A Markov chain Monte Carlo sampler for gene genealogies conditional on haplotype data

K.M. BURKETT, B. McNENEY and J. GRAHAM*

Statistics and Actuarial Science, Simon Fraser University,
8888 University Drive
Burnaby, BC, V5A 1S6, Canada
**E-mail: jgraham@sfu.ca*

The gene genealogy is a tree describing the ancestral relationships among genes sampled from unrelated individuals. Knowledge of the tree is useful for inference of population-genetic parameters such as migration or recombination rates. It also has potential application in gene-mapping, as individuals with similar trait values will tend to be more closely related genetically at the location of a trait-influencing mutation. One way to incorporate genealogical trees in genetic applications is to sample them conditional on observed genetic data. We have implemented a Markov chain Monte Carlo based genealogy sampler that conditions on observed haplotype data. Our implementation is based on an algorithm sketched by Zöllner and Pritchard but with several differences described herein. We also provide insights from our interpretation of their description that were necessary for efficient implementation. Our sampler can be used to summarize the distribution of tree-based association statistics, such as case-clustering measures.

Keywords: Gene genealogy; coalescent model; Markov chain Monte Carlo; genetic association studies; population genetics

1. Introduction

The variation observed in the human genome is a result of stochastic evolutionary processes such as mutation and recombination acting over time. The gene genealogy for a sample of copies of a locus from unrelated individuals is a tree describing these ancestral events and relationships connecting the copies. Knowledge of the tree is useful for inference of population-genetic parameters and it also has potential application in gene-mapping. However, the time scale for genealogical trees is on the order of tens of thousands of years, and there is therefore no way to know the true underlying tree for a random sample of genes from a population.

One way to incorporate genealogical trees in genetic applications is to sample them conditional on observed genetic data, for example using Markov chain Monte Carlo (MCMC) techniques. In population genetics, MCMC-based genealogy samplers have been implemented in order to estimate, for example, effective population sizes, migration rates and recombination rates. These approaches are reviewed in Ref. 1 and some software implementations are reviewed in Ref. 2. MCMC techniques have also been used in gene mapping methodology. In particular, Zöllner and Pritchard[3] implemented a coalescent-based mapping method in a program called LATAG (Local Approximation To the Ancestral Recombination Graph) that uses MCMC to sample genealogical trees. The LATAG approach involves sampling ancestral trees at a single focal point within a genomic region, rather than sampling the full ancestral recombination graph (ARG)[4,5] representation of the ancestral tree at *all* loci across the region. Focusing on a lower-dimensional latent variable, the genealogical tree, enables the LATAG approach to handle larger data sets than those that sample the ARG.[3]

In this paper, we describe our implementation of a genealogy sampler to sample ancestral trees at a locus conditional on observed haplotype data for surrounding SNPs. Since commonly-used genotyping technology does not provide haplotype information, haplotypes will typically be imputed from SNP genotype data. Our implementation is based on the genealogy sampler outlined by Zöllner and Pritchard,[3] herein called the ZP algorithm. As we were not interested in mapping *per se*, but rather in developing a stand-alone haplotype-based genealogy sampler that we could later extend to handle genotype data, we used their algorithm as a guide for developing our own haplotype-based sampler. During implementation, we filled in some of the details omitted from Zöllner and Pritchard and made some modifications to the algorithm. We therefore provide a brief background on the ZP algorithm, describe some of the missing details, and highlight where our implementation differs. Although not described here, our sampler can be used to summarize the distribution of tree-based association statistics, such as case-clustering measures.

2. Notation

The observed data, $\mathbf{H}$, is a vector of size n consisting of a sample of haplotypes from unrelated individuals. Each haplotype, $\mathbf{s}_i$, is a sequence composed of the alleles at L SNP markers. The allele at the j^{th} marker locus on the i^{th} haplotype is denoted $s_{i,j}$ and can take one of two values, 0 or 1.

Due to recombination, the genealogy across a genomic region is repre-

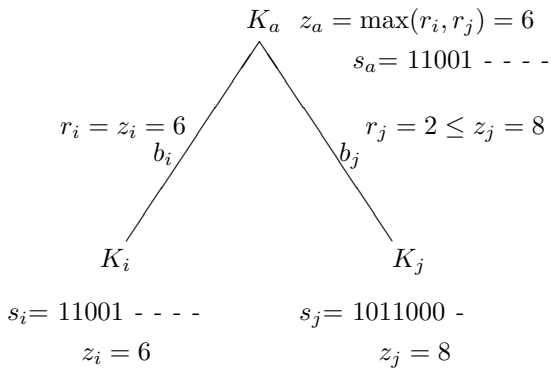

Fig. 1. Illustration of the definitions of the augmented variables on three nodes of τ_x. Each sequence is composed of $z_i - 1$ markers with alleles labelled 0 or 1. The focal point, x, is assumed to be to the left of the markers. Markers after recombination break points are labelled '-' and are not tracked because they are not co-inherited with the focal point.

sented by the ARG. However, the genealogy of a single locus is represented by a tree. Rather than sample from the ARG, the ZP algorithm and our implementation both approximate the ARG by sampling marginal trees at single focal points, x. The genealogy then consists of a tree topology τ_x and the node times $\mathbf{T}$. The nodes of the tree are labelled K_i and the internal branches of the tree have length b_i. There are n tip nodes, one for each sampled haplotype, $\mathbf{s}_i$. Each of the $n-1$ internal nodes corresponds to a coalescence event where two branches merge at a common ancestor. The root of the tree is the most recent common ancestor (MRCA) of the haplotypes in the sample. We are interested in sampling genealogies $(\tau_x, \mathbf{T})$ from the distribution $f(\tau_x, \mathbf{T}|\mathbf{H})$.

3. Overview of the ZP algorithm

We now give a brief overview of the ZP algorithm in order to better compare their algorithm with our approach in later sections. We first provide details of the target distribution, $f(\tau_x, \mathbf{T}|\mathbf{H})$, for the genealogies of the focal point x conditional on the haplotype data. We then describe the MCMC approach to sample from this distribution. For more information about the ZP algorithm and the corresponding gene-mapping approach, see Ref. 3.

3.1. *A distribution for* $(\tau_x, \mathbf{T})$ *conditional on haplotypes* $\mathbf{H}$

In the ZP algorithm, the distribution for $(\tau_x, \mathbf{T})$, conditional on $\mathbf{H}$ is modeled by augmenting $(\tau_x, \mathbf{T})$ with parameters for the mutation rate θ and the recombination rate ρ, and latent variables for the unobserved sequence states at nodes of the tree. These unobserved states consist of the haplotypes at the internal node, $\mathbf{S} = (s_{n+1}, s_{n+2}, \ldots, s_{2n-1})$. However, only the sequence that is passed to the present along with the focal point x is stored, which requires also storing the location of the closest recombination events to the focal point x on either side of x. The recombination processes on either side of x are assumed to be independent, which simplifies the description since only variables and the model corresponding to the right-hand process need to be described. The variables and model for recombination events to the left of the focal point are all defined similarly to the right-hand versions. The recombination-related variables corresponding to the right-hand recombination process are denoted $\mathbf{R} = (r_1, r_2, \ldots, r_n, r_{n+1}, \ldots, r_{2n-1})$. More information about $\mathbf{R}$ is given below and an illustration of the definitions of $\mathbf{s}$ and r for the individual nodes is given in Fig. 1. Let $\mathbf{A} = (\tau_x, \mathbf{T}, \theta, \rho, \mathbf{S}, \mathbf{R})$.

With the addition of the latent variables, samples from $\mathbf{A}$ are now desired from target distribution $f(\mathbf{A}|\mathbf{H})$. This distribution is given by

$$f(\mathbf{A}|\mathbf{H}) \propto \Pr(\mathbf{S}|\mathbf{R}, \mathbf{T}, \tau_x, \theta, \gamma) \Pr(\mathbf{R}|\mathbf{T}, \tau_x, \rho) h(\tau_x, \mathbf{T}) h(\theta) h(\rho). \quad (1)$$

The prior distribution for the topology and node times, $h(\tau_x, \mathbf{T})$, is the standard neutral coalescent model.[6,7] The prior distributions for the mutation and recombination rates, $h(\theta)$ and $h(\rho)$, are assumed to be uniform. The terms $\Pr(\mathbf{R}|\mathbf{T}, \tau_x, \rho)$ and $\Pr(\mathbf{S}|\mathbf{R}, \mathbf{T}, \tau_x, \theta, \gamma)$ are given below.

At each node, the sequence at a locus is stored only if that locus is co-inherited with the focal point. The sequence at the locus and the focal point will not be co-inherited when there is a recombination event between them. For node K_i, consider the maximum extent of sequence it leaves in at least one present descendant, and let z_i denote the index of the locus just beyond. Next, consider the maximum extent of sequence *from* K_i*'s parent node* that K_i leaves in at least one present descendant, and let r_i be the index of the locus just beyond. In Fig. 1, nodes K_i and K_j are siblings with parent node K_a. From the definitions of z and r it follows that $z_a = \max(r_i, r_j)$. Moreover, r_a depends on the r values of all K_a's descendants only through the r values of K_a's two children or, more specifically, their maximum z_a.

By noting that the r value for a node depends on the r values of all its

descendants through z, $\Pr(\mathbf{R}|\mathbf{T}, \tau_x, \rho)$ can be written as:

$$\Pr(\mathbf{R}|\mathbf{T}, \tau_x, \rho) = \prod_{i=1}^{2n-2} \Pr(r_i \mid z_i, b_i, \rho). \quad (2)$$

Recombination events along the branches of the tree and across the L loci are assumed to have a Poisson distribution with rate $\rho/2$ per unit of coalescence time. For each node then,

$$\Pr(r_i = c|z_i, b_i, \rho) = \begin{cases} 0 & c > z_i \\ \int_{d_{c-1}}^{d_c} \frac{b_i\rho}{2} \exp(-\frac{b_i\rho}{2}t)dt & 1 < c < z_i \\ \int_{d_{c-1}}^{\infty} \frac{b_i\rho}{2} \exp(-\frac{b_i\rho}{2}t)dt & c = z_i \end{cases}, \quad (3)$$

where the vector d consists of the locations of the L SNP markers.

The term $\Pr(\mathbf{S}|\mathbf{R}, \mathbf{T}, \tau_x, \theta, \gamma)$ is written by successively conditioning on parents:

$$\Pr(\mathbf{S}|\mathbf{R}, \mathbf{T}, \tau_x, \theta, \gamma) = \prod_{i=1}^{2n-1} \Pr(\mathbf{s}_i|\mathbf{s}_a, r_i, z_i, b_i, \theta, \gamma), \quad (4)$$

where K_a generically denotes the parent of K_i. We only track sequence that is co-inherited with the focal point, which consists of the sequence before the z_i^{th} marker. The first $r_i - 1$ loci have been inherited from K_a and therefore these loci are modeled by conditioning on the parental alleles. If the parental allele at a locus is different from its offspring's allele at the same locus, a mutation event has occurred on the branch between them. Mutation events at each locus along the branches of the tree are assumed to be Poisson distributed with rate $\theta/2$ per unit of coalescence time. Given that a mutation has occurred, the type of the new allele is chosen randomly from the two allelic types. Therefore, the probability of the $j^{th} < r_i$ allele is given by

$$\Pr(s_{i,j} = a_1|s_{a,j} = a_2, b_i, \theta) = \begin{cases} \frac{1}{2}(1 - e^{-\theta b_i/2}) & \text{if } a_1 \neq a_2 \\ \frac{1}{2}(1 - e^{-\theta b_i/2}) + e^{-\theta b_i/2} & \text{if } a_1 = a_2 \end{cases}, \quad (5)$$

where a_1 and a_2 are the allelic types at the j^{th} marker. Loci from r_i to $z_i - 1$ have recombined in from an unknown ancestor. For these markers, Ref. 3 assumed a first-order Markov model where the allele at a locus has a Bernoulli distribution with probability that depends on the allele at the previous locus. These probabilities, denoted γ, are estimated from the observed data. Therefore, for the i^{th} node, each term in Eqn. (4) is

$$\Pr(\mathbf{s}_i|\mathbf{s}_a, r_i, z_i, b_i, \theta, \gamma) = \left[\prod_{j=1}^{r_i-1} \Pr(s_{i,j}|s_{a,j}, b_i, \theta)\right] \Pr(h_{r_i \to z_i - 1}|\gamma), \quad (6)$$

where $h_{r_i \to z_i - 1}$ denotes the haplotype of the sequence between the r_i^{th} and $z_i - 1^{th}$ locus and $\Pr(s_{i,j}|s_{a,j}, b_i, \theta)$ is given in Eqn. (5).

3.2. *Overview of the proposal distributions used in the ZP algorithm*

MCMC is used to sample from the target distribution, $f(\mathbf{A}|\mathbf{H})$. A new value for $\mathbf{A}$, $\tilde{\mathbf{A}}$, is proposed from distribution $Q(\tilde{\mathbf{A}}|\mathbf{A})$. This value is then accepted or rejected according to the Metropolis-Hastings acceptance probability

$$\alpha(\mathbf{A}, \tilde{\mathbf{A}}) = \min\left\{1, \frac{f(\tilde{\mathbf{A}}|\mathbf{H})Q(\mathbf{A}|\tilde{\mathbf{A}})}{f(\mathbf{A}|\mathbf{H})Q(\tilde{\mathbf{A}}|\mathbf{A})}\right\}. \tag{7}$$

The ZP algorithm uses six update schemes to propose new values for $\mathbf{A}$. Each of the six schemes proposes new values for only a subset of the components of $\mathbf{A}$. The six update schemes are: (1) update θ; (2) update ρ; (3) local update of internal nodes; (4) major topology change; (5) minor topology change and (6) reorder coalescence events. These update schemes are illustrated in Fig. 2 and are summarized below.

(1) **Update θ:** Sample $\tilde{\theta}$ from a uniform distribution on $(\theta^{(t)}/2, 2\theta^{(t)})$, where $\theta^{(t)}$ is the value of θ at the t^{th} iteration.

(2) **Update ρ:** Sample $\tilde{\rho}$ from a uniform distribution on $(\rho^{(t)}/2, 2\rho^{(t)})$, where $\rho^{(t)}$ is the value of ρ at the t^{th} iteration.

(3) **Local updates of internal nodes:** For each node, starting at the tip nodes and ending at the MRCA, sample a new time since the present, $\tilde{t}_i$, recombination variable, $\tilde{r}_i$, and new sequence, $\tilde{\mathbf{s}}_i$. The proposal distribution for sampling each component was not provided.

(4) **Major topology change:** Randomly select a node to be moved, K_{c_1} in Fig. 2(A), from the set of nodes that can be moved without causing incompatible r values among neighbouring nodes. Referring to Fig. 2(C), after the topology change, loci 1 through $z_p - 1$ must pass to the present via either K_{c_2} or K_a. An incompatibility can occur if $z_p > \max(r_a, z_{c_2})$. A new sibling, K_s, is selected from the set of nodes having parents older than t_{c_1} and based on sequence similarity to $\mathbf{s}_{c_1}$. The topology change is made, as illustrated in Fig. 2(C), a new time for the parent $\tilde{K}_i$ is sampled from a uniform distribution with bounds $(\max(t_{c_1}, t_s), t_m)$, new r values are sampled for nodes K_{c_1}, K_{c_2}, K_s and $\tilde{K}_i$, and a new sequence is sampled for $\tilde{K}_i$.

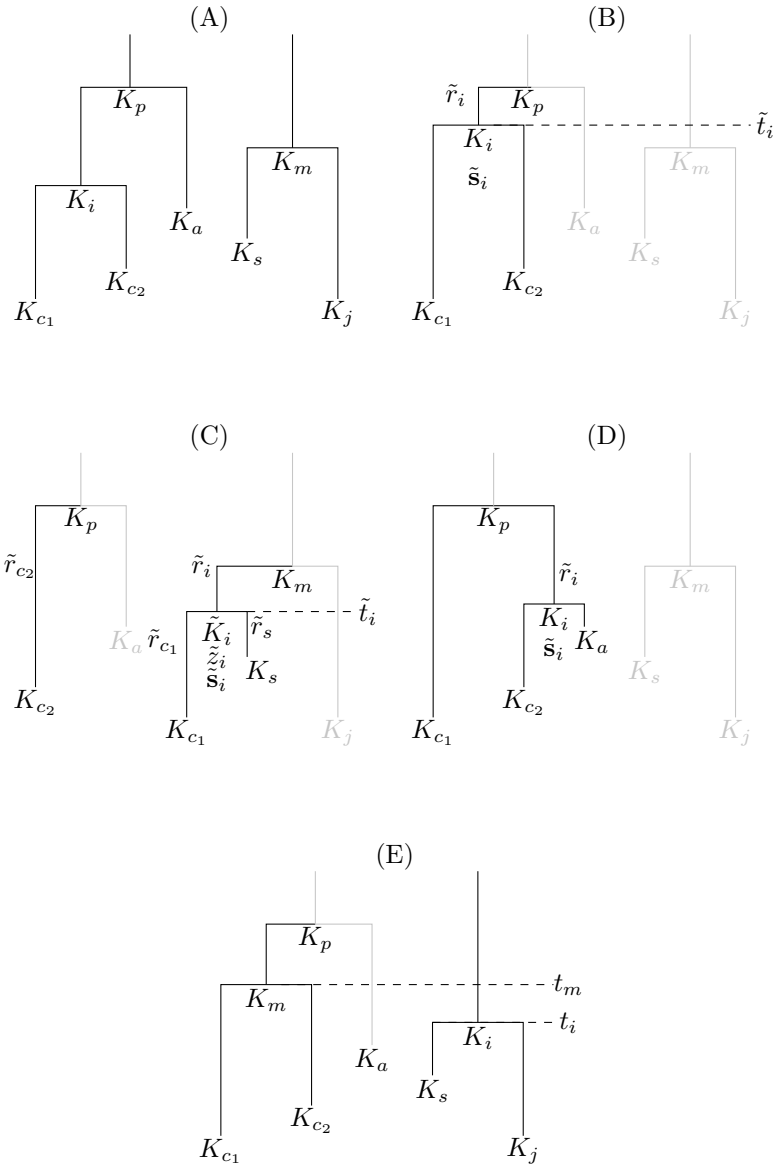

Fig. 2. Illustration of the update schemes for the ZP algorithm. (A) Example of two branches of the tree before any updates; (B) After local update to node K_i; (C) After a major topology change to node K_{c_1}; (D) After a minor topology change to node K_{c_1}; (E) After reordering coalescence events for nodes K_i and K_m. To highlight the nodes and branches modified during each update, those that remain the same are greyed out.

(5) **Minor topology change:** An internal node, K_{c_1} in Fig. 2(A), is first sampled. In the new topology its old sibling, K_{c_2}, and old aunt, K_a, become its new nieces. After the topology change, illustrated in Fig. 2(D), r and $\mathbf{s}$ value are sampled for the new parent node $\tilde{K}_i$.

(6) **Reorder coalescence events:** The order of coalescence events is modified by swapping the times for two nodes, K_i and K_m, as shown in Fig. 2(E). The two nodes are eligible to swap node times provided that, after the swap of times, K_i and K_j are not older than their respective parents or younger than their respective children.

At each iteration of the chain, combinations of these update schemes are used to propose new values for $\mathbf{A}$; for example, at the t^{th} iteration a major topology change and local updates might be used to propose $\tilde{\mathbf{A}}$. Information about the frequency with which each combination of updates is selected is not given, but it is stated that the ZP algorithm samples each combination with a probability that depends on the nature of the dataset.

4. Description of our genealogy sampler

In this section, we describe our haplotype-based genealogy sampler. As mentioned, we based our approach on that of Ref. 3 and therefore many of the details given in Sec. 3 apply equally to our sampler. In the description that follows, we therefore highlight the differences between our approach and that of the ZP algorithm. A more comprehensive description of our sampler can also be found in Ref. 8.

4.1. *The target distribution:* $f(\mathbf{A}|\mathbf{H})$

The target distribution is modeled as in Sec. 3.1, but we use an alternate prior distribution for the recombination rate, ρ. We do not assume a uniform prior distribution for ρ as there is some evidence that the distribution of recombination rates is roughly exponential in parts of the genome.[9] We therefore assume a gamma/exponential prior distribution for ρ.

4.2. *Proposal distributions for* t_i, r_i *and* $\mathbf{s}_i$

The local updates and the topology rearrangements sample new t, r, and $\mathbf{s}$ values for some nodes of the tree; however, the proposal distributions for sampling these new values were not given in Ref. 3. We therefore now provide information on our proposal distributions for these components. For a node K_i, our proposal distributions for t_i, r_i and $\mathbf{s}_i$ are all motivated

by the decomposition of $f(\mathbf{A}|\mathbf{H})$ given in Eqn. (1). That is, t_i is sampled conditional on other t values from an approximation to $h(t_i|T_{-i})$, r_i from an approximation to $\Pr(r_i|\mathbf{R}_{-i}, \mathbf{T}, \tau_x, \rho)$ and $\mathbf{s}_i$ from $\Pr(\mathbf{s}_i|\mathbf{S}_{-i}, \mathbf{R}, \mathbf{T}, \tau_x, \theta, \gamma)$.

Proposal distribution for t_i

The proposal distribution for t_i is motivated by the distribution $h(t_i|\mathbf{T}_{-i})$; however, rather than condition on all elements of $\mathbf{T}_{-i}$ in the proposal distribution, we condition only on the times of the nodes adjacent to K_i. Letting K_{c_1} and K_{c_2} be K_i's two children, and K_a be K_i's parent (see Fig. 2(A) for the node labels), the proposal distribution for t_i is

$$q(t_i|t_{c_1}, t_{c_2}, t_a) = \begin{cases} \exp(-(t_i - t_{2n-2}))I[t_i > t_{2n-2}] & i = 2n-1 \\ U(\max(t_{c_1}, t_{c_2}), t_a) & i \neq 2n-1 \end{cases}. \quad (8)$$

The motivation for this proposal distribution comes from the coalescent model, which assumes exponential inter-coalescence times with rate $\binom{j}{2}$ when there are j lineages left to coalesce. If K_i is the MRCA, *i.e.* $i = 2n-1$, $j = 2$ and the waiting time until the last two lineages merge, $t_i - t_{i-1} = t_{2n-1} - t_{2n-2}$, has an exp(1) distribution. On the other hand, if K_i is not the MRCA, the distribution of t_i conditional on t_{i-1} and t_{i+1} is uniform on (t_{i-1}, t_{i+1}). Rather than condition on the times of the adjacent coalescence events, *i.e.* t_{i-1} and t_{i+1}, we condition on the times of the adjacent nodes, t_{c_1}, t_{c_2} and t_a. Therefore, our uniform proposal distribution only approximates the true conditional distribution. However, this proposal distribution does allow for a re-ordering of coalescence events since the proposal distribution for t_i is not necessarily constrained by t_{i-1} and t_{i+1}.

Proposal distribution for r_i

Recall that the r variables are defined recursively so that the value of r_i depends on the r values of all of its descendants through z_i. In the description of the model for the r values given in Ref. 3 (page 1090), they explicitly let the probability of r_i, given in Eqn. (3) here, be 0 if the value is incompatible with r values of ancestral nodes. Therefore, the ZP algorithm seems to allow an $\tilde{r}_i$ to be sampled that is incompatible with other r values on the tree; such $\tilde{r}_i$ values would be immediately rejected as they are defined to have 0 probability under the target distribution.

In our updates to the r_i variables, rather than propose values that would be rejected due to incompatibility, we restrict the support of the proposed r_i to be compatible with all other r variables on the tree. The proposed value for r_i is constrained by the length of sequence that is passed to the present through node K_i; that is, by z_i. The proposed value may also be

constrained by r_a, the r value of an ancestor K_a, if the sequence material passes to the present through K_i. Therefore, the support for the proposal distribution for r_i, $\mathcal{S}(\mathbf{R}_{-i})$, depends on restrictions imposed by the r and z values of nodes in the vicinity of node K_i. The actual nodes that restrict the support depend on the update type and therefore more information about the restrictions is provided in Sec. 4.3.

With the support $\mathcal{S}(\mathbf{R}_{-i})$ giving the set of values for r_i that are compatible with the data at surrounding nodes, the conditional distribution for component r_i is

$$\Pr(r_i|\mathcal{S}(\mathbf{R}_{-i}), \mathbf{T}, \tau_x, \rho) = \frac{\Pr(r_i|z_i, b_i, \rho)}{\sum_{r^* \in \mathcal{S}(\mathbf{R}_i)} \Pr(r^*|z_i, b_i, \rho)} 1[r_i \in \mathcal{S}_i(\mathbf{R}_{-i})].$$

We thus take as our proposal distribution for r_i

$$q(r_i|\mathcal{S}(\mathbf{R}_{-i}), \mathbf{T}, \tau_x, \rho) = \Pr(r_i|\mathcal{S}(\mathbf{R}_{-i}), \mathbf{T}, \tau_x, \rho), \tag{9}$$

where $\Pr(r_i|z_i, b_i, \rho)$ is given in Eqn. (3).

Proposal distributions for $\mathbf{s}_i$

For our proposal distribution for component $\mathbf{s}_i$, recall that $\mathbf{s}_i$ is a vector, with L elements corresponding to the alleles at each locus. We therefore sample a new sequence, $\tilde{\mathbf{s}}_i$, by sampling a new allele at each locus $s_{i,j}$ starting at the first locus, $j = 1$, and finishing at $j = z_i - 1$. Since we are only sampling alleles at loci that are passed to at least one descendant at present in the tree, all alleles at markers z_i and above are given the value '-' with probability one, as shown in Fig. 1.

To motivate a proposal distribution for $s_{i,j}$, the allele at the j^{th} locus, we use information available from the allele at the $j-1^{th}$ locus, $s_{i,j-1}$, and the alleles at the j^{th} locus in the sequences of adjacent nodes. If the j^{th} locus was inherited from parent K_a, *i.e.* if $j < r_i$, then $s_{i,j}$ will be different from $s_{a,j}$ if a mutation occurred on the branch b_i. The probability of mutation events on branches of the tree was given in Eqn. (5). If this locus was not inherited from K_a, *i.e.* if $j \geq r_i$, then, by the first-order Markov haplotype model referred to in Eqn. (6), the probability of the allele depends on the allele at the $j-1^{th}$ locus. Similar arguments can be made for the inheritance of this locus between K_i and its children, K_{c_1} and K_{c_2}.

Our proposal distribution for $s_{i,j}$ is motivated by combining the inheritance of the j^{th} locus from K_a through nodes K_i, K_{c_1} and K_{c_2}. Let the

proposal distribution for the j^{th} locus be

$$\begin{aligned}
&q(s_{i,j}|s_{i,j-1}, s_{c_1,j}, s_{c_2,j}, s_{a,j}, \mathbf{R}, \mathbf{T}, \tau_x, \theta, \gamma) \\
&= \Pr(s_{i,j}|s_{i,j-1}, s_{c_1,j}, s_{c_2,j}, s_{a,j}, \mathbf{R}, \mathbf{T}, \tau_x, \theta, \gamma) \\
&\propto \Pr(s_{i,j}, s_{c_1,j}, s_{c_2,j}|s_{i,j-1}, s_{a,j}, \mathbf{R}, \mathbf{T}, \tau_x, \theta, \gamma) \\
&= \Pr(s_{i,j}|s_{i,j-1}, s_{a,j}, \mathbf{R}, \mathbf{T}, \tau_x, \theta, \gamma) \times \Pr(s_{c_1,j}|s_{i,j}, \mathbf{R}, \mathbf{T}, \tau_x, \theta, \gamma) \times \\
&\qquad \Pr(s_{c_2,j}|s_{i,j}, \mathbf{R}, \mathbf{T}, \tau_x, \theta, \gamma). \qquad (10)
\end{aligned}$$

The last line follows from the conditional independence of $s_{c_1,j}$ and $s_{c_2,j}$ given $s_{i,j}$, their parent's allele at the j^{th} locus.

If K_i is not the MRCA ($i \neq 2n-1$), the first term in Eqn. (10) is

$$\Pr(s_{i,j}|s_{i,j-1}, s_{a,j}, \mathbf{R}, \mathbf{T}, \theta, \gamma) = \begin{cases} \Pr(s_{i,j}|s_{a,j}, b_i, \theta) & j < r_i \\ \Pr(s_{i,j}|\gamma) & j = r_i \\ \Pr(s_{i,j}|s_{i,j-1}, \gamma) & r_i < j < z_i - 1 \end{cases},$$

with $\Pr(s_{i,j}|s_{a,j}, b_i, \theta)$ given in Eqn. (5), and $\Pr(s_{i,j}|\gamma)$ and $\Pr(s_{i,j}|s_{i,j-1}, \gamma)$ given by the first-order Markov haplotype model referred to in Eqn. (6). If K_i is the MRCA, then K_i does not have a parent in the tree and therefore the probability of the j^{th} locus is based on the haplotype model:

$$\Pr(s_{i,j}|s_{i,j-1}, \mathbf{R}, \mathbf{T}, \theta, \gamma) = \begin{cases} \Pr(s_{2n-1,j}|\gamma) & j = 1 \\ \Pr(s_{2n-1,j}|s_{2n-1,j-1}\gamma) & 1 < j \leq z_i - 1 \end{cases}.$$

The second and third terms of Eqn. (10) are modeled similarly. However, if K_i does not pass the j^{th} locus to K_{c_1} and/or K_{c_2} then these nodes provide no information about $s_{i,j}$. Hence the probabilities $\Pr(s_{c_1,j}|s_{i,j}, \mathbf{R}, \mathbf{T}, \tau_x, \theta)$ and $\Pr(s_{c_2,j}|s_{i,j}, \mathbf{R}, \mathbf{T}, \tau_x, \theta)$ are constant with respect to $s_{i,j}$ and so can be dropped. It follows that the second and third terms are:

$$\Pr(s_{c_1,j}|s_{i,j}, \mathbf{R}, \mathbf{T}, \theta) = \begin{cases} \Pr(s_{c_1,j}|s_{i,j}, b_{c_1}, \theta) & j < r_{c_1} \\ 1 & j \geq r_{c_1} \end{cases} \text{ and}$$

$$\Pr(s_{c_2,j}|s_{i,j}, \mathbf{R}, \mathbf{T}, \theta) = \begin{cases} \Pr(s_{c_2,j}|s_{i,j}, b_{c_2}, \theta) & j < r_{c_2} \\ 1 & j \geq r_{c_2} \end{cases}.$$

To summarize, the proposal distribution for $s_{i,j}$, given in Eqn. (10), is proportional to the product of the probabilities for the inheritance of the j^{th} locus from its parent K_a to its children K_{c_1} and K_{c_2}. The proposal probability for the full sequence, $\tilde{\mathbf{s}}_i$, is proportional to the product of the proposal probabilities for each allele at each locus.

4.3. *Update schemes for our sampler*

As mentioned, we used the outline given for the six update schemes, which are summarized in Sec. 3.2, as a starting point for our sampler. During

implementation, we made some changes to the updates schemes. First, we generally aimed to increase sampling efficiency by not proposing values that would result in the update being automatically rejected due to incompatibility with the unmodified values. For example, referring to Fig. 2(A), this can occur if $\tilde{r}_i$ is lower than the r value for K_i's parent K_a. An incompatibility can also occur with the sequences, $\mathbf{s}$, due to an update to the r's. For example, if $\tilde{r}_i$ leads to $\tilde{z}_a > z_a$ and $\mathbf{s}_a$ is not also updated, then the current value for $\mathbf{s}_a$ will not be possible since it will not have valid alleles at markers z_a to $\tilde{z}_a - 1$. Second, our algorithm does not use the sixth update, which reorders coalescence events by swapping node times, since the proposal distribution we use to update the t_i allows for a reordering of coalescence events. Finally, by default, at each iteration of the chain, only one of the five updates is used to propose new $\mathbf{A}$ rather than a combinations of the six; however, in our implementation the user can choose to perform combination of updates consecutively. Our modifications to the update schemes are now described.

(1) **Updating θ:** Our proposal distribution for θ is also uniform. However, the range of the uniform is set so that θ outside of the prior distribution, which would be rejected, can not be sampled.

(2) **Local updates:** In the local updates of the ZP algorithm, new values for t_i, r_i and $\mathbf{s}_i$ are proposed for each node, starting at the tip nodes and moving to the MRCA. Our local updates proceed from the first internal node to the MRCA and we choose to accept or reject the changes at each node before moving to the next node. At each internal node, K_i, we propose a new time, t_i, from the proposal distribution given in Sec. 4.2. We then propose new r_{c_1} and r_{c_2}, for the children K_{c_1} and K_{c_2} rather than for K_i, from proposal distribution given in the Sec. 4.2. We restrict the support of $\tilde{r}_{c_1}$ and $\tilde{r}_{c_2}$ so that the value of r_i is not incompatible with $\tilde{r}_{c_1}$ and $\tilde{r}_{c_2}$ (*i.e.* one of $\tilde{r}_{c_1}$ and $\tilde{r}_{c_2}$ must be greater than or equal to r_i). Finally, we propose new $\mathbf{s}_i$ from the proposal distribution given in Sec. 4.2. Our modifications to the local updates ensure that (1) the sampled r values are not immediately rejected due to incompatibility with other r values and (2) an update to $\mathbf{s}_a$ is not required (if r_i were updated, then by definition z_a might also change).

(3) **Major topology change:** We made several changes to the major topology rearrangement:

   (a) The similarity score used in the ZP algorithm was not provided. Our

similarity score between two sequences, $\mathbf{s}_i$ and $\mathbf{s}_c$, is

$$p_{ic} = \begin{cases} \frac{\sum_{j=1}^{\min(z_i-1,z_c-1)} 1(s_{i,j}=s_{c,j})}{\min(z_i-1,z_c-1)} & K_c \text{ is eligible to coalesce with } K_i \\ 0 & \text{otherwise} \end{cases}.$$

This score counts the alleles that $\mathbf{s}_i$ and $\mathbf{s}_c$ share in common. Ideally, we would only like to compare loci that both sequences inherited from their shared parent. However, when selecting the new sibling, we don't have this information as $\tilde{r}_i$ and $\tilde{r}_c$ haven't yet been sampled. We therefore only compare the loci that both would have inherited from their parent if no recombination events occur on the branches $\tilde{b}_i$ and $\tilde{b}_c$. This sequence is between locus 1 and $\min(z_i - 1, z_c - 1)$.

(b) We also exclude nodes in selecting K_s in order to avoid creating non-sensical trees and to differentiate the major from the minor topology change; the excluded nodes are the current sibling, parent, aunt, niece or grandparent of K_{c_1}.

(c) As with the local updates, we have structured our updates to the r variables so that the proposed values do not cause incompatibilities with unchanged values on the tree. Referring to Fig. 2(A) and (C) for the notation, this is done by restricting the support of $\tilde{r}_{c_2}$ so that $\tilde{z}_p = z_p$ and the supports of $\tilde{r}_s$ and $\tilde{r}_i$ so that $\tilde{z}_m = z_m$.

(4) **Minor topology change:** Our implementation of the minor topology change also differs substantially:

(a) Referring to Fig. 2(A) and (D), we impose the condition that $t_i > t_a$, or K_i must be older than its sibling K_a, to ensure that the topology change produces a valid tree. Since t_i and t_a are the same after the topology change, without this condition the parent $\tilde{K}_i$ could be younger than its child K_a and the update would be automatically rejected.

(b) The ZP algorithm imposed a constraint that K_{c_1} be an internal node. However, if tip nodes are not eligible for the topology change, the reverse rearrangement of $\tilde{\mathbf{A}}$ to $\mathbf{A}$ could have zero probability, the acceptance probability would be 0, and the update would automatically be rejected. We therefore remove this restriction.

(c) Even though the time of node $\tilde{K}_i$ is the same in $\tilde{\tau}_x$ as that of K_i in τ_x, the branch lengths b_{c_1} and b_a are not the same. For this reason, r_{c_1} and r_a could be unlikely given the new branch lengths. We therefore also sample new r_{c_1} and r_a.

(d) As with the major topology change and the local updates, we sample r values in such a way that compatibility with other r values is

ensured. For this update, this is achieved if we require that $\tilde{z}_p = z_p$. This results in a restricted support so that one of $\tilde{r}_{c_1}$ or $\tilde{r}_a$ is greater than or equal to z_p.

4.4. *Software and application*

The sampler was coded in C++ and currently runs at the command line; however, we plan to import the C++ code into `R`. Input options are provided to the program in a file and include the file names for the relevant datasets, a run name, chain length, burn-in, thinning, as well as prior parameters and initial values for the latent variables. Most options include sensible defaults. Output includes the scalar-valued output (the update performed, an acceptance indicator and the ρ and θ values) and the sampled trees.

The sampler was applied to haplotype data imputed from the Crohn's dataset[10] available in the `R gap` package. Each dataset consisted of 516 haplotypes of between 20 and 35 SNPs and was run on a separate 2.67 GHz core of a cluster computer. Eight million iterations, which took approximately three weeks, were required to achieve convergence. Due to space constraints, we refer the reader to Section 2.7 of Ref. 8 for the full results, including convergence diagnostics.

5. Discussion

In this paper, we have given a summary of our haplotype-based sampler. Our sampler is based on the approach outlined in Ref. 3 but we have provided some important details related to our implementation. In particular, all our proposal distributions for the recombination variable r lead to proposed values that do not cause incompatibilities with other variables on the tree and hence have non-zero probabilities under the target distribution. In contrast, the ZP algorithm can propose r values that are incompatible with ancestral nodes since they explicitly define the probability of such events to be zero under the target distribution. This would be inefficient as it would lead to the proposal being rejected. A related point is that we structured the local updates so that sequence at an ancestral node doesn't become incompatible with the proposed values. In further efforts to improve efficiency, we also modified the proposal distribution for the minor topology rearrangement so that transitions from $\tilde{\mathbf{A}}$ back to $\mathbf{A}$ are always possible. We have also eliminated the need for a separate update to reorder coalescence events. Finally, by default, we do not perform sets of updates in a sequential order, for example a major topology rearrangement followed

by the local updates; we have found that performance of the sampler is unaffected by this simplification.

Our sampler returns the tree and node times in Newick format. The trees can be read into existing software packages for graphical display and analysis. Any tree-based statistic that relies on the topology and node times, such as the case-clustering measure from Ref. 11, can be computed from a sampled tree. The distribution of the statistic can be summarized by computing the statistic on the MCMC sample of trees. More information and an example of this approach is provided in Ref. 8.

A drawback to using a genealogical sampler that conditions on haplotype data is that with SNP data the haplotypes are often unknown. Therefore, the haplotypes are also latent data in this context. The suggestion in Ref. 3 was to first use a program to statistically impute phase. However, since we are sampling trees based on a single imputation of the haplotypes, the imputation limits the space of trees that we sample from which could subsequently bias estimates. We are therefore working on an extension to the sampler to handle missing phase.

Acknowledgments

This work was supported by the Natural Sciences and Engineering Research Council of Canada and the Mathematics of Information Technology and Complex Systems Networks of Centres of Excellence. Portions of this work were undertaken while KB held a Canadian Institutes of Health Research Doctoral Research Award and a Michael Smith Foundation for Health Research (MSFHR) Senior Graduate Trainee Award, and while JG was a MSFHR Scholar.

References

1. M. Stephens, Inference under the coalescent, in *Handbook of Statistical Genetics*, eds. D. J. Balding, M. Bishop and C. Cannings (John Wiley and Sons, 2003) pp. 636–661, second edn.
2. M. K. Kuhner, *Trends in Ecology & Evolution* **24**, 86 (2009).
3. S. Zöllner and J. K. Pritchard, *Genetics* **169**, 1071 (2005).
4. R. C. Griffiths and P. Marjoram, *Journal of Computational Biology* **3**, 479 (1996).
5. R. C. Griffiths and P. Marjoram, An ancestral recombination graph, in *Progress in Population Genetics and Human Evolution*, eds. P. Donnelly and S. Tavare (Springer-Verlag, 1997) pp. 257–270.
6. J. F. C. Kingman, *Stochastic Processes and their Applications* **13**, 235 (1982).
7. R. R. Hudson, *Theoretical Population Biology* **23**, 183 (1983).

8. K. M. Burkett, Markov chain Monte Carlo sampling of gene genealogies conditional on observed genetic data, PhD thesis, Simon Fraser University, (Burnaby, Canada, 2011).
9. G. A. T. McVean, S. R. Myers, S. Hunt, P. Deloukas, D. R. Bentley and P. Donnelly, *Science* **304**, 581 (2004).
10. M. J. Daly, J. D. Rioux, S. F. Schaffner, T. J. Hudson and E. S. Lander, *Nature Genetics* **29**, 229 (2001).
11. M. J. Minichiello and R. Durbin, *American Journal of Human Genetics* **79**, 910 (2006).