Importance sampling on the coalescent with recombination

A thesis submitted for the degree of Doctor of Philosophy

Paul A. Jenkins
St. Edmund Hall, University of Oxford
Michaelmas Term 2008

Importance sampling on the coalescent with recombination

Paul A. Jenkins, St. Edmund Hall

DPhil thesis, Michaelmas Term 2008

Abstract

Performing inference on contemporary samples of homologous DNA sequence data is an important task. By assuming a stochastic model for ancestry, one can make full use of observed data by sampling from the distribution of genealogies conditional upon the sample configuration. A natural such model is Kingman's coalescent, with numerous extensions to account for additional biological phenomena. However, in this model the distribution of interest cannot be written down analytically, and so one solution is to utilize importance sampling.

In this context, importance sampling (IS) simulates genealogies from an artificial *proposal distribution*, and corrects for this by weighting each resulting genealogy. In this thesis I investigate in detail approaches for developing efficient proposal distributions on coalescent histories, with a particular focus on a two-locus model mutating under the infinite-sites assumption and in which the loci are separated by a region of recombination. This model was originally studied by Griffiths (1981), and is a useful simplification for considering the correlated ancestries of two linked loci. I show that my proposal distribution generally outperforms an existing IS method which could be recruited to this model.

Given today's sequencing technologies it is not difficult to find volumes of data for which even the most efficient proposal distributions might struggle. I therefore appropriate resampling mechanisms from the theory of sequential Monte Carlo in order to effect substantial improvements in IS applications. In particular, I propose a new resampling scheme and confirm that it ensures a significant gain in the accuracy of likelihood estimates. It outperforms an existing scheme which can actually diminish the quality of an IS simulation unless it is applied to coalescent models with care. Finally, I apply the methods developed here to an example dataset, and discuss a new measure for the way in which two gene trees are correlated.

Acknowledgements

I would like to express my unbounded gratitude to my supervisor Bob Griffiths, for introducing me to mathematical genetics, for offering his wisdom and encouragement, and for his patient reviewing of drafts of this work.

I have also been fortunate enough to be surrounded by many helpful researchers in the field, from whom I have learnt an inestimable amount. I would especially like to thank Niall Cardin, Yvonne Griffiths, Jotun Hein, Rune Lyngsø, Gil McVean, Paul Munday, Yun Song, and Damjan Vukcevic, for collectively many many hours of discussions and advice. I would also like to thank the administrators and IT support of the Statistics Department for their work, especially Beverley Lane; my office-mates Afie, Bence, Robin, and Yuqiang; and my examiners Chris Holmes and Carsten Wiuf.

My deepest gratitude goes to the staff and students of the Life Sciences Interface Doctoral Training Centre, who made my transition to DPhil such an invaluable experience, and whom I could not mention without thanking the DTC's core team at the time I joined: David Gavaghan, James Wakefield, Maureen York, and Charlotte Smith. Thanks also to the EPSRC for their generous financial support.

Finally, thank you to all my friends who kept me aware that there's more to life than maths, particularly everyone on the Old Black Horse quiz team, and the boys back home—Jon, Tom, Rob, Oli Softrock, Nick, and Paul. Thank you to my Mum and Dad for your prodigious and unwavering support, to my brother Joe, and to Doll Pickering for keeping the village updated on my work. Lastly, thank you Sarah, who began this degree as a stranger and ended it as my wife. I could not have wished for anyone else with whom to share my journey.

Contents

Chapter 1

Introduction

In recent years there have been rapid advances in DNA sequencing technologies, giving access to a vast wealth of genetic data. Potentially, one can address a wide variety of questions with this information, such as: What mutation and recombination rates would give rise to such data? What is the distribution of the time to the most recent common ancestor within and between loci? What demographic changes can be inferred from the data? Various summary statistics of the data might be employed for inference, such as the number of segregating sites for inferring mutation rates [1], but by their nature they discard some information. Preferable would be to make full use of the data; a useful approach in this setting is to assume that it has been generated from some stochastic model of ancestry [2]. By simulating from this model, one can compare what we expect from a given set of parameters to what we have observed, thus making inference much more intuitive. But what stochastic model should we use? In a sense, the decision is already made for us: for any of a range of simple population models that might apply here and have been used historically, on letting the population size grow large and rescaling time appropriately

the limiting process is the same. This is *Kingman's coalescent* [3, 4], with extensions to include additional biological processes like recombination [5, 6].

Computationally-intensive simulation using the coalescent is a popular method for performing inference. Yet even with today's technology, current methods often struggle to handle reasonably-sized datasets. Frequently one must focus either on simple genetic models of ancestry, or on approximate computational methods for inference, or both. Throughout this thesis I shall be particularly interested in a model for data assumed to have been derived from the infinite-sites model with a single recombination breakpoint, which is suited to the following rather broad question: given two adjacent and correlated gene trees, what can be inferred about the way they fit together?

The structure of this thesis is as follows. In the remainder of this chapter, I shall develop the theory of Kingman's coalescent, and illustrate how it is an attractive model with respect to commonly used models for finite populations by deriving it as a limit from the *Wright-Fisher* model. The use of the coalescent with various models of mutation is described, as is its extension to incorporate recombination. Although I will not develop it here, I merely mention that the coalescent lends itself to incorporating many other population genetic phenomena: migration, population growth, and so on. Then I describe how importance sampling may be used to perform inference on the coalescent, and in Chapter 2 I develop new importance sampling proposal distributions under a two-locus model with recombination, for each of a finite-alleles and infinite-sites model of mutation. Properties of the latter are investigated in detail. It is implemented in a program written in C++ and its performance compared with existing proposal distributions.

Even in the setting of simplified models in which I am interested, it is not diffi-

cult to obtain datasets for which these importance sampling methods still struggle. In Chapter 3 I therefore extend two rejuvenation procedures from sequential Monte Carlo—*stopping-time resampling* and *pilot-exploration resampling*—for use with coalescent models, and illustrate that provided they are adopted in a careful manner they can be of considerable benefit (whereas a more naïve adoption can actually diminish the efficiency of the importance sampling procedure).

In Chapter 4 I illustrate the use of this proposal distribution by performing ancestral inference on an example dataset, obtaining estimates including parameter values for the model, ages of mutations, and the time to the most recent common ancestor (TMRCA) of each locus. A natural problem in a two-locus model is to measure the extent of shared sequence ancestry between the histories of the loci, and so I also introduce a new way of summarizing the distribution of ancestral recombination graphs associated with a set of data—the "*gene graph*"—which reduces the state space of genealogies to one which is finite.

1.1 The coalescent

The coalescent [3, 4] (see also Hein *et al.* (2005) [7] for an introduction) could be derived from several models of reproduction, but in this section I shall focus on the Wright-Fisher model as an important example. The Wright-Fisher model applies to a constant population of $2M$ genes drawn from $2M$ haploid or M diploid individuals. Note that I invoke the term 'gene' loosely: throughout this thesis I shall use it interchangeably with 'sequence' or 'haplotype' to mean an appropriately sampled and phased stretch of DNA. We begin with a number of simplifying assumptions that will be inherited by the standard coalescent. The population is selectively neutral, that

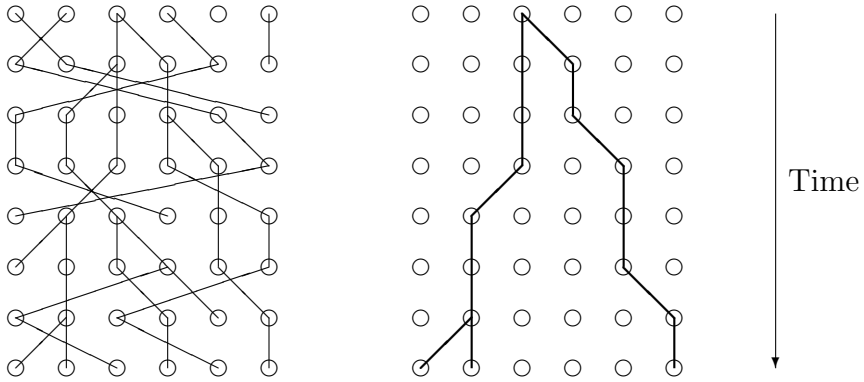


Figure 1.1: (*Left*): Simulation of the Wright-Fisher model for six genes. Successive generations are shown in each row, with time flowing downwards. Genes are joined to their parent by a black line. (*Right*): The ancestry of a subsample of three in the present generation.

is, all individuals have equal reproductive success; the population has no structure, so that for a given gene in the current generation, each individual in the previous generation is equally likely to be its parent; and there are no further complicating biological processes such as recombination. In aggregate these assumptions result in a rather unrealistic model biologically, but it is at least mathematically tractable. It is worth noting that each of the assumptions can be relaxed, resulting in suitably modified versions of the coalescent; indeed, we shall introduce recombination shortly. In the Wright-Fisher model we proceed with discrete, non-overlapping generations. The genealogy is constructed by each gene in the present generation independently sampling its parent from the previous generation uniformly at random. The process is commonly represented diagrammatically as in Figure 1.1. By tracing the parentage backwards from a sample of size n in the present generation, a genealogy for n genes can be sampled under this model.

As Figure 1.1 illustrates, in each generation some genes will have one or more offspring, while others have none. As one traces the present generation's ancestry

back in time, two genes will eventually share a common ancestor. In the example shown, the most recent common ancestor (MRCA) of the current generation is found seven generations back. This is also the MRCA of the subsample of three genes in Figure 1.1 (*right*).

The coalescent process derives from the limiting behaviour of this process as $M \to \infty$, with time suitably rescaled. It is straightforward to show that the probability that j genes were chosen from $j-1$ parents is

$$p_{j,j-1} = \frac{\binom{j}{2}}{2M} + O(M^{-2})$$

as $M \to \infty$, while $p_{j,j} = 1 - p_{j,j-1} + O(M^{-2})$ and all other possibilities are $O(M^{-2})$ or smaller. If we denote the time while there are k ancestors by τ_k and define $T_k^{(M)} = \frac{\tau_k}{2M}$, the waiting time scaled by $2M$, then

$$\mathbb{P}(T_k^{(M)} \leq t) = 1 - p_{k,k}^{\lfloor 2Mt \rfloor} \approx 1 - \left(1 - \frac{\binom{k}{2}}{2M}\right)^{\lfloor 2Mt \rfloor} \to 1 - \exp\left[-\binom{k}{2}t\right]$$

as $M \to \infty$. We deduce that embedded in the coalescent process is a death process $\{A_n(t), t \geq 0\}$ on the number of ancestors looking back in time, with successive death rates $\binom{k}{2}$ for $k = n, \ldots, 2$. The corresponding times T_k while there are k ancestors are independent exponentially distributed random variables with parameters $\binom{k}{2}$, and are said to be the distribution for the time of each *epoch*. Time is measured in units of $2M$ generations, and the model is a limit from the Wright-Fisher model as $M \to \infty$. This death process reaches a common ancestor with probability 1. Note that the death process is independent of the actual collection of coalescence events between pairs of lineages—of the *jump chain*, in Kingman's terminology. An

example coalescent tree is shown in Figure 1.2 (*left*).

The coalescent is mathematically appealing; much can be inferred easily from a model of this form. As a simple example, note that with T_k defined as above we can obtain the distribution of the total branch length in a coalescent tree. Its mean is

$$\mathbb{E}(L_n) = \sum_{k=2}^{n} k\mathbb{E}(T_k) = 2\sum_{k=1}^{n-1} \frac{1}{k}. \tag{1.1}$$

Beyond its mathematical elegance, there are at least three reasons why the coalescent is such a useful tool. The first has been mentioned already—it is robust to the underlying finite-population model provided time is rescaled appropriately. Here we have seen its relationship to the standard Wright-Fisher model, but many assumptions can be modified without destroying convergence to the coalescent, including the variance of the distribution in the number of offspring and a change in the population size. In these cases M is replaced by the *effective population size* M_e, which can be interpreted as the number of genes in a standard Wright-Fisher model with the same time-scaling on passing to a coalescent limit; see Nordborg (2001) [8] for further discussion. Second, the coalescent is retrospective rather than prospective. It simulates ancestry starting from the present, and, according to the results above, this requires only a collection of mutually independent exponential random variables along with an independently generated bifurcating tree (with each coalescence event chosen to occur uniformly at random to a pair of lineages). Contrast this with simulating from the Wright-Fisher model, for example, whose design—looking forwards in time at least—does not distinguish between events relevant to the contemporary sample and those affecting the whole population. Third, a consequence of the selective neutrality of the coalescent is that the mutation process can

be separated entirely from the genealogical process. Thus various mutational models can be superimposed onto the coalescent process according to the data of interest. Several important examples are introduced below, but we begin by adumbrating how to incorporate the mutational mechanism into the coalescent.

1.2 Models of mutation

Consider again the Wright-Fisher model; other starting points are possible. When each gene arises from its parent it mutates to another type with probability u, otherwise it shares the type of its parent. Denote the time until the first mutation along one lineage by $T^{(M)}$, measuring time in units of $2M$ generations. Then

$$\mathbb{P}(T^{(M)} \leq t) = 1 - (1-u)^{\lfloor 2Mt \rfloor} = 1 - \left(1 - \frac{\theta/2}{2M}\right)^{\lfloor 2Mt \rfloor} \to 1 - \exp\left(-\frac{\theta}{2}t\right)$$

as $M \to \infty$. In other words we suppose $u = O(M^{-1})$ as $M \to \infty$ while keeping $\theta = 4Mu$ fixed. Inter-arrival times of mutations along a single lineage are thus independent exponentially distributed random variables with parameter $\frac{\theta}{2}$; mutations occur as a Poisson process along the branches of the coalescent tree. This suggests two equivalent ways for simulating a coalescent tree with mutations. First, one can generate a coalescent tree without mutations and then scatter a Poisson-distributed number of mutations uniformly on the branches of the tree. The distribution of the number of mutations M_n on such a coalescent tree has mean

$$\mathbb{E}(M_n) = \mathbb{E}[\mathbb{E}(M_n|\, L_n)] = \mathbb{E}\left[\frac{\theta}{2}L_n\right] = \theta \sum_{k=1}^{n-1} \frac{1}{k}, \tag{1.2}$$

using (1.1). Second, one can generate the coalescent tree concurrently with its mutations. Note that while there are k lineages, as one goes back in time coalescences occur exponentially at rate $\binom{k}{2}$, while mutations occur exponentially at rate $k\frac{\theta}{2}$. By the properties of competing exponentials, this defines an exponential random variable with rate $\binom{k}{2} + k\frac{\theta}{2}$ which models the time until the next event. With probability $\frac{k-1}{k-1+\theta}$ it is a coalescence event on a randomly chosen pair of lineages, otherwise with probability $\frac{\theta}{k-1+\theta}$ it is a mutation event on a randomly chosen lineage. This procedure forms the basis of many urn models (see for example [9] and references therein), which can be formulated equivalently forwards in time. To digress briefly, it is non-trivial which of these two suggestions is used when it comes to designing Monte Carlo schemes. For example, the former is used in the pioneering importance sampling technique of Griffiths & Tavaré [9, 10, 11, 12, 13], while the approach of Kuhner *et al.* [14, 15] separates the simulation of genealogies from the set of possible mutation configurations thereon.

The model above is so far free from prescribing precisely *how* allelic types change during a mutation. We now consider some commonly used models of mutation. Given such a model, allelic types can be associated with each point on the tree. A way to assign such types would be to draw the allele of the MRCA from some stationary distribution, and allow this to cascade down the tree to the leaves, modifying the allele according to the prescribed model wherever a mutation event is encountered.

1.2.1 The finite-alleles model

The finite-alleles (or K-alleles) model is a rather general mutation model on a finite type space E, and is defined by a transition matrix P—at a mutation event in which the parent is of type $i \in E$, the type of the offspring is $j \in E$ with probability P_{ij}. The stationary distribution for choosing the type of the MRCA is usually chosen to coincide with the stationary distribution of P.

The finite-alleles model incorporates a variety of biological models. For example, we may wish to model the evolution of a single site, in which case $E = \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$, and P is determined by whatever model for base changes is appropriate. Perhaps the simplest such choice is the *Jukes-Cantor* model in which $P_{ij} = \frac{1}{4}$. There are many extensions to this model to account for things like unequal rates of transition/transversion and unequal base frequencies. An important special case of the finite-alleles model arises when the type of the offspring is independent of its parent: then we have parent independent mutation (PIM) given by $P_{ij} = P_j \ \forall i \in E$.

The single site model can be extended to a *finite-sites* model for a sequence of length L, provided we assume each site mutates independently. Then $E = \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}^L$, and if we denote by h_j the probability conditional on a mutation occurring that it occurs to the jth site, with this site's 4×4 transition matrix denoted by Q_j, then the transition matrix for the complete sequence is given by

$$P = \sum_{j=1}^{L} h_j I \otimes I \otimes \ldots \otimes Q_j \otimes \ldots \otimes I,$$

where $\otimes$ denotes direct product, and Q_j resides at the jth position in the product.

There are a number of other applications of the finite-alleles model, including a model for codon evolution in which the type space is over the 64 possible triplets of

nucleotides, and a truncated approximation to the stepwise mutation model (Section 1.2.4) whose type space is ordinarily countably infinite.

1.2.2 The infinite-alleles model

In the infinite-alleles model, the allelic type of a mutant is always assumed to be new. One can model this by letting $E = [0, 1]$ and labelling each new allele by drawing a uniform random number from $[0, 1]$. This serves as a model for data in which there are no quantitative differences between alleles—one knows only whether two genes have the same allele or not, and so the data is equivalent to the collection (α_j), where α_j is the number of alleles with j representatives in the sample. In the context of inference there is no reason to trace the ancestry of any lineage back beyond its most recent mutation, since this does not affect the data. It is conventional to treat a lineage beyond the mutation as non-ancestral; then we are interested only in the random forest of subtrees that trace each allelic class back to the last mutation.

1.2.3 The infinite-sites model

The infinite-sites model [16, 1] can be regarded as a limit in the finite-sites model as $L \to \infty$. Then each mutation always falls on a site that has not experienced a mutation anywhere else in the ancestry; repeat mutations and back mutations are not observed. While for much data this is biologically unrealistic, it does lead to some dramatic simplifications in applications such as the detection of recombination events [17] and the reconstruction of perfect phylogenies [18]. Where the infinite-sites model *is* used, it is best applied when the number of *segregating sites*—that is, sites exhibiting more than one type in a sample—is much less than the total

number of sites of a gene under investigation, due to a relatively long sequence of nucleotides of interest or to a low scaled mutation rate. A representation for this model is to label segregating sites by elements of $[0,1]$; the type space is $E = [0,1]^{\mathbb{Z}_+}$. The labelling can be arbitrary, as in the infinite-alleles model, or by mapping the co-ordinates of the gene to $[0,1]$ the label can then represent the location of that site. So a gene is of type $\boldsymbol{x} = (x_0, x_1, \ldots) \in E$ if $x_0, x_1, \ldots$ is the age-ordered sequence of sites at which mutations have occurred in the line of descent of that gene, where x_0 is the most recently mutated site. For two mutations occurring on the same branch it is not possible to determine their relative ages, which can be assigned arbitrarily for convenience.

For a sample $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) \in E^n$ with $n \in \mathbb{N}$, denote the jth co-ordinate of $\boldsymbol{x}_i$ by x_{ij}. Provided:

1. $\forall i, j, k,\ x_{ij} = x_{ik} \Longrightarrow j = k$,

2. if $i, i' \in \{1, \ldots, n\},\ j, j' \in \mathbb{Z}_+$, then $x_{ij} = x_{i'j'} \Longrightarrow x_{i(j+k)} = x_{i'(j'+k)},\ k = 0, 1, \ldots$,

3. $\exists\ j_1, \ldots, j_n \in \mathbb{Z}_+$ such that $x_{1j_1} = \ldots = x_{nj_n}$,

then the sample is equivalent to a *gene tree* [19], that is, a perfect phylogeny relating the mutation histories of each sequence. The conditions above ensure that (1) no mutation is repeated along a lineage, (2) whenever the same mutation is observed in two lineages then they must have shared the same ancestry further back in time, and (3) each lineage reaches the same root. Each segregating site has two alleles, and by assigning one as the ancestral type and one as the mutant, the root of the tree—the type ancestral at all sites—is assumed known. By toggling the ancestral/derived

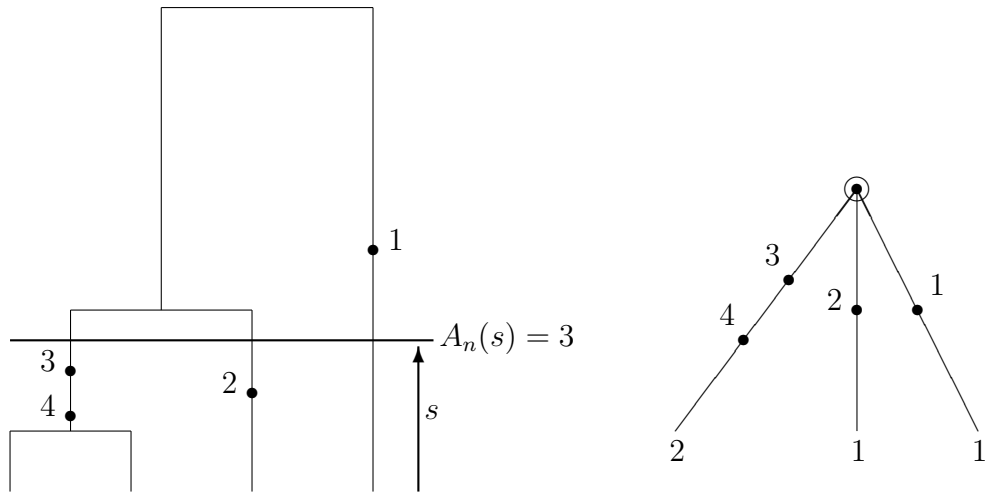


Figure 1.2: (*Left*): A coalescent tree for a sample of four genes with four mutations. The sample at the present time is at the leaves of the tree, and edges trace the ancestry backwards in time. Vertices represent coalescence events. The process continues backwards until a common ancestor is reached. $A_n(t)$ is the number of ancestors at time t back—for example, at time s there are three ancestors. (*Right*): The corresponding rooted gene tree, in which mutations are vertices and time information is lost. It is common to collapse identical paths back to the root and to indicate their multiplicities at the leaves.

state at each site one can obtain every rooted tree compatible with the data, and these are all associated with a single *unrooted* gene tree. In this work we shall always assume that the root is known. A rooted gene tree associated with a coalescent history is shown in Figure 1.2. Throughout this thesis I shall use the notation $\mathcal{T} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$, and it is common to collapse identical paths so that the data is equivalent to the pair $(\mathcal{T}, \boldsymbol{n})$, where $\boldsymbol{n}$ records the multiplicity of each distinct path [19]. Then $\boldsymbol{x}_i$ records one of d distinct sequence types rather than n sequences. For example, the gene tree in Figure 1.2 can be written in the following form:

```
2  :  4 3 0
1  :  2 0
1  :  1 0
```

This corresponds to the input format for R.C. Griffiths' program `genetree` [20];

each row is a distinct allele, represented by its multiplicity and then the path to the root. It is easy to see how to recover the mutation pattern of each sequence from the rooted tree—simply trace the sequence through the tree from the leaf back to the root, recording each mutation along the way. Conversely, efficient algorithms exist to construct the rooted gene tree from the mutation configuration [18], which can be stored via an *incidence matrix* $C \in M_{n,s}(\mathbb{Z}_2)$ (where s is the number of segregating sites), defined by

$$c_{ij} = \begin{cases} 1 & \text{if site } j \text{ on sequence } i \text{ is mutant,} \\ 0 & \text{if site } j \text{ on sequence } i \text{ is ancestral.} \end{cases}$$

Note that when the labelling of sites is entirely arbitrary, one is really interested only in equivalence classes of gene trees, given by: $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) \sim (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n)$ if there exists a bijection $\zeta : [0,1] \to [0,1]$ with $y_{ij} = \zeta(x_{ij})$ for $i = 1, \ldots, n$ and $j = 0, 1, \ldots$. The function ζ simply relabels mutations. One might further be interested in the equivalence class of gene trees in which *sequences* are also unlabelled, given by: $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) \simeq (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n)$ if there exists a bijection $\zeta : [0,1] \to [0,1]$ and a permutation σ of $\{1, \ldots, n\}$ with $y_{\sigma(i)j} = \zeta(x_{ij})$ for $i = 1, \ldots, n$ and $j = 0, 1, \ldots$. Members of the former equivalence class are usually referred to as *labelled trees*, the latter as *unlabelled trees*, and often we do not distinguish between an equivalence class and a representative member. The 'labelling' refers to sequences; in both cases labelling of sites is arbitrary. Indeed, mutation labels do not even have to reside in $[0,1]$—in Figure 1.2 for example, mutations are labelled by non-negative integers, and the root by 0. The sequences in labelled trees may be ordered or unordered. Note that an unordered but labelled tree is not quite the same as an unlabelled

tree, since more than one labelling might correspond to the same unlabelled tree. For example, in the gene tree shown in Figure 1.2 there are $\frac{4!}{2!1!1!} = 12$ orderings of the sequences at the tips of the gene tree, but after unlabelling the sequences we find that each ordering has been counted twice; the two sequence types with one mutation are equivalent in an unlabelled tree. Further discussion on the distinction between ordered/unordered and labelled/unlabelled trees is given by Griffiths & Tavaré (1995) [12].

A gene tree can be viewed as a condensation from the full coalescent history, since temporal information of events is lost. Vertices are mutations, so we cannot deduce the relative ordering of coalescence events occurring between immediate descendants below the mutation. From the perspective of inferring histories from a given collection of sequences, the utility of the infinite-sites model is clear: conditioning upon a known rooted gene tree gives us a clear head-start.

1.2.4 The stepwise mutation model

In a stepwise mutation model, $E = \mathbb{Z}$ and $P_{ij} = u_{j-i}$, $i, j = 0 \pm 1, \pm 2, \ldots$; the rate of mutation from one allele to another depends only on the distance between the alleles. This type space can be used to represent the amount of charge on the resulting protein, whereby it was observed that certain technologies could detect mutations only through this effect [21]. More recently it has been used to model microsatellite repeats, which are amenable to a similar model of evolution (see for example [22, 23] in the context of importance sampling). In both cases a natural choice for P is

$$P_{ij} = \begin{cases} \frac{1}{2} & \text{if } |j - i| = 1, \\ 0 & \text{otherwise.} \end{cases}$$

1.3 The coalescent with recombination

Recombination is an important biological process to affect the ancestry of a sample, and can be incorporated into the Wright-Fisher model and thence the coalescent [6]. In eukaryotes recombination typically occurs during meiosis. In its absence, the formation of a haploid gamete from a diploid pair of chromosomes would entail simply selecting one of the homologous pair at random; recombination ensures that the gamete is instead formed of a mosaic of the two chromosomes. Recombination comprises both the process of gene conversion and of chromosomal crossover; I shall abuse terminology and always assume it refers to the latter. The former has also been incorporated into the coalescent [24, 25].

During gamete formation, a recombination event can occur somewhere along a chromosome at a *recombination breakpoint*. To the left of the breakpoint the gamete is formed from one of the corresponding diploid pair of chromosomes, and to the right it is formed from the other of the pair. Looking backwards in a Wright-Fisher model, this can be modelled as follows. Denote by r the probability of a recombination event per gene per generation. With probability $1 - r$ the gene chooses a single parent as usual, and with probability r it chooses two parents—these are the two genes that, via recombination, each supplied genetic material to the offspring. In the analysis of real data, recombination rates are often measured in *centiMorgans* (cM). Two positions are separated by 1 cM if in a single generation the probability of a recombination between them is 0.01.

Following an argument similar to that given for mutation above, the coalescent process is obtained and is now a birth and death process. Deaths and mutation events occur as before while there are k ancestors, at exponential rates $\binom{k}{2}$ and $k\frac{\theta}{2}$

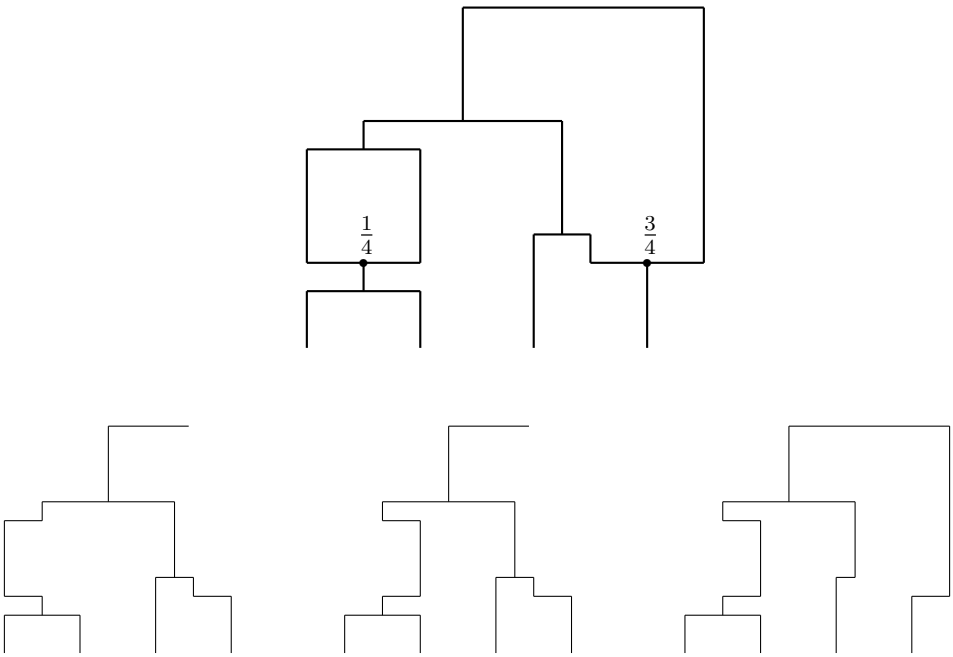


Figure 1.3: (*Top*): An ancestral recombination graph for a sample of four genes. Breakpoint positions are labelled. Embedded in this ARG are three marginal coalescent trees (*bottom*), for genetic material in $[0, \frac{1}{4}]$, $[\frac{1}{4}, \frac{3}{4}]$ and $[\frac{3}{4}, 1]$. Here each is traced back to the grand most recent common ancestor (GMRCA).

respectively, and births occur at an exponential rate $k\frac{\rho}{2}$, where $\rho = 4M_e r$ is the population-scaled recombination rate. A genealogy is no longer a bifurcating tree but is now a graph, known as the *ancestral recombination graph* (ARG) [26]. An example ARG is shown in Figure 1.3. It is convenient to represent the length of the gene as $[0, 1]$; then embedded in the ARG is a coalescent tree for each position $x \in [0, 1]$. To recover such a coalescent tree, one can trace lineages back from each leaf in the ARG. When we encounter a recombination event with breakpoint position $y \in [0, 1]$, go left if $x < y$, otherwise go right. The ARG is much more complicated than a coalescent tree, and inference is correspondingly much more difficult. Note that branches in the ARG need not be ancestral to *any* material in the extant sequences. The birth rate $k\frac{\rho}{2}$ includes recombination events that

occur within non-ancestral material, and so in this general formulation of the ARG there might be lineages which do not affect the ancestry of the sample (inference procedures can be made more efficient by modifying the birth rate according to the length of ancestral material; see Chapter 2). The MRCA for different positions is no longer necessarily the same individual, and indeed the GMRCA for the whole ARG need not coincide with any of the marginal MRCAs. So as $x \in [0, 1]$ is increased from 0 to 1, one encounters a collection of correlated coalescent trees for the genetic material at x, each embedded in the ARG. Indeed, this perspective can be used as a way to simulate ARGs from the coalescent process [27] as an alternative to the usual process of looking back in time [6]. When one simulates ARGs, a prescribed recombination breakpoint distribution Z on $[0, 1]$ is required. It is common to take a uniform distribution $Z \sim \mathrm{U}[0, 1]$, but another intriguing possibility is to consider the discrete distribution $Z \sim \mathrm{U}\left\{\frac{1}{m}, \ldots, \frac{m-1}{m}\right\}$, which gives rise to an m-locus model. By then letting $m \to \infty$ this can be used as a tool for deriving results on the infinite-sites model with recombination permitted to occur between any two sites, but it is also of interest in its own right [28]. The special case $m = 2$ is of particular interest [5, 29, 30, 31, 32].

1.4 Inference under the coalescent

A common estimation problem is one in which we'd like to estimate certain unknown parameters of the ancestral process that gave rise to the data. By assuming a model for this process it is possible to write down the likelihood as a function of the parameters, and, as we have seen, the coalescent provides a useful such model. For example, the mutation parameter θ could be estimated by the method of maximum

likelihood: a likelihood surface can be constructed by calculating $L(\theta)$ for various values of θ. I note in passing that this sort of inference can also be translated into a Bayesian framework. Instead of fixing θ we draw it from a prior distribution—for a description see Stephens (2001) [33]. For simplicity I shall focus on estimating θ as a goal of coalescent inference. The likelihood is

$$L(\theta) = p(\mathcal{D}|\,\theta) = \int p(\mathcal{D}|\,\mathcal{H},\theta)p(\mathcal{H}|\,\theta)\ \mathrm{d}\mathcal{H}, \tag{1.3}$$

where $\mathcal{D}$ is the observed data, and $\mathcal{H}$ is a coalescent history associated with it. In theory, by integrating over possible histories the right-hand side is straightforward to derive. $p(\mathcal{H})$ is known from the coalescent process, and $p(\mathcal{D}|\,\mathcal{H},\theta)$ is easy to calculate, depending on the form for $\mathcal{H}$: if $\mathcal{H}$ contains mutation information then it is simply an indicator function according to whether the leaves of the tree have the same configuration as the sample. If $\mathcal{H}$ is 'only' the genealogy then this quantity can still easily be calculated, for example by Felsenstein's peeling algorithm [34].

In practice, particularly for ARGs, the space of coalescent histories is prohibitively large, and we resort to a Monte Carlo approximation. A naïve approximation of (1.3) is thus

$$L(\theta) \approx \frac{1}{N}\sum_{i=1}^{N} p(\mathcal{D}|\,\mathcal{H}^{(i)},\theta), \tag{1.4}$$

where the $\mathcal{H}^{(i)}$ are independent samples from the distribution of $\mathcal{H}$—for brevity I shall denote the random variable whose outcomes are coalescent histories (under whatever specific model we have chosen) as $\mathcal{X}$. However, even for modestly-sized data, with extremely high probability each simulation will not even be consistent with $\mathcal{D}$. For this method to be accurate, the number of simulations N would be

exceedingly large. Importance sampling and Markov Chain Monte Carlo (MCMC) are two techniques recently applied in the field of population genetics which attempt to circumvent this problem, by sampling not from the distribution of $\mathcal{H}$ but from some other distribution that reduces the variance of each term in the summation in (1.4). Then each simulated run becomes relatively more important, and we do not have to reluctantly increase N as we wait for terms of any significance to appear during the course of the simulation. In this work we focus on importance sampling techniques; a summary of some alternatives is given in Section 1.4.2.

1.4.1 Importance sampling

Importance sampling (IS) [35] helps to overcome the problem described above by biasing simulated samples towards those of high probability, and weighting them accordingly. For example, a first-order correction would be only to simulate coalescent histories that are at least consistent with the data. Consider (suppressing the dependence on θ for convenience):

$$\begin{aligned} L(\theta) = \int p(\mathcal{D}|\,\mathcal{H})p(\mathcal{H})\ \mathrm{d}\mathcal{H} &= \int p(\mathcal{D}|\,\mathcal{H})\frac{p(\mathcal{H})}{q(\mathcal{H})}q(\mathcal{H})\ \mathrm{d}\mathcal{H} \\ &\approx \frac{1}{N}\sum_{i=1}^{N} p(\mathcal{D}|\,\mathcal{H}^{(i)})\frac{p(\mathcal{H}^{(i)})}{q(\mathcal{H}^{(i)})} =: \frac{1}{N}\sum_{i=1}^{N} w^{(i)}, \quad (1.5) \end{aligned}$$

where the $\mathcal{H}^{(i)}$ are drawn from $q(\mathcal{H})$, which can be any distribution on ancestries whose support includes $\{\mathcal{H} : p(\mathcal{D}|\,\mathcal{H}) > 0\}$. $q(\mathcal{H})$ is the *proposal distribution*, $w^{(i)}$ are the *IS weights*. The goal is to choose $q(\mathcal{H})$ in such a way as to reduce the variance of the estimator (1.5) as much as possible. $q(\mathcal{H})$ may depend on θ implicitly, which is known in this context as the *driving value*. The optimal choice for $q(\mathcal{H})$ would be

$q^*(\mathcal{H}) := p(\mathcal{H}|\,\mathcal{D})$; then every term in the sum (1.5) becomes

$$p(\mathcal{D}|\,\mathcal{H}^{(i)})\frac{p(\mathcal{H}^{(i)})}{p(\mathcal{H}^{(i)}|\,\mathcal{D})} = p(\mathcal{D}) \tag{1.6}$$

by Bayes' theorem, i.e. each term is exactly equal to the likelihood, and the variance is 0. An enterprising strategy then would be to try to choose $q(\mathcal{H})$ to match this optimal proposal as closely as possible. In general, obtaining $q^*(\mathcal{H})$ is as hard as the original problem.

Recall that interest lies in estimating a likelihood surface over a range of θ and thence the value of θ to maximize the likelihood. For efficiency, one can re-use the weighted collection of histories $((\mathcal{H}^{(1)}, w^{(1)}), \ldots, (\mathcal{H}^{(N)}, w^{(N)}))$ from a single driving value θ_0 by noting that

$$\hat{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N} p(\mathcal{D}|\,\mathcal{H}^{(i)}, \theta)\frac{p(\mathcal{H}^{(i)}|\,\theta)}{q(\mathcal{H}^{(i)})} = \frac{1}{N}\sum_{i=1}^{N} w^{(i)}\frac{p(\mathcal{H}^{(i)}|\,\theta)}{p(\mathcal{H}^{(i)}|\,\theta_0)} \tag{1.7}$$

is a likelihood estimate for $\theta \neq \theta_0$ [9]. However, remember that the proposal distribution assumes a mutation parameter θ_0 by design, so for θ far away from this driving value the likelihood estimate will be much poorer. A compromise is to perform IS independently on a set of points and then use interpolation, via a method such as bridge sampling [36] or kriging [23].

Importance sampling schemes also provide a method for ancestral inference. Suppose we are interested in estimating some property $h(\mathcal{H})$ of a history (for example

the TMRCA). Then

$$\begin{aligned}\mathbb{E}[h(\mathcal{X})|\,\mathcal{D}] &= \frac{\mathbb{E}[h(\mathcal{X})\mathbb{I}_{\{H_0=\mathcal{D}\}}(\mathcal{X})]}{\mathbb{E}[\mathbb{I}_{\{H_0=\mathcal{D}\}}(\mathcal{X})]} = \frac{\mathbb{E}[h(\mathcal{X})\mathbb{I}_{\{H_0=\mathcal{D}\}}(\mathcal{X})]}{p(\mathcal{D})} \\ &\approx \frac{\frac{1}{N}\sum_i h(\mathcal{H}^{(i)})w^{(i)}}{\frac{1}{N}\sum_j w^{(j)}} = \sum_i h(\mathcal{H}^{(i)})\left(\frac{w^{(i)}}{\sum_j w^{(j)}}\right), \qquad (1.8)\end{aligned}$$

where $\mathbb{I}_{\{H_0=\mathcal{D}\}}$ is an indicator function taking value 1 on those histories consistent with the data. It is worth making several remarks on equation (1.8). First, note that it is a result of Monte Carlo estimates of both the numerator and the denominator, re-using the same weights. The two estimates are of course not independent (which may incur a bias), but there is no reason why independent estimates—perhaps even based on different proposal distributions—may not be used, though I do not pursue this. Second, if $h(\mathcal{H}) = \mathbb{I}_{\mathcal{E}}(\mathcal{H})$, an indicator function for some event $\mathcal{E}$ associated with $\mathcal{H}$, then the left-hand side of (1.8) becomes $\mathbb{P}(\mathcal{E}|\,\mathcal{D})$; we thus have a way to estimate the probability of some event occurring in the history of the sequences. Third, since $\mathbb{E}[h(\mathcal{X})|\,\mathcal{D}] = \int h(\mathcal{H})p(\mathcal{H}|\,\mathcal{D})\,\mathrm{d}\mathcal{H}$, (1.8) tells us that the collection $\left(\mathcal{H}^{(1)}, ..., \mathcal{H}^{(N)}\right)$ with probability masses given by the normalized importance weights $\frac{w^{(i)}}{\sum_j w^{(j)}}$ is an empirical estimate of $p(\mathcal{H}|\,\mathcal{D})$.

Importance sampling techniques have been considered in some depth to investigate coalescent histories of a sample of genes under many different models, and in each case coalescent histories are generated *sequentially*. To explain the idea, first we clarify some notation. At each step in the recursion we obtain the next configuration back in time prior to the most recent event. Following Stephens & Donnelly (2000) [37], record each of these in the sequence $\mathcal{H} := (H_0, H_{-1}, \ldots, H_{-m})$, where H_0 is the configuration of the present and H_{-m} is the singleton MRCA (m may be

unknown, though it must be finite). The genes in H_k are unordered. $\mathcal{H}$ is defined as the *history* of the sample, and we assume it also contains information on the exact event that gives rise to each transition, wherever this cannot be inferred directly from the sequence $(H_0, H_{-1}, \ldots, H_{-m})$.

A natural class of proposal distributions on histories arises by randomly constructing histories *backwards* in time—then each simulated history is always consistent with $\mathcal{D}$; the support of $q(\mathcal{H})$ and $p(\mathcal{H}|\,\mathcal{D})$ are identical. Histories can be simulated from the distribution of $q(\mathcal{H})$ by prescribing backwards transition probabilities $q(H_{k-1}|\,H_k)$. The probabilities of interest can then be dealt with sequentially:

$$\begin{aligned} p(H_0) &= \sum_{\{H'_{-1}\}} \frac{p(H_0|\,H'_{-1})}{q(H'_{-1}|\,H_0)} q(H'_{-1}|\,H_0) p(H'_{-1}) \\ &= \mathbb{E}_q \left(\frac{p(H_0|\,H_{-1})}{q(H_{-1}|\,H_0)} p(H_{-1})|\,H_0 \right) \\ &= \mathbb{E}_q \left(\frac{p(H_0|\,H_{-1})}{q(H_{-1}|\,H_0)} \cdots \frac{p(H_{-m+1}|\,H_{-m})}{q(H_{-m}|\,H_{m+1})} p(H_{-m}) \right) \\ &= \mathbb{E}_q \left(\frac{p(\mathcal{H}_{\rightarrow})}{q(\mathcal{H}_{\leftarrow})/p(H_0)} \right), \end{aligned} \tag{1.9}$$

where H'_{-1} is a dummy variable summing over all states one event ago that could have given rise to the current configuration, and $p(\mathcal{H}_{\rightarrow})$, $q(\mathcal{H}_{\leftarrow})$ represent the probability of a history sample path evaluated as Markov chain probabilities in the forward and backwards directions respectively. Respective transition probabilities are p (from the coalescent process) and q (our proposal distribution). $\mathbb{E}_q$ denotes expectation with respect to q. The final expression in (1.9) expresses the expected weight concisely as a joint distribution on the data together with its history. The probability of the data is built into the numerator but not the denominator, which is corrected by dividing

by $p(H_0)$. Note that $\mathcal{H}$ is a sequence of intermediate ancestral configurations, and the time between events is not required to estimate the likelihood; these are conditionally independent given $\mathcal{H}$, and their densities can be integrated out. The integral in (1.3) then becomes a summation over sequences of intermediate states, which is a finite sum in the infinite-sites model (and in the absence of recombination).

The earliest implementation of an IS proposal distribution on coalescent histories was that of Griffiths & Tavaré (1994*a*) [9]. Their formulation was not in terms of importance sampling, but by renormalizing forward transition probabilities from a recursion for the probability $p(\boldsymbol{n})$ of a sample configuration $\boldsymbol{n}$. In other words, the proposal distribution is defined by

$$q(H_{k-1}|\, H_k) \propto p(H_k|\, H_{k-1}). \tag{1.10}$$

The connexion with IS was noted by Felsenstein *et al.* (1999) [38]. It was later observed that it is easy to construct situations in which this Griffiths-Tavaré scheme performs poorly, and more sophisticated approaches have had more success [37, 39]. On the other hand, its great appeal is its widespread applicability; if you can write down a recursion for the coalescent model of interest then you can perform importance sampling on it with this scheme. Examples of the application of the Griffiths-Tavaré scheme include finite-alleles data [9], infinite-sites data [10, 11, 12], stepwise mutation models [22], deterministically varying population size [10], recombination [40], subdivided population structure with migration [20], selection [41], Λ-coalescents [42], and epidemiological data [43].

The essential problem with the Griffiths-Tavaré scheme is that it is greedy, in the sense that it takes decisions of high (forward) probability early on which it

may then have to compensate for further back in the history. It maximizes the probability of the most recent part of the history $p(H_k|\, H_{k-1})$ at the expense of the term encompassing the rest of the probability, $p(H_{k-1})$. A more efficient proposal distribution, which redistributes 'attention' back towards the $p(H_{k-1})$ term was suggested by Stephens & Donnelly (2000) [37]. This was achieved by characterizing the backwards transition probabilities in terms of the function π, defined as follows: if $E = \{1, ..., d\}$ is the type space for a collection of n genes, with multiplicities $\boldsymbol{n} = (n_i)_{i \in E}$, then $\pi[i|\, \boldsymbol{n}]$ is the probability that an additional type chosen at random from the population is of type i given a sample configuration $\boldsymbol{n}$. Omitting recombination for now, backwards transition probabilities can be characterized in terms of what shall here-onwards be referred to as the *sampling distribution* π:

$$p(H_{k-1}|\, H_k) = \begin{cases} \dfrac{n_j - 1}{n + \theta - 1} \dfrac{n_j}{n} \dfrac{1}{\pi[j|\, \boldsymbol{n} - \boldsymbol{e}_j]} & \text{if } H_{k-1} = \boldsymbol{n} - \boldsymbol{e}_j, \\ \\ \dfrac{\theta P_{ij}}{n + \theta - 1} \dfrac{n_j}{n} \dfrac{\pi[i|\, \boldsymbol{n} - \boldsymbol{e}_j]}{\pi[j|\, \boldsymbol{n} - \boldsymbol{e}_j]} & \text{if } H_{k-1} = \boldsymbol{n} + \boldsymbol{e}_i - \boldsymbol{e}_j, \end{cases} \tag{1.11}$$

where $\boldsymbol{e}_i$ and $\boldsymbol{e}_j$ are unit vectors whose entries are all 0, except for the ith and jth respectively whose entries are 1. P is the transition matrix familiar from Section 1.2.1. Equation (1.11) is exact. It is obtained by applying Bayes' theorem

$$p(H_{k-1}|\, H_k) = p(H_k|\, H_{k-1}) \frac{p(H_{k-1})}{p(H_k)}, \tag{1.12}$$

and substituting for the right-hand fraction by applying the *symmetry condition*

$$\pi[i|\, \boldsymbol{n} - \boldsymbol{e}_j] p(\boldsymbol{n} - \boldsymbol{e}_j) = \frac{n_i + 1 - \delta_{ij}}{n} p(\boldsymbol{n} - \boldsymbol{e}_j + \boldsymbol{e}_i), \tag{1.13}$$

where δ_{ij} is the Kronecker delta. Equation (1.13) follows from the property of *exchangeability*: for an ordered configuration of types $\mathcal{A}_n = (a_1, \ldots, a_n)$ and any permutation σ of $\{1, \ldots, n\}$, we have that $p(a_{\sigma(1)}, \ldots, a_{\sigma(n)}) = p(a_1, \ldots, a_n)$. Thus the probability $p(\boldsymbol{n})$ of an unordered configuration is proportional to the probability of *any* corresponding ordered configuration: $p(\boldsymbol{n}) = \binom{n}{\boldsymbol{n}} p(\mathcal{A}_n)$, where $\binom{n}{\boldsymbol{n}}$ is the multinomial co-efficient $\frac{n!}{\prod_{i \in E} n_i!}$, the number of ordered configurations corresponding to $\boldsymbol{n}$. Equation (1.13) then follows by noting that both sides are equal to

$$\binom{n-1}{\boldsymbol{n} - \boldsymbol{e}_j} p(a_1, \ldots, a_{l-1}, a_{l+1}, \ldots, a_{n-1}, i),$$

where the omitted type is $a_l = j$.

We expect π to be intractable, otherwise from (1.11) we would have $p(\mathcal{H} | \, \mathcal{D})$ and hence $p(\mathcal{D})$. But by phrasing the problem in this way, good proposal distributions can be obtained by seeking good approximations to the distribution π. Stephens & Donnelly (2000) [37] propose such an approximation $\hat{\pi}$ given by

$$\hat{\pi}[j | \, \boldsymbol{n}] = \sum_{i \in E} \sum_{m=0}^{\infty} \frac{n_i}{n} \left(\frac{\theta}{n + \theta} \right)^m \frac{n}{n + \theta} \left(P^m \right)_{ij} . \tag{1.14}$$

$\hat{\pi}$ is the stationary distribution in a Markov chain with transition probability matrix $\frac{\theta P + U}{n + \theta}$, where U is the $d \times d$ matrix with each row equal to $\boldsymbol{n}$. $\hat{\pi}$ can be sampled from efficiently by interpreting (1.14) as follows: select a gene from the existing sample $\boldsymbol{n}$ uniformly at random and mutate it a Geometric$\left(\frac{n}{n+\theta}\right)$ number of times according to the transition matrix P. This captures the idea that the next sampled type will look similar, but not necessarily identical to, previously observed types. It transpires that $\hat{\pi}$ has a number of nice properties. For example, when substituted into (1.11)

to obtain a proposal distribution, the total backwards probability for a type i with multiplicity n_i is $\frac{n_i}{n}$. The IS procedure can then be implemented efficiently by selecting a gene uniformly at random to be involved in the next event back in time, and then evaluating $\hat{\pi}$ only for those types relevant to events involving this gene. Several justifications for the choice of $\hat{\pi}$ are given by De Iorio & Griffiths (2004*a*) [39], to which I shall return in detail in Chapter 2. The Stephens-Donnelly proposal distribution has been extended to include recombination by Fearnhead & Donnelly (2001) [36], who augmented it with a hidden Markov model running along positions of the sequence to describe the source sequence of the newly sampled gene. Thus, jumps in the hidden Markov model correspond to recombination events. I consider this approach in a two-locus, infinite-sites setting in Chapter 2.

When applied to the infinite-sites model, the Stephens-Donnelly proposal distribution is simplified by exploiting the property that, according to the proposal, each gene is equally likely to be involved in the next event back in time. When histories are of the form $H_k = (\mathcal{T}, \boldsymbol{n})$, *the choice of gene defines the event.* A gene can either coalesce with an identical copy (if its multiplicity is greater than 1) or lose its most recent mutation (if its multiplicity is 1), but not both, and—if its multiplicity is 1 and its most recent mutation is also observed on another lineage—possibly neither. The proposal distribution is therefore uniform on all possible immediate history changes going back in time: proceed by choosing a gene at random that could have been involved in the next event back in time, and undo that unique event. This is very easy to implement and leads to an efficient importance sampler. However, it can still be improved by noting that there is information in the gene tree telling us which of the set of possible events is more likely to have occurred most recently back in time. Looking at Figure 1.2 for example, there are three possible events

that could lead to the previous configuration: either a coalescence of the two identical genes, the removal of the mutation labelled 1, or the removal of the mutation labelled 2. Intuitively, one might argue that the coalescence is more likely to be the more recent event, simply because the resulting lineage has more events that it must subsequently be involved in before the MRCA can be reached. Recently, Hobolth *et al.* (2008) [44] have suggested a modified proposal distribution for infinite-sites data, which takes events further back in time into account by basing backwards proposal probabilities on exact results known for single sites [45].

1.4.2 Alternatives to importance sampling

Although the focus here is on importance sampling, it is worth noting that there are a number of other computationally-intensive approaches to similar problems, which are briefly reviewed below. I have focused on Markov Chain Monte Carlo (MCMC) techniques, the major alternative to IS, and omitted some 'faster' methods such as rejection algorithms and approximate Bayesian computation (ABC)—for a more detailed review see Marjoram & Tavaré (2006) [2]. In Section 1.4.2.2 I also discuss some recent heuristic attempts to cope with the specific complexities arising from the presence of recombination.

1.4.2.1 Markov Chain Monte Carlo

Markov Chain Monte Carlo (MCMC) is a general procedure for obtaining samples from an otherwise difficult to explore distribution of interest. For example, when confronted with ancestral inference problems of the form (1.8), we should be interested in the posterior distribution of genealogies given the data, $p(\mathcal{H}|\,\mathcal{D})$. MCMC

proceeds by constructing a Markov chain whose stationary distribution is the distribution of interest. Then an observation of the current state of the Markov chain sufficiently far into the future is a sample from this distribution. One can continue to evolve the same chain and sample it repeatedly at regular intervals. Provided the intervals are sufficiently large that successive samples are approximately independent, the collection of samples is still a sample from the posterior distribution, by ergodicity. The number of steps before sampling commences is the *burn-in*, and the interval between subsequent samplings is the *thinning interval*. These parameters are chosen such that independence of samples from the initial state and from each other approximately hold. The Markov chain is determined by a proposal distribution $q(\mathcal{H}, \mathcal{H}')$ that models the transition rate between any two states, from $\mathcal{H}$ to $\mathcal{H}'$. A common implementation is the Metropolis-Hastings algorithm [35], which permits almost any proposal distribution provided $q(\mathcal{H}, \mathcal{H}') > 0 \iff q(\mathcal{H}', \mathcal{H}) > 0$, and provided we introduce the additional acceptance step after drawing $\mathcal{H}'$ from $q(\mathcal{H}, \mathcal{H}')$:

- With probability $\min\left(1, \frac{p(\mathcal{H}'|\,\mathcal{D})}{p(\mathcal{H}|\,\mathcal{D})} \frac{q(\mathcal{H}', \mathcal{H})}{q(\mathcal{H}, \mathcal{H}')}\right)$ accept the proposed genealogy, otherwise set $\mathcal{H}' = \mathcal{H}$ and accept that.

The ratio $\frac{p(\mathcal{H}'|\,\mathcal{D})}{p(\mathcal{H}|\,\mathcal{D})}$ can be calculated even though, individually, the numerator and denominator cannot. Using the Metropolis-Hastings algorithm, MCMC on coalescent histories thus simplifies to choosing a good proposal distribution for exploring $p(\mathcal{H}|\,\mathcal{D})$. Unlike IS, MCMC transition probabilities are based on *local* moves from one topology to another. Well-designed transitions can combine relatively large local moves with relatively high acceptance probabilities, so that the posterior distribution is well-explored, and the subsequently thinned samples are independent.

An early example in the case of coalescent histories was that of Kuhner *et al.* (1995) [14], with modifications in Felsenstein *et al.* (1999) [38]. Given an existing coalescent tree, a new one is proposed as follows:

1. Choose a node in the tree uniformly at random and disconnect it from its parent, erasing the lineage between them.

2. Simulate a new lineage from the node backwards in time, which coalesces at a constant rate with each contemporaneous lineage (including that of the MRCA if it survives past the root of the tree).

As in the case of IS, this core algorithm can be extended to incorporate numerous other parameters (see Felsenstein *et al.* (1999) [38] and references therein), such as recombination [15]. A disadvantage of the above algorithm is that it ignores the likely allelic state of each lineage; conditional on the data, a lineage is more likely to coalesce with some than with others. Several other MCMC proposal distributions have also been suggested—for examples, see [46, 47, 2].

Even when the burn-in and thinning interval are set to be large, a disadvantage of MCMC by comparison with IS is that it is difficult to evaluate when the Markov chain has converged. Of course, IS has its own issues when it comes to diagnostic procedures—these are discussed with respect to my proposal distributions in Chapter 2.

1.4.2.2 Heuristic approaches to dealing with recombination

Under the standard coalescent model, datasets of reasonable size can prove to be difficult to handle in the context of likelihood inference. When recombination is included, the situation becomes even worse. One way to measure the complexity of

performing inference on a given dataset $\mathcal{D}$ is to enumerate all possible intermediate ancestral configurations (ACs) H_k appearing in any history $\mathcal{H}$ of the data. When recombination is introduced, the size of such a set grows several orders of magnitude more quickly than in its absence, as n and s grow [48]. However, as Song *et al.* (2006) [48] have noted, this growth can be curbed somewhat by considering the subset of ACs appearing only in ARGs with at most $R_{\min}(\mathcal{D})$ recombination events, the minimum possible number of recombination events associated with the data. Similarly, one can consider some reasonable $R \geq R_{\min}(\mathcal{D})$ to obtain a larger collapsed space of ACs. An approximate method for likelihood inference then suggests itself: evaluate the integral as in (1.3), but truncate the range of the integrand only to those histories that can be constructed from the set of ACs appearing in histories with at most R recombination events. This has been implemented by Lyngsø *et al.* (2008) [49], although it is currently limited to small datasets. The underlying idea of this approach is one of parsimony: it is expected that much of the total probability mass resides in histories constructed from ACs that are parsimonious in the number of recombination events; enough so that summing only over these histories is a reasonable approximation.

A popular alternative to importance sampling is the *product of approximate conditionals* (PAC) model of Li & Stephens (2003) [50]. The same sampling distribution π as appears in (1.11) is exploited, this time by writing the likelihood as

$$p(\mathcal{D}) = \pi[a_1]\pi[a_2|\, a_1] \ldots \pi[a_n|\, a_1, \ldots, a_{n-1}], \tag{1.15}$$

(suppressing dependence on the parameters of interest as usual), where $(a_1, \ldots, a_n)$ is an ordering of the unordered sample $\mathcal{D}$. A PAC-likelihood approximation can be

obtained by substituting an approximate sampling distribution $\hat{\pi}$ for π in (1.15), though a consequence is that the estimate now depends on the particular ordering chosen. On the other hand, this is a much faster method than IS in general, and circumventing the need to simulate a set of genealogies dispenses with the dependence on a genealogical model. In principle, any approximate sampling distribution $\hat{\pi}$ can be used, though it is propitious to incorporate coalescent ideas in its definition. A number of approximate sampling distributions are in existence. These include Ewens' sampling formula, which assumes an infinite-alleles model of mutation and at one locus is in fact exact [51]; sampling formulae based on those for single loci, coupled with a hidden Markov model to track the recombination process along the sequence [36, 50, 52, 53]; and an approximation derived from the general principles of the coalescent [39, 32].

Finally, it is worth mentioning some alternatives to (1.15), falling broadly under the heading of *composite (marginal) likelihoods* [54]. These attempt to approximate the full likelihood by a product of marginal densities of subsets of the data. They need not be treated as alternatives to IS, since it is often the case that the marginal densities themselves can be approximated using IS. Two important examples of composite likelihoods are:

1. The *pairwise likelihood* of Hudson (2001) [55], in which the likelihood is estimated by the product of the (scaled-)likelihoods of observing the configurations of each *pair* of sites, i.e. each pair is assumed to have been obtained from independent two-locus genealogies—and this likelihood is much easier to estimate. Due to its speed, this model has been extended successfully to additional models of mutation [56], and to obtain an estimate of the genetic map of the human genome [57, 58].

2. The *composite likelihood* of Fearnhead & Donnelly (2002) [59], in which the data $\mathcal{D}$ is split into R subregions $\mathcal{D}_1, \ldots, \mathcal{D}_R$, and the likelihood (as a function of both recombination and mutation parameters) is estimated by

$$L_C(\rho, \theta) = \prod_{r=1}^{R} p(\mathcal{D}_r | \, \rho, \theta). \tag{1.16}$$

This too has been extended for genomic-scale single nucleotide polymorphism (SNP) data for detecting recombination hotspots [60, 61]. Also employed in these studies is an *approximate marginal likelihood* [59], whereby genealogies for segregating sites below a certain minor allele frequency (MAF) threshold are not constructed by the IS procedure—instead they are approximated by the genealogies of neighbouring SNPs.

Since subregions in (1.16) are non-overlapping, the composite likelihood approximation is equivalent to assuming $\rho = \infty$ at the boundaries of each subregion. The composite likelihood is therefore well-suited to data for which recombination hotspot positions are known—a particular application of this is given in Chapter 3 (Section 3.3.2).

1.5 Diffusion processes

Before proceeding, we need one final piece of mathematical machinery. Diffusion processes in population genetics have a long history, pre-dating the coalescent, but for brevity I shall focus on them only in the context of their relevance to importance sampling, where they crop up in providing a justification for the development of IS proposal distributions [39]. Technical details are omitted. For further details on

diffusion models see for example Kimura (1964) [62].

A *diffusion process* $\{X_t\}_{t\geq 0}$ is a continuous space and time Markov process which traces out a continuous path as time evolves. It is determined by its *transition density function* $p : \mathbb{R}_+ \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}_+$, defined by

$$\mathbb{P}(X_t \in I \,|\, X_0 = x) = \int_I p(t, x, y)\, \mathrm{d}y,$$

$\forall I \subseteq \mathbb{R}$. Often this density is unknown, and instead a diffusion can be characterized by its *infinitesimal mean* and *infinitesimal variance*

$$\begin{aligned} \mu(x,t) &:= \lim_{\delta t \to 0} \frac{1}{\delta t} \mathbb{E}\left(\delta X_t \,|\, X_t = x\right), \\ \sigma^2(x,t) &:= \lim_{\delta t \to 0} \frac{1}{\delta t} \mathbb{E}\left[(\delta X_t)^2 \,|\, X_t = x\right], \end{aligned}$$

where $\delta X_t = X_{t+\delta t} - X_t$. We shall be interested only in *time-homogeneous* processes for which $\mu(x,t) = \mu(x)$ and $\sigma^2(x,t) = \sigma^2(x)$. These functions can be interpreted as approximately the mean and variance of the displacement δX_t during $(t, t+\delta t)$ for small δt, since it is easy to show that

$$\begin{aligned} \mathbb{E}\left(\delta X_t \,|\, X_t = x\right) &= \mu(x)\delta t + o(\delta t), \\ \mathrm{var}\left(\delta X_t \,|\, X_t = x\right) &= \sigma^2(x)\delta t + o(\delta t). \end{aligned}$$

Typically, diffusion processes are used to model the frequency of an allele in a large population as it changes in time. Such a process can be derived as a process from a Wright-Fisher model as $M \to \infty$. For example, in the simple case of two alleles A_1 and A_2, say, with symmetric mutation probability u, we can write $Y(k)$ for

the number of A_1 alleles at generation k. Then, by considering the scaled process $X_M(t) = \frac{Y(\lfloor 2Mt \rfloor)}{2M}$ for $t \geq 0$ we obtain

$$\begin{aligned}
&\mathbb{E}\left[X_M\left(t+\frac{1}{2M}\right) - X_M(t) \middle| X_M(t) = \frac{i}{2M}\right] \\
&\qquad = \frac{1}{2M}\mathbb{E}\left[Y\left(\lfloor 2Mt \rfloor + 1\right) - i \middle| Y\left(\lfloor 2Mt \rfloor\right) = i\right] \\
&\qquad = \frac{1}{2M}\left[2M\left(\frac{i}{2M}(1-u) + \frac{2M-i}{2M}u\right) - i\right] \\
&\qquad = \frac{1}{2M}2Mu\left(1-\frac{i}{M}\right) \\
&\qquad = \frac{1}{2M}\frac{\theta}{2}(1-2x),
\end{aligned}$$

where $\theta = 4Mu$, and $x = \frac{i}{2M}$. Thus, writing $\delta t = \frac{1}{2M}$ and letting $M \to \infty$ we get $\mu(x) = \frac{\theta}{2}(1-2x)$; in the same way, $\sigma^2(x) = x(1-x)$. Similar calculations for more complicated Wright-Fisher models incorporating a larger type space or multiple loci result in multi-dimensional diffusions with these infinitesimal moments still known. Higher-order moments are $o(\delta t)$.

A convenient representation is to consider, for some arbitrary $f : \mathbb{R} \to \mathbb{R}$ whose first- and second-order derivatives exist:

$$\begin{aligned}
\mathbb{E}\left[f(X_{t+\delta t}) \middle| X_t = x\right] &= \mathbb{E}\left[f(x+\delta X_t) \middle| X_t = x\right] \\
&= \left[f(x) + \mathbb{E}(\delta X) f'(x) + \frac{\mathbb{E}(\delta X^2)}{2} f''(x) + o(\delta t)\right] \\
&= f(x) + \left[\mu(x) f'(x) + \frac{\sigma^2(x)}{2} f''(x)\right] \delta t + o(\delta t), \quad (1.17)
\end{aligned}$$

and hence

$$\begin{aligned}\mathbb{E}\left[f(X_{t+\delta t}) - f(X_t)\right] &= \mathbb{E}\left[\mathbb{E}\left[f(X_{t+\delta t})\right] \mid X_t\right] - \mathbb{E}\left[f(X_t)\right] \\ &= \mathbb{E}\left[\mu(X_t)f'(X_t) + \frac{\sigma^2(X_t)}{2}f''(X_t)\right]\delta t + o(\delta t).\end{aligned}$$

Dividing by δt and letting $\delta t \to 0$:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}\left[f(X_t)\right] = \mathbb{E}\left[\mu(X_t)f'(X_t) + \frac{\sigma^2(X_t)}{2}f''(X_t)\right]. \tag{1.18}$$

Defining the *generator* of the diffusion as

$$\mathscr{L}f(x) := \lim_{\delta t \to 0} \frac{\mathbb{E}\left[f(X_{t+\delta t}) - f(x) \mid X_t = x\right]}{\delta t},$$

we have from (1.17) that $\mathscr{L}f(x) = \mu(x)f'(x) + \frac{\sigma^2(x)}{2}f''(x)$, and so (1.18) becomes

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}\left[f(X_t)\right] = \mathbb{E}\left[\mathscr{L}f(X_t)\right].$$

In particular, if the diffusion admits a stationary distribution X as $t \to \infty$, then

$$\mathbb{E}\left[\mathscr{L}f(X)\right] = 0. \tag{1.19}$$

By applying it to various functions f, equation (1.19) can prove very useful in obtaining results on the genealogical process, as we shall discover in the next chapter.

Chapter 2

Importance sampling on the coalescent with recombination

2.1 Introduction

In the previous chapter a number of approaches for designing IS proposal distributions were introduced. A method of particular interest is that developed by De Iorio & Griffiths (2004a) [39], since it is based on general mathematical principles and can be extended to a variety of models. Let us consider the method in more detail.

Assume a finite-alleles model of mutation (Section 1.2.1), so that an unordered sample is represented by the vector of multiplicities $\boldsymbol{n} = (n_i)_{i \in E}$, and we are interested in the quantity $p(\boldsymbol{n})$. A well-known recursion for $p(\boldsymbol{n})$ [9] is

$$\begin{aligned} p(\boldsymbol{n}) &= \frac{n-1}{n+\theta-1} \sum_{j \in E} \frac{n_j - 1}{n-1} p(\boldsymbol{n} - \boldsymbol{e}_j) \\ &\quad + \frac{\theta}{n+\theta-1} \sum_{j \in E} \sum_{i \in E} \frac{n_i + 1 - \delta_{ij}}{n} P_{ij} p(\boldsymbol{n} + \boldsymbol{e}_i - \boldsymbol{e}_j), \end{aligned} \tag{2.1}$$

with boundary conditions $p(\boldsymbol{e}_j) = \pi_j^*$ for $j \in E$. It is common to assume that $(\pi_j^*)_{j\in E}$ is the stationary distribution of P, the equilibrium distribution of the allele of a sampled gene in the absence of any further information.

Let $\mathcal{C}_j$ be the event that a gene of type j is the first to be involved in an event back in time. Then

$$\begin{aligned}\mathbb{P}(\boldsymbol{n}, \mathcal{C}_j) = p(\boldsymbol{n})\mathbb{P}(\mathcal{C}_j|\,\boldsymbol{n}) \;\; &= \;\; \frac{n-1}{n+\theta-1}\frac{n_j-1}{n-1}p(\boldsymbol{n}-\boldsymbol{e}_j) \\ &\quad + \frac{\theta}{n+\theta-1}\sum_{i\in E}\frac{n_i+1-\delta_{ij}}{n}P_{ij}p(\boldsymbol{n}+\boldsymbol{e}_i-\boldsymbol{e}_j),\end{aligned} \tag{2.2}$$

by simply partitioning on the possible first event back in time. An approximate sampling distribution $\hat{\pi}[j|\,\boldsymbol{n}]$ is obtained first by approximating $\mathbb{P}(\mathcal{C}_j|\,\boldsymbol{n})$ with

$$\widehat{\mathbb{P}}(\mathcal{C}_j|\,\boldsymbol{n}) = \frac{n_j}{n}, \tag{2.3}$$

and then assuming that (1.13) applies to $\hat{\pi}[j|\,\boldsymbol{n}]$ and its corresponding sample probability defined by

$$\hat{p}(\boldsymbol{n}) = \sum_{j\in E}\hat{\pi}[j|\,\boldsymbol{n}-\boldsymbol{e}_j]\hat{p}(\boldsymbol{n}-\boldsymbol{e}_j). \tag{2.4}$$

Substituting $\widehat{\mathbb{P}}(\mathcal{C}_j|\,\boldsymbol{n})$ for $\mathbb{P}(\mathcal{C}_j|\,\boldsymbol{n})$ and $\hat{p}$ for p in (2.2) yields:

$$n_j(n-1+\theta)\hat{p}(\boldsymbol{n}) = n(n_j-1)\hat{p}(\boldsymbol{n}-\boldsymbol{e}_j) + \theta\sum_{i\in E}(n_i+1-\delta_{ij})P_{ij}\hat{p}(\boldsymbol{n}+\boldsymbol{e}_i-\boldsymbol{e}_j). \tag{2.5}$$

Then substitute $\hat{\pi}$ for $\hat{p}$ using (1.13) to obtain the definition (1.14). One can also show [39] that if the symmetry condition *did* hold for $\hat{\pi}$ then we would have $\hat{p}(\boldsymbol{n}) = p(\boldsymbol{n})$.

We can see why the approximation (2.3) is reasonable. If P has stationary

distribution $(P_j)_{j\in E}$, then unconditional on the data the probability that the next event back in time occurs to a gene of type j is $\mathbb{P}(\mathcal{C}_j) = P_j$. This is because of the symmetry as to which gene is involved in the event and because P_j is the expected proportion of type j genes at a time in the sample's history, given no further information. Equation (2.3) can thus be seen as an approximation to the population allele frequencies using the sample.

De Iorio & Griffiths (2004*a*) [39] provide another way to obtain the approximation (1.14), via a consideration of the corresponding diffusion process of allele frequencies. Such frequencies $\boldsymbol{X} = (X_i)_{i\in E}$ are distributed according to the stationary distribution in a diffusion process with state space

$$K = \left\{ x = (x_i)_{i\in E} : \, x_i \geq 0, \sum_{i\in E} x_i = 1 \right\},$$

and generator

$$\mathscr{L} = \sum_{j\in E} L_j \frac{\partial}{\partial x_j}, \text{ where } L_j = \frac{1}{2}\sum_{i\in E} x_i(\delta_{ij} - x_j)\frac{\partial}{\partial x_i} + \frac{\theta}{2}\sum_{i\in E} x_i(P_{ij} - \delta_{ij}).$$

The sample probability is then a multinomial draw from $\boldsymbol{X}$:

$$p(\boldsymbol{n}) = \binom{n}{\boldsymbol{n}} \mathbb{E}\left(\prod_{i\in E} X_i^{n_i} \right). \tag{2.6}$$

By applying the generator equation (1.19) to $f(\boldsymbol{X}) := \prod_{i\in E} X_i^{n_i}$, one recovers (2.1). The technique for obtaining an approximate sampling distribution $\hat{\pi}[j\,|\,\boldsymbol{n}]$ is to assume that there exists a distribution for $\boldsymbol{X}$ with expectation operator $\widehat{\mathbb{E}}$ such that

(1.19) holds component-wise for the same $f(\boldsymbol{X})$; that is,

$$\widehat{\mathbb{E}}\left[L_j\left(\frac{\partial}{\partial X_j}\prod_{i\in E}X_i^{n_i}\right)\right]=0. \tag{2.7}$$

This can be simplified to obtain (2.5), where

$$\hat{p}(\boldsymbol{n})=\binom{n}{\boldsymbol{n}}\widehat{\mathbb{E}}\left(\prod_{i\in E}X_i^{n_i}\right).$$

By also assuming the symmetry condition (1.13) for $\hat{p}(\boldsymbol{n})$ and its associated $\hat{\pi}[j|\,\boldsymbol{n}]$, once again we obtain the Stephens-Donnelly approximate sampling distribution (1.14). Note that this method leads to the same approximation without invoking the coalescent model explicitly.

An advantage of the diffusion approach is that it can be applied to generators for a wide variety of models. As we have noted, when applied to the generator for the standard, neutral coalescent model mutating under a finite-alleles model, we recover the Stephens-Donnelly approximation. It has also been applied to the case with population structure [63], the stepwise mutation model [23], and to a two-locus model with recombination [32].

The goal of this chapter is to apply the techniques of De Iorio & Griffiths (2004*a*) [39] to a particular model of interest: data assumed to have been derived from the infinite-sites model with a single recombination breakpoint. As before, the technique results in an approximation to the sampling distribution $\pi[j|\,\boldsymbol{n}]$, the probability that when we sample an additional allele from the population it is of type j, given an existing sample configuration $\boldsymbol{n}$. When recombination is involved, there is still the question of how to deal with non-ancestral material efficiently, even after an approxi-

mate sampling distribution $\hat{\pi}[j|\,\boldsymbol{n}]$ has been obtained. This is an important problem, and one that also applies to the finite-alleles model of mutation. Before proceeding to the infinite-sites case, first I consider how to develop a proposal distribution in the finite-alleles setting that circumvents the need to integrate over non-ancestral lineages in putative ARGs (Section 2.2), given an approximate sampling distribution. A similar argument then permits the development of a proposal distribution in the infinite-sites setting too. This model is introduced (Section 2.3) and then a complete proposal distribution is developed using the technique of De Iorio & Griffiths (2004*a*) [39]. The section includes details of the computer implementation of this proposal (Section 2.3.4), techniques for improving and tracking its efficiency, and a comparison with existing IS proposal distributions that could apply to this model (Section 2.3.5). Finally, in Section 2.4 I discuss ways that the proposal could be extended and improved.

2.2 The two-locus, finite-alleles model

2.2.1 A recursion for the sample probability

We begin by introducing some notation. Recall that a gene is represented as the interval $[0, 1]$, and in a two-locus model the recombination breakpoint distribution is given by $\mathbb{P}\left(Z = \frac{1}{2}\right) = 1$. The two loci are then represented by $[0, \frac{1}{2}]$ and $[\frac{1}{2}, 1]$, which we denote respectively as A and B. Their respective transition matrices are P^{A} and P^{B}, with mutation parameters θ_{A} and θ_{B}, and type spaces $E_{A} = \{1, \ldots, d_{A}\}$ and $E_{B} = \{1, \ldots, d_{B}\}$. Griffiths *et al.* (2008) [32] apply the method of the previous section to this model to obtain an approximate sampling distribution $\hat{\pi}[(i, j)|\,\boldsymbol{n}]$,

conditional on the sample configuration $\boldsymbol{n} = (n_{ij})_{(i,j)\in E_A\times E_B}$. Their exposition was in terms of the diffusion approximation (2.7), but it could just as easily have been developed in terms of a coalescent argument based on the recursion (2.1). In the two-locus case this becomes

$$\begin{aligned} p(\boldsymbol{n}) &= \frac{n-1}{n-1+\theta_A+\theta_B+\rho}\sum_{(i,j)\in E_A\times E_B}\frac{n_{ij}-1}{n-1}p(\boldsymbol{n}-\boldsymbol{e}_{ij}) \\ &+ \frac{\theta_A}{n-1+\theta_A+\theta_B+\rho}\sum_{(i,j)\in E_A\times E_B}\sum_{k\in E_A}\frac{n_{kj}+1-\delta_{ik}}{n}P^A_{ki}p(\boldsymbol{n}+\boldsymbol{e}_{kj}-\boldsymbol{e}_{ij}) \\ &+ \frac{\theta_B}{n-1+\theta_A+\theta_B+\rho}\sum_{(i,j)\in E_A\times E_B}\sum_{l\in E_B}\frac{n_{il}+1-\delta_{jl}}{n}P^B_{lj}p(\boldsymbol{n}+\boldsymbol{e}_{il}-\boldsymbol{e}_{ij}) \\ &+ \frac{\rho}{n-1+\theta_A+\theta_B+\rho}\sum_{(i,j)\in E_A\times E_B}\sum_{(k,l)\in E_A\times E_B}\left(\frac{n_{ij}+\delta_{ik}\delta_{jl}}{n_{ij}}\frac{n_{il}+1-\delta_{jl}}{n+1}\right. \\ &\qquad \left.\times\frac{n_{kj}+1-\delta_{ik}}{n}p(\boldsymbol{n}+\boldsymbol{e}_{il}+\boldsymbol{e}_{kj}-\boldsymbol{e}_{ij})\right), \end{aligned} \tag{2.8}$$

where $\boldsymbol{e}_{ij}$ is a unit matrix whose (i,j)th entry is δ_{ij}, and so on. One way to obtain this recursion is to apply the generator equation (1.19) to

$$Q(\boldsymbol{n};\boldsymbol{X}) := \binom{n}{\boldsymbol{n}}\prod_{(i,j)\in E_A\times E_B}X_{ij}^{n_{ij}},$$

the probability of obtaining the (unordered) configuration $\boldsymbol{n}$ when n genes are sampled from a population with allele frequencies $\boldsymbol{X}$. The appropriate two-locus gen-

erator is $\mathscr{L} = \sum_{(i,j)\in E_A\times E_B} L_{ij}\frac{\partial}{\partial x_{ij}}$ given in [32], where

$$\begin{aligned} L_{ij} &= \frac{1}{2}\sum_{(k,l)\in E_A\times E_B} x_{ij}(\delta_{ik}\delta_{jl} - x_{kl})\frac{\partial}{\partial x_{kl}} + \frac{\theta_A}{2}\sum_{k\in E_A} x_{kj}(P^A_{ki} - \delta_{ki}) \\ &\quad + \frac{\theta_B}{2}\sum_{l\in E_B} x_{il}(P^B_{lj} - \delta_{lj}) + \frac{\rho}{2}\left(\sum_{(k,l)\in E_A\times E_B} x_{il}x_{kj} - x_{ij}\right). \end{aligned} \tag{2.9}$$

As a slight digression, perhaps a more succinct way to obtain (2.8) is to apply the generator equation to the *joint probability generating function* for the sample configuration instead. This is based on a suggestion of R.C. Griffiths who used a very similar approach for finding results on two-locus homozygosities in various two-locus models [5]. Some details, which will prove useful when we consider a related recursion later, are as follows.

Given population allele frequencies $\boldsymbol{x}$, the joint probability generating function for the number of each type in the sample is given by

$$G_n(\boldsymbol{s};\boldsymbol{x}) = \sum_{\{\hat{\boldsymbol{n}}:\,|\hat{\boldsymbol{n}}|=n,\,\hat{n}_{ij}\geq 0\}}\left[\prod_{(i,j)\in E_A\times E_B} s_{ij}^{\hat{n}_{ij}}\right] Q(\hat{\boldsymbol{n}};\boldsymbol{x}) = \left(\sum_{(i,j)\in E_A\times E_B} s_{ij}x_{ij}\right)^n,$$

where $\boldsymbol{s} = (s_{ij})$. The right-hand equality is a result of the multinomial theorem, and the algebra is much simpler when using this form. At stationarity we can apply

the generator equation (1.19) to $G_n(\boldsymbol{s};\boldsymbol{X})$:

$$\begin{aligned}
0 &= \mathbb{E}\left[\mathscr{L}G_n(\boldsymbol{s};\boldsymbol{X})\right] \\
&= \mathbb{E}\Bigg[\sum_{(i,j)\in E_A\times E_B}\Bigg(\frac{n(n-1)}{2}s_{ij}^2X_{ij}G_{n-2}(\boldsymbol{s};\boldsymbol{X}) + \frac{n\theta_A}{2}\sum_{k\in E_A}s_{ij}P_{ki}^AX_{kj}G_{n-1}(\boldsymbol{s};\boldsymbol{X}) \\
&\quad + \frac{n\theta_B}{2}\sum_{l\in E_B}s_{ij}P_{lj}^BX_{il}G_{n-1}(\boldsymbol{s};\boldsymbol{X}) + \frac{n\rho}{2}\sum_{(k,l)\in E_A\times E_B}s_{ij}X_{il}X_{kj}G_{n-1}(\boldsymbol{s};\boldsymbol{X})\Bigg) \\
&\quad - \frac{n(n-1+\theta_A+\theta_B+\rho)}{2}G_n(\boldsymbol{s};\boldsymbol{X})\Bigg],
\end{aligned}$$

which becomes, after some re-arrangement:

$$\begin{aligned}
&(n-1+\theta_A+\theta_B+\rho)\mathbb{E}[G_n(\boldsymbol{s};\boldsymbol{X})] = \qquad (2.10)\\
&\quad (n-1)\sum_{(i,j)\in E_A\times E_B}s_{ij}^2\sum_{\substack{\{\hat{\boldsymbol{n}}:|\hat{\boldsymbol{n}}|=n-2,\\ \hat{n}_{ij}\geq 0\}}}\left[\prod_{(q,r)\in E_A\times E_B}s_{qr}^{\hat{n}_{qr}}\right]\mathbb{E}[X_{ij}Q(\hat{\boldsymbol{n}};\boldsymbol{X})] \\
&\quad + \theta_A\sum_{(i,j)\in E_A\times E_B}\sum_{k\in E_A}s_{ij}P_{ki}^A\sum_{\substack{\{\hat{\boldsymbol{n}}:|\hat{\boldsymbol{n}}|=n-1,\\ \hat{n}_{ij}\geq 0\}}}\left[\prod_{(q,r)\in E_A\times E_B}s_{qr}^{\hat{n}_{qr}}\right]\mathbb{E}[X_{kj}Q(\hat{\boldsymbol{n}};\boldsymbol{X})] \\
&\quad + \theta_B\sum_{(i,j)\in E_A\times E_B}\sum_{l\in E_B}s_{ij}P_{lj}^B\sum_{\substack{\{\hat{\boldsymbol{n}}:|\hat{\boldsymbol{n}}|=n-1,\\ \hat{n}_{ij}\geq 0\}}}\left[\prod_{(q,r)\in E_A\times E_B}s_{qr}^{\hat{n}_{qr}}\right]\mathbb{E}[X_{il}Q(\hat{\boldsymbol{n}};\boldsymbol{X})] \\
&\quad + \rho\sum_{(i,j)\in E_A\times E_B}\sum_{(k,l)\in E_A\times E_B}s_{ij}\sum_{\substack{\{\hat{\boldsymbol{n}}:|\hat{\boldsymbol{n}}|=n-1,\\ \hat{n}_{ij}\geq 0\}}}\left[\prod_{(q,r)\in E_A\times E_B}s_{qr}^{\hat{n}_{qr}}\right]\mathbb{E}[X_{il}X_{kj}Q(\hat{\boldsymbol{n}};\boldsymbol{X})].
\end{aligned}$$

Note that

$$\mathbb{E}[G_n(\boldsymbol{s};\boldsymbol{X})] = \sum_{\{\hat{\boldsymbol{n}}:|\hat{\boldsymbol{n}}|=n,\,\hat{n}_{ij}\geq 0\}}\left[\prod_{(i,j)\in E_A\times E_B}s_{ij}^{\hat{n}_{ij}}\right]\mathbb{E}[Q(\hat{\boldsymbol{n}};\boldsymbol{X})]$$

is itself a probability generating function, since $\mathbb{E}[Q(\hat{\boldsymbol{n}}; \boldsymbol{X})]$ is the probability of obtaining a sample configuration $\hat{\boldsymbol{n}}$ when sampling from population allele frequencies at stationarity (as in equation (2.6)). The remaining expectations in (2.10) can be interpreted similarly.

Now, (2.10) holds for arbitrary $\boldsymbol{s}$, and so co-efficients of each polynomial in $\boldsymbol{s}$ must be equal. In particular, the co-efficients of $\prod_{(q,r)\in E_A\times E_B} s_{qr}^{n_{qr}}$ are $\mathbb{E}[Q(\boldsymbol{n}; \boldsymbol{X})]$—the probability of interest. Hence:

$$\begin{aligned} p(\boldsymbol{n}) &= \frac{n-1}{n-1+\theta_A+\theta_B+\rho} \sum_{(i,j)\in E_A\times E_B} \mathbb{E}[X_{ij} Q(\boldsymbol{n}-2\boldsymbol{e}_{ij}; \boldsymbol{X})] \\ &\quad + \frac{\theta_A}{n-1+\theta_A+\theta_B+\rho} \sum_{(i,j)\in E_A\times E_B} \sum_{k\in E_A} P^A_{ki} \mathbb{E}[X_{kj} Q(\boldsymbol{n}-\boldsymbol{e}_{ij}; \boldsymbol{X})] \\ &\quad + \frac{\theta_B}{n-1+\theta_A+\theta_B+\rho} \sum_{(i,j)\in E_A\times E_B} \sum_{l\in E_B} P^B_{lj} \mathbb{E}[X_{il} Q(\boldsymbol{n}-\boldsymbol{e}_{ij}; \boldsymbol{X})] \\ &\quad + \frac{\rho}{n-1+\theta_A+\theta_B+\rho} \sum_{(i,j)\in E_A\times E_B} \sum_{(k,l)\in E_A\times E_B} \mathbb{E}[X_{il} X_{kj} Q(\boldsymbol{n}-\boldsymbol{e}_{ij}; \boldsymbol{X})], \end{aligned} \tag{2.11}$$

where, by exchangeability in the sampling order,

$$\mathbb{E}[X_{ij} Q(\boldsymbol{n}-2\boldsymbol{e}_{ij}; \boldsymbol{X})] = \pi[(i,j)|\, \boldsymbol{n}-2\boldsymbol{e}_{ij}] p(\boldsymbol{n}-2\boldsymbol{e}_{ij}) = \frac{n_{ij}-1}{n-1} p(\boldsymbol{n}-\boldsymbol{e}_{ij}). \tag{2.12}$$

Similarly:

$$\mathbb{E}[X_{kj} Q(\boldsymbol{n}-\boldsymbol{e}_{ij}; \boldsymbol{X})] = \frac{n_{kj}+1-\delta_{ik}}{n} p(\boldsymbol{n}+\boldsymbol{e}_{kj}-\boldsymbol{e}_{ij}), \tag{2.13}$$

$$\mathbb{E}[X_{il} Q(\boldsymbol{n}-\boldsymbol{e}_{ij}; \boldsymbol{X})] = \frac{n_{il}+1-\delta_{jl}}{n} p(\boldsymbol{n}+\boldsymbol{e}_{il}-\boldsymbol{e}_{ij}), \tag{2.14}$$

$$\begin{aligned} \mathbb{E}[X_{il} X_{kj} Q(\boldsymbol{n}-\boldsymbol{e}_{ij}; \boldsymbol{X})] &= \frac{n_{ij}+\delta_{ik}\delta_{jl}}{n_{ij}} \frac{n_{il}+1-\delta_{jl}}{n+1} \\ &\quad \times \frac{n_{kj}+1-\delta_{ik}}{n} p(\boldsymbol{n}+\boldsymbol{e}_{il}+\boldsymbol{e}_{kj}-\boldsymbol{e}_{ij}). \end{aligned} \tag{2.15}$$

Substituting (2.12)–(2.15) into (2.11) recovers (2.8), as required.

2.2.2 Restricting the recursion to reduced ARGs

Given $\hat{\pi}[(i,j)|\,\boldsymbol{n}]$, a proposal distribution can then be defined in a manner analogous to the one-locus case (1.11). However, an inspection of equation (2.8) suggests that this naïve approach will result in a disastrously inefficient scheme; as we trace back through the recursion, no distinction is made between ancestral and non-ancestral material. Recombination events occurring in non-ancestral material result in one of the parents being *entirely* non-ancestral to the sample. The history of this lineage has no effect on the sample, and if possible we should like to integrate analytically over all possible histories of such lineages, by the principle of Rao-Blackwellization. A similar observation is made by Griffiths (1991) [31], who noted that in the two-locus case genes can be classified into one of four categories: ancestral only at locus A, ancestral only at locus B, ancestral at both loci, or ancestral at neither. Denote the number of each type of gene at time t back by $n_{A}(t)$, $n_{B}(t)$, $n_{C}(t)$, and $n_{D}(t)$ respectively. $(n_{A}(t), n_{B}(t), n_{C}(t), n_{D}(t))$ is a Markov jump process with known transition rates. The marginal transition rates of $(n_{A}(t), n_{B}(t), n_{C}(t))$ do not depend on $n_{D}(t)$, so this too is Markovian; we need not keep track of $n_{D}(t)$.

One way to measure the loss of efficiency of a proposal distribution that concerns itself with entire ARGs is to consider the number of recombination events, which we shall denote R_n^1 (and which is a random variable). Also denote the number of recombination events only occurring inside ancestral material as R_n, so that $R_n \leq R_n^1$. The total number of events in an ARG is $2R_n^1 + n - 1 + M_n$, where M_n is the random variable describing the total number of mutation events. The factor of

2 comes from the fact that for each recombination event an additional coalescence is also required, on top of the $n-1$ needed in the absence of recombination. For a given dataset, a reasonably efficient importance sampling scheme q will satisfy

$$\mathbb{E}_q\left(R_n^1 | \, \mathcal{D}\right) \approx \mathbb{E}\left(R_n^1 | \, \mathcal{D}\right). \tag{2.16}$$

To see this, note that an importance sampling scheme that is close to optimal will have a similar distribution to $q^*(\mathcal{H}) = p(\mathcal{H} | \, \mathcal{D})$, which satisfies

$$\mathbb{E}_{q^*}\left(R_n^1 | \, \mathcal{D}\right) = \sum_{k=0}^{\infty} k q^*(R_n^1 = k | \, \mathcal{D}) = \sum_{k=0}^{\infty} k \int_{\{\mathcal{H}:\, R_n^1 = k\}} p(\mathcal{H} | \, \mathcal{D}) \, \mathrm{d}\mathcal{H} = \mathbb{E}\left(R_n^1 | \, \mathcal{D}\right);$$

here, the approximation is exact. (Indeed, a stronger condition holds, since the two distributions for $R_n^1 | \, \mathcal{D}$ are the same. We are using only the first-order moment.) Of course, a given IS scheme will perform well for some datasets and poorly for others, and so in (2.16) we might take the expectation across datasets to get:

$$\mathbb{E}\left[\mathbb{E}_q\left(R_n^1 | \, \mathcal{D}\right)\right] \approx \mathbb{E}\left(R_n^1\right). \tag{2.17}$$

So, averaged across datasets, the number of recombination events we expect to observe in ARGs generated by the proposal distribution q should be similar to the number observed in ARGs generated from the coalescent process. If an importance sampler deviates from this equality, it indicates a bias in the number of recombination events it should be generating.

The point of all this is to argue that provided a proposal distribution is reasonably efficient such that (2.17) holds, its computational burden is closely reflected by the number of recombination events observed. A quantity of interest is therefore $\mathbb{E}\left(R_n^1\right)$.

Ethier & Griffiths (1990) [30] showed that

$$\mathbb{E}\left(R_n^1\right) = \rho \int_0^1 \frac{1-(1-x)^{n-1}}{x} e^{\rho x}\,\mathrm{d}x. \tag{2.18}$$

In particular, $\mathbb{E}\left(R_n^1\right) \sim e^{\rho}$ as $\rho \to \infty$. Less problematic is the result $\mathbb{E}\left(R_n^1\right) \sim \rho \log n$ for fixed ρ as $n \to \infty$, which also holds for R_n [26]. This slow growth with n can be understood by the intuition that increasing the sample size increases the rate of coalescence quadratically near the leaves, where branch lengths are shorter and recombination events are relatively less important. I note in passing that it is clear that equation (2.18) also applies when the breakpoint distribution is continuous. In this case an explicit expression for $\mathbb{E}(R_n)$ is also known ([17], provided material that has reached a common ancestor is also defined to be non-ancestral), whereas in the two-locus case $\mathbb{E}(R_n)$ is known only recursively—though an important result is that $\lim_{\rho\to\infty} \mathbb{E}(R_n) < \infty$ (Griffiths (1991) [31]).

For increasing ρ, an exponential increase in computational burden is clearly undesirable, compared to this bound on $\mathbb{E}(R_n)$. What we should really like is for our importance sampler to sample from a distribution of *reduced ARGs*. That is, to sample from a process which allows recombination events to occur only to genes carrying material ancestral to the sample. Monte Carlo schemes that simulate complete ARGs, such as that of Nielsen (2000) [47], clearly suffer from a not insubstantial disadvantage. Other schemes (such as [40, 36]) trace back only lineages carrying ancestral material, and it would be desirable for the one we develop here to do this too. A crucial observation is as follows: by sampling only from possible histories of genes carrying ancestral material (rather than complete ARGs), we have altered the forward probability of the data—in other words, $\boldsymbol{n}$ no longer satisfies the recursion

(2.8). We are no longer sampling from ARGs but equivalence classes of ARGs: they can be partitioned by the configuration of the lineages carrying ancestral material. Two ARGs $\mathcal{H}^{(i)}$, $\mathcal{H}^{(j)}$ are *equivalent* in this partition iff the embedded histories of genes carrying ancestral material are the same. An illustration is given in Figure 2.2, in which equivalence is denoted $\mathcal{H}^{(i)} \overset{1}{\sim} \mathcal{H}^{(j)}$. A way to sample from these reduced histories is to tag genes according to which loci are ancestral (as I mentioned above), and then consider the corresponding marginal recursion for the configuration only of genes carrying ancestral material. I develop an IS proposal distribution by this method in the next subsection, assuming that an approximate sampling distribution $\hat{\pi}[(i,j)|\,\boldsymbol{n}]$ has already been obtained; note that even though backwards transition rules are obtained from a suitably modified recursion (2.19) below, the approximation $\hat{\pi}[(i,j)|\,\boldsymbol{n}]$ can still be derived directly from (2.8). Also note that the following proposal distribution has also been published in Griffiths, Jenkins & Song (2008) [32]; throughout the thesis, references to this paper outside the next subsection are merely citations of it.

2.2.3 A proposal distribution

Denote a gene of type $(i,j) \in E_A \times E_B$ which is ancestral at locus A only, at locus B only, and at both loci as $(i,j)^A$, $(i,j)^B$, and $(i,j)^C$ respectively, with corresponding multiplicities n_{ij}^A, n_{ij}^B, and n_{ij}^C. The state space for genes in this system can then be denoted $(i,j,\gamma) \in E_A \times E_B \times \Gamma$, where $\Gamma = \{\mathrm{A}, \mathrm{B}, \mathrm{C}\}$ and γ indicates at which loci the gene is ancestral (A only, B only, or both). Define $\boldsymbol{n}_\gamma = (n_{ij}^\gamma)_{(i,j)\in E_A\times E_B}$ and $n^\gamma = \sum_{(i,j)\in E_A\times E_B} n_{ij}^\gamma$ for $\gamma \in \Gamma$, so that $\boldsymbol{n} = (\boldsymbol{n}_A, \boldsymbol{n}_B, \boldsymbol{n}_C)$ and $n = n^A + n^B + n^C$. $(\boldsymbol{n}_A(t), \boldsymbol{n}_B(t), \boldsymbol{n}_C(t))$ is thus a Markov process going backwards in time, with

transition rates defined according to the recursion below. Notice that in order to rein our computation in to a tractable form, this is at the expense of more complicated coalescence terms in the recursion; the ancestral statuses γ_1, γ_2 of two genes of type (i, j) are irrelevant to whether they can coalesce. There is therefore a term in the recursion for coalescences of all combinations of pairs. An equation for $p(\boldsymbol{n})$ under this reduced scheme is:

$$\begin{aligned} D_0 p(\boldsymbol{n}) \;=\; n \sum_{(i,j)\in E_A\times E_B} \Bigg[& \sum_{\gamma\in\Gamma} (n_{ij}^{\gamma}-1)p(\boldsymbol{n}-\boldsymbol{e}_{ij}^{\gamma}) + 2n_{ij}^{C}\left[p(\boldsymbol{n}-\boldsymbol{e}_{ij}^{A})+p(\boldsymbol{n}-\boldsymbol{e}_{ij}^{B})\right] \\ & + 2(n_{ij}^{C}+1)p(\boldsymbol{n}-\boldsymbol{e}_{ij}^{A}-\boldsymbol{e}_{ij}^{B}+\boldsymbol{e}_{ij}^{C}) \\ & + \theta_A \sum_{k\in E_A}\sum_{\gamma\in\Gamma} P_{ki}^{A}\frac{n_{kj}^{\gamma}+1-\delta_{ik}}{n} p(\boldsymbol{n}-\boldsymbol{e}_{ij}^{\gamma}+\boldsymbol{e}_{kj}^{\gamma}) \\ & + \theta_B \sum_{l\in E_B}\sum_{\gamma\in\Gamma} P_{lj}^{B}\frac{n_{il}^{\gamma}+1-\delta_{jl}}{n} p(\boldsymbol{n}-\boldsymbol{e}_{ij}^{\gamma}+\boldsymbol{e}_{il}^{\gamma}) \qquad (2.19) \\ & + \rho \sum_{(k,l)\in E_A\times E_B} \frac{(n_{il}^{A}+1)(n_{kj}^{B}+1)}{n\,(n+1)} p(\boldsymbol{n}+\boldsymbol{e}_{il}^{A}+\boldsymbol{e}_{kj}^{B}-\boldsymbol{e}_{ij}^{C}) \Bigg], \end{aligned}$$

where $D_0 = n(n-1)+n\theta_A+n\theta_B+\rho n^C$, $\boldsymbol{n}-\boldsymbol{e}_{ij}^{A}$ denotes $(\boldsymbol{n}_A-\boldsymbol{e}_{ij}, \boldsymbol{n}_B, \boldsymbol{n}_C)$, and so on. Equation (2.19) can be justified in a number of ways. A direct method is via the usual 'backwards-forwards' coalescent argument [9, 7], by conditioning on the most recent event going back in time. The six terms on the right-hand side correspond to the following events, summing over each event for all possible allelic states (i, j):

1. Coalescence of two identical types (i, j, γ), for each ancestral status $\gamma \in \Gamma$.

2. Coalescence of a type (i, j, C) with either (i, j, A) or (i, j, B). The resulting gene is of type (i, j, C), so there is a net loss of one gene of type (i, j, A) or (i, j, B) respectively.

3. Coalescence of a type (i, j, A) with (i, j, B). The resulting type is (i, j, C), since the ancestor is now ancestral at both loci.

4. Mutation of a type (i, j, γ) to (k, j, γ), for each $\gamma \in \Gamma$.

5. Mutation of a type (i, j, γ) to (i, l, γ), for each $\gamma \in \Gamma$.

6. Recombination of a type (i, j, C), resulting in additional types (i, j, A) and (i, j, B).

To justify the co-efficients for each event, consider as an example the term representing a mutation at locus A. By the rules of competing exponentials, the probability going backwards of the most recent event back in time being a mutation at locus A is $\frac{n\theta_A}{D_0}$. Suppose it occurred to a type (i, j, γ). Now, given that this is the most recent event, what is the probability going forwards that the correct mutation occurred to give rise to the present configuration $\boldsymbol{n}$? It is the probability that a gene picked uniformly at random from the previous configuration $\boldsymbol{n} - \boldsymbol{e}_{ij}^{\gamma} + \boldsymbol{e}_{kj}^{\gamma}$ is of the correct type (k, j, γ), and that it mutates forwards to a type (i, j, γ)—this probability is $P_{ki}^{A} \frac{n_{kj}^{\gamma} + 1 - \delta_{ik}}{n}$. Then we simply need to sum over all possible (i, j, γ) and possible previous configurations, times the probability of the previous configuration. Other terms can be dealt with in a similar way.

The co-efficients in (2.19) provide the correct forward probabilities. Given solutions for $\hat{\pi}[(i, j)^{\gamma} | \, \boldsymbol{n}]$, backwards transition probabilities $\hat{p}(H_{k-1} | \, H_k)$ can easily be defined using Bayes' theorem (1.12) and the symmetry condition (1.13). This extends the distribution (1.11) for the case with recombination, and by working from the modified recursion (2.19) we need no longer trace the lineages of entirely non-ancestral genes. The forward transition probabilities and the ratio $\frac{\hat{p}(H_{k-1})}{\hat{p}(H_k)}$ for this

scheme are shown in Table 2.1, whence the backward transition probabilities and IS weights can be derived (e.g. the importance weights are simply the reciprocal of $\frac{\hat{p}(H_{k-1})}{\hat{p}(H_k)}$).

We denote the subsequent sampling of *two* genes by

$$\begin{aligned}\pi[\{(i,l)^A,(k,j)^B\}|\,\boldsymbol{n}] &:= \pi[(i,l)^A|\,\boldsymbol{n}]\pi[(k,j)^B|\,\boldsymbol{n}+\boldsymbol{e}_{il}^A] \\ &= \pi[(k,j)^B|\,\boldsymbol{n}]\pi[(i,l)^A|\,\boldsymbol{n}+\boldsymbol{e}_{kj}^B], \end{aligned} \tag{2.20}$$

the approximation of which, $\hat{\pi}[\{(i,l)^A,(k,j)^B\}|\,\boldsymbol{n}]$, appears in Table 2.1. In general, exchangeability in the sampling order of the two genes in (2.20) will *not* hold for the approximation—a point also noted by Stephens & Donnelly (2000) [37]—and so we propose to use the following symmetrized definition:

$$\begin{aligned}\hat{\pi}[\{(i,l)^A,(k,j)^B\}|\,\boldsymbol{n}] &= \frac{1}{2}\Big[\hat{\pi}[(i,l)^A|\,\boldsymbol{n}]\hat{\pi}[(k,j)^B|\,\boldsymbol{n}+\boldsymbol{e}_{il}^A] \\ &\qquad + \hat{\pi}[(k,j)^B|\,\boldsymbol{n}]\hat{\pi}[(i,l)^A|\,\boldsymbol{n}+\boldsymbol{e}_{kj}^B]\Big]. \end{aligned} \tag{2.21}$$

Note that this symmetrization procedure also appears when solving the system of equations for $\hat{\pi}[(i,j)|\,\boldsymbol{n}]$ in [32].

Finally, Table 2.1 assumes that we have solutions for $\hat{\pi}[(i,j)^\gamma|\,\boldsymbol{n}]$, $\gamma\in\Gamma$, when we only have solutions $\hat{\pi}[(i,j)|\,\boldsymbol{n}]$. But of course the sampling distribution of types (i,j) should not depend on our assignation of which loci are ancestral; we therefore use

$$\hat{\pi}[(i,j)^A|\,\boldsymbol{n}] = \hat{\pi}[(i,j)^B|\,\boldsymbol{n}] = \hat{\pi}[(i,j)^C|\,\boldsymbol{n}] = \hat{\pi}[(i,j)|\,\boldsymbol{n}].$$

H_{k-1}	$p(H_k \mid H_{k-1})$	$\frac{\hat{p}(H_{k-1})}{\hat{p}(H_k)}$
Coalescence :		
$\boldsymbol{n} - \boldsymbol{e}_{ij}^A$	$\frac{n(2n_{ij}^C + n_{ij}^A - 1)}{D_0}$	$\frac{n_{ij}^A}{n} \cdot \frac{1}{\hat{\pi}[(i,j)^A \mid \boldsymbol{n} - \boldsymbol{e}_{ij}^A]}$
$\boldsymbol{n} - \boldsymbol{e}_{ij}^B$	$\frac{n(2n_{ij}^C + n_{ij}^B - 1)}{D_0}$	$\frac{n_{ij}^B}{n} \cdot \frac{1}{\hat{\pi}[(i,j)^B \mid \boldsymbol{n} - \boldsymbol{e}_{ij}^B]}$
$\boldsymbol{n} - \boldsymbol{e}_{ij}^C$	$\frac{n(n_{ij}^C - 1)}{D_0}$	$\frac{n_{ij}^C}{n} \cdot \frac{1}{\hat{\pi}[(i,j)^C \mid \boldsymbol{n} - \boldsymbol{e}_{ij}^C]}$
$\boldsymbol{n} - \boldsymbol{e}_{ij}^A - \boldsymbol{e}_{ij}^B + \boldsymbol{e}_{ij}^C$	$\frac{2n(n_{ij}^C + 1)}{D_0}$	$\frac{n_{ij}^A n_{ij}^B \hat{\pi}[(i,j)^C \mid \boldsymbol{n} - \boldsymbol{e}_{ij}^A - \boldsymbol{e}_{ij}^B]}{n(n_{ij}^C + 1)\hat{\pi}[\{(i,j)^A, (i,j)^B\} \mid \boldsymbol{n} - \boldsymbol{e}_{ij}^A - \boldsymbol{e}_{ij}^B]}$
Mutation :		
$\boldsymbol{n} - \boldsymbol{e}_{ij}^A + \boldsymbol{e}_{kj}^A$	$\frac{\theta_A P_{ki}^A (n_{kj}^A + 1 - \delta_{ik})}{D_0}$	$\frac{n_{ij}^A}{n_{kj}^A + 1 - \delta_{ik}} \cdot \frac{\hat{\pi}[(k,j)^A \mid \boldsymbol{n} - \boldsymbol{e}_{ij}^A]}{\hat{\pi}[(i,j)^A \mid \boldsymbol{n} - \boldsymbol{e}_{ij}^A]}$
$\boldsymbol{n} - \boldsymbol{e}_{ij}^B + \boldsymbol{e}_{kj}^B$	$\frac{\theta_A P_{ki}^A (n_{kj}^B + 1 - \delta_{ik})}{D_0}$	$\frac{n_{ij}^B}{n_{kj}^B + 1 - \delta_{ik}} \cdot \frac{\hat{\pi}[(k,j)^B \mid \boldsymbol{n} - \boldsymbol{e}_{ij}^B]}{\hat{\pi}[(i,j)^B \mid \boldsymbol{n} - \boldsymbol{e}_{ij}^B]}$
$\boldsymbol{n} - \boldsymbol{e}_{ij}^C + \boldsymbol{e}_{kj}^C$	$\frac{\theta_A P_{ki}^A (n_{kj}^C + 1 - \delta_{ik})}{D_0}$	$\frac{n_{ij}^C}{n_{kj}^C + 1 - \delta_{ik}} \cdot \frac{\hat{\pi}[(k,j)^C \mid \boldsymbol{n} - \boldsymbol{e}_{ij}^C]}{\hat{\pi}[(i,j)^C \mid \boldsymbol{n} - \boldsymbol{e}_{ij}^C]}$
$\boldsymbol{n} - \boldsymbol{e}_{ij}^A + \boldsymbol{e}_{il}^A$	$\frac{\theta_B P_{lj}^B (n_{il}^A + 1 - \delta_{jl})}{D_0}$	$\frac{n_{ij}^A}{n_{il}^A + 1 - \delta_{jl}} \cdot \frac{\hat{\pi}[(i,l)^A \mid \boldsymbol{n} - \boldsymbol{e}_{ij}^A]}{\hat{\pi}[(i,j)^A \mid \boldsymbol{n} - \boldsymbol{e}_{ij}^A]}$
$\boldsymbol{n} - \boldsymbol{e}_{ij}^B + \boldsymbol{e}_{il}^B$	$\frac{\theta_B P_{lj}^B (n_{il}^B + 1 - \delta_{jl})}{D_0}$	$\frac{n_{ij}^B}{n_{il}^B + 1 - \delta_{jl}} \cdot \frac{\hat{\pi}[(i,l)^B \mid \boldsymbol{n} - \boldsymbol{e}_{ij}^B]}{\hat{\pi}[(i,j)^B \mid \boldsymbol{n} - \boldsymbol{e}_{ij}^B]}$
$\boldsymbol{n} - \boldsymbol{e}_{ij}^C + \boldsymbol{e}_{il}^C$	$\frac{\theta_B P_{lj}^B (n_{il}^C + 1 - \delta_{jl})}{D_0}$	$\frac{n_{ij}^C}{n_{il}^C + 1 - \delta_{jl}} \cdot \frac{\hat{\pi}[(i,l)^C \mid \boldsymbol{n} - \boldsymbol{e}_{ij}^C]}{\hat{\pi}[(i,j)^C \mid \boldsymbol{n} - \boldsymbol{e}_{ij}^C]}$
Recombination :		
$\boldsymbol{n} - \boldsymbol{e}_{ij}^C + \boldsymbol{e}_{il}^A + \boldsymbol{e}_{kj}^B$	$\frac{\rho(n_{il}^A + 1)(n_{kj}^B + 1)}{(n+1)D_0}$	$\frac{n_{ij}^C (n+1)\hat{\pi}[\{(i,l)^A, (k,j)^B\} \mid \boldsymbol{n} - \boldsymbol{e}_{ij}^C]}{(n_{il}^A + 1)(n_{kj}^B + 1)\hat{\pi}[(i,j)^C \mid \boldsymbol{n} - \boldsymbol{e}_{ij}^C]}$

Table 2.1: Forward transition probabilities $p(H_k \mid H_{k-1})$ and the ratio $\frac{\hat{p}(H_{k-1})}{\hat{p}(H_k)}$ for a two-locus, finite-alleles model. The constant D_0 is defined as $D_0 = n(n-1) + n\theta_A + n\theta_B + \rho n^C$, and the multiplicity of H_k is $\boldsymbol{n} = (\boldsymbol{n}^A, \boldsymbol{n}^B, \boldsymbol{n}^C)$.

From the resulting known backwards transition probabilities, an importance sampling scheme is defined. By tracking the allelic state at *all* loci of genes carrying ancestral material, we do not have to resort to imputing types at non-ancestral loci.

The scheme developed above has the advantage that we needn't concern ourselves with the history of genes that interact with genes ancestral to the sample but which are not themselves ancestral. Even better would be not to concern ourselves with *loci* that are not ancestral to the sample. In order to achieve this, we could further reduce the state space of ARGs by considering *fragments* of genes. Then each fragment is ancestral wherever it is defined, and we have a coarser partition on ARGs: Two ARGs $\mathcal{H}^{(i)}$, $\mathcal{H}^{(j)}$ are *equivalent* in this partition iff the embedded histories of ancestral material are the same (thus, two ARGs equivalent under the partition defined in the previous section must also be equivalent in this one). In Figure 2.2 this equivalence is denoted $\mathcal{H}^{(i)} \overset{2}{\sim} \mathcal{H}^{(j)}$. Further notation for this partition will be developed in the next section. First, note that we could have attempted to develop a proposal distribution under this state space for the finite-alleles model considered above. However, this is at the expense of simplicity—the coalescence terms become even more complicated, and it becomes correspondingly more difficult to derive a sampling distribution $\hat{\pi}$ for each possible type. However, there is one situation in which introducing a state space of fragments is vital, which is the motivating model mentioned originally. Modify the current model so that each locus mutates under the infinite-sites assumption. Now the type space for a new allele at each locus is countably infinite; it will have some combination of previously observed segregating sites plus some natural number of new mutations. Why can't we just use the same scheme? If, as in the previous scheme, we attempted to assign alleles to non-ancestral loci arising from recombination events, we should have to consider the

distribution across this infinite sum. On the other hand, a scheme operating under a fragment state space need consider only the sampling distribution for the observed and inferred types of a given sample—and for any given sample this list is finite. Hence, reducing the state space of ARGs in this example becomes an appropriate way to simplify the problem, from approximating a probability distribution with infinite support to one with finite support. The price we pay is having to deal with a much larger collection of possible events going back in time, as different types of fragment can coalesce with each other. This results in a proposal distribution with a rather excessive amount of fine details, but which ultimately is straightforward to implement, as we shall discover in the next section. First, we elaborate on a description of the model.

2.3 The two-locus, infinite-sites model

The two-locus, infinite-sites model was studied by Griffiths (1981) [5], and will be a particular focus here. For brevity I shall refer to it as the G_{81} model. To recap, each locus is regarded as a sequence of completely linked sites mutating under the infinite-sites model; there is no recombination within loci. It can be thought of as a limiting process of a two-locus, finite-sites model using the notation of Section 2.2 with L sites, and then letting $L \to \infty$. At a single locus, infinite-sites data is equivalent to a rooted gene tree. At two loci, the data is equivalent to two marginal gene trees. From a graph-theoretical perspective, nothing further can be inferred about the way the two trees are related, other than that they are connected at the tips. Further back than the present, an immediate recombination on each branch cannot be ruled out, whereupon the two gene trees could have independent histories.

In this construction, mutations are assumed to be labelled by the loci in which they reside, but because there is no recombination within loci, mutation positions can otherwise be ignored.

The emphasis on this model can be justified in a number of ways. First is its mathematical simplicity. We shall only ever work with two correlated gene trees, which is easier to handle than the random number of gene trees that follows from a more complicated breakpoint distribution. Of course, employment of this model depends on the data with which we are working. It is most reasonable when recombination within loci is much less than between loci and when the infinite-sites assumption may be employed—for example, a region with a narrow, hot recombination hotspot compared to a low background rate, and whose flanking genetic material is long with a low mutation rate. Second, gene trees are quite informative about the history of the data compared to other models of mutation. Looking at the way two gene trees are entwined allows us to focus on particular questions of ancestral inference. For example, what can be deduced about the joint ancestry of the two loci? Which branches in the ARG are shared by both loci? What is the probability that the MRCA of the two loci are the same individual? Questions such as these are generally not well-answered, and I shall return to them in Chapter 4. Third, until recently it was not known quite how applicable to real data this model would be. But several studies of the last few years have shown substantial variation in fine-scale recombination rate [57, 58, 64]. For example, Myers *et al.* (2005) [58] identify more than $25,000$ hotspots across the human genome, where a hotspot is defined to be a 200 kb sequence window for which a likelihood ratio test rejects the hypothesis that a central 2 kb region does not have an elevated local recombination rate relative to that of its surroundings. Given the difficulties in performing

inference on realistically-sized datasets under a process as complicated as the ARG, this study utilized a composite likelihood method based on the pairwise likelihood of Hudson (2001) [55] (see also Section 1.4.2.2). Notice that given an IS proposal distribution on the G_{81} model, the pairwise likelihood is easily extended from pairs of sites to pairs of *loci*—by which we mean blocks of segregating sites for which we might have an independent belief that recombination within the block is absent. In any case, in light of the complicated nature of inference on the ARG, focusing on data which is well-modelled by the G_{81} model is a good place to start.

The program `ms` [65] provides a way to simulate data from the ARG, with switches available for it to conform to the G_{81} model. Throughout this thesis I shall make regular use of `ms` for simulation studies, and will denote by $\texttt{ms}(n, \theta, \rho)$ the random variable whose outcomes are datasets drawn from `ms` under the G_{81} model with sample size n and mutation and recombination parameters θ and ρ respectively. Unless otherwise stated, the mutation parameters θ_A, θ_B for the two loci are equal (and $\theta_A + \theta_B = \theta$).

2.3.1 A recursion for the sample probability

To emphasize the different nature of fragments and half-fragments we will follow Ethier & Griffiths (1990) [30] and use a, b, c rather than n^A, n^B, n^C for the multiplicities of each type: denote a gene of type i at locus A which is ancestral only at locus A by $(i, *)$; a gene of type j at locus B which is ancestral only at locus B by $(*, j)$; and a gene of types i, j at each locus respectively, ancestral at both loci, by (i, j). Denote their corresponding multiplicities a_i, b_j, c_{ij}. Define $\boldsymbol{a} = (a_i)_{i \in I_A}$, $\boldsymbol{b} = (b_j)_{j \in I_B}$, $\boldsymbol{c} = (c_{ij})_{(i,j) \in I_A \times I_B}$, with $\boldsymbol{n} = (\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c})$ and $n = a + b + c$,

where $a = \sum_{i \in I_A} a_i$, $b = \sum_{(i,j) \in I_B} b_j$, $c_{i\cdot} = \sum_{j \in I_B} c_{ij}$ and $c_{\cdot j} = \sum_{i \in I_A} c_{ij}$. These summations should really be over the uncountably infinite type space described in Section 1.2.3. However, we shall only ever work with some previously observed dataset with n sequences and s segregating sites, in which case we can take I_A and I_B to index only the list of observed and inferred types associated with this dataset. Inferred types are states associated with nodes in the gene tree at one of the loci, but which are not observed in the sample directly. I_A and I_B are then finite sets. Each entry in the index corresponds to a path $(x_0, x_1, \ldots)$ to the root, and the dataset is equivalent to the pair of marginal gene trees $\mathcal{T} = (\mathcal{T}_A, \mathcal{T}_B)$ with multiplicity $\boldsymbol{n}$. In this construction, we will usually observe an initial dataset $\boldsymbol{n} = (\mathbf{0}, \mathbf{0}, \boldsymbol{c})$, but sequences with missing data at a locus can be accommodated easily using $\boldsymbol{a}$ or $\boldsymbol{b}$. The IS proposal distribution we are aiming for will also handle missing data in this way. To develop a proposal distribution, the method of De Iorio & Griffiths (2004*a*) [39] detailed in Section 2.1 suggests working directly from a recursion for $(\mathcal{T}, \boldsymbol{n})$. The recursion we shall work with will be for the quantity $p(\mathcal{T}, \boldsymbol{n})$ defined by

$$p(\mathcal{T}, \boldsymbol{n}) = \binom{n}{\boldsymbol{n}} q(\mathcal{T}, \boldsymbol{n}), \tag{2.22}$$

where $q(\mathcal{T}, \boldsymbol{n})$ is the probability of observing an ordered, labelled sample $(\mathcal{T}, \boldsymbol{n})$. Note that unless $a = b = 0$, $p(\mathcal{T}, \boldsymbol{n})$ is not exactly the probability of an unordered dataset—which would be $\binom{a}{\boldsymbol{a}}\binom{b}{\boldsymbol{b}}\binom{c}{\boldsymbol{c}} q(\mathcal{T}, \boldsymbol{n})$. Instead, we have 'thrown' fragments and half-fragments together, a consequence of which is that it is possible to have $p(\mathcal{T}, \boldsymbol{n}) > 1$, as in a similar recursion considered by Griffiths & Marjoram (1996) [40]. Although we could work with $q(\mathcal{T}, \boldsymbol{n})$ or $\binom{a}{\boldsymbol{a}}\binom{b}{\boldsymbol{b}}\binom{c}{\boldsymbol{c}} q(\mathcal{T}, \boldsymbol{n})$, we choose to work with (2.22) so that terms in its recursion are analogous to those of equation (2.19)

and the recursion in [40], retaining conceptual clarity.

An equation satisfied by $p(\mathcal{T}, \boldsymbol{n})$ is:

$$\begin{aligned}
Dp(\mathcal{T}, \boldsymbol{n}) &= n \sum_{i:\, a_i \geq 1} (a_i + 2c_{i\cdot} - 1) p(\mathcal{T}, \boldsymbol{n} - \boldsymbol{e}_i^A) \\
&\quad + n \sum_{j:\, b_j \geq 1} (b_j + 2c_{\cdot j} - 1) p(\mathcal{T}, \boldsymbol{n} - \boldsymbol{e}_j^B) \\
&\quad + n \sum_{(i,j):\, c_{ij} \geq 2} (c_{ij} - 1) p(\mathcal{T}, \boldsymbol{n} - \boldsymbol{e}_{ij}^C) \\
&\quad + n \sum_{i:\, a_i \geq 1} \sum_{j:\, b_j \geq 1} 2(c_{ij} + 1) p(\mathcal{T}, \boldsymbol{n} - \boldsymbol{e}_i^A - \boldsymbol{e}_j^B + \boldsymbol{e}_{ij}^C) \\
&\quad + \theta_A \sum_{\substack{i:\, a_i = 1, \\ c_{i\cdot} = 0, i \to k}} (a_k + 1) p(\mathcal{T}'_{i-}, \boldsymbol{n} - \boldsymbol{e}_i^A + \boldsymbol{e}_k^A) \\
&\quad + \theta_A \sum_{\substack{i:\, a_i = 0, \exists j: \\ c_{ij} = c_{i\cdot} = 1, i \to k}} (c_{kj} + 1) p(\mathcal{T}'_{i-}, \boldsymbol{n} - \boldsymbol{e}_{ij}^C + \boldsymbol{e}_{kj}^C) \\
&\quad + \theta_B \sum_{\substack{j:\, b_j = 1, \\ c_{\cdot j} = 0, j \to l}} (b_l + 1) p(\mathcal{T}'_{j-}, \boldsymbol{n} - \boldsymbol{e}_j^B + \boldsymbol{e}_l^B) \\
&\quad + \theta_B \sum_{\substack{j:\, b_j = 0, \exists i: \\ c_{ij} = c_{\cdot j} = 1, j \to l}} (c_{il} + 1) p(\mathcal{T}'_{j-}, \boldsymbol{n} - \boldsymbol{e}_{ij}^C + \boldsymbol{e}_{il}^C) \\
&\quad + \frac{\rho}{n+1} \sum_{(i,j):\, c_{ij} \geq 1} (a_i + 1)(b_j + 1) p(\mathcal{T}, \boldsymbol{n} + \boldsymbol{e}_i^A + \boldsymbol{e}_j^B - \boldsymbol{e}_{ij}^C), \quad (2.23)
\end{aligned}$$

where $D = n(n-1) + (a+c)\theta_A + (b+c)\theta_B + \rho c$, $\boldsymbol{n} - \boldsymbol{e}_i^A$ denotes $(\boldsymbol{a} - \boldsymbol{e}_i, \boldsymbol{b}, \boldsymbol{c})$, and so on. The notation $i \to k$ denotes that when the most recent mutation is removed from type i, the resulting type is k. Then $\mathcal{T}'_{i-}$ denotes the corresponding pair of gene trees with this mutation removed. In this recursion, two possible types of mutation have been condensed into a single summation: after removing the mutation, the resulting type is either present in the sample already or it is not. Usually these are treated separately, but for brevity they are combined into a single term here. For

example, in the terms multiplying θ_A, k may or may not be a new type, with a_k or c_{kj} possibly 0. Notationally, if we allow $\boldsymbol{n}$ to contain empty rows as placeholders for these inferred types, then the combination of the two types of mutation into a single summation is written consistently. Then $\mathcal{T}'_{i-} = \left((\mathcal{T}_A)'_{i-}, \mathcal{T}_B\right)$ represents the removal of the appropriate mutation in the appropriate marginal tree, and the corresponding entry in I_A can be deleted. If the MRCA is assumed to be the ancestral type at each site then boundary conditions for (2.23) are $p(\mathcal{T}, (\mathbf{0}, \mathbf{0}, \boldsymbol{e}_{ij})) = p(\mathcal{T}, (\boldsymbol{e}_i, \mathbf{0}, \mathbf{0})) = p(\mathcal{T}, (\mathbf{0}, \boldsymbol{e}_j, \mathbf{0})) = 1$, for $i \in I_A$, $j \in I_B$.

Equation (2.23) will prove important, and so I offer several ways to derive it. First, it is a generalization of Golding's recursion [29, 30], which also allows samples to have their alleles unrestricted at one locus but which deals with infinite-alleles mutation. To obtain (2.23), use (2.22) to substitute for $p(\mathcal{T}, \boldsymbol{n})$ in Golding's recursion (e.g. [30], their equation (2.2)). The mutation terms also require modification to account for an infinite-sites model. The result of a mutation should not be that the lineage becomes non-ancestral, but simply that the segregating site to have appeared most recently is removed, with the forward co-efficient modified accordingly.

Second, (2.23) is a special case of the recursion considered by Griffiths & Marjoram (1996) ([40], their equation (1)). They were interested in a uniform model of recombination, $Z \sim \mathrm{U}[0,1]$, which resulted in a (much more difficult to work with) integro-recursion. By replacing the breakpoint distribution with $Z = \delta(\frac{1}{2})$, the Dirac delta function at $\frac{1}{2}$, much of their terminology simplifies immediately. For example, their term for a coalescence of two different types is $2n \sum (n_k + 1 - \delta_{ik} - \delta_{jk}) Q(\mathbf{A}, \mathbf{M}, \mathbf{n}^k_{ij})$, where the summation is over all distinct coalescable pairs of type i and j and resulting in type k. With a fixed breakpoint, each such pair is one of the following pairs of multiplicities: a_i and c_{ij}, in which case $\delta_{ik} + \delta_{jk} = 1$, $n_k = c_{ij}$,

and the forward co-efficient becomes $2nc_{ij}$; b_j and c_{ij}, in which case $\delta_{ik} + \delta_{jk} = 1$, $n_k = c_{ij}$, and the forward co-efficient becomes $2nc_{ij}$; or a_i and b_j, in which case $\delta_{ik} + \delta_{jk} = 0$, $n_k = c_{ij}$, and the forward co-efficient becomes $2n(c_{ij} + 1)$. This accounts for the relevant parts of the first, second and fourth terms on the right-hand side of (2.23), and other terms can be handled in a similar fashion.

An important point is that in [40] the quantity of interest $Q(\mathbf{A}, \mathbf{M}, \mathbf{n})$ is really a joint density of the sample configuration together with the positions of the mutations, which are therefore labelled by their position. The concept of the equivalence class $\sim$ discussed in Chapter 1 no longer holds, and to recover it we should integrate over all possible mutation positions within each locus. We need to be careful not to integrate over permutations of the positions of identical segregating columns in the incidence matrix. A way to achieve this is as follows. Suppose a locus has s segregating sites with positions $\boldsymbol{s} = (s_{11}, \ldots, s_{1\kappa_1}, s_{21}, \ldots, s_{2\kappa_2}, \ldots, s_{m1}, \ldots, s_{m\kappa_m})$, where the first index groups identical columns—for s_{ij}, s_{kl}, if $i = k$ then these denote mutations on the same branch of the marginal coalescent history at this locus. So we have partitioned the collection of segregating sites by the m branches they occupy, with κ_l sites on the lth branch. To consider each possible set of mutation positions, the correct range of integration is therefore $\{\boldsymbol{s} \in [0,1]^s : 0 < s_{l1} < s_{l2} < \ldots < s_{l\kappa_l} < 1,\ l = 1, \ldots, m\}$, and hence

$$p(\mathcal{T}, \boldsymbol{n}) = \int_{\substack{0<s_{11}<\ldots<s_{1\kappa_1}<1 \\ \vdots \\ 0<s_{m1}<\ldots<s_{m\kappa_m}<1}} \cdots \int Q(\mathbf{A}, \mathbf{M}, \mathbf{n}) \, \mathrm{d}\boldsymbol{s} = \frac{1}{\kappa_1! \ldots \kappa_m!} Q(\mathbf{A}, \mathbf{M}, \mathbf{n}),$$

the latter equality holding because in the absence of recombination within the locus, the density is independent of site positions and can be taken outside the integral.

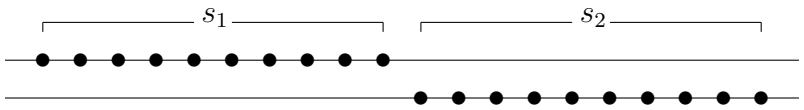


Figure 2.1: An example dataset with two sequences and $s = s_1 + s_2$ segregating sites. Each line is a sequence, and balls represent mutations differing from the root type. When sites are unlabelled, there are $\frac{s!}{s_1!s_2!}$ ways to remove mutations back to the root. When they are labelled there are $s!$ ways. If a recursion for the likelihood of this data is calculated, we therefore observe that $Q(\mathbf{A}, \mathbf{M}, \mathbf{n}) = s_1!s_2!p(\mathcal{T}, \boldsymbol{n})$.

Another way to see where the factor $\frac{1}{\kappa_1!\ldots\kappa_m!}$ comes from is to notice that in (2.23) the mutation term is a summation over all possible *sequences* with removable mutations, whereas in the recursion for $Q(\mathbf{A}, \mathbf{M}, \mathbf{n})$ the summation is over all possible *removable mutations*. In the former case, the state space is one of unlabelled mutations, in which removing one of the κ_l mutations on a branch is a single event from one state to another. In the latter case, for which mutations are distinct, removing each mutation qualifies as a distinct event. To take an illustrative example, consider the dataset shown in Figure 2.1. Here, there are $\frac{s!}{s_1!s_2!}$ paths back through the recursion for the gene tree representative of an equivalence class with arbitrarily labelled sites, whereas there are $s!$ paths back through the recursion for the joint density of a gene tree together with site positions. Each path contributes the same amount. Thus, to finish converting the recursion for $Q(\mathbf{A}, \mathbf{M}, \mathbf{n})$ into equation (2.23), either divide the answer by the combinatorial factor above or adjust the summation over removable mutations to a summation over sequences containing removable mutations. Finally, different mutation rates can be permitted at each locus.

A third way to obtain (2.23) is via a consideration of the underlying diffusion process. One can repeat the steps above for applying the generator equation (1.19) to the joint probability generating function for a sample configuration. However,

the measure-valued diffusion for the infinite-sites process is much more complicated than its finite-alleles counterpart. An alternative approach is to aim for a 'fragment' recursion for the finite-alleles model and then to derive the infinite-sites recursion directly from that, as follows.

For a recursion in terms of fragments, one can apply the generator equation (1.19) directly to

$$\widetilde{Q}(\boldsymbol{n};\boldsymbol{X}) := \binom{n}{\boldsymbol{n}} \prod_{(i,j)\in E_A\times E_B} X_{ij}^{c_{ij}} \prod_{i\in E_A} X_{i\cdot}^{a_i} \prod_{j\in E_B} X_{\cdot j}^{b_j} \tag{2.24}$$

where $X_{i\cdot} = \sum_{j\in E_B} X_{ij}$, $X_{\cdot j} = \sum_{i\in E_A} X_{ij}$ (again noting that by throwing fragments together $\widetilde{Q}(\boldsymbol{n};\boldsymbol{x})$ is not a probability), or to the corresponding generating function

$$\begin{aligned}\widetilde{G}_n(\boldsymbol{s},\boldsymbol{t},\boldsymbol{u};\boldsymbol{X}) &= \sum_{\substack{\{\hat{\boldsymbol{n}}:|\hat{\boldsymbol{n}}|=n,\\ \hat{c}_{ij}\geq 0,\hat{a}_i\geq 0,\hat{b}_j\geq 0\}}} \left[\prod_{(i,j)\in E_A\times E_B} s_{ij}^{\hat{c}_{ij}} \prod_{i\in E_A} t_i^{\hat{a}_i} \prod_{j\in E_B} u_j^{\hat{b}_j}\right] \widetilde{Q}(\hat{\boldsymbol{n}};\boldsymbol{X}) \\ &= \left(\sum_{(i,j)\in E_A\times E_B} (s_{ij}+t_i+u_j)X_{ij}\right)^n, \end{aligned} \tag{2.25}$$

which is equivalent to repeating the steps in Section 2.2.1 and then setting $s_{ij} \mapsto s_{ij} + t_i + u_j$. In either case the recombination term in the generator (2.9) is re-written as $\frac{\rho}{2}(x_{i\cdot}x_{\cdot j} - x_{ij})$. The result is a finite-alleles recursion with fragments. Finally, define the transition matrix for L-sites mutation with two alleles at each site and each site mutating independently, and let $L \to \infty$ to recover (2.23)—an outline is in Appendix B.

Fourth and finally, one can obtain (2.23) directly from a 'backwards-forwards' coalescent argument. The details of each term are very similar to those considered

in obtaining (2.19), and so are omitted.

2.3.2 A proposal distribution

To obtain a proposal distribution we shall need to approximate sampling distributions for fragments: $\hat{\pi}[(i,*)|\,\mathcal{T},\boldsymbol{n}]$ and $\hat{\pi}[(*,j)|\,\mathcal{T},\boldsymbol{n}]$. We will utilize a number of symmetry conditions, which are gathered below. To see how these are derived, consider as an example:

$$p(\mathcal{T},\boldsymbol{n})\frac{a_i}{n} = p(\mathcal{T},\boldsymbol{n}-\boldsymbol{e}_i^A)\pi[(i,*)|\,\mathcal{T},\boldsymbol{n}-\boldsymbol{e}_i^A], \tag{2.26}$$

which holds by noting that both sides are equal to $\mathbb{E}[X_i.\widetilde{Q}(\boldsymbol{n}-\boldsymbol{e}_i^A;\boldsymbol{X})]$. An intuitive interpretation is that the left-hand side is equal to the probability of obtaining the unordered sample $\boldsymbol{n}$ and then selecting a type $(i,*)$ uniformly at random from this sample to be ordered. When fragments are thrown together this occurs with probability $\frac{a_i}{n}$. This is then seen to be the same as the right-hand side—which is also the probability of the unordered sample $\boldsymbol{n}-\boldsymbol{e}_i^A$ together with one type $(i,*)$ 'ordered' on account of it being known to be sampled most recently. We will need to substitute for a ratio of probabilities, and so it is convenient to write (2.26) as

$$\frac{p(\mathcal{T},\boldsymbol{n}-\boldsymbol{e}_i^A)}{p(\mathcal{T},\boldsymbol{n})} = \frac{a_i}{n}\frac{1}{\pi[(i,*)|\,\mathcal{T},\boldsymbol{n}-\boldsymbol{e}_i^A]}.$$

More complicated ratios can be dealt with sequentially, as in the one-locus case [37]:

$$\begin{aligned}\frac{p(\mathcal{T}'_{i-}, \boldsymbol{n} - \boldsymbol{e}_i^A + \boldsymbol{e}_k^A)}{p(\mathcal{T}, \boldsymbol{n})} &= \frac{p(\mathcal{T}'_{i-}, \boldsymbol{n} - \boldsymbol{e}_i^A + \boldsymbol{e}_k^A)}{p(\mathcal{T}'_{i-}, \boldsymbol{n} - \boldsymbol{e}_i^A)} \frac{p(\mathcal{T}'_{i-}, \boldsymbol{n} - \boldsymbol{e}_i^A)}{p(\mathcal{T}, \boldsymbol{n})} \\ &= \frac{\pi[(k,*)|\, \mathcal{T}'_{i-}, \boldsymbol{n} - \boldsymbol{e}_i^A]}{(a_k+1)/n} \frac{1/n}{\pi[(i,*)|\, \mathcal{T}'_{i-}, \boldsymbol{n} - \boldsymbol{e}_i^A]} \\ &= \frac{1}{a_k+1} \frac{\pi[(k,*)|\, \mathcal{T}'_{i-}, \boldsymbol{n} - \boldsymbol{e}_i^A]}{\pi[(i,*)|\, \mathcal{T}'_{i-}, \boldsymbol{n} - \boldsymbol{e}_i^A]}. \end{aligned} \tag{2.27}$$

Note that there is some choice in the order in which this fraction is decomposed; a different order provides a different condition for π. Similar symmetry conditions can be used to obtain ratios $\frac{p(H_{k-1})}{p(H_k)}$ in terms of π for all possible previous sample configurations back in time H_{k-1}, where $H_k = (\mathcal{T}, \boldsymbol{n})$. By assuming that they also apply to the approximation $\hat{\pi}$ and the corresponding $\hat{p}$, Bayes' rule (1.12) can be used to define backwards IS proposal probabilities in terms of $\hat{\pi}$. The relevant conditions are collected in Table 2.2.

We are finally in a position to utilize the approach of De Iorio & Griffiths (2004*a*) [39] to obtain a proposal distribution, by applying the approximation detailed in (2.2)–(2.3) to the recursion (2.23). In the infinite-sites setting, it might be the case that a gene is of multiplicity 1 but that none of its mutations are removable. Then it cannot be involved in the next event back in time until another gene mutates to the same type, after which point it can be involved in a coalescence. The appropriate approximation is therefore to assume that the next event back in time occurs uniformly at random amongst all genes in the present configuration *which can be involved in the next event back in time*. In the one-locus case, this is enough to define the entire proposal distribution, since given the choice of gene the next event is determined uniquely. The resulting proposal distribution corresponds to that also

H_{k-1}	$p(H_k \mid H_{k-1})$	$\frac{\hat{p}(H_{k-1})}{\hat{p}(H_k)}$
Coalescence :		
$\boldsymbol{n} - \boldsymbol{e}_i^A$	$\frac{n(a_i + 2c_{i\cdot} - 1)}{D}$	$\frac{a_i}{n} \cdot \frac{1}{\hat{\pi}[(i,*) \mid \mathcal{T}, \boldsymbol{n} - \boldsymbol{e}_i^A]}$
$\boldsymbol{n} - \boldsymbol{e}_j^B$	$\frac{n(b_j + 2c_{\cdot j} - 1)}{D}$	$\frac{b_j}{n} \cdot \frac{1}{\hat{\pi}[(*,j) \mid \mathcal{T}, \boldsymbol{n} - \boldsymbol{e}_j^B]}$
$\boldsymbol{n} - \boldsymbol{e}_{ij}^C$	$\frac{n(c_{ij} - 1)}{D}$	$\frac{c_{ij}}{n} \cdot \frac{1}{\hat{\pi}[(i,j) \mid \mathcal{T}, \boldsymbol{n} - \boldsymbol{e}_{ij}^C]}$
$\boldsymbol{n} - \boldsymbol{e}_i^A - \boldsymbol{e}_j^B + \boldsymbol{e}_{ij}^C$	$\frac{2n(c_{ij} + 1)}{D}$	$\frac{a_i b_j \hat{\pi}[(i,j) \mid \mathcal{T}, \boldsymbol{n} - \boldsymbol{e}_i^A - \boldsymbol{e}_j^B]}{n(c_{ij}+1)\hat{\pi}[\{(i,*),(*,j)\} \mid \mathcal{T}, \boldsymbol{n} - \boldsymbol{e}_i^A - \boldsymbol{e}_j^B]}$
Mutation :		
$\boldsymbol{n} - \boldsymbol{e}_i^A + \boldsymbol{e}_k^A$	$\frac{\theta_A(a_k + 1)}{D}$	$\frac{1}{a_k+1} \cdot \frac{\hat{\pi}[(k,*) \mid \mathcal{T}'_{i-}, \boldsymbol{n} - \boldsymbol{e}_i^A]}{\hat{\pi}[(i,*) \mid \mathcal{T}'_{i-}, \boldsymbol{n} - \boldsymbol{e}_i^A]}$
$\boldsymbol{n} - \boldsymbol{e}_j^B + \boldsymbol{e}_l^B$	$\frac{\theta_B(b_l + 1)}{D}$	$\frac{1}{b_l+1} \cdot \frac{\hat{\pi}[(*,l) \mid \mathcal{T}'_{j-}, \boldsymbol{n} - \boldsymbol{e}_j^B]}{\hat{\pi}[(*,j) \mid \mathcal{T}'_{j-}, \boldsymbol{n} - \boldsymbol{e}_j^B]}$
$\boldsymbol{n} - \boldsymbol{e}_{ij}^C + \boldsymbol{e}_{kj}^C$	$\frac{\theta_A(c_{kj} + 1)}{D}$	$\frac{1}{c_{kj}+1} \cdot \frac{\hat{\pi}[(k,j) \mid \mathcal{T}'_{i-}, \boldsymbol{n} - \boldsymbol{e}_{ij}^C]}{\hat{\pi}[(i,j) \mid \mathcal{T}'_{i-}, \boldsymbol{n} - \boldsymbol{e}_{ij}^C]}$
$\boldsymbol{n} - \boldsymbol{e}_{ij}^C + \boldsymbol{e}_{il}^C$	$\frac{\theta_B(c_{il} + 1)}{D}$	$\frac{1}{c_{il}+1} \cdot \frac{\hat{\pi}[(i,l) \mid \mathcal{T}'_{j-}, \boldsymbol{n} - \boldsymbol{e}_{ij}^C]}{\hat{\pi}[(i,j) \mid \mathcal{T}'_{j-}, \boldsymbol{n} - \boldsymbol{e}_{ij}^C]}$
Recombination :		
$\boldsymbol{n} - \boldsymbol{e}_{ij}^C + \boldsymbol{e}_i^A + \boldsymbol{e}_j^B$	$\frac{\rho(a_i+1)(b_j+1)}{(n+1)D}$	$\frac{c_{ij}(n+1)\hat{\pi}[\{(i,*),(*,j)\} \mid \mathcal{T}, \boldsymbol{n} - \boldsymbol{e}_{ij}^C]}{(a_i+1)(b_j+1)\hat{\pi}[(i,j) \mid \mathcal{T}, \boldsymbol{n} - \boldsymbol{e}_{ij}^C]}$

Table 2.2: Forward transition probabilities $p(H_k \mid H_{k-1})$ and the ratio $\frac{\hat{p}(H_{k-1})}{\hat{p}(H_k)}$ for a two-locus, infinite-sites model. The constant D is defined as $D = n(n-1) + (a+c)\theta_A + (b+c)\theta_B + \rho c$, and the multiplicity of H_k is $\boldsymbol{n} = (\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c})$.

suggested by Stephens & Donnelly (2000) [37]. In the two-locus case, to obtain a proposal distribution requires much more work, since the choice of gene does not determine the next event. For example, if we chose a type $(i, *)$ with multiplicity > 1, it could coalesce with another $(i, *)$, or coalesce with a type (i, j), or coalesce with a type $(*, j)$—for *any* $j \in I_B$. However, even without trying to determine the relative probabilities of these events, one can argue that a proposal distribution uniform on genes is undesirable. To illustrate, consider a gene of type (i, j) whose multiplicity is 1 and for which none of its segregating sites can be removed; the only event in which it can be involved is a recombination event. In the proposal distribution, this event would occur with probability $\frac{1}{m^\circ}$, where m° is the number of genes that can be involved in the next event back in time, and it is straightforward to show that the corresponding SIS weight is $\frac{\rho m^\circ}{(n+1)D}$. That is, the probability of this event is independent of ρ while the weight is strongly affected by it. For $\rho \ll 1$ the importance sampler will generate too many recombination events, each contributing a very small SIS weight, and similarly for $\rho \gg 1$ too few will be generated. If we persevere through the details of the rest of the proposal distribution, analogous pathologies arise in other cases. Essentially the same problem is identified by Cardin (2006) ([53], Section 4.2.2.1) in the proposal distribution of Fearnhead & Donnelly (2001) [36].

A natural question to ask is why the above problem does not apply to mutation events. By choosing uniformly amongst genes, backwards probabilities are also independent of θ while the SIS weights are not, as in the single-locus implementation of `genetree`. But in this setting, the number of mutation (and coalescence) events is *constrained in advance*. For example, if $\theta \ll 1$ then any mutation event encountered by the importance sampler will cause the current SIS weight to suffer, but this is

offset by the fact that any genealogy will also encounter this event at some point. This is closely related to the concept of the correlation between the current and final weight of a partially reconstructed genealogy, examined in detail in Chapter 3. Since the number of recombination events is unconstrained, there is the opportunity for 'frivolity' in the behaviour of an importance sampler in terms of the number of recombination events it simulates, whereas the behaviour we *want* is evident from (2.17) (which also holds for R_n). Deviations from this property are usually most evident when $\rho \ll 1$ or $\rho \gg 1$. While we should take care to design an importance sampling scheme such that (2.17) holds, the corresponding equation for the number of mutation events is exact for *any* scheme. In light of (2.17), we should expect to have to treat recombination events separately in the proposal distribution. Note that in an infinite-sites model with subdivided population structure, migration events are unconstrained in exactly the same way. The solution I offer for recombination, below, could also be applied to the proposal distribution of De Iorio & Griffiths (2004*b*) ([63], Appendix B) for the subdivided population model.

An alternative to applying the method of De Iorio & Griffiths (2004*a*) [39] directly is as follows. Any proposal distribution can be written in the form

$$q(H_{k-1}|\, H_k) = \begin{cases} q(\mathrm{R})q(i,j|\, \mathrm{R}) & \text{for } \mathrm{E} = \mathrm{R},\ i \in I_A, j \in I_B, \\ \\ [1 - q(\mathrm{R})]\, q(i,j,\mathrm{E}|\, \mathrm{R}^{\complement}) & \text{for } \mathrm{E} \neq \mathrm{R},\ i \in I_A^*, j \in I_B^*, \end{cases}$$

where $q(\mathrm{R})$ is the probability that the next event back in time is a recombination event, $q(i,j|\, \mathrm{R})$ is the probability that a type (i,j) recombines given that the next event is a recombination, $q(i,j,\mathrm{E}|\, \mathrm{R}^{\complement})$ is the probability of some other event E occur-

ring to type (i, j) given that the next event back in time is not a recombination, and $I_A^* = I_A \cup \{*\}$, $I_B^* = I_B \cup \{*\}$ provide a shorthand for summing over both fragments and half-fragments. In this form it is clear that

$$\sum_{(i,j)\in I_A \times I_B} q(i, j|\,\mathrm{R}) = 1, \text{ and } \sum_{\substack{(i,j)\in I_A^* \times I_B^* \\ \mathrm{E}\neq\mathrm{R}}} q(i, j, \mathrm{E}|\,\mathrm{R}^{\complement}) = 1.$$

I propose to decouple recombination events in the proposal distribution by applying the method of De Iorio & Griffiths separately to each of these classes of events, after conditioning upon whether or not a recombination has occurred. All that is required by this method is to choose the value $q(\mathrm{R})$, an additional degree of freedom we have introduced. A natural choice is the relative rate of recombination events unconditional on the data: $q(\mathrm{R}) = \frac{\rho c}{D}$.

Given that a recombination occurs, selecting uniformly at random from among the genes that can recombine implies that

$$q(i, j|\,\mathrm{R}) = \frac{c_{ij}}{c}.$$

To deal with $q(i, j, \mathrm{E}|\,\mathrm{R}^{\complement})$, we apply the method of De Iorio & Griffiths to (2.23) after setting $\rho = 0$. (This will solve for backwards proposal probabilities, but forward coefficients are unchanged and still include ρ.) The technique is equivalent to writing the left-hand side of (2.23) as

$$\widetilde{D}\left(\sum_{i\in I'_A} \frac{a_i}{n^\circ} p(\mathcal{T}, \boldsymbol{n}) + \sum_{j\in I'_B} \frac{b_j}{n^\circ} p(\mathcal{T}, \boldsymbol{n}) + \sum_{(i,j)\in(I_A\times I_B)'} \frac{c_{ij}}{n^\circ} p(\mathcal{T}, \boldsymbol{n})\right),$$

where $\widetilde{D} = D - \rho c = n(n-1) + (a+c)\theta_A + (b+c)\theta_B$; n° denotes the number of genes

that can be involved in a coalescence or mutation in the next event back in time; and I'_A, I'_B, and $(I_A \times I_B)'$ are only those indices corresponding to types that can be involved in such events. Next, equate terms inside the summations on either side of the recursion. There is some ambiguity here, since some terms on the right-hand side of the recursion involve more than one type of gene. For example, the term $2n(c_{ij}+1)p(\mathcal{T}, \boldsymbol{n}-\boldsymbol{e}_i^A-\boldsymbol{e}_j^B+\boldsymbol{e}_{ij}^C)$ represents the coalescence of types $(i,*)$ and $(*,j)$. The factor of 2 in the forward co-efficient accounts for the two ways of creating this unordered pair. To deal with terms of this type, I treat it as two separate events: "Choose a type $(i,*)$ to be the gene involved in the next event back in time, which is then chosen to coalesce with a $(*,j)$", and vice versa. Each is assigned forward co-efficient $\frac{n(c_{ij}+1)}{D}$. Both events result in a change of state $\boldsymbol{n} \mapsto \boldsymbol{n}-\boldsymbol{e}_i^A-\boldsymbol{e}_j^B+\boldsymbol{e}_{ij}^C$, but each could have its own SIS weight, so splitting the events would increase the Monte Carlo variance of the likelihood estimate. This can be avoided. If each event had proposal probability q_1, q_2 respectively, with respective forward co-efficients p_1, p_2, then when either of the events are chosen a shared weight of $\frac{p_1+p_2}{q_1+q_2}$ correctly eradicates this artificially introduced variation. In the example above, q_1+q_2 is the total probability of this change of state and $p_1+p_2=\frac{2n(c_{ij}+1)}{D}$, as required. Other examples involving more than one type are treated in the same way. This device is purely for mathematical convenience, as it allows us to solve the equation systems for $\hat{\pi}$, but a penalty is that for this event two proposal probabilities need to be calculated rather than one. In a more extreme case like $\boldsymbol{n} \mapsto \boldsymbol{n}-\boldsymbol{e}_i^A$, up to $|I_A|+1$ proposal probabilities might need to be considered, though each consideration entails only a few simple arithmetic operations.

The terms on the right-hand side of our approximation equation depend on the events in which a gene can be involved. In the one-locus model, De Iorio & Griffiths

(2004a) ([39], Proposition 3) deal with three cases: a coalescence of a gene with multiplicity > 1, a mutation to a type new to the sample, and a mutation to a type already present in the sample. For brevity I have been collapsing the latter two, yet even after this simplification there are many more cases to consider in the two-locus recursion. Writing the current configuration as $H_k = (\mathcal{T}, \boldsymbol{n})$, we split up the possibilities into the following nine cases. Some are more tractable than others. In what follows, denote a type with multiplicity 1 and a removable mutation as a *singleton*.

(i) $H_{k-1} = (\mathcal{T}, \boldsymbol{n} - \boldsymbol{e}_i^A)$ or $H_{k-1} = (\mathcal{T}, \boldsymbol{n} - \boldsymbol{e}_i^A - \boldsymbol{e}_j^B + \boldsymbol{e}_{ij}^C)$. Coalescence of a non-singleton type $(i, *)$ with either $(i, *)$, (i, j) or $(*, j)$, for some $j \in I_B$. Equating terms either side of the recursion yields

$$\widetilde{D}\frac{a_i}{n^\circ}\hat{p}(\mathcal{T}, \boldsymbol{n}) = n(a_i + c_{i\cdot} - 1)\hat{p}(\mathcal{T}, \boldsymbol{n} - \boldsymbol{e}_i^A) + n\sum_{j:\, b_j \geq 1}(c_{ij} + 1)\hat{p}(\mathcal{T}, \boldsymbol{n} - \boldsymbol{e}_i^A - \boldsymbol{e}_j^B + \boldsymbol{e}_{ij}^C). \tag{2.28}$$

Apply the first and fourth conditions in Table 2.2 to (2.28) to obtain an equation for $\hat{\pi}$:

$$\widetilde{D}\frac{a_i}{n^\circ} = a_i\left(\frac{a_i + c_{i\cdot} - 1}{\hat{\pi}[(i, *)|\, \mathcal{T}, \boldsymbol{n} - \boldsymbol{e}_i^A]} + \sum_{j:\, b_j \geq 1} b_j \frac{\hat{\pi}[(i, j)|\, \mathcal{T}, \boldsymbol{n} - \boldsymbol{e}_i^A - \boldsymbol{e}_j^B]}{\hat{\pi}[\{(i, *), (*, j)\}|\, \mathcal{T}, \boldsymbol{n} - \boldsymbol{e}_i^A - \boldsymbol{e}_j^B]}\right),$$

$$\implies \hat{\pi}[(i, *)|\, \mathcal{T}, \boldsymbol{n} - \boldsymbol{e}_i^A] = \frac{n^\circ}{\widetilde{D}}\left(a_i + c_{i\cdot} - 1 + \sum_{j:\, b_j \geq 1} b_j \frac{\hat{\pi}[(i, j)|\, \mathcal{T}, \boldsymbol{n} - \boldsymbol{e}_i^A - \boldsymbol{e}_j^B]}{\hat{\pi}[(*, j)|\, \mathcal{T}, \boldsymbol{n} - \boldsymbol{e}_i^A - \boldsymbol{e}_j^B]}\right).$$

It is not clear at first sight how to deal with terms multiplying the b_j. A simple and fast solution is to interpret these ratios as a measure of the linkage disequilibrium between the two alleles i, j; it is an estimate of the probability of selecting type i at locus A, given type j at locus B, and given the sample $(\boldsymbol{a} - \boldsymbol{e}_i, \boldsymbol{b} - \boldsymbol{e}_j, \boldsymbol{c})$ (see Chapter 4

for further discussion on linkage disequilibrium). By using a sample estimate $\frac{c_{ij}}{c_{\cdot j}}$ of this term, one can obtain a solution for $\hat{\pi}[(i,*)|\,\mathcal{T}, \boldsymbol{n} - \boldsymbol{e}_i^A]$. However, this estimate suffers from the zero frequency problem: alleles not already present in the sample are assumed to have zero probability of being observed in the future. A partial solution is to assume a Dirichlet($\boldsymbol{u}$) prior across observed and inferred types $i \in I_A$ for a fixed j; we presume u_i prior observations for (i,j). One such choice is to take uniform pseudocounts $u_i = \epsilon \in (0,\infty)\ \forall i \in I_A$, where ϵ controls the strength of the prior. Letting $\epsilon \to 0$ focuses on those types that have been observed, while $\epsilon \to \infty$ results in the observed data having no effect. The posterior estimate becomes

$$\frac{\hat{\pi}[(i,j)|\,\mathcal{T}, \boldsymbol{n} - \boldsymbol{e}_i^A - \boldsymbol{e}_j^B]}{\hat{\pi}[(*,j)|\,\mathcal{T}, \boldsymbol{n} - \boldsymbol{e}_i^A - \boldsymbol{e}_j^B]} = \frac{c_{ij} + \epsilon}{c_{\cdot j} + |I_A|\epsilon} =: \kappa_A[(i,j)|\,\boldsymbol{c}], \tag{2.29}$$

and hence

$$\hat{\pi}[(i,*)|\,\mathcal{T}, \boldsymbol{n} - \boldsymbol{e}_i^A] = \frac{n^\circ}{\widetilde{D}} \left(a_i + c_{i\cdot} - 1 + \sum_{j:\, b_j \geq 1} \frac{b_j(c_{ij} + \epsilon)}{c_{\cdot j} + |I_A|\epsilon} \right). \tag{2.30}$$

Note that if in (2.29) we set $\boldsymbol{n} - \boldsymbol{e}_i^A - \boldsymbol{e}_j^B \mapsto \boldsymbol{n}$ and then sum over the index i, we obtain $\hat{\pi}[(\cdot,j)|\,\mathcal{T}, \boldsymbol{n}] = \hat{\pi}[(*,j)|\,\mathcal{T}, \boldsymbol{n}]$, as we would hope. Also observe that setting $\kappa_A[(i,j)|\,\boldsymbol{c}] = 1$ corresponds to ignoring any information we have about linkage disequilibrium, while $\kappa_A[(i,j)|\,\boldsymbol{c}] = 0$ corresponds to importance sampling on the sequentially Markov coalescent [66]. The merits of the approach in (2.29) and possible choices for ϵ are discussed in Section 2.3.3.

(ii) $H_{k-1} = (\mathcal{T}, \boldsymbol{n} - \boldsymbol{e}_i^A + \boldsymbol{e}_k^A)$. Mutation of a singleton type $(i,*)$ ($a_i = 1$, $c_{i\cdot} = 0$)

with $b = 0$ so no coalescence is possible involving this type. Set

$$\widetilde{D}\frac{1}{n^\circ}\hat{p}(\mathcal{T},\boldsymbol{n}) = \theta_{A}(a_k+1)\hat{p}(\mathcal{T}'_{i-},\boldsymbol{n}-\boldsymbol{e}_i^{A}+\boldsymbol{e}_k^{A}).$$

Using the fifth row of table Table 2.2 we have

$$\frac{\hat{\pi}[(i,*)|\,\mathcal{T}'_{i-},\boldsymbol{n}-\boldsymbol{e}_i^{A}]}{\hat{\pi}[(k,*)|\,\mathcal{T}'_{i-},\boldsymbol{n}-\boldsymbol{e}_i^{A}]} = \frac{n^\circ}{\widetilde{D}}\theta_{A},$$

whence the proposal probability for this event can be written down.

(iii) Coalescence or mutation of a singleton type $(i,*)$ ($a_i = 1$, $c_{i\cdot} = 0$) with $b > 0$. Set

$$\widetilde{D}\frac{1}{n^\circ}\hat{p}(\mathcal{T},\boldsymbol{n}) = n\sum_{j:\,b_j\geq 1}\hat{p}(\mathcal{T},\boldsymbol{n}-\boldsymbol{e}_i^{A}-\boldsymbol{e}_j^{B}+\boldsymbol{e}_{ij}^{C}) + \theta_{A}(a_k+1)\hat{p}(\mathcal{T}'_{i-},\boldsymbol{n}-\boldsymbol{e}_i^{A}+\boldsymbol{e}_k^{A}),$$

yielding, by the fourth and fifth rows of Table 2.2:

$$\frac{\widetilde{D}}{n^\circ} = \sum_{j:\,b_j\geq 1} b_j\frac{\hat{\pi}[(i,j)|\,\mathcal{T},\boldsymbol{n}-\boldsymbol{e}_i^{A}-\boldsymbol{e}_j^{B}]}{\hat{\pi}[\{(i,*),(*,j)\}|\,\mathcal{T},\boldsymbol{n}-\boldsymbol{e}_i^{A}-\boldsymbol{e}_j^{B}]} + \theta_{A}\frac{\hat{\pi}[(k,*)|\,\mathcal{T}'_{i-},\boldsymbol{n}-\boldsymbol{e}_i^{A}]}{\hat{\pi}[(i,*)|\,\mathcal{T}'_{i-},\boldsymbol{n}-\boldsymbol{e}_i^{A}]}. \quad (2.31)$$

This equation is recursive with respect to a subtree of $\mathcal{T}$, and so might recursively lead to a consideration of all subtrees of $\mathcal{T}$. An analogous situation arises in the recursion for a single-locus with subdivided population structure, and in their paper De Iorio & Griffiths (2004*b*) ([63], Appendix B) suggest an easily computable approximate solution. I resort to this approximation, defined here by taking

$$\frac{\hat{\pi}[(i,j)|\,\mathcal{T},\boldsymbol{n}-\boldsymbol{e}_i^{A}-\boldsymbol{e}_j^{B}]}{\hat{\pi}[\{(i,*),(*,j)\}|\,\mathcal{T},\boldsymbol{n}-\boldsymbol{e}_i^{A}-\boldsymbol{e}_j^{B}]} = \frac{\hat{\pi}[(k,*)|\,\mathcal{T}'_{i-},\boldsymbol{n}-\boldsymbol{e}_i^{A}]}{\hat{\pi}[(i,*)|\,\mathcal{T}'_{i-},\boldsymbol{n}-\boldsymbol{e}_i^{A}]} = \frac{\widetilde{D}}{n^\circ(\theta_{A}+b)}$$

for each $j \in I_B$. That is, backwards transition probabilities are proportional to the co-efficients multiplying each unknown $\hat{\pi}$ term.

(iv–vi) The three cases above can be treated similarly for a half-fragment ancestral only at locus B, with an analogous sample approximation

$$\frac{\hat{\pi}[(i,j)|\,\mathcal{T}, \boldsymbol{n} - \boldsymbol{e}_i^A - \boldsymbol{e}_j^B]}{\hat{\pi}[(i,*)|\,\mathcal{T}, \boldsymbol{n} - \boldsymbol{e}_i^A - \boldsymbol{e}_j^B]} = \frac{c_{ij} + \epsilon}{c_{i\cdot} + |I_B|\epsilon} =: \kappa_B[(i,j)|\,\boldsymbol{c}]. \tag{2.32}$$

(vii) Coalescence of a type (i,j) which is non-singleton at both loci. We proceed with a similar treatment to the cases above: apply the relevant approximation, and then utilize the first, second and third rows of Table 2.2 to yield, after some re-arrangement,

$$\hat{\pi}[(i,j)|\,\mathcal{T}, \boldsymbol{n} - \boldsymbol{e}_{ij}^C] = \frac{n^\circ}{\widetilde{D}} \left(c_{ij} - 1 + a_i \frac{\hat{\pi}[(i,j)|\,\mathcal{T}, \boldsymbol{n} - \boldsymbol{e}_{ij}^C]}{\hat{\pi}[(i,*)|\,\mathcal{T}, \boldsymbol{n} - \boldsymbol{e}_i^A]} + b_j \frac{\hat{\pi}[(i,j)|\,\mathcal{T}, \boldsymbol{n} - \boldsymbol{e}_{ij}^C]}{\hat{\pi}[(*,j)|\,\mathcal{T}, \boldsymbol{n} - \boldsymbol{e}_j^B]} \right). \tag{2.33}$$

Equation (2.33) also holds when either of a_i, b_j are 0. The same sample approximations $\kappa_A[(i,j)|\,\boldsymbol{c} - \boldsymbol{e}_{ij}]$ (2.29) and $\kappa_B[(i,j)|\,\boldsymbol{c} - \boldsymbol{e}_{ij}]$ (2.32) can be used to estimate terms on the right-hand side, and hence solve for $\hat{\pi}[(i,j)|\,\mathcal{T}, \boldsymbol{n} - \boldsymbol{e}_{ij}^C]$:

$$\hat{\pi}[(i,j)|\,\mathcal{T}, \boldsymbol{n} - \boldsymbol{e}_{ij}^C] = \frac{n^\circ}{\widetilde{D}} \left(c_{ij} - 1 + a_i \frac{c_{ij} - 1 + \epsilon}{c_{i\cdot} - 1 + |I_B|\epsilon} + b_j \frac{c_{ij} - 1 + \epsilon}{c_{\cdot j} - 1 + |I_A|\epsilon} \right). \tag{2.34}$$

(viii) Coalescence or mutation of a type (i,j) which has a removable mutation at precisely one locus. For brevity we deal with the case in which the type has a removable mutation at locus A; the case for locus B is similar. Equating terms in

the recursion and applying the second and seventh rows of Table 2.2 gives

$$\frac{\widetilde{D}}{n^\circ} = b_j \frac{1}{\hat{\pi}[(*,j)|\,\mathcal{T}, \boldsymbol{n} - \boldsymbol{e}_j^B]} + \theta_A \frac{\hat{\pi}[(k,j)|\,\mathcal{T}'_{i-}, \boldsymbol{n} - \boldsymbol{e}_{ij}^C]}{\hat{\pi}[(i,j)|\,\mathcal{T}'_{i-}, \boldsymbol{n} - \boldsymbol{e}_{ij}^C]},$$

from which an approximate solution (as in case (iii)) can be obtained by setting

$$\frac{\widetilde{D}}{n^\circ(b_j + \theta_A)} = \frac{1}{\hat{\pi}[(*,j)|\,\mathcal{T}, \boldsymbol{n} - \boldsymbol{e}_j^B]} = \frac{\hat{\pi}[(k,j)|\,\mathcal{T}'_{i-}, \boldsymbol{n} - \boldsymbol{e}_{ij}^C]}{\hat{\pi}[(i,j)|\,\mathcal{T}'_{i-}, \boldsymbol{n} - \boldsymbol{e}_{ij}^C]}.$$

(ix) Mutation of a type (i,j) which has a removable mutation at both loci. This is (somewhat) similar to the previous case—equating terms in the summation yields

$$\frac{\widetilde{D}}{n^\circ} = \theta_A \frac{\hat{\pi}[(k,j)|\,\mathcal{T}'_{i-}, \boldsymbol{n} - \boldsymbol{e}_{ij}^C]}{\hat{\pi}[(i,j)|\,\mathcal{T}'_{i-}, \boldsymbol{n} - \boldsymbol{e}_{ij}^C]} + \theta_B \frac{\hat{\pi}[(i,l)|\,\mathcal{T}'_{j-}, \boldsymbol{n} - \boldsymbol{e}_{ij}^C]}{\hat{\pi}[(i,j)|\,\mathcal{T}'_{j-}, \boldsymbol{n} - \boldsymbol{e}_{ij}^C]},$$

from which an approximate solution can again be obtained by setting

$$\frac{\widetilde{D}}{n^\circ(\theta_A + \theta_B)} = \frac{\hat{\pi}[(k,j)|\,\mathcal{T}'_{i-}, \boldsymbol{n} - \boldsymbol{e}_{ij}^C]}{\hat{\pi}[(i,j)|\,\mathcal{T}'_{i-}, \boldsymbol{n} - \boldsymbol{e}_{ij}^C]} = \frac{\hat{\pi}[(i,l)|\,\mathcal{T}'_{j-}, \boldsymbol{n} - \boldsymbol{e}_{ij}^C]}{\hat{\pi}[(i,j)|\,\mathcal{T}'_{j-}, \boldsymbol{n} - \boldsymbol{e}_{ij}^C]}.$$

The complete proposal distribution is given in Table 2.3. For brevity some terms in the table have been left in terms of $\hat{\pi}$—their evaluations are given by the expressions above. In practice, the weights of different events leading to the same change of state will be combined—also discussed above. In the table we have assumed $n^\circ > 0$, which need not always hold. If $n^\circ = 0$ then we set $q(\mathrm{R}) = 1$ and adjust the weights accordingly.

A number of approximating decisions were made in order to derive this proposal distribution. In the next subsection I shall discuss the implications of these approximations, and provide some justification for them. First though, there is one

Case	H_{k-1}	$p(H_{k-1} \mid H_k)$	IS weight
R	$\boldsymbol{n} - \boldsymbol{e}^C_{ij} + \boldsymbol{e}^A_i + \boldsymbol{e}^B_j$	$\frac{\rho c}{D} \cdot \frac{c_{ij}}{c}$	$\frac{(a_i+1)(b_j+1)}{(n+1)c_{ij}}$
(i)	$\boldsymbol{n} - \boldsymbol{e}^A_i$	$\frac{a_i(c_{i\cdot}+a_i-1)}{D\hat{\pi}[(i,*) \mid \mathcal{T}, \boldsymbol{n}-\boldsymbol{e}^A_i]}$	$\frac{n\hat{\pi}[(i,*) \mid \mathcal{T}, \boldsymbol{n}-\boldsymbol{e}^A_i]}{a_i}$
	$\boldsymbol{n} - \boldsymbol{e}^A_i - \boldsymbol{e}^B_j + \boldsymbol{e}^C_{ij}$	$\frac{a_i b_j \kappa_{A}[(i,j) \mid \boldsymbol{c}]}{D\hat{\pi}[(i,*) \mid \mathcal{T}, \boldsymbol{n}-\boldsymbol{e}^A_i]}$	$\frac{n(c_{ij}+1)\hat{\pi}[(i,*) \mid \mathcal{T}, \boldsymbol{n}-\boldsymbol{e}^A_i]}{a_i b_j \kappa_{A}[(i,j) \mid \boldsymbol{c}]}$
(ii)	$\boldsymbol{n} - \boldsymbol{e}^A_i + \boldsymbol{e}^A_k$	$\frac{\widetilde{D}}{D} \cdot \frac{1}{n^\circ}$	$\frac{n^\circ \theta_{A}(a_k+1)}{\widetilde{D}}$
(iii)	$\boldsymbol{n} - \boldsymbol{e}^A_i - \boldsymbol{e}^B_j + \boldsymbol{e}^C_{ij}$	$\frac{\widetilde{D}}{D} \cdot \frac{1}{n^\circ} \cdot \frac{b_j}{\theta_{A}+b}$	$\frac{n(c_{ij}+1)}{b_j} \frac{n^\circ(\theta_{A}+b)}{\widetilde{D}}$
	$\boldsymbol{n} - \boldsymbol{e}^A_i + \boldsymbol{e}^A_k$	$\frac{\widetilde{D}}{D} \cdot \frac{1}{n^\circ} \cdot \frac{\theta_{A}}{\theta_{A}+b}$	$\frac{n^\circ(a_k+1)(\theta_{A}+b)}{\widetilde{D}}$
(iv)	$\boldsymbol{n} - \boldsymbol{e}^B_j$	$\frac{b_j(c_{\cdot j}+b_j-1)}{D\hat{\pi}[(*,j) \mid \mathcal{T}, \boldsymbol{n}-\boldsymbol{e}^B_j]}$	$\frac{n\hat{\pi}[(*,j) \mid \mathcal{T}, \boldsymbol{n}-\boldsymbol{e}^B_j]}{b_j}$
	$\boldsymbol{n} - \boldsymbol{e}^A_i - \boldsymbol{e}^B_j + \boldsymbol{e}^C_{ij}$	$\frac{a_i b_j \kappa_{B}[(i,j) \mid \boldsymbol{c}]}{D\hat{\pi}[(*,j) \mid \mathcal{T}, \boldsymbol{n}-\boldsymbol{e}^B_j]}$	$\frac{n(c_{ij}+1)\hat{\pi}[(*,j) \mid \mathcal{T}, \boldsymbol{n}-\boldsymbol{e}^B_j]}{a_i b_j \kappa_{B}[(i,j) \mid \boldsymbol{c}]}$
(v)	$\boldsymbol{n} - \boldsymbol{e}^B_j + \boldsymbol{e}^B_l$	$\frac{\widetilde{D}}{D} \cdot \frac{1}{n^\circ}$	$\frac{n^\circ \theta_{B}(b_l+1)}{\widetilde{D}}$
(vi)	$\boldsymbol{n} - \boldsymbol{e}^A_i - \boldsymbol{e}^B_j + \boldsymbol{e}^C_{ij}$	$\frac{\widetilde{D}}{D} \cdot \frac{1}{n^\circ} \cdot \frac{a_i}{\theta_{B}+a}$	$\frac{n(c_{ij}+1)}{a_i} \frac{n^\circ(\theta_{B}+a)}{\widetilde{D}}$
	$\boldsymbol{n} - \boldsymbol{e}^B_j + \boldsymbol{e}^B_l$	$\frac{\widetilde{D}}{D} \cdot \frac{1}{n^\circ} \cdot \frac{\theta_{B}}{\theta_{B}+a}$	$\frac{n^\circ(b_l+1)(\theta_{B}+a)}{\widetilde{D}}$
(vii)	$\boldsymbol{n} - \boldsymbol{e}^C_{ij}$	$\frac{c_{ij}(c_{ij}-1)}{D\hat{\pi}[(i,j) \mid \mathcal{T}, \boldsymbol{n}-\boldsymbol{e}^C_{ij}]}$	$\frac{n\hat{\pi}[(i,j) \mid \mathcal{T}, \boldsymbol{n}-\boldsymbol{e}^C_{ij}]}{c_{ij}}$

Continued on next page

Continued from previous page

Case	H_{k-1}	$p(H_{k-1} \mid H_k)$	IS weight
	$\boldsymbol{n} - \boldsymbol{e}_i^A$	$\frac{a_i c_{ij} \kappa_B[(i,j) \mid \boldsymbol{c} - \boldsymbol{e}_{ij}]}{D \hat{\pi}[(i,j) \mid \mathcal{T}, \boldsymbol{n} - \boldsymbol{e}_{ij}^C]}$	$\frac{n \hat{\pi}[(i,j) \mid \mathcal{T}, \boldsymbol{n} - \boldsymbol{e}_{ij}^C]}{a_i \kappa_B[(i,j) \mid \boldsymbol{c} - \boldsymbol{e}_{ij}]}$
	$\boldsymbol{n} - \boldsymbol{e}_j^B$	$\frac{b_j c_{ij} \kappa_A[(i,j) \mid \boldsymbol{c} - \boldsymbol{e}_{ij}]}{D \hat{\pi}[(i,j) \mid \mathcal{T}, \boldsymbol{n} - \boldsymbol{e}_{ij}^C]}$	$\frac{n \hat{\pi}[(i,j) \mid \mathcal{T}, \boldsymbol{n} - \boldsymbol{e}_{ij}^C]}{b_j \kappa_A[(i,j) \mid \boldsymbol{c} - \boldsymbol{e}_{ij}]}$
(viii)	$\boldsymbol{n} - \boldsymbol{e}_j^B$	$\frac{\widetilde{D}}{D} \cdot \frac{1}{n^\circ} \cdot \frac{b_j}{b_j + \theta_A}$	$\frac{n n^\circ (b_j + \theta_A)}{b_j \widetilde{D}}$
	$\boldsymbol{n} - \boldsymbol{e}_{ij}^C + \boldsymbol{e}_{kj}^C$	$\frac{\widetilde{D}}{D} \cdot \frac{1}{n^\circ} \cdot \frac{\theta_A}{b_j + \theta_A}$	$\frac{(c_{kj} + 1) n^\circ (b_j + \theta_A)}{\widetilde{D}}$
	$\boldsymbol{n} - \boldsymbol{e}_j^A$	$\frac{\widetilde{D}}{D} \cdot \frac{1}{n^\circ} \cdot \frac{a_i}{a_i + \theta_B}$	$\frac{n n^\circ (a_i + \theta_B)}{a_i \widetilde{D}}$
	$\boldsymbol{n} - \boldsymbol{e}_{ij}^C + \boldsymbol{e}_{il}^C$	$\frac{\widetilde{D}}{D} \cdot \frac{1}{n^\circ} \cdot \frac{\theta_B}{a_i + \theta_B}$	$\frac{(c_{il} + 1) n^\circ (a_i + \theta_B)}{\widetilde{D}}$
(ix)	$\boldsymbol{n} - \boldsymbol{e}_{ij}^C + \boldsymbol{e}_{kj}^C$	$\frac{\widetilde{D}}{D} \cdot \frac{1}{n^\circ} \cdot \frac{\theta_A}{\theta_A + \theta_B}$	$\frac{n^\circ (c_{kj} + 1)(\theta_A + \theta_B)}{\widetilde{D}}$
	$\boldsymbol{n} - \boldsymbol{e}_{ij}^C + \boldsymbol{e}_{il}^C$	$\frac{\widetilde{D}}{D} \cdot \frac{1}{n^\circ} \cdot \frac{\theta_B}{\theta_A + \theta_B}$	$\frac{n^\circ (c_{il} + 1)(\theta_A + \theta_B)}{\widetilde{D}}$

Table 2.3: Proposal distribution for the two-locus infinite-sites model with recombination. The constants D, $\widetilde{D}$ are given by $D = n(n-1) + (a+c)\theta_A + (b+c)\theta_B + \rho c$, $\widetilde{D} = D - \rho c$, so that the probability in this proposal that a recombination does not occur is $1 - q(\mathrm{R}) = \frac{\widetilde{D}}{D}$. The expression for $\hat{\pi}[(i,*) \mid \mathcal{T}, \boldsymbol{n} - \boldsymbol{e}_i^A]$ is given by (2.30), and analogously for $\hat{\pi}[(*,j) \mid \mathcal{T}, \boldsymbol{n} - \boldsymbol{e}_j^B]$. $\hat{\pi}[(i,j) \mid \mathcal{T}, \boldsymbol{n} - \boldsymbol{e}_{ij}^C]$ is given by (2.34), and $\kappa_A[(i,j) \mid \boldsymbol{c}]$, $\kappa_B[(i,j) \mid \boldsymbol{c}]$ are respectively given by equations (2.29) and (2.32).

additional modification we can make to further improve its efficiency, and indeed this also applies to the proposal distribution developed in Section 2.2.3. Since mutations occurring at a locus beyond its MRCA are not segregating, they carry no information about the genealogy, and so there is no gain in tracing a lineage this far back. By ignoring such lineages, we are effectively marginalizing over them, and sampling from an even further reduced space of reduced ARGs. There exists an even coarser partition, in which two ARGs $\mathcal{H}^{(i)}$, $\mathcal{H}^{(j)}$ are *equivalent* iff the embedded histories of ancestral material up to each marginal MRCA are the same. Indeed, for this reason it is convenient to *define* a lineage to be non-ancestral when all its ancestral material is older than the corresponding MRCA. This equivalence class is illustrated in Figure 2.2, in which equivalence is denoted $\mathcal{H}^{(i)} \overset{3}{\sim} \mathcal{H}^{(j)}$. I mentioned in Section 2.2.2 that in a two-locus model there exists a recursion for $\mathbb{E}(R_n)$, the expected number of recombination events occurring in ancestral material. The same recursion can be used to calculate this quantity under the new definition of ancestral material—one simply needs to adjust its boundary conditions to exclude recombination events occurring in this new definition of non-ancestral material [67].

To throw away this newly identified material in the proposal distribution, one simply needs to 'clean-up' a locus by setting $(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}) \mapsto (\boldsymbol{0}, \boldsymbol{b} + \boldsymbol{c}_B, \boldsymbol{0})$ as soon as we reach $a + c = 1$, or $(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}) \mapsto (\boldsymbol{a} + \boldsymbol{c}_A, \boldsymbol{0}, \boldsymbol{0})$ as soon as we reach $b + c = 1$. Here $\boldsymbol{c}_A$, $\boldsymbol{c}_B$ denote the marginal data at each locus of $\boldsymbol{c}$. There are two things in this process of which we should be wary. First, for the likelihood to be calculated correctly we also need to adjust the combinatorial factor in (2.22). For example, suppose we reach a state $(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{0})$ with $a = 1$ and clean it to $(\boldsymbol{0}, \boldsymbol{b}, \boldsymbol{0})$. Then $\binom{n}{\boldsymbol{n}} \mapsto \binom{n-1}{\boldsymbol{n}-\boldsymbol{e}_i^A}$, and so we must multiply the SIS weight by $\frac{n}{a_i}$ to get the correct answer. Other cases follow similarly. Second, if we had *not* introduced this clean-up procedure then as

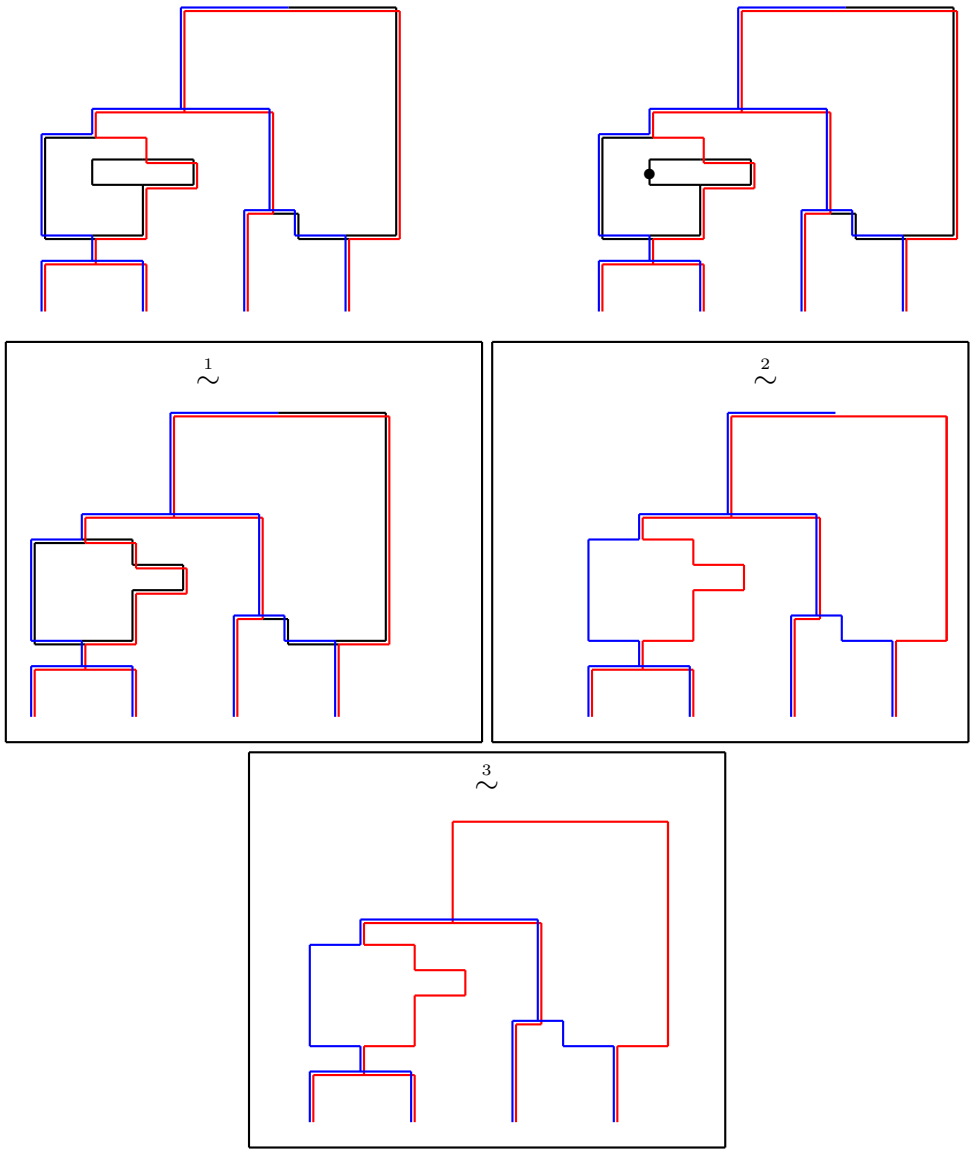


Figure 2.2: (*Top*): Two inequivalent two-locus ARGs. Lineages with ancestral material at the first locus are indicated in blue and at the second locus in red, and non-ancestral loci are shown in black. Some of this history does not affect the sample, so we consider equivalence classes of ARGs. Representatives of these equivalence classes retaining only the relevant embedded regions of the ARGs are shown below. ($\overset{1}{\sim}$): Only genes carrying ancestral material at at least one locus are traced. ($\overset{2}{\sim}$): Only loci carrying ancestral material are traced. ($\overset{3}{\sim}$): Loci are traced back only as far as their MRCA. In this equivalence class, members contain 'minimal' information about the history which determines the sample (and is therefore informative about the model): two coalescent trees together with knowledge of their overlap.

it stands our proposal distribution would be incorrect. Equation (2.23) is valid only as far back as the first MRCA; beyond that it does not take account of the fact that on the lineage ancestral to this MRCA any mutations are not observed in the data. Since we *are* using this clean-up procedure, the point is moot. Finally, note that the IS proposal distributions of Griffiths & Marjoram (1996) [40] and Fearnhead & Donnelly (2001) [36] implement analogous procedures and avoid tracing marginal genealogies farther back than their MRCA.

2.3.3 Properties of the proposal distribution

As desired, the IS proposal distribution simulates from a reduced space of ARGs, circumventing a consideration of all non-ancestral material, both at loci linked to ancestral material and on lineages carrying material older than its MRCA. This means we avoid relying on an imputation procedure. Since the proposal distribution is based on the principles of De Iorio & Griffiths (2004*a*) [39], it inherits a number of other nice properties. First, it can handle any sequences with missing data at a locus, as mentioned above. Moreover, if a sample has all its sequences with information at only one locus, say $\boldsymbol{n} = (\boldsymbol{a}, \mathbf{0}, \mathbf{0})$, then the IS proposal distribution automatically generates a genealogy only at this locus, and it is simulated from exactly the same distribution as `genetree` (and the infinite-sites proposal of Stephens & Donnelly (2000) [37]); that is, pick one of these half-fragments uniformly at random and the event is then determined. Similarly, in the case $\theta_A = \theta_B$, we could input data in the form $\boldsymbol{n} = (\mathbf{0}, \mathbf{0}, \boldsymbol{c})$, and by letting $\rho \to 0$ the proposal converges in distribution to that of `genetree` (up to a labelling of mutations by their locus). Conversely, if $\rho \to \infty$ then $q(\mathrm{R}) \to 1$ whenever $c > 0$. It is also worth noting that we would

obtain identical solutions for $\hat{\pi}$—and hence the same proposal distribution—had we worked with $\binom{a}{\boldsymbol{a}}\binom{b}{\boldsymbol{b}}\binom{c}{\boldsymbol{c}}q(\mathcal{T},\boldsymbol{n})$ instead of $p(\mathcal{T},\boldsymbol{n})$.

It would be of interest to confirm that the decoupling procedure introduced above really does offer an improvement. To achieve this I implemented a similar IS proposal distribution without decoupling recombination events. That is, *first* choose a gene uniformly at random, and then determine the relative probabilities for each event in which it can be involved. Corresponding approximations such as (2.34) become more complicated, as they now incorporate a term representing a recombination. After further work, I implemented a complete proposal distribution (details omitted), and empirical likelihood estimates were compared with those of the existing proposal. It was immediately clear that across a variety of parameter values the existing proposal outperformed one which is uniform on the choice of genes. The latter also lacks some of the properties above, such as convergence in distribution to that of `genetree` as $\rho \to 0$, as I discussed as a motivation for the decoupling method. One way to visualize the relative performance of the two proposals is by a consideration of the cumulative distribution of the set of IS weights generated by each scheme. An example is shown in Figure 2.3. As is clear from the graph, a scheme which is uniform on genes performs relatively poorly. An interpretation of its curve is that the likelihood estimate is dominated by only a handful of genealogies, with $\sim 1\%$ of genealogies contributing $\sim 95\%$ of the estimate. The decoupled IS scheme performs better—broadly comparable to that of `genetree`, even though it has to handle a much larger state space of genealogies. It converges in distribution to that of `genetree` as $\rho \to 0$, i.e. when the same random number seeds are used, then the blue curve tends to the red curve, whereas the magenta curve actually becomes even more peaked as $\rho \to 0$ (data not shown). Here, the effective sample

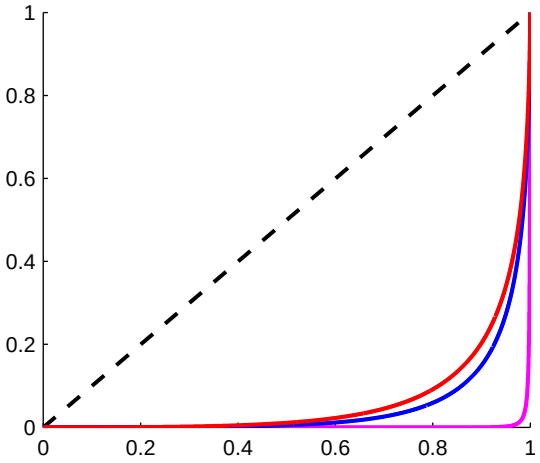


Figure 2.3: Comparison of the cumulative distribution of $N = 100,000$ normalized weights of various IS proposal distributions operating on a dataset drawn from $\texttt{ms}(10, 5, 0)$. Shown are distributions for the scheme developed in Section 2.3.2 (blue) with driving values $\theta_0 = 5$, $\rho_0 = 1$; an alternative scheme with the same driving values but which chooses uniformly among genes (magenta)—see text for details; and the scheme of `genetree` (red), which is identical to the other two schemes when $\rho = 0$. The distribution of an optimal scheme is shown as a dashed line.

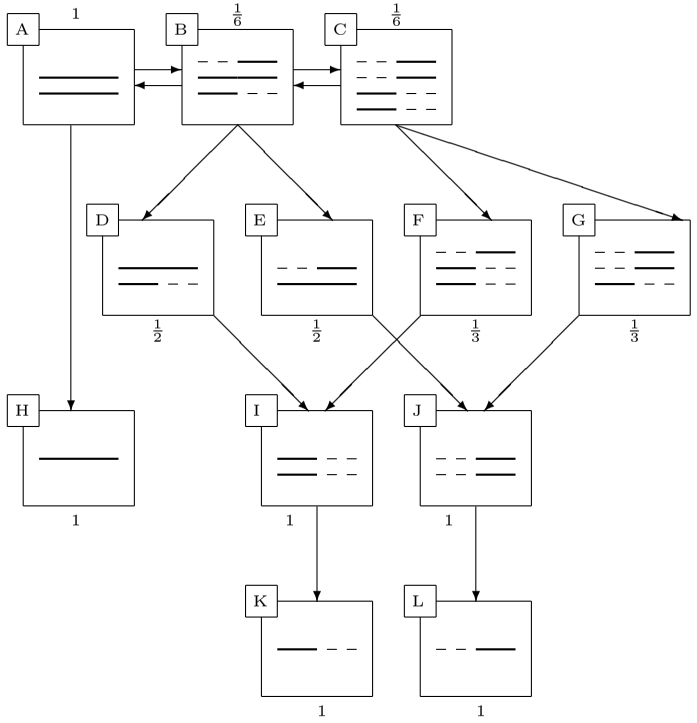
size (see Section 2.3.4.2) for my proposal is 2946 compared to 189 for the alternative. Similar graphs were observed for other parameter values.

Another welcome consequence of the decoupling procedure is this. In the case $\theta_A = \theta_B = 0$, the proposal distribution is *optimal* for all choices of $\boldsymbol{n} = (a, b, c)$ and all values of ρ. This is clear from the fact that if $\theta_A = \theta_B = 0$ then for any configuration of fragments sampled there is only one possible dataset $\mathcal{D}$; namely, that which has no segregating sites at any point which is ancestral. Hence $p(\mathcal{H} | \mathcal{D}) = p(\mathcal{H})$, whose distribution is governed by a Markov chain going back in time with known transition rates [31]. In particular, the rate of recombination is $\frac{\rho c}{D}$, coincident with our choice of $q(\mathrm{R})$; and the rate of coalescence is $\frac{n(n-1)}{D}$ and uniform among all pairs, as is the proposal distribution in this case. It is worth noting that even though it is obvious that $p(\mathcal{T}, \boldsymbol{n}) = \binom{n}{\boldsymbol{n}}$ $\forall \rho \geq 0$, different histories still contribute

different amounts to this likelihood, and it is certainly not the case that *any* proposal distribution would sample from the posterior distribution of histories optimally. Since in the optimal proposal distribution every weight is the same, a necessary condition for optimality is that during the exploration through the Markov chain of previous sample configurations, the current SIS weight is a single-valued function of the current state. This is true in for example Figure 2.4, which gives an exhaustive illustration of the optimality of our proposal distribution in the case $\boldsymbol{n} = (0, 0, 2)$, an example whose corresponding Markov chain was also analysed in detail by Simonsen & Churchill (1997) [68]. In a similar way, the IS proposal distribution can be recast in terms of a Markov chain.

An aspect of the proposal distribution that also warrants further investigation is the introduction of $\kappa_A[(i,j)|\,\boldsymbol{c}]$ and $\kappa_B[(i,j)|\,\boldsymbol{c}]$. An alternative would simply be to apply the approximation of De Iorio & Griffiths (2004*b*) ([63], Appendix B) as we have in other seemingly intractable cases—(iii), (vi), (viii) and (ix) in Table 2.3. Here, this approximation is equivalent to setting $\kappa_A[(i,j)|\,\boldsymbol{c}] = \kappa_B[(i,j)|\,\boldsymbol{c}] = 1$. To compare the relative performance of these definitions with (2.29) and (2.32), I performed the following experiment. Draw 100 datasets from $\texttt{ms}(20, 5, \rho)$ for each $\rho \in \{0, 10^{-10}, 0.1, 0.5, 1, 2, 5, 10, 20, 10^{10}\} =: \Upsilon$, and perform $N = 100,000$ runs of importance sampling with driving values equal to the true parameter values, for each of the two definitions. That is, perform the experiment for each of $\kappa_A[(i,j)|\,\boldsymbol{c}] = 1$ and $\kappa_A[(i,j)|\,\boldsymbol{c}] = \frac{c_{ij}+1}{c_{\cdot j}+|I_A|}$ (choosing $\epsilon = 1$ here), and similarly for κ_B. The relative performance of the two schemes can be measured by:

- The self-reported effective sample size (ESS, Section 2.3.4.2) of each likelihood estimate, and



Transition	$q(H_{k-1}\mid H_k)$	$p(H_k\mid H_{k-1})$	$\frac{p(H_k\mid H_{k-1})}{q(H_{k-1}\mid H_k)}$	
A → B	$\frac{\rho}{1+\rho}$	$\frac{\rho}{6(1+\rho)}$	$\frac{1}{6}$	
A → H	$\frac{1}{1+\rho}$	$\frac{1}{1+\rho}$	1	
B → A	$\frac{2}{6+\rho}$	$\frac{12}{6+\rho}$	6	
B → C	$\frac{\rho}{6+\rho}$	$\frac{\rho}{6+\rho}$	1	
B → D	$\frac{2}{6+\rho}$	$\frac{6}{6+\rho}$	3	
B → E	$\frac{2}{6+\rho}$	$\frac{6}{6+\rho}$	3	
C → B	$\frac{2}{3}$	$\frac{2}{3}$	1	
C → F	$\frac{1}{6}$	$\frac{1}{3}$	2	
C → G	$\frac{1}{6}$	$\frac{1}{3}$	2	
D → I	–	–	2	(†)
E → J	–	–	2	(†)
F → I	–	–	3	(†)
G → J	–	–	3	(†)
I → K	1	1		
J → L	1	1		

Figure 2.4: The Markov chain on states A–L corresponding to optimal importance sampling on $\boldsymbol{n} = (0, 0, 2)$ when $\theta_A = \theta_B = 0$; the initial state is A. Possible transitions are shown by arrows. Each sequence is represented by a line, with non-ancestral material dashed. Also annotated is the current SIS weight with each state, which is a single-valued function here. Note that the exit states H, K, and L each have final weight 1, indicating that the distribution of weights from this IS scheme is a point mass on the true likelihood and is therefore optimal. Corresponding forward and backward probabilities, together with the SIS weight accrued at each step, are shown in the accompanying table. (†) These entries are not really weights, but are accrued by the MRCA 'clean-up' procedure (see text). The proposal distribution is still optimal without a clean-up procedure.

- The likelihood estimate itself, whose accuracy can be measured by its relative error with respect to an independent 'true' value estimated from $N = 10,000,000$ runs (and whose definition of κ_A, κ_B is given by (2.29), (2.32) respectively).

Note that the second of these could be replaced by the accuracy of the *maximum likelihood estimate* (MLE) of the parameters, but since the likelihood surface contains all the information required to obtain the MLE, I shall focus only on the latter (see [36] for further discussion).

Results will depend on ρ. The collection Υ was chosen to investigate a wide range of recombination parameter values, with $\rho = 10^{-10}$ included to check the limiting behaviour as $\rho \to 0$, and $\rho = 10^{10}$ used as a computational approximation to $\rho = \infty$. I will refer to the set Υ throughout the thesis. Results for a selection of elements in Υ are presented in Figure 2.5.

As is clear from the figure (*top*), across a range of values for ρ importance sampling with $\epsilon = 1$ generally reports a higher effective sample size, indicating a superior likelihood estimate. For low recombination rates, this effect is diminished by the fact that the two schemes differ much less in the decisions they make. Indeed, for $\rho = 0$ the correlation is 1. (For each dataset, the two schemes used the same random number seed in order to minimize any variation not attributable to their definitions. This increases the correlation between the schemes, but any difference between them is more likely to be the signal in which we are interested.) As ρ increases, the difference between them becomes more pronounced. Using $\kappa = 1$, the ESSs of simulated datasets were clearly lower than when using $\epsilon = 1$. A summary statistic to confirm this observation is the slope of a line of linear regression, plotted in red. For all values of ρ examined, the slope was greater than 1 and increasing

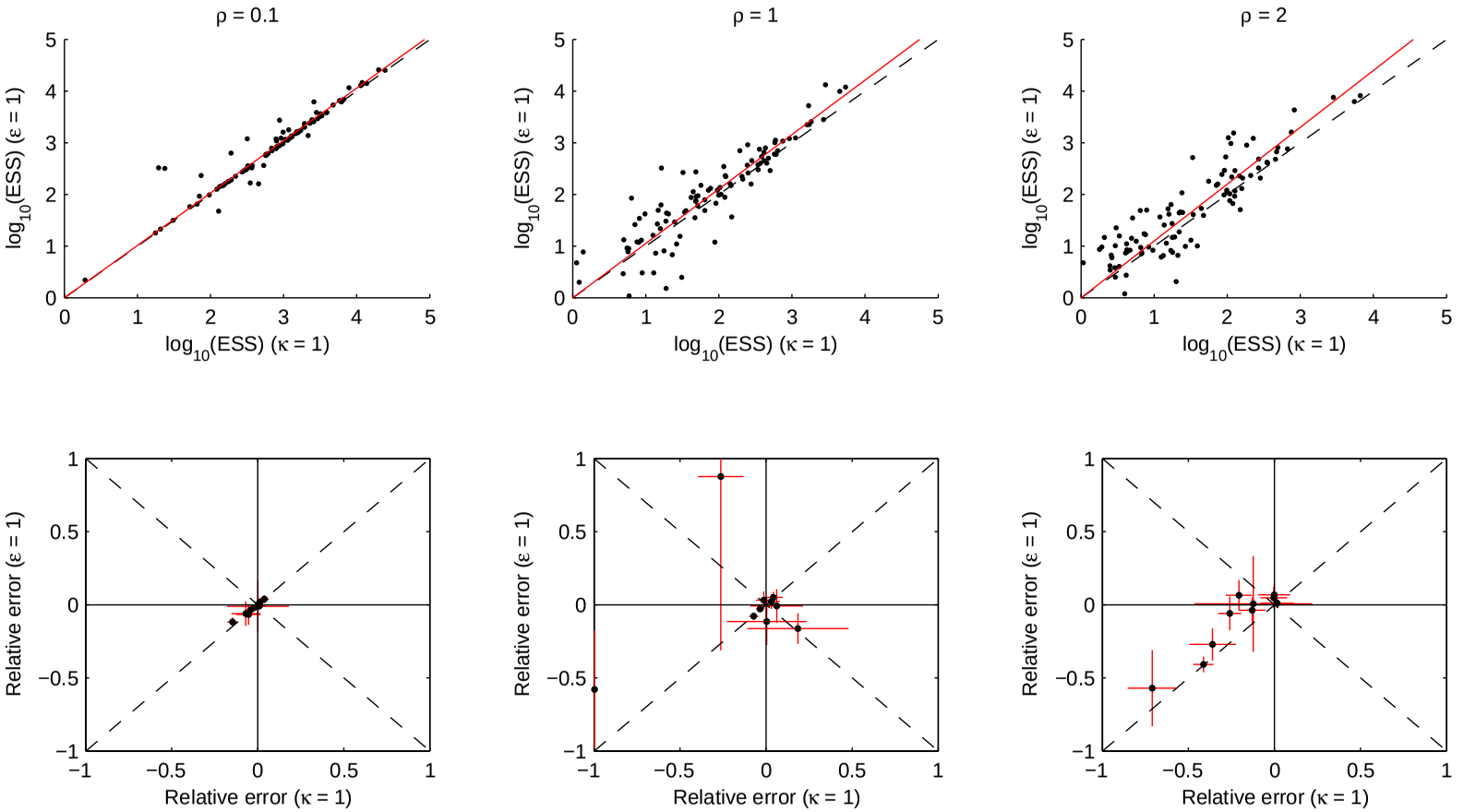


Figure 2.5: (*Top*): The relative performance of two IS schemes $\kappa = 1$ (where κ denotes both κ_A and κ_B) versus $\epsilon = 1$, as measured by the effective sample size on a $\log_{10}$ scale, for $N = 100,000$ runs. Plotted are 100 simulated datasets from each of six representative values of ρ. Shown in red are lines of linear regression, assuming zero intercept, and dashed lines indicate the diagonals (dashed). (*Bottom*): Comparisons of the relative errors of the two schemes on 10 randomly selected datasets from the plot above. Crosses show ± 1 standard error. True values were estimated by one long run of $N = 10,000,000$.

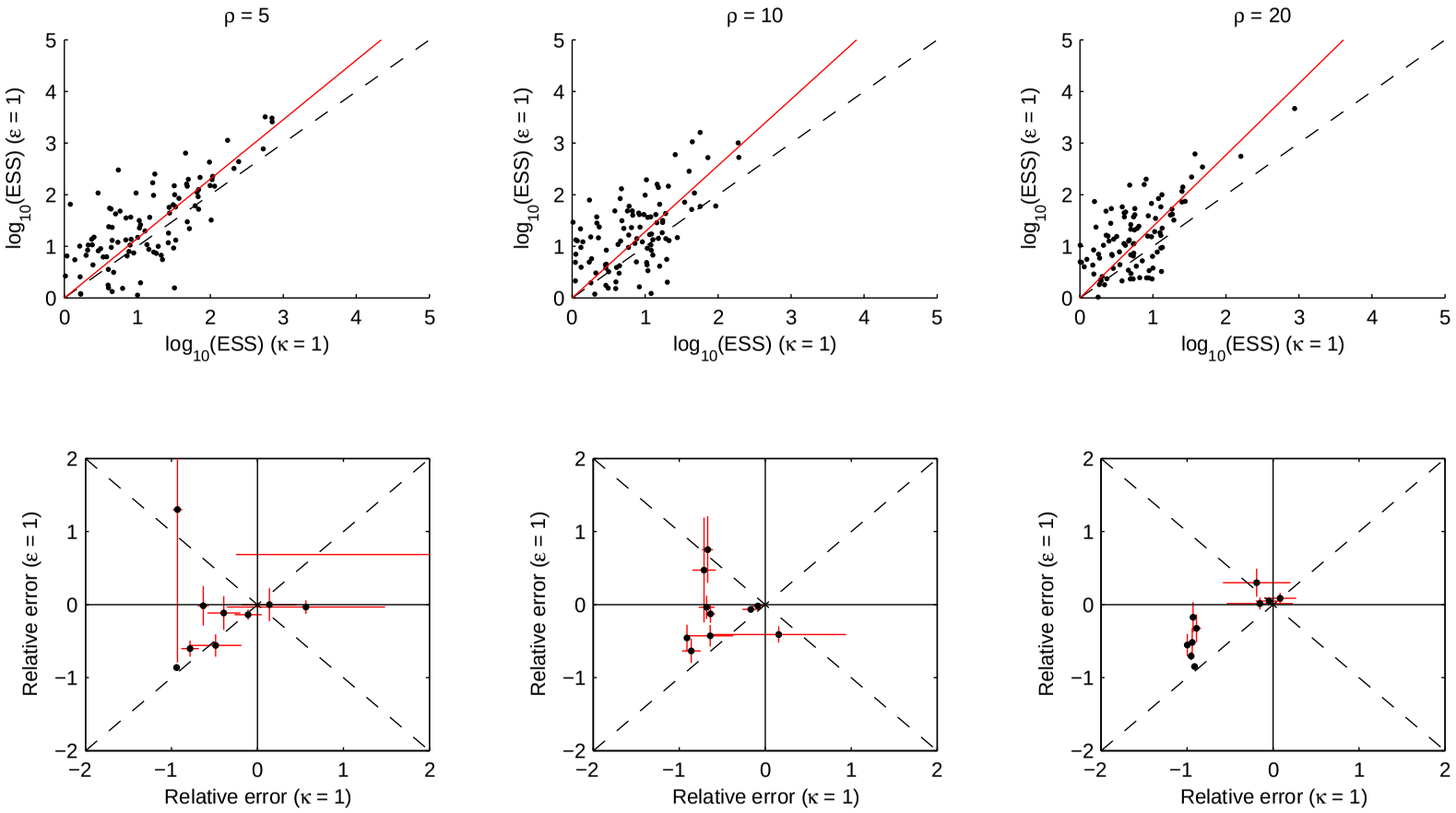


Figure 2.5: (*Continued*).

with ρ.

To verify that the ESS was not providing spurious results, the relative errors of ten randomly chosen datasets under each scheme were also plotted (*bottom*); an estimate of the true likelihood was computationally expensive, prohibiting the inclusion of all 100 datasets. Again, for small values of ρ there is little between the relative errors, but as ρ increases it becomes evident that the relative errors for $\epsilon = 1$ are generally smaller, as evinced by most of the points lying inside the horizontal cone. There also seems to be some evidence that where a scheme is inaccurate, it is more likely to be an underestimate than an overestimate (see Section 2.3.4.2). To optimize the choice of ϵ, I performed similar experiments, comparing pairs of possible values for ϵ. I found almost no effect as strong as those shown in Figure 2.5, henceforth I simply set $\epsilon = 1$.

2.3.4 A computer program

2.3.4.1 Implementation

I have implemented the proposal distribution of Section 2.3.2 in the C++ program `rita` (`r`ecombining, `i`nfinite-sites, `t`wo-locus `a`ncestries), which incorporates a number of additional features:

- It can approximate a likelihood hypersurface by providing likelihood estimates over a set of gridpoints for $(\theta_A, \theta_B, \rho)$, either performing importance sampling independently at each gridpoint or using a single driving value $(\theta_{A0}, \theta_{B0}, \rho_0)$ and then applying the method of (1.7) (replacing θ with $(\theta_A, \theta_B, \rho)$).

- It performs each of the resampling schemes discussed in Chapter 3.

- It performs ancestral inference on each of the properties discussed in Chapter 4.

Documentation for `rita` is given in Appendix C. It utilizes a random number generator previously implemented by R.C. Griffiths and based on that of Marsaglia & Zaman (1991) [69]. C++ was chosen for its speed, which is obviously an important factor for an importance sampler. I carried out extensive debugging to minimize the possibility of error. That `rita` should have the same output as `genetree` for data at a single locus, and should be optimal when $\theta_A = \theta_B = 0$, proved very useful in checking that it operates correctly. I also checked its output against exact solutions of (2.23) for some simple datasets.

It is worth noting that, by construction of the proposal distribution, selecting the next event at each step is straightforward. It can be decomposed into the following:

1. Determine whether the next event is a recombination, which occurs with probability $\frac{\rho c}{D}$. If it is, select one of the c genes that can recombine uniformly at random and calculate its SIS weight from Table 2.3 (unless $n^\circ = 0$, in which case a recombination event occurs with probability 1 and the SIS weight is modified accordingly).

2. Otherwise, select one of the n° genes that can coalesce or mutate uniformly at random.

3. Determine into which of the cases (i)–(ix) in Table 2.3 the gene falls, and select the next event using the probabilities given in this category.

4. Calculate the forward co-efficients (p_i) and backwards probabilities (q_i) associated with each way of choosing this event; the SIS weight is then $\frac{\sum_i p_i}{\sum_i q_i}$.

This process is implemented efficiently; usually it entails only a few arithmetic operations. Moreover, one does not need to consider $\hat{\pi}$ for every type. At worst, step 4 above necessitates the consideration of $O(\max\{|I_A|, |I_B|\})$ types.

2.3.4.2 Diagnostics

Given a weighted sample from an importance sampler, we should like to assess the confidence we can have in its estimate. A natural assessment for the likelihood estimate (1.5), which is simply the sample mean of the IS weights, is given by the *standard error* (s.e.)

$$s_e = \frac{\sigma_W}{\sqrt{N}},$$

where $\sigma_W = \sqrt{\text{var}(W)}$, and W is the random variable whose outcomes are the IS weights associated with histories $\mathcal{H}$ drawn from the proposal distribution q. By the central limit theorem, for large N we can use s_e to approximate the standard deviation of the likelihood estimate about the true likelihood. σ_W depends on the magnitude of the weights, and so for comparison across datasets, say, one might measure the *co-efficient of variation*

$$cv = \frac{\sigma_W}{\mu_W},$$

where $\mu_W = \mathbb{E}(W)$. A related quantity is the *effective sample size* (ESS)

$$\text{ESS} = \frac{N}{1 + cv^2} = \frac{N\mu_W^2}{\mu_W^2 + \sigma_W^2}. \tag{2.35}$$

Of course, neither σ_W nor μ_W are known (the latter is the likelihood of the data, the thing we are trying to estimate), and so we replace them by the sample estimates

$\widehat{\sigma}_W$, $\widehat{\mu}_W$, with corresponding definitions for $\widehat{s}_e$, $\widehat{cv}$, and $\widehat{\text{ESS}}$. Wherever we refer to these statistics, it is implicit that we are using these estimates based on the sample, and for convenience the hat will be suppressed.

The effective sample size is a useful quantity, since it has an interpretation in (1.8) as an approximation of the relative efficiency of the proposal distribution to the target distribution—which is $p(\mathcal{H}|\,\mathcal{D})$ here—scaled by N [35]. That is, N samples from the proposal distribution explore the posterior distribution approximately as well as ESS samples taken directly from the latter. So the larger the ESS, the more efficient the proposal distribution, with the optimal importance sampler having ESS = N. This is evident from (2.35), for which $\sigma_W^2 = 0$ under optimal sampling; each weight is the same and is equal to the likelihood. A nice property of the effective sample size is that it is independent of $h(\mathcal{H})$, and so provides a measure of efficiency for a variety of functions.

However, one needs to be extremely careful when substituting $\widehat{\sigma}_W$ for σ_W, for the following reason. In general, the distribution of weights of less efficient importance samplers will exhibit a large positive skew. In an extreme case like $q(\mathcal{H}) = p(\mathcal{H})$, this is manifest as almost all of the weights at 0 and only a handful at 1. Importance samplers are not this extreme in general, but often the skew is still evident—as in Figure 2.3, for example. A sample from such a highly skewed distribution may, with high probability (depending on the skew and the sample size), contain no realizations from the tail of the distribution, and therefore $\widehat{\sigma}_W$ is likely to *underestimate* σ_W. Thus, it is easy for our ESS to overestimate the truth, giving misleading confidence in our results. For the same reason, an insufficient sample size leads to an underestimated likelihood more often than an overestimated one, even though such a sample is unbiased. This observation is also made by Stephens & Donnelly

(2000) [37]. An inflated ESS is a consequence of genealogies with high weight being overlooked, which is also associated with an underestimate of the likelihood. The effective sample size is therefore negatively correlated with the likelihood estimate [53]. The distribution of ESS estimates is itself also skewed [43].

In the absence of more reliable measures, we shall track the ESS as a rough guide to the performance of an importance sampler, but because of the caveats above, we should interpret it with care. A way to confirm the reliability of a likelihood estimate is to re-run the importance sampler independently several times. Close agreement between plots of these independently generated likelihood surfaces provides a quick visual indication that we may have confidence in the estimate. Another strategy [36] is to track the ESS as N increases; an example is given in Figure 2.6. From the graph, it is clear that for this dataset $N = 50,000$ is insufficient to estimate the ESS, whereas $N = 1,000,000$ is much more reasonable. The discovery of genealogies with large weights causes noticeable drops in the ESS, particularly early on, and until the tail has been explored sufficiently, the ESS tends to overestimate the truth. This effect diminishes as N becomes large; asymptotically the ESS tends towards a linear increase. Figure 2.6 complements a similar plot of Stephens & Donnelly (2000) ([37], their Fig. 12), who tracked estimates of the likelihood and standard deviation of the IS weights for increasing N.

2.3.5 Comparison with existing proposal distributions

2.3.5.1 Fearnhead & Donnelly (2001)

It would be interesting to compare the proposal distribution developed in this chapter with existing IS schemes for infinite-sites models with recombination, such as

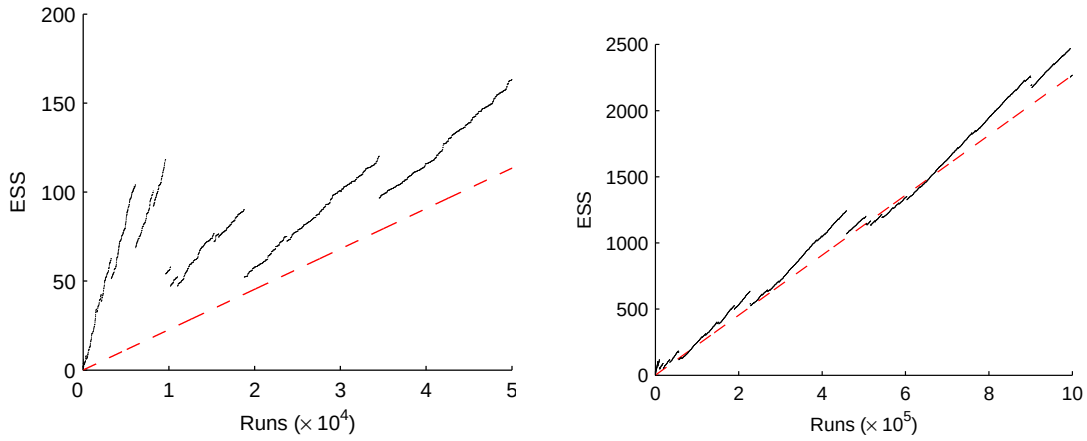


Figure 2.6: The change in ESS as the number of runs N increases, up to $N = 50,000$ (*left*) and up to $N = 1,000,000$ (*right*). These are compared with the slope of linear increase based on the ESS at $N = 1,000,000$ (dashed line). The dataset is the same as that in Figure 2.3, here using driving values $\theta_0 = 5$, $\rho_0 = 10$.

those implemented in `recom` [40] and `infs` [36]. Each of these permits a continuous breakpoint distribution, but this can be modified without too much difficulty. However, a technical problem arises with the latter scheme, suggesting that it is inapplicable unless recombination *is* allowed between all sites. An explanation is as follows.

To model the sampling distribution for new sequences, Fearnhead & Donnelly (2001) [36] extend Stephens & Donnelly's "look-down-and-copy" model, whereby new sequences are "copied" from existing ones, and this copying is imperfect, in order to incorporate mutations that could have occurred since their MRCA. The source sequence at each site in the copying process is governed by a hidden Markov model, so that jump probabilities model recombination events between sites. In the infinite-sites model, each mutation may occur only once, and in `infs` *this includes the copying process*. That is, for any mutation already observed somewhere in the rest of the sample, the emission probability for it appearing when copying from a

Figure 2.7: (*Left*): An example dataset $\boldsymbol{n} = \{\alpha, \delta_1, \delta_2\}$. A recombination occurring to α between the positions of the second and third segregating sites results in types β and γ (*right*), with non-ancestral material represented by dashed lines. See text for details.

sequence must be zero. This imposes a strong restriction on the hidden process: for a newly sampled sequence, any mutation on it (which is not on this sequence uniquely) can have been copied only from other sequences also exhibiting that mutation. A consequence is that in order to "copy" a given sequence, one might rely on jumps in the hidden process whether or not we have recombination in the model. An example is shown in Figure 2.7. In this dataset, suppose without loss of generality that the next event back in time occurred to α. Then in the proposal ([36], their Appendix B) we need to write down $\hat{\pi}[\alpha|\,\{\delta_1, \delta_2\}]$. The sequence of hidden states at each locus for this copying procedure can only be $(\delta_1, \delta_2, \delta_1)$ or $(\delta_2, \delta_2, \delta_1)$ (each with emission probability $\frac{\theta}{n-1+\theta}\frac{n-1}{n-1+\theta}$, the probability of precisely one new mutation before a coalescence), and since in both cases there is at least one jump, it is straightforward to show that $\hat{\pi}[\alpha|\,\{\delta_1, \delta_2\}] = O(\rho)$. Since each proposal probability for this sequence has a term $\hat{\pi}[\alpha|\,\{\delta_1, \delta_2\}]^{-1}$, setting $\rho = 0$ incurs a division by 0. For example, the probability of a recombination event occurring to α between the second and third segregating sites is proportional to

$$\rho\frac{\hat{\pi}[\beta|\,\{\delta_1, \delta_2\}]\hat{\pi}[\gamma|\,\{\delta_1, \delta_2, \beta\}]}{\hat{\pi}[\alpha|\,\{\delta_1, \delta_2\}]}, \tag{2.36}$$

where β and γ are the new types after the recombination event (Figure 2.7).

In fact, a strategy to rescue the proposal distribution is to note that the *numerators* in terms like (2.36) also look like $O(\rho)$, and so in the limit as $\rho \to 0$ it converges to a valid proposal distribution. Alas, it is one that still proposes recombination events with non-zero probability, and the reason for this can also be seen from Figure 2.7. Consider the event whose probability is given by (2.36). To propose this, we need to write down $\hat{\pi}[\beta|\,\{\delta_1, \delta_2\}]$ and $\hat{\pi}[\gamma|\,\{\delta_1, \delta_2, \beta\}]$. Since each of these new sequences have stretches which are non-ancestral, the crucial point is that the hidden states associated with each of these sampling events does *not* require any jumps. For example, at its ancestral locus β can be copied from (δ_1), and similarly γ can be copied from (δ_2, δ_2). The end result is that in the proposal distribution, the probability of this recombination event is $\rho\frac{O(1)}{O(\rho)} = O(1)$; it does not disappear as we let $\rho \to 0$. If, as illustrated, we suppose that the segregating sites are equally spaced along $[0, 1]$, then one can show that the probability $q(\alpha, \mathrm{R})$ of a recombination event occurring to α satisfies

$$\lim_{\rho\to 0} q(\alpha, \mathrm{R}) = \frac{1}{3} \cdot \frac{1280(2+\theta)\theta}{1280(2+\theta)\theta + (16+5\theta)^2(16+3\theta)},$$

which is ~ 0.10 when $\theta = 1$, for example, and clearly this cannot be overlooked. Further detailed discussion of the mis-specification of Fearnhead & Donnelly's proposal distribution for particular parameter values is given by Cardin (2006) [53].

2.3.5.2 Griffiths & Marjoram (1996)

We must therefore content ourselves with a comparison of the performance against `recom`. The source code for this program is designed to handle recombination at *any* position, which requires sites to be labelled by position. The computational

machinery for such a program is far more elaborate than is needed here, and so I found it simpler to re-implement their proposal distribution inside `rita`. This also enables a fairer comparison of running times. Although the time taken to determine the next event at each step was comparable for both methods, `recom` often made poor decisions—taking the genealogy no closer to the MRCA—so I found that the total running time for this scheme was frequently more than twice as high (data not shown).

To compare the two proposal distributions, I repeated the experiment shown in Figure 2.5, this time comparing $\epsilon = 1$ against the proposal distribution of `recom`. Results are shown in Figure 2.8. It is clear that my proposal outperforms that of Griffiths & Marjoram (1996) [40] across a range of recombination parameter values. For large ρ, the relative error of the latter scheme for most datasets was very close to -1, indicating that—compared to the true likelihood—all of the weights were effectively zero. Under my proposal distribution, each point in Figure 2.8 (*top*) required a running time of the order of 1 minute on a 3 GHz desktop PC with 512 MB RAM, though this was very variable for different datasets.

2.4 Discussion

In this chapter I have applied the approach of De Iorio & Griffiths (2004*a*) [39] to obtain an IS proposal distribution for a two-locus model, for both finite-alleles and infinite-sites models of mutation. The former utilizes the approximate sampling distribution $\hat{\pi}[(i,j)|\,\boldsymbol{n}]$ developed by Griffiths *et al.* (2008) [32], while the latter derives such expressions concurrently with the proposal distribution. The latter has also been implemented in a C++ program, `rita`, and I have verified that it

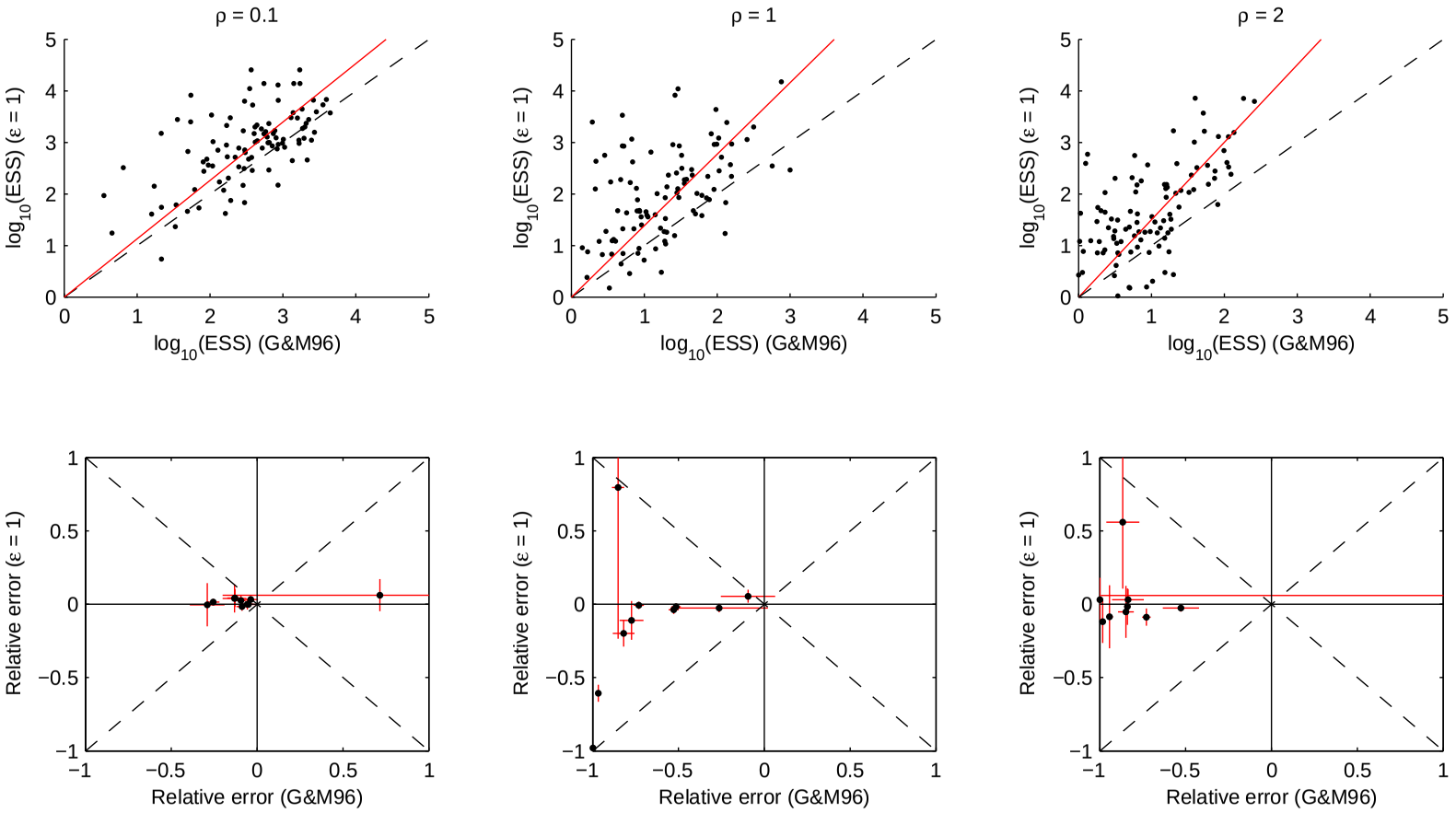


Figure 2.8: (*Top*): The relative performance of two proposal distributions: that of Griffiths & Marjoram (1996) [40] versus that of `rita` (Table 2.3), as measured by the effective sample size on a $\log_{10}$ scale, for $N = 100,000$ runs. Plotted are 100 simulated datasets from each of six representative values of ρ. Shown in red are lines of linear regression, assuming zero intercept, and dashed lines indicate the diagonals. (*Bottom*): Comparisons of the relative errors of the two schemes on 10 randomly selected datasets from the plot above. Crosses show ± 1 standard error. True values were estimated by one long run of $N = 10,000,000$.

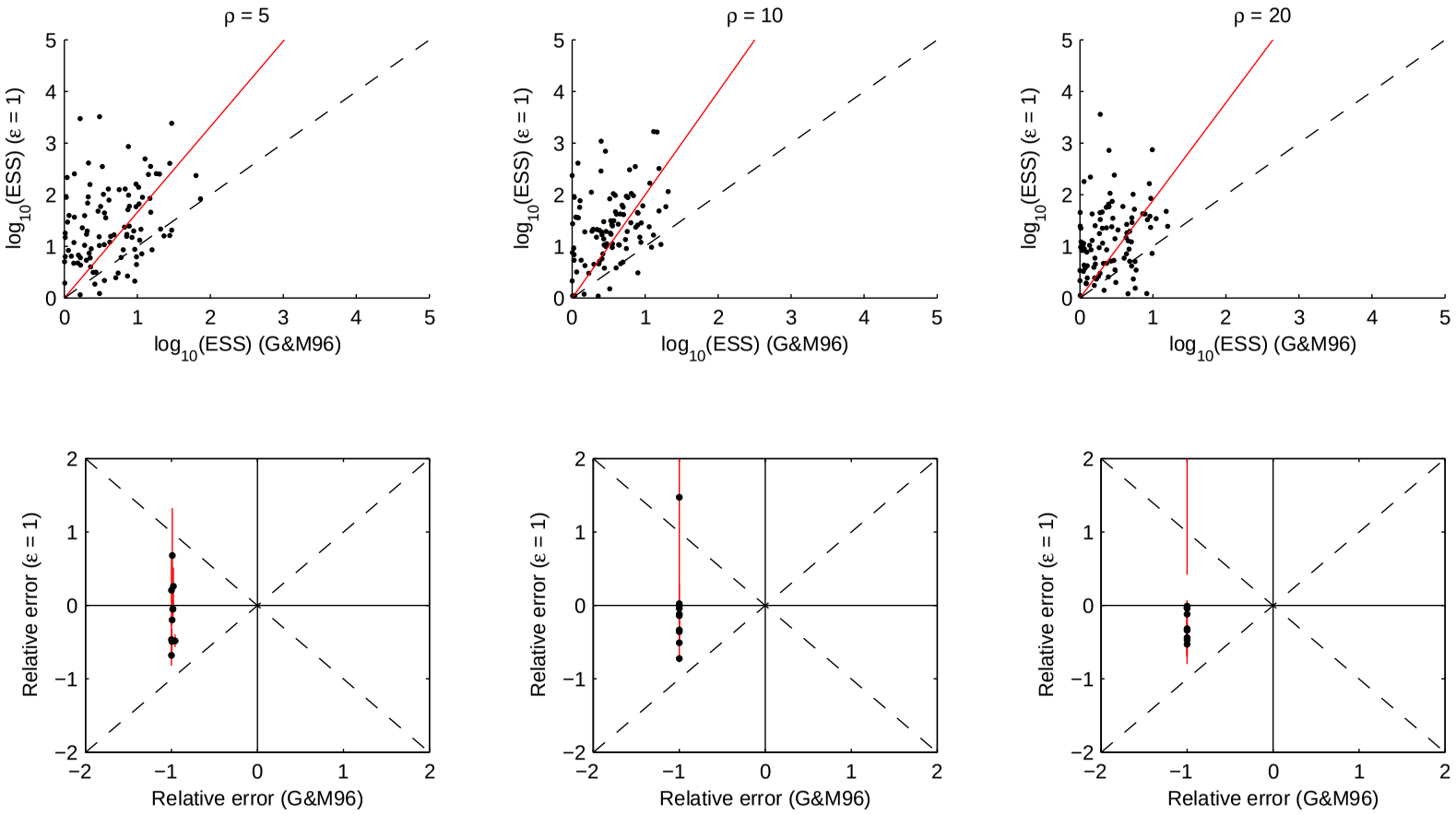


Figure 2.8: (*Continued*).

significantly outperforms the only existing proposal distribution under this model—a modification to that of Griffiths & Marjoram (1996) [40]. Our interest in this G_{81} model is as a simple way to treat infinite-sites loci for which the recombination rate between them far exceeds the rate within. Making these modelling assumptions is useful because we can then work with two neighbouring gene trees which tell us much about the correlated ancestries of the two loci—but of course, it is worth noting that importance sampling under a *finite-alleles* model is equally important. I have not pursued it here, but it will also be of interest to implement the proposal distribution in Section 2.2.3 and to compare it to other methods. For example, Y.S. Song [personal communication] has suggested an alternative more akin to the fragment state space of Section 2.3.2 and which uses imputation. The finite-alleles model is also of interest as a way to develop proposal distributions in the infinite-sites case, simply by considering a finite-sites model with L sites and then letting $L \to \infty$. The infinite-sites recursion can be obtained in this way (Appendix B), and it would be interesting to determine whether a proposal distribution might be similarly obtainable. I discuss this further in Chapter 5.

The proposal distribution implemented in `rita` has a number of other welcome properties, and it is worth summarizing them here. First, it is computationally efficient. Each step entails only a few arithmetic operations, and by working directly from a recursion whose state space is collections of *fragments* of sequences, we dispense with the countably infinite state space of alleles for newly sampled loci (any newly observed locus can have any $s \in \mathbb{N}$ new mutations). Second, and also a consequence of the fragment state space, `rita` can handle missing data, since it accepts any input of the form $\boldsymbol{n} = (\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c})$. For example, suppose data on two loci had been collected separately. Retrospectively, one can ask how the two marginal genealogies

fit together, by inputting data as $\boldsymbol{n} = (\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{0})$. Third, being able to input data in this way means that `rita` contains a one-locus importance sampler as a special case. Since it uses De Iorio & Griffiths' technique, it coincides exactly with the importance sampler of `genetree` for a panmictic population. That is, select a gene uniformly at random, and the next event is then determined. Events uninformative about the locus of interest, such as recombinations, are automatically bypassed. Inputting data at two loci and letting $\rho \to 0$ yields convergence in distribution to the same importance sampler (up to the labelling of mutations by locus). This contrasts with Fearnhead & Donnelly (2001) [36] which does not converge to a valid proposal distribution as $\rho \to 0$. Fourth, as I have noted above, in the special case $\theta_A = \theta_B = 0$, the proposal distribution is optimal for any $\boldsymbol{n}$ and for any ρ. Fifth, there is no restriction on loci being physically proximate on a chromosome; they could be at any distance, and this suggests a natural extension to Hudson's pairwise likelihood [55]. Rather than pairs of sites, consider pairs of loci. Then the only restriction is that we have some independent belief that recombination within loci is negligible, but given this restriction it could cut down the number of pairwise comparisons significantly. Finally, a *disadvantage* of this importance sampler is that by moving to a fragment state space there are many more different types of coalescence event. This results in some complicated expressions for $\hat{\pi}$ in the proposal distribution, such as (2.31), and to solve them I have resorted to a quick approximation ([63], Appendix B). Although these affect only some decisions, it is not clear whether this is the most efficient way to proceed, and a more systematic way to solve such equations would be desirable.

Applying De Iorio & Griffiths' technique directly to a two-locus, infinite-sites recursion with recombination does not yield an effective proposal distribution, primar-

ily because selecting genes uniformly at random cannot satisfy (2.16). An important property of this model is that during the reconstruction of a genealogy, the number of some classes of event (mutations and 'effective' coalescences) are restricted, while others (recombinations) are not, and so we must be careful to design a proposal distribution which strikes the right balance between the two. I have offered a solution to this problem, by *decoupling* recombination events in the proposal distribution and applying De Iorio & Griffith's technique separately to each class. Such a solution is very general and could be extended to other models of mutation. Note that there is still room for further improvement, since it is clear that choosing sequences uniformly at random to recombine conditional upon a recombination occurring is not an optimal strategy. The recombination of some sequences will resolve incompatibilities in the data, while others will not. This problem also applies to other proposal distributions, such as that of `infs` [36], and there it is confounded by having to choose a suitable breakpoint location as well as the best sequence. One should like to 'aim' the recombination events at the more appropriate sequences. An *ad hoc* solution would be to measure the effect of each recombination event on a simple non-parametric statistic like $R_{\min}(\mathcal{D})$ (also suggested in [53]), and to incorporate this into the proposal somehow. It would be interesting to find a more systematic way of achieving this, akin to the more abstract technique of De Iorio & Griffiths (2004*a*) [39].

A notable omission from this chapter is to obtain a proposal distribution in the fragment state space directly from De Iorio & Griffiths' generator approximation (2.7), by applying it to the appropriate function $\widetilde{Q}(\boldsymbol{n}; \boldsymbol{X})$ (2.24) (or to its generating function $\widetilde{G}_n(\boldsymbol{s}, \boldsymbol{t}, \boldsymbol{u}; \boldsymbol{X})$ (2.25)). Unfortunately, this results in some complicated expressions for $\hat{\pi}$ which I found to be intractable. Moreover, the technique intro-

duces terms on the right-hand side of the recursion for $\hat{p}(\boldsymbol{n})$ which do not seem to correspond to genuine genealogical events, such as $\hat{p}(\boldsymbol{n} - 2\boldsymbol{e}_{i\cdot} + \boldsymbol{e}_{ij})$. It is possible that approximations other than (2.3) and (2.7) would be more appropriate here, and progress on this matter would be welcome.

Chapter 3

Improving the efficiency of sequential importance samplers

3.1 Introduction

3.1.1 Sequential importance resampling

In recent years, the development of *sequential Monte Carlo* (SMC) methods for tackling a variety of on-line inference problems have received much attention [70]. Briefly, the model comprises a sequence of hidden states $\{x_t \in \mathscr{X} : t \in \mathbb{N}\}$ on some space $\mathscr{X}$, evolving in a Markovian way, with known transition probabilities. Observations $\{y_t \in \mathscr{Y} : t \in \mathbb{N}_+\}$ are made, and assumed to be conditionally independent given $\{x_t \in \mathscr{X} : t \in \mathbb{N}\}$, with known—possibly non-linear— marginal distribution $p(y_t|\, x_t)$. Typically, SMC involves simulating a sequence of 'particles' under a prescribed transition model $q(x_t|\, x_{t-1})$, to provide an empirical estimate of the posterior distribution $p(\boldsymbol{x}_{0:t}|\, \boldsymbol{y}_{1:t})$ of the evolution of the model conditional on the observed

data, where $\boldsymbol{z}_{t_1:t_2}$ denotes $(z_{t_1}, \ldots, z_{t_2})$. Analogous to the framework described for coalescent histories, at time t each particle i is weighted in the posterior by

$$w_t^{(i)} \propto w_{t-1}^{(i)} \frac{p(x_t^{(i)} | \, x_{t-1}^{(i)}, \boldsymbol{y}_{1:t})}{q(x_t^{(i)} | \, x_{t-1}^{(i)}, \boldsymbol{y}_{1:t})}.$$

The constant of proportionality is determined by the condition $\sum_i w_t^{(i)} = 1$. A crucial aspect of these methods is the resampling step: as time progresses, one importance weight tends to dominate the collection of particles, and the empirical posterior looks more and more like a point-mass. To overcome this, at some stage the collection of particles is resampled according to their weights. Then particles with relatively large weights are multiplied, and those with relatively small weights are discarded. There is plenty of scope for variation in the details of how and when to perform resampling, which I shall discuss below.

The degeneracy of weights is common in SMC schemes. For example, in the setting of sequential imputations Kong *et al.* (1994) [71] showed that the sequential importance weights w_t are a martingale in t. Accounting for given observations $\boldsymbol{y}_{1:t}$, we have

$$\mathbb{E}\left[\mathrm{var}(W_t | \, \boldsymbol{Y}_{1:t})\right] = \mathrm{var}(W_t),$$

with the right-hand side increasing in t. So the variance of weights is observed to be increasing in trend, and their degeneracy to a single non-zero weight as t increases is certain.

In our SIS framework for coalescent histories, this is not observed. Here, the

evolution of the SIS weights depends on the entire history:

$$\mathbb{E}(W_k) = \mathbb{E}_q \left(\frac{p(H_0 | H_{-1})}{q(H_{-1} | H_0)} \cdots \frac{p(H_{-k+1} | H_{-k})}{q(H_{-k} | H_{-k+1})} \right),$$

(recall (1.9)) and certainly will not exhibit Markovian behaviour; future amendments to the history of the sample depend on those parts of the history that have been addressed already. We should not therefore expect the variance of the (normalized) weights always to grow indefinitely; this is illustrated in Figure 3.1. In practice, for realistic parameters, it does seem to be common for the variance of normalized weights to drift upwards, but with discernible spikes along the way. A useful way to think of this is as follows. From start to finish, an importance sampler's run on a single genealogy has a number of 'probability hurdles' to overcome. For example, when `genetree` operates on infinite-sites data, it has to deal with a fixed number of coalescence and mutation events. With each event there is an associated importance weight, which comprises an *intrinsic* component—attributed to the probability of the event having occurred, and depending in turn on the parameters of the model—and an *extrinsic* component, which depends on when the importance sampler chooses to deal with that event. An importance sampler can vary only the extrinsic part of the weight, by dealing with the features of the gene tree in a chosen order, up to the constraints imposed by the topology of the tree. Suppose a collection of histories are generated in parallel. Following Liu & Chen (1998) [72], call each run a "stream". At some points on the run, events with a relatively large or relatively small intrinsic weight will have been dealt with by some, but not all, of the streams, increasing the variance of weights across the streams. After further steps, all streams will have dealt with this event, the unusually large or small intrinsic weight will be accounted for

in every sequential importance weight, and the variance of the normalized weights is reduced. In the infinite-sites setting, an example of a large hurdle would be a mutation event when there is a small value of θ. An example gene tree contriving this hurdle is shown in Figure 3.1 (*top*), corresponding to the gene tree given by

```
19  :  0
 1  :  1 0
```

A more realistic example is shown in Figure 3.1 (*bottom*), in which the extrinsic components of the weights are relatively more important. Nevertheless, to distinguish between what I have called the intrinsic and extrinsic components of sequential weights is still important. Resampling is based on the current collection of sequential weights, which are assumed to be in positive correlation with the final weights, but there are two opposing forces acting on this correlation which we need to distinguish. Extrinsic components increase the correlation—for example, bad decisions taken by the importance sampler early on will decrease the weight of a stream, and the weight might not recover, while intrinsic components can *decrease* the correlation—a stream suffers early on for a payoff in the future, as other streams catch up. Therefore one needs to be wary: these factors affect the variance of normalized weights, which is often used as a measure of how 'well' a collection of streams is performing, and hence when to perform resampling. I shall return to this measure below.

3.1.2 When and how to resample

Conventional SIS evaluates runs serially, but given the computing power to do this in parallel then one can exploit resampling procedures. That is, generate N independent samples from $q(H_{-1} \,|\, H_0)$, which we denote by $\left\{x_1^{(1)}, \ldots, x_1^{(N)}\right\}$, keeping track of each stream's importance weight. Then update each stream and its impor-

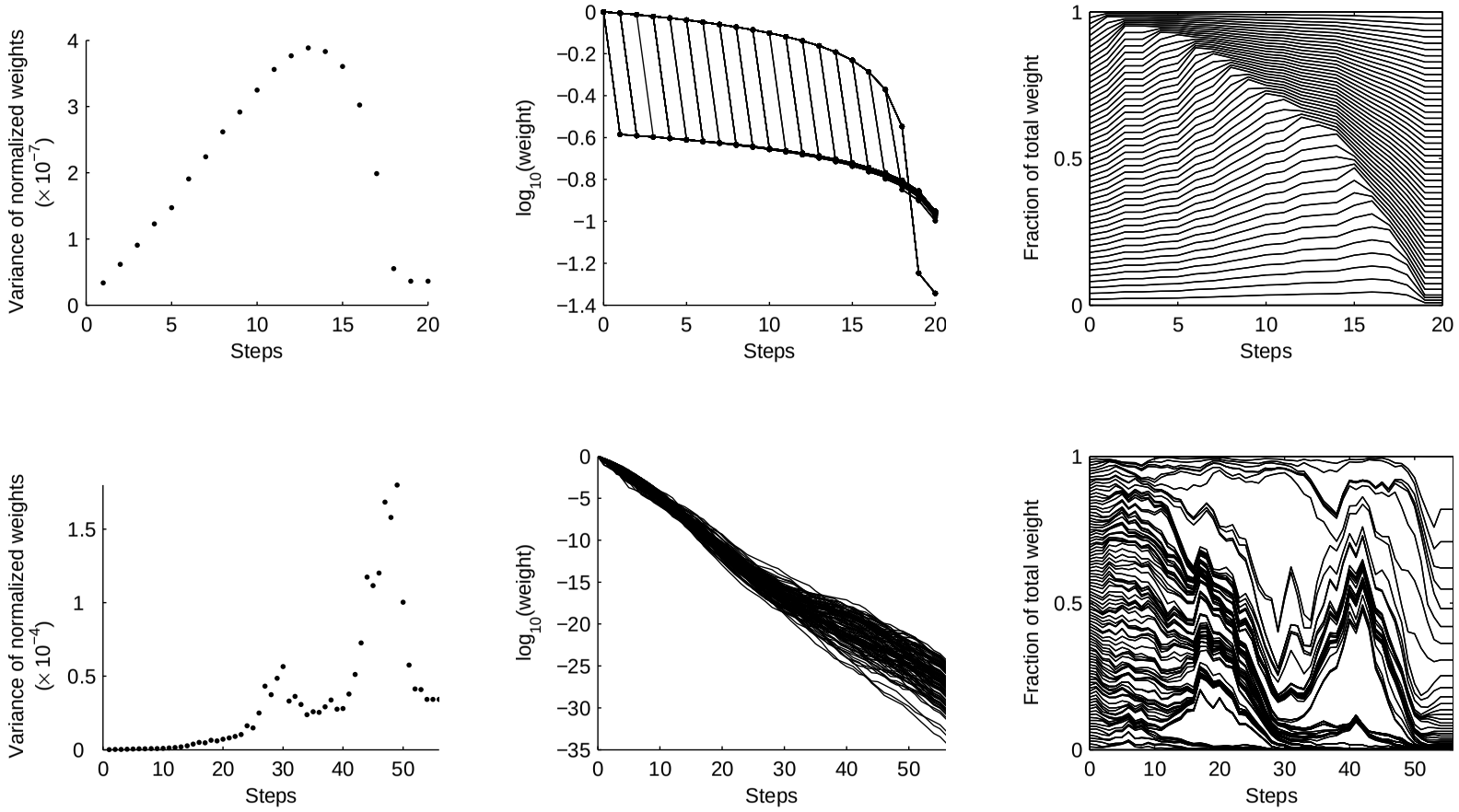


Figure 3.1: The sample variance of the normalized sequential importance weights of 1000 genealogies (*left*). Also shown are sample trajectories of the SIS weights of 100 such genealogies (*middle*), and their relative contribution to the current total (*right*), ordered by their weight at the final step. (*Top*): A contrived example with a single large hurdle (a mutation) in a gene tree (see text). Notice how each sequential weight is dominated by whether or not the mutation has yet been dealt with, and this effect is much larger than the variance in weights at the MRCA. (*Bottom*): A more realistic example, drawn from `ms`$(20, 10, 0)$. Here it is still the case that spikes in variance are relatively very large, and in this example larger than the variance at the MRCA (though this is not always observed).

tance weight in turn by drawing from $q(H_{-2}|\, H_{-1})$, and so on. By some criterion this process can be halted and the current collection of streams resampled. Two commonly used resampling schedules are *deterministic* and *dynamic*. In a deterministic schedule, resampling is carried out at checkpoints chosen in advance, such as after every λ steps. In a dynamic schedule, at each step cv^2 is checked and resampling is performed only if it exceeds some threshold. Note that it is possible to resample too often. Aside from the computational burden, which is $O(N)$, resampling can only reduce the number of distinct streams. This loss in variation is offset against an expected future gain in accuracy; in problems requiring Monte Carlo integration, it is invariably preferable to finish with a sample of genealogies from regions of relatively high posterior probability than from across the whole posterior. But immediately after resampling, the spread of this sample can only be reduced. Thus for example, resampling at the final stage of a run can only be harmful to accuracy. Similarly, resampling when the co-efficient of variation is already small is likely only to make things worse, since a small co-efficient of variation indicates that the streams are approximately all of the same utility to the estimate. It is therefore usually the case that a dynamic schedule outperforms a deterministic one, as the former is calibrated by definition only to resample when necessary. Further discussion and examples supporting this observation are given by Liu & Chen (1998) [72]. Here-onwards I consider only dynamic resampling: resampling whenever $cv^2 > B$, for some $B \in \mathbb{R} \cup \{\infty\}$. Note that $B = 0$ corresponds to resampling at every step, while $B = \infty$ corresponds to no resampling.

The simplest and oldest resampling scheme is the bootstrap filter [73]. Given a weighted sample $S_t = \left\{ \left(x_t^{(1)}, w_t^{(1)} \right), \ldots, \left(x_t^{(N)}, w_t^{(N)} \right) \right\}$, generate a new representation $\tilde{S}_t$ as follows:

- For each $\tilde{x}_t^{(i)}$, draw i from a categorical random variable on $\{1, \ldots, N\}$ with probabilities proportional to $\boldsymbol{w}_t := \left(w_t^{(1)}, \ldots, w_t^{(N)}\right)$. Thus the resulting number of copies of each stream follows a Multinomial$\left(N, \frac{\boldsymbol{w}_t}{\sum_j w_t^{(j)}}\right)$ distribution.

- Set each $\tilde{w}_t^{(i)} = \frac{\sum_j w_t^{(j)}}{N}$, the sample mean of the weights.

Streams are drawn in a multinomial fashion, in proportion to their current weight, which achieves the stated aim of randomly removing streams with low weights and duplicating streams of large weight. The cost is increased variation introduced by the multinomial sampling. Alternatives to this scheme aimed at reducing the variation have been proposed, including *stratified resampling* [74] and *residual resampling* [72]. In stratified resampling, the aim is to avoid sampling from groups of streams whose total normalized weight is less than $\frac{1}{N}$. This can be achieved as follows. Line up the normalized weights on $[0, 1]$ in order of magnitude. Divide $[0, 1]$ into N equal subintervals, and draw a uniform random variable on each: $U_j \sim \mathrm{U}[\frac{j-1}{N}, \frac{j}{N})$, retaining the streams whose weights on the unit interval contain a realization of one of these random variables.

When importance sampling on coalescent histories, there is no other requirement for streams to be sorted by weight, so stratified resampling introduces the additional burden of a sorting algorithm. Preferable might be residual resampling, whose aims are as follows: reduce the variance in multinomial sampling by selecting in advance the largest integer below the expected number of retained copies of each stream, and simply make up the difference with multinomial sampling. By adjusting the parameters of the multinomial sampling of these 'residuals' correctly, the expected number of copies of each stream is unchanged overall, while the variance in each number is reduced. This is achieved as follows:

- Pre-select $\Lambda_i := \lfloor \frac{N w_t^{(i)}}{\sum_j w_t^{(j)}} \rfloor$ copies of each $x_t^{(i)}$ to be retained. Define $N_r := N - \Lambda_1 - \ldots - \Lambda_N$ (the remaining number of slots to be filled).

- Obtain N_r i.i.d. draws from S_t with probabilities proportional to

$$\left\{ \frac{N w_t^{(i)}}{\sum_j w_t^{(j)}} - \Lambda_i : i = 1, \ldots, N \right\}.$$

- Set each $\tilde{w}_t^{(i)} = \frac{\sum_j w_t^{(j)}}{N}$, the sample mean of the weights.

As well as reducing the Monte Carlo variance, residual sampling also imposes less computational burden, and "does not seem to have disadvantages in other aspects" [72]. Here-onwards I consider only residual resampling.

Whatever resampling scheme is chosen, provided

1. the number $X_t^{(i)}$ of retained copies of stream $x_t^{(i)}$ satisfies $\mathbb{E}(X_t^{(i)}) \propto w_t^{(i)}$, and
2. the weight of the retained sample is set to $\tilde{w}_t^{(i)} = \frac{\sum_j w_t^{(j)}}{N}$,

then the resulting collection still provides a consistent estimate of the likelihood as $N \to \infty$ (or whatever integral we are approximating with a Monte Carlo estimate). Rubin (1987) [75] demonstrated the utility of the resampling step, which here I adopt for the language of importance sampling on the coalescent. For simplicity, suppose that the resampling takes place after a weighted collection of entire genealogies has been simulated. Genealogies are generated from the distribution $q(\mathcal{H})$ and resampled from the distribution proportional to $w(\mathcal{H})$, and so, asymptotically, the ultimate weighted collection is drawn from

$$q(\mathcal{H}) \frac{w(\mathcal{H})}{\int w(\mathcal{H}) q(\mathcal{H}) \, \mathrm{d}\mathcal{H}} = \frac{p(\mathcal{D}|\, \mathcal{H}) p(\mathcal{H})}{\int p(\mathcal{D}|\, \mathcal{H}) p(\mathcal{H}) \, \mathrm{d}\mathcal{H}} = \frac{p(\mathcal{H}|\, \mathcal{D})}{\int p(\mathcal{H}|\, \mathcal{D}) \, \mathrm{d}\mathcal{H}} = p(\mathcal{H}|\, \mathcal{D}),$$

each with weight $\frac{1}{N}\sum_j w^{(j)} \to \mathbb{E}_q(w(\mathcal{H})) = p(\mathcal{D})$ as $N \to \infty$. In other words, this is the optimal proposal distribution! The likelihood estimate is still asymptotically consistent, even when the resampling step is performed at some intermediate point of the sequential importance run [76]. Suppose that there are d steps in the sequential importance sampler, and resampling is performed at step $t \leq d$. Denote the partially generated genealogy at step t by $\mathcal{H}_{-t}$. As N gets large, resampling is equivalent to letting each sample survive with a probability $Cw(\mathcal{H}_{-t})$, where $w(\mathcal{H}_{-t})$ is the current weight associated with $\mathcal{H}_{-t}$, and C is a constant. Note that as $N \to \infty$ these survival probabilities become independent, and we no longer need to condition on the size of the resampled population being N. Recall that $\mathcal{X}$ denotes the random variable whose outcomes are coalescent histories $\mathcal{H}$ under the coalescent model, and denote by $\mathcal{Y}$ the random variable whose outcomes are coalescent histories drawn from q. Also denote the event "a history $\mathcal{H}$ is drawn from $\mathcal{Y}$ and survived resampling" by $\mathcal{V}$. We can equate $\mathbb{P}(\mathcal{V})N$, the approximate size of the surviving population, with

$$\mathbb{E}\left[\sum_{j=1}^{N} \mathbb{I}_{\mathcal{V}}(\mathcal{H}^{(j)})\right] = \sum_{j=1}^{N} \mathbb{P}(\mathcal{H}^{(j)} \text{ survived}) = C\sum_{j=1}^{N} w^{(j)},$$

where $\mathbb{I}_{\mathcal{V}}$ is an indicator for surviving resampling. Hence, in order to satisfy requirement 2 above, we must assign a weight $\frac{\mathbb{P}(\mathcal{V})}{C}$ to each surviving history as $N \to \infty$. The weight of a completed genealogy is thus

$$w'(\mathcal{H}) = \frac{\mathbb{P}(\mathcal{V})}{C} \cdot \frac{p(H_{-t}|\, H_{-t-1})}{q(H_{-t-1}|\, H_{-t})} \cdots \frac{p(H_{-m+1}|\, H_{-m})}{q(H_{-m}|\, H_{-m+1})} p(H_{-m}),$$

and hence

$$\begin{aligned}
\mathbb{E}_q[w'(\mathcal{Y})|\,\mathcal{V}]\mathbb{P}(\mathcal{V}) &= \mathbb{E}_q[w'(\mathcal{Y})\mathbb{I}_{\mathcal{V}}(\mathcal{Y})] \\
&= \int w'(\mathcal{H})\mathbb{P}(\mathcal{H} \text{ survived})q(\mathcal{H})\,\mathrm{d}\mathcal{H} \\
&= \int w'(\mathcal{H})Cw(\mathcal{H}_{-t})q(\mathcal{H})\,\mathrm{d}\mathcal{H} \\
&= \int w(\mathcal{H})\mathbb{P}(\mathcal{V})q(\mathcal{H})\,\mathrm{d}\mathcal{H} \\
&= \mathbb{P}(\mathcal{V})p(\mathcal{D}). \qquad (3.1)
\end{aligned}$$

A Monte Carlo estimate for $p(\mathcal{D})$ is thus the sample mean of the weights of surviving histories. The argument above resembles a rejection method for simulating partial histories from the distribution proportional to $p(\mathcal{D}|\,H_0)p(H_0|\,H_{-1})\ldots p(H_{-t+1}|\,H_{-t})$, using envelope function $\frac{1}{\mathbb{P}(\mathcal{V})}q(\mathcal{H}_{-t})$.

If one wishes to estimate the likelihood (of θ, say) away from the driving value θ_0, this can be accommodated [35] as usual by the Monte Carlo estimate

$$\hat{L}(\theta) = \frac{1}{N}\sum_{j=1}^{N} w'(\mathcal{H}^{(j)})\frac{p(\mathcal{H}^{(j)}|\,\theta)}{p(\mathcal{H}^{(j)}|\,\theta_0)}.$$

Requirement 1 above can be relaxed, so that resampling is based on any probability vector $\boldsymbol{a}$. Requirement 2 is modified accordingly, so that a stream sampled with probability $a^{(i)}$ has new weight $\frac{w_t^{(i)}}{Na^{(i)}}$. For example, $a^{(i)} \propto \sqrt{w_t^{(i)}}$ causes the resampler to behave as if the weights were more similar to each other, ensuring greater diversity in the resampled population [70]. One might even wish to incorporate *future* information into $\boldsymbol{a}$—this is discussed in Section 3.3.

3.1.3 Stopping-time resampling

It can be advantageous to allow different streams to proceed a different number of steps before resampling is performed between them. Instead, they can proceed until some *stopping-time* criterion is reached. We can formalize the definition of stopping-time resampling (SISSTR) [76] as follows. Let the sequence of stopping times $1 < T_1 < \ldots < T_L < d$ be defined on the sample path $\left\{x_1^{(i)}, x_2^{(i)}, \ldots\right\}$, so that the event $\{T_l = t\}$ is measurable with respect to the σ-field generated by $\left\{x_1^{(i)}, \ldots, x_t^{(i)}, T_1, \ldots, T_{l-1}\right\}$; after $\left\{x_1^{(i)}, \ldots, x_t^{(i)}, T_1, \ldots, T_{l-1}\right\}$ is observed, we know whether T_l has been reached, and it does not for example depend on the activity of any other streams. At each of these stopping times, resampling is performed iff $cv^2 > B$, where B is a tuning parameter specified in advance. Chen *et al.* (2005) [76] proved that, provided $\mathbb{P}(T_l \leq d) = 1$ (or $\mathbb{P}(T_l < \infty) = 1$ if $d = \infty$), (3.1) still holds when the time of resampling is drawn from the random variable T_l, and when this random variable can vary across streams. SISSTR has been applied successfully to coalescent histories of a single locus, mutating under a finite-alleles model [77, 76]. In this application, T_l is defined to be the step of the lth coalescence event. Chen *et al.* (2005) [76] claim that it has also been applied successfully to a model of infinite-sites mutation with varying population size.

In the next section, I discuss why a simple application of stopping-time resampling under the infinite-sites model is problematic. I propose a new definition for stopping times in Section 3.2.2, and in Section 3.2.3 I extend it to incorporate recombination. The effect of using these stopping times on simulated data is investigated in Section 3.2.4, and generally shown to yield an improvement both in terms of the correlation between current and final weight (Section 3.2.4.1), and in terms of the

accuracy of likelihood estimates (Section 3.2.4.2). In Section 3.2.5, I conclude with a discussion of possible extensions of these new stopping times.

Also in this chapter (Section 3.3), I adapt and modify an approach complementary to stopping-time resampling, that of *pilot-exploration resampling* [78]. This is introduced in Section 3.3.1, and its relationship with stopping-time resampling is discussed. It is adapted for the coalescent in Section 3.3.2, and in Section 3.3.3 I perform an extensive experiment on simulated data to confirm both that its use can be an improvement to likelihood estimates (Section 3.3.3.1), as expected, and also that its use can be a preferable way to assign computing resources under certain conditions (Section 3.3.3.2). Finally, in Section 3.4 I conclude the chapter by speculating on a third improvement procedure that was unsuccessful, but which might still have potential.

3.2 A new definition of stopping times

3.2.1 Motivation

The motivation behind SISSTR is that standard resampling techniques do not perform well on a certain class of problems, and might even make the sample worse. These are problems for which the current SIS weight of a stream is poorly correlated with its final weight. In 'most' SMC problems this is not the case, owing to the underlying Markov process. Streams that have thus far performed badly are still expected to do about as well as other streams in the future, and the relatively poor current weight will simply be propagated to the end. Under the coalescent model, the reverse could be true. A stream with a relatively poor current weight might

be in such a state because it is already quite close to the MRCA—and because of this observation, its final weight could actually be relatively high. To relate this idea to concepts discussed earlier, after a fixed number of steps different streams will have encountered different quantities of weight that is *intrinsic* to the dataset. One should expect a more natural timescale for coalescent models to be stopping times defined at each coalescence event. This gives a better indication as to how far a stream really is from the MRCA. Resampling at stopping times should therefore result in a fairer procedure, in which streams are compared when they have roughly the same expected future number of steps remaining. The definition of a stopping-time should aim to separate intrinsic from extrinsic weight, and to stop streams when they have accounted for the same amount of intrinsic weight. Resampling then punishes or rewards streams only on the extrinsic components of their weights, which result from the decisions taken by the importance sampler. I aim to extend the definition of stopping-time, ultimately to apply to our model of interest, which I have been denoting as G_{81}. I shall first reconsider stopping times under the one-locus, infinite-sites model, keeping in mind the motivation for using stopping times that I have just discussed.

3.2.2 A stopping-time for infinite-sites data

Let us continue with Chen *et al.*'s definition for stopping times. Consider the two partially reconstructed genealogies given by intermediate datasets shown in Figure 3.2, which represent histories reached at T_1 when importance sampling on the one-locus gene tree given by

```
3 :  0
1 : 10  9  8  7  6  5  4  3  2  1  0
```

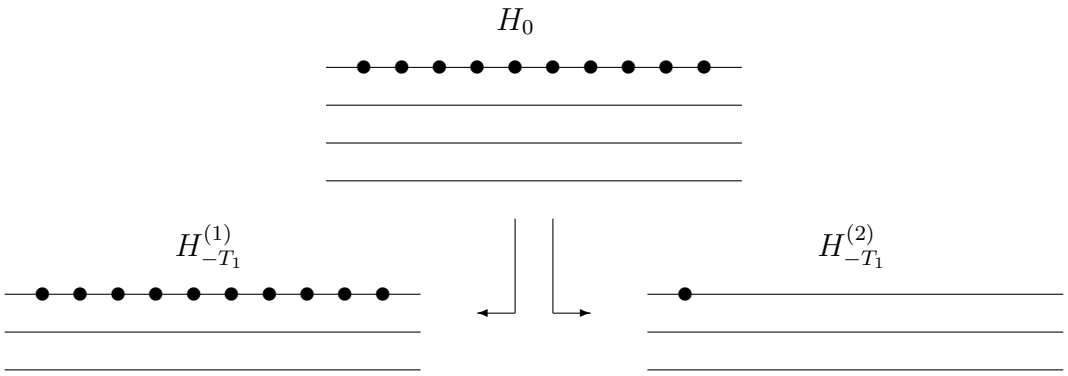


Figure 3.2: An example dataset H_0 (*top*) on which an importance sampler such as `genetree` can operate. When Chen *et al.*'s stopping-time is used, two partial genealogies $H^{(1)}_{-T_1}$, $H^{(2)}_{-T_1}$ that could be observed at the first stopping time are illustrated (*bottom*).

Although each genealogy has reached the same stopping-time, one is clearly much closer to the MRCA than the other. Resampling on their current weights is not a good strategy. For example, when $\theta = 1$ then $w^{(1)}_{-T_1} = w^{(2)}_{-T_1}$, and so we expect to resample them in the same quantity. Yet $p\left(H^{(1)}_{-T_1}\right)/p\left(H^{(2)}_{-T_1}\right) \approx 10^{-6}$; conditional on observing these two partially reconstructed genealogies, the second makes one million times the contribution to the likelihood compared to the first. This example is somewhat contrived, but the point is clear. Under the infinite-sites model, mutation information should be incorporated into the definition of a stopping-time just as much as coalescent information. Indeed, contrary to the claim of Chen *et al.* (2005) [76], my results on simulated data suggest that using only the number of coalescence events often diminishes the quality of the sample, compared to doing no resampling at all (see Section 3.2.4). That a standard implementation of stopping-time resampling might be inappropriate for infinite-sites data is noted independently by Hobolth *et al.* (2008) [44].

The advantage of utilizing resampling on infinite-sites data is that the total number of mutation events is known in advance, and, unlike a finite-alleles model, every mutation event is a move towards the MRCA. This provides us with two natural timescales for the reconstruction of a genealogy by the importance sampler—the number of coalescence events and the number of mutation events. By this measure, we can project each intermediate dataset H_{-t} onto $(C, M) \in \{0, \ldots, n-1\} \times \{0, \ldots, s\}$, where C denotes the number of coalescence events required to reach H_{-t}, M denotes the number of mutation events, and s is the number of segregating sites of the full dataset H_0. Since n and s are known in advance, it is straightforward to define a metric based on these statistics between any two intermediate datasets; or between a dataset and the MRCA, for which $(C, M) = (n-1, s)$. Stopping times can then be expressed in terms of the distance from the starting point (or equivalently from the end point—the MRCA) with respect to some metric, and there is a one-one correspondence between stopping-time definitions and metrics. For example, the stopping-time of Chen *et al.* (2005) [76]—which hereafter I refer to as CXL—is associated with the metric $d_{CXL}[H_{-t_1}, H_{-t_2}] = d_{CXL}[(C_1, M_1), (C_2, M_2)] = |C_2 - C_1|$ (the first of these definitions is in fact a pseudometric). An intermediate dataset H_{-t} projected onto (C, M) is then determined to have reached each stopping time T_l by the condition $d_{CXL}[H_{-t}, H_0] = d_{CXL}[(C, M), (0, 0)] = C \geq l$, for $l = 1, 2, \ldots$

Two possible extensions of d_{CXL} to incorporate mutations are:

- $d_{OR}[(C_1, M_1), (C_2, M_2)] := |C_2 - C_1| + |M_2 - M_1|$, and
- $d_{AND}[(C_1, M_1), (C_2, M_2)] := \min\{|C_2 - C_1|, |M_2 - M_1|\}$.

In the first case, a partially reconstructed genealogy reaches the lth stopping-time when the sum of its coalescence and mutation events reaches l (hence "OR". A

coalescence *or* mutation event brings it closer to the next stopping-time. d_{OR} is the Manhattan metric.) In the second case, a partially reconstructed genealogy reaches the lth stopping-time when the number of its coalescence events *and* the number of its mutation events reaches l. Other metrics are of course possible, but for now I shall consider those suggested here. While they attempt to incorporate mutation information, each has its own disadvantages. OR treats coalescences and mutations equally, so if one event tends to have much smaller weight than the other (much smaller intrinsic weight), genealogies which happen to have incorporated more of this type of event at a given stopping-time will be punished unduly. AND explicitly avoids this problem, but with the cost of forcing coalescence and mutation events to proceed at the same rate. Assuming the weight of each event is less than 1, it encourages the reconstruction of genealogies with precisely 1 mutation event during each epoch. If $s \ll n$ then mutations might be squashed towards the tips of the coalescent tree. If $s \gg n$ then mutations might be squashed towards the root.

Each of these problems can be addressed by proposing modified versions of the above metrics which I shall call scaled-OR (sOR) and scaled-AND (sAND). Enlarge the projection space $\{0,\ldots,n-1\}\times\{0,\ldots,s\}$ to $[0,\nu(n-1)]\times[0,\nu\mu s]$, for $\nu \in (0,\infty)$, $\mu \in (0,\infty)$—parameters we specify in advance. Metrics can be defined on this space by:

- $d_{sOR}[(C_1,M_1),(C_2,M_2)] := \nu\left(|C_2-C_1| + \mu|M_2-M_1|\right)$, and
- $d_{sAND}[(C_1,M_1),(C_2,M_2)] := \nu \min\left\{|C_2-C_1|, \mu|M_2-M_1|\right\}$.

In each case we assume $s > 0$. If $s = 0$, take $d_{sOR} = d_{sAND} = d_{CXL}$. Consider first μ. The basic problem with the unscaled metrics is that mutation events will tend to be 'worth' a different amount in weight compared to coalescence events.

μ represents an exchange rate between the two, and a natural choice would be $\mu = \frac{n-1}{s}$. Then, given n sequences and s segregating sites in a dataset, $\frac{n-1}{s}$ mutation events are equivalent to one coalescence event in bringing a partially reconstructed genealogy one step closer to the next stopping-time. As a modification to AND, this encourages coalescence and mutation events to proceed at a more appropriate rate. As a modification to OR, it identifies coalescence and mutation events with their real 'worth' by assuming that $n-1$ coalescence events is equivalent progress to s mutation events. But recall that this space is merely a proxy for the true distance, that of weight. The value of a mutation event to a stopping-time should depend only on its weight for a given θ, rather than on how many mutations actually have been observed. I propose to incorporate this by rescaling the number of mutation events using $\mathbb{E}(S_n)$ rather than s; any discrepancy between the two is then accounted for. This is a subtle point, and deserves elaboration. Suppose for example that θ is small, but that n and s are both relatively large. The current weight of any genealogy will be dominated by the number of mutation events it has encountered, and we would like stopping times to be similarly dominated in order to reflect that: $d[(C, M), (0, 0)] \propto M$. Using $\mu = \frac{n-1}{s}$ would not rescale the metric much, apportioning coalescences and mutations a similar distance. On the other hand, since $\mathbb{E}(S_n)$ is small, $\mu = \frac{n-1}{\mathbb{E}(S_n)}$ would correctly amplify the distance accorded to each mutation event.

Consider next the role of ν. When a metric is defined on a continuous space it introduces additional flexibility in the number of stopping times we can use. For example, suppose we wanted 20 stopping times. A natural choice would be to place them at each ventile between 0 and the maximum distance from 0 as determined by the metric, which corresponds to the distance of a genealogy at the MRCA from 0.

The distance is proportional to ν. So instead of choosing the value of each stopping-time, we can equivalently fix stopping times to increment in units of 1 and vary ν to choose the total size of the space and hence the total number used. This is simply for mathematical convenience—now, stopping-times increment in size 1 on both the discrete and continuous metric spaces. In the latter case, varying ν allows us to choose the grain of the stopping times. There is an upper limit to ν—corresponding to a finest grain that is still practical—beyond which every move of the importance sampler crosses more than one stopping-time. Henceforth, I will set ν so that sOR and sAND encounter $n-1$ stopping times, the same number as CXL.

The five metrics proposed in this section are illustrated in Figure 3.3. Note that of the new metrics, only sOR enjoys the properties that, for the lth stopping-time T_l:

- $T_l \to \inf\{t \in \mathbb{N} : M \geq l\}$ as $\theta \to 0$, and
- $T_l \to \inf\{t \in \mathbb{N} : C \geq l\}$ as $\theta \to \infty$.

Because of this and the unnatural way that the AND metrics force rates of coalescence and mutation, it is expected that sOR will outperform the others. The relative performance of the metrics on a variety of datasets is considered in the Section 3.2.4, but before we proceed it is worth noting that there is scope for still further improvement. In principle any set of contours could be used in Figure 3.3, and in particular, at present no account is taken of during which epoch a mutation arises. We know (recall (1.2)) that, denoting the number of mutations while k ancestors by S_n^k,

$$\mathbb{E}(S_n^k) = \frac{\theta}{k-1}. \tag{3.2}$$

It would be desirable to rescale mutation events by this quantity—or even to incorporate further topological information derived from the gene tree—but the major obstacle to this approach is that we can no longer obtain the projection of an intermediate dataset H_{-t}; its distance depends on the entire partially reconstructed genealogy $\mathcal{H}_{-t} = (H_0, \ldots, H_{-t})$. There is no longer a sense of how far a genealogy is from the MRCA, since there are many possible distances—one for each $\mathcal{H}_{-m}$. I do not pursue this further, but simply note that these metrics have the potential for further refinement.

3.2.3 Incorporating recombination

To incorporate recombination to the above discussion, we need to account both for the existence of two mutation processes, one at each locus, and for the recombination events themselves. Let us consider each in turn.

Suppose the dataset is taken from two loci. In the absence of recombination, a suitable extension to each metric is clear. Here I extend sOR—the case for sAND follows similarly. d_{sOR} becomes

$$\begin{aligned} d_{sOR}[(C_1, M_1^A, M_1^B), (C_2, M_2^A, M_2^B)] &= \nu\left(|C_2 - C_1| \right. \\ &\quad \left. + \mu_A |M_2^A - M_1^A| + \mu_B |M_2^B - M_1^B|\right), \end{aligned}$$

where $\mu_A = \frac{n-1}{\mathbb{E}(S_n^A)}$, $\mu_B = \frac{n-1}{\mathbb{E}(S_n^B)}$; S_n^A, S_n^B denote the number of segregating sites at locus A and B respectively. The metric now operates on a three-dimensional projection. Next, re-introduce recombination. The importance sampler now operates on a birth and death process, and the number of coalescence events is no longer fixed. The sample size can rise and fall, and so a natural modification to the stopping-time is

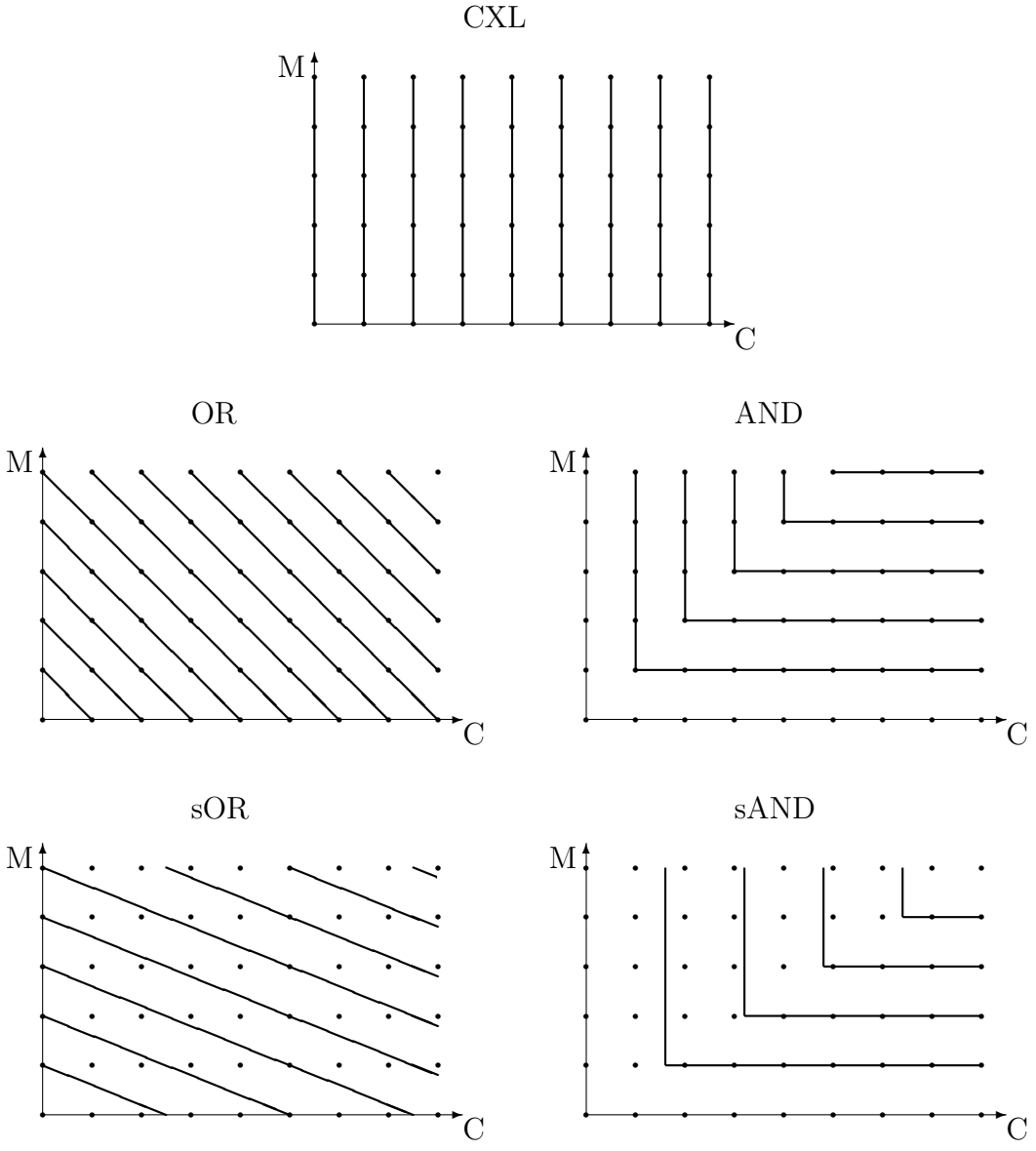


Figure 3.3: Illustration of stopping times (solid lines) described in the text. A sequential importance sampling run can be projected onto a walk across an $(n-1)\times s$ grid, as the number of coalescence events C and the number of mutation events M increment in steps. In the example shown, the rescaling of stopping times under sOR corresponds to a small value of θ; now, more than two coalescence events are required to reach the next stopping-time compared with only one mutation event. The coarseness is set so that each mutation event advances by precisely one stopping-time.

to refer not to the lth coalescence event, but instead to the first time the sample size decreases by l below n. This strategy is also employed by Larribe (2003) [79]. In our setting, in which individual genes are permitted to be non-ancestral at one or other locus, sample sizes can be fractional, and so we need a further modification. Stopping times should refer not to sample size, but to the length of ancestral material $\xi := \frac{1}{2}(a+b+2c)$. Putting all this together we have

$$\begin{aligned} d_{sOR}[(\xi_1, M_1^A, M_1^B), (\xi_2, M_2^A, M_2^B)] &= \nu\left(|\xi_2 - \xi_1|\right. \\ &\quad \left. + \mu_A|M_2^A - M_1^A| + \mu_B|M_2^B - M_1^B|\right), \end{aligned} \tag{3.3}$$

with

$$\begin{aligned} \mu_A &= \frac{a+c-1}{\mathbb{E}(S_n^A)} = \frac{a+c-1}{\theta_A \sum_{j=1}^{a+c-1} \frac{1}{j}}, \\ \mu_B &= \frac{b+c-1}{\mathbb{E}(S_n^B)} = \frac{b+c-1}{\theta_B \sum_{j=1}^{b+c-1} \frac{1}{j}}, \\ \nu &= \frac{n-1}{\xi + \mu_A s_n^A + \mu_B s_n^B}, \end{aligned}$$

where ξ, a, b, c, s_n^A and s_n^B refer to the initial dataset H_0. ν is chosen so that there are $n-1$ stopping times in total, as in CXL, though in neither case do we perform resampling at the last stopping-time (which coincides with reaching the MRCA). Using the definition (3.3), stopping times are then defined by $T_l = \inf\{t \in \mathbb{N} : d_{sOR}[H_{-t}, H_0] \geq l\}$.

What of the recombination events themselves? It is tempting to include them in the metric alongside the other types of event, but recombination events differ in that they do not necessarily bring us closer to the MRCA, nor are the total number

of events known in advance. They occupy a similar role to mutation events under different mutation models, such as the finite-alleles model. In both cases, the event *might* bring us closer to the MRCA in some sense, but it is not guaranteed to, nor is it obvious how to measure this is so. I omit the use of the number of recombination events in these metrics, and defer discussion of possible ways of utilizing them to Section 3.2.5. Let us now consider the relative performance of these stopping times.

3.2.4 Results

As I have discussed, we expect OR and sOR to provide the greater improvements over CXL, and this is borne out by many of the results. For brevity I shall concentrate on the comparison between CXL and sOR, and other metrics will be omitted in places. Where this occurs, it is generally the case that AND and sAND provide little or no improvement over CXL, and OR provides an improvement intermediate between CXL and sOR. AND and sAND can be thought of as a sort of negative control: it is not enough merely to account for mutation events in the metric somehow; they must be incorporated in a useful way.

3.2.4.1 Correlation of weights

The purpose of resampling is that we expect there to be a correlation between the current and final weight of each history. Resampling discards those with a relatively poor current weight and multiplies those with a relatively large current weight, and this process hinges on the current weight acting as a good indicator of final weight. Good stopping times should therefore stop histories at such a time that this correlation is evident. Before actually performing any resampling, we can investigate

the evolution of the correlation of weights during a run for different definitions of stopping times. Figure 3.4 illustrates the correlation between the current and final weight at each stopping-time for an example dataset drawn from $\texttt{ms}(20, 5, 0)$.

As l increases, the correlation between the current and the final weight improves, as one would expect. A heuristic argument can be made that for large t, the current SIS weight W_t is approximately log-normally distributed ([80], Section 5.3), and so a simple measure of this correlation is the sample correlation coefficient υ of these logarithmic plots (I use υ rather than the usual ρ, to avoid confusion with the latter's use as the scaled recombination rate).

In Figure 3.4, the plot for CXL agrees qualitatively with those of other datasets (not shown), and with a similar plot of Larribe (2003) [79] which uses a different proposal. A caveat is that the speed of convergence of correlation depends on the complexity of the dataset, as we shall see below.

It is clear that the correlation in weights improves more rapidly for stopping times defined by sOR than for those defined by CXL. In the example shown, this is most easily observed around the 13th stopping-time. However, it is not obvious from these plots alone whether comparing T_{13} directly, for example, is meaningful. By this measure, one could design a new stopping-time with an excellent correlation at the 13th stopping-time, simply by defining T_{13} to arise much closer to the MRCA. Similarly, we need a way to compare stopping schemes that have a different number of stopping times. One solution is to calibrate the stopping-time in terms of the fraction of the complete genealogy that has been reconstructed. For each stream this can be calculated as the number of steps in the importance sampler up to the stopping-time, divided by the total number of steps required to reconstruct the complete genealogy. When $\rho = 0$ the denominator is $n - 1 + s$, the same across

Figure 3.4: Evolution of correlation of SIS weights. For a dataset drawn from $\texttt{ms}(20, 5, 0)$, 1000 genealogies were reconstructed using $\theta_0 = 5$, $\rho_0 = 0$. The lth plot, reading from left to right, shows current weight (x-axis) versus final weight (y-axis) at the lth stopping-time. Stopping times are defined by CXL (*top*) and sOR (*bottom*). Weights are on a $\log_{10}$ scale; each axis runs from -40 to 0.

streams. A stopping-time can then be calibrated by taking the mean of this fraction across all streams. The distribution of stopping times by the mean fraction of ARG recovered is shown in Figure 3.5 (*left*), for CXL and sOR. Since the number of mutations arising while there are k ancestors *increases* with diminishing k (according to (3.2)), earlier stopping times under CXL have recreated less of the total ARG than later ones. On the other hand, for $\theta = 5$, stopping times under sOR are much more evenly spread out throughout the recovery of the ARG. This implies that, by failing to calibrate stopping times according to the fraction of ARG recovered, the performance of CXL is underestimated. The correlation at T_{13} is notably less for CXL than for sOR, but this is because under CXL T_{13} occurs much earlier! The evolution of the correlation co-efficient with stopping times correctly calibrated is shown in Figure 3.5 (*right*). Even after calibration, the improvement in correlation is more rapid for sOR, and appears to be more consistent for different datasets. Nevertheless, in all schemes there is a considerable dependence on dataset of these correlation curves.

The above procedure was repeated for each $\rho \in \Upsilon$. Figure 3.6 shows a similar plot, but for datasets drawn from $\texttt{ms}(20, 5, 5)$. As ρ increases, the performance of different stopping times start to look more alike, which can be explained as follows. For larger values of ρ, more and more of the steps taken in the reconstruction of the ARG are recombination and re-coalescence events, and these are essentially ignored by all of the stopping-time definitions. Since recombination events are linear in rate compared to quadratic coalescence events, relatively more events occur nearer the MRCA of a reconstructed ARG. This makes the skew in distribution of stopping times under CXL towards the tips of the ARG even more pronounced (Figure 3.6 (*top left*)).

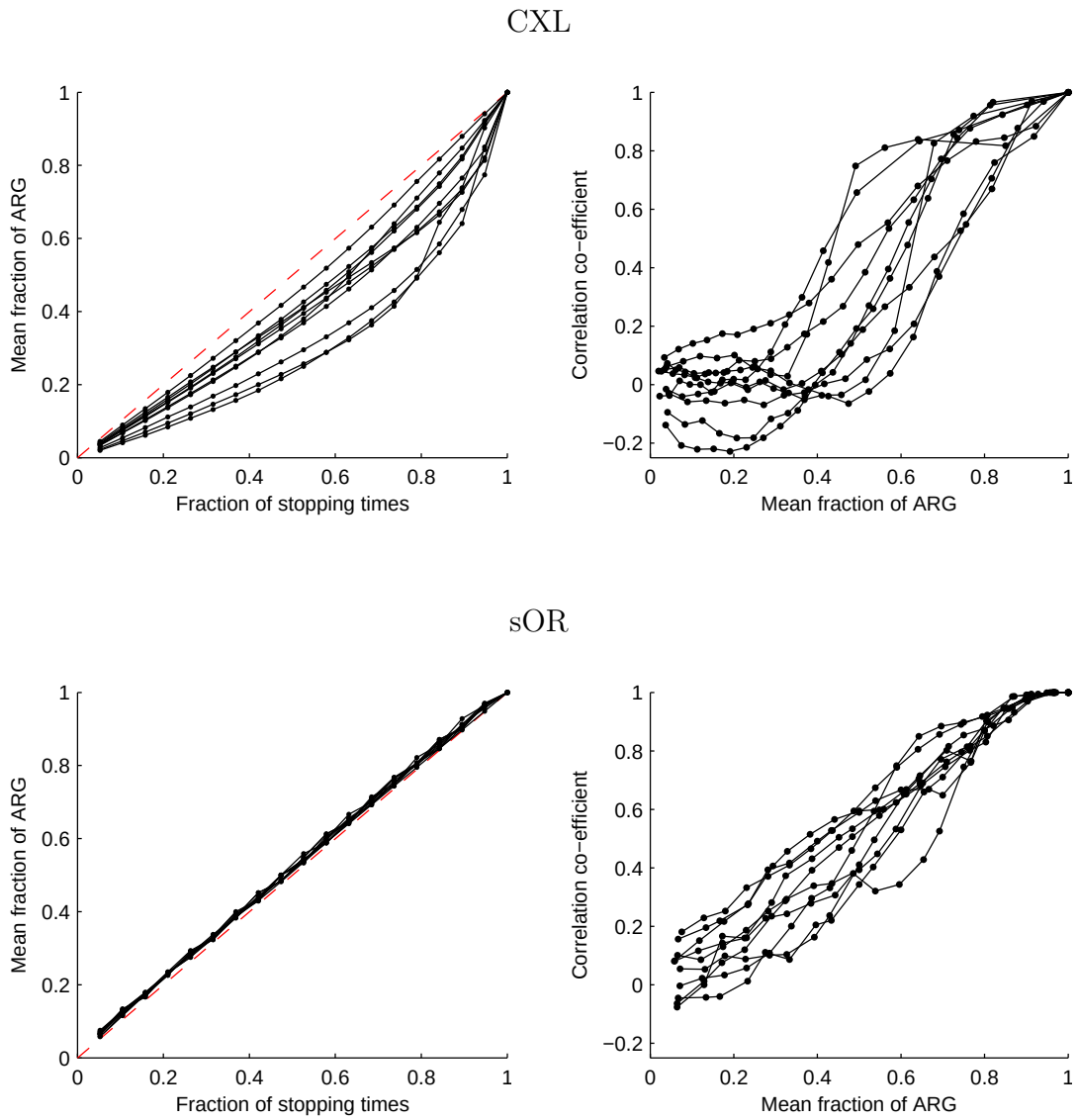


Figure 3.5: The distribution of stopping times throughout the reconstruction of the ARG (*left*) using driving values $\theta_0 = 5$, $\rho_0 = 0$. The distribution is measured by the mean fraction across 1000 streams of the total number of steps which have been used by a given stopping-time. Also shown (*right*) is the improvement in correlation v between the current and final SIS weights during the reconstruction of the ARG, for 10 datasets drawn from $\texttt{ms}(20, 5, 0)$. Stopping times are defined by CXL (*top*) and sOR (*bottom*).

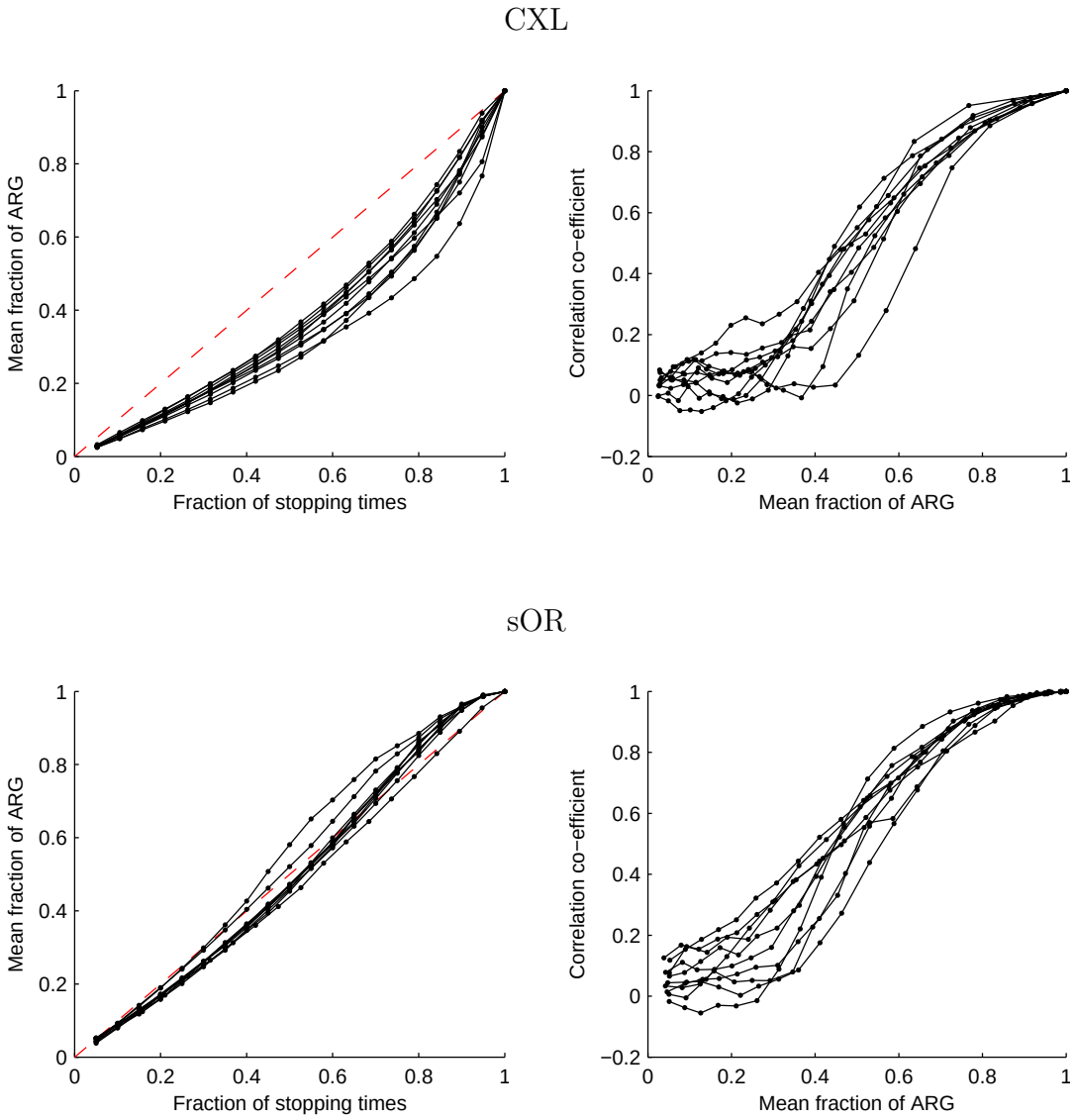


Figure 3.6: A similar plot to Figure 3.5, but for 10 datasets drawn from $\texttt{ms}(20, 5, 5)$, and using driving values $\theta_0 = 5$, $\rho_0 = 5$.

To compare the improvement in correlation directly, I took the mean of these correlation trajectories across datasets (using linear interpolation between stopping times), and plotted the resulting trajectories together. This was repeated for all schemes and for values of $\rho \in \Upsilon$: datasets were drawn from $\texttt{ms}(20, 5, \rho)$, and in the reconstruction of ARGs the corresponding driving value was also set to ρ. Results for several values of ρ are shown in Figure 3.7. For all values of ρ tested, sOR has the greatest correlation throughout the importance sampling run (with some brief exceptions). The improvement over CXL is greatest at $\rho = 0$, and as ρ increases all the schemes begin to behave more similarly, as noted above. For $\rho = 10^{10}$, a computational approximation to $\rho = \infty$, the schemes are almost indistinguishable, and the growth in correlation is approximately linear. At $\rho = 0$, OR and sOR display very similar behaviours; only for intermediate values of ρ is the new scaling that I have introduced a noticeable improvement over OR. This difference is greatest around $\rho = 10$. Although the difference between OR and sOR is generally small, all the investigation so far has been for $n = 20$ and $\theta = 5$, for which $\mu_A = \mu_B = 2.14$. This value is apparently not too far from 1, at which we have OR = sOR; the latter is expected to perform better away from this exchange rate. The effect of varying θ will be investigated in Section 3.2.4.4.

3.2.4.2 Fine-tuning the dynamic resampling parameter

In Section 3.2.4.1 we discovered that the new stopping schemes improve the correlation between the current and final weight of an importance sampling run, and this holds throughout the run. It remains to investigate whether this translates into an improvement in accuracy of likelihood estimates, assuming the dynamic resampling parameter B can be chosen correctly for each scheme. To deal with this assumption,

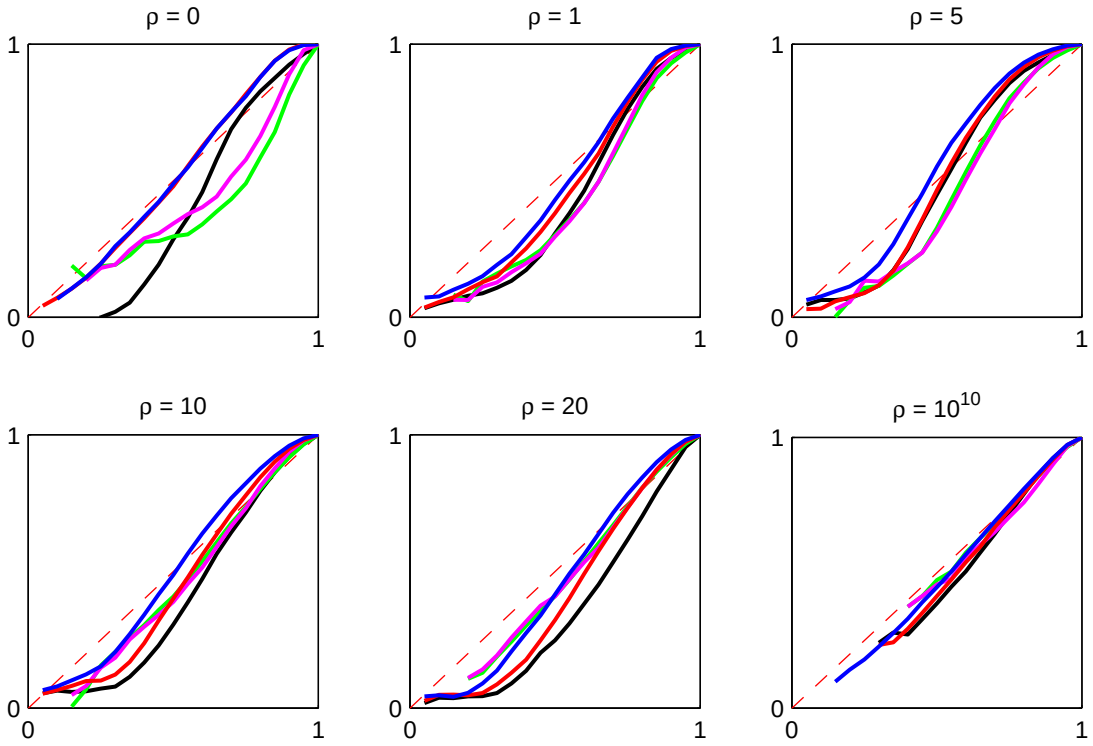


Figure 3.7: The correlation co-efficient v (y-axis) of current SIS weight against final weight, versus the mean fraction of ARG recovered at each stopping-time (x-axis). Plots are averaged across 10 datasets drawn from $\texttt{ms}(20, 5, \rho)$; 1000 genealogies were reconstructed for each dataset and for each scheme, using driving values $\theta_0 = 5$ and $\rho_0 = \rho$. Stopping schemes are shown as: CXL (black), OR (red), sOR (blue), AND (green) and sAND (magenta).

Stopping scheme	$\hat{L}(1)$	Standard error	Relative error	ESS	Resampling events
None	3.828×10^{-7}	1.524×10^{-9}	3.399×10^{-3}	8631	0
CXL	3.705×10^{-7}	8.614×10^{-9}	-2.903×10^{-2}	1561	2
OR	3.847×10^{-7}	1.058×10^{-9}	8.256×10^{-3}	9297	3
sOR	3.817×10^{-7}	0.206×10^{-9}	4.145×10^{-4}	9971	1
AND	3.817×10^{-7}	9.309×10^{-9}	5.288×10^{-4}	1440	2
sAND	3.829×10^{-7}	0.848×10^{-9}	3.669×10^{-3}	9532	1
'True' value	3.815×10^{-7}	4.611×10^{-11}	0	8725613	0

Table 3.1: Likelihood estimates of dataset given in Figure 3.2 for $\theta = 1$ under various stopping schemes, together with estimated standard errors, estimated ESS, and number of resampling events incurred. Estimates are based on $10,000$ runs with $B = 1$, except for the 'true' value which is based on $10,000,000$ runs with $B = \infty$. The 'true' value is used to estimate the relative errors.

here I will investigate the accuracy of likelihood estimation and the effect of varying B simultaneously.

As a simple confirmation of the potential of these new schemes, first return to the example dataset given in Figure 3.2. Likelihood estimates $\hat{L}(\theta)$ evaluated at $\theta = 1$ under each stopping scheme are given in Table 3.1.

Again, this example was chosen to emphasize the improvement in the new schemes, but the differences are striking. CXL significantly reduces the accuracy of the likelihood estimate, and this is reflected in a larger relative error from the true value, a greater standard error, and a reduced ESS. OR, sAND, and sOR result in the reverse trend by each of these measures, and sOR provides the most accurate estimate—reducing the relative error by an order of magnitude, though the complexity of this dataset is such that even without resampling the likelihood estimate is reasonable. For this reason, we shall proceed by looking at some larger simulated datasets. Firstly though, Table 3.1 raises some further issues of which we should be

wary. $B = 1$ was chosen so that a reasonable number of resampling events would be incurred by each scheme, but because of the variation in placement and number of stopping times—as well as the stochasticity inherent in the importance sampler—this will result in a different number of resampling events between schemes. One cannot be sure *a priori* whether variation in likelihood estimates is simply due to the incursion of different numbers of resampling events for different schemes. In fact—for this example at least—varying B between schemes to ensure the same number of resampling events does not result in any qualitative changes to the data in the table (not shown). Nevertheless, one should still take this into account before making any inferences, as I shall do below. How well specified the number of resampling events is by a particular value of B is also considered in Section 3.2.4.3. Secondly, I have chosen to include both the standard error and ESS as diagnostic tools in assessing the accuracy. This is because, for realistic examples, an accurate estimate of the true likelihood will obviously be unavailable. It is important that these estimates of accuracy are good indicators of the true, underlying and hidden, estimate of accuracy—such as that reported by the relative error. The ESS is not without its flaws, however, as discussed in Section 2.3.4.2. The ESS is really a measure of the diversity of the sample, which is an indirect indication of the accuracy of the likelihood estimate. Phenomena which reduce the diversity of the sample without necessarily improving the likelihood estimate will create a discrepancy between the reported ESS and the true accuracy of the estimate. One such phenomenon has been mentioned already: an extremely poor IS proposal distribution with all its mass concentrated around very few genealogies will yield both a poor likelihood estimate and a high ESS. Another is the resampling procedure, which can only reduce the diversity of the sample for an expected *future* payoff in accuracy. Resampling

too often will increase the ESS. Resampling so much such that only one genealogy is left represented in the sample will result in an optimal ESS of N, but it does not follow that the resulting likelihood estimate will be optimal. In the absence of other diagnostic tools, we proceed anyway with surveilling the ESS—and for Table 3.1 at least, the relationship between ESS and relative error is as we would hope.

To investigate the effect of varying B and using each stopping scheme, I performed the following experiment, based on the observations above. Draw 10 datasets from $\texttt{ms}(20, 5, \rho)$ and for each dataset run the importance sampler driving at these parameters, i.e. $\theta_A = \theta_B = 2.5$ and ρ itself, for $10,000$ runs. Resample according to the schedule parameter B. Repeat this 25 times independently, to obtain a set of independent likelihood estimates. Repeat all of this for each stopping scheme; for each $B = 2^k$, $k = -6, \ldots, 15$; and for each $\rho \in \Upsilon$. The chosen range of B encompasses in most cases the entire range of responses, from resampling at every step to no resampling. Datasets of this size were chosen such that they would be complicated enough for different stopping schemes to have detectably different effects on likelihood estimates, yet simple enough that computational power is available to perform a number of experiments with a wide range of parameter values, and simple enough to obtain one accurate estimate from $10,000,000$ runs without resampling. This permits the continued estimation of an absolute relative error as the 'true' measure of accuracy (assuming there is no systematic bias in our estimates). Examples of the results produced are shown for a dataset drawn from $\texttt{ms}(20, 5, 0)$ (Figure 3.8), a dataset from $\texttt{ms}(20, 5, 5)$ (Figure 3.9), and a dataset from $\texttt{ms}(20, 5, 10^{10})$ (Figure 3.10). The basic measure of accuracy used in these plots is the median absolute relative error across 25 independent likelihood estimates.

One can make a number of observations from these figures and from the re-

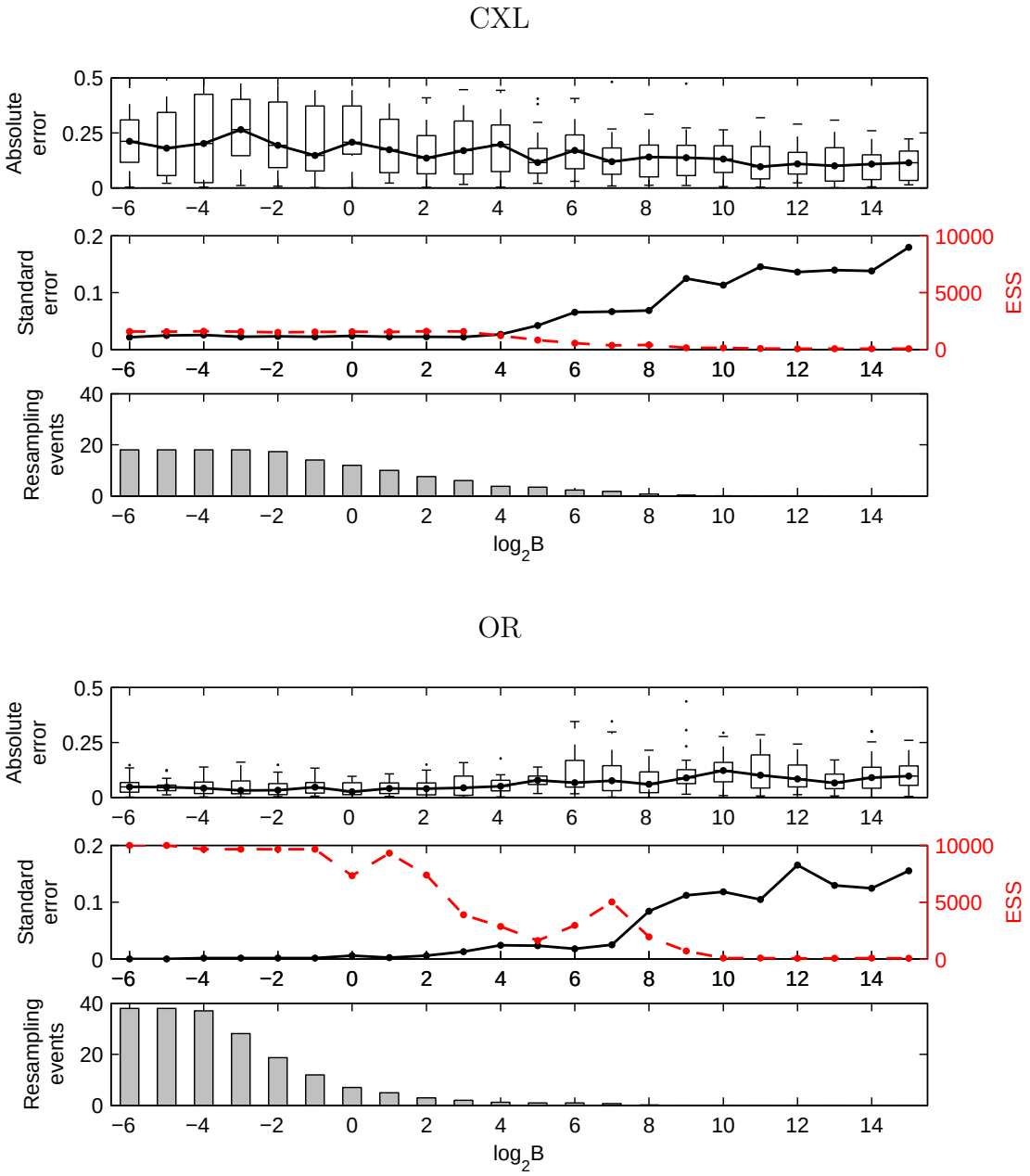


Figure 3.8: Accuracy of likelihood estimates for an example dataset drawn from $\texttt{ms}(20, 5, 0)$, and importance sampling performed at these values. Results are for CXL (*top*), OR (*bottom*), and sOR (*next page*). For each stopping scheme and for various values of B, 25 independent experiments of $10,000$ runs were performed: shown is a boxplot for absolute errors from each set of experiments, the mean reported standard error, mean ESS (*dashed line*), and mean number of resampling events. Errors are given relative to the estimated true likelihood of 4.096×10^{-20}.

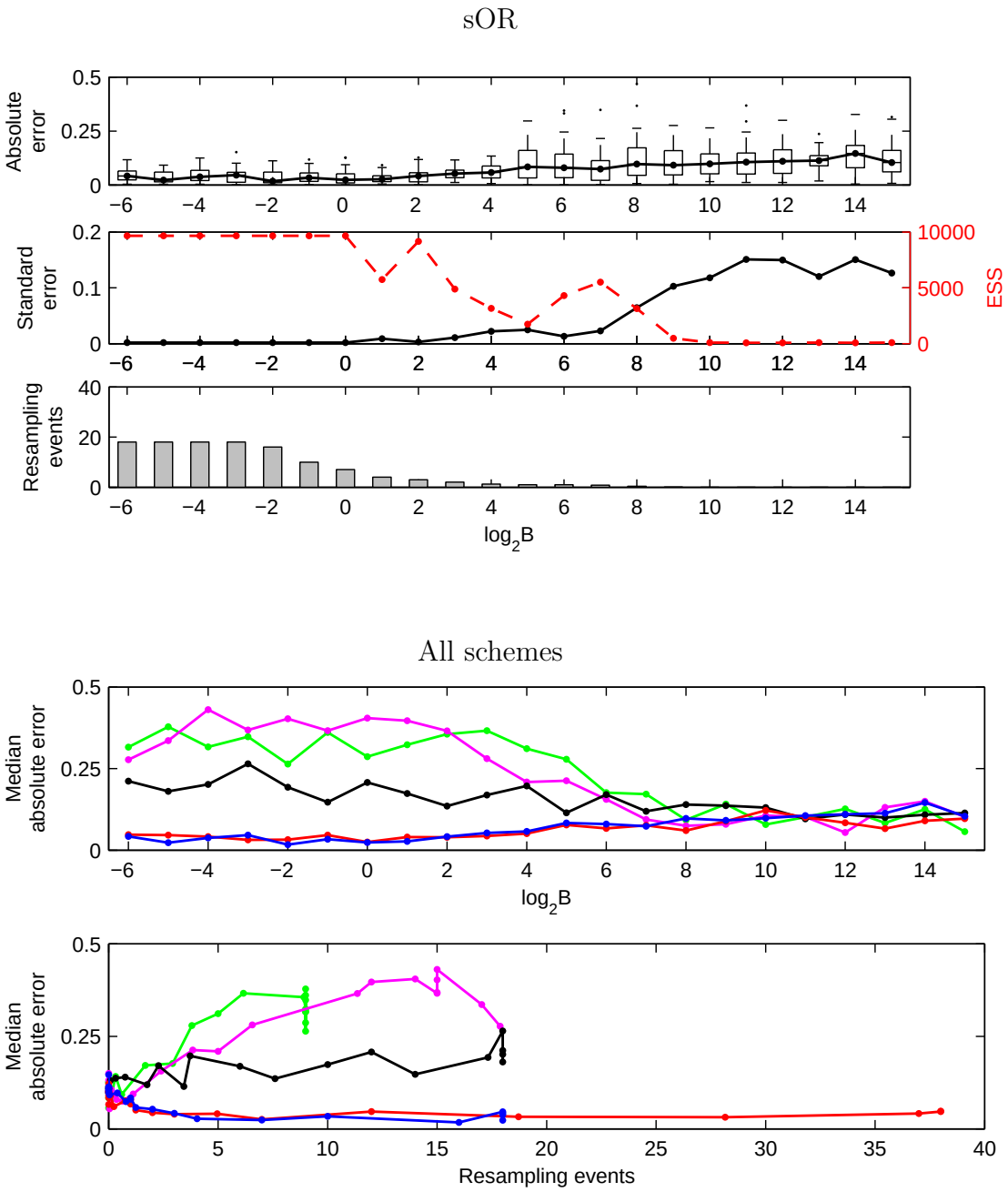


Figure 3.8: (*Continued*). *Bottom*: A comparison of (median) absolute relative error for each stopping scheme, plotted against B and against the number of resampling events. Stopping schemes are shown as: CXL (black), OR (red), sOR (blue), AND (green) and sAND (magenta).

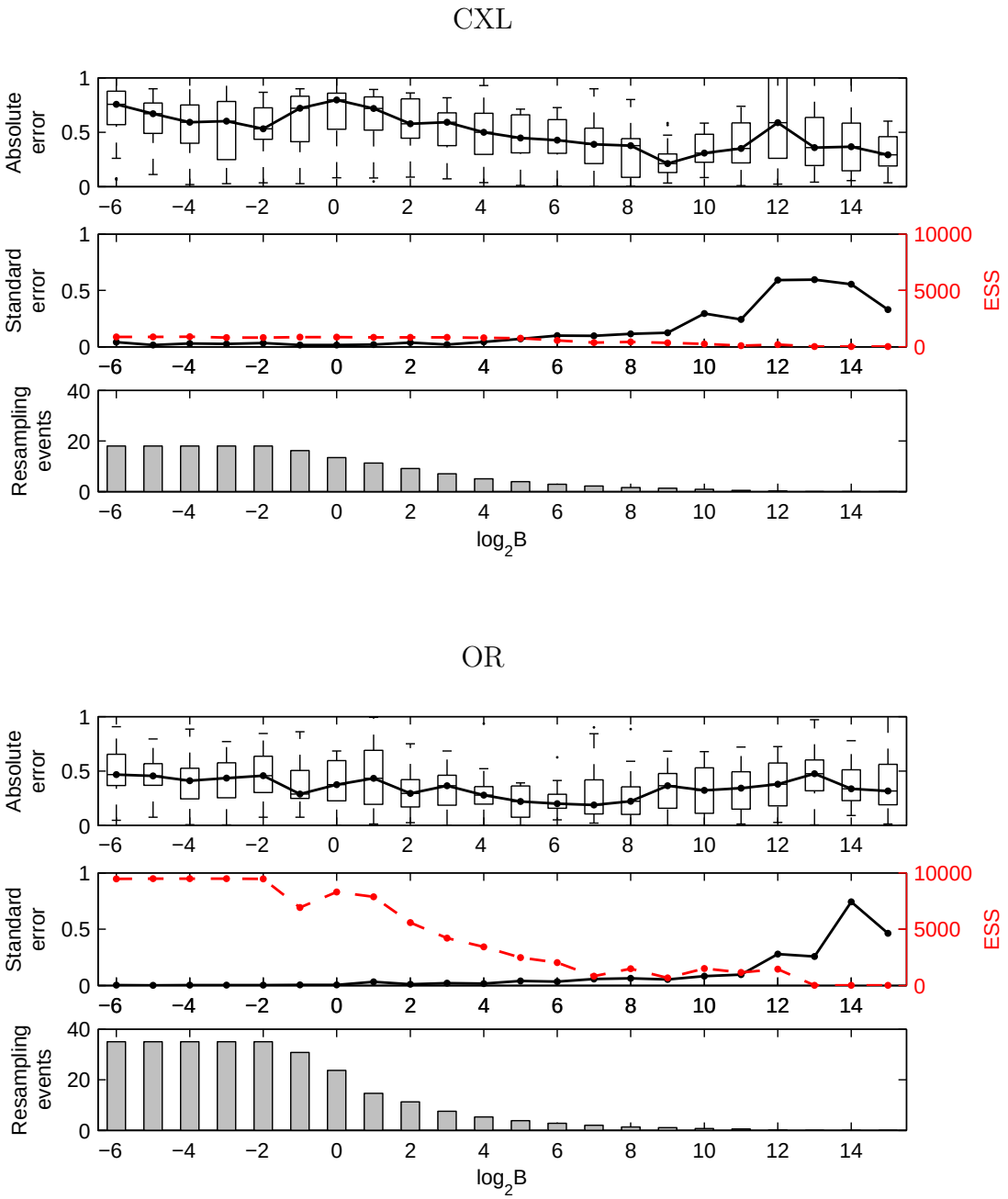


Figure 3.9: A similar plot to Figure 3.8, but for a dataset drawn from `ms`$(20, 5, 5)$, and using driving values $\theta_0 = 5$, $\rho_0 = 5$. Units for absolute error and standard error are given relative to the estimated true likelihood of 2.981×10^{-18} (i.e. the top graphs are of the absolute relative error).

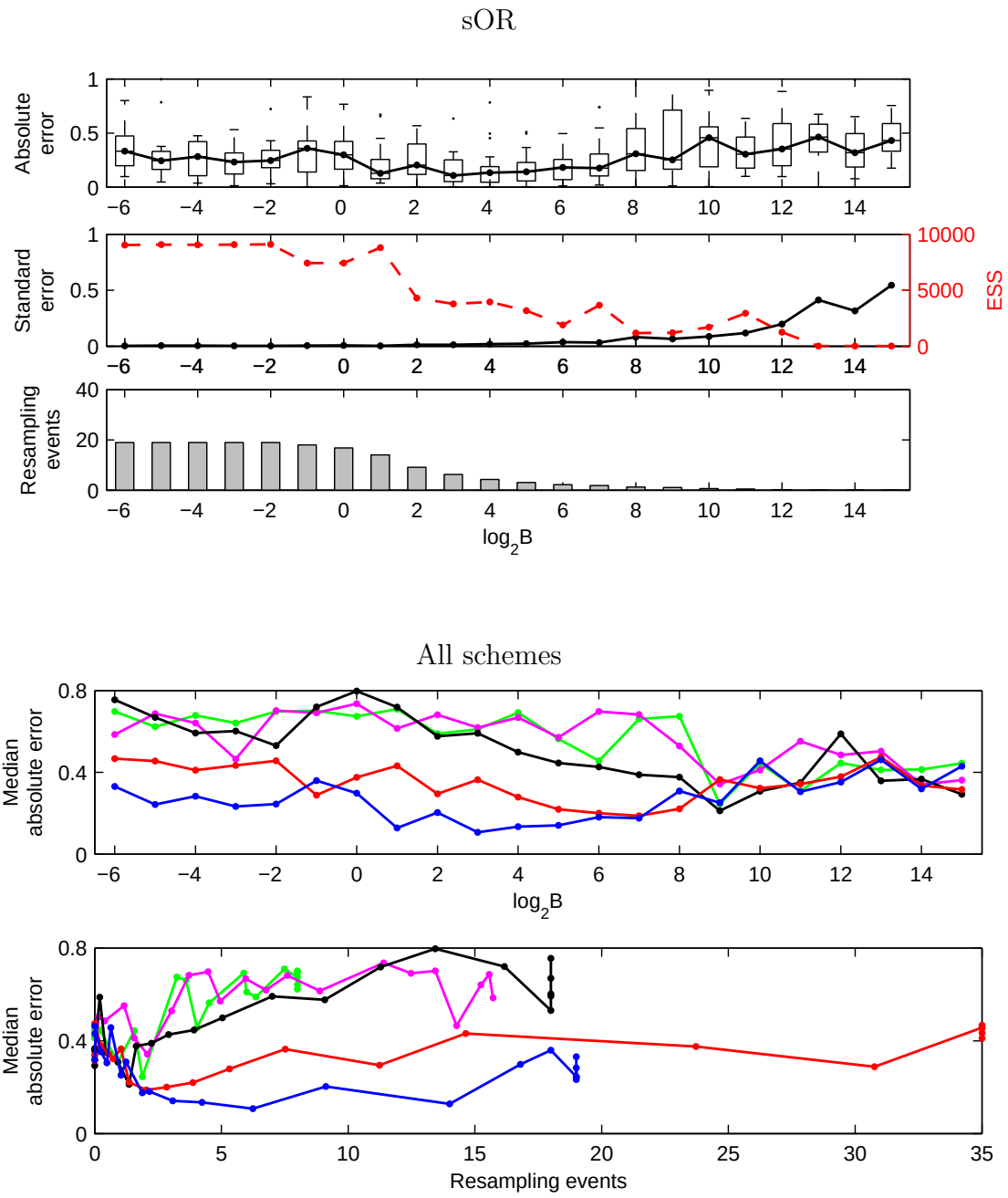


Figure 3.9: (*Continued*).

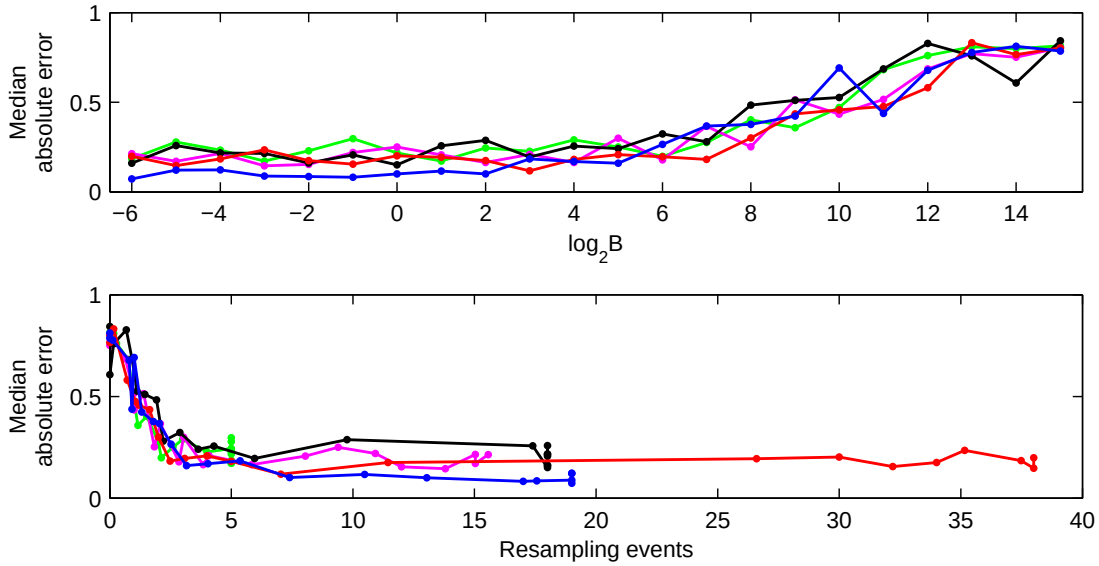


Figure 3.10: A similar plot to Figure 3.8, but for a dataset drawn from $\texttt{ms}(20, 5, 10^{10})$, and using driving values $\theta_0 = 5$, $\rho_0 = 10^{10}$. Units for absolute error are given relative to the estimated true likelihood of 6.745×10^{-19}.

maining results (not shown for brevity). As one increases B from 0 towards ∞, the mean number of resampling events gently declines from its maximum towards 0, as we would expect. The other plotted measures show visible trends too. Most importantly, we can see the effect that resampling has on the accuracy of the likelihood estimate, as evinced by the plot of the absolute relative error. These trends are obtained from a distribution with a large variance, particularly for CXL whose inter-quartile range is relatively large, but in most cases a clear pattern is still visible. For the example given in Figure 3.8, as B increases there is a clear *improvement* in accuracy under CXL; resampling generally diminishes the quality of the sample. This agrees with the intimations of the results in the previous section. On the other hand, the trends for OR and sOR show a *reduction* in accuracy as B increases; resampling improves the quality of the sample. Moreover—for these parameter values at least—as B approaches 0 and more and more resampling events take place,

the absolute relative error does not begin to increase. That is, doing too much resampling does not worsen the likelihood estimate significantly, and if the absolute relative error has a global minimum away from $B = 0$ then the plot is so flat that it is difficult to pick out. So when confronted with this sort of dataset, the advice for tuning B is then clear: setting it too low is preferable to setting it too high. In the former case, the only price seems to be paid in the modest computational burden of performing the resampling. Provided $\log_2 B \leq 4$ and OR or sOR is used, the median absolute relative error comfortably falls within 0.1, and reducing B further offers little improvement.

The coloured plot in Figure 3.8 compares the various schemes directly, calibrated both by B and by the number of resampling events carried out by each scheme, using the same data. In terms of absolute relative error, the superiority of OR and sOR is clear (as is the inferiority of AND and sAND!). For a given amount of resampling, there appears to be an almost negligible difference between OR and sOR at these parameter values. The advantage of sOR is in the fact that the number of resampling events is usually much fewer; sOR is therefore far easier to tune with respect to B. In this dataset, for small values of B, OR employs twice as many resampling events for no improvement in accuracy. This is likely to be even more exaggerated for other choices of parameter values (see Section 3.2.4.4). It should also be noted that these trends were generally repeated for other datasets drawn using the same parameters, albeit with exceptions. In particular, $N = 10,000$ gave very accurate likelihood estimates for some datasets, such that resampling could provide no further improvement. For example, of the ten datasets drawn from $\texttt{ms}(20, 5, 0)$ four showed no obvious visible improvement under sOR for any value of B, and in these cases the relative error rarely exceeded 0.04 anyway.

Our diagnostic tools, ESS and s.e., also display clear trends as B varies in these graphs. There is a general tendency for the estimates of the ESS to be largest where the likelihood estimates are most accurate, but with some important caveats. Observe the graphs for CXL in isolation. The ESS is low throughout, but is lowest for large B. Taken on its own this might lead one to infer that resampling actually improves the estimate, which is not the case. Instead, all we can do is appraise the ESS in absolute terms—compared to an optimal value of $10,000$, the ESS is poor for all B, suggesting a poor likelihood estimate. But we must also take care when dealing with a large ESS. The trends for OR and sOR show a declining ESS with increasing B—this time correctly—yet they report *optimal* values of $10,000$ for a range of small values of B when the relative error is clearly non-zero. This is a consequence of the ESS measuring accuracy indirectly by reporting the diversity of a sample, when resampling has manipulated this diversity artificially. Together, these features agree with the observation of Fearnhead & Donnelly (2001) [36], that "while a low estimated ESS is indicative of a poor estimate, a large estimated ESS does not guarantee an accurate one". We should continue to be vigilant when making inferences from the ESS, and similar warnings should be considered when using the reported standard error.

Figure 3.9 shows a similar plot, but for a dataset drawn from $\texttt{ms}(20, 5, 5)$. Qualitatively, many features of these results are similar to the case $\rho = 0$: CXL generally diminishes the estimate while OR and sOR improve it, the latter more so. The estimated values for ESS and s.e. provide only indirect guides to accuracy. These observations are seen in other simulated datasets. The improvement of OR and sOR over CXL this time is slightly less distinguished, as discussed in Section 3.2.4.1; also clear here is the superiority of sOR over OR. There is another important feature

to be seen in these graphs, and it is this. In all three of the graphs of CXL, OR, and sOR, doing too much resampling can *diminish* the quality of the sample; there appears to be a global minimum in the absolute relative error for all three schemes around 2–5 resampling events. (This range is of course a function both of the dataset involved and of N.) Aside from the small computational cost incurred from resampling, this provides a further incentive not to reduce B indefinitely. Although the minimum is relatively flat, this pattern was observed in a number of other datasets generated from large values of $\rho \in \Upsilon$—except for $\rho = 10^{10}$, for which no amount of resampling appeared to be too much (Figure 3.10). As in Section 3.2.4.1, the performances of different schemes begin to look very similar for large ρ. Finally, it is also worth noting that of all the datasets investigated, a small number actually suffered after resampling under any of the schemes, as discussed above.

As an aside, instead of using the median absolute relative error as a guide to accuracy, I also considered the mean absolute relative error and the co-efficient of variation

$$\widetilde{cv} = \frac{\sqrt{\frac{1}{K}\sum_{j=1}^{K}\left[\hat{L}_j(\theta_A, \theta_B, \rho) - L(\theta_A, \theta_B, \rho)\right]^2}}{L(\theta_A, \theta_B, \rho)},$$

where $\hat{L}_j(\theta_A, \theta_B, \rho)$ is the likelihood estimate from the jth of K experiments, and $L(\theta_A, \theta_B, \rho)$ is the 'true' likelihood. $\widetilde{cv}$ can be seen as a normalized version of mean squared error (MSE), and differs from the mean absolute relative error only in using the square of the differences in the sum (and then square-rooting) rather than the absolute difference. But these two alternatives suffer from essentially the same problem, which is that the distribution of likelihood estimates has a very large tail. The occasional experiment can have an error of several orders of magnitude,

entirely swamping any signal. I found that, qualitatively, these measures gave similar outputs—only far more noisily. Thus I have focused on the median as a more useful measure, as has been noted by others (e.g. [55, 36, 50], though these authors were discussing the distribution of the MLE rather than the actual likelihood). This point can be illustrated by re-considering the data in Figures 3.8, 3.9, and 3.10; these plots are repeated in Figures 3.11, 3.12, and 3.13, this time using the mean absolute relative error rather than the median.

As is evident from these figures, the general trends are the same, particularly for the 'easy' dataset generated using $\rho = 0$. But the volatility in the standard errors, and indeed in the means themselves, suggest that for the occasional experiment, the likelihood estimate is very far from the truth—enough even to affect the mean of 25 experiments. For example, this is clearly the case in Figure 3.12 (OR—the middle plot, at $\log_2 B = 1$) and in Figure 3.13 (sOR—lower plot, at $\log_2 B = 14$, for which the mean absolute relative error is greater than 10), but these freak occurrences seem to be independent of stopping scheme and of B, and are observed not infrequently. In the absence of resampling, the occasional large weight can result in a large overestimate of the likelihood. What these plots demonstrate is that there is the possibility that resampling will not always cure the problem. By over-relying on them, the resampling procedure can duplicate these large weights and perhaps even exacerbate the problem.

3.2.4.3 Robustness of the resampling procedure

In the previous section, calibrations were made against the mean number of resampling events across experiments, but it is worth noting that the distribution of resampling events is tight about this mean; the number is well specified by the choice

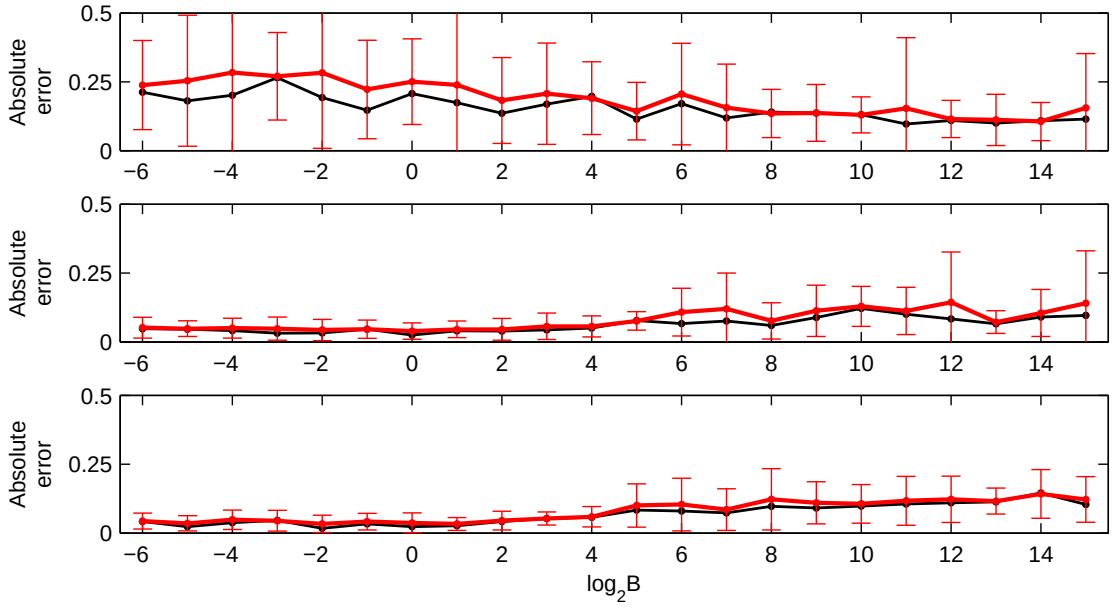


Figure 3.11: The same simulation results as in Figure 3.8 for CXL (*top*), OR (*middle*), and sOR (*bottom*), comparing mean absolute relative error (red line, with standard errors) with the median absolute relative error (black line).

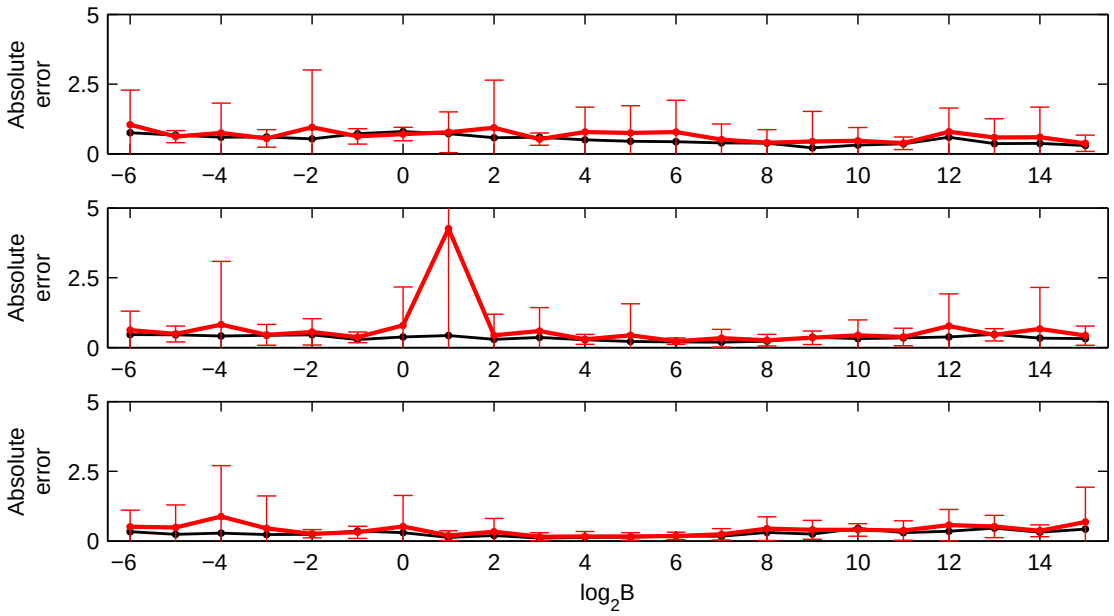


Figure 3.12: The same simulation results as in Figure 3.9 for CXL (*top*), OR (*middle*), and sOR (*bottom*), comparing mean absolute relative error (red line, with standard errors) with the median absolute relative error (black line).

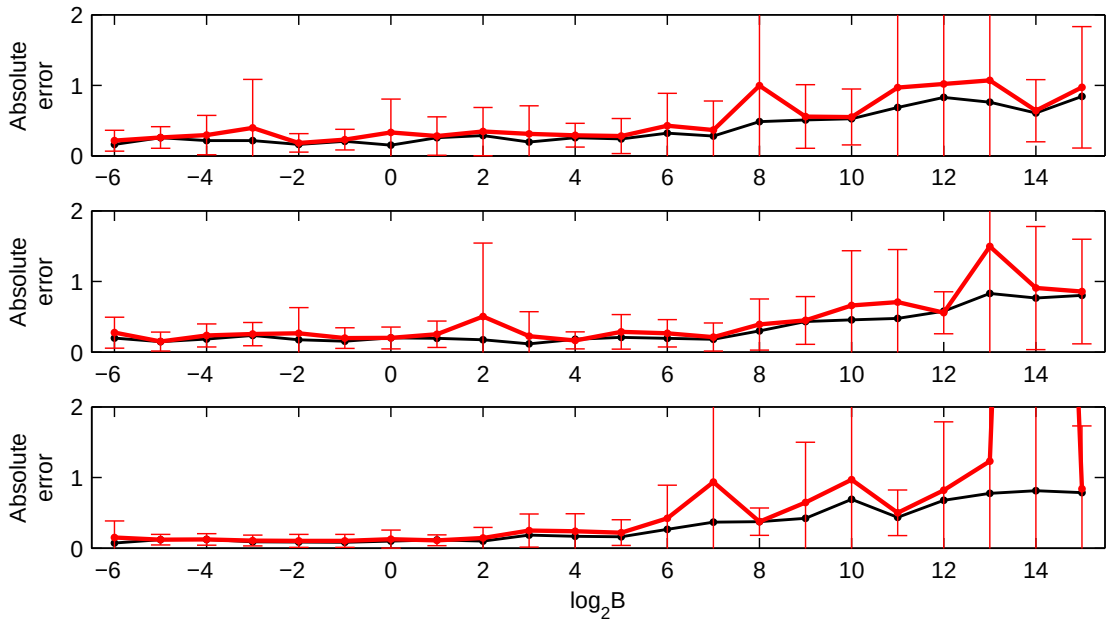


Figure 3.13: The same simulation results as in Figure 3.8 for CXL (*top*), OR (*middle*), and sOR (*bottom*), comparing mean absolute relative error (red line, with standard errors) with the median absolute relative error (black line).

of B. To illustrate, the distribution of the number of resampling events under CXL for the dataset of Figure 3.8 is given in Table 3.2. Similar results were obtained for larger ρ (though with slightly more variance) and for other stopping schemes.

The trajectory of cv^2 during an experiment will follow a sawtooth pattern as it drifts upwards and then becomes zero after each resampling event. As $N \to \infty$, this trajectory and the number of resampling events become deterministic. Table 3.2 shows that even for smaller N (and even for $N = 100$—data not shown) this trajectory is reasonably robust to the stochasticity of the importance sampler. This supports the observation of Figure 3.1, that the variance of the normalized weights during a run can be strongly controlled by the topology of the trees induced by the data.

	$\log_2 B$														
	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10	11
I.N.M. -2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
I.N.M. -1	0.00	0.00	0.00	0.00	0.00	0.40	0.00	0.28	0.00	0.00	0.24	0.24	0.00	0.00	0.00
I.N.M.	1.00	0.68	1.00	1.00	1.00	0.60	1.00	0.72	0.56	0.72	0.76	0.76	0.68	0.92	1.00
I.N.M. $+1$	0.00	0.32	0.00	0.00	0.00	0.00	0.00	0.00	0.44	0.28	0.00	0.00	0.32	0.08	0.00
I.N.M. $+2$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Value of I.N.M.	18	17	14	12	10	8	6	4	3	2	2	1	0	0	0
I.N.M. -2	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
I.N.M. -1	0.00	0.00	0.00	0.04	0.04	0.48	0.16	0.48	0.20	0.20	0.12	0.00	0.00	0.00	0.00
I.N.M.	1.00	0.60	1.00	0.96	0.96	0.52	0.80	0.48	0.72	0.68	0.60	0.68	0.96	1.00	1.00
I.N.M. $+1$	0.00	0.40	0.00	0.00	0.00	0.00	0.00	0.04	0.08	0.08	0.24	0.28	0.04	0.00	0.00
I.N.M. $+2$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.04	0.04	0.00	0.00	0.00
Value of I.N.M.	18	17	14	12	10	8	6	4	3	2	1	0	0	0	0

Table 3.2: *Top*: The distribution of the number of resampling events about its mean, for each set of experiments in Figure 3.8 using stopping scheme CXL. For each value of B, shown is the frequency of the 25 experiments about the integer nearest the mean (I.N.M.) of that set of experiments. In no experiment is the number of resampling events more than 2 away from the I.N.M. *Bottom*: the same procedure, repeated for only 1,000 runs in each experiment. The distribution is only slightly more spread about the I.N.M.

θ	$\mathbb{E}(S_n)$	μ_A, μ_B
0.2	0.9	109.4
2.0	9.0	10.9
50.0	224.0	0.4

Table 3.3: Examples of values of θ for which, with $n = 50$, the exchange rates μ_A and μ_B are not close to 1. Such values cause OR and sOR to diverge in their behaviour.

3.2.4.4 The effect of mutation rate on stopping-times

Thus far I have considered only datasets simulated from $n = 20$ and $\theta = 5$, for which $\mu_A = \mu_B = 2.14$. We neither expected nor observed significant differences between OR and sOR for many of the examples considered. In this section I set $n = 50$—a more realistic sample size for which importance sampling becomes relevant—and investigate the effect on likelihood estimates of importance sampling on datasets generated under different values of θ. A selection of results are illustrated here, with exchange rates μ_A and μ_B given in Table 3.3. For each of these exchange rates I performed the following experiment. Simulate a dataset from $\texttt{ms}(50, \theta, 0.1)$, and run the importance sampler using these parameters as driving values to generate a likelihood surface for θ_A. Repeat this five times independently for each stopping scheme. A cross-section of the complete likelihood hypersurface, with θ_B and ρ fixed, was chosen for ease of exposition. To ensure that the simulated dataset was not too atypical of these parameter values, I repeatedly simulated from $\texttt{ms}(50, \theta, 0.1)$ and accepted only the first dataset to satisfy $|\mathbb{E}(S_n) - s| \leq 2$. I chose $\rho = 0.1$ as a value not so large that the stopping schemes begin to behave similarly, yet not so small that the stopping schemes need not have to deal with recombination events. The number of runs to generate each curve was chosen to be small enough that there existed variation between the five curves, and hence resampling would still be beneficial. Results are presented in Figures 3.14, 3.15, and 3.16.

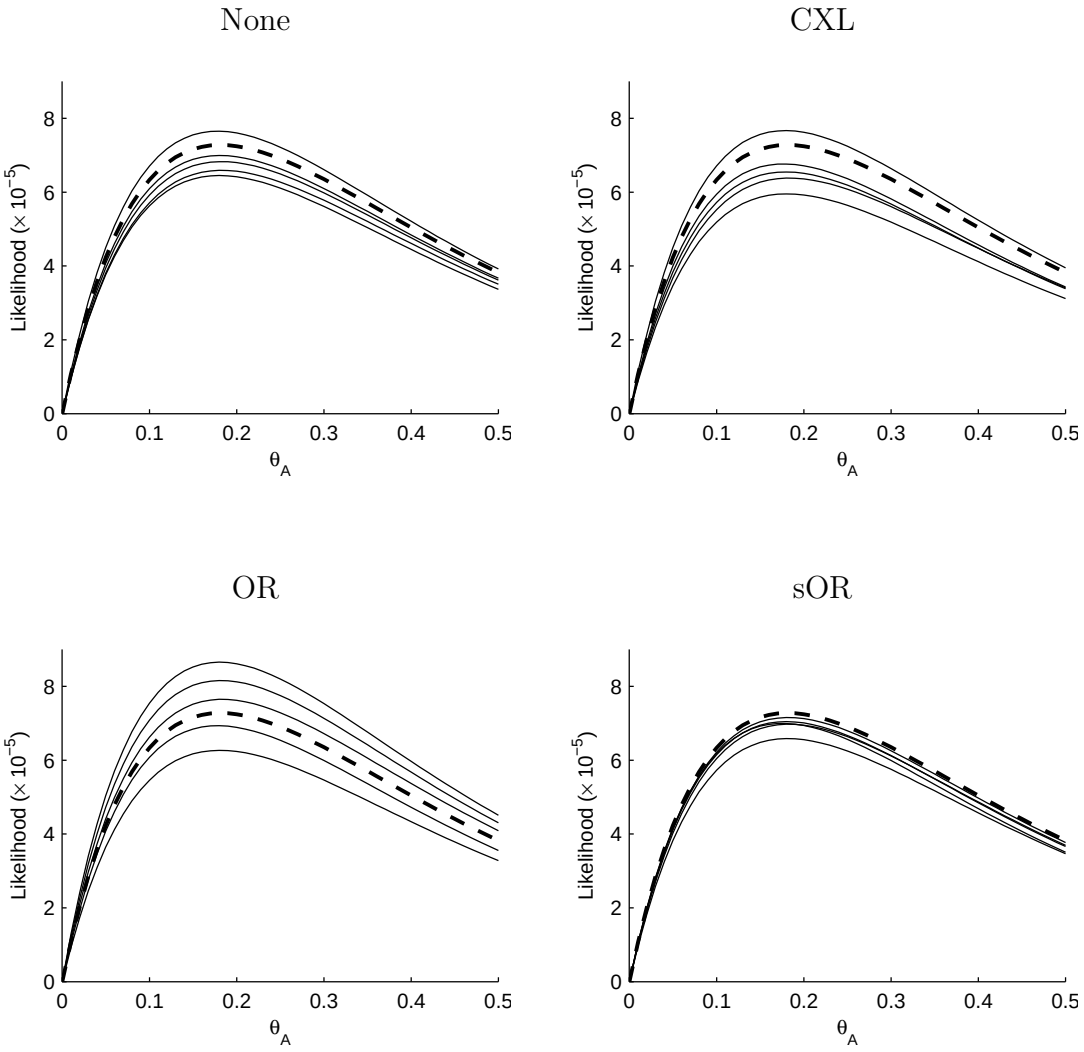


Figure 3.14: Five independent likelihood curves for θ_A using a dataset drawn from `ms`$(50, 0.2, 0.1)$; the importance sampler used these parameters as driving values, and each curve is based on 100 runs. Results are shown for no resampling (*top left*), CXL (*top right*), OR (*bottom left*), and sOR (*bottom right*); each stopping scheme took place according to $B = 1$. The dashed curve shows the 'true' surface, estimated from 10,000,000 runs at $B = \infty$.

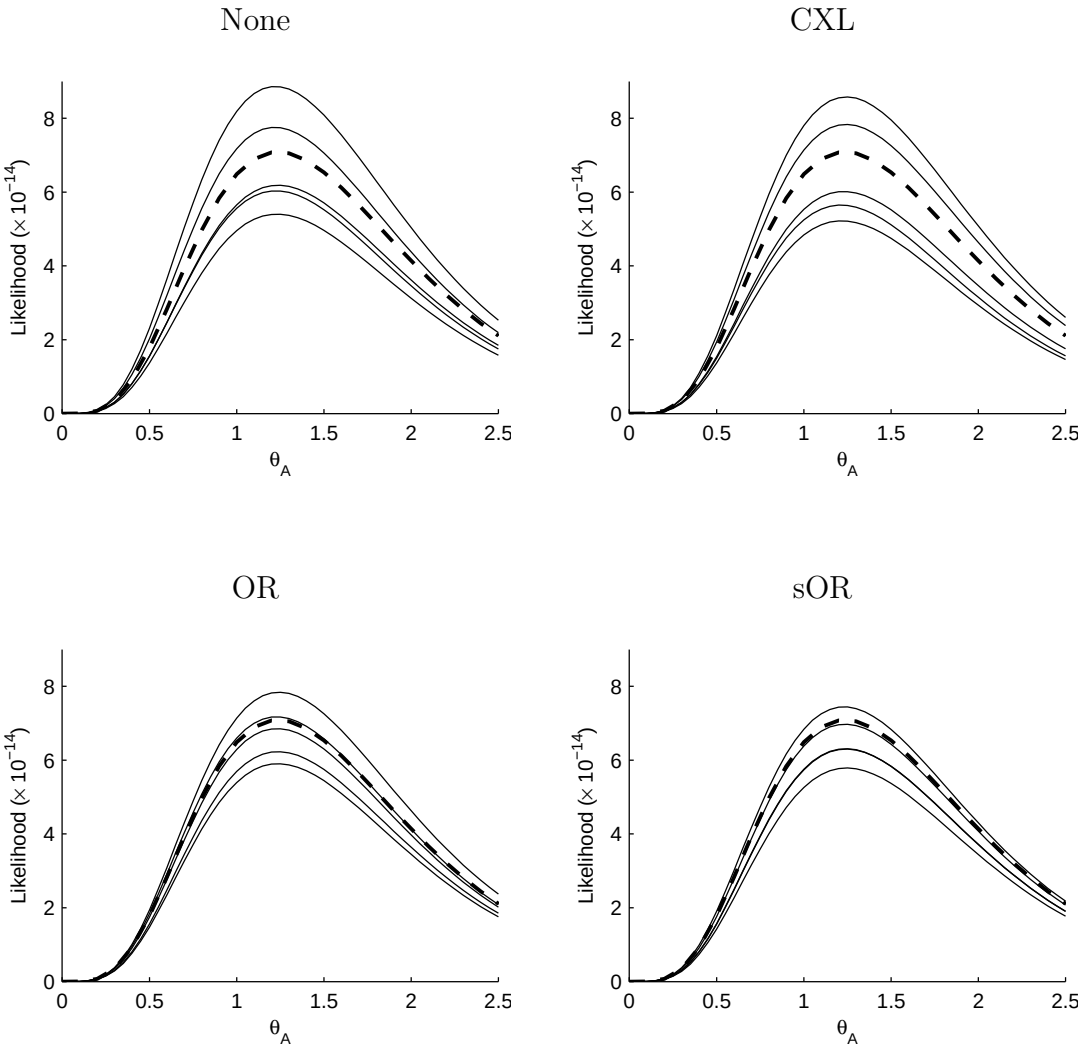


Figure 3.15: A similar plot to Figure 3.14, but for a dataset drawn from `ms`$(50, 2, 0.1)$, and using driving values $\theta_0 = 2$, $\rho_0 = 0.1$, $N = 1000$. To ensure a reasonable amount of resampling, here I set $B = 10$.

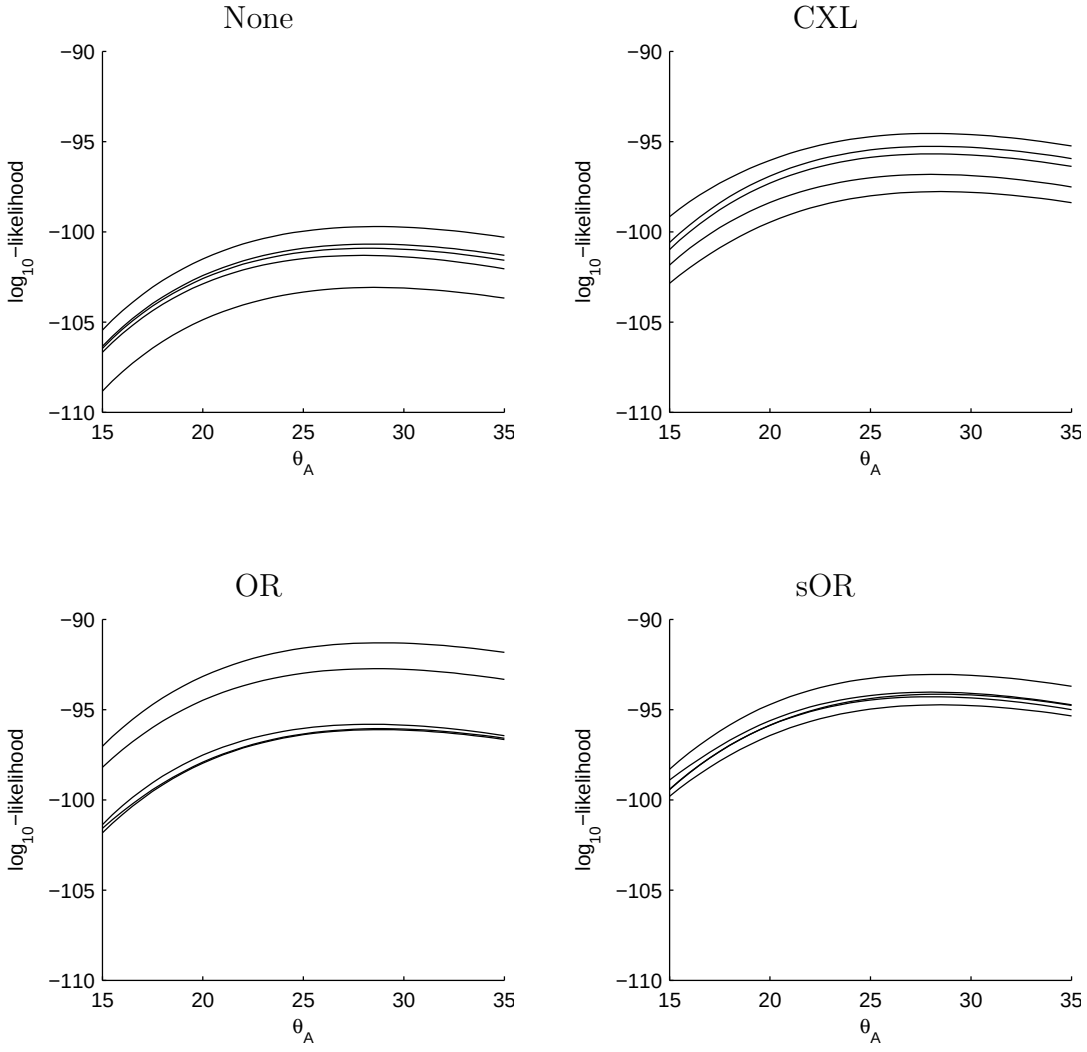


Figure 3.16: A similar plot to Figure 3.14, but for a dataset drawn from $\texttt{ms}(50, 50, 0.1)$, and using driving values $\theta_0 = 50$, $\rho_0 = 0.1$, $N = 10,000$. Here, $B = 25$. Given the complexity of this dataset, an estimate of the true curve is unavailable.

The dataset on which Figure 3.14 is based is exceedingly simple; each locus had only one segregating site. Hence only 100 runs still provided reasonable estimates of the likelihood curve (*top left*). Because of the low mutation rate and the failure of CXL and OR to account for the mutation event's large effect on the current SIS weight, it is clear that these schemes provide *less* accurate likelihood surfaces. Conversely, resampling according to sOR brings each of the five curves very close to the truth. It is encouraging that for low mutation rates the difference between OR and sOR becomes apparent, and that owing to the way sOR is designed it offers substantial improvement.

Similar inferences can be made from Figure 3.15, although now the exchange rates μ_A and μ_B are ten times smaller, the difference between OR and sOR is diminished. Indeed, at these parameter values both OR and sOR offer similar improvement, while CXL offers no reduction in variation about the true curve compared to doing no resampling at all.

The dataset on which Figure 3.16 is based is much more complicated. An attempt to estimate the true curve from $10,000,000$ runs took $\sim$ 1 week on a 3 GHz PC, and the resulting likelihood estimate at the driving value yielded an ESS of only 2! Clearly then, this dataset falls into the territory in which resampling becomes essential. We can judge the performance of the various stopping schemes only on their self-reported accuracy. Recall that, for a proposal distribution which has an insufficient number of runs to explore the sample space, its inaccuracy is likely to be an *underestimate* (Section 2.3.4.2). This appears to be the case for the curve with no resampling. Visually, all three of CXL, OR, and sOR seem to remedy this probable underestimate. That all three schemes might be an improvement for very complicated datasets was also touched upon above. But only sOR significantly

reduces the variation *between* the five curves, implying that its estimated likelihood curves might be more accurate. This is substantiated by an average ESS of 5672; corresponding values for CXL and OR are 168 and 3184 respectively. sOR is thus demonstrated to estimate the likelihood more accurately both for relatively small and relatively large values of θ, though the latter of these conclusions should be accepted tentatively and with the caveats discussed in Section 3.2.4.2.

3.2.5 Discussion

Resampling offers great potential to rejuvenate a sample of SIS runs propagated in parallel, and stopping-time resampling in particular offers a more natural timescale on which resampling can take place. For 'unconstrained' mutation models such as a finite-alleles model, simply taking the number of coalescence events as a yardstick works well [76]. But as I have shown here, ignoring mutation events under more rigid mutation models like the infinite-sites model can be counter-productive. In the previous sections I have proposed a new stopping scheme, called scaled-OR, and verified its improved performance under a variety of parameter values. Moreover, the recasting of the problem in terms of metric spaces aids intuition when it comes to designing other stopping schemes. For example, although I have not attempted to do so, it is simple to consider defining a metric to incorporate recombination events. We should not be interested in every recombination event—merely those that bring a partially reconstructed genealogy closer to the MRCA. A simple measure for this is the *minimum* number of recombination events required to explain the current dataset. Recently, efficient algorithms to solve this problem have been developed [81, 82, 83], and incorporating them into an appropriate metric is a possibility for

fruitful future research. It is unclear whether these algorithms are efficient enough to invoke at every step of every genealogy, and so simpler measures, based on a lower bound for the minimum number of recombination events [17] or on some other non-parametric summary, could prove more useful. Moreover, a similar approach might also be used to incorporate mutation information under different mutation models. Consider Figure 3.2: a similar situation is easily constructed under a stepwise mutation model. To use an example cited by Stephens & Donnelly (2000) [37], let the initial dataset consist of the collection of alleles $H_0 = \{5, 5, 5, 5, 5, 5, 11\}$. Naïvely using the stopping times of CXL, two possible intermediate datasets at the first stopping-time are $H^{(1)}_{-T_1} = \{5, 5, 5, 5, 5, 6\}$ and $H^{(2)}_{-T_1} = \{5, 5, 5, 5, 5, 16\}$. Under a Griffiths-Tavaré proposal distribution (1.10) with symmetric stepwise mutations, each of these intermediate datasets is equally likely to be drawn from q (assuming only the gene with initial allele 11 has undergone an overall mutation upon reaching T_1). Yet it is clear just by looking at the mutation configuration that $H^{(1)}_{-T_1}$ is much closer to the MRCA. The improved importance sampler for stepwise mutation models of De Iorio *et al.* (2005) [23] builds this observation into the proposal distribution, but the point is that it should be built into the stopping scheme too. Taking inspiration from the suggestion for recombination events above, a simple measure might be the minimum number of *mutation* events required to explain the data. Under the stepwise model, this is $a_{(n)} - a_{(1)}$, where $a_{(n)}, a_{(1)} \in \mathbb{Z}$ are the largest and smallest allelic types in the sample respectively, and we assume the allele of the MRCA is unknown. Extensions to other models of mutation are also possible.

3.3 An adaptation of pilot-exploration resampling

3.3.1 Introduction

Historically, an important motivation for the development of sequential importance sampling techniques has been to tackle the chain growth problem, including the very first applications of SIS [84, 85]; enrichment methods [86], which serve as an alternative to resampling; and future scanning [87]. In this section, I propose a way to adopt a recent development in the latter of these—the future scanning method of Zhang & Liu (2002) [78], known as *pilot-exploration resampling* (PER)—to importance sampling on coalescent histories, including the G_{81} model of interest. Zhang & Liu (2002) actually implement two potentially independent lookahead procedures: what I shall denote the *exhaustive* lookahead of Meirovitch (1982) [87] and a *stochastic* lookahead, which is new. They implement the former both at each step of the SIS procedure and at the resampling step, while the latter is implemented only at the resampling step. Their motivation was in the optimization problem of finding polymer conformations of ground energy, but it can be adopted to Monte Carlo integration straightforwardly.

There are two important motivations for looking at PER. First, in the previous sections, one reason for developing sophisticated stopping times was as an indirect way to estimate the expected final weight of each stream. Here, we estimate the expected future weight of each stream directly. Second, a more obvious way to improve a Monte Carlo estimate is simply to increase the sample size N, but when resampling is employed this entails running N streams in parallel. The computational burden is then not merely a linear increase in running time, but an additional burden on memory of maintaining N streams simultaneously. During the investiga-

tion of stopping-time resampling in Section 3.2 I found that, in my implementation, increasing N beyond $50,000$ exhausted the 512 MB RAM of a desktop PC—any further would entail reading and writing to files. To find a way of further improving accuracy without significantly increasing memory requirements would be an important benefit, and PER fulfils that role. (It should be noted that a machine with 1 GB RAM easily handled $N = 250,000$.)

The lookahead procedures function as follows. In the exhaustive lookahead, the key observation of Meirovitch (1982) [87] was that the optimal proposal distribution (which for simplicity I translate into coalescent notation) can be written as

$$\begin{aligned} q^*(H_{k-1} | H_k) &= \frac{\sum_{\{\mathcal{H}' : H'_{k-1} = H_{k-1}\}} p(\mathcal{H}' | \mathcal{D})}{\sum_{\{\mathcal{H}'\}} p(\mathcal{H}' | \mathcal{D})} \\ &= \frac{\sum_{\{\mathcal{H}' : H'_{k-1} = H_{k-1}\}} p(H'_{-m}) \prod_{j=k}^{-m+1} p(H'_j | H'_{j-1})}{\sum_{\{\mathcal{H}'\}} p(H'_{-m}) \prod_{j=k}^{-m+1} p(H'_j | H'_{j-1})}, \end{aligned} \tag{3.4}$$

where $\mathcal{H}' = (H_k, H'_{k-1}, \ldots, H'_{-m})$ is a dummy variable summing over all histories compatible with the current configuration H_k. In other words, partition all histories compatible with the data by the set of next configurations $\{H_{k-1}\}$ and select the next configuration H_{k-1} with probability proportional to the sum of all posterior probabilities of histories with H_{k-1} as their next configuration. Of course, evaluating this sum is computationally prohibitive, but an IS proposal distribution to approximate q^* can be defined by summing not over coalescent histories all the way back to H_{-m}, the MRCA, but by looking only δ steps back. In other words:

$$q(H_{k-1} | H_k) = \frac{\sum_{\{\mathcal{H}' : H'_{k-1} = H_{k-1}\}} \prod_{j=k}^{k+1-\delta} p(H'_j | H'_{j-1})}{\sum_{\{\mathcal{H}'\}} \prod_{j=k}^{k+1-\delta} p(H'_j | H'_{j-1})}. \tag{3.5}$$

The summation is still exhaustive over histories going δ steps back, but can be evaluated for δ sufficiently small. Note that this technique *defines* an IS proposal distribution; it is not one that can be applied to any given proposal distribution. The exhaustive lookahead is however an improvement over the standard proposal distribution used on the chain growth problem, because this corresponds to a simple one-step lookahead. It is clear that incorporating more future information is an improvement on incorporating only the next step—which is the case in the standard setting of $\delta = 1$:

$$q(H_{k-1}|\, H_k) = \frac{p(H_k|\, H_{k-1})}{\sum_{\{H'_{k-1}\}} p(H_k|\, H'_{k-1})}. \tag{3.6}$$

But this is precisely the Griffiths-Tavaré proposal distribution (1.10)! The exhaustive lookahead technique is thus guaranteed to be an improvement only over (3.6). For other proposal distributions, such as that of Stephens & Donnelly (2000) [37], an explicit attempt is made to take an approach more sophisticated than simply maximizing the probability of that part of the history in the vicinity of the current configuration. So, except for very large δ, the exhaustive lookahead procedure will not be an improvement. Since my proposal distribution is based on this more sophisticated approach, the exhaustive lookahead is thus not applicable. An example of combining a two-step lookahead with the Griffiths-Tavaré proposal distribution is in Munday [43].

A situation in which a lookahead procedure *is* applicable is at the resampling step, and this is the setting for the (stochastic) pilot-exploration resampling of Zhang & Liu (2002) [78]. It also deals with the rapidly increasing computational burden with increasing lookahead parameter by sending out a pilot exploration "team" to spy on future information, rather than to evaluate it exhaustively. The method

from their paper proceeds as follows. Residues are added according to some proposal distribution using (an exhaustive) δ-step lookahead. At the resampling step, suppose a set $S_t = \left\{ \boldsymbol{x}_t^{(j)} : j = 1, \ldots, N \right\}$ of partial conformations has been constructed. PER then consists of the following:

1. For each partial conformation $\boldsymbol{x}_t^{(j)} \in S_t$, a team of m "members" are sent out to explore Δ steps ahead. More precisely, we build on $\boldsymbol{x}_t^{(j)}$ the next Δ residues m independent times by SIS to get $\left\{ (x_{t+1}^{(j)l}, \ldots, x_{t+\Delta}^{(j)l}) : l = 1, \ldots, m \right\}$. An exhaustive ϱ-step lookahead can be applied in the generation of these pilot paths.

2. For each generated pilot path l of conformation $\boldsymbol{x}_t^{(j)}$, compute its Boltzmann weight $b_t^{(j)l} = \pi_{t+\Delta+\varrho-1}(\boldsymbol{x}_t^{(j)}, x_{t+1}^{(j)l}, \ldots, x_{t+\Delta}^{(j)l})$.

3. The (unnormalized) resampling probability $a_t^{(j)}$ for conformation $\boldsymbol{x}_t^{(j)}$ is calculated as the αth power of the mean of $b_t^{(j)l}$, $l = 1, \ldots, m$.

4. A resampling step for the set S_t is performed with the probability vector proportional to $\left\{ a_t^{(1)}, \ldots, a_t^{(N)} \right\}$.

Resampling is conducted after every λ residues. The purpose of SIS in their paper is to generate samples of polymers of length d drawn from the Boltzmann distribution $\pi_d(\boldsymbol{x}_d)$, with an energy function defined by pairwise interactions between different types of non-covalently bonded residues (see [78] for details). An approximation for the marginal distribution $\pi_d(\boldsymbol{x}_t)$ of a partial conformation $\boldsymbol{x}_t$ is to use $\pi_t(\boldsymbol{x}_t)$, the Boltzmann distribution of a polymer of length t, and these define the numerator in the SIS weights.

3.3.2 Pilot-exploration resampling on a two-locus, infinite-sites coalescent model

The algorithm described above has eight user-set parameters: τ (a temperature parameter in the Boltzmann distribution), N, δ, λ, m, Δ, ϱ, and α. For simplicity, I will consider only $\alpha = \frac{1}{2}$ as in [70, 78], to avoid placing too much emphasis on estimated future information. δ and ϱ are inapplicable, as discussed above, and we employ dynamic resampling rather than choose a value for λ. In the coalescent framework, the eight parameters therefore map to $(\theta_A, \theta_B, \rho)$, N, B, m and Δ; only two of which are new—team size m and exploration distance Δ.

To implement the method on the G_{81} model, I introduce two further modifications. Firstly, the purpose of pilot-exploration is to obtain a Monte Carlo estimate of the expected SIS weight accumulated a certain number of steps in the future. In a coalescent framework, Δ steps might mean different progression for different streams—i.e. a more natural timescale is required as a measure of progress. This problem is identical to that discussed in Section 3.2, and so in addition to performing resampling using the idea of *stopping times* introduced in that section, one can also define Δ in units of stopping times. Thus, a history $\mathcal{H}^{(j)}_{-T_k}$ that has reached stopping-time T_k will have pilot exploration search as far ahead as $\mathcal{H}^{(j)}_{-T_{k+\Delta}}$ (rather than $\mathcal{H}^{(j)}_{-(T_k+\Delta)}$). Secondly, PER can be viewed as a heuristic method for spying ahead and obtaining future information for use in the resampling procedure. Its main advantage should be the gain in efficiency, but if it uses the same proposal distribution as the main IS algorithm then its exploration will be no less cumbersome. I will investigate a quick simplification that is intended to improve the speed of exploration while still uncovering the important features of the stream's future

SIS weight, as follows: given a current configuration $H^{(j)}_{-T_k}$ that is equivalently represented as two gene trees in the two-locus model, perform IS independently on each gene tree using the one-locus proposal distribution (equal to that of `genetree`), and estimate the expected future SIS weight by the product of the mean of each of these weights. This is a composite likelihood estimate $L_C(0,\theta)$ of equation (1.16), with ρ assumed to be 0 within loci for this model. Depending on the parameters of the model, this estimate should still provide a reasonable indication of future trends in the weight. For example, for large ρ the histories of the two gene trees will be almost independent anyway.

The SISPER procedure for two-locus, coalescent histories at stopping-time T_k with $cv^2 > B$ is then as follows:

1. For each partial history $\mathcal{H}^{(j)}_{-T_k}$ currently weighted by $w^{(j)}_{0:k}$, a team of m members are sent out to explore Δ stopping times ahead. More precisely, we build on $\mathcal{H}^{(j)}_{-T_k}$ m independent times by SIS to get $\left\{(H^{(j)l}_{-(T_k+1)},\ldots,H^{(j)l}_{-(T_{k+\Delta})}) : l = 1,\ldots,m\right\}$.

2. For each generated pilot path l of $\mathcal{H}^{(j)}_{-T_k}$, compute its SIS weight $w^{(j)l}_{k:k+\Delta}$.

3. The (unnormalized) resampling probability $a^{(j)}_{-T_k}$ for $\mathcal{H}^{(j)}_{-T}$ is calculated as

$$a^{(j)}_{-T_k} = w^{(j)}_{0:k}\sqrt{\frac{1}{m}\sum_{l=1}^{m} w^{(j)l}_{k:k+\Delta}}.$$

4. A resampling step on $\left\{\mathcal{H}^{(1)}_{-T_k},\ldots,\mathcal{H}^{(N)}_{-T_k}\right\}$ is performed with the probability vector proportional to $\left\{a^{(1)}_{-T_k},\ldots,a^{(N)}_{-T_k}\right\}$.

If the composite approximation of independent loci is used, the SIS of pilot explo-

ration is performed at each locus independently, and the 3rd step in this procedure is replaced with

3b. The (unnormalized) resampling probability $a^{(j)}_{-T_k}$ for $\mathcal{H}^{(j)}_{-T_k}$ is calculated as

$$a^{(j)}_{-T_k} = w^{(j)}_{0:k} \sqrt{\frac{1}{m}\sum_{l=1}^{m} w^{A(j)l}_{k:k+\Delta}} \sqrt{\frac{1}{m}\sum_{l=1}^{m} w^{B(j)l}_{k:k+\Delta}},$$

where $w^{A(j)l}_{k:k+\Delta}$, $w^{B(j)l}_{k:k+\Delta}$ are the SIS weights accrued by the lth explorer of stream j, between stopping times T_k and $T_{k+\Delta}$, for gene trees at locus A and B respectively. I shall refer to these two algorithms as *SISPER* and *composite SISPER*.

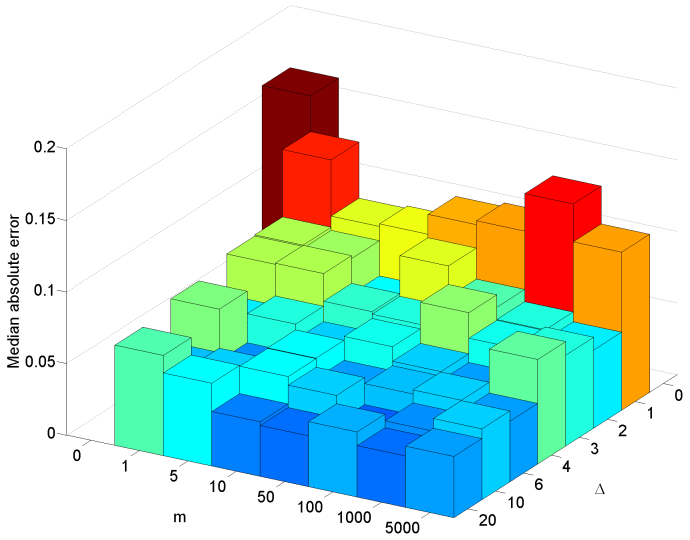
3.3.3 Results

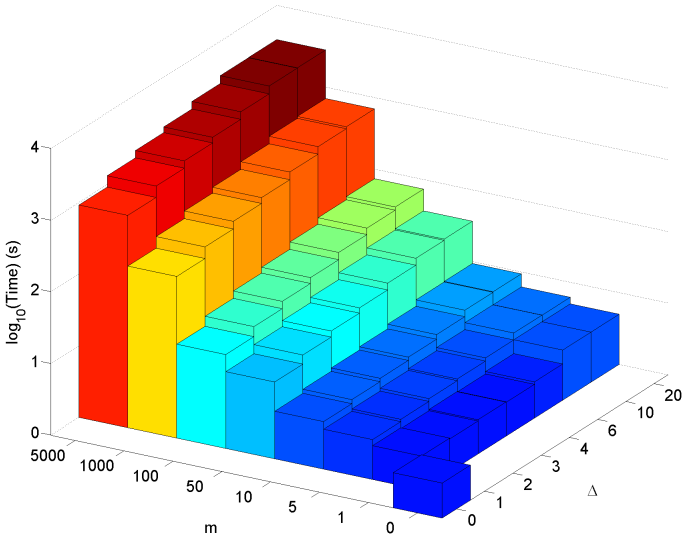
3.3.3.1 Gauging the magnitude of improvement

To estimate the scale of the improvement from the SISPER procedure described above, I performed the following experiment, for a dataset drawn from `ms`$(20, 5, 1)$. For each $m \in \{0, 1, 5, 10, 50, 100, 1000, 5000\}$ and each $\Delta \in \{0, 1, 2, 3, 4, 6, 10, 20\}$, run the importance sampler for 10,000 runs driving at the true parameter values. Resample at stopping times defined by sOR with $B = 4$. The dataset is sufficiently simple that an estimate of the 'true' likelihood is available from $10,000,000$ runs with $B = \infty$, and the performance of each experiment can be measured by the absolute relative error with respect to this true value. This experiment was repeated 25 times independently for each (m, Δ) pair. Results are presented in Figure 3.17. Shortly, we will be concerned with the trade-off between the time spent exploring ahead and the time simply increasing N, so the running-time for each experiment is also shown. Of course, running times are only partly determined by the algorithm itself; there

are also factors dependent on the hardware and software used, and this author's programming design. This should be kept in mind, but the given running times can still provide a general comparison for the relative computational burden of different choices of parameters.

Figure 3.17 gives us a good indication of how to proceed, with one caveat. That the changes in absolute error do not change monotonically along either axis indicates that, even when each bar is based upon the median of 25 experiments, there is still a considerable amount of noise. Nevertheless, clear trends are visible. For SISPER (*top left*), increasing either m or Δ improves the accuracy of the likelihood estimate, as one would expect. Even for $(m, \Delta) = (1, 1)$ there is a clear improvement. It is also evident that increasing m alone or Δ alone—but not both together—provides little or no further improvement. Again, this is as we might expect. Large Δ and small m corresponds to sending only a handful of explorers a long way ahead, which would provide a picture of the evolution of each partially reconstructed genealogy far into the future, but a very dim picture—so dim as to be almost entirely useless. Similarly, large m and small Δ, which corresponds to sending a large team only a few steps forward, provides a very clear picture—but this time only of the immediate future of each partially reconstructed genealogy. The graph suggests that both strategies perform badly, whereas the most rapid improvement in accuracy follows from increasing m and Δ *together*. This agrees with a rule of thumb suggested by Zhang & Liu (2002) [78] in the case of polymer growth: that m and Δ should be of comparable magnitude. In the coalescent setting, with a dataset of this size and Δ measured in units of sOR stopping times, an efficient choice appears to be $m = 50$, $\Delta = 4$. As is clear from the figure, if either parameter is increased further then we encounter rapidly diminishing returns. Running-times increase very quickly





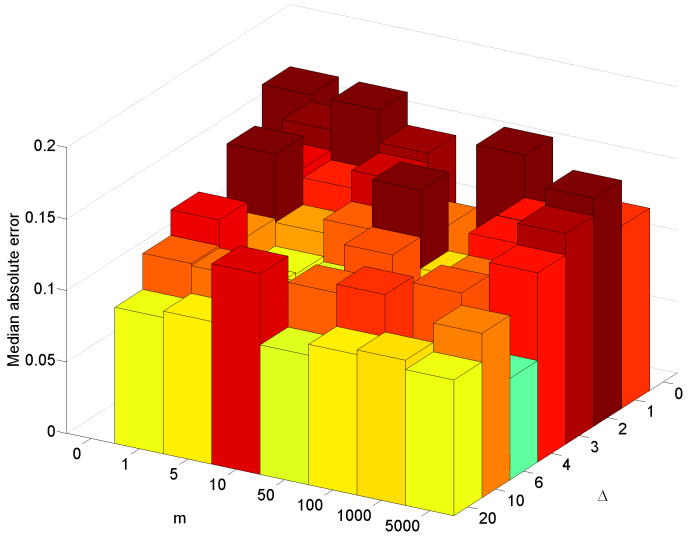


Figure 3.17: (*Left*): The absolute error of the importance sampler run on a dataset drawn from $\texttt{ms}(20, 5, 1)$, driving at the true parameter values for $10,000$ runs and for various choices of SISPER parameters m and Δ. Absolute error is measured in units of the estimated true likelihood of 2.058×10^{-12}, and each bar is the median of 25 independent experiments. Results are for SISPER (*top left*) and composite SISPER (*bottom left*). Also shown (*top right*) is the mean running time for each set of parameter values under SISPER—note that axes have been reversed here, for visual clarity. An almost identical plot was obtained for composite SISPER (not shown). Bars are colour-coded by height.

(Figure 3.17 (*top right*)) with little pay-off, particularly with increasing m.

When the composite SISPER method is applied, there are two competing effects on running-time. Each stream now requires $2m$ explorers, doubling any fixed overhead associated with initiating an exploration. On the other hand, each explorer now operates on only one locus, simplifying the importance sampler significantly, and making it much quicker to reach the next stopping-time in the absence of recombination events. In the example of Figure 3.17, these effects more-or-less perfectly cancel out, but the resulting accuracies (*bottom left*) are clearly much worse. This implies that omitting to account for the dependencies of the two gene trees on each other reduces the predictive power of the exploration team noticeably, at least for a dataset drawn with $\rho = 1$. For greater ρ, there are savings both from ignoring the greater number of recombination events and from the correlation between the two gene trees actually diminishing, so that the composite likelihood approximation tends towards the truth. I will perform some further investigation into composite SISPER, which may still be of benefit in these cases.

3.3.3.2 The trade-off between number of runs and the amount of pilot exploration

Section 3.3.3.1 raises further questions. Do there exist parameter values for which composite SISPER might perform more efficiently than vanilla SISPER? Is it worth performing either version of SISPER at all, given that the additional computation time could be used simply to increase N? To answer these questions, I performed the following experiment. For each value of $\rho \in \Upsilon$, draw 10 datasets from $\texttt{ms}(20, 5, \rho)$ and perform importance sampling on each, testing four strategies:

(I) Importance sampling with $N = 10,000$ and STR with stopping scheme sOR

and $B = 4$.

(II) As in (I) but also with SISPER employed, with $m = 50$ and $\Delta = 4$.

(III) As in (I) but also with composite SISPER employed, with $m = 50$ and $\Delta = 4$.

(IV) As in (I) but with $N = 50,000$.

Repeat each experiment 25 times independently. Strategy (IV) was included as an alternative method for improving efficiency. I found that, for most parameter values, taking $N = 50,000$ increased the running time by a similar magnitude to employing SISPER, or slightly less. As mentioned above, an important point is that in my implementation $N = 50,000$ represents the limit of how far N can be increased when importance sampling is run in parallel, a consequence of having to store streams simultaneously. So if we wanted a comparison with even larger N, it may not be feasible anyway. A representative selection of results for this experiment are shown in Figure 3.18.

As usual, there is a noticeable amount of noise even across 25 independent experiments and a great deal of variation between datasets. However, clear patterns are still visible, and we can make a number of inferences. For several values of ρ, it is clear that SISPER provides an improvement in accuracy comparable to simply increasing N, for a comparable increase in computing time. Consider for example the plots for $\rho = 2$. The median absolute error for (II) and (IV) are very similar for each dataset, though the running time in the former case lies mostly in the range 1–3$\times$ that of (IV). (III) offers a running time intermediate to that of (II) and (IV) but seems to offer slightly less accuracy than either. As ρ increases, the superiority of (II) over (IV) becomes apparent, but at a cost of growing running time. This

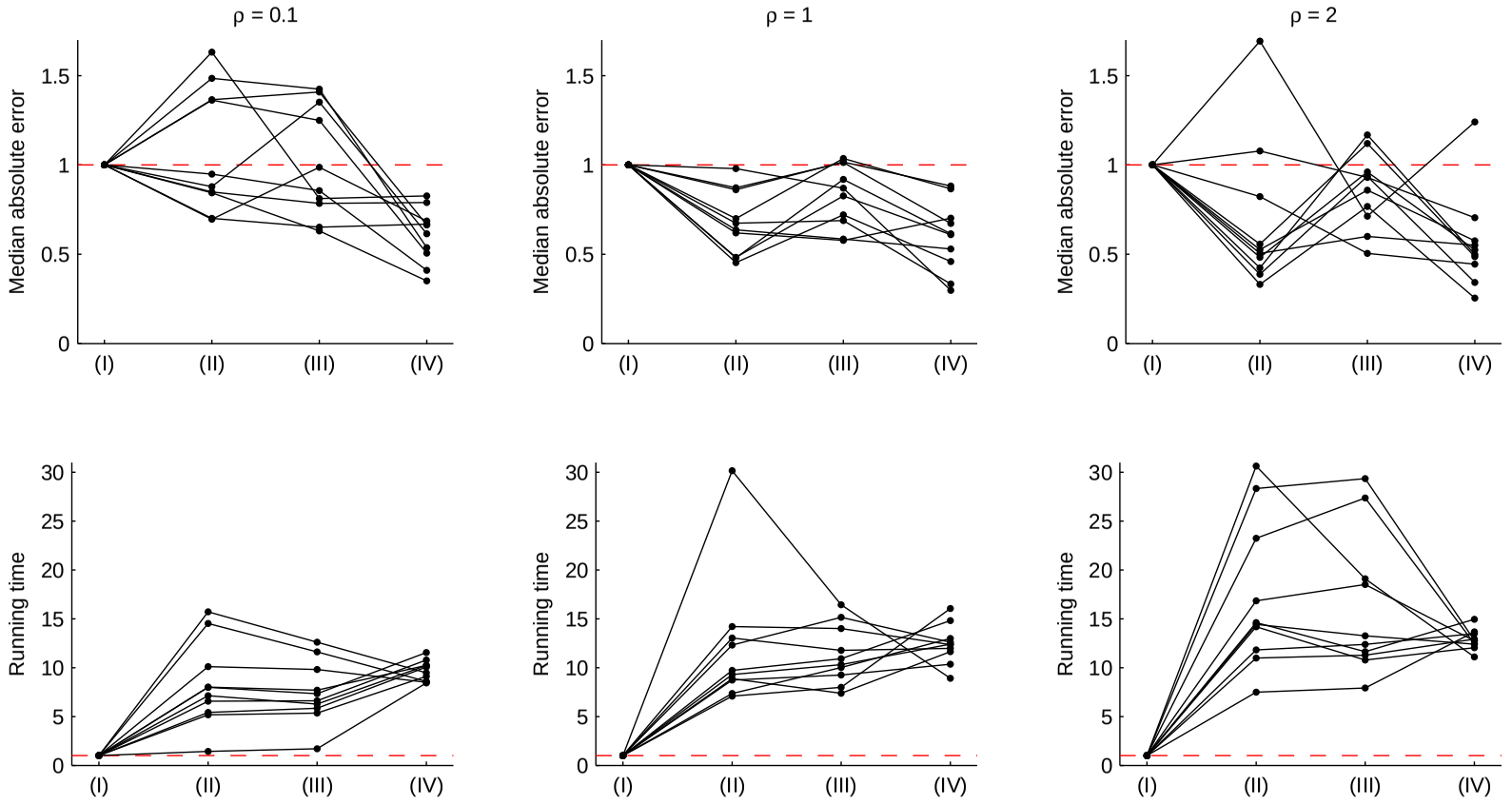


Figure 3.18: (*Top*): The performance of SISPER compared to simply increasing N, for six representative values of $\rho \in \Upsilon$ and 10 datasets drawn from $\texttt{ms}(20, 5, \rho)$. The importance sampler was run for $N = 10,000$, driving at the true parameter values with sOR resampling and $B = 4$; plotted is the absolute error from the 'true' value (estimated by $N = 10,000,000$ and $B = \infty$). Each data point is the median of 25 independent experiments. To enable comparisons between datasets, each point is calibrated by setting the error for STR without PER to 1. Shown are results for (I) STR only, (II) SISPER with $m = 50$ and $\Delta = 4$, (III) composite SISPER with $m = 50$ and $\Delta = 4$, and (IV) STR only with $N = 50,000$. (*Bottom*): Running times, similarly calibrated.

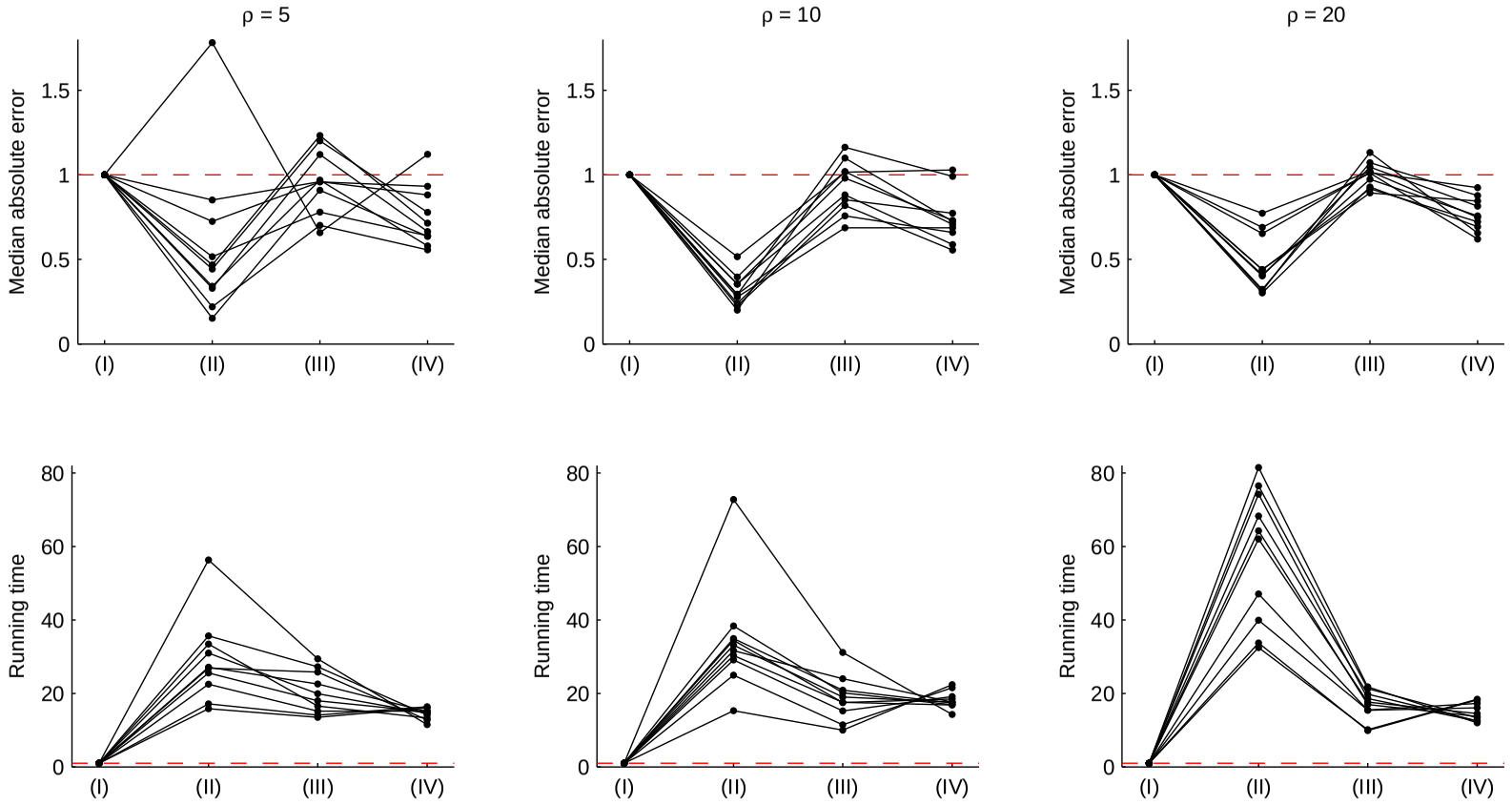


Figure 3.18: (*Continued*).

is most obvious in the plot for $\rho = 20$, in which the trends for all datasets seem to exhibit less absolute error under (II) than under (IV), but the running time for (II) lies in the range 1.5–4× that of (IV). A convenient compromise might be (III), whose improvement in accuracy is generally smaller, but its running times are very similar to (IV). For smaller values of ρ, all of (II), (III), and (IV) generally offer very similar performances and running times. Curiously, for very small ρ (as in $\rho = 0.1$ in Figure 3.18), for several datasets SISPER offers almost no improvement at all.

In summary, then, we may make the following tentative conclusions, based on the parameter values on which I have focused:

1. For small values of ρ, neither SISPER nor composite SISPER offer significant improvement, and a much more reliable way to utilize any additional computing resources is simply to increase N.

2. For larger values of ρ, both SISPER and composite SISPER can offer a similar improvement to increasing N—or at worst, not much less improvement—which provides us with considerable reassurance with respect to choosing a strategy to employ; if we have 'mis-specified' whether SISPER is applicable then the resulting loss in accuracy is not too dramatic.

3. The previous observation may be moot in some situations, since there exists a limit to how much N can be increased. In our example, increasing N to $50,000$ introduced $40,000$ new simultaneous streams, whereas SISPER introduced only 50. In this circumstance, SISPER and composite SISPER are suitable alternatives.

4. For large values of ρ, the running time of SISPER continues to grow whereas

that of composite SISPER does not, and so the latter usurps the former as the most efficient strategy.

3.3.4 Discussion

While of interest in its own right, SISPER offers a neat solution to the problem of having to run a large number of streams in parallel in sequential Monte Carlo. In this section I have shown how to adapt the SISPER method introduced by Zhang & Liu (2002) [78] to importance sampling on the coalescent, co-opting a strategy designed for an optimization problem into one of improving the estimate of a Monte Carlo integral. I have proposed some modifications to their implementation, designed to optimize it for the specific nature of the coalescent model and to take advantage of the progress made in Section 3.2 regarding stopping-time resampling. I have also investigated a faster *composite* version of SISPER—taking inspiration from recent interest in this approach in the literature [54]—and verified that it shows promise here too.

SISPER is complementary to stopping-time resampling; on its own, the latter is hindered by its need to run a large number of streams in parallel, whereas the former makes efficient use of a small number of extra streams. In terms of storage power alone, Section 3.3.3.2 shows that the latter's $10,050$ streams provides similar improvement to the former's $50,000$. It is this that provides encouragement for the further development of the method. An aspect that might discourage us is its large number of parameters. For reasons of space I have had to focus on data generated under particular values of n, θ and ρ; then choose values for N and B; and then choose values for m and Δ. The total space is large, and there is much scope for

further refinement. There is also scope for refinement of the composite approach to exploration; it is not clear whether other fast approximations to the exploration procedure might be more efficient.

3.4 Discussion

There have been many recent developments in the field of sequential Monte Carlo. In this chapter, I have shown how two such developments—*stopping-time resampling* of Chen *et al.* (2005) [76], and *pilot-exploration resampling* of Zhang & Liu (2002) [78]—may be adapted for use with the coalescent, even in more complicated settings like the G_{81} model. There is still much scope for further improvement, as discussed in Sections 3.2.5 and 3.3.4, and moreover, characterizing stopping times in terms of metrics suggests a useful way to extend them for other models. When applied properly, resampling is a useful tool to combine with importance sampling on the coalescent. It can be viewed as an automated way to track the progress of each stream *empirically*, and in this way it unifies some previous observations: Griffiths & Tavaré (1994*a*) [9] suggested that abandoning streams with an excessive number of mutation events (or alternatively, total number of events) would reduce the simulation time and could be used to approximate the full likelihood. This has also been implemented by Nielsen (1997) [22] in the setting of stepwise mutations. Stephens & Donnelly (2000) [37] suggested using a rejection control technique [70] which is similar in approach to resampling; it also measures the success of a stream by its weight, but instead it *discards* those that drop below some threshold. As the theory behind resampling improves, availing of it could become routine.

Finally, it is worth discussing briefly a lookahead procedure I implemented but

which did not seem to provide any improvement. Its motivation was the exhaustive lookahead discussed in Section 3.3.1. As mentioned, that lookahead is not really appropriate unless the Griffiths-Tavaré proposal distribution is used. But one might consider a different, analytic, 'lookahead' as follows. A disadvantage of any importance sampler that incorporates recombination is that it will waste a great deal of time repeatedly undergoing excursions of recombination and then re-coalescence of the same chunks of genetic material. While these will have a (possibly small [66]) effect on the likelihood, they require a relatively large amount of computing time, particularly for larger recombination rates. It is desirable to avoid dealing with these cycles. Ideally, we would have some way of making (2.23) truly recursive; that is, we never return to a state that has already been visited during this run of the chain. While that goal seems out of reach, we can make a 'first-order' correction towards it by eliminating the need for short excursions of the form $\boldsymbol{n} \mapsto \boldsymbol{n} - \boldsymbol{e}_{ij}^{C} + \boldsymbol{e}_{i}^{A} + \boldsymbol{e}_{j}^{B} \mapsto \boldsymbol{n}$, i.e. a recombination and re-coalescence of the same material, as happens for example after the recombination at position $y = \frac{1}{4}$ in Figure 1.3.

For brevity, rewrite the recursion for the sample configuration $\boldsymbol{n}$ of interest by gathering all but the recombination term, as

$$p(\boldsymbol{n}) = \sum_{(i,j)\in E_A\times E_B} \left(r(\boldsymbol{n}, \boldsymbol{m}_R^{(i,j)}) p(\boldsymbol{m}_R^{(i,j)}) + \sum_{\boldsymbol{m}^{(i,j)}\in\mathcal{A}^{i,j}} r(\boldsymbol{n}, \boldsymbol{m}^{(i,j)}) p(\boldsymbol{m}^{(i,j)}) \right), \quad (3.7)$$

where the $\boldsymbol{m}^{(i,j)}$ are possible previous configurations prior to the most recent event back in time which involved the type (i, j); $\boldsymbol{m}_R^{(i,j)}$ is a recombination of this type, and $\mathcal{A}^{i,j}$ is the set of all other events. $r(\boldsymbol{n}, \boldsymbol{m})$ are the known forward co-efficients (resembling the notation of [11]). So (3.7) simply partitions the recombination event, to be treated separately. Now, noting that $p(\boldsymbol{m}_R^{(i,j)})$ also satisfies (3.7), extract the

complementary re-coalescence term:

$$p(\boldsymbol{n}) = \sum_{(i,j)\in E_A\times E_B} \left(r(\boldsymbol{n}, \boldsymbol{m}_R^{(i,j)}) \left[p^{i,j}(\boldsymbol{m}_R^{(i,j)}) + r(\boldsymbol{m}_R^{(i,j)}, \boldsymbol{n}) p(\boldsymbol{n}) \right] + \sum_{\boldsymbol{m}^{(i,j)}\in\mathcal{A}^{i,j}} r(\boldsymbol{n}, \boldsymbol{m}^{(i,j)}) p(\boldsymbol{m}^{(i,j)}) \right), \tag{3.8}$$

where $p^{k,l}(\boldsymbol{n})$ denotes a sample configuration satisfying a restricted version of (3.7):

$$p^{k,l}(\boldsymbol{n}) = \sum_{(i,j)\in E_A\times E_B} \left(r(\boldsymbol{n}, \boldsymbol{m}_R^{(i,j)}) p(\boldsymbol{m}_R^{(i,j)}) + \sum_{\boldsymbol{m}^{(i,j)}\in\mathcal{B}_{k,l}^{i,j}} r(\boldsymbol{n}, \boldsymbol{m}^{(i,j)}) p(\boldsymbol{m}^{(i,j)}) \right), \tag{3.9}$$

with

$$\mathcal{B}_{k,l}^{i,j} = \begin{cases} \mathcal{A}^{i,j} & \text{if } (k,l) \neq (i,j), \\ \mathcal{A}^{i,j} \setminus \left\{ \boldsymbol{m}_C^{k,l} \right\} & \text{if } (k,l) = (i,j). \end{cases} \tag{3.10}$$

Here, $\boldsymbol{m}_C^{k,l}$ is the configuration complementary to a recombination of (k,l); it is the re-coalescence of the resulting fragments. In other words, in (3.8), $p^{i,j}(\boldsymbol{m}_R^{(i,j)})$ satisfies the same recursion conditional on the next event back in time not being a re-coalescence of the material that just underwent recombination—the term for this event has been dealt with separately.

The point of all this is that (3.8) can now be written in the following form:

$$p(\boldsymbol{n}) = \frac{\displaystyle\sum_{(i,j)\in E_A\times E_B} \left(r(\boldsymbol{n}, \boldsymbol{m}_R^{(i,j)}) p^{i,j}(\boldsymbol{m}_R^{(i,j)}) + \sum_{\boldsymbol{m}^{(i,j)}\in\mathcal{A}^{i,j}} r(\boldsymbol{n}, \boldsymbol{m}^{(i,j)}) p(\boldsymbol{m}^{(i,j)}) \right)}{1 - \displaystyle\sum_{(i,j)\in E_A\times E_B} r(\boldsymbol{n}, \boldsymbol{m}_R^{(i,j)}) r(\boldsymbol{m}_R^{(i,j)}, \boldsymbol{n})}. \tag{3.11}$$

It is now straightforward to apply our IS scheme to this 'wrapped' equation rather

than to (3.7); each forward term is simply adjusted by a factor of

$$(1-\lambda(\boldsymbol{n}))^{-1} := \left(1 - \sum_{(i,j)\in E_A\times E_B} r(\boldsymbol{n}, \boldsymbol{m}_R^{(i,j)}) r(\boldsymbol{m}_R^{(i,j)}, \boldsymbol{n})\right)^{-1},$$

and on the right-hand side the configuration $\boldsymbol{m}_R^{(i,j)}$ now satisfies the slightly modified recursion (3.9) rather than (3.7). These induce two complicating factors, addressed below, but first consider—what does (3.11) represent? Let's call a string of recombinations/re-coalescences of the same genetic material a *first-order excursion*. The probability distribution of ARGs can be thought of as a distribution on ARGs with no first-order excursions, compounded with some geometric random variable X_k at each step H_k. X_k represents the number of first-order excursions observed at each step, and is geometrically distributed with parameter $\lambda(\boldsymbol{n})$ because of the Markov nature of the ARG generation process. The recursion (3.11) effectively integrates over each such X_k; the factor $(1-\lambda(\boldsymbol{n}))^{-1}$ is simply the total expected forward probability accumulated by first-order excursions at each step. Note that since $\lambda(\boldsymbol{n})$ is a summation over all allelic types, we have dealt with first-order excursions occurring to *any* allelic type (and indeed a string of first-order excursions each occurring to a possibly different type).

There are two further complications before this scheme can be made practical:

- We also need to perform importance sampling for $p^{k,l}(\boldsymbol{n})$,
- The existing IS scheme designed for (3.7) may no longer be optimal (nor even efficient).

To deal with the first problem, we can repeat the same 'wrapping' manoeuvre on (3.9). It is straightforward to repeat these steps, and the solution obtained (by

analogy with (3.11)) is:

$$p^{k,l}(\boldsymbol{n}) = \frac{1}{1-\lambda(\boldsymbol{n})}\left[\lambda(\boldsymbol{n})r(\boldsymbol{n},\boldsymbol{m}_C^{k,l})p(\boldsymbol{m}_C^{k,l}) + \sum_{(i,j)\in E_A\times E_B}\left(r(\boldsymbol{n},\boldsymbol{m}_R^{(i,j)})p^{i,j}(\boldsymbol{m}_R^{(i,j)}) + \sum_{\boldsymbol{m}^{(i,j)}\in\mathcal{B}_{k,l}^{i,j}} r(\boldsymbol{n},\boldsymbol{m}^{(i,j)})p(\boldsymbol{m}^{(i,j)})\right)\right]. \quad (3.12)$$

Equation (3.12) is identical to (3.11), except for the additional factor of $\lambda(\boldsymbol{n})$ in the numerator. This represents the fact that, under the equation system for $p^{k,l}(\boldsymbol{n})$, re-coalescence of types $(k,*)$ and $(*,l)$ can occur only conditional on the fact that a further excursion first takes place—in other words, it can occur only conditional on the original excursion from state $\boldsymbol{n}$ being higher than first-order (for which we have not attempted to account).

The second problem seems insurmountable. Without it, the above set of equations defines a recursion on a projection of ARGs in which first-order excursions are accounted for analytically, by simply adjusting the forward co-efficients appropriately and cutting out the relevant pathways to prevent first-order excursions actually appearing in the ARGs generated by the importance sampler. The problem is that by maintaining the same backwards probabilities in the IS proposal distribution, *too many* recombination events will be observed. Only those leading to greater than first-order excursions should be generated. This suggests an obvious way to modify the IS proposal distribution; adjust it such that the probability of a recombination event is $\frac{\rho c}{D} - \lambda(\boldsymbol{n})$ rather than $\frac{\rho c}{D}$. The former is the *a priori* relative rate of recombination events that lead to higher than first-order excursions. My simulation work suggests that on average this scheme leads to slightly worse performance in terms of convergence to the true likelihood, even in simple cases such as that shown in

Figure 2.4—in which case exploration of the Markov chain is no longer conducted optimally. Extensive debugging suggests this is not simply an error in the program. I have discussed the potential of this approach anyway, since it illustrates the direct link between the recursion of interest and the potential for improving the importance sampler. Indeed, this method might be rescued by accounting for higher than merely first-order excursions.

Chapter 4

Ancestral inference on the coalescent with recombination

4.1 Introduction

In Chapter 2 some rather involving algebra eventually arrived us at a practicable and efficient importance sampler for the G_{81} model. A prime historical motivation for the development of similar importance samplers has been for performing ancestral inference conditional on a set of data. For example, they might be used for estimating the age of a particular mutation or of the TMRCA [11, 40, 88, 20]. Allowing for more complicated models introduces additional questions of interest. One example is migration between isolated sub-populations—then one might estimate the distribution among subpopulations for the location in which a mutation arose [20]. Another example is recombination: as well as obvious issues like the distribution of the number and location of recombination events [40], there is the more general problem of how two gene trees are correlated. It is not even clear how

to ask such a question, much less answer it. How should we measure the 'sharing' of ancestry between loci? Wiuf & Hein (1999*b*) [89] gave one previous definition in this context. In this chapter I shall return to this problem, and offer a new way to extend it. Its use will be illustrated on a real dataset, but before we proceed it will be necessary to describe briefly how to obtain and process such data and to estimate the parameters associated with it. In particular, by assuming data conforms to the G_{81} model we may not allow any *incompatibilities* within loci. In this context, an incompatibility is defined by each instance of a failure of the *three gamete test*. If the sub-matrix formed by intersecting any three rows and any two columns of the data's incidence matrix is of the form

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix},$$

then this pattern can be explained only after invoking a parallel mutation, a back mutation or a recombination event between the two sites. Here we have assumed that the ancestral type at each site is 0; in the absence of this assumption, the [0 0] haplotype must also be observed and this is now a *four* gamete test. Observing no incompatibilities in the data is equivalent to successfully imposing a partial order on the columns of the incidence matrix, in the following sense. Let O_i denote the set of row indices for which the ith column is 1. Then there are no incompatibilities iff $\forall i, j$, either $O_i \subseteq O_j$ or $O_j \subseteq O_i$ or $O_i \cap O_j = \emptyset$. The inheritance of alleles at nearby sites is thus strongly correlated; they are said to be in *linkage disequilibrium* (LD). Of course, recombination acts to break down LD, which is therefore an indirect measure of the local rate of recombination. A common measure [7] of LD for two

sites is the squared correlation co-efficient of allele frequencies:

$$r^2 = \frac{(p_{11} - p_1 q_1)^2}{p_0 p_1 q_0 q_1},$$

where p_0, p_1 are the marginal allele frequencies at the first site; q_0, q_1 are the marginal allele frequencies at the second site; and p_{11} is the frequency of the [1 1] haplotype. Plots of LD can be used to check agreement with the G_{81} model.

The rest of this chapter is structured as follows. In the next section I introduce an example dataset, and provide details on the pre-processing required for our IS proposal distribution to be applicable, whence MLEs for θ and ρ can be obtained. In Section 4.3 I perform ancestral inference on this dataset. Section 4.3.1 answers obvious questions such as the ages of MRCAs and the distribution of the number of recombination events. Section 4.3.2 attempts to introduce a new method for visualizing the correlation between the two gene trees, via what I have called a *gene graph*. The aim of the gene graph is to summarize the topological nature of the overlap of the two gene trees, while discarding 'unimportant' features of the ARG, in much the same way that a gene tree can be seen as a summary of a given coalescent history. For a fixed dataset the space of gene graphs is finite, and in Section 4.3.2.1 I offer a way to calculate the size of this space.

4.2 Real data

A natural choice for the analysis of real data is SeattleSNPs [90], a resource for the high quality resequencing of genes underlying inflammatory responses in humans. The density of SNPs, that is, single nucleotides whose types vary in the population,

is high—approximately one per 200 base pairs (bp)—which compares with about one per kilobase (kb) in Phase II of the HapMap [64], for example. Since entire genes are resequenced, potentially any SNP in the sequence of interest and present in the sampled population will be captured, including singletons. Indeed, the SeattleSNPs quality control protocols include a procedure for re-sequencing any putatively singleton SNPs. This goes some way to counteract the negative correlation between allele frequency and genotyping error; one would expect widespread genotyping errors to cause an excess of singleton mutations. The inclusion of very low frequency SNPs is important: they cluster on or near the external branches of a genealogy, and ignoring them would distort the distribution of histories conditional on the data, and hence create a bias in any ancestral inference. Many other genotyping projects are designed with a focus on disease-association studies, for which SNPs with a very low MAF are less informative. For example, the Phase II HapMap [64] deliberately excludes SNPs shown in a previous study [91] to have MAF ≤ 0.05.

The high quality of SeattleSNPs data reduces issues relating to ascertainment bias, but might not completely eliminate them. Inference we make on this data should be correspondingly tentative. Two other potential problems are as follows. First, SeattleSNPs seeks out genes important to inflammatory responses—linked to many disorders such as asthma, coronary artery disease and stroke. *A priori*, coding regions of such genes would be strong candidates for purifying selection, a mechanism omitted from our coalescent model. We might simply have to endure this, but the effects can be partially quantified by examining statistics such as Tajima's D [7]. A second problem is that it is known that recombination hotspots occur preferentially within 50 kb of genes but outside the transcribed domain [58], and therefore a candidate gene containing a central hotspot might not be a representative example.

Note however that the IS scheme we are using is restricted only to syntenic loci—they do not necessarily have to be proximate, although it will be true of the example chosen here.

4.2.1 Pre-processing

The gene I shall use in this chapter is Complement Component 2 (C2), an 18 kb gene residing in the major histocompatibility complex and in association with age-related macular degeneration [92]. For this gene, SeattleSNPs sequenced DNA from a panel of 24 African-American (AA) individuals and 23 European individuals, the latter derived from the Center for the Study of Human Polymorphisms (CEPH). These samples correspond to those used in the study of Hinds *et al.* (2005) [91]. Here, I shall omit the European population which often shows evidence of a population bottleneck, departing from the standard coalescent model—though African-American populations can also show evidence of admixture [93, 64] (see also Hein *et al.* (2005) [7] for an overview of recent human evolution in a coalescent framework). Tajima's D for the remaining sample is -0.99, indicative of some departure from the neutral model but not significantly so (p-value 0.19, in a test which accounts for the presence of recombination [93]. Negative values of Tajima's D reflect an excess of low frequency polymorphisms, which may be explained by population growth or purifying selection.) C2 was also chosen since we have independent evidence of a central region of elevated recombination, surrounded by a very low background rate—namely, the genetic map inferred from the Phase II HapMap data (release 22, build 36) [57, 64]. This source suggests a total map distance of 0.0078 cM across the 21.5 kb sequenced, of which 0.0032 cM (41%) can be localized to a central segment of 2 kb. (Despite

this elevation, by the criterion of the HapMap project [58] it is not strong enough to constitute a hotspot.) The precise location of the breakpoint in the G_{81} model is identified below.

SeattleSNPs is a genotyping resource; it does not infer haplotypes ("*phase*" them) experimentally. To obtain an estimate of the set of haplotypes I ran the software `PHASE` version 2.1 [94, 52] using its modelling option of one hotspot of unknown location, and increasing its default parameters to 5000 runs, burn-in 100 and thinning interval 2, to improve accuracy. `PHASE` implements an MCMC algorithm in a Bayesian framework for sampling from the posterior distribution of haplotypes conditional upon the genotype data, and for obtaining an estimate of the list of haplotypes of maximum probability; haplotypes are unobserved random quantities in this setting.

After removing five insertion/deletion polymorphisms (indels) and two SNPs identified by SeattleSNPs as being non-synonymous, I took the best guess of `PHASE` to be the true set of haplotypes. CEPH individuals were included in the `PHASE` analysis; despite population stratification, it is suggested that their inclusion improves the haplotype inference of AA individuals [52]. Note that the phase of SNPs of very low MAF might be difficult or impossible to infer. SeattleSNPs offers haplotypes taken directly from `PHASE` output with such SNPs removed. Rather than use their suggested haplotypes and distort ancestral inference, as discussed above, I preferred to keep low MAF SNPs and simply take `PHASE`'s best guess anyway (which for singletons could correspond to a random assignment to one of the two haplotypes). A more sophisticated approach might be to use the approximate marginal likelihood of Fearnhead & Donnelly (2002) [59].

To *root* the data—that is, determine the ancestral allele at each site—I performed

a `BLAST` alignment [95] of the SeattleSNPs C2 reference sequence against the reference genome for the chimpanzee (build 2) [96]. The top scoring alignment provided homologous nucleotides for 19,454 bp (90.3% of the sequence)—these were taken to be the ancestral alleles. For nucleotides for which possible homologs could not be inferred, the allele of major frequency was taken to be ancestral. This is based on the observation that, conditional on a site being segregating, the probability that the mutant appears in j of n sequences is

$$\frac{j^{-1}}{\sum_{k=1}^{n-1} k^{-1}},$$

for $j = 1, \ldots, n-1$ [97]. Thus, ignoring all other information in the data, it is more probable that the minor of the two alleles is the mutant.

Before we can perform IS, two tasks remain. We need both to determine a suitable location for the recombination breakpoint and to remove a set of sequences and sites such that the remaining dataset is compatible with the G_{81} model. The latter can be used to guide the former, as explained below. Indeed, I found this to be quite a fast way to locate the breakpoint (whose true location we have independent evidence for, from the HapMap Phase II genetic map). The procedure works as follows. R.C. Griffiths has implemented a greedy algorithm by Lenhard (1997) [98], called `prune`, which removes the sequence or site involved in the most incompatibilities in the data, iterating until none remain. Sequences are weighted by their multiplicity, so that sequences with high multiplicities carry an extra penalty for their removal. Selection among sequences or sites scoring the same number of incompatibilities is at random. Now, it is straightforward to modify `prune` such that recombination events at a single fixed breakpoint position do not count as incompatibilities. So

with each putative breakpoint position one can associate the number of pruning events required to make the data compatible, and one could take the position with the fewest number to be the correct breakpoint for the G_{81} model. Tests of this procedure on a number of genes for which the HapMap genetic map is available suggest that this quick method is quite accurate. Figure 4.1 gives some examples. In each case the position with the fewest number of pruning events coincides with the position with the greatest rate of recombination. One could argue that this pattern is an artefact: we'd expect the centre of a gene to require the fewest number of pruning events simply by virtue of it being at the centre; then the breakpoint covers more pairwise site comparisons. This certainly seems to explain at least some of the pattern: the gene GATA3 exhibits a relatively flat recombination rate, and choosing a breakpoint near the centre still seems slightly to be preferable (Figure 4.1 (*top right*)). For genes with stronger hotspots of recombination, however, there is a clear alignment between the two plots (*bottom*). For gene HPGD, even after the hotspot is located it still requires a large number of pruning events, and so the G_{81} model is probably not suitable.

I took the recombination breakpoint of C2 to be that which minimized the required number of pruning events. Assuming it to be at the midpoint of the two adjacent SNPs, its co-ordinate on the chromosome was $32,011,323$, very close to the peak of recombination from the genetic map. With this assumption at most a further 7 pruning events were needed, reflecting incompatibilities that can be explained only by a back or parallel mutation, or by a recombination elsewhere. This left a compatible dataset of 48 sequences and 43 segregating sites, with 21 at locus A and 22 at locus B.

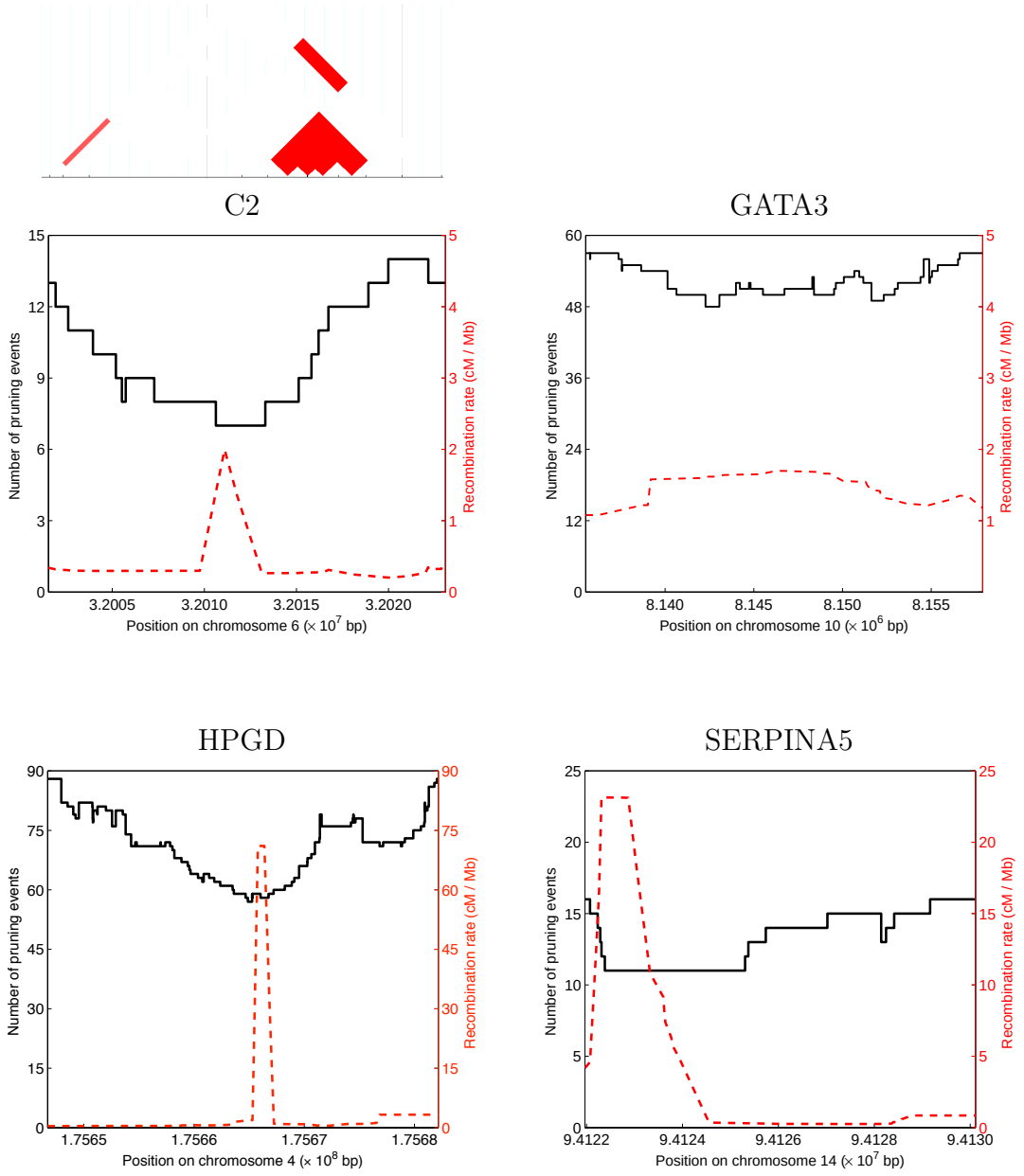


Figure 4.1: Plotted is the number of pruning events required to make the SeattleSNPs data compatible with the G_{81} model, for each putative breakpoint position (solid line). Overlaid is the genetic map derived from Phase II of the HapMap project [64] (dashed line). Also plotted above C2 is a pairwise LD plot of r^2 taken from the Yoruban sample of the Phase II HapMap: white shows no LD, pink moderate LD and red strong LD. It is clear that LD is stronger within at least the right-hand of the two proposed loci, than between them.

θ_{A0}	ρ_0	$\hat{\theta}_A$	$\hat{\rho}$	$L(\hat{\theta}_A, \hat{\rho})$ $(\times 10^{-57})$
5	3	5.7	5.0	0.39
5	4	5.9	6.1	4.34
5	5	5.8	5.4	1.52
5	6	5.9	4.1	0.99
6	3	6.0	3.9	1.11
6	4	5.9	5.2	1.27
6	5	6.0	5.3	1.02
6	6	5.7	6.3	0.82
7	3	5.8	4.5	0.21
7	4	5.9	4.7	0.36
7	5	6.3	4.6	2.45
7	6	5.8	6.3	2.54
All		5.9	5.4	1.29

Table 4.1: MLEs of θ_A $(= \theta_B)$ and ρ using various combinations of driving values, for the SeattleSNPs African-American C2 data under the G_{81} model. Each is based on 25 independent experiments of $N = 60,000$ runs and using sOR resampling with $B = 80$. The 'All' estimate is based on the mean of the weights across all 300 experiments (but is still influenced by the choices of driving value).

4.2.2 Parameter estimates

Given this dataset, one can obtain a joint maximum likelihood estimate $(\hat{\theta}_A, \hat{\theta}_B, \hat{\rho})$ for $(\theta_A, \theta_B, \rho)$. Since the number of segregating sites per kb at each locus were almost identical, henceforth for simplicity I assume that $\theta_A = \theta_B$: the likelihood hypersurface is then reduced to a surface. Trial-and-error simulations suggested that the MLE resides somewhere in $(\hat{\theta}_A, \hat{\rho}) \in [5, 7] \times [3, 6]$. To home in I performed the following experiment. For each pair of driving values $(\theta_{A0}, \rho_0) \in \{5, 6, 7\} \times \{3, 4, 5, 6\}$, obtain an estimate of the likelihood surface from 25 independent experiments with $N = 60,000$ and with sOR resampling $(B = 80)$. Note that it would be preferable to have one large experiment with all streams subject to the same resampling events, but this would require much more memory capacity. Results are shown in Table 4.1.

From the table it is evident that $\hat{\theta}_A$ is extremely robust to the choice of driving value, $\hat{\rho}$ less so. This reflects the fact that the choices made by the proposal distribution depend strongly on ρ_0 but less on θ_{A0}. Each likelihood surface estimate is accurate only in the vicinity of the driving values, and this is more pronounced along the ρ-axis. Away from the driving values it is more probable that the likelihood will be underestimated, so that MLEs are 'dragged' towards the driving value. This seems to be confirmed in Table 4.1: lower driving values for ρ result in a lower estimate $\hat{\rho}$, and higher driving values in a higher estimate. Nevertheless, one can have a reasonable idea of the location of maximum likelihood. To obtain an accurate estimate I ran 300 independent experiments with $N = 60,000$, $B = 80$, and all with driving values $(\theta_{A0}, \theta_{B0}, \rho_0) = (6, 6, 5)$, which is now believed to be close to the MLE. The resulting likelihood surface is presented in Figure 4.2. Construction of the surface required approximately 42 hours on a computer with a 2.4 GHz processor and 1 GB RAM, and by the argument above we can be reasonably confident that the surface is accurate in the vicinity of the MLE. The surface gave $\hat{\theta}_A = 6.07$ and $\hat{\rho} = 4.74$, which I take to be the true values hereafter. The MLE compares with Watterson's moment estimator [1] of $\frac{\tilde{\theta}}{2} = 4.84$, which is obtained by substituting the observed number of segregating sites s into (1.2). The likelihood at the MLE was estimated as 1.46×10^{-57}. That all the likelihood estimates in Table 4.1 are within an order of magnitude of this value is encouraging. As I noted previously, more sophisticated approaches to combining driving values are available—such as bridge sampling [36] and kriging [23]—but I do not pursue them here.

Given a biologically plausible value for r, the probability of a recombination in the gene per generation, we can estimate the effective population size M_e using $\rho = 4M_e r$, and subsequently estimate u, the mutation rate per site per generation,

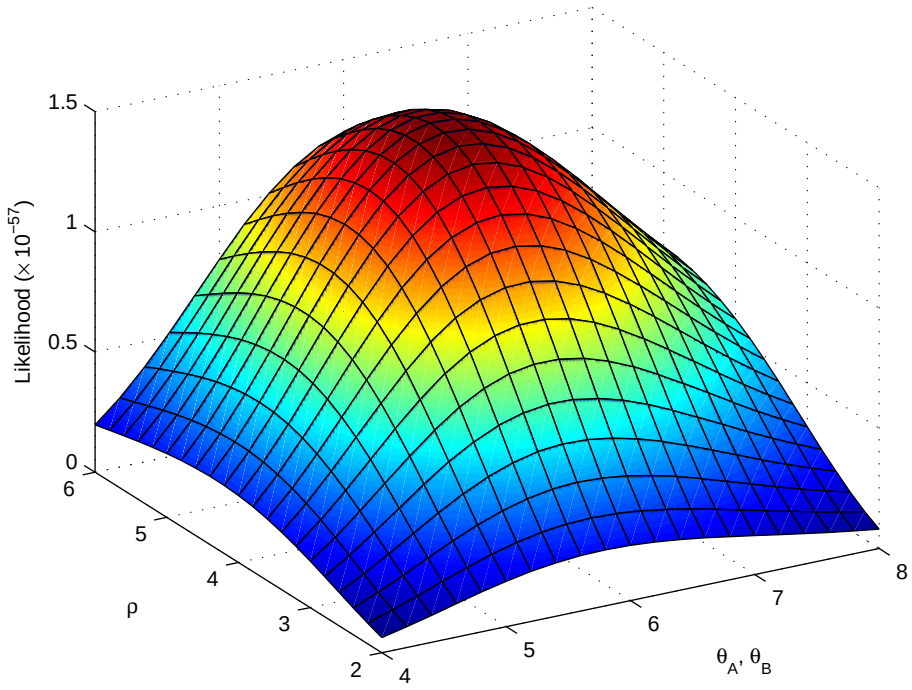


Figure 4.2: Estimated likelihood surface for SeattleSNPs African-American C2 data, assuming a G_{81} model and $\theta_A = \theta_B$, and based on 300 independent IS surface estimates of $N = 60,000$ runs utilizing sOR resampling ($B = 80$). Driving values are $\theta_{A0} = \theta_{B0} = 6$, $\rho_0 = 5$, which are close to the true MLE (see text). The surface is coloured by height.

using $\theta = 4M_e uL$. Here, L is the total length of sequence scanned for variation by SeattleSNPs; for C2 this is $L = 18,626$ [93]. Taking $r = 0.0078$ cM from the Phase II HapMap genetic map for C2 [64], as above, leads to estimates of $\widehat{M_e} = 15,200$ and $\hat{u} = 5.4 \times 10^{-9}$. The former is reassuringly close to the estimate of $15,700$ by Myers *et al.* (2005) [58], who used separate genotype data [91] from the same panel of individuals. The latter is biologically plausible; typical estimates are of the order of 10^{-8} per site per generation [99]. My underestimate could be explained by the fact that some sites were removed in Section 4.2.1, and for simplicity I have ignored the purifying selection invariably acting at some sites.

4.3 Ancestral inference

Given a weighted set of histories, ancestral inference is usually performed via equation (1.8) for various h. For example, if T_{MRCA} denotes the TMRCA then by simulating times between events in each history, the discrete distribution with atoms of mass $\frac{w^{(i)}}{\sum_j w^{(j)}}$ at $(T^{(i)}_{\mathrm{MRCA}}, \mathcal{H}^{(i)})$ is an approximation of $(T_{\mathrm{MRCA}}, \mathcal{H})|\,\mathcal{D}$. Setting $h(\mathcal{H}^{(i)}) = T^{(i)}_{\mathrm{MRCA}}$, equation (1.8) then estimates the mean of this distribution. Other moments can be estimated similarly.

In fact, this approach is needlessly inefficient, since the density $T_{\mathrm{MRCA}}|\,\mathcal{H}$ is known analytically, being the sum of independent exponential distributions (and is thus gamma distributed). A preferable estimate of lower variance is therefore

$$\mathbb{E}[T_{\mathrm{MRCA}}|\,\mathcal{D}] \approx \sum_{i=1}^{N} \left(\frac{w^{(i)}}{\sum_j w^{(j)}} \right) \mathbb{E}[T_{\mathrm{MRCA}}|\,\mathcal{H}^{(i)}]. \tag{4.1}$$

Indeed, the complete density for $T_{\mathrm{MRCA}}|\,\mathcal{D}$ can be estimated in this way, and given the set $\{(\mathcal{H}^{(i)}, w^{(i)}) : i = 1, \ldots, N\}$ this estimate can be written down analytically, being a weighted sum of independent gamma distributions [100]. Even when an analytic solution like this is not available, one can still improve on the original suggestion by appending N' independent estimates (e.g. of times between events) to each history—rather than merely 1—and using

$$\mathbb{E}[h(\mathcal{X})|\,\mathcal{D}] \approx \sum_{i=1}^{N} \left(\frac{w^{(i)}}{\sum_j w^{(j)}} \right) \frac{1}{N'} \sum_{k=1}^{N'} h(\mathcal{H}^{(i,k)}),$$

where $\mathcal{H}^{(i,k)}$ is the ith history using the kth of its associated set of times. Further discussion is given by Stephens (2000) [100].

	A	B	A∨B	A	A∨B
$\mathbb{E}[T_{\mathrm{MRCA}} \mid \mathcal{D}]$	1.04	1.50	1.53	1.96	2.45
	632,000	912,000	930,000	1,191,000	1,490,000
Standard error	0.19	0.27	0.24	0.00	0.57
	116,000	164,000	182,000	0	347,000

Table 4.2: (*Left*): Estimated TMRCAs in coalescent units (*top*) and in years (*bottom*) for each locus of the SeattleSNPs African-American C2 data, and the time until both loci have reached a common ancestor (A∨B). (*Right*): The same estimates for $n = 48$, unconditional on the data. The TMRCA for a single locus is known exactly from coalescent theory, and the estimate for A∨B was based on importance sampling on a monomorphic sample with $\theta_0 = 0$ and $N = 1,000,000$ (though there also exist recursions for calculating this quantity exactly [31]).

4.3.1 General inference

By recording the expected time between the events of each simulated genealogy, one can apply (4.1) in order to estimate the TMRCAs of our example dataset (Table 4.2), and ages of mutations can be obtained similarly. Recall that coalescent units are in terms of $2M_e$ generations, and typically a generation time of 20 years is assumed for humans [88], allowing conversion into years. As the table shows, estimates are of the order of 10^6 years, which for example compares with an estimate by Harding *et al.* (1997) [88] of $\sim 800,000$ years for $n = 326$ sequences of the human β-globin gene.

As discussed above, straightforward modification of (4.1) permits the estimation of other features in the distribution of genealogies, and I conducted one large simulation of $N = 160,000$ with sOR resampling ($B = 150$) for accurate estimates thereof. For example, a question of interest is the probability that the MRCA at locus A and the MRCA at locus B is the same individual. That is, if $T_{\mathrm{A}} := \inf\{t \in \mathbb{R}_+ : a+c = 1\}$ and $T_{\mathrm{B}} := \inf\{t \in \mathbb{R}_+ : b + c = 1\}$ are the TMRCAs for each locus then we wish to

estimate $\mathbb{P}(T_{\mathrm{A}} = T_{\mathrm{B}})$. For our example dataset I estimated $\mathbb{P}(T_{\mathrm{A}} = T_{\mathrm{B}}) = 0.098$. By importance sampling on a monomorphic sample driving at $\theta_0 = 0$ one can compare this with the same probability *unconditional* on the data; using $N = 1,000,000$ I estimated this to be 0.077. Note that it is also possible to find this unconditional probability exactly. One can write down a recursion for $\mathbb{P}(T_{\mathrm{A}} = T_{\mathrm{B}})$ in terms of a, b, c and ρ. If the degree of this recursion is $a+b+2c$ then terms on the right-hand side are of degree less than or equal to $a+b+2c$. Knowing terms of strictly lesser degree, one can find the rest by solving a tridiagonal system of equations. Griffiths (1991) [31] solves a closely related problem in this way—the probability that the MRCA at one locus is ancestral to or is equal to the other, which applies to two-locus ARGs equivalent under $\overset{2}{\sim}$ but not under $\overset{3}{\sim}$ (Figure 2.2).

A second question of inference is the expected number of recombination events, which I estimated to be $\mathbb{E}(R_n|\,\mathcal{D}) = 11.7$, slightly less than the unconditional value of $\mathbb{E}(R_n) = 13.7$. As above, Griffiths (1991) [31] tabulates some exact values for $\mathbb{E}(R_n)$, though for ARGs equivalent under $\overset{2}{\sim}$ rather than under $\overset{3}{\sim}$.

In Chapter 2 I discussed the importance of estimating the distribution of the number of recombination events accurately, and first it is worth a brief digression to revisit the idea here. Figure 4.3 provides an empirical estimate of the distribution $R_n|\,\mathcal{D}$, and we can use this to compare it with the distribution of the number of recombination events generated by the proposal. As is clear from these histograms, an accurate estimate of $R_n|\,\mathcal{D}$ is forthcoming only when resampling is activated, but to evaluate the proposal distribution in isolation we should also consider its performance without resampling. For this dataset the proposal seems to generate slightly too many recombination events, which could be explained as follows. Currently the proposal distribution generates recombination events at the correct relative rate

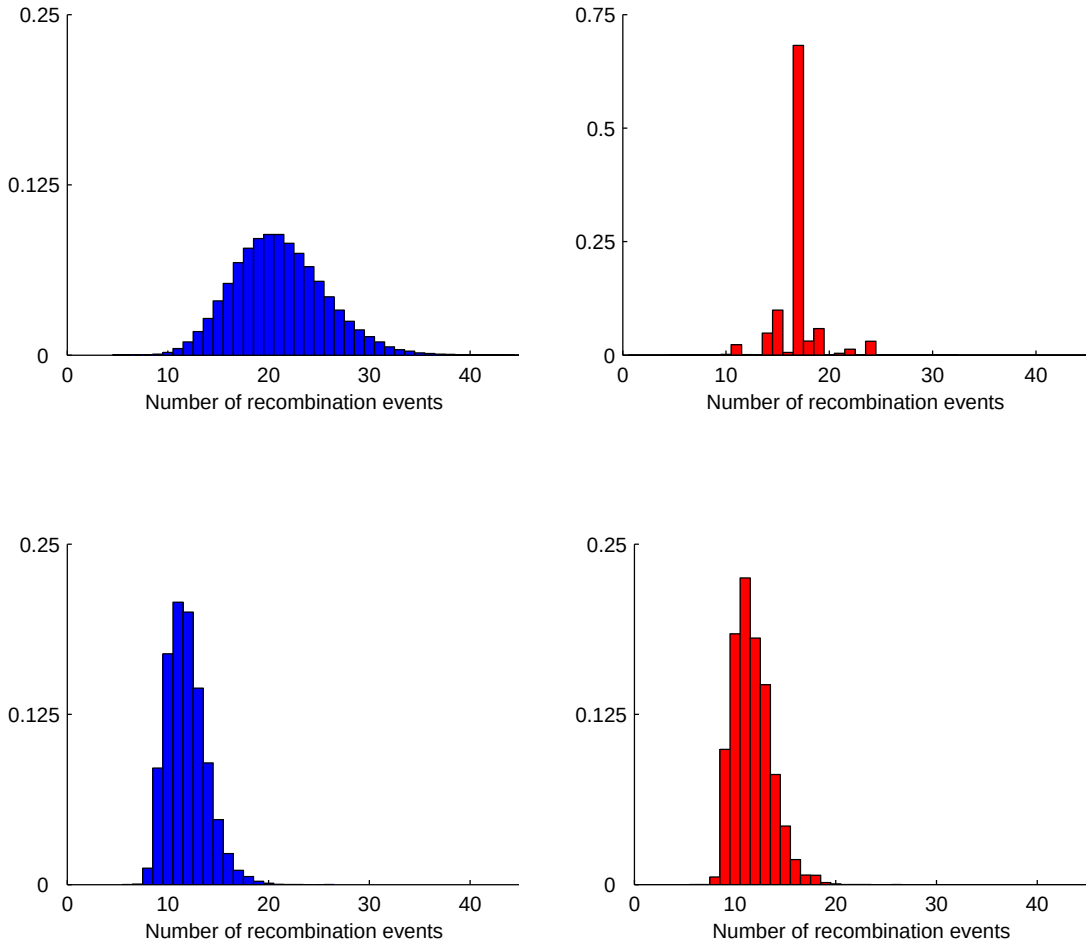


Figure 4.3: Empirical estimates of the probability mass functions for the number of recombination events in a genealogy generated by the proposal distribution (*left*) and for the number of recombination events in a genealogy that gave rise to the data, i.e. $R_n|\,\mathcal{D}$ (*right*). Data is based on the SeattleSNPs African-American panel for the C2 gene. Estimates are based on $N = 160,000$ simulated genealogies without resampling (*top*) and with sOR resampling, $B = 150$ (*bottom*).

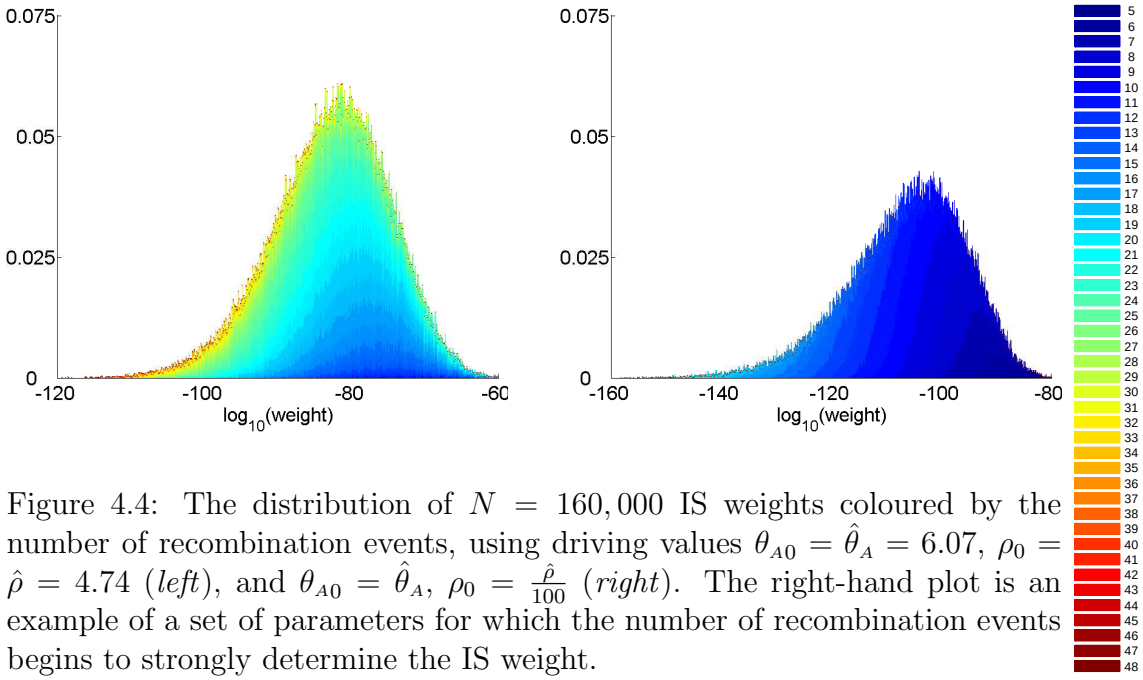


Figure 4.4: The distribution of $N = 160,000$ IS weights coloured by the number of recombination events, using driving values $\theta_{A0} = \hat{\theta}_A = 6.07$, $\rho_0 = \hat{\rho} = 4.74$ (*left*), and $\theta_{A0} = \hat{\theta}_A$, $\rho_0 = \frac{\hat{\rho}}{100}$ (*right*). The right-hand plot is an example of a set of parameters for which the number of recombination events begins to strongly determine the IS weight.

$\frac{\rho c}{D}$ *unconditional* on the data, but it has no control over when or where they are placed on the genealogy. When there are incompatibilities in the data to be overcome, wasted recombination events can be corrected only by proposing yet more recombination events. This recalls the discussion in Section 2.4; we should like to aim recombination events at sequences that are 'ripe' for recombination. Figure 4.4 shows the relationship between the IS weight of each genealogy and the number of its recombination events; here resampling is not used since it would distort any pattern through the correlation of different weights. For this dataset there is a weak positive correlation between the parsimoniousness of a genealogy (in terms of its number of recombination events) and its weight, verifying that if we generate an excessive number of recombinations then we pay a price in the posterior interest in this genealogy. This becomes even more important for smaller values of ρ, for which the number of recombination events begins to determine the magnitude of

the IS weight (*right*). These plots are encouraging for the method of Lyngsø *et al.* (2008) [49], who proposed to approximate the likelihood of the data by excluding ancestral configurations reachable only with an excessive number of recombination events. The intuition supported here is that for small ρ, ARGs with a minimal or near minimal number of recombination events dominate the contribution to the likelihood.

4.3.2 A "gene graph"

It remains to address the question of how the two gene trees fit together. There seems to be scant literature addressing the nature of the 'overlap' of lineages. One exception is Wiuf & Hein (1999*b*) [89], who suggested a definition for the shared ancestry of two sequences. This was proposed to be the total expected time that the ancestral configuration in the ARG is equal to the sample configuration, and in their construction sequences were labelled. One could extend this for more sequences, for example by considering a pairwise composite definition; or relax it to count sharing in lineages individually rather than an 'all-or-none' consideration of entire ancestral configurations. Here however I intend to propose a statistic that gives a more complete overview of precisely which lineages are shared and which are not. To motivate the definition, consider the existing options for summarizing the histories of the two loci. One possibility would be to record the 'most likely' ARG seen during IS. But this is of limited value, first because there is no guarantee that the most likely ARG is 'typical'; and second because any ARG contains a great deal of superfluous events like repeated recombinations and re-coalescences of the same material, in which our summary should not be interested. What we need is some

partition on two-locus ARGs analogous to the partition on coalescent histories that collapses them into gene trees. In this case the gene trees are equivalent to the data. In the two-locus case two marginal gene trees are equivalent to the data but they carry no information on their overlap. It would therefore be of benefit to define some intermediate mathematical object which summarizes the topological information in the ARG and which contains more information than just the data itself. I shall denote this object a "*gene graph*", defined as follows.

Recall the encoding of an infinite-sites dataset at a single locus, which records the age ordered sequence of sites at which mutations have occurred in the line of descent of that gene. A gene is of type $\boldsymbol{x} = (x_0, x_1, \ldots) \in E$ where $E = [0,1]^{\mathbb{Z}_+}$, say. In a two-locus model we can record the mutations occurring in each locus individually, so that a gene is of type $(\boldsymbol{x}, \boldsymbol{y}) \in E_A \times E_B$, where $E_A = [0, \frac{1}{2}]^{\mathbb{Z}_+}$ and $E_B = [\frac{1}{2}, 1]^{\mathbb{Z}_+}$. These provide the path back to the root in each marginal gene tree. Extend this to a gene graph as follows. As one traces the lineage of a locus back through the ARG, there will be times during which it will occupy a gene whose other locus is also ancestral to the sample, and this other locus will also undergo mutations creating new segregating sites. In the path of the locus of interest, *also record these mutations at the other locus.* That is, associate with each gene in the sample a type $(\boldsymbol{x}, \boldsymbol{y}) \in E \times E$, recording the age ordered sequence of mutations encountered on *either* locus in the history of each locus. Thus, genes of identical type in the sample may have different histories in the gene graph.

One might be able to infer more age orderings from a gene graph than from the marginal gene trees alone. The gene graph must comply with the marginal age orderings, and any additional age orderings must not contradict each other. By analogy with the conditions given in Chapter 1 for a gene tree, a gene graph

$(\mathcal{G}_A, \mathcal{G}_B) = ((\boldsymbol{x}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{x}_n, \boldsymbol{y}_n)) \in E^{2n}$ must satisfy the following:

1. $\forall i, j, k,\ x_{ij} = x_{ik} \Longrightarrow j = k$,
2. $\forall i, j, k,\ y_{ij} = y_{ik} \Longrightarrow j = k$,
3. if $i, i' \in \{1, \ldots, n\},\ j, j' \in \mathbb{Z}_+$, then $x_{ij} = x_{i'j'} \Longrightarrow x_{i(j+k)} = x_{i'(j'+k)},\ k = 0, 1, \ldots,$
4. if $i, i' \in \{1, \ldots, n\},\ j, j' \in \mathbb{Z}_+$, then $y_{ij} = y_{i'j'} \Longrightarrow y_{i(j+k)} = y_{i'(j'+k)},\ k = 0, 1, \ldots,$
5. $\exists\ j_1, \ldots, j_n \in \mathbb{Z}_+$ such that $x_{1j_1} = \ldots = x_{nj_n}$,
6. $\exists\ j_1, \ldots, j_n \in \mathbb{Z}_+$ such that $y_{1j_1} = \ldots = y_{nj_n}$,
7. $\forall i, i', j, k, l,\ x_{ij} = y_{i'k},\ x_{il} = y_{i'm},\ j < l \Longrightarrow k < m$.

The first six conditions ensure that each marginal set of paths correspond to a perfect phylogeny. The seventh specifies that, since the two sets of paths share the same set of nodes, any pair of nodes appearing in paths on both 'sides' of the gene graph are age ordered in the same way. An example of an ARG and its corresponding gene graph is given in Figure 4.5. Notice that the paths of locus A induce an age ordering on two of the mutations at locus B, which are not discernible from the marginal gene tree of locus B alone.

Armed with the concept of a gene graph, we are now in a position to investigate the nature of shared sequence ancestry for the C2 data. A visual summary of the 'likely' way the two gene trees entwine is the mode in the weighted distribution of gene graphs obtained from IS, which should represent a subspace of high posterior probability in the space of ARGs. I modified `rita` to record the gene graph from

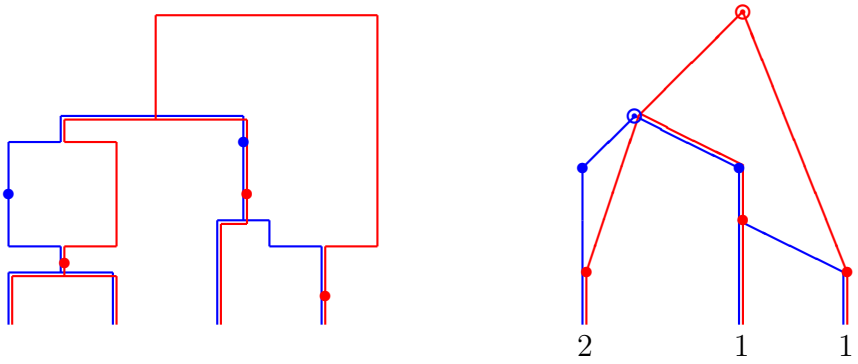


Figure 4.5: An example ARG (*left*) and its corresponding gene graph with leaves labelled by their multiplicity (*right*). Marginal MRCAs are denoted by hollow circles.

each reconstructed ARG, and created a weighted collection of gene graphs from 10 independent experiments of $N = 5,000$, each using sOR resampling ($B = 25$). For convenience gene graphs were leaf-labelled. The $50,000$ simulated ARGs mapped to $14,018$ leaf-labelled gene graphs, and the mode of these accounted for 11.8% of the total weight. Keep in mind however that these statistics probably overestimate the achievable reduction in the space of ARGs, since resampling will have destroyed diversity and greatly reduced the actual number of gene graphs observed. The modal gene graph is presented in Figure 4.6.

It is clear from Figure 4.6 that the gene graph provides a useful visual summary of the nature of the correlation between histories at the two loci. For example, the clade drawn as the left 'half' of the gene tree of locus B seems to share a much greater proportion of ancestry with locus A than the clade composing the right half. A similar pattern was observed for other gene graphs of relatively high posterior probability (not shown). The gene graph also provides a useful point estimate of a set of sequence types likely to have undergone a recombination event. Indeed, it is easy to reconstruct an ARG with the minimal number of recombination events conditional on this gene graph, since necessary recombinations are shown wherever

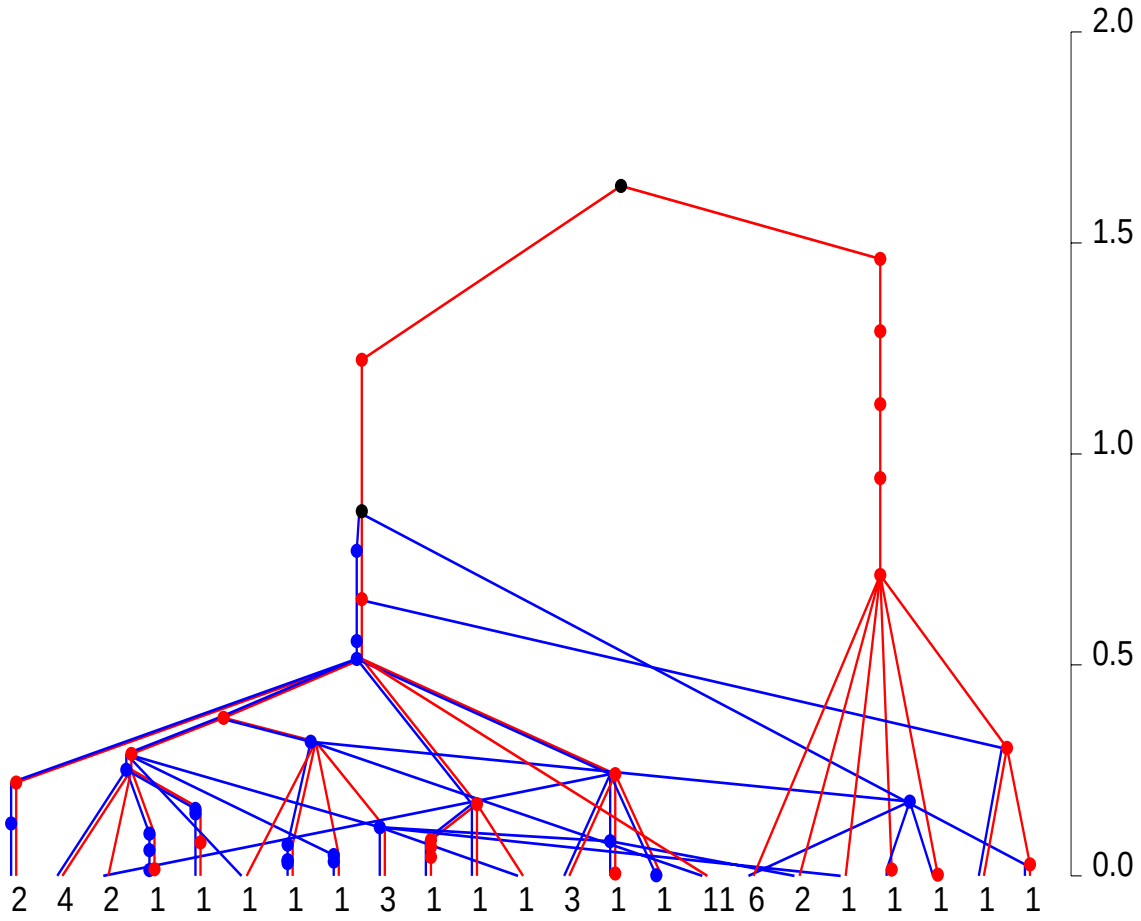


Figure 4.6: An estimate of the gene graph of highest probability given the SeattleSNPs AA panel data for the C2 gene under the G_{81} model. Leaves are labelled by their multiplicities. The axis provides the timescale, in coalescent units. The height of each mutation and of each of the two MRCAs is its expected age conditional on this gene graph.

two previously joined lineages in the gene graph subsequently diverge. Such an ARG can be created by tracing the path of each locus back to its root: coalesce whenever it encounters other branches from either locus, remove mutations when vertices of the gene graph are reached, and recombine only whenever it diverges from its partner lineage. There are 13 such divergences in Figure 4.6. If one could reconstruct *every* gene graph associated with a dataset, a useful consequence is that $R_{\min}(\mathcal{D})$ follows immediately in this way. I turn to the question of enumerating gene graphs in the next subsection.

4.3.2.1 Counting gene graphs

Just like ARGs, one can investigate the distribution of gene graphs. For example, one can write down a recursion for their probability similar to (2.8). Unlike ARGs, the set of gene graphs corresponding to a given dataset is finite, and evaluating the size of this set implies a natural counting problem. In this subsection I develop a method for solving this problem. For convenience I shall work with (leaf-) labelled gene graphs; unlabelled gene graphs could then be enumerated using this result.

We begin with a given dataset in the form of two marginal gene trees and a multiplicity vector $(\mathcal{T}_A, \mathcal{T}_B, \boldsymbol{n})$, and since identical types can have distinct histories in a gene graph let us re-encode this data by treating a type i with multiplicity n_i as n_i repeated types of multiplicity 1. The data is now $(\mathcal{T}_A, \mathcal{T}_B)$ with possibly non-distinct rows, and a suppressed multiplicity vector of 1s. Paths to the root in a corresponding gene graph will be denoted $(\mathcal{G}_A, \mathcal{G}_B)$. Introduce a set of *leaf nodes* $Z_n = \{z_1, \ldots, z_n\}$ which label each sequence, so the paths of mutations to the root are each prefixed by the appropriate sequence label. Denote the collection of sites on each tree by $A_n = \{a_1, \ldots, a_{s_n^A}\}$, $B_n = \{b_1, \ldots, b_{s_n^B}\}$. The MRCA will be represented

either by the set $R_0 = \{0\}$ or by $R_1 = \{0_A, 0_B\}$, according to whether the MRCA of the two trees coincide. Let $M_j := Z_n \cup A_n \cup B_n \cup R_j$, the set of all the nodes in a gene graph. We will introduce order relations $\omega \subseteq M_j \times M_j$ to represent age orderings on these nodes induced by the gene graph. For an order relation ω on a set X, written $<_\omega$, denote the *transitive closure* of ω by ω^*; that is, $y_1 <_{\omega^*} y_2$ iff $\exists$ $\{x_1, \ldots, x_m\} \subseteq X$ (possibly empty) such that $y_1 <_\omega x_1 <_\omega \ldots <_\omega x_m <_\omega y_2$. Denote the age ordering inferred from $\mathcal{T}_A$ by t_A. That is, for $m_1, m_2 \in M_0$, $m_1 <_{t_A} m_2$ iff $\exists i, j$ with $m_1 = x_{ij}$ and $m_2 = x_{i(j+1)}$, where $x_{ij}, x_{i(j+1)} \in \mathcal{T}_A$. The transitive closure t_A^* gives the complete age ordering deducible from $\mathcal{T}_A$, so for $m_1 <_{t_A^*} m_2$ read: "m_1 is younger than m_2 in $\mathcal{T}_A$". Define t_B and t_B^* similarly. The number of gene graphs can be enumerated by counting all possible pairs of age orderings (α, β), where $\alpha, \beta \subseteq M_j \times M_j$, which are compatible with a partial order on M_j and with the age orderings induced by the data $(\mathcal{T}_A, \mathcal{T}_B)$. α and β will encode the corresponding paths for locus A and B of the gene graph respectively, and so $(\alpha \cup \beta)^*$ encodes the age ordering deducible from the complete gene graph.

I now make use of the following observations:

1. Conditional on $\text{MRCA}_A = \text{MRCA}_B$, $\exists$ a bijection between the set of gene graphs consistent with $(\mathcal{T}_A, \mathcal{T}_B)$ and the set of pairs of order relations (α, β) each on M_0 that satisfy:

   (a) $(\alpha \cup \beta)^*$ is asymmetric; there are no two nodes in the gene graph each younger than the other. Note that since $(\alpha \cup \beta)^*$ is transitive, irreflexivity is necessary and sufficient for this to hold.

   (b) $\alpha^*|_{M_0 \setminus B_n} = t_A^*$, and $\beta^*|_{M_0 \setminus A_n} = t_B^*$. That is, the restriction of the age ordering of $\mathcal{G}_A$ to the nodes of $\mathcal{T}_A$ coincides precisely with the age ordering

discernible from $\mathcal{T}_A$ alone. Similarly for locus B.

(c) $\forall a \in A_n \cup B_n$, $\forall b, c \in M_0$, $a <_\alpha b$ and $a <_\alpha c \implies b = c$; any mutation has at most one outgoing edge in $\mathcal{G}_A$. Similarly for $\mathcal{G}_B$.

(d) $\forall a \in A_n \cup B_n$, $\exists b \in M_0$ with $a <_\alpha b \iff \exists c \in M_0$ with $c <_\alpha a$; for $\mathcal{G}_A$, a mutation has an outgoing edge iff it has at least one incoming edge.

(e) $\forall a \in A_n \cup B_n$, $\exists b \in M_0$ with $a <_\beta b \iff \exists c \in M_0$ with $c <_\beta a$; for $\mathcal{G}_B$, a mutation has an outgoing edge iff it has at least one incoming edge.

(f) $\forall z \in Z_n$ $\nexists m \in M_0$ with $m <_{(\alpha\cup\beta)} z$; the leaves of the gene graph have no incoming edges.

(g) $\exists m_1, m_2 \in M_0$ such that $m_1 \neq m_2$, $m_1 <_\alpha 0$, $m_2 <_\alpha 0$; and similarly for β. Additionally, $\nexists m \in M_0$ such that $0 <_{(\alpha\cup\beta)} m$; the root of the gene graph has at least two incoming edges at each locus and has no outgoing edges.

2. Conditional on $\text{MRCA}_A \neq \text{MRCA}_B$, $\exists$ a bijection between the set of gene graphs consistent with $(\mathcal{T}_A, \mathcal{T}_B)$ and the set of pairs of order relations (α, β) each on M_1 that satisfy analogous properties as above—simply replace all references to R_0 with R_1, and replace (1g) with:

   (g) $\exists m_1, m_2 \in M_1$ such that $m_1 \neq m_2$, $m_1 <_\alpha 0_A$, $m_2 <_\alpha 0_A$; and similarly for β and 0_B. Additionally, $\nexists m \in M_1$ such that $0_A <_\alpha m$ and $\nexists m \in M_1$ such that $0_B <_\beta m$; the roots of the gene graph have at least two incoming edges and no outgoing edges within their respective loci.

In arguing for the correctness of these observations, from here-onwards I will deal only with observation 1; observation 2 can be treated similarly. To argue for this

bijection, first consider $\mathcal{G} \mapsto (\alpha, \beta)$. Given a gene graph $\mathcal{G}$, define (α, β) as follows. For each sequence $\boldsymbol{x}_i$ in $\mathcal{G}_A$ and for each pair of consecutive entries $(x_{ij}, x_{i(j+1)})$ put $x_{ij} <_\alpha x_{i(j+1)}$. For each sequence $\boldsymbol{y}_i$ in $\mathcal{G}_B$ and for each pair of consecutive entries $(y_{ij}, y_{i(j+1)})$ put $y_{ij} <_\beta y_{i(j+1)}$. Consider the case when any one of the properties listed above does not hold.

If (1a) fails then there is a mutation which is younger than itself; it is clear that $\mathcal{G}$ cannot have been a gene graph.

Suppose (1b) fails: without loss of generality consider only locus A, and the two cases (i) $\alpha^*|_{M_0 \setminus B_n} \subset t_A^*$, and (ii) $\alpha^*|_{M_0 \setminus B_n} \supset t_A^*$ (a third, trivial, possibility is $\alpha^*|_{M_0 \setminus B_n} \cap t_A^* = \emptyset$). In case (i), $\exists\ a, b$ with $a <_{t_\alpha^*} b$ such that $a \not<_{\alpha^*} b$. So there is a lineage in $\mathcal{T}_A$ containing both mutations a and b, with a younger than b. This is not the case in the corresponding lineage of the gene graph, and so $\mathcal{G}$ is not consistent with $\mathcal{T}_A$. In case (ii), $\exists\ a, b$ with $a <_{\alpha^*|_{M_0 \setminus B_n}} b$ such that $a \not<_{t_\alpha^*} b$. So there is a lineage in $\mathcal{G}_A$ containing both mutations a and b, with a younger than b, and this is not the case in the corresponding lineage of $\mathcal{T}_A$. Thus $\mathcal{G}$ is not consistent with $\mathcal{T}_A$.

It is clear that for any gene graph it must be the case that, at each locus, each leaf has an outgoing edge and no incoming edge; each mutation has at most one outgoing edge and at least one outgoing edge iff it has an incoming edge; and the root has at least two incoming and no outgoing edges. If any of (1c–1g) does not hold then one of these properties is violated and $\mathcal{G}$ cannot have been a gene graph. Therefore a violation of any of the properties (1a–1g) above yields a contradiction, and so the map $\mathcal{G} \mapsto (\alpha, \beta)$ is well-defined.

Next argue for the converse, $(\alpha, \beta) \mapsto \mathcal{G}$. Let (α, β) be a pair of order relations satisfying properties (1a–1g). It remains to show that they correspond uniquely to a gene graph $\mathcal{G}$ consistent with $(\mathcal{T}_A, \mathcal{T}_B)$. Define each $\boldsymbol{x}_i$ in $\mathcal{G}_A$ by forming the sequence

resulting from tracing the path $z_i <_\alpha \ldots <_\alpha 0$, and each $\boldsymbol{y}_i$ in $\mathcal{G}_B$ by forming the sequence resulting from tracing the path $z_i <_\beta \ldots <_\beta 0$. Properties (1c–1g) ensure that these paths are well-defined and unique. Property (1a) ensures that the set of mutations $A_n \cup B_n$ can be strictly partially ordered by age. Property (1b) ensures that the marginal topologies of $\mathcal{G}$ on $Z_n \cup A_n \cup R_0$ and $Z_n \cup B_n \cup R_0$ coincide precisely with the marginal trees $(\mathcal{T}_A, \mathcal{T}_B)$, so that $\mathcal{G}$ is consistent with the data. Thus, we have simplified the problem of counting gene graphs into one of counting subsets of $M_0 \times M_0$ satisfying particular properties.

A counting strategy now suggests itself: simply consider each pair of order relations (α, β) in turn and count those satisfying properties (1a–1g) above. However, this entails the construction of $4^{|M_0|}$ order relations and discarding the vast majority, a vastly inefficient exercise. Progress can be made by constructing each order relation in a systematic way, and abandoning it as soon as any of the properties are violated. To facilitate this, define order relations $\bar{\alpha}, \bar{\beta}$. As we sequentially construct α, β we aim to use these to keep track of those orderings which are prohibited from α and β. In other words, given a partially constructed α we have that $(a, b) \in \bar{\alpha} \implies (a, b) \cup \alpha$ violates one of (1a–1g). It is our aim to insert elements into $\bar{\alpha}$ and $\bar{\beta}$ so that violations of these properties are avoided as early in the construction of α and β as possible. Some of the properties can be prohibited earlier than others. For example, by putting (m, z) into $\bar{\alpha}$ and $\bar{\beta}$ initially, $\forall m \in M_0$ and $\forall z \in Z_n$, property (1f) can always be satisfied. I now describe a recursive method for constructing each (α, β) satisfying these properties.

For convenience suppose the elements of M_0 can be put into order by their labels: $(z_1, \ldots, z_n, a_1, \ldots, a_{s_n^A}, b_1, \ldots, b_{s_n^B}, 0)$. Henceforth, order relations will be constructed row-wise, where α and β are thought of as $|M_0| \times |M_0|$ matrices, and the ith row

refers to the ith element of this list. Denote $m_0 = |M_0|$. Initialize $\bar{\alpha}$ and $\bar{\beta}$ with $\bar{\alpha}_0$, $\bar{\beta}_0$, defined as follows. Begin with $\bar{\alpha}_0 = \emptyset$, and insert the following elements:

- $\{(m, z) : m \in M_0, z \in Z_n\}$. Edges incoming to leaves are prohibited.
- $\{(0, m) : m \in M_0\}$. Edges outgoing from the root are prohibited.
- $\{(m, m) : m \in M_0\}$. Reflexivities in α are prohibited.
- $\forall m_k, m_l \in M_0 \setminus B_n, m_k <_{t_A} m_l \implies (m_k, m) \in \bar{\alpha}_0$, $\forall m \in M_0 \setminus (B_n \cup \{m_l\})$. Lineages inconsistent with $\mathcal{T}_A$ are prohibited.
- $\forall m_k, m_l \in M_0 \setminus A_n, m_k <_{t_B^*} m_l \implies (m_l, m_k) \in \bar{\alpha}_0$. Lineages inconsistent with the age ordering deduced from $\mathcal{T}_B$ are prohibited.

$\bar{\beta}_0$ is defined similarly. Now, given some data $(\mathcal{T}_A, \mathcal{T}_B)$, a partially constructed α and β and some corresponding knowledge $\bar{\alpha}, \bar{\beta}$ about prohibited extensions of these order relations, define a recursive function $f_{i,j}(\alpha, \bar{\alpha}, \beta, \bar{\beta})$ as follows:

$$
f_{i,0}(\alpha, \bar{\alpha}, \beta, \bar{\beta}) \quad = \quad \begin{cases} \sum\limits_{\text{(I)}} f_{i+1,0}\left(\alpha \cup \{(i,k)\}, \bar{\alpha}', \beta, \bar{\beta}\right) \\ \qquad\qquad \text{if } 1 \le i \le n + s_n^A; \\ \sum\limits_{\text{(I)}} f_{i+1,0}\left(\alpha \cup \{(i,k)\}, \bar{\alpha}', \beta, \bar{\beta}\right) + f_{i+1,0}(\alpha, \bar{\alpha}, \beta, \bar{\beta}) \\ \qquad\qquad \text{if } n + s_n^A + 1 \le i \le n + s_n^A + s_n^B; \\ 0 \qquad\qquad \text{if } i = m_0 \textit{ and} \text{ any of the following:} \\ \qquad\qquad\quad \exists j \ge n+1 : \sum_k \alpha_{jk} > 0 \text{ and } \sum_l \alpha_{lj} = 0, \text{ or} \\ \qquad\qquad\quad \alpha^*|_{M_0 \setminus B_n} \ne t_A^*, \text{ or } \sum_l \alpha_{lm_0} \le 1; \\ f_{m_0,1}(\alpha, \bar{\alpha}, \beta, \bar{\beta} \cup (\alpha^*)^t) \\ \qquad\qquad \text{otherwise.} \end{cases}
$$

$$f_{m_0,j}(\alpha,\bar{\alpha},\beta,\bar{\beta}) \;=\; \begin{cases} \displaystyle\sum_{\text{(II)}} f_{m_0,j+1}(\alpha,\bar{\alpha},\beta\cup\{(j,k)\},\bar{\beta}') & \\ \qquad\qquad \text{if } 1\le j\le n \text{ or } n+s_n^A+1\le j\le n+s_n^A+s_n^B; & \\ \displaystyle\sum_{\text{(II)}} f_{m_0,j+1}(\alpha,\bar{\alpha},\beta\cup\{(j,k)\},\bar{\beta}') + f_{m_0,j+1}(\alpha,\bar{\alpha},\beta,\bar{\beta}) & \\ \qquad\qquad \text{if } n+1\le j\le n+s_n^A; & \\ 0 \qquad \text{if } j=m_0 \textit{ and} \text{ any of the following:} & \\ \qquad\qquad \exists i\ge n+1 \text{ with } \textstyle\sum_k \beta_{ik}>0 \text{ and } \sum_l \beta_{li}=0, \text{ or} & \\ \qquad\qquad \beta^*|_{M_0\setminus A_n}\neq t_B^*, \text{ or } \textstyle\sum_l \beta_{lm_0}\le 1; & \\ 1 \qquad \text{otherwise.} & \end{cases}$$

Summation (I) is over all k such that $(i,k)\notin\bar{\alpha}$ and $(\alpha\cup\{(i,k)\})^*|_{M_0\setminus B_n}\subseteq t_A^*$. Summation (II) is over all k such that $(j,k)\notin\bar{\beta}$ and $(\beta\cup\{(j,k)\})^*|_{M_0\setminus A_n}\subseteq t_B^*$ and $(\alpha\cup\beta\cup\{(j,k)\})^*$ is asymmetric. Using this recursion, the number of gene graphs consistent with $(\mathcal{T}_A,\mathcal{T}_B)$ is $f_{1,0}(\emptyset,\bar{\alpha}_0,\emptyset,\bar{\beta}_0)$. In words, the function defined above proceeds as follows. We recursively construct an (α,β) by conditioning on each row i of α and then given an α, conditioning on each row j of β. Test for violations of each of the properties outlined above as soon as possible. So, the definition of $f_{i,0}(\alpha,\bar{\alpha},\beta,\bar{\beta})$ says: suppose we have already considered each row $1,\ldots,i-1$. Proceed by conditioning on each column k in turn for which (i,k) has not already been outlawed via $\bar{\alpha}$. Add (i,k) to the current α, and add all entries to $\bar{\alpha}$ that we can deduce immediately to be outlawed in α—denoted above as $\bar{\alpha}'$. This is the union of the existing $\bar{\alpha}$ with $\{(k,i)\}\cup\{(k,m):(m,i)\in\alpha^*\}\cup\{(m,p):(k,m)\in\alpha^*,(p,i)\in\alpha^*\}$, the set with those entries that would entail $(k,i)\in\alpha^*$, creating an inconsistent age ordering on nodes. We also count only k for which adding (i,k) to α does not violate property (1b). If $n+s_n^A+1\le i\le n+s_n^A+s_n^B$ then we also consider

the case $f_{i+1,0}(\alpha, \bar{\alpha}, \beta, \bar{\beta})$, which corresponds to a mutation on locus B being absent from every path in $\mathcal{G}_A$.

The proposed order relation α is completed when $i = m_0$. The definition $f_{m_0,0}(\alpha, \bar{\alpha}, \beta, \bar{\beta})$ then performs some final tests on α: does there exist a non-leaf node with an outgoing edge but no incoming edges (property 1d)? Is the age ordering of nodes in tree A the same in α as in t_A (property 1b)? Does the root have fewer than two incoming edges (property 1g)? If each of these tests is passed then we repeat the process by setting $j = 1$ and conditioning recursively on each row of β, after adding the elements of $(\alpha^*)^t$ to $\bar{\beta}$ (recall 1a). The conditioning on each row j of β proceeds in a similar fashion.

When (α, β) is finally constructed and passes all the tests, it is counted. $\bar{\alpha}_0$ and $\bar{\beta}_0$ ensure that properties (1f) and (1g) are always satisfied, and the tests described in the definition of f ensure that properties (1a), (1b), (1d), and (1e) are satisfied. Finally, property (1c) is automatically satisfied since at each stage of the construction of α and β the row is incremented, so that each node can have at most one outgoing edge. Thus, (α, β) corresponds to a gene graph. The recursion developed here provides, therefore, both the number of gene graphs consistent with the data and an explicit way to reconstruct each of them.

I implemented this function to examine the growth in the number of gene graphs with the complexity of a dataset, as measured by n and s. An illustration for small datasets is given in Table 4.3, and it is obvious from the table that growth is very rapid. Despite the acceleration in counting gene graphs as outlined above, the procedure still runs in exponential time, and enumerating the gene graphs associated with a dataset as complex as the SeattleSNPs African-American C2 data is out of reach by this method.

	s		
n	0	1	2
2	6	19	92
3	14	61	626
4	30	355	3974
5	62	1383	24566

Table 4.3: The maximum number of leaf-labelled gene graphs associated with a dataset, across datasets with a fixed number of n sequences and s segregating sites.

4.4 Discussion

In this chapter I have illustrated the use of the IS proposal distribution for the G_{81} model by applying it to a real dataset, inferring MLEs for θ and ρ and also performing ancestral inference. Issues of pre-processing were addressed in order to make the data compatible with the model, but the resulting parameter estimates are in good agreement with other studies. Although not a focus of this chapter, the question of how best to prune a dataset was discussed. The greedy algorithm applied here is a natural solution, but there may be more sophisticated approaches which reduce the number of pruning events even further, or prioritize certain prunes based on biological considerations. To take an extreme example, consider an individual who is homozygous at a site in a population in which all other individuals are homozygous for the *other* allele. This individual's two haplotypes could be involved in an incompatibility, yet they are also more likely to exist as a result of a genotyping error. One could argue that it would be preferable to weight this site such that its removal is favoured. An alternative approach would be to allow a small number of back or parallel mutations in the coalescent model explicitly.

The potential of the IS proposal distribution was also demonstrated by addressing questions of ancestral inference for the dataset. Aside from obvious questions—ages

of MRCAs, ages of mutations, number of recombination events—I have also suggested a way to summarize the probable overlap of two adjacent gene trees, in a construct termed the *gene graph*. Here I have largely restricted its use to a visual summary of the histories of the loci, but there is much potential for further research into the distribution of gene graphs, and how well they capture the important features of the histories they represent. A starting point is that I have offered a way to calculate the number of gene graphs associated with a dataset. A problem closely related to gene graphs is the measuring of shared sequence ancestry, and it would also be of interest to obtain exact results on the nature of shared ancestry unconditional on a dataset, extending results of Wiuf & Hein (1999*b*) [89] and possibly mimicking the recursive approach of Griffiths (1991) [31]. Where the corresponding value for a dataset deviates from the unconditional distribution, this could then be informative about possible past events that have shaped the data. I return to this idea in the next chapter.

Chapter 5

Discussion

Performing inference on a dataset in which recombination is allowed to occur has proven to be a challenging problem. In this thesis I have addressed two main issues: the design of efficient IS proposal distributions and the appropriation of sequential Monte Carlo resampling techniques for rejuvenating the weighted sample of the IS procedure. Underlying this research is an interest in the nature of the correlation of neighbouring loci, and for this reason I have focused on a two-locus model separated by a fixed region of recombination. There are several other advantages to this approach: two-locus models are of interest in their own right [5, 29, 30, 31, 32] but can also be used as a starting point for guiding the extension to multiple loci. Indeed, this would be a natural next step for much of the work in this thesis. It is also convenient that the biological nature of recombination has been shown to some extent to support the idealization of blocks of completely linked loci, in the ubiquity of recombination hotspots throughout the genome [57, 58, 64].

Another modelling simplification of which I have frequently made use is the infinite-sites assumption. One might argue that together with selective neutrality,

constant population size, complete linkage within loci, panmictic mating, and so on, the resulting model is an over-simplification. One answer to this is inherent in Kingman's coalescent: it is robust to deviations from many of these assumptions [8]. Moreover, one can find real data which really does behave not unlike the infinite-sites model—recall the modest amount of pruning required of the dataset in Chapter 4—and the gains from assuming perfect conformity to the model are considerable. In the context of IS, one has knowledge of a perfect phylogeny at each locus which gave rise to the data. In the course of performing IS, the sequence of ancestral configurations from the sample back to the root is a parsimonious one; each coalescent and mutation event is guaranteed to bring us closer to the root. Defining a concept of 'distance' from the root is then straightforward, as I have attempted in Chapter 3. By contrast, in a finite-alleles model for example, the concept of distance from the root is much more subtle, and in the course of IS there is scope for vast meanderings in the sequence of ancestral configurations.

Once we accept these modelling assumptions, a main achievement in this thesis has been to design and implement an efficient IS proposal distribution on a coarse partition of ARGs which circumvents the need to consider allelic states at non-ancestral loci. The proposal has been illustrated on an example dataset. A major guide in this endeavour has been the method of De Iorio & Griffiths (2004*a*) [39], who provided a very general method for constructing IS proposal distributions. A consequence of using their method is that in the case of data available only at one locus, my proposal distribution coincides precisely with theirs (which in turn coincides with that of Stephens & Donnelly (2000) [37]). In the two-locus case, a downside is that by using a fragment state space, the resulting equation system for the approximate sampling distribution $\hat{\pi}[(i,j)|\,\mathcal{T},\boldsymbol{n}]$ is at least as complicated as the original

recursion. Nevertheless, after a simple *ad hoc* solution [63], testing of the proposal distribution suggests that it is quite efficient. Unlike the one-locus case, in which the set of histories is finite and there is therefore some 'bound' on just how badly an IS procedure can perform, in a two-locus model there is no bound on the number of recombination events that could be proposed. It is therefore important to propose a reasonable number of recombination events in any history, and while my proposal tries to address this problem, results alluded to in Section 4.3.1 suggest that there is still the need for further refinement. The problem is confounded in multi-locus models, in which one must determine both when and where on the sequence a recombination event should occur. De Iorio & Griffiths' technique is currently the most successful way to obtain an IS proposal distribution, but nevertheless a tough challenge for the future would be to identify in what circumstances a technique like theirs proposes poor choices, and whether there is a general approach for solving the problem. As a motivation, consider the following. For a single locus mutating under a finite-alleles model, one arrives at the approximate sampling distribution given in (1.14). This says: pick an existing type uniformly at random and mutate it a geometric number of times forwards through the transition matrix P. One interpretation of this is that the current sample is an approximation of the population allele frequencies at the time T that the new type coalesced with its closest relative. But if propagation through P is 'slow' then the current sample would be unrepresentative

of the allele frequencies back at time T. For example, suppose

$$P = \begin{bmatrix} \frac{\epsilon}{3} & 1-\epsilon & \frac{\epsilon}{3} & \frac{\epsilon}{3} \\ \frac{\epsilon}{3} & \frac{\epsilon}{3} & 1-\epsilon & \frac{\epsilon}{3} \\ \frac{\epsilon}{3} & \frac{\epsilon}{3} & \frac{\epsilon}{3} & 1-\epsilon \\ 1-\epsilon & \frac{\epsilon}{3} & \frac{\epsilon}{3} & \frac{\epsilon}{3} \end{bmatrix},$$

where $\epsilon \ll 1$, and we have say $\boldsymbol{n} = (0, 5, 0, 0)$, and θ small. Denote the type space by $E = \{1, 2, 3, 4\}$. Then samples from $\hat{\pi}[i|\,\boldsymbol{n}]$ will mostly be of type 2, with perhaps a few of type 3, since (1.14) now follows a geometric random variable with a low success probability. But by consideration of the evolution of types through P, at time T we might expect there to have been some of type 1 in the population, and new samples ought to reflect this. A possible correction to (1.14) would be to have a geometric number of mutations *backwards* through P followed by a geometric number forwards. The two geometric random variables would be correlated by their dependence on T. Of course, suggesting this change immediately destroys many of the properties of (1.14) that make it so practicable [37], as well as the coalescent and diffusion assumptions underlying its derivation [39], but it does at least illustrate that specific problems with the approach are identifiable. Unfortunately, the example given here also applies to the infinite-sites model. One could try to obtain an approximate sampling distribution for infinite-sites data by writing down a transition matrix for a *finite*-sites model with a large number of sites L, and then take a limit in (1.14) as $L \to \infty$. But in the limit, by the observation above, mutating only forwards through P in an L-site model tends to only *adding* segregating sites to existing types. This immediately prohibits the observation of many likely haplotypes having

the misfortune not to have been seen in the sample already. It seems that we are thus precluded from deriving a proposal distribution for infinite-sites data in this way. Together with the technical problems discussed in Chapter 2—for example, can a 'fragment diffusion' be constructed such that a proposal distribution is obtained after applying the approximation (2.7)?—many possible refinements to the design of IS proposal distributions present themselves. It is worth noting that Hobolth *et al.* (2008) [44] have also observed limitations in the Stephens-Donnelly proposal distribution, by showing that it is based on an infinite-*alleles* mutation mechanism: it ignores some of the fine structure in the gene tree and proposes the same changes as if it had access only to an allele frequency spectrum. Beyond refinements to the proposal distribution, there is always of course the opportunity for incorporating more realistic phenomena such as multiple loci, population growth, migration, and gene conversion. An ambitious approach for handling many of these simultaneously would resemble Felsenstein *et al.*'s idea of an "evolutionary genetics black box" [38].

A second contribution of this thesis has been to consider how to apply Chen *et al.*'s [76] stopping-time resampling procedure to infinite-sites data with recombination. Despite the suggestion that their stopping scheme should transfer to infinite-sites data, I did not find this to be the case; resampling can actually have an adverse affect if applied in this way. Hobolth *et al.* (2008) [44] have observed independently the inapplicability of stopping-time resampling to infinite-sites data. I have proposed a new stopping scheme, "scaled-OR", which enables resampling to be applied successfully. It casts the problem explicitly in terms of the 'distance' of an ancestral configuration from the root, which should help in the evaluation of any other proposed refinements. One further suggestion I mentioned in Chapter 3 would be to incorporate at each step $R_{\min}(\mathcal{D})$—provided it could be calculated or estimated

efficiently—which would cover a third distance dimension. Other stopping schemes may be possible, incorporating further modelling features.

Other strategies from sequential Monte Carlo can be similarly exploited. I have also adapted the pilot-exploration resampling technique of Zhang & Liu (2002) [78] and shown that it has the potential for even further improvement in the accuracy of likelihood estimation. It is particularly useful in this setting because it deals head-on with a feature that distinguishes coalescent importance sampling from most sequential Monte Carlo models: the highly non-Markov nature of transitions through the state space. In the construction of a coalescent history, a decision of unusually low weight will likely be compensated for later on, because the weight is *intrinsic* to the data, in the terminology I used in Chapter 3. Spying ahead on the future weight is an obvious, if rather crude, solution to this problem. Pilot-exploration resampling is also complementary to standard resampling techniques as—unlike the core parallel IS procedure—it makes efficient use of a small amount of memory. My conclusions for this method have been somewhat tentative, however, because of the large number of parameters involved and because the pilot-exploration phase requires running times of the same order of magnitude as the IS procedure itself. Nonetheless, for many of the examples considered in this thesis, and for the optimization problems of the original paper, it has been shown to be successful, and doubtless it could find application in many other fields.

Finally, it is worth emphasising the potential for the methods developed here. Two-locus models can tell us much about the fine-scale nature of the affect of recombination on two neighbouring ancestries. The IS methods in this thesis allow the measurement of such properties, but it should be re-iterated that the theory for these ideas *unconditional* on a particular dataset could also be extended. In

Chapter 4 I suggested one definition to aid one's intuition about inference of shared ancestry—what I have called a *gene graph*—and applied it to an example dataset, but there are many other related questions that could be addressed. As motivation, let us consider briefly a possible extension to Wiuf & Hein's definition of shared sequence ancestry [89], which counts the total branch length of each lineage *individually* if it is ancestral at both loci (and does not exclude lineages for which the loci are ancestral to *different* contemporary sequences). Call this quantity T_S. It is straightforward to show that for a dataset $\boldsymbol{n} = (\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c})$, the total shared ancestry $T_S(a, b, c; \rho)$ of the sample satisfies

$$\begin{aligned}
\mathbb{E}[T_S(a,b,c;\rho)] &= \frac{2c}{n(n-1)+\rho c} + \frac{\rho c}{n(n-1)+\rho c}\mathbb{E}[T_S(a+1,b+1,c-1;\rho)] \\
&\quad + \frac{2ab}{n(n-1)+\rho c}\mathbb{E}[T_S(a-1,b-1,c+1;\rho)] \\
&\quad + \frac{a(a+2c-1)}{n(n-1)+\rho c}\mathbb{E}[T_S(a-1,b,c;\rho)] \\
&\quad + \frac{b(b+2c-1)}{n(n-1)+\rho c}\mathbb{E}[T_S(a,b-1,c;\rho)] \\
&\quad + \frac{c(c-1)}{n(n-1)+\rho c}\mathbb{E}[T_S(a,b,c-1;\rho)]. \qquad (5.1)
\end{aligned}$$

This can be obtained for example as a special case of a recursion considered by Griffiths (1991) ([31], his Equation (2.7)). It sums over two-locus ARGs equivalent under $\overset{2}{\sim}$. For equivalence under $\overset{3}{\sim}$, the relevant terms corresponding to reaching a MRCA are modified accordingly: $\mathbb{E}[T_S(a, b, c; \rho)] = 0$ if $a+c = 1$ or if $b+c = 1$; under $\overset{3}{\sim}$ there can be no sharing of ancestry beyond the youngest marginal MRCA. For small samples, the solution to (5.1) can be written down analytically, by standard Markov chain analysis [68, 89]. For example, if $\boldsymbol{n} = (0, 0, 2)$ then there are only three non-absorbing states (A, B, and C in Figure 2.4), and the recursion reduces

to

$$\begin{bmatrix} 1 & -\frac{\rho}{1+\rho} & 0 \\ -\frac{2}{6+\rho} & 1 & -\frac{\rho}{6+\rho} \\ 0 & -\frac{2}{3} & 1 \end{bmatrix} \begin{bmatrix} \mathbb{E}[T_S(0,0,2;\rho)] \\ \mathbb{E}[T_S(1,1,1;\rho)] \\ \mathbb{E}[T_S(2,2,0;\rho)] \end{bmatrix} = \begin{bmatrix} \frac{2}{1+\rho} \\ \frac{2}{6+\rho} \\ 0 \end{bmatrix},$$

and hence

$$\mathbb{E}[T_S(0,0,2;\rho)] = 2\left[1 - \frac{\rho(\rho^2+6\rho+24)}{\rho^3+13\rho^2+96\rho+108}\right],$$

which declines rapidly from 2 when $\rho = 0$ to 0 as $\rho \to \infty$. There is great potential for this sort of analysis, which could occupy another thesis.

Appendix A

List of symbols

A	The left-hand locus $[0, \frac{1}{2}]$ in a two-locus model
A_n	The set of segregating sites at locus A, in a sample of size n
$\mathcal{A}_n$	An ordered configuration of types $(a_1, \ldots, a_n)$ sampled from a population
a	The number of sequences in the sample which are ancestral only at locus A
α_j	In the infinite-alleles model, the number of alleles with j representatives in the sample
α	(Chapter 3): The power to which an estimated future weight is raised in a pilot-exploration resampling procedure
α	(Chapter 4): An order relation on the nodes M_j of a gene graph, equivalent to $\mathcal{G}_A$
$\bar{\alpha}$	An order relation recording prohibited age orderings of the nodes M_j of a gene graph at locus A
B	The right-hand locus $[\frac{1}{2}, 1]$ in a two-locus model
B	The tuning parameter in a dynamic sequential importance resampling schedule
B_n	The set of segregating sites at locus B, in a sample of size n
b	The number of sequences in the sample which are ancestral only at locus B
β	An order relation on the nodes M_j of a gene graph, equivalent to $\mathcal{G}_B$

$\bar{\beta}$	An order relation recording prohibited age orderings of the nodes M_j of a gene graph at locus B
C	Used to denote genes ancestral at both loci in a two-locus model
c	The number of sequences in the sample which are ancestral both at locus A and at locus B
$\mathcal{C}_j$	The event that a gene of type j is the first to be involved in an event back in time
cv	The co-efficient of variation of sequential importance weights, $\frac{\sigma_W}{\mu_W}$
D	$n(n-1) + (a+c)\theta_A + (b+c)\theta_B + \rho c$
D_0	$n(n-1) + n\theta_A + n\theta_B + \rho n^C$
$\widetilde{D}$	$n(n-1) + (a+c)\theta_A + (b+c)\theta_B$
d	The number of alleles in a finite-alleles model, $\lvert E\rvert$; or the number of distinct sequences observed in a sample from an infinite-sites model
d_A	The number of alleles in a two-locus finite-alleles model, $\lvert E_A\rvert$
d_B	The number of alleles in a two-locus finite-alleles model, $\lvert E_B\rvert$
δ_{ij}	The Kronecker delta
$\delta(x)$	The Dirac delta function at x
δ	The number of steps in an exhaustive lookahead procedure
Δ	The number of steps in a pilot-exploration procedure
$\mathcal{D}$	A sampled dataset, equivalent to the configuration H_0 at the tips of a coalescent tree consistent with such data
E	The type space in a finite-alleles model
E_A	The type space in a finite-alleles model at locus A only
E_B	The type space in a finite-alleles model at locus B only
$\mathcal{E}$	An event associated with a coalescent history $\mathcal{H}$
ESS	The effective sample size, $\frac{N}{1+cv^2}$
$\boldsymbol{e}_j$	A unit vector whose jth entry is 1 and all others are 0
$\boldsymbol{e}_{ij}$	A unit matrix whose (i,j)th entry is 1 and all others are 0
ϵ	A prior parameter for the number of observations of type (i,j) in the two-locus, infinite-sites proposal distribution
$\mathcal{G}$	A gene graph, $(\mathcal{G}_A, \mathcal{G}_B)$
$\mathcal{G}_A$	The vector of paths to the root at locus A of a gene graph
$\mathcal{G}_B$	The vector of paths to the root at locus B of a gene graph
Γ	The set $\{\mathrm{A}, \mathrm{B}, \mathrm{C}\}$
γ	An element of Γ to indicate at which loci a gene is ancestral to the sample

G_{81}	A shorthand for a two-locus model with a single recombination breakpoint position separating the loci, and within loci are a sequence of completely linked sites mutating under the infinite-sites model—also considered by Griffiths (1981) [5]
H_0	The configuration at the tips of a coalescent tree
H_k	The configuration in a coalescent tree $-k$ events back in time, $k = 0, -1, \ldots$
H_{-m}	The most recent common ancestor of a coalescent tree, encountered after m events
$\mathcal{H}$	A coalescent history, determined by a sequence of intermediate configurations $(H_0, H_{-1}, \ldots, H_{-m})$
$\mathcal{H}_{-t}$	A partially reconstructed history t steps back
I	A subset of $\mathbb{R}$
I_A	An indexing set for the set of observed and inferred types associated with the gene tree of a dataset at locus A
I_B	An indexing set for the set of observed and inferred types associated with the gene tree of a dataset at locus B
I_A^*	$I_A \cup \{*\}$
I_B^*	$I_B \cup \{*\}$
J_k^j	The set of sequences in a finite-sites model which differ from type j at one site and which, if they replaced type j in the sample, would alter the number of segregating sites by k
K	The state space of a diffusion process
κ_l	The number of segregating sites on the lth branch of a coalescent tree
$\kappa_A[(i,j)\vert\, \boldsymbol{c}]$	In the two-locus, infinite-sites proposal distribution, the estimated posterior probability of observing a type (i,j) given $\boldsymbol{c}$ and that the type at locus B is j, (2.29)
$\kappa_B[(i,j)\vert\, \boldsymbol{c}]$	In the two-locus, infinite-sites proposal distribution, the estimated posterior probability of observing a type (i,j) given $\boldsymbol{c}$ and that the type at locus A is i, (2.32)
L	The number of sites in a finite-sites model
L_n	Total branch length of a coalescent tree of n genes
Λ_i	The number of copies of stream i guaranteed to be retained in a residual resampling scheme
$\mathscr{L}$	The generator of a diffusion process

M	The number of individuals in a diploid Wright-Fisher model
M_n	The random variable whose outcomes are the number of mutation events in a genealogy
M_j	The set of nodes in a gene graph, $Z_n \cup A_n \cup B_n \cup R_j$
m_j	$\lvert M_j \rvert$
M_e	The effective population size of a Wright-Fisher model
m	The team size in a pilot-exploration procedure
m°	The number of genes that can be involved in the next event back in time
$\texttt{ms}(n, \theta, \rho)$	The random variable whose realizations are draws from Hudson's `ms` program [65] with sample size n, two loci each with mutation parameter $\theta/2$, and separated by a breakpoint with recombination parameter ρ
μ_A	A tuning parameter for stopping schemes sOR and sAND
μ_B	A tuning parameter for stopping schemes sOR and sAND
μ_W	$\mathbb{E}(W) = p(\mathcal{D}) = L(\theta)$
N	The number of runs in an IS scheme
N_r	$N - \Lambda_1 - \ldots - \Lambda_N$
$\boldsymbol{n}$	The vector of multiplicities for each distinct type in the sample
$\boldsymbol{n}_A$	The vector of multiplicities for each distinct type in the sample which is ancestral only at locus A
$\boldsymbol{n}_B$	The vector of multiplicities for each distinct type in the sample which is ancestral only at locus B
$\boldsymbol{n}_C$	The vector of multiplicities for each distinct type in the sample which is ancestral at both loci
n	Sample size $n^A + n^B + n^C$, or $a + b + c$ in a fragment model
n^A	The number of sequences in the sample which are ancestral only at locus A. The alleles at both loci are still specified
n^B	The number of sequences in the sample which are ancestral only at locus B. The alleles at both loci are still specified
n^C	The number of sequences in the sample which are ancestral at both loci
n_i	The multiplicity of type $i \in E$
n°	The number of genes that can be involved in a coalescence or mutation as the next event back in time
ν	A tuning parameter for stopping schemes sOR and sAND
O_i	The set of row indices of an incidence matrix for which the ith column is 1

P	The transition matrix in a finite-alleles model
P^A	The transition matrix in a two-locus finite-alleles model at locus A
P^B	The transition matrix in a two-locus finite-alleles model at locus B
p	The probability of a sample configuration/intermediate configuration/entire history drawn from the coalescent process, with the appropriate meaning implied from the context of use
q	Either the IS proposal distribution, or a version of p for an ordered sample, with the appropriate meaning implied from the context of use
q^*	The optimal IS proposal distribution—on histories this is $p(\mathcal{H}\vert\,\mathcal{D})$
θ	The population scaled coalescent mutation rate $4M_e u$; the total such rate $\theta_A + \theta_B$ in a two-locus model
θ_A	The population scaled coalescent mutation rate at locus A
θ_B	The population scaled coalescent mutation rate at locus B
θ_{A0}	An importance sampling driving value for θ_A
θ_{B0}	An importance sampling driving value for θ_B
$\hat{\theta}_A$	A maximum likelihood estimate of θ_A
$\hat{\theta}_B$	A maximum likelihood estimate of θ_B
$\tilde{\theta}$	Watterson's estimate [1] of θ
R_n^0	The random variable whose outcomes are the number of recombination events in an ancestral recombination graph
R_n	The random variable whose outcomes are the number of recombination events occurring within ancestral material in an ancestral recombination graph
R_0	A set containing the node which is the MRCA of both locus A and locus B, $\{0\}$
R_1	A set containing the nodes which are the distinct MRCAs of locus A and B, $\{0_A, 0_B\}$
$R_{\min}(\mathcal{D})$	The minimum possible number of recombination events that could have occurred in giving rise to the data $\mathcal{D}$
r	The recombination probability per gene per generation in a Wright-Fisher model
$r(\boldsymbol{n}, \boldsymbol{m})$	The forward co-efficient in a recursion for the probability of the sample configuration, encapsulating a change of state backwards in time from $\boldsymbol{n}$ to $\boldsymbol{m}$

ρ	The population scaled coalescent recombination rate $4M_e r$
ρ_0	An importance sampling driving value for ρ
$\hat{\rho}$	A maximum likelihood estimate of ρ
S_n	The random variable whose outcomes are the number of segregating sites in samples drawn from a coalescent model
S_n^A	The random variable whose outcomes are the number of segregating sites at locus A in samples drawn from a coalescent model
S_n^B	The random variable whose outcomes are the number of segregating sites at locus B in samples drawn from a coalescent model
S_n^k	The random variable whose outcomes are the number of segregating sites while k ancestors in histories drawn from a coalescent model
S_t	A weighted collection $\left\{(x_t^{(1)}, w_t^{(1)}), \ldots, (x_t^{(N)}, w_t^{(N)})\right\}$ of streams in a sequential Monte Carlo scheme
s	The total number of segregating sites $s_n^A + s_n^B$; realizations of S_n
s_n^A	The number of segregating sites at locus A; realizations of S_n^A
s_n^B	The number of segregating sites at locus B; realizations of S_n^B
σ	A permutation of $\{1, \ldots, n\}$
σ_W	$\sqrt{\mathrm{var}(W)}$
T_k	The time of the epoch while there are k ancestors
T_l	The lth stopping-time for a stopping-time resampling scheme
T_{MRCA}	The time to the most recent common ancestor
T_{A}	The time to the most recent common ancestor at locus A, $\inf\{t \in \mathbb{R}_+ : a + c = 1\}$
T_{B}	The time to the most recent common ancestor at locus B, $\inf\{t \in \mathbb{R}_+ : b + c = 1\}$
T_S	The total branch length in a two-locus ARG of lineages ancestral at both loci, and ancestral not necessarily to the same sequence
$\mathcal{T}$	The vector of paths to the root of a gene tree representing each distinct allele
$\mathcal{T}_A$	The marginal gene tree at locus A
$\mathcal{T}_B$	The marginal gene tree at locus B
t_A	An order relation on the nodes of M_j of a gene graph, encapsulating the gene tree $\mathcal{T}_A$
t_B	An order relation on the nodes of M_j of a gene graph, encapsulating the gene tree $\mathcal{T}_B$

U	The $d \times d$ matrix whose rows are all equal to $\boldsymbol{n}$
u	The mutation probability per gene per generation in a Wright-Fisher model
$\boldsymbol{u}$	The parameters of the Dirichlet prior for the number of observations of types (i, j) with one index fixed, for the two-locus, infinite-sites proposal distribution
Υ	The set $\{0, 10^{-10}, 0.1, 0.5, 1, 2, 5, 10, 20, 10^{10}\}$
υ	The correlation co-efficient of current versus final sequential importance sampling weight, with weights measured on a $\log_{10}$ scale
$\mathcal{V}$	The event "a history $\mathcal{H}$ is drawn from $\mathcal{Y}$ and survived resampling"
W	The random variable whose outcomes are importance sampling weights of histories $\mathcal{H}$ drawn from q
W_t	The random variable whose outcomes are current sequential importance sampling weights after t steps, associated with partially reconstructed genealogies drawn from q
w_t	The current sequential importance sampling weight after t steps, associated with a partially reconstructed genealogy drawn from q; realizations of W_t
$\boldsymbol{X}$	The random variable which is a matrix (X_{ij}) of allele frequencies for each type $(i, j) \in E_A \times E_B$ in a diffusion process
$\mathcal{X}$	The random variable whose outcomes are coalescent histories $\mathcal{H}$ under the coalescent model
$(x_{i0}, \ldots)$	The sequence of mutation labels for the ith type in a gene tree, from the present back to the root
x_t	A stream at the tth stage of a Sequential Monte Carlo scheme
ξ	Total length of ancestral material in a sample of n sequences, $\frac{1}{2}(a + b + 2c)$
$\mathscr{X}$	The state space for the hidden process in sequential Monte Carlo
$\mathcal{Y}$	The random variable whose outcomes are coalescent histories $\mathcal{H}$ drawn from q
$\mathscr{Y}$	The state space of observations in sequential Monte Carlo
Z	The recombination breakpoint distribution on $[0, 1]$
Z_n	A set of leaf nodes in a gene graph for n sequences, $\{z_1, \ldots, z_n\}$
ζ	A bijection $\zeta : [0, 1] \to [0, 1]$ which relabels mutations in a gene tree

Appendix B

Deriving an infinite-sites recursion from a finite-sites recursion

For simplicity I shall consider only a one-locus model—multi-locus models follow in a similar manner. Suppose a sample $\boldsymbol{n}$ has s segregating sites, and is mutating under a finite-sites model with L-sites. For notational simplicity, assume symmetric mutation at each site, each with transition matrix

$$Q = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

So there are 2^L alleles in a type space E of binary strings of length L, and here we suppose that the root is known to be the zero string, though this can be relaxed easily. The complete transition matrix is

$$P = \frac{1}{L} \sum_{j=1}^{L} I \otimes I \otimes \ldots \otimes Q \otimes \ldots \otimes I,$$

where Q is in the jth position in the direct product. A recursion for a sample under this model is

$$\begin{aligned} p(\boldsymbol{n}) &= \frac{n-1}{n-1+\theta}\sum_{j:\,n_j\geq 2}\frac{n_j-1}{n-1}p(\boldsymbol{n}-\boldsymbol{e}_j) \\ &\quad + \frac{\theta}{n-1+\theta}\sum_{j:\,n_j\geq 1}\frac{1}{L}\sum_{i\in J_{-1}^j\cup J_0^j\cup J_1^j}\frac{n_i+1}{n}p(\boldsymbol{n}-\boldsymbol{e}_j+\boldsymbol{e}_i), \end{aligned} \tag{B.1}$$

where J_k^j denotes the set of sequences that differ from type j at one site and which, if they replaced type j in the sample, would alter the number of segregating sites by k. This recursion can be derived in a manner similar to the recursions obtained in Chapter 2. In this construction sites are distinguished by their position. We shall be interested in the probability of a sample whose sites are not labelled by their position, which is given by the quantity

$$p(\boldsymbol{n}') = \binom{L}{s}\binom{s}{\boldsymbol{\kappa}^L}p(\boldsymbol{n}),$$

where $\boldsymbol{n}'$ is the sample configuration without regard to the position of each mutation, and $\boldsymbol{\kappa}^L = (\kappa_1^L, \ldots, \kappa_m^L)$ is a vector of the multiplicities of identical segregating columns. To see that this is true, note that $\binom{L}{s}$ is the number of sets of s positions we could have chosen to be segregating, and $\binom{s}{\boldsymbol{\kappa}^L}$ is the number of ways of allocating a given configuration amongst these positions. $p(\boldsymbol{n})$ is independent of the positions

of the sites, so this is a sum of $\binom{L}{s}\binom{s}{\boldsymbol{\kappa}^L}$ equal terms. Substitute for $p(\boldsymbol{n}')$ in (B.1):

$$\begin{aligned} p(\boldsymbol{n}') &= \frac{n-1}{n-1+\theta}\sum_{j:\,n_j\geq 2}\frac{n_j-1}{n-1}p((\boldsymbol{n}-\boldsymbol{e}_j)') \\ &\quad + \frac{\theta}{n-1+\theta}\sum_{j:\,n_j\geq 1}\frac{1}{L}\sum_{i\in J_{-1}^j}\frac{n_i+1}{n}\frac{L-s+1}{\kappa_l^L}p((\boldsymbol{n}-\boldsymbol{e}_j+\boldsymbol{e}_i)') \\ &\quad + \frac{\theta}{n-1+\theta}\sum_{j:\,n_j\geq 1}\frac{1}{L}\sum_{i\in J_0^j}\frac{n_i+1}{n}\frac{\kappa_{l'}^L+1}{\kappa_l^L}p((\boldsymbol{n}-\boldsymbol{e}_j+\boldsymbol{e}_i)') \\ &\quad + \frac{\theta}{n-1+\theta}\sum_{j:\,n_j\geq 1}\frac{1}{L}\sum_{i\in J_1^j}\frac{n_i+1}{n}\frac{\kappa_{l'}^L+1}{L-s}p((\boldsymbol{n}-\boldsymbol{e}_j+\boldsymbol{e}_i)'). \end{aligned}$$

On the right-hand side, the second term represents the removal of a singleton mutation, the third term represents a mutation at a segregating but non-singleton site, and the fourth term represents a mutation that creates a new segregating site back in time. In the second and third terms, κ_l^L is the entry in $\boldsymbol{\kappa}^L$ reduced by one when a mutation occurs to j, resulting in i (so l depends implicitly on i and j). In the third and fourth terms, $\kappa_{l'}^L$ is the multiplicity increased by one as a result of the mutation (similarly determined by i and j).

Let $L\to\infty$. We have that $\boldsymbol{\kappa}^L\to\boldsymbol{\kappa}$, where $\boldsymbol{\kappa}$ is as defined in Chapter 2: it is the vector giving the number of mutations on each branch of a coalescent tree associated with the data. The recursion becomes

$$\begin{aligned} p(\boldsymbol{n}') &= \frac{n-1}{n-1+\theta}\sum_{j:\,n_j\geq 2}\frac{n_j-1}{n-1}p((\boldsymbol{n}-\boldsymbol{e}_j)') \\ &\quad + \frac{\theta}{n-1+\theta}\sum_{j:\,n_j\geq 1}\sum_{i\in J_{-1}^j}\frac{n_i+1}{n}\frac{1}{\kappa_l}p((\boldsymbol{n}-\boldsymbol{e}_j+\boldsymbol{e}_i)'), \end{aligned}$$

where κ_l becomes the number of removable mutations from type j, and is now the

same for all i (since the i's correspond to mutations residing on the same branch of the coalescent tree). The third and fourth terms disappear; only sample configurations compatible with a gene tree, $\boldsymbol{n}' = (\mathcal{T}, \boldsymbol{n})$, have non-zero probability. Since $|J^j_{-1}| = \kappa_l$ and since $(\boldsymbol{n} - \boldsymbol{e}_j + \boldsymbol{e}_i)'$ is the same for each $i \in J^j_{-1}$ after unordering the sites, we recover the infinite-sites recursion [12], with $\boldsymbol{n}$ now representing the multiplicity of types *in a gene tree*:

$$\begin{aligned} p(\mathcal{T}, \boldsymbol{n}) &= \frac{n-1}{n-1+\theta} \sum_{j:\, n_j \geq 2} \frac{n_j - 1}{n-1} p(\mathcal{T}, \boldsymbol{n} - \boldsymbol{e}_j) \\ &\quad + \frac{\theta}{n-1+\theta} \sum_{\substack{j:\, n_j = 1 \\ j \to i}} \frac{n_i + 1}{n} p(\mathcal{T}'_{j-}, \boldsymbol{n} - \boldsymbol{e}_j + \boldsymbol{e}_i). \end{aligned}$$

Appendix C

Instructions for `rita`

This section describes how to use the C++ program `rita` (**r**ecombining, **i**nfinite-sites, **t**wo-locus **a**ncestries, v1.0), the program described in Section 2.3.4 for performing importance sampling on the G_{81} model. It compiles in Linux or in Windows with standard libraries, and is available on request. `rita` utilizes a random number generator previously implemented by R.C. Griffiths and based on that of Marsaglia & Zaman (1991) [69].

To compile in Linux, go to the source file directory and type

```
g++ *.cpp -o rita
```

which creates a file called `rita`. The basic command line is then

```
./rita input_file tA tB r N S
```

where `input_file` is the input dataset, (`tA`, `tB`, `r`) is the vector of driving values for $(\theta_A, \theta_B, \rho)$, `N` is the number of runs and `S` is a positive integer which serves as the random number seed. Further options are specified using command-line switches, described below.

C.1 The input file

The format for the input dataset mimics that of `genetree` [20], which accepts inputs for gene trees such as:

```
2  :  4 3 0
1  :  2 0
1  :  1 0
```

This describes the gene tree shown in Figure 1.2. Each row holds information on a distinct allele, with the multiplicity followed by a path to the root. In the two-locus case, each row should be of the form:

`n : `s_1 s_2 `... 0 | `s_3 s_4 `... 0 |`

Here, `n` is the multiplicity for this allele and s_1, s_2, s_3, s_4, ... should be the distinct positive integers $\{1, \ldots, s\}$ labelling each mutation and providing the path to the root. The first path encodes the marginal gene tree at locus A and should use the labels $\{1, \ldots, s_n^A\}$, and the second path should use the labels $\{s_n^A + 1, \ldots, s\}$ for locus B. Each root is given by `0`. The multiplicity is separated by a `:` symbol, and the end of each locus is identified by a `|` symbol. Each character should be separated with whitespace. For each allele, data can be missing at one locus but not both. For example, if the genes in the gene tree above were observed to be monomorphic at another locus, except for one gene whose data was missing there, then the input data might look like:

```
1  :  4 3 0  |  0  |
1  :  4 3 0  |     |
1  :  2 0    |  0  |
1  :  1 0    |  0  |
```

Multiple datasets can be input in a single file; each should be separated by a `//` symbol (also surrounded by whitespace). Importance sampling using the same options

can then be performed on each dataset by using the `-D` switch—see below.

C.2 The output file

The basic command line will output an estimate of the likelihood, standard error and ESS to the screen and to a file called `likelihoods.txt` in a tab-delimited format (with a header row). It will also show the number of resampling events and the time in seconds for this experiment. In the output file, the row is preceded by a `0` to label the dataset and a `0` to label the experiment; multiple datasets and multiple independent experiments can then be reported in a single file using the `-D` and `-s` switches respectively—see below. Further output files are created depending on other switches used; these are also described below.

C.3 Other switches

Importance sampling on multiple datasets using -D

Multiple datasets can be input in a single file as described above, but by default IS is performed only on the first one. To perform IS on more than one dataset using the same options (except the seed, which is not reset between datasets) append the switch `-D u`, where `u` is the number of datasets. In the output file they are labelled `0`, ..., `u-1`. For example,

```
./rita input_file 2.5 2.5 0 10000 1 -D 10
```

performs $N = 10,000$ runs of IS using driving values $(\theta_{A0}, \theta_{B0}, \rho_0) = (2.5, 2.5, 0)$ on each of the 10 datasets in `input_file`.

Independent replicates of importance sampling on the same dataset using -s

To get an idea of the variability in likelihood estimates, IS can be repeated several times using the same options (except the seed, which is not reset between each experiment). To achieve this, append the switch `-s v`, where `v` is the number of independent experiments. In the output file they are labelled `0`, ..., `v-1`. For example,

```
./rita input_file 2.5 2.5 0 10000 1 -s 25
```

performs $N = 10,000$ runs of IS using driving values $(\theta_{A0}, \theta_{B0}, \rho_0) = (2.5, 2.5, 0)$ independently 25 times on the dataset in `input_file`.

The switches `-D` and `-s` can be combined; then `likelihoods.txt` contains `uv` likelihood estimates.

Importance sampling over a grid of θ_A values using -a

The likelihood can be estimated at a range of values for θ_A by appending the switch `-a L R P`, where the likelihood is estimated for θ_A in the range `[L, R]` at `P` equally spaced points. In the absence of the parallel switch `-p` (see below) this is achieved by performing IS *independently* at each of these points, using each point as a driving value. For example,

```
./rita input_file 2.5 2.5 0 10000 1 -a 15 35 5
```

performs $N = 10,000$ runs of IS five times independently, using driving values $(15, 2.5, 0)$, $(20, 2.5, 0)$, ..., $(35, 2.5, 0)$ on the dataset in `input_file`. The original specification for $\theta_{A0} = 2.5$ is now ignored, but any number is still required here for the remaining options to occupy the correct position in the command line.

A file containing the estimated likelihood at each of the values in the vector for θ_A is created, called `surface_data#.txt`, where `#` is the label for the dataset (`0` by default). The file is in tab-delimited format; each row contains the current values for θ_A, θ_B, ρ, the estimated likelihood, and the standard error. For longer simulations each row can be time-consuming, and so this file is constructed step-by-step in a `surface_temp.txt` file, which is then deleted when IS on the current dataset is complete. Since several driving values are in use, this replaces the `likelihoods.txt` file which is no longer applicable.

Importance sampling over a grid of θ_B values using -b

This operates in a way perfectly analogous to θ_A, and can be used in combination with it to produce an estimated likelihood surface.

Importance sampling over a grid of θ values using -ab

This switch is used in the same way as `-a` with the additional assumption that $\theta_A = \theta_B$. For example,

```
./rita input_file 2.5 2.5 0 10000 1 -ab 15 35 5
```

performs $N = 10,000$ runs of IS five times independently, using driving values $(15, 15, 0)$, $(20, 20, 0)$, ..., $(35, 35, 0)$ on the dataset in `input_file`. It should not be used in conjunction with either `-a` or `-b`.

Importance sampling over a grid of ρ values using -r

This operates in a way perfectly analogous to θ_A, and can be used in combination with `-a` and `-b` to produce an estimated likelihood hypersurface.

Estimating a likelihood surface from a single driving value using `-p`

Adding the `-p` switch to the command line now re-interprets instances of `-a`, `-b`, `-ab` and `-r`. Likelihood estimates over the gridpoints specified by these switches are now produced from a single driving value, using the method of (1.7). For example,

```
./rita input_file 25 2.5 0 10000 1 -a 15 35 5 -p
```

performs $N = 10,000$ runs of IS using driving value $(25, 2.5, 0)$ on the dataset in `input_file`. The likelihood estimate at this driving value is output to the file `likelihoods.txt`, and the estimate at five points in the range $\theta_A \in [15, 35]$ are output to `surface_data0.txt`.

Specifying ϵ using `-e`

The parameter ϵ in the proposal distribution can be specified using `-e`. For example:

```
./rita input_file 25 2.5 0 10000 1 -e 10
```

uses $\epsilon = 10$ rather than the default $\epsilon = 1$. Special cases are `-e 0`, which specifies $\kappa_A = \kappa_B = 1$, and `-e -1`, which replaces the whole scheme with that of Griffiths & Marjoram (1996) [40].

Appending `r` to the switch allows one to test a range of ϵ values, and subsequently appending `s` specifies that each ϵ should utilize the same random number seed. This minimizes differences in likelihood estimates which are not explicable by the choice of ϵ. For example,

```
./rita input_file 25 2.5 0 10000 1 -ers 10 100 10
```

performs IS nine times—once for each $\epsilon \in \{10, 20, \ldots, 100\}$, using the same seed each time.

Stopping-time resampling using `-B#`

Stopping-time resampling can be initiated using `-B#`, where `#` is either `0` (none), `1` (CXL), `2` (AND), `3` (OR), `4` (sOR), or `5` (sAND). The next number in the command line determines B. A range of B values can be tested by appending `r` to the switch—for example,

```
./rita input_file 2.5 2.5 0 10000 1 -B4 80
```

performs sOR resampling with $B = 80$, while

```
./rita input_file 2.5 2.5 0 10000 1 -B4r 0 100 11
```

performs IS eleven times independently, with sOR resampling and respective thresholds $B = 0, 10, \ldots, 100$.

Resampling is memory-hungry. At present each stream is stored entirely in RAM, and this can be used up quickly—particularly if a whole likelihood hypersurface is constructed with each. It is advised that you keep a close eye on the memory usage when invoking this switch.

Pilot-exploration resampling using `-B#h`, `-m`, `-t`

Pilot-exploration resampling can be invoked by appending `h` to the switch for resampling, or appending `hc` for composite pilot-exploration resampling. The threshold for B (or range of thresholds when `-B#r` was used) should then be followed by values for Δ, m and α. Additional switches `-t` and `-m` can be used to replace these single values with ranges respectively for Δ and m. For example,

```
./rita input_file 2.5 2.5 0 10000 1 -B4h 80 4 50 0.5
```

performs SISPER with stopping times defined by sOR, and parameters $B = 80$, $\Delta = 4$, $m = 50$, $\alpha = 0.5$, while

```
./rita input_file 2.5 2.5 0 10000 1 -B4h 80 4 50 0.5 -t 1 4 4
```

replaces this experiment with five independent estimates, using each $\Delta \in \{1, 2, 3, 4\}$ (the original 4 is now ignored). This can be used for testing the effect of varying Δ.

Varying the parameter values exponentially

Appending `e` to any switch specifying a range of parameter values—that is, `-a`, `-b`, `-r`, `-er`, `-B#r`, `-m`, and `-t`—replaces the set of points in this range with 10 raised to them (2 in the case of `-B#r`). For example,

```
./rita input_file 2.5 2.5 0 10000 1 -re -4 4 9
```

performs IS nine times independently on the dataset in `input_file`, using driving values $(2.5, 2.5, 10^{-4})$, $(2.5, 2.5, 10^{-3})$, ..., $(2.5, 2.5, 10^{4})$.

Controlling the precision of estimates using -o

The switch `-o` can be used to specify the number of decimal places displayed in outputs to the screen and to files. Note that it does not affect precision of the program. The switch should be followed by two integers, specifying the number of decimal places for displaying parameters and for displaying likelihood estimates, respectively. For example,

```
./rita input_file 2.5 2.5 0 10000 1 -o 2 3
```

would display the driving values as $(2.50, 2.50, 0.00)$ and the likelihood would look like 1.234×10^{-5}. Default values are 2 and 6 respectively.

Inference and debugging with -d

Inferential and debugging output from IS can be specified with the switch `-d w`, where `w` is the level of output, from 1 to 6. These levels are nested, so that level `w+1` collects all the information in level `w`. Levels are as follows:

1. `Inference`: Outputs basic inference to screen, including estimated ages of mutations and MRCAs, number of recombination events, and $\mathbb{P}(T_A = T_B)$, each with an estimated standard error. When IS is over a range of values for $(\theta_A, \theta_B, \rho)$, this inference is output to a file `inference_data#.txt` for each combination of parameters, where `#` is the label for the dataset (`0` by default). When `v` independent experiments are performed using `-s`, the files are named `inference_data#_exp##.txt`, with `##` running from `0` to `v-1`.

2. `End weights`: Outputs a column vector with the weight at the end of each run to a file called `histograms_data#.txt`, where `#` is the label for the dataset. A suffix for multiple experiments can be added, as above. If a range of driving values are used, for example i values for θ_{A0}, j values for θ_{B0}, and k values for ρ_0, then each file is named `histograms_data#_u_v_w.txt`, where `u` runs from $0, \ldots, i-1$, `v` runs from $0, \ldots, j-1$, and `w` runs from $0, \ldots, k-1$.

3. `All weights` (with resampling activated): Outputs the complete set of weights at each stopping-time to a file `intermediate_weights_data#.txt`, where `#` is the label for the dataset. A suffix for multiple experiments or multiple driving values can be added, as above. Interspersing each block of N weights in the file is a vector of N integers, recording the total number of moves each stream underwent to reach this stopping-time.

4. `All stats`: Replaces the column of N weights in `histograms_data#.txt` with an $N \times (s+4)$ matrix. Each row records, for this run: its weight, the expected age of the MRCA at locus A, the expected age of the MRCA at locus B, the expected age of each mutation, the number of recombination events.

5. `Gene graphs`: Below the column of weights in `histograms_data#.txt`, also output is the leaf-labelled gene graph from each run, in the following format:

AB	:	I					
0	:	x_{00}	x_{01}	...	y_{00}	y_{01}	...
1	:	x_{10}	x_{11}	...	y_{10}	y_{11}	...
	⋮				⋮		
n-1	:	$x_{(n-1)0}$	$x_{(n-1)1}$	...	$y_{(n-1)0}$	$y_{(n-1)1}$	...

   Here, `I` is an indicator for the coincidence of the two MRCAs (so is either `0` or `1`), and below this the ith row represents the path from the ith leaf back to the root, in the form introduced in Section 4.3.2. If the marginal MRCAs are distinct, their labels in these paths are `0` and `s+1` for locus A and B respectively, otherwise it is `0` for both. Gene graphs in this file are separated by a `//` symbol.

6. `Full`: Full debugging information, including an output to the screen of the marginal gene trees associated with every intermediate ancestral configuration of each run.

A less well-supported option: -l

The switch `-l` implements the analytic lookahead procedure discussed in Section 3.4. Extensive debugging suggests that its results are correct; they are simply less accurate than when this switch is not used.

Bibliography

[1] G. A. Watterson. On the number of segregating sites in genetical models without recombination. *Theoretical Population Biology*, 7:256–276, 1975.

[2] P. Marjoram and S. Tavaré. Modern computational approaches for analysing molecular genetic variation data. *Nature Reviews Genetics*, 7:759–770, 2006.

[3] J. F. C. Kingman. On the genealogy of large populations. *Journal of Applied Probability*, 19:27–43, 1982.

[4] J. F. C. Kingman. The coalescent. *Stochastic processes and their applications*, 13(3):235–248, 1982.

[5] R. C. Griffiths. Neutral two-locus multiple allele models with recombination. *Theoretical Population Biology*, 19:169–186, 1981.

[6] R. R. Hudson. Properties of a neutral allele model with intragenic recombination. *Theoretical Population Biology*, 23:183–201, 1983.

[7] J. Hein, M. H. Schierup, and C. Wiuf. *Gene genealogies, variation and evolution*. Oxford University Press, 2005.

[8] M. Nordborg. Coalescent theory. In D. Balding, M. Bishop, and C. Cannings, editors, *Handbook of Statistical Genetics*, chapter 7. Wiley, Chichester, UK, 2001.

[9] R. C. Griffiths and S. Tavaré. Simulating probability distributions in the coalescent. *Theoretical Population Biology*, 46:131–159, 1994.

[10] R. C. Griffiths and S. Tavaré. Sampling theory for neutral alleles in a varying environment. *Philosophical Transactions of the Royal Society B*, 344:403–410, 1994.

[11] R. C. Griffiths and S. Tavaré. Ancestral inference in population genetics. *Statistical Science*, 9:307–319, 1994.

[12] R. C. Griffiths and S. Tavaré. Unrooted genealogical tree probabilities in the infinitely-many-sites model. *Mathematical Biosciences*, 127:77–98, 1995.

[13] R. C. Griffiths and S. Tavaré. Computational methods for the coalescent. In P. Donnelly and S. Tavaré, editors, *Progress in population genetics and human evolution*, volume 87, pages 165–182. Springer-Verlag, Berlin, 1997.

[14] M. K. Kuhner, J. Yamato, and J. Felsenstein. Estimating effective population size and mutation rate from sequence data using Metropolis-Hastings sampling. *Genetics*, 140:1421–1430, 1995.

[15] M. K. Kuhner, J. Yamato, and J. Felsenstein. Maximum likelihood estimation of recombination rates from population data. *Genetics*, 156:1393–1401, 2000.

[16] M. Kimura. The number of heterozygous nucleotide sites maintained in a finite population due to steady flux of mutations. *Genetics*, 61:893–903, 1969.

[17] R. R. Hudson and N. L. Kaplan. Statistical properties of the number of recombination events in the history of a sample of DNA sequences. *Genetics*, 111:147–164, 1985.

[18] D. Gusfield. Efficient algorithms for inferring evolutionary trees. *Networks*, 21:19–28, 1991.

[19] R. C. Griffiths. Genealogical-tree probabilities in the infinitely-many-site model. *Journal of Mathematical Biology*, 27:667–680, 1989.

[20] M. Bahlo and R.C. Griffiths. Inference from gene trees in a subdivided population. *Theoretical Population Biology*, 57:79–95, 2000.

[21] T. Ohta and M. Kimura. A model of mutation appropriate to estimate the number of electrophoretically detectable alleles in a genetic population. *Genetical Research*, 22:201–204, 1973.

[22] R. Nielsen. A likelihood approach to populations samples of microsatellite alleles. *Genetics*, 146:711–716, 1997.

[23] M. De Iorio, R. C. Griffiths, R. Leblois, and F. Rousset. Stepwise mutation likelihood computation by sequential importance sampling in subdivided population models. *Theoretical Population Biology*, 68:41–53, 2005.

[24] C. Wiuf and J. Hein. The coalescent with gene conversion. *Genetics*, 155:451–462, 2000.

[25] C. Wiuf. A coalescence approach to gene conversion. *Theoretical Population Biology*, 57:357–367, 2000.

[26] R. C. Griffiths and P. Marjoram. An ancestral recombination graph. In P. Donnelly and S. Tavaré, editors, *Progress in population genetics and human evolution*, volume 87, pages 257–270. Springer-Verlag, Berlin, 1997.

[27] C. Wiuf and J. Hein. Recombination as a point process along sequences. *Theoretical Population Biology*, 55:248–259, 1999.

[28] N. Kaplan and R. R. Hudson. The use of sample genealogies for studying a selectively neutral m-loci model with recombination. *Theoretical Population Biology*, 28:382–396, 1985.

[29] G. B. Golding. The sampling distribution of linkage disequilibrium. *Genetics*, 108:257–274, 1984.

[30] S. N. Ethier and R. C. Griffiths. On the two-locus sampling distribution. *Journal of Mathematical Biology*, 29:131–159, 1990.

[31] R. C. Griffiths. The two-locus ancestral graph. In I. V. Basawa and R. L. Taylor, editors, *Selected proceedings of the Sheffield symposium on applied probability: 18. IMS Lecture Notes—Monograph series*, volume 18, pages 100–117, 1991.

[32] R. C. Griffiths, P. A. Jenkins, and Y. S. Song. Importance sampling and the two-locus model with subdivided population structure. *Advances in Applied Probability*, 40(2):473–500, 2008.

[33] M. Stephens. Inference under the coalescent. In D. Balding, M. Bishop, and C. Cannings, editors, *Handbook of Statistical Genetics*, chapter 8. Wiley, Chichester, UK, 2001.

[34] J. Felsenstein. Evolutionary trees from DNA sequences: a maximum likelihood approach. *Journal of molecular evolution*, 17:368–376, 1981.

[35] J. S. Liu. *Monte Carlo strategies in scientific computing*. Springer-Verlag, New York, 2001.

[36] P. Fearnhead and P. Donnelly. Estimating recombination rates from population genetic data. *Genetics*, 159:1299–1318, 2001.

[37] M. Stephens and P. Donnelly. Inference in molecular population genetics. *Journal of the Royal Statistical Society: Series B*, 62:605–655, 2000.

[38] J. Felsenstein, M. K. Kuhner, J. Yamato, and P. Beerli. Likelihoods on coalescents: a Monte Carlo sampling approach to inferring parameters from population samples of molecular data. In F. Seillier-Moiseiwitsch, editor, *Statistics in molecular biology and genetics*, volume 33 of *IMS lecture notes—monograph series*, pages 163–185. Institute of Mathematical Statistics and American Mathematical Society, Hayward, CA, 1999.

[39] M. De Iorio and R. C. Griffiths. Importance sampling on coalescent histories I. *Advances in Applied Probability*, 36:417–433, 2004.

[40] R. C. Griffiths and P. Marjoram. Ancestral inference from samples of DNA sequences with recombination. *Journal of Computational Biology*, 3(4):479–502, 1996.

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

[42] M. Birkner and J. Blath. Computing likelihoods for coalescents with multiple collisions in the infinitely many sites model. *Journal of Mathematical Biology*, 57:435–465, 2008.

[43] P. J. Munday. *Importance sampling in spatial epidemics*. PhD thesis, University of Oxford, In preparation.

[44] A. Hobolth, M. Uyenoyama, and C. Wiuf. Importance sampling for the infinite sites model. Pre-print, 7 March 2008.

[45] C. Wiuf and P. Donnelly. Conditional genealogies and the age of a neutral mutant. *Theoretical Population Biology*, 56:183–201, 1999.

[46] I. J. Wilson and D. J. Balding. Genealogical inference from microsatellite data. *Genetics*, 150:499–510, 1998.

[47] R. Nielsen. Estimation of population parameters and recombination rates from single nucleotide polymorphisms. *Genetics*, 154:931–942, 2000.

[48] Y. S. Song, R. Lyngsø, and J. Hein. Counting all possible ancestral configurations of sample sequences in population genetics. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 3(3):239–251, 2006.

[49] R. B. Lyngsø, Y. S. Song, and J. Hein. Accurate computation of likelihoods in the coalescent with recombination via parsimony. In M. Vingron and L. Wong, editors, *Research in Computational Molecular Biology*, pages 463–477. Springer Berlin/Heidelberg, 2008.

[50] N. Li and M. Stephens. Modeling linkage disequilibrium and identifying recombination hotspots using single-nucleotide polymorphism data. *Genetics*, 165:2213–2233, 2003.

[51] W. J. Ewens. The sampling theory of selectively neutral alleles. *Theoretical Population Biology*, 3:87–112, 1972.

[52] M. Stephens and P. Scheet. Accounting for decay of linkage disequilibrium in haplotype inference and missing-data imputation. *American Journal of Human Genetics*, 76:449–462, 2005.

[53] N. Cardin. *Approximating the coalescent with recombination.* PhD thesis, University of Oxford, 2006.

[54] C. Varin. On composite marginal likelihoods. *Advances in Statistical Analysis*, 92(1):1–28, 2008.

[55] R. R. Hudson. Two-locus sampling distributions and their application. *Genetics*, 159:1805–1817, 2001.

[56] G. McVean, P. Awadalla, and P. Fearnhead. A coalescent-based method for detecting and estimating recombination from gene sequences. *Genetics*, 160:1231–1241, 2002.

[57] G. A. T. McVean, S. R. Myers, S. Hunt, P. Deloukas, D. R. Bentley, and P. Donnelly. The fine-scale structure of recombination rate variation in the human genome. *Science*, 304:581–584, 2004.

[58] S. Myers, L. Bottolo, C. Freeman, G. McVean, and P. Donnelly. A fine-scale map of recombination rates and hotspots across the human genome. *Science*, 310:321–324, 2005.

[59] P. Fearnhead and P. Donnelly. Approximate likelihood methods for estimating local recombination rates. *Journal of the Royal Statistical Society: Series B*, 64:657–680, 2002.

[60] P. Fearnhead, R. M. Harding, J. A. Schneider, S. Myers, and P. Donnelly. Application of coalescent methods to reveal fine-scale rate variation and recombination hotspots. *Genetics*, 167:2067–2081, 2004.

[61] P. Fearnhead and N. G. C. Smith. A novel method with improved power to detect recombination hotspots from polymorphism data reveals multiple hotspots in human genes. *American Journal of Human Genetics*, 77:781–794, 2005.

[62] M. Kimura. Diffusion models in population genetics. *Journal of Applied Probability*, 1:177–232, 1964.

[63] M. De Iorio and R. C. Griffiths. Importance sampling on coalescent histories II. *Advances in Applied Probability*, 36:434–454, 2004.

[64] The International HapMap Consortium. A second generation human haplotype map of over 3.1 million SNPs. *Nature*, 449(7164):851–861, 2007.

[65] R. R. Hudson. Generating samples under a Wright-Fisher neutral model of genetic variation. *Bioinformatics*, 18:337–338, 2002.

[66] G. A. T. McVean and N. J. Cardin. Approximating the coalescent with recombination. *Philosophical Transactions of the Royal Society B*, 360:1387–1393, 2005.

[67] C. Wiuf and D. Posada. A coalescent model of recombination hotspots. *Genetics*, 164:407–417, 2003.

[68] K. L. Simonsen and G. A. Churchill. A Markov chain model of coalescence with recombination. *Theoretical Population Biology*, 52:43–59, 1997.

[69] G. Marsaglia and A. Zaman. A new class of random number generators. *The Annals of Applied Probability*, 1(3):462–480, 1991.

[70] A. Doucet, J. F. G. de Freitas, and N. J. Gordon, editors. *Sequential Monte Carlo methods in practice*. Springer, New York, 2001.

[71] A. Kong, J. S. Liu, and W. H. Hong. Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association*, 89(425):278–288, 1994.

[72] J. S. Liu and R. Chen. Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association*, 98(443):1032–1044, 1998.

[73] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F. Radar and Signal Processing*, 140(2):107–113, 1993.

[74] G. Kitagawa. Monte Carlo filter and smoother for non-Gaussian state space models. *Journal of Computational and Graphical Statistics*, 5(1):1–25, 1996.

[75] D. B. Rubin. A noniterative sampling/importance resampling alternative to the data augmentation algorithm for creating a few imputations when fractions of missing information are modest: the SIR algorithm. *Journal of the American Statistical Association*, 82(398):543–546, 1987.

[76] Y. Chen, J. Xie, and J. S. Liu. Stopping-time resampling for sequential Monte Carlo methods. *Journal of the Royal Statistical Society: Series B*, 67:199–217, 2005.

[77] Y. Chen and J. S. Liu. Discussion on "Inference in molecular population genetics" by M. Stephens and P. Donnelly. *Journal of the Royal Statistical Society: Series B*, 62:644–645, 2000.

[78] J. L. Zhang and J. S. Liu. A new sequential importance sampling method and its application to the two-dimensional hydrophobic-hydrophilic model. *Journal of Chemical Physics*, 117(7):3492–3498, 2002.

[79] F. Larribe. *Cartographie génétique fine par le graphe de recombinaison ancestral*. PhD thesis, University of Montréal, 2003.

[80] R. Thomson. *The shape of a coalescent tree*. PhD thesis, Monash University, 1998.

[81] Y. S. Song and J. Hein. Parsimonious reconstruction of sequence evolution and haplotype blocks: Finding the minimum number of recombination events. In G. Benson and R. Page, editors, *Algorithms in Bioinformatics*, pages 287–302. Springer-Verlag, Berlin, 2003.

[82] Y. S. Song and J. Hein. Constructing minimal ancestral recombination graphs. *Journal of Computational Biology*, 12(2):147–169, 2005.

[83] R. B. Lyngsø, Y. S. Song, and J. Hein. Minimum recombination histories by branch and bound. In R. Casadio and G. Myers, editors, *Algorithms in Bioinformatics*, pages 239–250. Springer Berlin/Heidelberg, 2005.

[84] J. M. Hammersley and K. W. Morton. Poor man's Monte Carlo. *Journal of the Royal Statistical Society: Series B*, 16:23–38, 1954.

[85] M. N. Rosenbluth and A. W. Rosenbluth. Monte Carlo calculation of the average extension of molecular chains. *Journal of Chemical Physics*, 23(2):356–359, 1955.

[86] F. T. Wall and J. J. Erpenbeck. New method for the statistical computation of polymer dimensions. *Journal of Chemical Physics*, 30(3):634–637, 1959.

[87] H. Meirovitch. A new method for simulation of real chains: scanning future steps. *Journal of Physics A: Mathematical and General*, 15(12):L735–L741, 1982.

[88] R. M. Harding, S. M. Fullerton, R. C. Griffiths, J. Bond, M. J. Cox, J. A. Schneider, D. S. Moulin, and J. B. Clegg. Archaic African *and* Asian lineages in the genetic ancestry of modern humans. *American Journal of Human Genetics*, 60:772–789, 1997.

[89] C. Wiuf and J. Hein. The ancestry of a sample of sequences subject to recombination. *Genetics*, 151:1217–1228, 1999.

[90] SeattleSNPs. NHLBI program for Genomic Applications, SeattleSNPs, Seattle, WA (URL: `http://pga.gs.washington.edu`) [July 2008].

[91] D. A. Hinds, L. L. Stuve, G. B. Nilsen, E. Halprein, E. Eskin, D. G. Ballinger, K. A. Frazer, and D. R. Cox. Whole-genome patterns of common DNA variation in three human populations. *Science*, 307:1072–1079, 2005.

[92] B. Gold, J. E. Merriam, J. Zernant, L. S. Hancox, A. J. Taiber, K. Gehrs, K. Cramer, J. Neel, J. Bergeron, G. R. Barile, R. T. Smith, the AMD Genetics Clinical Study Group, G. S. Hageman, M. Dean, and R. Allikmets. Variation in factor B (*BF*) and complement component 2 (*C2*) genes is associated with age-related macular degeneration. *Nature Genetics*, 38(4):458–462, 2006.

[93] J. M. Akey, M. A. Eberle, M. J. Rieder, C. S. Carlson, M. D. Shriver, D. A. Nickerson, and L. Kruglyak. Population history and natural selection shape patterns of genetic variation in 132 genes. *PLoS Biology*, 2(10):e286, 2004.

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

[95] Z. Zhang, S. Schwartz, L. Wagner, and W. Miller. A greedy algorithm for aligning DNA sequences. *Journal of Computational Biology*, 7(1–2):203–214, 2000.

[96] URL: `http://www.ncbi.nlm.nih.gov/BLAST` [July 2008].

[97] R. C. Griffiths and S. Tavaré. The age of a mutation in a general coalescent tree. *Stochastic Models*, 14:273–195, 1998.

[98] J. Lenhard. *Kritische Untersuchung einer Methode zur Schätzung Phylogenetischer Größen.* PhD thesis, Johann Wolfgang Goethe University, 1997.

[99] M. W. Nachman and S. L. Crowell. Estimate of the mutation rate per nucleotide in humans. *Genetics*, 156:297–304, 2000.

[100] M. Stephens. Times on trees, and the age of an allele. *Theoretical Population Biology*, 57:109–119, 2000.