Simulation of ancestral selection graphs for Monte Carlo integration

Nicoleen Cloete Geoff K. Nicholls* David J. Scott†

(Received 8 August 2003, revised 31 March 2004)

Abstract

An ancestral selection graph is a realization of an genealogy-process model which incorporates natural selection. The space of ancestral graphs is a countable union of spaces of unequal dimensions. We give a Markov Chain Monte Carlo algorithm simulating ancestral selection graphs. Output can be used to estimate expectations for functions defined on the space of ancestral graphs.

Contents

1 Introduction **C392**
1.1 Ancestral selection graphs C394
1.2 The space of ancestral selection graphs C396

*Department of Mathematics, Private Bag 92019, Auckland, New Zealand.
mailto:cloete@math.auckland.ac.nz
†Department of Statistics, University of Auckland, New Zealand.
See http://anziamj.austms.org.au/V45/CTAC2003/Cloe for this article, Published June 8, 2004. ISSN 1446-8735

2 Sampling ancestral selection graphs **C398**
2.1 Metropolis-Hastings algorithm C398
2.2 Posterior distribution C399
2.3 Moves C400
2.3.1 Add an edge C401
2.3.2 Delete an edge C403

3 Discussion **C403**

References **C404**

1 Introduction

We wish to compute the expected value $\mathsf{E}\{h(x)\}$ of some function h over the domain Ω of a probability density $f(x)$. Recall the basic Monte Carlo procedure. We draw N samples $x_i \sim f$, $i = 1, 2, \ldots, N$ and form an estimate $\bar{h} = \frac{1}{N}\sum_i h(x_i)$ of $\mathsf{E}\{h(x)\}$. The samples are distributed according to f, however they need not be independent. It is sometimes convenient to use a correlated sequence of states from a Markov chain with state space Ω and equilibrium f. If the Markov chain is geometrically ergodic, a central limit theorem applies, and $\bar{h}$ and its standard error provide a suitable estimate and uncertainty measure. This is the basis of the Markov chain Monte Carlo method for estimating expectations.

It is only since 1995 and the work of Green, that it has been feasible to draw samples $x \sim f$ from generic densities defined on spaces for which all states do not have equal dimension, so that the state vector has random dimension. We wish to compute expectations in a probability distribution defined over a certain class of graphs (ancestral selection graphs, see Figure 1 for an instance). The number of vertices varies from one graph to another. Because each vertex is carries a real scalar parameter, the state dimension

varies from one state to another. Densities of this kind arises in a class of stochastic models of ancestry with natural selection.

In this paper we will define the density and state space of interest, and show how it may be sampled using Markov chain Monte Carlo and the methods of Green [3]. This sampling scheme will be used in later work to estimate expectations for quantities of interest. The density we treat can be sampled in a much more straightforward way using the graph process by which it is defined. However, the framework we give extends to cases involving data. We expand on this point in Section 2.2. This is not the case for any direct implementation of the graph process itself. Other simulation methods have been given which are effective, in particular the importance sampling method due to Slade [7]. These do simulate the process conditioned on certain types of data, and will be more efficient than our MCMC methods for many cases of interest. However, there remain data types (such as viral DNA sequence data) for which MCMC simulation is the only straightforward option. These are cases in which there is no distribution 'close' to the posterior distribution which can feasibly be sampled by direct methods to yield iid samples.

A phylogenetic tree $T = (V, E)$ on a set $\mathcal{A} = \{\infty, \in, \ni \ldots, \backslash\}$ is a binary tree with vertices $V = V_L \cup V_A$. Here V_L is a set of n leaf vertices (each with a distinct label from $\mathcal{A}$) and V_A is the set of internal vertices. Associated with each vertex $v \in V$ is a function $B : V \mapsto \mathcal{S}$, where $\mathcal{S}$ is a countable state space and indicates a type at that vertex. If $B_u \neq B_v$, a change of type has occurred along the edge between vertex u and vertex v.

Kingman [4] defined a stochastic process, the n coalescent, to model ancestral relations for a group of organisms n evolving without natural selection in a larger background population of size N. Realizations of this process are phylogenetic trees with n leaves. We are interested in a related process, modelling two types of organisms $\mathcal{S} = \{0, 1\}$ evolving with natural selection. The selective advantage of type 1 individuals over type 0 is represented by supposing type 1 individuals have a higher birth rate. The offspring of an individual is of a different type to its parent with probability μ, independent

of the type of the parent. At a birth event offspring replace a randomly chosen member of the population.

This selection model can be graphically represented by the so-called biased-voter model. In the "diffusive limit", $N \to \infty$, the random biased-voter graph converges in distribution to a random graph process called the ancestral selection graph. A full description of the biased-voter model and associated dual process can be found in the paper by Neuhauser and Krone. [5]

In Section 1.1 we give a description of the ancestral selection graphs as introduced by Krone-Neuhauser [5]. A description of the space of ancestral selection graphs is given in Section 1.2. We go into some detail, because it is the nature of this space that warrants the novel Monte Carlo methods presented in Section 2. In Section 2.2 we develop our motivation for studying this problem, in more technical terms.

1.1 Ancestral selection graphs

The ancestral selection graph can be thought of as a list of coalescing and branching events. We generate the graph from the present back into the past (so time t increases into the past). Let the elements of $\mathcal{A} = \{1, 2, \ldots, n\}$ label a set of n particles. At a coalescent event, two particles merge resulting in one particle, while at a branching event one particle splits into two. In the process defined below, coalescent events dominate. The process stops the moment the number of particles drop to one. It realizes a directed, connected graph $\mathcal{G} = (V, E)$. We call the branching and coalescing process itself the ancestral selection graph process, $\{\mathcal{G}(t)\}$. Let M denote the total number of events in one realization. In Figure 1, $M = 5$, $n = 4$.

$\{\mathcal{G}(t)\}$ consists of two sub-processes, a set-valued process $\{\mathcal{A}(t) : t \geq 0\}$, denoting the set of particles at time t and a jump process $\{\mathcal{R}_m : m \geq 1\}$, giving the times at which events occur.

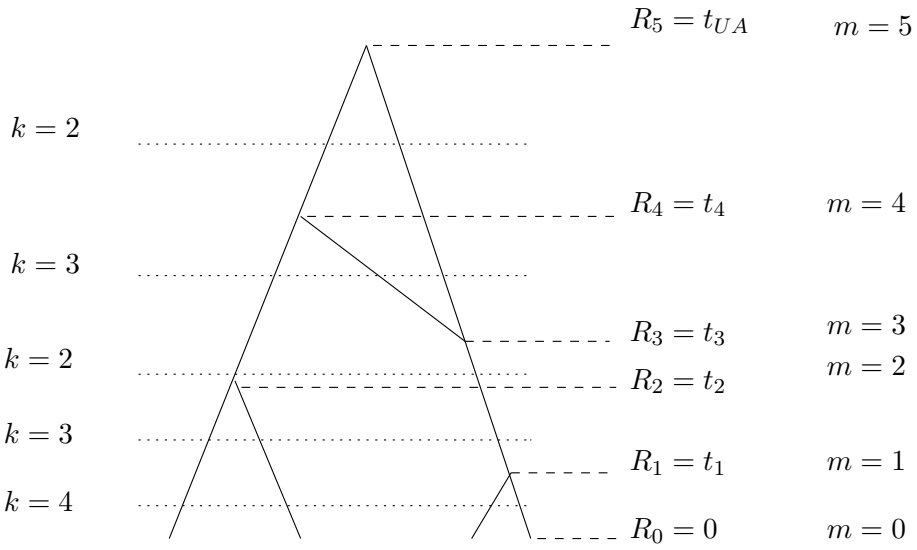


FIGURE 1: Ancestral selection graph

Let $t = R_0$ denote the time associated with the leaves. At time $t = R_0$, $\{\mathcal{A}(t)\} = \mathcal{A}$, the set of labels associated with the terminal nodes of the graph. As branching and coalescing events occur the number of lineage labels in this set may increase or decrease. For each $t \geq R_0$, $\{\mathcal{A}(t)\}$ is a subset of $\{1, 2, 3, 4, \ldots\}$. Let $|\mathcal{A}(t)|$ denote the cardinality of the set $\mathcal{A}(t)$. The process stops when the final coalescence occurs and $|\mathcal{A}(t)| = 1$.

The vertex at this time is called the ultimate ancestor (UA). The time of the UA vertex is denoted T_{UA} with $T_{UA} = \inf\{t : |\mathcal{A}(t)| = 1\}$.

We now define the time dynamics in detail. Let R_m, $m \geq 1$ be a time at which either a coalescing or a branching event occurs. Let $\sigma \geq 0$ be a real positive constant. While $|\mathcal{A}(R_{m-1})| = k$, coalescing happens at rate $\binom{k}{2}$ and branching happens at a rate $\sigma k/2$. The time intervals $\{R_m - R_{m-1}\}$ are independent and exponentially distributed with rate parameter $\binom{k}{2} + \frac{\sigma k}{2}$. If a branching event happens at time R_m a random particle j branches resulting in

an extra particle with label $n+m$, so that $\mathcal{A}(R_m) = \mathcal{A}(R_m^-) \cup \{n+m\}$. When two random ancestors i, j coalesce with $i < j$, then $\mathcal{A}(R_m) = \mathcal{A}(R_m^-) \backslash \{j\}$ and the resulting ancestor is labelled i.

The selection parameter σ determines the amount of branching in the ancestral graph. If $\sigma = 0$ the ancestral selection graph collapses to the n coalescent of Kingman. Large values of σ results in many branching events.

1.2 The space of ancestral selection graphs

In this section we define a probability space, $(\Gamma, \mathcal{F}, \mathbf{P})$, where Γ is the sample space of all ancestral graphs, $\mathcal{F}$ is a suitable σ-algebra of subsets of Γ and $\mathbf{P}$ is a probability measure on Γ.

For a graph $g \in \Gamma$, let $V_c \subset V_A$ be the set of vertices where coalescing of two edges occur, $V_b = V_A \setminus V_c$ the set of vertices where an edge branches and V_L the set of leaf tips, that is, the set of vertices representing the data, $V_g = V_c \cup V_b \cup V_L$. We define corresponding sets of edges, $E_g = E_b \cup E_c$, with E_b the set of edges branching at the top and E_c the set of edges coalescing at the top ("up" is into the past, increasing t). Let N_c denote the number of coalescing events and N_b the number of branching events. Then $N_c = n + N_b - 1$.

An ancestral selection graph g is a directed graph. Associated with each vertex $v \in V_g$ is a time t_v, measured in years, increasing from the leaves to the root vertex of the graph (UA). Edge $\langle v, w \rangle \in E_g$ is directed from the "child" v to w, the "parent", so our convention is $t_v \leq t_w$.

For each vertex, $v \in V_g$, let d_{in} and d_{out} denote the in-degree and the out-degree. Define the set of all admissible ancestral graph topologies on N_b branching vertices and n leaves as $\Gamma_{N_b,n} = \{(V_g, E_g)$ such that for $v \in V_L$, $u \in V_b$ and $w \in V_c$, the respective degrees are $d_{\text{in}}(v) = 0$, $d_{\text{out}}(v) = 1$, $d_{\text{in}}(u) = 1$, $d_{\text{out}}(u) = 2$, $d_{\text{in}}(w) = 2$, $d_{\text{out}}(w) = 1\}$. Let t_A denote the

ordered set of times associated with the vertices $v \in V_A$. Let

$$\chi_{V,E} = \{t_A : t_A \in [R_0, \infty)^{N_b+N_c} \text{ and for } \langle i,j\rangle \in E_g, t_i < t_j\},$$

denote the space of vertex times for a given topology.

The space Γ of selection graphs is then

$$\Gamma = \bigcup_{N_b=0}^{\infty} \bigcup_{\{V,E\}\in\Gamma_{N_b,n}} \bigcup_{t_A\in\chi_{V,E}} \{(V,E,t_A)\}.$$

For $g \in \Gamma$ let $g = (V_g, E_g, t_A(g))$ be an ancestral selection graph with $M = N_b + N_c$ ancestral vertices. Let $\nu(t_A)$ denote the element of volume in χ_{V_g,E_g} so that

$$d\nu(t_A) = dt_1 dt_2 dt_3 \cdots dt_M .$$

The element of measure dg is $d\nu(t_A(g))$ with counting measure over distinct topologies.

For each set $B \in \mathcal{F}$ let $\mathbf{P}(B)$ give the probability that any given realization $\mathcal{G} = g$ of the ancestral selection graph process is in B, that is, $\mathbf{P}(B) = \Pr\{\mathcal{G} \in B\}$. The distribution $\mathbf{P}$ has a density $\pi(g)$ with respect to the measure dg, so that $\mathbf{P}(dg) = \pi(g)dg$, where

$$\pi(g) = \left(\frac{\sigma}{2}\right)^{N_b} \prod_{m=1}^{N_b+N_c} e^{-\rho_m \tau_m},$$

$\rho_m = \binom{k_m}{2} + \frac{\sigma k_m}{2}$, k_m is the number of lineages in the graph between event m and event $m+1$, and τ_m is the time between events m and $m+1$.

In order to compute the expectation of a state function $f : \Gamma \to \Re$, for example $f(g) = t_{UA}(g)$, the time to ultimate coalescence, we compute

$\mathsf{E}_{\mathcal{G}}\{f(g)\} = \int_\Gamma f(g)\pi(g)dg$, that is,

$$\int_\Gamma f(g)\,\pi(g)\,dg = \sum_{N_b=0}^{\infty} \sum_{\{V,E\}\in\Gamma_{N_b,N_c}} \int_{\chi_{V,E}} f(V,E,t_A)\,\pi(V,E,t_A)\,dt_1 dt_2 \cdots dt_{N_b+N_c}\,.$$

In the next section we show how to evaluate such expressions numerically using Markov chain Monte Carlo averaging.

2 Sampling ancestral selection graphs

2.1 Metropolis-Hastings algorithm

Using the Metropolis-Hastings algorithm we construct a Markov Chain $\{\mathcal{G}_n\}$, $n = 0, 1, 2, \ldots$ of random variables converging (geometrically) to the equilibrium density π on Γ.

The Metropolis-Hastings algorithm consists of two steps. Suppose the chain is in state $\mathcal{G}_n = g$. First, draw a candidate state g' in the following way. Draw uniform random variates $u = (u_0, u_1, u_2, \ldots)$ according to some fixed simple density q and compute $g' = \psi(g, u)$, where ψ is a fixed mapping. We are to some extent (details below) free to choose q and ψ. We suppose there is unique $u' = (u'_1, u'_2, \ldots)$ such that $g = \psi(g', u')$. The new state is accepted with probability $\alpha(g'|g)$, where

$$\alpha(g'|g) = \min\left[1, \frac{\pi(g')}{\pi(g)}\frac{q(u')}{q(u)}\left|\frac{\partial(g',u')}{\partial(g,u)}\right|\right] \tag{1}$$

If the new state is accepted, $\mathcal{G}_{n+1} = g'$, else $\mathcal{G}_{n+1} = g$.

The above expression for the acceptance probability is needed in the variable dimension setting (but is convenient in any case). It is due to Green [3]. Some details of the algorithm may be clarified in the example below.

Roughly speaking, the density q and the mapping ψ must be chosen so that the Markov chain is irreducible on Γ. Since Γ is continuous we need the chain to be π-irreducible (in the sense of [8]). A proof that this property holds for the chain below is straightforward though lengthy. The above form for the acceptance probability ensures the Markov chain is reversible with respect to $\pi(g)$. It follows that the sequence $\{\mathcal{G}_n\}$, $n = 0, 1, 2, \ldots$ is ergodic with unique equilibrium distribution π. The Metropolis-Hastings birth-death process we describe below is similar to that presented in [2], one of the first instances of this type of Markov chain Monte Carlo.

2.2 Posterior distribution

Let D denote data, observed at the leaf tips of the tree. This data is a realization of a type mutation process which propagates down the graph from the ultimate ancestor. The leaf-data is informative of graph structure, since leaves with a recent common ancestor are likely to have similar type values. All parameters of the mutation process are assumed known.

Consider the problem of recovering the graph g from leaf data D. The posterior density $\mathbf{P}(g|D)$ contains all available information concerning g. The posterior density, is proportional to the product of two terms, a likelihood function, $\mathbf{P}(D|g)$, and the prior density $\pi(g)$. In a Monte-Carlo approach we summarize $\mathbf{P}(g|D)$ using samples drawn from $\mathbf{P}(g|D)$.

When we write a computer program implementing Markov chain Monte Carlo for this problem a very large part of the work is the problem of representing the evolving graphical state, and designing and implementing reversible MCMC updates that act on the graph. In other words, the MCMC

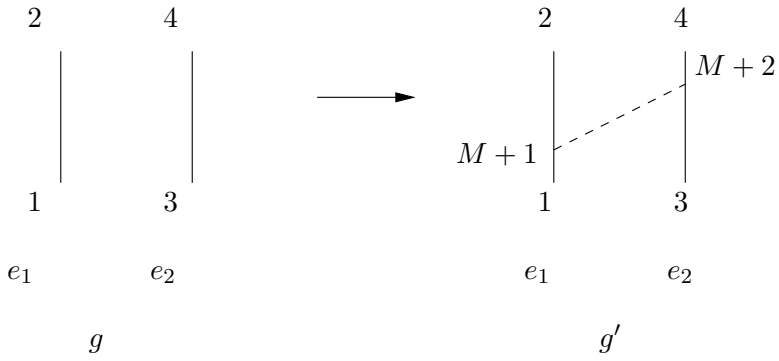
simulation of $g \sim \pi(g)$ is a large part of the problem of making MCMC simulation of $g \sim \mathbf{P}(g|D)$. The transition matrix of a MCMC algorithm for the prior density can be used as a proposal density for an MCMC simulation of the full posterior Note that this is quite different from, and much more efficient than, a rejection algorithm using the prior to generate iid draws which are rejected by the likelihood. We are drawing from a transition matrix, so the proposed state differs at just a few nodes from the current state.

Perfect sampling using coupling from the past was used by Fearnhead [1] to simulate the joint distribution $\mathbf{P}(g, D)$. The Fearnhead algorithm is inspired by, but differs radically from, the original perfect sampling by coupling from the past, due to Propp and Wilson [6]. Almost all perfect sampling algorithms to date need the MCMC to be stochastically monotone relative to a partial order defined on the space states. The lack of many applications of perfect sampling to data reflects the fact that it is in general very difficult to find such an order. Data tends to destroy the kind of simple symmetric combinatorial structures needed for monotonicity.

2.3 Moves

A hybrid strategy, consisting of two moves, is implemented to move around in the space of graphs. With probability $p^* = \frac{1}{2}$ one chooses to add an edge while with probability $p^\dagger = 1 - p^* = \frac{1}{2}$ one deletes an edge from a graph g. Variate $u_0 \sim U(0, 1)$ is used to simulate this decision. The role of variates $u_1, u_2, \ldots$ depends on the outcome of this first choice.

2.3.1 Add an edge



Suppose state $g = (V, E, t_A)$ has $M = N_b + N_c$ ancestral vertices. Choose two edges $e_1, e_2 \in E$ uniformly at random with replacement. Without loss of generality, suppose these are $e_1 = \langle 1, 2\rangle$ and $e_2 = \langle 3, 4\rangle$. The edge above the root can be chosen (so for the purpose of this algorithm it is convenient to regard E as containing such an edge). Points corresponding to new vertices with labels $M+1$ and $M+2$ are chosen on e_1 and e_2 using uniform random variates u_1 and u_2, respectively. Let $t'_{M+1} = t_1 + (t_2 - t_1)u_1$ and $t'_{M+2} = t_3 + (t_4 - t_3)u_2$. Connect vertices $M+1$ and $M+2$ with an edge e. Let $\tau_1 = t_2 - t_1$ and $\tau_2 = t_4 - t_3$ be the lengths of edges e_1 and e_2 respectively. The new state $g' = (V', E', t'_A) = (V \cup \{M+1, M+2\}, E_g \cup \{e\}, (t_A, t_{M_1}, t_{M+2})$.

Notice $t_A \in [R_0, \infty)^{N_b+N_c}$ while $t'_A \in [R_0, \infty)^{N_b+N_c+2}$ so the state dimension changes. The Jacobian for the transformation is the determinant of an

$M+2 \times M+2$ matrix,

$$\frac{\partial(g,u')}{\partial(g,u)} = \begin{array}{c|cccccccccc} & \mathbf{t_1} & \mathbf{t_2} & \mathbf{t_3} & \mathbf{t_4} & \mathbf{t_5} & \dots & \mathbf{t_M} & \mathbf{u_1} & \mathbf{u_2} \\ \hline \mathbf{t'_1} & 1 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ \mathbf{t'_2} & 0 & 1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ \mathbf{t'_3} & 0 & 0 & 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ \mathbf{t'_4} & 0 & 0 & 0 & 1 & 0 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & 0 & 0 \\ \mathbf{t'_{M+1}} & 1-u_1 & u_1 & 0 & 0 & 0 & \dots & 0 & \tau_1 & 0 \\ \mathbf{t'_{M+2}} & 0 & 0 & 1-u_2 & u_2 & 0 & \dots & 0 & 0 & \tau_2 \end{array}$$

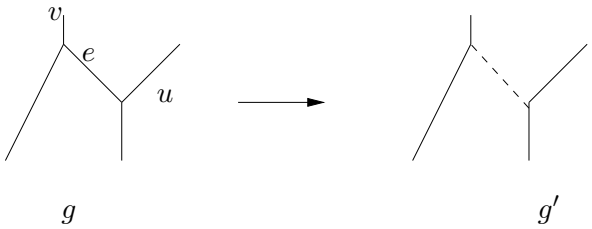
that is $\partial(g,u')/\partial(g,u) = \tau_1\tau_2$. Because there are two ways to choose e_1 and e_2 (either order), the generation probability is $q(u) = p^* 2/|E|^2$.

The generation process for the reverse move is given in the next Section, so that the acceptance probability is

$$\alpha(g'|g) = \min\left[1, \frac{\pi(g')}{\pi(g)}\frac{p^\dagger}{p^*}\frac{1}{2}\frac{|E|^2}{|E'_D|} \times \tau_1\tau_2\right]$$

We have omitted two special cases. When one of the edges e_1, e_2 is the edge above the root, an exponential density is used to choose the new vertex location. The new vertex will be the new ultimate ancestor if the update is accepted. The acceptance probability must be modified to account for the revised proposal probability. Another special case is when $e_1 = e_2$, a case we call a "bubble". In that case $q(u) = p^*/|E|^2$. The probability to propose the reverse move is altered, also. The ratio $q(u')/q(u)$ becomes $|E|^2/|E'_D|$.

2.3.2 Delete an edge



Let E_D be the set of "deletable" edges in g. Directed edge $e = \langle u, v \rangle$ is in E_D iff $v \in V_c$ and $u \in V_b$. An edge e is drawn from E_D. The new state g' is generated by deleting edge e from g. The generation probability is $q(u) = p^\dagger / |E_D|$. The acceptance probability is

$$\alpha(g'|g) = \min \left[1, \frac{\pi(g')}{\pi(g)} \frac{p^*}{p^\dagger} \frac{2|E_D|}{|E'|^2} \times \frac{1}{\tau_1 \tau_2} \right].$$

When one of the edges e_1, e_2 is the edge above the root and edges which are part of bubbles are again special cases. There is a third special case for deletion: if at any time t in the interval (t_u, t_v) the number of edges drops to one, so $|A(t)| = 1$ then the candidate is rejected (so $\mathcal{G}_{n+1} = g$). The section of the new graph g' above t is above the ultimate ancestor of g', and hence g' is not in Γ.

3 Discussion

Most of the work in implementing the MCMC goes into the careful planning of the graph data structure used to represent an evolving graph with a time-varying number of nodes. The vertex and edge birth and death operators should be given a careful modular implementation, in order to avoid much special case handling. The program was checked by making comparisons with analytical results from Neuhauser and Krone [5].

Acknowledgements: Thank you to Prof. Allen Rodrigo of The School of Biological Sciences, University of Auckland, who suggested this project.

References

[1] Paul Fearnhead. Perfect sampling from popultion genetic models with selection. *Theoretical Population Biology*, 59(4):263–279, 2000. C400

[2] C. J. Geyer and J. Møller. Simulation procedures and likelihood inference for spatial point processes. *Scand. J. Statist.*, 21:359–373, 1994. C399

[3] Peter J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732, 1995. C393, C399

[4] J. F. C. Kingman. The coalescent. *Stochastic Processes and their Applications*, 13:235–248, 1982. C393

[5] Stephen M. Krone and Claudia Neuhauser. Ancestral processes with selection. *Theoretical Population Biology*, 51:210–237, 1997. C394, C403

[6] J. G. Propp and D. B. Wilson. Exact sampling with coupled markov chains and applications to statistical mechanics. *Random Structures and Algorithms*, 9:223–252, 1996. C400

[7] Paul F. Slade. Simulation of selected genealogies. *Theoretical Population Biology*, 57:35–49, 2000. C393

[8] Luke Tierney. Markov chains for exploring posterior distributions. *Annals of Statistics*, 22(4):1701–1728, 1994. C399