Bayesian inference in molecular phylogeography using Markov chain Monte Carlo

Yuttapong Thawornwattana

2018

Thesis submitted to the University College London

for the degree of Master of Philosophy

I, Yuttapong Thawornwattana, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Chapters 3 and 4 involve two projects that are joint work with Ziheng Yang (ZY) and Daniel Dalquen (DD).

- Chapter 3 and parts of Chapter 1 are adapted from a published paper: Thawornwattana, Y., Dalquen, D., and Yang, Z. (2018). Designing simple and efficient Markov chain Monte Carlo proposal kernels. *Bayesian Anal.*, 13(4):1033–1059. doi:10.1214/17-BA1084.
- Chapter 4 is adapted from an accepted manuscript: Thawornwattana, Y., Dalquen, D., and Yang, Z. (2018). Coalescent analysis of phylogenomic data confidently resolves the species relationships in the *Anopheles gambiae* species complex. *Mol. Biol. Evol.* https://doi.org/10.1093/molbev/msy158.

For both projects, ZY designed the study. DD performed initial data analysis. I performed subsequent analysis and wrote a draft manuscript. All authors edited the manuscript for Chapter 3. I and ZY edited the manuscript for Chapter 4.

Abstract

The Bayesian approach to phylogenetic inference allows quantification of all aspects of uncertainty using probability. Markov chain Monte Carlo (MCMC), a class of algorithms based on iterative simulation, is often considered a gold standard for approximate Bayesian inference. However, MCMC is computationally intensive and there are many design decisions to be made when using it in practice. We discuss few principles for designing simple and efficient MCMC algorithms. In particular, we propose several new proposal kernels for MCMC based on the idea of introducing negative correlations in the simulation draws. In many cases, these kernels can lead to efficiency >100%. Using practical examples, we illustrate that a sequence of well-designed one-dimensional proposals can be more efficient than a single d-dimensional proposal, and that variable transformations can be used as a general strategy for designing efficient MCMC. Next, we turn to the problem of species tree inference in the *Anopheles gambiae* species complex from whole-genome data. This is a challenging problem due to complex effects of recent and rapid radiation, introgression, chromosome inversions and natural selection. We extract over 80,000 coding and noncoding loci from the genomes of six members of this species complex and perform Bayesian inference using MCMC under the multispecies coalescent model, which takes into account genealogical heterogeneity across the genome and uncertainty in the gene trees. We obtain a robust species tree estimate, consistent with chromosome inversions. Using simulation informed by the real data, we conclude that species trees from previous studies are erroneous as a result of methodological artefacts. We also found evidence of gene flow between certain pairs of species based on direct estimation of migration rates under the isolation-with-migration model. The results highlight the importance of accommodating incomplete lineage sorting and introgression in phylogenomic analyses of species that arose through recent radiative speciation events.

Impact statement

The Bayesian approach to statistical data analysis is being used to answer a wide variety of problems in science and engineering. It provides a convenient way of formulating complex models of the observed data and comes with a natural way of quantifying uncertainty of quantities in the model using probability. However, computation remains a major limitation in practical applications. In the first part of this thesis, we illustrate how several general principles can be used to design simple and efficient simulation algorithms for Bayesian inference. In many examples considered, an increase in efficiency can be substantial compared with commonly used algorithms and many state-of-the-art algorithms. Higher efficiency means shorter running time required to achieve a desired level of accuracy. Thus using more efficient algorithms can considerably speed up an analysis pipeline. This is particularly important in applications involving large and complex models such as those in phylogenomics, where a single analysis can take weeks or months to run.

In the second part of the thesis, we perform Bayesian analysis of the species phylogeny of the *Anopheles gambiae* species complex, a group of mosquitoes that are principal vectors for human malaria in Africa. Malaria remains a major public health issue in Africa, accounting for 90% of the malaria cases and 91% of all malaria deaths worldwide in 2016 (World Health Organization, 2017). Here, we robustly infer the evolutionary history of this species complex, including species branching orders and divergent times as well as introgression events between species. Unlike previous work, our approach takes into account genealogical heterogeneity across the genome in a single probabilistic model. The knowledge of the species phylogeny forms a foundation for studying the evolution and epidemiology of medically important traits such as vectorial capacity and insecticide resistance. A better understanding of the ecological and evolutionary processes underlying these traits will ultimately lead to development of more effective strategies for malaria control and elimination.

Acknowledgments

It has been a privilege to be supervised by Ziheng Yang. I have benefited immensely from his expertise, ideas and advice. Fulfilling the intellectual standard he sets has been demanding but also rewarding.

I was fortunate to receive constant academic and social support as well as good humour from several individuals, in particular, members of Yang, Telford and Paola groups, who also made my time at UCL memorable. Special thanks go to many of my officemates for putting up with me over the years: Daniel Dalquen, Anne Zakrzewski, Kostas Angelis, Jose Barba-Montoya, Jiao, Johannes Girstmair and Steven Müller. Mario dos Reis has been an invaluable source of advice since my undergraduate days. I benefited a lot from this enthusiasm and encouragement. Thanks to my examiners Carolin Kosiol and Garrett Hellenthal for reviewing this work, making insightful comments and catching some mistakes. I am solely responsible for any errors that remain. I also owe a great deal to many wonderful friends at UCL and my parents without whom I might have not made it this far.

Finally, I am grateful to the Development and Promotion of Science and Technology Talents Project (DPST) for their long-term investment in funding my study for many years.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction to Bayesian data analysis and Markov chain Monte Carlo methods

1.1 Bayesian approach to statistical data analysis

Bayesian data analysis provides a powerful statistical framework for addressing scientific questions of interest by combining data with the scientific knowledge about the problem. All the unknowns and the observed data are expressed in terms of probability distributions, which naturally provide a measure of uncertainty. Bayesian data analysis is an iterative process consisting of following three steps (Box, 1980; Gelman et al., 2014).

1. **Model building**. First, a probabilistic model is constructed to provide a mechanism for generating the observed data based on the scientific knowledge. This is a probability distribution $p(y|\theta)$ of the observed data y conditioned on the parameters θ in the generative process we wish to learn about from the data, called the *likelihood* or *data model*. The parameters are also given a probability distribution $p(\theta)$, called the *prior distribution*. These two components give the joint probability distribution $p(y, \theta) = p(y|\theta)p(\theta)$, which we simply refer to as the *model*. Using probability distributions to describe parameters allow direct uncertainty quantification of quantities of interest in all aspects.

2. **Posterior inference**. Second, computation is performed to produce a probability distribution of the model parameters given the observed data, $p(\theta|y)$, called the *posterior distribution*, using the relationship

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)}. \tag{1.1}$$

   We elaborate on inference computation in more detail in Section 1.2.

3. **Model checking**. Third, the fitted model is evaluated and improved. Assessing and criticising the validity of the model assumptions are important parts of scientific investigations, with direct

implications on the conclusions drawn from the inference. For instance, sensitivity to the prior specification and other modelling assumptions can be used to assess the robustness of inferred posterior quantities. In *posterior predictive checking*, similarity of summary statistics of data simulated from the fitted model and the real data is used to indicate how well the model fits the data (Gelman et al., 1996a).

This three-step iterative process allows systematic improvements of the model. For example, parts of the model may be revised or extended to address any inadequacies identified, making it more realistic. New data may be added to the model. The model may also be simplified to ease the computation. This process is in line with the scientific method where hypotheses can be refined or falsified based on new observations and knowledge (Gelman and Shalizi, 2013).

As an example, in Chapter 4, we are interested in inferring a species tree that represents the evolutionary history of a group of mosquito species using whole genome sequences. The first step in Bayesian analysis is to build a joint model of the sequence data and species tree using our knowledge about how genomic sequences evolve. Second, the posterior distribution of the species tree is computed using a numerical algorithm. Last, two main approaches of model checking are employed: (1) the robustness of the species tree estimate is assessed by repeating the inference using a different prior specification or a different data preprocessing pipeline, and (2) how well the model fits the data is assessed by performing simulation studies using parameter values from the fitted model.

One major challenge of Bayesian data analysis is the computation of posterior quantities of interest, particularly for complex models and/or large datasets, as we shall see in Chapter 4. In fact, Chapter 3 is devoted to improving efficiency of the posterior computation.

1.2 Inference computation

Inference computation in Bayesian statistics is largely concerned with the following quantities: (1) posterior distribution $p(\theta|y)$, as well as its moments and marginal distributions, (2) *prior predictive distribution* $p(y) = \int p(y, \theta)\, d\theta$ (also called *marginal likelihood*), which is useful for comparing different data models, (3) *posterior predictive distribution* $p(\tilde{y}|y) = \int p(\tilde{y}|\theta, y)p(\theta|y)\, d\theta = \int p(\tilde{y}|\theta)p(\theta|y)\, d\theta$. Except for simple models, these quantities are computationally intractable and have to be approximated.

In this thesis, we primarily focus on computing posterior quantities such as mean and variance, which are integrals with respect to the posterior distribution, and not on estimating the entire posterior distribution. It is important to note that this integral estimation problem is typically an easier problem than estimating the entire distribution, also known as the density estimation problem.

Numerical integration methods can be used to approximate posterior integrals. Deterministic approximations such as quadrature rules work well only for low-dimensional integrals, with assumptions on the

smoothness of the integrand. For high-dimensional integrals, a class of randomised algorithms known as *Monte Carlo algorithms* is often used. This is the topic of the next section.

1.3 Monte Carlo algorithms

1.3.1 Monte Carlo integration

We review how an integral with respect to a probability distribution of interest, such as its mean and variance, can be estimated using only samples generated from this distribution, which will be referred to as the *target distribution*, or simply the *target*. We start with an estimator that uses independent samples from the target distribution, referred to as a *simple Monte Carlo estimator*. Let π be the target distribution with support $X \subset \mathbf{R}^d$. In Bayesian inference, π will often be the posterior distribution. Let $f : X \to \mathbf{R}$ be an absolutely integrable function and let

$$\pi(f) \coloneqq \mathbf{E}_\pi f(x) = \int_X \pi(x) f(x)\, dx \tag{1.2}$$

denote the expected value of f with respect to π. Given samples $x_{1:N}$ from π, Monte Carlo integration gives an unbiased approximation of $\pi(f)$ by

$$\hat{\pi}_N(f) \coloneqq \frac{1}{N} \sum_{n=1}^{N} f(x_n). \tag{1.3}$$

Provided $\pi(f) < \infty$, the estimator $\hat{\pi}_N(f)$ converges almost surely to $\pi(f)$ as $N \to \infty$ by the Strong Law of Large Numbers. This estimator is unbiased since $\mathbf{E}\hat{\pi}_N(f) = \mathbf{E}\pi(f)$, with variance

$$\mathrm{Var}(\hat{\pi}_N(f)) = \frac{1}{N} V_f, \tag{1.4}$$

where $V_f \coloneqq \mathrm{Var}_\pi(f(x)) = \mathbf{E}_\pi(f(x) - \mathbf{E}_\pi f(x))^2$ is the variance of f under π, provided f is square integrable, i.e. $\pi(f^2) < \infty$. Thus the root mean square error (RMSE)[1] of $\hat{\pi}_N(f)$ is simply its standard deviation $\sqrt{V_f/N} = O_{N\to\infty}(N^{-1/2})$. This means that an extra digit of accuracy in the estimate requires about 100 times more samples. Moreover, when $\pi(f^2) < \infty$ holds, we know from the Central Limit Theorem (CLT) that the *Monte Carlo error* $\hat{\pi}_N(f) - \pi(f)$ converges in distribution to the normal distribution $N(0, \sqrt{V_f/N})$. When $d = 1$ (univariate target), this leads to an approximate $100(1-\alpha)\%$ two-sided confidence interval $\hat{\pi}_N(f) \pm \Phi^{-1}(1-\alpha/2)\sqrt{\frac{\hat{V}_f}{N}}$, where $\alpha \in (0,1)$ and $\hat{V}_f$ is an estimate of V_f such as $\hat{V}_f = \frac{1}{N-1}\sum_{n=1}^{N}(f(x_n) - \hat{\pi}_N(f))^2$, and Φ is the distribution function of the standard normal distribution $N(0,1)$. In this case, the

[1] Also called the *root mean quadratic error*, it is given by $\sqrt{\mathbf{E}\,|\hat{\pi}_N(f) - \pi(f)|^2}$, with

$$\mathbf{E}\,|\hat{\pi}_N(f) - \pi(f)|^2 = |\mathbf{E}\hat{\pi}_N(f) - \pi(f)|^2 + \mathrm{Var}(\hat{\pi}_N(f)),$$

where the first term on the right hand side is the bias squared and the second term is the variance of $\hat{\pi}_N(f)$, and the expectations are taken with respect to π.

standard error of the Monte Carlo estimate, which is the standard deviation of the asymptotic distribution of the Monte Carlo error, coincides with the RMSE. Compared with numerical quadrature methods such as trapezoid rule or Simpson rule, this Monte Carlo standard error $O_{N\to\infty}(N^{-1/2})$ decays at a much slower rate[2]. But unlike those quadrature rules, Monte Carlo error does not get worse as the target dimension d increases. In addition, numerical quadrature methods often require certain smoothness assumptions on f, whereas Monte Carlo integration does not. Hence Monte Carlo integration can be competitive as a generic method for estimating integrals, particularly for high-dimensional ones, which are common in modern applications.

1.3.2 Sampling methods

But how do we obtain samples $x_{1:N}$ from the target distribution π to use in the Monte Carlo estimator $\hat{\pi}_N(f)$ (1.3)? There are recipes for sampling from a wide range of standard distributions and processes, see e.g. Devroye (1986). In principle, any scalar random variate on the reals $\mathbf{R}$ can be generated from uniform random variates by inverting its distribution function. This method is referred to as *inverse transform sampling* (Algorithm 1.1). The idea also works for multivariate and discrete distributions.

Algorithm 1.1 Inverse transform sampling

Suppose x has distribution function F with inverse $F^{-1}(u) \coloneqq \inf\{x \in \mathbf{R} : F(x) \geq u\}$ for $u \in (0, 1)$.
Return: a random variate x with distribution function F

1. draw $u \sim U(0, 1)$
2. set $x \leftarrow F^{-1}(u)$

In practice, however, the transformation used in the inverse transform sampling is not always available. An alternative approach is to sample from another distribution q and transform the samples into those from the desired distribution π. In *rejection sampling* (von Neumann, 1951), points are generated uniformly under a curve proportional to q and are accepted if they fall under a curve proportional to π. The accepted points are samples from π. A good sampling distribution q should be proportional to π to keep the rejection rate small. It is used in many algorithms for generating non-uniform random variates, particularly from truncated distributions.

Instead of rejecting samples, *importance sampling* (Hammersley and Morton, 1954; Rosenbluth and Rosenbluth, 1955) gives a weight to each sample, and all weighted samples are used to compute the Monte Carlo estimate of (1.2), based on the identity $\int_X \pi(x)f(x)\,dx = \int_X \frac{\pi(x)}{q(x)}q(x)f(x)\,dx$. To obtain independent draws from π, resampling can be performed on the weighted samples (x_i, w_i) where $x_i \sim q$ and $w_i \coloneqq \frac{\pi(x_i)}{q(x_i)}$. Importance sampling has many uses in Bayesian computation such as improving an analytic posterior approximation, providing starting points for iterative simulation (Section 1.3.3), estimating out-of-sample

[2] $O(N^{-3})$ for trapezoid rule and $O(N^{-4})$ for Simpson rule as $N \to \infty$.

prediction accuracy via leave-one-out cross-validation as well as forming a basis for a large class of Monte Carlo algorithms called *sequential Monte Carlo (SMC)*.

The main drawback of both rejection sampling and importance sampling is that they do not scale well to high-dimensional targets. In rejection sampling, the rejection rate can become impractically high. In importance sampling, although all samples are used, they can be unreliable, and the variance of the estimator can become arbitrary large or even infinite. One way to improve the estimator is to generate samples sequentially and learn from the past samples. This idea leads to the use of Markov chain simulation to explore the target distribution π, where the location of the next sample depends on the current sample.

1.3.3 Markov chain Monte Carlo (MCMC)

1.3.3.1 Markov chains

Markov chain Monte Carlo (MCMC) is a class of iterative simulation algorithms that provides a generic construction of a Markov chain whose stationary distribution matches the target distribution of interest, and generates approximate[3] samples that can be used for density estimation and Monte Carlo integration. An introduction to MCMC methods can be found in Andrieu et al. (2003); Craiu and Rosenthal (2014). Roberts and Rosenthal (2004) and Diaconis (2009) provide more analytical backgrounds. Book-length treatments are in Robert and Casella (2004); Liu (2008); Brooks et al. (2011). Robert and Casella (2011) give a historical perspective on the development of MCMC.

To simulate a Markov chain[4] $(x_t)_{t\geq 0}$ with stationary distribution π on X, we first initialise the chain $x_0 \sim \nu$ from an initial distribution ν. Then for $t \geq 1$, x_t is generated from a *transition kernel* $P : x \mapsto P_x$, which gives a probability distribution on X conditioned on the previous sample x_{t-1}, denoted $P_{x_{t-1}}$, with probability density $p(x_t|x_{t-1})$. For correctness of the simulation, we require that P is *stationary* (or *invariant*) with respect to π, meaning that *global balance* condition $\pi P = \pi$ is satisfied, where $\pi P := \int_X P_x \, d\pi(x)$, or in terms of density,

$$\int_X p(y|x)\pi(x)\,dx = \pi(y). \tag{1.5}$$

This condition says that the conditional distribution P_x at any point x is, on average, the stationary distribution π. In practice, simulating from a transition kernel is done via an appropriate deterministic function $\phi : X \times U \to X$ such that $x_t = \phi(x_{t-1}, u_t)$, where u_t is a collection of uniform random variates providing a source of randomness. Next, we introduce the notion of a reverse transition kernel and reversibility of a Markov chain. For each transition kernel $P : x \to P_x$, we can define a *reverse transition kernel* $R : y \mapsto R_y$,

[3] Except for when exact sampling is possible (see e.g. Huber (2016)), finite samples from a Markov chain are only approximately from the target distribution, and exact samples are only achieved at the infinite sample size limit, which is not practical.

[4] We focus exclusively on *first-order discrete-time homogeneous* Markov chains on $X \subset \mathbf{R}^d$, which are the most commonly used class of Markov chains for MCMC. The use of continuous-time Markov chains in MCMC is possible, see e.g. Bierkens et al. (2016); Bouchard-Côté et al. (2018).

with density $r(\cdot|y)$, such that the *detailed balance* (or *local balance*) condition $dP_x(y)\,d\pi(x) = dR_y(x)\,d\pi(y)$ is satisfied. In terms of density, this is

$$p(y|x)\pi(x) = r(x|y)\pi(y). \tag{1.6}$$

In the special case when P and R are equivalent, the chain is said to be *reversible*. This is a strictly stronger notion than stationarity, but is easier to verify since it only involves a pair of states instead of the entire state space.

We now move on to convergence of Markov chains, which implies the asymptotic correctness of the simulation. For a Markov chain to converge to a specified target distribution π, (1) it must have a unique stationary distribution and (2) this distribution must be π. First, for a Markov chain with transition kernel P to converge to a unique stationary distribution ν, P must satisfy certain conditions[5] which almost always hold in practical applications. Note that if ν is a stationary distribution, then it satisfies the global balance condition (1.5). Second, to ensure that this stationary distribution ν matches the desired target distribution π, many recipes are available. Popular constructions include Metropolis–Hastings (MH) algorithm (Metropolis et al., 1953; Hastings, 1970) and Gibbs sampler (Geman and Geman, 1984). The MH algorithm is generic and the resulting Markov chain is reversible, and is thus stationary. The Gibbs sampler requires simulation from full conditional distributions, and the resulting Markov chain needs not be reversible, depending on the sampling order of the target variables.

Example 1.1 (MH algorithm)**.** The Metropolis–Hastings (MH) algorithm is a popular MCMC algorithm that also forms a basis for many advanced MCMC samplers. Given the chain is currently at a state x, a potential next state x' is generated from a *proposal kernel* Q on X, with density $q(x'|x)$. The chain then moves to x' with probability

$$\alpha(x, x') \coloneqq \min\left(1, \frac{\pi(x')}{\pi(x)}\frac{q(x|x')}{q(x'|x)}\right). \tag{1.7}$$

Otherwise it stays at x. The resulting Markov chain has the transition kernel P with density

$$p(x'|x) = \begin{cases} q(x'|x)\alpha(x, x') & \text{if } x' \neq x, \\ 1 - \int_X q(x'|x)\alpha(x, x')\,dx & \text{if } x' = x. \end{cases}$$

By construction, this Markov chain is reversible with respect to π, with $\pi(x)p(x'|x) = \pi(x')p(x|x')$ for almost every $x, x' \in X$. If the proposal kernel Q is irreducible and aperiodic, the transition kernel P will also be irreducible, aperiodic and is π-invariant. Notice that the target density appears as a *ratio* in the acceptance probability $\alpha(x, x')$. This makes the algorithm applicable when π is only known up to a

[5] Specifically, P must be *irreducible* (meaning that the chain can move from any state to any other in a finite number of steps) and *aperiodic* (meaning that there are no regions of the state space that can only be visited at a regular time interval; this is to avoid deterministically repeating a sequence of states as cycles of length greater than one); see e.g. Theorem 1 in Tierney (1994) or Theorem 4 in Roberts and Rosenthal (2004).

Algorithm 1.2 Metropolis-Hastings (MH) algorithm

Inputs: number of samples N, number of burn-in iterations N_0, initial distribution ν_0, proposal kernel Q
Returns: samples $x_{1:N}$ approximately from π

1. initialise $x_0 \sim \nu_0$
2. for $n = 1, \ldots, N_0 + N$
   a) draw $x' \sim Q_{x_{n-1}}$
   b) compute acceptance ratio $a = \frac{\pi(x')q(x_{n-1}|x')}{\pi(x_{n-1})q(x'|x_{n-1})}$
   c) if $a > 1$ or $u < a$ where $u \sim U(0, 1)$, then set $x_n = x'$
   otherwise set $x_n = x_{n-1}$
   d) if $n > N_0$, record x_n

Note: Step 2c sets $x_n = x'$ with probability $\min(1, a(x_{n-1}, x'))$

normalising constant, which is usually the case for posterior distributions (1.1).

Several proposal kernels can be combined in two main ways to form a new kernel (Tierney, 1994). A *mixture of kernels* can be used to combine proposals with different features to improve the mixing property of the chain (Guan and Krone, 2007), and is particularly popular when the target is a mixture distribution (Holden et al., 2009; Bai et al., 2011) or consists of components with different dimensions (called trans-dimensional problems) (Green, 1995). A *cycle of kernels* applies a sequence of kernels in turn. A special case is when each kernel only samples a subset of variables of a multivariate target. This strategy is useful, for instance, when a multivariate state can be grouped into blocks of variables, and each block is sampled separately. Grouping highly correlated variables into the same block can improve mixing and convergence properties of the algorithm (Roberts and Sahu, 1997). An example of using cycle of kernels is Gibbs sampling, where each kernel is a conditional distribution. In Section 3.2, we empirically demonstrate that for a multivariate target, applying a cycle of low-dimensional random walk kernels can be more efficient than using a single multivariate kernel.

Example 1.2 (Gibbs sampler)**.** For a multivariate target, Gibbs sampler alternately draws samples from conditional distributions of a subset of the components given the rest of the variables. The sampling order can be deterministic or randomised. For a certain class of models such as directed acyclic graphs (DAGs), which include hierarchical regression models, the conditional distributions often have a standard form, so sampling can be done exactly. When direct sampling from the conditional is not possible, any MH kernel may be used.

Although there is no tuning parameter in the Gibbs kernels as in the random walk MH (Section 1.3.4.1), parameterisation of the model can greatly affect the efficiency. Various reparameterisation schemes can be applied to improve efficiency by reducing correlations among the components (Papaspiliopoulos et al., 2007; Yu and Meng, 2011; Xu et al., 2013).

1.3.3.2 Monte Carlo estimator using Markov chain samples

A simulated path of the Markov chain can be used to estimate an expectation under π (1.2) using a Monte Carlo estimator similar to (1.3). However, since the samples are only approximately from the target distribution π, the estimator is no longer unbiased. In practice, we discard samples from early iterations to reduce the bias. The estimator is thus

$$\hat{\pi}_N(f) \coloneqq \frac{1}{N} \sum_{n=N_0+1}^{N_0+N} f(x_n),$$

where N_0 is the waiting time for the process to 'get close' to the target distribution π, and N is the number of samples used to construct the estimator. The initial N_0 iterations are referred to as the *burn-in*, also known as the *warm-up* or *initial transient* phase of the simulation. The bias is expected to decrease as the burn-in time N_0 increases. This estimator is consistent in the sense that it converges to $\pi(f)$ almost surely as $N \to \infty$ (see e.g. Theorem 3 in Tierney (1994)). The variance of this estimator is more complex than that of the simple Monte Carlo estimator (1.4) due to autocorrelations in the Markov chain samples. Under certain ergodicity assumptions, the univariate Central Limit Theorem (CLT) holds for $\sqrt{N}\hat{\pi}(f)$ (see e.g. Theorems 4 and 5 in Tierney (1994), or Jones (2004)):

$$\sqrt{N}(\hat{\pi}_N(f) - \pi(f)) \xrightarrow{d} N(0, v) \tag{1.8}$$

as $N \to \infty$, where $\xrightarrow{d}$ denotes convergence in distribution, with

$$v \coloneqq \lim_{N\to\infty} N\mathrm{Var}(\hat{\pi}(f)) = V_f \tau, \tag{1.9}$$

where $V_f \coloneqq \mathrm{Var}_\pi(f(x))$ is the variance of $f(x)$ under π, $\rho_k \coloneqq \mathrm{Cor}(f(x_n), f(x_{n+k}))$ is the lag-k autocorrelation and $\tau \coloneqq 1 + 2\sum_{k=1}^{\infty} \rho_k$ is called the *integrated autocorrelation time*. This CLT result provides an asymptotic approximation of the variance of $\hat{\pi}_N(f)$ as $\frac{v}{N} = \frac{V_f \tau}{N}$. Thus increasing the number of post-burn-in iterations N reduces the variance of $\hat{\pi}_N(f)$. When the computer storage is limited and the running time is not an issue, it is common to 'thin' the chain by only recording one sample in every k iterations. This will reduce autocorrelations in the recorded samples but may increase the variance of the estimator. Finally, while some authors advocate the use of long burn-in such as $N_0 = N$ (Gelman et al., 2014), we prefer $N_0 \ll N$ to put more emphasis on the precision of the estimate since unbiasedness seems a less important property in practical applications.

1.3.3.3 Efficiency measures of the estimator

The presence of autocorrelations in the Markov chain samples makes the 'effective' sample size (ESS), denoted N_e, appear larger or smaller than what would be expected if the samples were uncorrelated.

Specifically, N_e is such that

$$\mathrm{Var}(\hat{\pi}_N(f)) = \frac{V_f}{N_e},$$

where the left hand side is asymptotically $\frac{V_f \tau}{N}$ based on (1.9), and the right hand side is the variance of the estimator using N_e uncorrelated samples (1.4). Thus

$$N_e = \frac{V_f}{V_f \tau / N} = \frac{N}{\tau}.$$

So we see that if the samples are uncorrelated, $\tau = 1$ and $N_e = N$, while positive autocorrelations tend to make $\tau > 1$ and $N_e < N$, and negative autocorrelations tend to make $\tau < 1$ and $N_e > N$. The ratio of the variance of $\hat{\pi}(f)$ for uncorrelated samples to the variance for MCMC samples of the same size, i.e.

$$E \coloneqq \frac{V_f / N}{v / N} = \frac{V_f}{v} = \frac{1}{\tau}, \tag{1.10}$$

can be used as a measure of statistical efficiency of the estimator $\hat{\pi}_N(f)$ (e.g. Gelman et al. (1996b)). From this expression, we see that $E = 1$ when there are no correlations, $E < 1$ when the samples are positively correlated and $E > 1$ when the samples are negatively correlated. This observation is exploited in Chapter 3 where we construct a 'super-efficient' estimator by directly injecting negative correlations into the Markov chain samples. In fact, the use of negative correlations is a common theme in reducing Monte Carlo error, known as *antithetic coupling* or *antithetic variates* (Hammersley and Morton, 1956).

In practice, the calculation of N_e and E involves the term $\sum_{k=1}^{\infty} \rho_k$ in τ, which needs to be approximated. Geyer (1992) provides several estimators for reversible Markov chains such as those from the MH algorithm. In this thesis, we use the so-called *initial positive sequence estimator* from Geyer (1992): given a simulation sequence $x_{1:N}$ from MCMC, $\sum_{k=1}^{\infty} \rho_k$ is estimated by $\sum_{k=1}^{K} \hat{\rho}_k$ where $\hat{\rho}_k$ is a sample-based estimate of the lag-k autocorrelation and K is the largest positive integer for which $\hat{\rho}_K + \hat{\rho}_{K+1} < 0$. Our experience (see Chapter 3) suggests that small sample sizes (say, $N < 10^5$) do not give reliable estimation, especially when E is small, which makes it harder to estimate. We typically use $N = 10^7$ to 10^8 after burn-in.

An alternative approach to estimate E is to use a closed form expression of v on a finite state space (Kemeny and Snell, 1960; Peskun, 1973):

$$v = f^\top B(2Z - I - A)f,$$

where $Z \coloneqq (I - P + A)^{-1}$, P is the transition probability matrix, $A \coloneqq (\pi(y))_{x,y \in X}$ (each row is π), and $f = (f_1, \ldots, f_{|X|})^\top$ with $f_x \coloneqq f(x)$ for $x \in X$. To apply this formula to a continuous state space $X \subset \mathbf{R}$, we truncate X to a finite interval $[x_L, x_U]$ for some pre-specified values of x_L and x_U, which is then divided into K bins of width $\Delta = \frac{x_U - x_L}{K}$. For each bin $k = 1, \ldots, K$, we use the midpoint $x_k \coloneqq x_L + (k - 1/2)\Delta$ as its representative. Then we compute the required matrices in the above expression on the

discretised space. Note that this method requires an analytic expression of the proposal density $q(x'|x)$ to compute P, whereas the positive sequence estimator described earlier only requires the proposal ratio $\frac{q(x|x')}{q(x'|x)}$ appearing in the MH acceptance probability (1.7). More details are described in Gelman et al. (1996b) and Yang and Rodríguez (2013).

Another commonly used efficiency measure is the *expected square jump distance (ESJD)*, also called *first-order efficiency*, defined as

$$E_\pi^2 := \mathbf{E}(f(x) - f(x'))^2 = 2(1 - \rho_1)\mathrm{Var}_\pi(f(x)),$$

where x and x' are consecutive elements of a stationary chain. Thus maximising E_π^2 is equivalent to minimising the first-order autocorrelation ρ_1. Both E_π^2 and ρ_1 can be directly estimated from the simulation draws. Note that E_π^2 depends on the target variance while ρ_1 does not, thus ρ_1 may be preferred as an efficiency measure.

While we will be focusing on the mixing of Markov chains at stationarity, as measured by E or ρ_1, we will also consider two measures of the convergence rate of Markov chains, namely the absolute value of the second largest eigenvalue of P, denoted $|\lambda|_2$, and the largest total variation distance to the target distribution after n steps among all possible starting points, denoted δ_n. These measures can be estimated by discretising the state space. Specifically, given the stationary distribution $\pi = (\pi_1, \ldots, \pi_K)$ and the probability transition matrix $P = (p_{ij})_{1 \le i,j \le K}$ on the discretised (i, j) state space, we calculate $|\lambda|_2 = \max_{i=2,\ldots,K} |\lambda_i|$, where $\lambda_1 \ge \lambda_2 \ge \ldots \ge \lambda_K$ are eigenvalues of P, and $\delta_n = \max_{i=1,\ldots,K} \sum_{j=1}^K |p_{ij}^n - \pi_j|$, where p_{ij}^n denotes the entry (i, j) of the matrix P^n. We used $n = 8$ in this thesis. From our experience (Figure 3.4), these convergence rate-based measures appear to be less useful for the purpose of constructing efficient Monte Carlo estimators (as oppose to density estimation); see also Yang and Rodríguez (2013).

1.3.4 Design decisions in the MH algorithm

1.3.4.1 Choice of the proposal kernel

For the MH algorithm (Example 1.1), we know that given a proposal kernel Q, the choice of the acceptance probability α in (1.7) is optimal in terms of minimising the asymptotic variance (1.9) of the estimator $\hat{\pi}(f)$; see Peskun (1973) for discrete state spaces and Tierney (1998) for continuous state spaces. However, what features the proposal kernel Q should have to minimise the asymptotic variance is not clear. The most common choice of Q is $x' = x + \sigma u$ where u has a Gaussian or uniform distribution with unit variance. This is referred to as the *(additive) random walk MH* algorithm, with a *step-size parameter* $\sigma > 0$ chosen by the user. Another commonly used proposal is the *Langevin proposal* $x' = x + \frac{\sigma^2}{2}\nabla_x \log \pi(x) + \sigma u$ with $u \sim N(0, I)$, which makes use of the target's gradient information to bias the proposal towards a

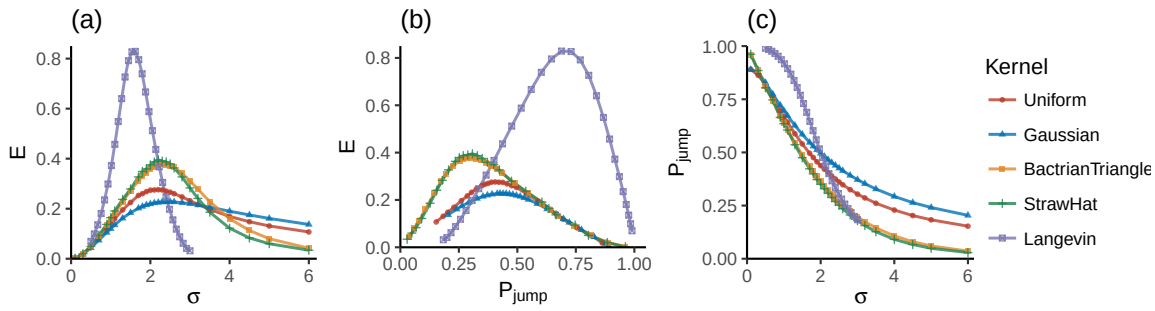


Figure 1.1: Efficiency as a function of the step-size parameter (σ) and the expected acceptance probability (P_{jump}) in the MH algorithm for the $N(0, 1)$ target using five different proposal kernels.

local mode of the target. The MH algorithm using the Langevin proposal is referred to as *Metropolis-adjusted Langevin algorithm (MALA)*. Although the Langevin kernel, as well as its related *Hamiltonian Monte Carlo (HMC)* kernel (Neal, 2011), is provably more efficient than the random-walk MH in terms of the asymptotic variance of the estimator when the step-size is chosen optimally (Figure 1.2, middle plot; Roberts and Rosenthal (1998)), its efficiency is considerably more sensitive to the choice of σ (Figure 1.1a). As a result, it is harder to tune σ to achieve good performance in practice. Moreover, the Langevin and HMC kernels also requires gradient computation, which may be not available analytically for complex models, or computationally intractable for large datasets.

It is well known that a poor choice of σ can adversely affect the mixing and convergence properties of the MH algorithm. Determining the optimal choice of the step-size parameter σ is an active area of research, known as *optimal scaling*. Gelman et al. (1996b) estimated an optimal σ that minimises v (1.9) for the Gaussian random walk MH kernel $q(x'|x) = N(x'|x, \sigma^2)$ for estimating the mean of the $N(0, 1)$ target to be about 2.38, with the corresponding expected acceptance probability

$$P_{\text{jump}} := \int_X \int_X \pi(x)\alpha(x, x')q(x'|x)\, dx'\, dx \tag{1.11}$$

to be about 0.44. When the target distribution π is d-dimensional with identically distributed components, Roberts et al. (1997) show that for the Gaussian kernel $q(x'|x) = N(x'|x, \sigma^2 I_d)$, the optimal step-size that minimises v is the one that leads to $P_{\text{jump}} \approx 0.234$ as $d \to \infty$ (Figure 1.2). Recent work covers more complex algorithms such as Multiple-try Metropolis (Bédard et al., 2012), delayed rejection MH (Bédard et al., 2014), HMC (Beskos et al., 2013), as well as the MH and Metropolis-adjusted Langevin algorithm (MALA) in the infinite-dimensional setting (Beskos et al., 2009; Pillai et al., 2012). However, even for the simple random walk MH algorithms, optimal scaling analysis has beed limited to the Gaussian random walk. A few exceptions are the Langevin proposal (Roberts and Rosenthal, 1998) and the Cauchy distribution (Neal and Roberts, 2011). Beyond this small collection of proposal kernels, there is no general theory available.

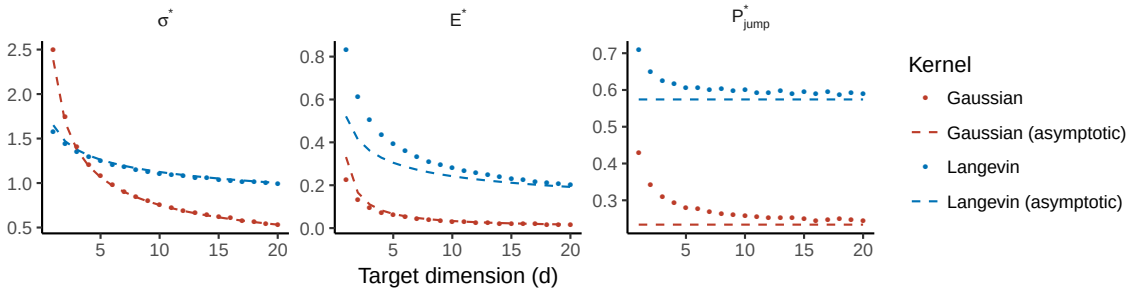


Figure 1.2: Optimal step-size σ^* and the corresponding efficiency E^* and expected acceptance probability P^*_{jump} for Gaussian and Langevin kernels as the dimension of the $N_d(0, I)$ target increases. The values were calculated using 10^7 MCMC samples, compared with the asymptotic values as $d \to \infty$ from theoretical analysis (Roberts et al., 1997; Roberts and Rosenthal, 1998).

1.3.4.2 Choice of the step-size parameter

Ideally, the proposal step-size σ should be set to give the optimal efficiency E if accuracy of the estimate is of interest. A computationally intensive approach is to run the algorithm for a range of σ values and choose the one that gives the highest efficiency, referred to as *grid evaluation*; see Figure 1.1a for an example. This is expensive and may not be practical in real applications. In practice, we employ an approach called *automatic scale adjustment* (Yang and Rodríguez, 2013), where we monitor P_{jump} and use it to adjust σ for a one-dimensional proposal. The optimal P_{jump} appears to be around 0.4 for unimodal kernels (such as Gaussian and uniform kernels) and 0.3 for bimodal kernels (such as the Bactrian-type kernels of Yang and Rodríguez (2013) and new kernels introduced in Chapter 3), based on a monotone decreasing relationship between σ and P_{jump} (Figure 1.1c), and the maximum efficiency is achieved at some intermediate σ (Figure 1.1a) and P_{jump} (Figure 1.1b). Specifically, we use the relationship $P_{\text{jump}}(\sigma) = \frac{2}{\pi}\tan^{-1}(2/\sigma)$ for the $N(0, 1)$ target and $x'|x \sim N(x, \sigma^2)$ kernel (Gelman et al., 1996b), to obtain the update formula

$$\sigma^* = \sigma \frac{\tan(\frac{\pi}{2} P_{\text{jump}})}{\tan(\frac{\pi}{2} P^*_{\text{jump}})}, \tag{1.12}$$

where σ is the current step-size, P_{jump} is the observed acceptance proportion, while σ^* and P^*_{jump} are the optimal ones. P^*_{jump} is specified by the user. We update σ several times during the burn-in.

1.3.4.3 Assessing convergence of simulation

In theory, the simulation must be run for long enough to achieve approximate convergence. In practice, it is often effective to perform simulation of multiple independent Markov chain sequences starting from different points and compare the estimates of posterior quantities of interest. If the estimates from independent runs do not agree at a desired level of accuracy, a longer run maybe required.

1.3.5 Summarising posterior inference from simulation draws

For continuous scalar variables, there are several choices of $100(1-\alpha)\%$ *posterior intervals* for $\alpha \in (0,1)$. The *central interval* (also called *equal-tailed interval*) lies between $\alpha/2$ and $1-\alpha/2$ quantiles which can be straightforwardly estimated from simulation by calculating appropriate quantiles. The *highest posterior density (HPD) region* contains $100(1-\alpha)\%$ of the highest posterior mass so that any point in this region has probability density at least as high as those outside. This is the shortest interval of the specified probability coverage and always contains the mode of the distribution. The HPD region can be estimated from simulation by calculating the empirical shortest probability intervals (Chen and Shao, 1999). Liu et al. (2015b) describe an algorithm that gives more stable estimates of the shortest probability intervals. For unimodal and symmetric distributions, the HPD region coincides with the central interval. For asymmetric distributions, such as those with a mode located at a boundary of the distribution support, the HPD region appears to be more representative as an inferential summary of the distribution.

For a tree structure, the *maximum a posteriori (MAP) tree* is the tree with the highest posterior probability (Rannala and Yang, 1996). However, this summary is not useful when the MAP tree has a low posterior probability. Alternatively, a $100(1-\alpha)\%$ *credibility set* (or *posterior set*) contains trees with highest posterior probabilities such that the total probability of trees in this set is at least $1-\alpha$ (Rannala and Yang, 1996; Mau and Newton, 1997). This is a discrete version of the HPD region, and is more useful when it contains only one or a few trees. Other common summaries of trees are a *majority-rule consensus (MRC) tree*, which contains clades that appear in at least half of the tree samples, and a *maximum clade credibility (MCC) tree*, which is the tree with the largest product or sum of the posterior clade probabilities. Many more tree summaries are discussed in Heled and Bouckaert (2013).

Chapter 2

Introduction to species tree inference using multispecies coalescent model

2.1 Phylogenetics using genome-scale sequence data

A wide variety of genome sequencing methods is now available for generating different types of genome-wide sequence data that can be used for phylogenetic inference. In this thesis, we will be focusing on using whole genome sequences. An advantage of using whole genomes is that different types of genomic regions (or loci) can be extracted to suit the phylogenetic questions of interest. Examples include coding regions and various kinds of noncoding regions such as introns, ultraconserved elements (UCEs) (McCormack et al., 2012) and conserved nonexonic elements (CNEEs) (Edwards et al., 2017). The main limitation, however, is that it is still costly and time-consuming to obtain genomes sequenced at high-coverage that are well assembled and well annotated. Consequently, only a small fraction of organisms have whole genomes available. More cost-effective methods for generating phylogenetic data involve sequencing only specific parts of the genome, known as targeted sequencing or genomic partitioning, reviewed in Lemmon and Lemmon (2013). These methods include transcriptome sequencing, restriction-site-associated DNA (RAD) sequencing and sequence capture. In contrast to whole genome sequencing, these methods require selection of genomic regions prior to sequencing experiment, and the amount of sequence data is usually much smaller. Non-sequence data can also be used to infer species trees, for instance, gene order, gene copy number and large-scale chromosomal features such as duplications and inversions.

Besides the choice of genomic regions, various other data processing steps can be performed prior to the actual species tree inference. These include subsampling of loci to filter out less informative positions (Molloy and Warnow, 2018), and assessing alignment quality and homology (Lemmon and Lemmon, 2013). However, the effect of these data processing steps on the downstream inference is still subject to debate, see e.g. Lanier et al. (2014); Sayyari et al. (2017); Molloy and Warnow (2018). Moreover, the genome assembly pipeline can also affect the sequence data quality. For example, in assembling diploid genomes,

it is common to 'haploidify' the sequences to ease the assembly process, particularly in polymorphic genomes, including the mosquito genomes analysed in Chapter 4 (Neafsey et al., 2015; Fontaine et al., 2015). However, the resulting consensus sequences may not represent any real haplotypes and could potentially increase the systematic error in the species tree inference. The use of phased diploid sequences should be preferred when available. Eriksson et al. (2018) illustrate the importance of using phased alleles for species tree inference in the context of understanding polyploid origins in plants. More studies are needed to quantify the effect of using haploidified consensus genomes and genome phasing on species tree inference.

2.2 Species tree inference from multilocus data

The increasing availability of genome-scale data provides an unprecedented opportunity to resolve contentious phylogenetic hypotheses. However, modelling and inference with the genome-wide data are fraught with difficulties due to complex interactions of various evolutionary processes operating at different levels. Here, we assume that the data consist of different genomic regions (or loci), referred to as multilocus data. Each locus consists of multiple sequences from different species. Each species may have more than one sequence. For diploid species, we may get either two sequences from a phased genome or one sequence from a consensus genome (Section 2.1) which is more commonly seen in assembled genomes. Finally, we assume that sequences in each locus are orthologous and are correctly aligned. In practice, misidentification of orthologues and misalignment can be problematic. Checking orthology and alignment quality should be performed whenever possible to reduce systematic error in species tree inference (Lemmon and Lemmon, 2013).

Given multilocus data, a traditional approach in phylogenetics is to either concatenate all loci into a single alignment and infer a single tree (called a *supermatrix* or *concatenation* approach), or to infer a gene tree for each locus and combine all trees into a single tree (called a *supertree* approach). However, there are problems with these methods when applied to genome-wide data. First, both concatenation and supertree approaches ignore the variation in the gene tree structures across genomic loci. Moreover, it has been shown via theoretical analysis and simulation studies that concatenation can lead to inconsistent estimates of the species tree with high support (Kubatko and Degnan, 2007; Liu and Edwards, 2009; Roch and Steel, 2015), and apparent substitution rate variation and biased branch length estimates (Mendes and Hahn, 2016), particularly when the species tree has short internal branches due to rapid successive speciation events. The supertree methods are heuristic and do not properly account for uncertainty in the gene trees.

Szöllősi et al. (2015) review gene-centric approaches for modelling species trees taking into account potentially different evolutionary histories at different loci across the genome. The variation in locus-wise genealogical histories, also called *gene trees*, can be attributed to several factors (Maddison, 1997; Nich-

ols, 2001; Degnan and Rosenberg, 2009). The variation can be due to random factors such as differential fixation of polymorphic alleles in different lineages, known as *incomplete lineage sorting (ILS)*, or non-random factors such as selection, gene flow between species, gene duplications and losses, as well as large structural features of the chromosomes such as inversions and heterochromatin regions which tend to reduce recombination. In Szöllősi et al. (2015), a three-level hierarchical model is suggested as a working model for species evolution: (1) a species tree at the top level for genome evolution, (2) a 'locus' tree at the middle level for gene family evolution and (3) a gene tree at the innermost level for sequence evolution. Generative models for each of these levels exist, but methods that fully capture all three levels in a single probabilistic model are currently lacking. While it might be easy to come up with such a model, inference computation is likely to be a major challenge in practice.

For single-copy loci, the concept of locus tree or gene family does not apply and can be omitted. This simplifies the modelling task. We will assume in this thesis that loci in the sequence alignments are single-copy orthologues. Consequently, gene duplications and losses can be eliminated as a mechanism for gene tree heterogeneity. With this assumption, the *multispecies coalescent (MSC)* framework (Rannala and Yang, 2003, 2017) provides a natural way to model the species tree and gene tree levels, capturing the heterogeneity in the evolutionary history of individual loci while being computationally tractable. We note that there are supertree methods that combine inferred gene trees according to the MSC model, treating estimated gene trees as input data. However, these methods, which we refer to as *approximate* or *summary coalescent methods*, tend to have identifiability and inconsistency issues, primarily due to loss of information in using gene tree estimates as inputs instead of sequence data as well as poor gene tree estimates when sequence data from individual loci have little phylogenetic information (Xu and Yang, 2016; Shi and Yang, 2018). There is increasing evidence favouring the MSC model as opposed to concatenation or summary coalescent methods (Ogilvie et al., 2016, 2017). We describe the MSC model in detail in the next section. An alternative approach to account for ILS is to directly model polymorphisms in the sequence data using a continuous-time Markov process for populations on tree, where the states represent allele frequencies and the transitions between states reflect mutations and random genetic drift. This class of models is referred to as *polymorphism-aware phylogenetic models (PoMo)* (De Maio et al., 2013, 2015; Schrempf et al., 2016). The number of states depends on the population size, which is constant throughout the species tree, and is fixed at a small value for computational tractability in practice. This population size parameter can be viewed as a discretisation approximation of the actual allele frequencies in the large population limit.

2.2.1 The multispecies coalescent (MSC) model

We first introduce notation before describing the model. A species tree (S, Θ) consists of a rooted binary tree structure S and scalar parameters $\Theta = \{\theta, \tau\}$ for population sizes (θ) and divergence times (τ). For

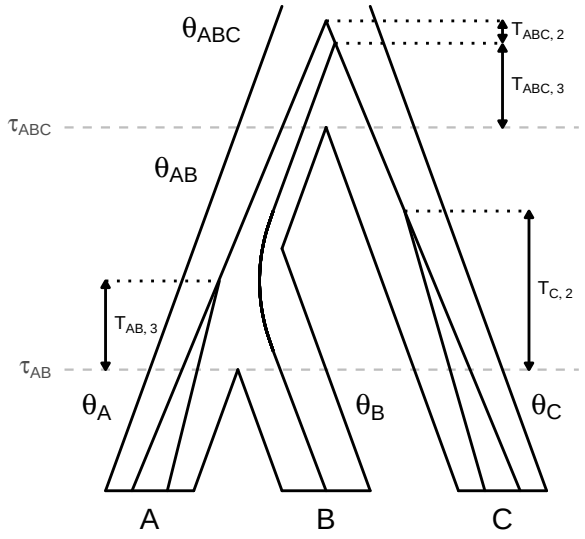


Figure 2.1: An example of a realisation of a gene tree from the MSC model for a given species tree $((A, B), C)$ of three species, with two sequences from A, one sequence from B and two sequences from C. The species tree has two divergence times τ_{AB}, τ_{ABC} and five population size parameters $\theta_A, \theta_B, \theta_C, \theta_{AB}, \theta_{ABC}$ corresponding to five branches of the species tree.

diploid species, the population size parameter is $\theta = 4N\mu$ where N is the effective population size and μ is the mutation rate per site per generation. Assuming that the mutation rate μ is constant throughout the species tree, both θ and τ can be measured in the units of mutations per site per generation. Each edge i of the species tree is associated with a population with size θ_i and the divergence time τ_i of the younger node. All extant populations have a zero divergence time. Thus for a tree with K leaves, there are $2K - 1$ θs (number of edges, including one for the root population) and $K - 1$ τs (number of internal nodes).

We now describe a generative process under the MSC model, followed by the calculation of the probability density. Given a species tree (S, Θ), a *gene tree* (G, t) with a binary tree *structure* G and *coalescent times* t is generated bottom-up via a coalescent process that sequentially merges a pair of lineages at a rate that depends on the population size θ, starting from extant populations at the terminal edges of the species tree. For each population $i = (u, v)$ (an edge on the species tree), where u is the parent node of v in the species tree S, let n_v and n_u be the numbers of lineages at times τ_v and τ_u, respectively. For example, for extant populations, n_v will be the number of sequences in the data. The coalescent process in this population starts at time τ_v with n_v lineages. Then given at time $t \in (\tau_v, \tau_u)$ with n_t remaining lineages, a *waiting time* T_{i,n_t} for the next coalescent event is drawn from the exponential distribution $\text{Exp}(\binom{n_t}{2}\frac{2}{\theta_i})$. This is the time since the last coalescent event or since τ_v, which ever is more recent, looking backwards in time. If $T_{i,n_t} < \tau_u$, the next coalescent event occurs, in which case, a pair of the remaining lineages is chosen uniformly at random to coalesce, and the process of sampling the next waiting time is repeated. Otherwise, no more coalescent event occurs until τ_u. The process terminates at the root population after all lineages have merged. An example of a realisation from this process is given in Figure 2.1.

The probability density of a given gene tree (G, t) under this process can be calculated independently on

each branch of the species tree as

$$p(G, t|S, \Theta) = \prod_i p(G_i, t_i|S, \Theta), \tag{2.1}$$

where the product is over $2K - 1$ branches of the species tree. For each branch i of the species tree, there are two types of events. The first type is a coalescent event where a random pair of the remaining lineages merge into a single lineage. Suppose there are n_t remaining lineages, the next coalescent event occurs at time $T_{i,n_t} \sim \text{Exp}(\binom{n_t}{2}\frac{2}{\theta_i})$, with probability

$$\frac{1}{\binom{n_t}{2}} \times \frac{2}{\theta_i}\binom{n_t}{2} e^{-\frac{2}{\theta_i}\binom{n_t}{2}T_{i,n_t}} = \frac{2}{\theta_i} e^{-\frac{2}{\theta_i}\binom{n_t}{2}T_{i,n_t}}.$$

Note that there are $n_v - n_u$ coalescent events in branch i. The second type of events occurs in the final time period where no more coalescence occurs until τ_u. This happens with probability $e^{-\binom{n_u}{2}\frac{2}{\theta_i}\Delta T}$ where $\Delta T \coloneqq \tau_u - \tau_v - \sum_{j=n_v}^{n_u+1} T_{i,j}$ is the remaining time before τ_u since the last coalescent event. Combining all the ingredients gives

$$\begin{aligned} p(G_i, t_i|S, \Theta) &= \left(\prod_{j=n_v}^{n_u+1} \frac{2}{\theta_i} e^{-\binom{j}{2}\frac{2}{\theta_i}T_{i,j}}\right) e^{-\binom{n_u}{2}\frac{2}{\theta_i}\left(\tau_u - \tau_v - \sum_{j=n_v}^{n_u+1} T_{i,j}\right)} \\ &= \left(\frac{2}{\theta_i}\right)^{n_v - n_u} \left(\prod_{j=n_{i,v}}^{n_u+1} e^{-\frac{j(j-1)}{\theta_i}T_{i,j}}\right) e^{-\frac{n_u(n_u-1)}{\theta_i}\left(\tau_u - \tau_v - \sum_{j=n_v}^{n_u+1} T_{i,j}\right)}. \end{aligned} \tag{2.2}$$

The coalescent times t_i (since the beginning of the process) are given by $t_{i,j} = \tau_v + \sum_{k=n_v}^{j} T_{i,k}$ for $j = n_v, n_v - 1, \ldots, n_u + 1$. As an example, for the gene tree in Figure 2.1, we have

$$\begin{aligned} p(G, t|S, \Theta) = {} & e^{-\frac{2}{\theta_A}\tau_{AB}} && \text{(population } A\text{)} \\ & \times \frac{2}{\theta_C} e^{-\frac{2}{\theta_C}T_{C,2}} && \text{(population } C\text{)} \\ & \times \frac{2}{\theta_{AB}} e^{-3\frac{2}{\theta_{AB}}T_{AB,3}} \times e^{-\frac{2}{\theta_{AB}}(\tau_{ABC} - T_{AB,3})} && \text{(population } AB\text{)} \\ & \times \frac{2}{\theta_{ABC}} e^{-3\frac{2}{\theta_{ABC}}T_{ABC,3}} \times \frac{2}{\theta_{ABC}} e^{-\frac{2}{\theta_{ABC}}T_{ABC,2}}. && \text{(population } ABC\text{)} \end{aligned}$$

Another example of gene tree density calculation is given at the end of Section 2.3.1 for the case of three species and three sequences.

This general formulation of the MSC model for any number of species and any number of sequences at each locus together with an MCMC algorithm for Bayesian inference was described in Rannala and Yang (2003). Earlier work recognising the distinction between gene divergence times and species divergence times appeared in Gillespie and Langley (1979); Hudson (1983); Tajima (1983); Takahata and Nei (1985). The case of three species and any number of sequences was studied by Pamilo and Nei (1988); Takahata (1989). Maximum likelihood inference of model parameters for three species and one sequence per species was

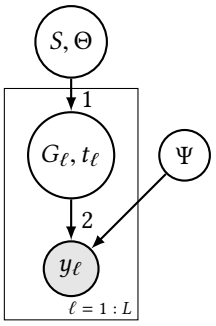


Figure 2.2: Model for species tree estimation. The shaded nodes y_ℓ indicates observed data. Other variables are parameters to be inferred. 1 = MSC model, 2 = sequence evolution model, L = number of loci.

implemented in Takahata et al. (1995), using the infinite sites model for sequence evolution. Yang (2002) extended Takahata et al. (1995) to use the JC model (Jukes and Cantor, 1969) and performed Bayesian inference using MCMC.

2.2.2 Species tree estimation

We now describe a generative model for multilocus sequence data $y = y_{1:L}$, where L is the number of loci, using the MSC model from the previous section, illustrated in Figure 2.2. The species tree (S, Θ) is generated from a probability distribution $p(S, \Theta) = p(S)p(\Theta)$ (described below). Then to generate a sequence alignment y_ℓ at each locus $\ell = 1, \ldots, L$, we first generate a gene tree (G_ℓ, t_ℓ) according to the MSC model (Section 2.2.1). Next, the sequence alignment y_ℓ is generated from a continuous-time Markov process along the branches of the gene tree (G_ℓ, t_ℓ) from root to leaves. This process may have additional parameters for the rate matrix of the Markov process, denoted Ψ. Assuming each locus evolves independently, we have the data model

$$p(y|\Psi, S, \Theta) = \prod_{\ell=1}^{L} p(y_\ell|\Psi, S, \Theta),$$

with

$$p(y_\ell|\Psi, S, \Theta) = \sum_{G_\ell} \int p(y_\ell|G_\ell, t_\ell, \Psi)p(G_\ell, t_\ell|S, \Theta)\, dt_\ell, \tag{2.3}$$

where $p(y_\ell|G_\ell, t_\ell, \Psi)$ is from the Markov process of sequence evolution (2 in Figure 2.2) and $p(G_\ell, t_\ell|S, \Theta)$ is from the MSC model (2.1) (1 in Figure 2.2). This data model assumes that the sequence data y_ℓ at each locus and the species tree (S, Θ) are conditionally independent given the gene tree (G_ℓ, t_ℓ) at that locus. The posterior distribution of the species tree is

$$p(S, \Theta, \Psi|y) \propto p(S)p(\Theta)p(\Psi) \prod_{\ell=1}^{L} p(y_\ell|\Psi, S, \Theta).$$

However, marginalising out all the gene trees $(G_\ell, t_\ell)_{\ell=1}^L$ and the Ψ parameters in (2.3) is only possible in the simplest cases of two or three species. In practice, we perform inference on the full joint posterior

$$p(S, \Theta, (G_\ell, t_\ell)_{\ell=1}^L, \Psi|y) \propto p(S)p(\Theta)p(\Psi)\prod_{\ell=1}^{L} p(y_\ell|G_\ell, t_\ell, \Psi)p(G_\ell, t_\ell|S, \Theta). \tag{2.4}$$

We describe the prior specification next. For each population i, we use an inverse Gamma prior $\theta_i \sim \text{InvG}(a_\theta, b_\theta)$, which has mean $\frac{b_\theta}{a_\theta - 1}$. The root divergence time is given $\tau_0 \sim \text{InvG}(a_\tau, b_\tau)$. The divergence times $\tau_1, \ldots, \tau_{K-2}$ for non-root nodes, where K is the number of species in the data, are uniform on the interval $(0, \tau_0)$ and sum to τ_0. These are generated using the symmetric Dirichlet(1), so that $p(\tau_1, \ldots, \tau_{K-2}) = \frac{(K-3)!}{\tau_0^{K-2}}$. Using the inverse Gamma priors allows analytic marginalisation of the θ parameter, which improves mixing and convergence properties of MCMC sampling (Rannala and Yang, 2017). The species tree prior $p(S)$ is the uniform distribution over rooted binary trees (Yang and Rannala, 2014). More complex models of species trees such as a birth-and-death process can also be used, with an extra computation cost.

There are two main implementations for posterior inference under the model (2.4): BPP (Yang, 2015; Rannala and Yang, 2017) and StarBEAST2 (Ogilvie et al., 2017). In Chapter 4, we will be using BPP to infer mosquito species trees. In particular, we will assume the JC model (Jukes and Cantor, 1969) for nucleotide evolution on the gene trees, so there are no parameters Ψ.

2.2.3 Modelling assumptions and limitations

In the full model (2.4) for species tree inference, we assume that (1) the gene trees are conditionally independent given the species tree, implying free recombination among loci, and (2) all alignment sites within each locus share the same gene tree, implying no recombination within each locus. These assumptions are mainly for computational convenience. Thus, ideally, loci should be short and far apart. Preprocessing of whole-genome data can be performed so that these requirements approximately hold. The sequence evolution model $p(y_\ell|G_\ell, t_\ell, \Psi)$ can be any Markov process on tree, possibly with mutation rate variation across loci and across branches of the species tree or gene tree. Using branch-specific rates on the species tree can improve the accuracy of species tree estimation compared with using a constant rate, but at a considerable computation cost (Ogilvie et al., 2017).

We now discuss some of the assumptions made by the MSC model for individual loci: (1) loci are evolving neutrally, (2) ILS is the only factor that can make gene trees differ from the species tree, (3) the population size is constant within each population, and (4) the mutation rate is constant throughout the gene tree. The first assumption may be dealt with by selecting non-coding regions of the genome since they are less likely to be under selection compared with the coding regions. Nonetheless, when the species are closely related, the coding regions also appear to be useful for estimating the species tree structure since they tend to

experience similar negative selection, which primarily reduces the neutral mutation rate without affecting much on the gene tree structure; see Shi and Yang (2018) and Chapter 4. The second assumption can be unrealistic for many datasets since other factors may also cause gene trees to disagree with the species tree, for example, gene flow and ancestral population structure (such as subdivided ancestral populations). The last two assumptions have a similar effect as selection, i.e. they are likely to mostly affect the coalescent times. More studies are needed to quantify the impact of each of these factors and their interactions on the accuracy and bias of the species tree estimates.

Various studies have assessed the robustness of the species tree estimates when the model assumptions are violated, reviewed in Liu et al. (2015a); Edwards et al. (2016). In particular, species tree estimation is shown to be robust to within-locus recombination via simulation studies (Lanier and Knowles, 2012). Shi and Yang (2018) discuss the impacts of several modelling assumptions made by the MSC model in the context of species tree inference among closely related species, in comparison with the concatenation and summary coalescent methods. Gruenstaeudl et al. (2016) demonstrate using posterior predictive checking (see Section 1.1) that the MSC model poorly fits data simulated with migration. The next section reviews how gene flow can be incorporated into the MSC framework.

2.3 Inferring gene flow

So far, the MSC model (Section 2.2.1) does not allow gene flow between populations. However, empirical evidence for gene flow in natural populations is plentiful (Mallet et al., 2016; Nieto Feliner et al., 2017). A wide range of methods is available for inferring gene flow from genomic data. Payseur and Rieseberg (2016) survey methods for inferring gene flow between diverging lineages from genome data as well as recent empirical genomic studies of hybridisation, reproductive isolation and speciation. They also identify important factors that need to be accounted for in current methods, which include variation of the migration rates across genome, over time and over geographical ranges, and the effect of selection on introgressed loci. More discussion on the modelling and inference aspect of speciation with gene flow, accounting for selection and recombination is given in Sousa and Hey (2013). Harrison and Larson (2014) review genomic studies of gene flow in hybrid zones. Tigano and Friesen (2016) discuss the interplay between natural selection and gene flow, and mechanisms that lead to adaptive introgression. Folk et al. (2018) review the current progress in genomic studies of hybridisation and introgression, emphasising the importance of factors such as polyploidy, climate and geographical distribution in addition to incomplete lineage sorting. Ignoring gene flow can lead to inconsistent estimates of species trees (Solís-Lemus et al., 2016; Long and Kubatko, 2018) or bias in the estimates of divergence times and population sizes (Leaché et al., 2014).

The MSC model can be extended to allow migration between lineages by adding migration events to the

coalescent process. But unlike coalescent events, migration simultaneously affects multiple populations that exist in the same period of time. Thus contemporary branches of the species tree that share the same time period are no longer independent. There is a need to keep track of the population in which each sequence resides, and the state of the coalescent process involves all contemporary populations. Consequently, the number of possible states of the coalescent process becomes combinatorially large in both the number of species and the number of sequences per species. Although the probability density of a gene tree can be derived based on a sample path of the process with coalescent and migration events, the expression is complex even for the simplest case of two species with two sequences, and requires numerical approximations to integrate out the migration history (Beerli and Felsenstein, 1999; Wang and Hey, 2010). It is much simpler to compute the gene tree probability density directly from the probability transition matrix of the coalescent process using the standard theory of continuous-time Markov process (Hobolth et al., 2011). The price to pay is the need to evaluate the entire transition matrix. Although there is a technique for reducing the size of the state space (Andersen et al., 2014), the approach remains impractical when the number of populations or the number of sequences is large (Andersen et al., 2014, Table 1). This type of models for migration between populations that are related through a tree structure falls under the banner of *isolation-with-migration (IM)* models. They are most studied in the case of two populations and two sequences (Hey and Nielsen, 2004; Wilkinson-Herbots, 2008). Only a few studies look at the case of more than two populations related through a tree structure (Hey, 2010; Andersen et al., 2014).

In the population genetics literature, models of migration between populations date back to at least Wright (1931, 1943). Early studies such as Li (1976) and Strobeck (1987) characterised the distribution of a pairwise nucleotide difference under various models of population structure, such as the infinite island model (Wright, 1931), the finite island model (Maruyama, 1970) and the stepping-stone model (Kimura and Weiss, 1964). The coalescent process with migration under these population structure models was studied by Takahata (1988), Notohara (1990), Nath and Griffiths (1993) and Wilkinson-Herbots (1998), and was referred to as the *structured coalescent*. However, this type of models assumes that the population structure is constant through time, ignoring the evolutionary relationships among the populations. Wakeley (1996) considered the structured coalescent of two populations that diverged from a common ancestral population at some time in the past. This is a two-population IM model, which can be considered an extension of the multispecies coalescent model of Takahata and Nei (1985) (Section 2.2.1). This model allows estimation of the population sizes, species divergence time and migration rates from sequence data (Nielsen and Wakeley, 2001; Hey and Nielsen, 2004). Their estimation procedure was subsequently developed into the IMa program (Hey and Nielsen, 2007) and its variants (Hey, 2010; Sethuraman and Hey, 2016).

Inference computation in the IM model is demanding since the calculation of gene tree density is considerably more involved compared with that for the MSC model. In practice, a trade-off between the number

of sequences and the number of loci must be made. The approach of Hey and Nielsen (2004, 2007); Hey (2010) is applicable for many sequences, but only for a small number of loci. Other methods that are based on explicit likelihood calculations work for genome-scale data, but are limited to only two or three species. In this case, different methods exist for likelihood calculation. For instance, Wang and Hey (2010) explicitly integrate out the migration history in the coalescent model for gene trees. Hobolth et al. (2011) and Andersen et al. (2014) use the transition probability matrix of the coalescent model which provides simpler and closed form expressions for gene trees, assuming constant migration rate over time. Lohse et al. (2011) and Lohse et al. (2016) use generating functions for gene trees, assuming the infinite-sites model for mutations. In Chapter 4, we will be considering the IM model for three species and three sequences, implemented in the program 3s (Zhu and Yang, 2012; Dalquen et al., 2017).

Finally, we note that there is an alternative approach that uses a phylogenetic network to model hybridisation or gene flow as a single event in time. By contrast, the IM-type models allow migration to occur continually over a period of time. We believe this is a more realistic approach to model gene flow between closely related species. In addition, Bayesian inference of phylogenetic networks with multispecies coalescent using MCMC is highly computationally intensive. Existing implementations only work for a few species (<10) and a small number of loci (~100) (Zhang et al., 2018; Wen and Nakhleh, 2018).

2.3.1 The isolation-with-migration (IM) model for three species and three sequences

We consider the IM model for three species (denoted 1, 2 and 3), assuming a fixed species tree $((1, 2), 3)$, and three sequences (denoted a, b and c) where gene flow is only allowed between the sister species 1 and 2. The program 3s (Zhu and Yang, 2012; Dalquen et al., 2017) implements maximum likelihood inference under this model. The species tree has two divergence times (τ_0, τ_1), five population size parameters $(\theta_1, \ldots, \theta_5)$ and two migration rates (M_{12} and M_{21}, for gene flow from and to population 1, respectively) (Figure 2.3). As for the MSC model, both τs and θs are in the units of the number of mutations per site. The migration rate $M_{ij} = N_j m_{ij}$ is the number of individuals in population j that come from population i. The two divergence times τ_0, τ_1 partition the time into three periods: $(0, \tau_1)$, (τ_1, τ_0) and (τ_0, ∞). There are eighteen possible gene tree structures since there are six possible unlabelled tree structures that differ by the time periods in which the two coalescent events occur, and each tree can be labelled at the tips in three different ways (Figure 2.3). Given three sequences a, b and c at a locus, we use the notation G_{kj} for the gene tree, with $k = 1, \ldots, 6$ and $j = a, b, c$, where $G_{kc} : ((a, b), c)$, $G_{ka} : ((b, c), a)$ and $G_{kb} : ((c, a), b)$. For example, both G_{5c} and G_{6c} represent the tree $((a, b), c)$, but G_{5c} has $\tau_1 < t_1 < \tau_0$ and $t_0 > \tau_0$, whereas G_{6c} has $t_1, t_0 > \tau_0$.

Given multilocus data $y_{1:L}$, where each locus has three sequences and each sequence can come from any

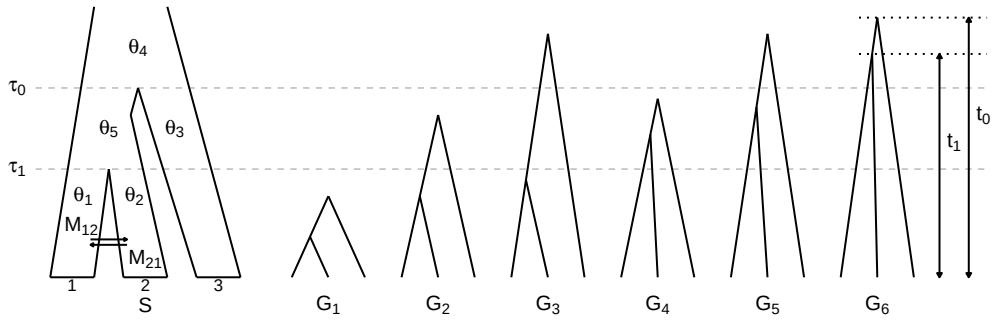


Figure 2.3: Species tree with three species S: $((1, 2), 3)$ and six possible gene tree structures $G_1, \ldots, G_6$. The species tree S has two divergence times τ_1 and τ_0, five population size parameters $\theta_1, \ldots, \theta_5$, and two migration rates M_{12}, M_{21} between species 1 and 2. Each gene tree has two coalescent times t_0 and t_1. Given three sequences a, b, c at each locus, there are three possible gene trees for each gene tree structure that differ by tip label: $G_{kc} : ((a, b), c)$, $G_{ka} : ((b, c), a)$ and $G_{kb} : ((c, a), b)$ for $k = 1, \ldots, 6$. Adapted from Figure 1 in Dalquen et al. (2017).

of the three species, the joint model of the sequence data and the gene trees $(G_\ell, t_\ell)_{\ell=1}^L$ is

$$p(y_{1:L}, (G_\ell, t_\ell)_{\ell=1}^L|\Theta) = \prod_{\ell=1}^{L} p(y_\ell, G_\ell, t_\ell|\Theta) = \prod_{\ell=1}^{L} p(y_\ell|G_\ell, t_\ell)p(G_\ell, t_\ell|\Theta),$$

where $p(y_\ell|G_\ell, t_\ell)$ is the standard phylogenetic likelihood on the gene tree (see Figure 2.2), but now $p(G_\ell, t_\ell|\Theta)$ comes from the IM model (described below) and $\Theta = \{\tau_0, \tau_1, \theta_1, \ldots, \theta_5, M_{12}, M_{21}\}$. The likelihood to be maximised is then given by marginalising out the gene tree, i.e.

$$p(y_\ell|\Theta) = \sum_k \int p(y_\ell, G_{\ell,k}, t_\ell|\Theta)\, dt_\ell.$$

For three species and three sequences, the integral over the two coalescent times is two-dimensional and can be calculated using a deterministic numerical quadrature method such as Gaussian quadrature, and the summation involves at most eighteen terms (Dalquen et al., 2017). The number of possible gene trees varies depending on the number of sequences from each species. For example, when there is one sequence from each species, denoted $1_a 2_b 3_c$, or simply 123, and gene flow is only allowed between species 1 and 2, there are five possible gene trees: G_{3c}, G_{5c}, G_{6a}, G_{6b} and G_{6c} since the second coalescent event must occur after τ_0 and must involve species 3.

When one of the sequences is from species 3 (the other two are from either species 1 or 2) and migration is restricted to the two sister species during the time period $(0, \tau_1)$, the model reduces to the case of two species and two sequences. For any sequence data of the form 112, 123 and 223, the coalescent process has the state space $\{113, 123, 223, 13|23\}$, where 13|23 denotes the state where the two sequences have coalesced in either species 1 or 2 (and the other sequence is in species 3), with the transition rate matrix Q given by

	113	123	223	13|23
113	$-2m_1 - \frac{2}{\theta_1}$	$2m_1$	0	$\frac{2}{\theta_1}$
123	m_2	$-m_2 - m_1$	m_1	0
223	0	$2m_2$	$-2m_2 - \frac{2}{\theta_2}$	$\frac{2}{\theta_2}$
13|23	0	0	0	0

where $m_1 \coloneqq \frac{m_{21}}{\mu} = \frac{m_{21}}{\theta_1/4N_1} = \frac{4M_{21}}{\theta_1}$ and $m_2 \coloneqq \frac{4M_{12}}{\theta_2}$ are migration rates per unit of time. Let $P(t) = e^{Qt}$ be the corresponding transition probability matrix at time $t > 0$. Let $s \in \{113, 123, 223\}$ denote the initial state from the data. The probability density for a gene tree with tree structure G_k and coalescent times t_1, t_0 (for the first and second coalescent events, respectively) is given by

$$p(G_{3c}, t_0, t_1|\Theta) = \left(P_{s,113}(t_1)\frac{2}{\theta_1} + P_{s,223}(t_1)\frac{2}{\theta_2}\right)\frac{2}{\theta_4}e^{-\frac{2}{\theta_4}(t_0-\tau_0)}, \quad t_1 \in (0,\tau_1), t_0 \in (\tau_0, \infty),$$

$$p(G_{5c}, t_0, t_1|\Theta) = \left(P_{s,113}(\tau_1) + P_{s,123}(\tau_1) + P_{s,223}(\tau_1)\right)\frac{2}{\theta_5}e^{-\frac{2}{\theta_5}(t_1-\tau_1)}\frac{2}{\theta_4}e^{-\frac{2}{\theta_4}(t_0-\tau_0)}, \quad t_1 \in (\tau_1,\tau_0), t_0 \in (\tau_0, \infty),$$

and for $j = a, c, b$,

$$p(G_{6j}, t_0, t_1|\Theta) = \left(P_{s,113}(\tau_1) + P_{s,123}(\tau_1) + P_{s,223}(\tau_1)\right)e^{-\frac{2}{\theta_5}(\tau_0-\tau_1)}\frac{2}{\theta_4}e^{-\binom{3}{2}\frac{2}{\theta_4}(t_1-\tau_0)}\frac{2}{\theta_4}e^{-\frac{2}{\theta_4}(t_0-t_1)},$$
$$t_1 \in (\tau_0, \infty), t_0 \in (t_1, \infty).$$

Integrating out t_0 and G_k gives the marginal density for t_1 as

$$p(t_1|\Theta) = \begin{cases} \frac{2}{\theta_1}P_{s,113}(t_1) + \frac{2}{\theta_2}P_{s,223}(t_1) & \text{if } t_1 < \tau_1, \\ \left(1 - P_{s,13|23}(\tau_1)\right)\frac{2}{\theta_5}e^{-\frac{2}{\theta_5}(t_1-\tau_1)} & \text{if } \tau_1 < t_1 < \tau_0, \\ 3\left(1 - P_{s,13|23}(\tau_1)\right)e^{-\frac{2}{\theta_5}(\tau_0-\tau_1)}\frac{2}{\theta_4}e^{-\frac{6}{\theta_4}(t_1-\tau_0)} & \text{if } t_1 > \tau_0. \end{cases} \tag{2.5}$$

Similarly, integrating out t_1 and G_k gives the marginal density for t_0 as

$$\begin{aligned} p(t_0|\Theta) = {} & P_{s,13|23}(\tau_1)\frac{2}{\theta_4}e^{-\frac{2}{\theta_4}(t_0-\tau_0)} + \left(1 - P_{s,13|23}(\tau_1)\right)\frac{2}{\theta_4}e^{-\frac{2}{\theta_4}(t_0-\tau_0)}\left(1 - e^{-\frac{2}{\theta_5}(\tau_0-\tau_1)}\right) \\ & + 3\left(1 - P_{s,13|23}(\tau_1)\right)e^{-\frac{2}{\theta_5}(\tau_0-\tau_1)}\frac{1}{2}\left(\frac{2}{\theta_4}e^{-\frac{2}{\theta_4}(t_0-\tau_0)} - \frac{2}{\theta_4}e^{-3\frac{2}{\theta_4}(t_0-\tau_0)}\right), \quad t_0 \in (\tau_0, \infty). \end{aligned} \tag{2.6}$$

Finally, the marginal distribution of the gene tree structure is given by

$$\begin{aligned} p(G_{3c}|\Theta) &= P_{s,13|23}(\tau_1) \\ p(G_{5c}|\Theta) &= \left(1 - P_{s,13|23}(\tau_1)\right)\left(1 - e^{-\frac{2}{\theta_5}(\tau_0-\tau_1)}\right) \\ p(G_{6j}|\Theta) &= \frac{1}{3}\left(1 - P_{s,13|23}(\tau_1)\right)e^{-\frac{2}{\theta_5}(\tau_0-\tau_1)} \qquad \text{for } j = a, b, c. \end{aligned} \tag{2.7}$$

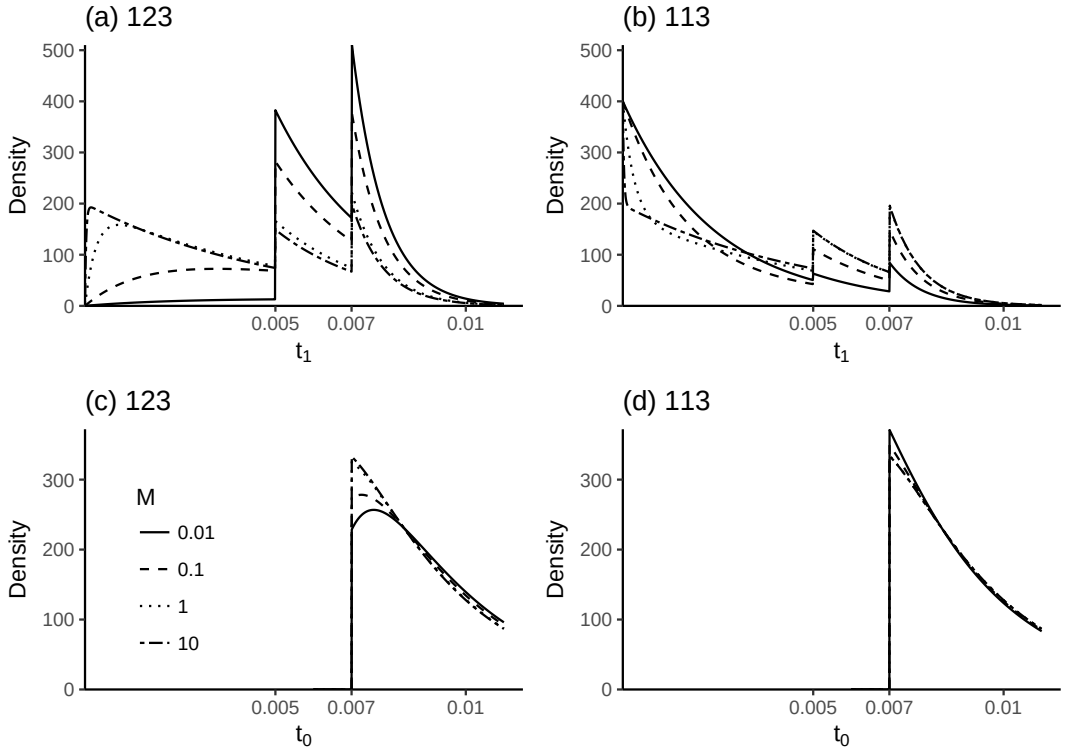


Figure 2.4: Probability density of the first coalescent time (t_1, top row), $p(t_1|\Theta)$ (2.5), and the second coalescent time (t_0, bottom row), $p(t_0|\Theta)$ (2.6), for two sequence configurations: 123 (left column) and 113 (right column). Parameters are $\tau_1 = 0.005$, $\tau_0 = 0.007$, $\theta_1 = \theta_2 = \theta_4 = \theta_5 = 0.005$ and $M_{12} = M_{21} =: M$, with $M = 0.01, 0.1, 1, 10$.

A complete description of the gene tree density $p(G_k, t_0, t_1|\Theta)$ for every sequence configuration is given in Dalquen et al. (2017) in terms of waiting times since the previous coalescent event instead of coalescent times.

In the special case when there is no gene flow, thus $M_{12} = M_{21} = 0$, the transition probability matrix $P(t)$ during the time period $(0, \tau_1)$ has a simple form[1]

	113	123	223	13|23
113	$e^{-\frac{2}{\theta_1}t}$	0	0	$1 - e^{-\frac{2}{\theta_1}t}$
123	0	1	0	0
223	0	0	$e^{-\frac{2}{\theta_2}t}$	$1 - e^{-\frac{2}{\theta_2}t}$
13|23	0	0	0	1

[1]The rate matrix Q can be diagonalised as $Q = S\Lambda S^{-1}$ with

$$Q = \begin{pmatrix} -\frac{2}{\theta_1} & 0 & 0 & \frac{2}{\theta_1} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{2}{\theta_2} & \frac{2}{\theta_2} \\ 0 & 0 & 0 & 0 \end{pmatrix}, \qquad S = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}, \qquad \Lambda = \operatorname{diag}\left(0, 0, -\frac{2}{\theta_2}, -\frac{2}{\theta_1}\right),$$

which gives $P(t) = e^{Qt} = Se^{\Lambda t}S^{-1}$ where $e^{\Lambda t} = \operatorname{diag}(1, 1, e^{-\frac{2}{\theta_2}t}, e^{-\frac{2}{\theta_1}t})$.

which gives the following the probability density for gene trees. For the 113 data, we have

$$p(G_{3c}, t_0, t_1) = \frac{2}{\theta_1} e^{-\frac{2}{\theta_1} t_1} \frac{2}{\theta_4} e^{-\frac{2}{\theta_4}(t_0 - \tau_0)}, \quad t_1 \in (0, \tau_1), t_0 \in (\tau_0, \infty),$$
$$p(G_{5c}, t_0, t_1) = e^{-\frac{2}{\theta_1}\tau_1} \frac{2}{\theta_5} e^{-\frac{2}{\theta_5}(t_1 - \tau_1)} \frac{2}{\theta_4} e^{-\frac{2}{\theta_4}(t_0 - \tau_0)}, \quad t_1 \in (\tau_1, \tau_0), t_0 \in (\tau_0, \infty),$$
$$p(G_{6j}, t_0, t_1) = e^{-\frac{2}{\theta_1}\tau_1} e^{-\frac{2}{\theta_5}(\tau_0 - \tau_1)} \frac{2}{\theta_4} e^{-\binom{3}{2}\frac{2}{\theta_4}(t_1 - \tau_0)} \frac{2}{\theta_4} e^{-\frac{2}{\theta_4}(t_0 - t_1)}, \quad t_1 \in (\tau_0, \infty), t_0 \in (t_1, \infty), \quad j = a, b, c.$$

The same expressions hold for the 223 data, but with θ_1 replaced by θ_2. For the 123 data, we have

$$p(G_{5c}, t_0, t_1) = \frac{2}{\theta_5} e^{-\frac{2}{\theta_5}(t_1 - \tau_1)} \frac{2}{\theta_4} e^{-\frac{2}{\theta_4}(t_0 - \tau_0)}, \quad t_1 \in (\tau_1, \tau_0), t_0 \in (\tau_0, \infty),$$
$$p(G_{6j}, t_0, t_1) = e^{-\frac{2}{\theta_5}(\tau_0 - \tau_1)} \frac{2}{\theta_4} e^{-\binom{3}{2}\frac{2}{\theta_4}(t_1 - \tau_0)} \frac{2}{\theta_4} e^{-\frac{2}{\theta_4}(t_0 - t_1)}, \quad t_1 \in (\tau_0, \infty), t_0 \in (t_1, \infty),$$

This case was considered in Takahata et al. (1995); Yang (2002). All these expressions can also be derived from the MSC model (Section 2.2.1) using (2.2), where each population can be considered independently.

Example 2.1. We give an example of the IM model with divergence times $\tau_1 = 0.005$, $\tau_0 = 0.007$, population sizes $\theta_1 = \theta_2 = \theta_4 = \theta_5 = 0.005$ and a symmetric migration rate $M_{12} = M_{21} =: M$. The probability densities of the two coalescent times are shown in Figure 2.4 for two sequence configurations: 123 and 113. We see that for the 123 data, when the migration rate is low, the first coalescence (between the sequences in species 1 and 2) is more likely to occur after τ_1 (looking backwards in time), either in the common ancestral population of species 1 and 2, or in the root population (Figure 2.4a). Higher migration rates make the first coalescence more likely to occur before τ_1, i.e. in either species 1 or 2. As a result, the gene tree G_{5c} in Figure 2.3 is more likely when the migration rate is low, while G_{3c} becomes more likely as the migration rate increases (Figure 2.5a). The 113 data has the opposite trend, i.e. when the migration rate is low, the coalescence between the two sequences in species 1 is more likely to occur before τ_1. This case is also reminiscent of a single-population coalescent model where the coalescent time follows an exponential distribution. As the migration rate increases, there is more chance that one of the two sequences in species 1 will migrate to species 2. This makes the coalescence become increasing more likely to occur after τ_1 (Figure 2.4b). Also unlike the 123 data, the gene tree G_{3c} remains the most likely tree (Figure 2.5b). The effect of the migration rate on the second coalescence is much less pronounced (Figure 2.4c-d). The effect is larger if the first coalescence is more likely to occur after τ_0 due to the requirement that $t_0 > t_1$.

2.3.2 Modelling assumptions and limitations

The IM model shares the same assumptions with the MSC model (Section 2.2.3) while allowing for gene flow between populations. However, gene flow may interact with other non-random factors such as recombination rate variation across the genome, and could be confounded with selection at linked sites

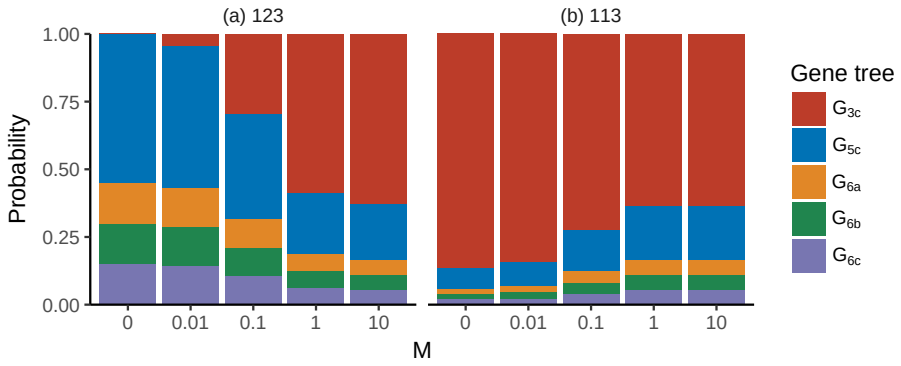


Figure 2.5: Gene tree probabilities $p(G_k|\Theta)$ (2.7) for two sequence data configurations: 123 and 113. Parameters are the same as in Figure 2.4. See Figure 2.3 for description of gene trees.

(Nachman and Payseur, 2012). The effect of such interactions on the estimates from the IM model requires further investigation. Furthermore, the constant migration rates over time and across the genome can be unrealistic for some datasets. In particular, two important cases are not captured by the IM model, namely, sympatric speciation with gene flow decreasing over time as reproductive or geographical barriers are forming, and gene flow during secondary contact of allopatric populations. Ignoring temporal variation in the gene flow rate can lead to biased estimates of model parameters (Innan and Watanabe, 2006; Becquet and Przeworski, 2009). Innan and Watanabe (2006) used time-dependent migration rates, but the resulting process is no longer time-homogeneous, and the calculation of transition probabilities becomes more complex or the likelihood must be approximated. One practical approach is to use piecewise-constant migration rates, in which case, the time can be split into intervals of constant migration rate. An example of this types of models is the isolation-with-initial-migration (IIM) where migration is only allowed during an initial period after divergence (Wilkinson-Herbots, 2012; Costa and Wilkinson-Herbots, 2017). Other violation of model assumptions can also produce biased estimates. These include ancestral population structure, population size dynamics, selection, gene flow between populations that are not part of the species tree, and the model of sequence evolution (Becquet and Przeworski, 2009; Strasburg and Rieseberg, 2010).

For species tree estimation, the joint model would be similar to (2.4), except that $p(G_\ell, t_\ell|S, \Theta)$ comes from the IM model. However, calculation of this gene tree density is a major computational difficulty due to the state space of the coalescent process being combinatorially large for arbitrary numbers of sequences and populations involved in migration (Andersen et al., 2014). Dalquen et al. (2017) suggest a workaround by reducing the problem to three species and at most three sequences per locus. For the case of three species, the 3s program can be used to infer the species tree indirectly by performing separate analysis for each of the three possible trees $((1, 2), 3)$, $((1, 3), 2)$ and $((2, 3), 1)$. The most probable species tree is expected to have the highest likelihood value among the three trees.

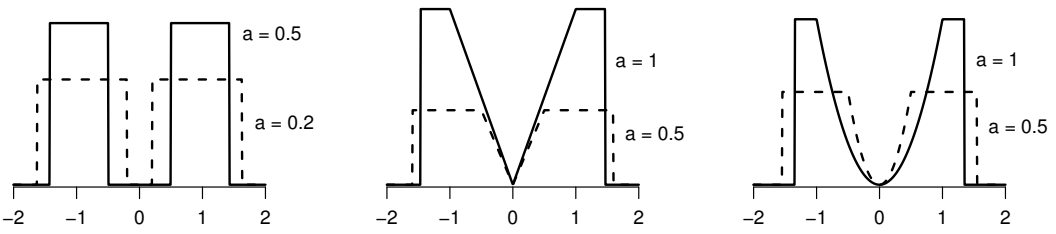
Chapter 3

Designing simple and efficient MCMC proposal kernels

In this chapter, we address the problem of designing efficient proposal kernels for the Metropolis–Hastings (MH) algorithm (Section 1.3.3), with each kernel implemented at an approximately optimal scale with respect to the asymptotic variance of the estimator (Section 1.3.4).

Yang and Rodríguez (2013) empirically demonstrated that using the so-called Bactrian kernels can substantially improve the asymptotic efficiency of the mean estimator for a range of univariate target distributions, compared with the uniform random walk, which is in turn more efficient than the Gaussian random walk. Here, we extend this work by proposing new proposal kernels and evaluate their statistical efficiency at the optimal step-size. We first present two classes of new one-dimensional proposal kernels for the MH algorithm (Section 3.1). In Section 3.1.1, we introduce three new bimodal kernels called Box, Airplane and StrawHat, and find that they have similar performance to the earlier Bactrian kernels, suggesting that the general shape of the proposal matters, but not the specific distributional form. We then propose a new class of kernels called the Mirror kernels in Section 3.1.2. This class of kernels directly introduces negative correlations in the Markov chain by generating new values around the 'mirror image' of the current value on 'the other side' of the target distribution, and in many cases achieves efficiency >100%.

For general multidimensional targets, we illustrate using several examples how a sequence of one-dimensional kernels and variable transformation can be used to improve the efficiency of the estimator (Section 3.2). We end the chapter with discussion about limitations of our work and connections with previous work (Section 3.3). We also give few examples of how variable transformation might be used as a general strategy for designing efficient MCMC kernels.



(a) Box (b) Airplane (c) StrawHat

Figure 3.1: Box, Airplane and StrawHat proposals. Each proposal is a one-parameter family of distributions with parameter a.

3.1 New one-dimensional proposals

These proposals attempt to reduce autocorrelations in the Markov chain, thereby improving the precision of the resulting MCMC estimates (see Section 1.3.3.3). One simple approach is to use a bimodal distribution with two modes on both sides of the current position. We describe three such proposals, called *Box*, *Airplane* and *StrawHat* (Figure 3.1). They have a bimodal shape similar to the Bactrian-type kernels given in Yang and Rodríguez (2013), and are symmetric, with $q(x'|x) = q(x|x')$. Another approach is to use non-symmetric kernels that directly induces negative correlations in the Markov chain, called the *Mirror kernel* (Figure 3.2).

For each of the proposal kernels described below, we first introduce a standard distribution version with zero mean and unit variance. Then given a current point x of the Markov chain, we give the proposal density with mean x and variance σ^2.

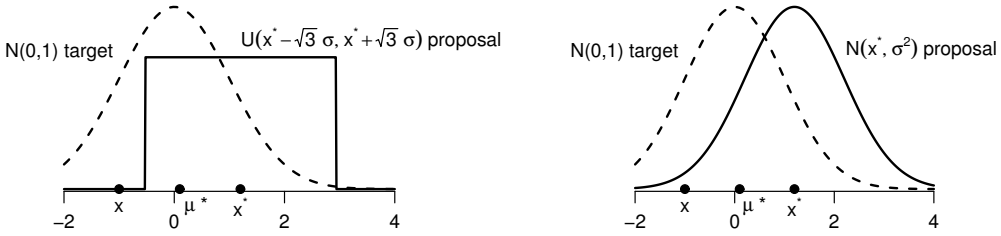
3.1.1 Bimodal kernels

3.1.1.1 Box

Given x, we generate x' uniformly from two intervals, one on each side of x (Figure 3.1a). The standard box distribution is $p(y; a) := \frac{1}{2(b-a)}$, $a \le |y| \le b$, where $b := \frac{1}{2}(\sqrt{12 - 3a^2} - a)$, and a is a parameter taking values in the interval $[0, 1)$. When $a = 0$, this is $U(-\sqrt{3}, \sqrt{3})$, which is the uniform kernel. In the proposal, we set $x' := x + \sigma y$, where y has the standard box distribution, with density $q(x'|x) = \frac{1}{2\sigma(b-a)}$, $\sigma a \le |x' - x| \le \sigma b$. To sample from $q(x'|x)$, draw $y \sim U(a, b)$ and $u \sim U(0, 1)$. If $u < \frac{1}{2}$, set $y \leftarrow -y$. Then set $x' \leftarrow x + \sigma y$.

3.1.1.2 Airplane

The standard Airplane distribution $p(y; a)$ is $\frac{1}{2b-a}$ if $a \le |y| \le b$ and $\frac{1}{2b-a}$ if $|y| < a$, where b is the root of $4b^3 - 12b + 6a - a^3 = 0$ with $b > a$, and $a \in [0, \sqrt{2})$ is a parameter (Figure 3.1b). The proposal density with mean x and variance σ^2 is $q(x'|x) = \frac{1}{\sigma(2b-a)}$ if $\sigma a \le |x' - x| \le \sigma b$ and $q(x'|x) = \frac{1}{\sigma a(2b-a)}|x' - x|$ if



(a) MirrorU (b) MirrorN

Figure 3.2: Examples of the proposal distribution for the two Mirror kernels when the current point is $x = -1$ and the estimated "centre" of the target distribution is $\mu^* = 0.1$. The proposal is centred at the mirror point $x^* = 2\mu^* - x$.

$|x' - x| < \sigma a$. To sample from $q(x'|x)$, draw $u_1, u_2, u_3 \sim U(0,1)$ independently. If $u_1 < \frac{a}{2b-a}$, set $y \leftarrow a\sqrt{u_2}$, otherwise draw $y \sim U(a,b)$. If $u_3 < \frac{1}{2}$, set $y \leftarrow -y$. Then set $x' \leftarrow x + \sigma y$.

3.1.1.3 StrawHat

The standard StrawHat distribution $p(y;a)$ is $\frac{3}{2(3b-2a)}$ if $a \le |y| \le b$ and $\frac{3}{2a^2(3b-2a)}y^2$ if $|y| < a$, where b is the root of $5b^3 - 15b + 10a - 2a^3$ with $b > a$, and $a \in [0, \sqrt{5/3})$ is a parameter. The proposal density $q(x'|x)$ can be derived similarly as for the Airplane kernel. To sample from $q(x'|x)$, draw $u_1, u_2, u_3 \sim U(0,1)$ independently. If $u_1 < \frac{a}{3b-2a}$, set $y \leftarrow a u_2^{1/3}$, otherwise draw $y \sim U(a,b)$. If $u_3 < \frac{1}{2}$, set $y \leftarrow -y$. Then set $x' \leftarrow x + \sigma y$.

For any of these three kernels (Box, Airplane, StrawHat), when $a = 0$, the kernel reduces to the uniform kernel. We note that if a is too close to its upper limit, the efficiency tends to drop off quickly as σ becomes too large (Figure 3.6). In practice, we suggest using $a = 0.5$ ($b = 1.43$) for Box, $a = 1$ ($b = 1.47$) for Airplane and $a = 1$ ($b = 1.35$) for StrawHat. Each kernel then has a step-size (σ) which can be adjusted to achieve good mixing.

3.1.2 Mirror kernels

In the Mirror kernel, we generate values around a point on 'the other side' of the target distribution that is the 'mirror image' of the current point x. Specifically, let μ^* be an estimate of the location of the target such as the mean or median. The proposal kernel is centred at $x^* := 2\mu^* - x$, the point with the same distance from μ^* as the current point x (Figure 3.2). We consider two variants, using either the uniform or Gaussian distribution. In the *MirrorU* kernel, we have

$$x'|x \sim U(2\mu^* - x - \sqrt{3}\sigma, 2\mu^* - x + \sqrt{3}\sigma), \tag{3.1}$$

and in the *MirrorN* kernel, we have

$$x'|x \sim N(2\mu^* - x, \sigma^2). \tag{3.2}$$

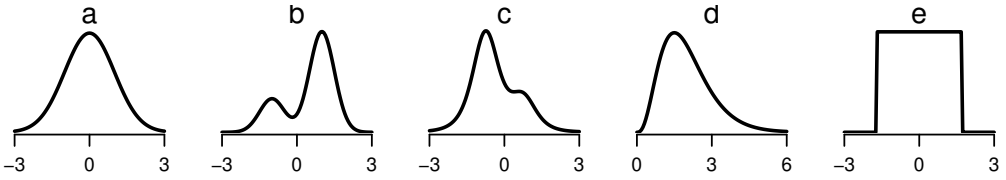


Figure 3.3: Five target distributions: (a) standard normal $N(0,1)$, (b) mixture of two normals $\frac{1}{4}N(-1,\frac{1}{4}) + \frac{3}{4}N(1,\frac{1}{4})$, (c) mixture of two t_4 distributions $\frac{3}{4}t_4(-\frac{3}{4},s^2) + \frac{1}{4}t_4(\frac{3}{4},s^2)$, (d) gamma $G(4,2)$ and (e) uniform $U(-\sqrt{3},\sqrt{3})$.

Both have mean $2\mu^* - x$ and variance σ^2.

For example, consider the $N(0,1)$ target. If μ^* is the true mean (μ) of the target, the optimal asymptotic efficiency (for estimating μ) is achieved by having $\sigma = 0$, in which case $E = \infty$ with $P_{\text{jump}} = 1$. However, in that case, the chain does not sample from the target, and E for estimating other functions may be 0. In general, if μ^* is close to the true mean, one would prefer a small σ to achieve a high efficiency (E), but a small σ may lead to slow convergence to the target distribution. On balance, we suggest two choices of the step-size: $\sigma = \hat{s}$ or $\sigma = \frac{1}{2}\hat{s}$, where $\hat{s}$ is the estimated target standard deviation. Both μ^* and $\hat{s}$ are initialised to some values at the beginning, and samples from the burn-in are used to estimate μ^* and $\hat{s}$.

If the target support is not the whole real line, the proposed value may lie outside the target support. While one could reject such values, rejection is not workable if all possible proposed values are outside the target support (i.e. when the proposal and target supports do not overlap). Reflection is another possibility but there are two problems. First, reflection would defeat the purpose of moving to the other side of the target. Second, with a small step-size, the reverse move $x' \to x$ of the MirrorU move with reflection may not be possible, thus breaking the detailed balance condition (1.6). As an example, consider a target with support $[0,\infty)$. For MirrorU (3.1) with $\mu^* = 1.5$ and step-size $\sigma = 1$, suppose the current value is $x = 5$. Then $x^* = 2\mu^* - x = -2$ and $x' \sim U(-3.73, -0.27)$. Suppose the proposed value is $x' = -0.2$, which is reflected to $x' = 0.2$. Now it is not possible to reach $x = 5$ from $x' = 0.2$ in the reverse direction because from $x' = 0.2$, we have $(x')^* = 2.8$ and the proposal is $x \sim U(1.07, 4.53)$.

Instead of rejection or reflection, we transform the target support X onto the real line $\mathbf{R}$ before applying the Mirror move. For instance, if $X = [a,\infty)$, we apply the Mirror move on the transformed variable $y := \log(x-a)$, with the proposal ratio $\frac{q(x|x')}{q(x'|x)} = \frac{x'-a}{x-a}$. For $X = [a,b]$, we use $y := \log\frac{x-a}{b-x}$, with $\frac{q(x|x')}{q(x'|x)} = \frac{(b-x')(x'-a)}{(b-x)(x-a)}$. With these log transformations, the original variable in the X space is multiplied by a random factor, and the Mirror proposal is referred to as *Mirror Multiplier*.

3.1.3 Experiments

We considered the following five target distributions (Figure 3.3): (a) standard normal distribution $N(0,1)$, with mean 0; (b) mixture of two normal distributions $\frac{1}{4}N(-1,\frac{1}{4}) + \frac{3}{4}N(1,\frac{1}{4})$, with mean $\frac{1}{2}$; (c) mixture of two t_4 distributions $\frac{3}{4}t_4(-\frac{3}{4},s^2)+\frac{1}{4}t_4(\frac{3}{4},s^2)$, where $s = \frac{1}{8}\sqrt{\frac{37}{2}}$, with mean $-\frac{3}{8}$; (d) gamma distribution $G(4,2)$,

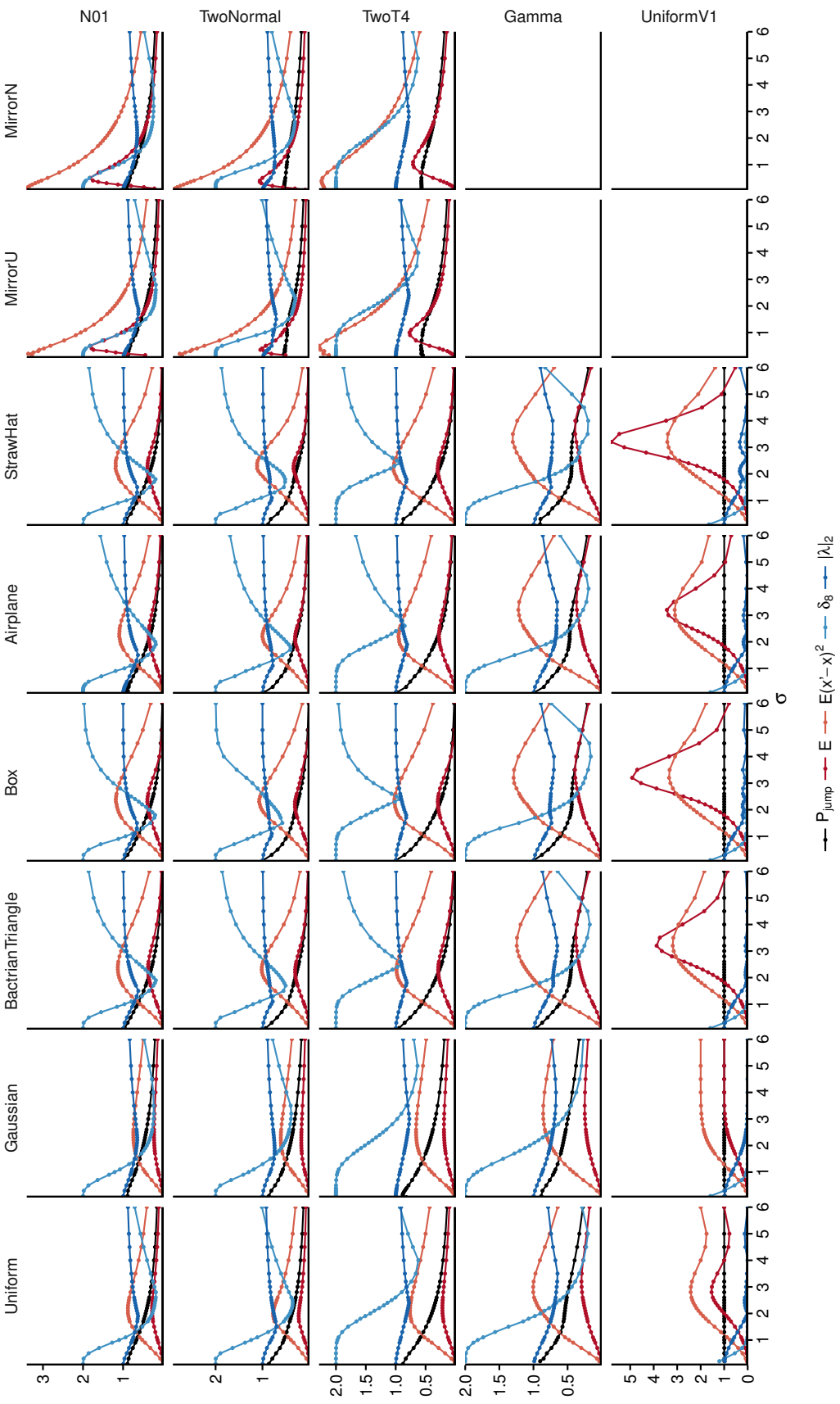


Figure 3.4: Efficiency (E) of eight proposal kernels for estimating the mean of five target distributions. Parameter: $a = 0.5$ for Box, $a = 1$ for Airplane and StrawHat, and $\mu^* = 0.1$ for MirrorU and MirrorN (the true means for N01, TwoNormal and TwoT4 are 0, $\frac{1}{2}$ and $-\frac{3}{8}$, respectively). The results for MirrorU and MirrorN kernels for gamma and uniform targets, which require a variable transformation, are in Figure 3.5.

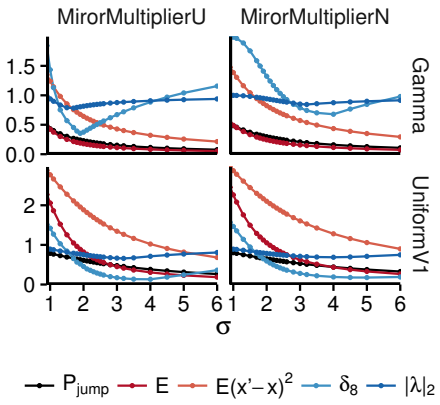


Figure 3.5: Efficiency (E) of the mirror multiplier kernels for estimating the mean of the gamma and uniform target distributions. For gamma target (mean 2), we used $\mu^* = 1.5$, i.e. we applied the mirror kernel to $\log x$, with mean 0.563 and $\log \mu^* = 0.405$. For uniform target (mean 0), we used $\mu^* = 0.1$, i.e. we applied the mirror kernel to $\log \frac{x-a}{b-x}$, with mean 0 and $\log \frac{\mu^*-a}{b-\mu^*} = 0.116$.

with mean 2; (e) uniform distribution $U(-\sqrt{3}, \sqrt{3})$, with mean 0. Each of the five targets has variance 1. Note that even though the density in (b) has two modes, we focus here on simple targets with a single mode; we do not expect the proposals discussed here to work well when the target has multiple peaks separated by deep valleys.

Note that the targets (d) and (e) have a constrained support. Sampling from targets with constrained support is often dealt with using rejection or truncated proposals (or truncated full conditionals in the context of Gibbs sampling) (Gelfand et al., 1992; Browne, 2006). We note that rejection can be very inefficient if a large proportion of proposed values are discarded, while the truncated variables can be expensive to simulate, often based on the inverse transform method (Algorithm 1.1); see e.g. Devroye (1986, p. 38). It is simpler and typically more efficient (in terms of the amount of computation involved as well as the asymptotic variance of the estimator) to use reflection (e.g. Yang and Rodríguez (2013)). For example, if x has support on the interval $[a, \infty)$ and if the kernel is symmetric with $q(x'|x) = q(x|x')$, we generate $x' \sim q(x'|x)$, and set $x' \leftarrow 2a - x'$ if $x' < a$. The proposal ratio is 1.

We evaluated five new proposals (Box, Airplane, StrawHat, MirrorU and MirrorN; Figures 3.1 and 3.2) described in Section 3.1, together with the uniform, Gaussian and BactrianTriangle proposals from Yang and Rodríguez (2013). We used 5×10^7 iterations after a burn-in of 10^4 iterations. Figures 3.4 and 3.5 show the performance of eight proposal kernels applied to five targets plotted against the proposal step-size σ. We observed large variations in efficiency as σ changes, emphasising the importance of choosing σ to achieve high efficiency. We also note that for the uniform and Gaussian kernels, the optimal σ for the convergence rate (measured by δ_8 and $|\lambda|_2$) was larger than that for mixing, while the opposite was true for the bimodal kernels.

The Box, Airplane and StrawHat kernels had similar efficiency to the Bactrian-type kernels from Yang and Rodríguez (2013), with Box and StrawHat generally performing slightly better than the BactrianTriangle

Table 3.1: Efficiency and convergence rate measures of proposal kernels for estimating the mean of the five one-dimensional target distributions (all have variance 1).

Kernel	optimal σ	P_{jump}	E	E_π^2	ρ_1	δ_8	$\lvert\lambda\rvert_2$
Target $N(0,1)$							
Uniform	2.2	0.405	0.276	0.879	0.560	0.230	0.671
Gaussian	2.5	0.426	0.228	0.744	0.628	0.286	0.657
BactrianTriangle ($m = 0.95$)	2.3	0.304	0.377	1.131	0.434	0.442	0.829
Box ($a = 0.5$)	2.3	0.290	0.394	1.150	0.410	0.608	0.857
Airplane ($a = 1$)	2.2	0.334	0.360	1.096	0.452	0.296	0.789
StrawHat ($a = 1$)	2.2	0.308	0.395	1.188	0.406	0.488	0.838
MirrorU ($\mu^* = 0.1$)	0.5	0.821	1.823	2.815	−0.408	1.828	0.865
MirrorN ($\mu^* = 0.1$)	0.5	0.828	1.824	2.884	−0.442	1.840	0.880
Target $\frac{1}{4}N(-1,\frac{1}{4}) + \frac{3}{4}N(1,\frac{1}{4})$							
Uniform	1.9	0.385	0.227	0.771	0.614	0.454	0.746
Gaussian	2.2	0.388	0.171	0.608	0.696	0.501	0.750
BactrianTriangle ($m = 0.95$)	2.2	0.271	0.303	1.010	0.495	0.705	0.880
Box ($a = 0.5$)	2.2	0.261	0.308	1.057	0.472	0.806	0.894
Airplane ($a = 1$)	2.2	0.283	0.304	1.004	0.498	0.603	0.863
StrawHat ($a = 1$)	2.2	0.269	0.339	1.114	0.443	0.693	0.878
MirrorU ($\mu^* = 0.1$)	0.35	0.525	1.045	2.503	−0.252	1.983	0.884
MirrorN ($\mu^* = 0.1$)	0.35	0.525	1.058	2.534	−0.267	1.980	0.893
Target $\frac{3}{4}t_4(-\frac{3}{4},s^2) + \frac{1}{4}t_4(\frac{3}{4},s^2)$							
Uniform	2.2	0.366	0.218	0.760	0.620	1.276	0.794
Gaussian	2.6	0.377	0.192	0.659	0.670	1.157	0.791
BactrianTriangle ($m = 0.95$)	2.3	0.276	0.289	0.986	0.507	1.054	0.881
Box ($a = 0.5$)	2.3	0.254	0.296	1.025	0.488	1.014	0.894
Airplane ($a = 1$)	2.2	0.295	0.277	0.954	0.523	1.147	0.852
StrawHat ($a = 1$)	2.2	0.272	0.300	1.041	0.480	1.086	0.884
MirrorU ($\mu^* = 0.1$)	1.0	0.550	0.769	1.922	0.039	1.964	0.925
MirrorN ($\mu^* = 0.1$)	1.0	0.542	0.710	1.964	0.018	1.960	0.931
Target $G(4,2)$							
Uniform	3.2	0.464	0.297	0.998	0.501	0.388	0.652
Gaussian	3.5	0.463	0.249	0.856	0.572	0.450	0.674
BactrianTriangle ($m = 0.95$)	3.5	0.403	0.378	1.241	0.379	0.213	0.665
Box ($a = 0.5$)	3.5	0.398	0.392	1.284	0.358	0.200	0.702
Airplane ($a = 1$)	3.5	0.412	0.371	1.209	0.395	0.224	0.654
StrawHat ($a = 1$)	3.5	0.414	0.388	1.302	0.349	0.206	0.717
Target $U(-\sqrt{3},\sqrt{3})$							
Uniform	2.8	1	1.537	2.425	−0.212	0.000	0.216
Gaussian	∞	1	1.000	2.000	0.000	0.000	0.000
BactrianTriangle ($m = 0.95$)	3.2	1	3.875	3.190	−0.595	0.022	0.604
Box ($a = 0.5$)	3.2	1	4.916	3.346	−0.673	0.060	0.682
Airplane ($a = 1$)	3.2	1	3.439	3.107	−0.554	0.013	0.562
StrawHat ($a = 1$)	3.2	1	5.801	3.421	−0.710	0.091	0.719

Note.—Parameter μ^* for the Mirror kernels was chosen arbitrarily to be 0.1 and not optimised; see text.

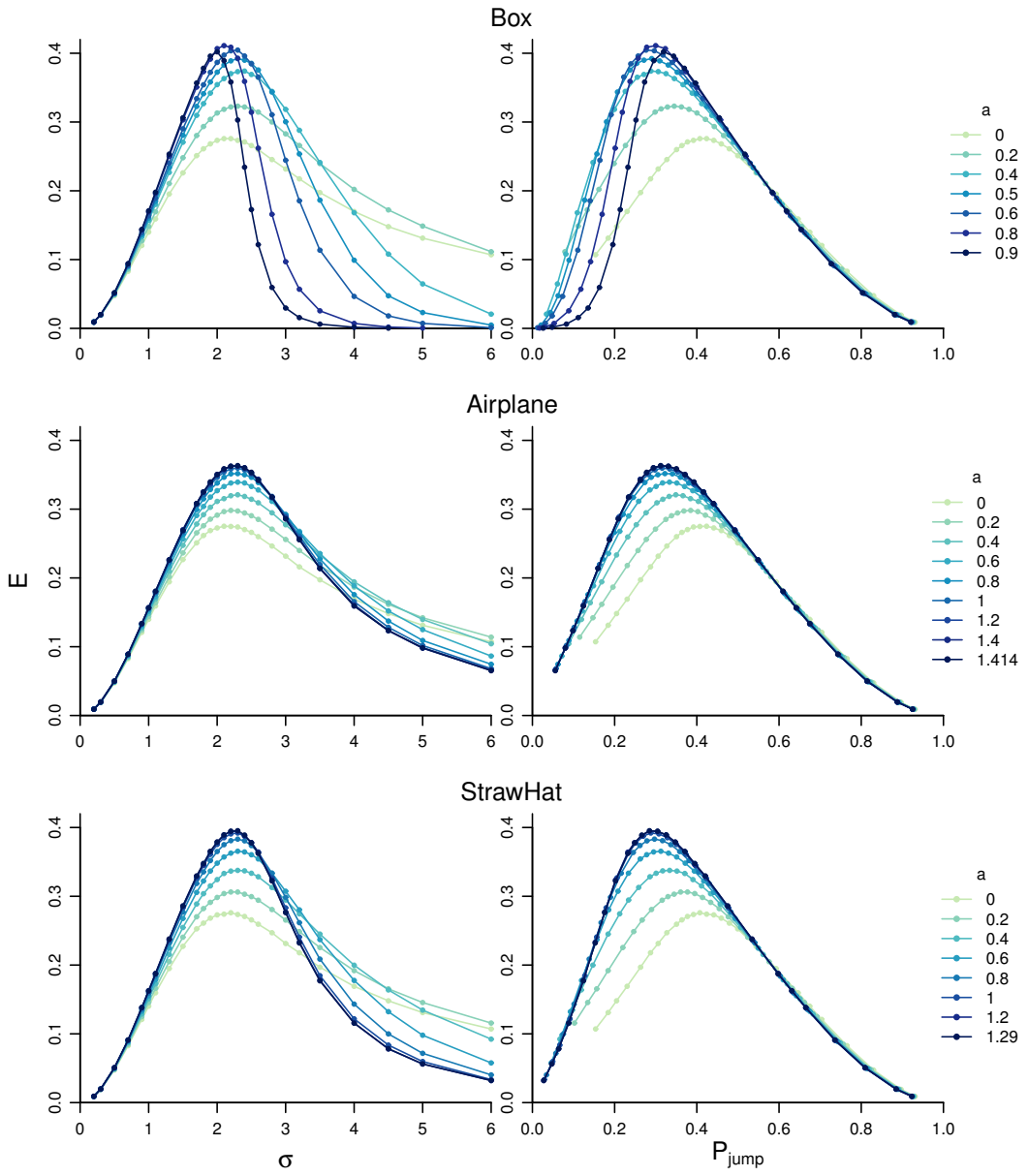


Figure 3.6: Effect of the parameter a for Box, Airplane and StrawHat kernels on the efficiency for estimating the mean of $N(0, 1)$.

kernel (Table 3.1). In addition, all these bimodal kernels were better than the unimodal Gaussian and uniform kernels. Thus the detail of the distributional form appeared to be less important. Among these bimodal kernels, we prefer the StrawHat as it tends to achieve high efficiency without being too sensitive to the choice of step-size (Figure 3.6).

For the MirrorU and MirrorN kernels, we fixed μ^* to 0.1 for all targets in this Section, except the gamma target, where we used $\mu^* = 1.5$. Using a fixed μ^* allowed us to optimise the step length σ and obtain smooth efficiency curves (Figure 3.4) without averaging over many simulation replicates. The two Mirror kernels generally achieved several-fold improvements in efficiency, and are 'super-efficient', with $E > 1$, in most cases (Table 3.1). In practical applications, we suggest setting $\mu^* = \hat{\mu}$ and $\sigma = \hat{s}$ or $\sigma = \frac{1}{2}\hat{s}$, with both the target mean μ and standard deviation s estimated during the burn-in (Section 3.1.2). If the estimated

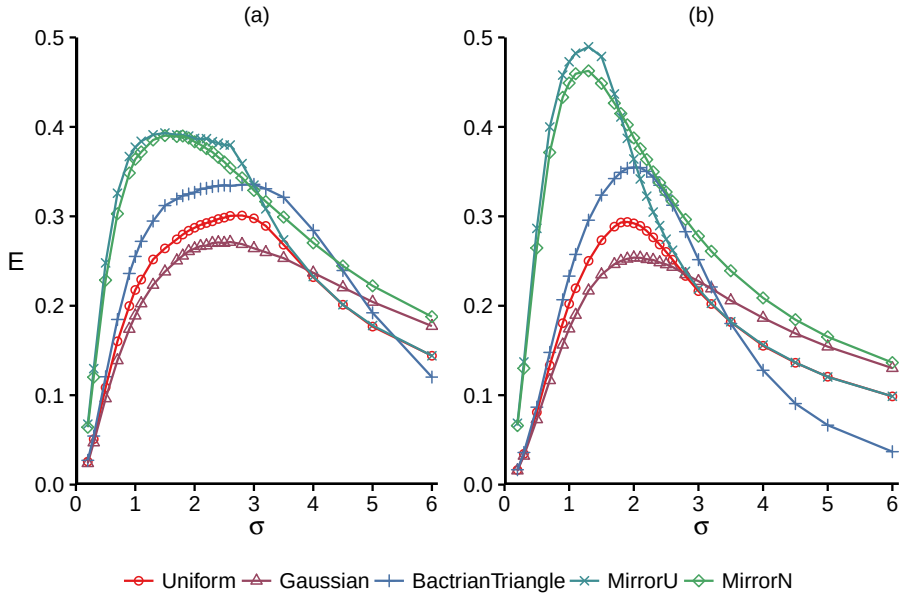


Figure 3.7: Efficiency (E) of five proposal kernels for estimating a tail probability of the normal distribution $N(0,1)$: (a) $\mathbf{P}(x > 2.3263) = 0.01$, and (b) $\mathbf{P}(x > 1.2815) = 0.1$. For MirrorU and MirrorN kernels, μ^* was fixed to 0.1.

mean is closer to the true mean than the fixed μ^* used in our experiments, performance will be better as well. For the $N(0,1)$ target, the efficiency, averaged over 10 replicates, is 1.290 for $\sigma = \hat{s}$, and 2.815 for $\sigma = \frac{1}{2}\hat{s}$ for MirrorN, compared with $E = 1.824$ when $\mu^* = 0.1$ is fixed and σ is optimised in Table 3.1.

We note that the ranking of the proposal kernels was largely the same across these five targets (Table 3.1), suggesting that this pattern may hold for fairly arbitrary targets. For the Box, Airplane and StrawHat kernels, the optimal P^*_{jump} was reasonably stable across the targets evaluated. We suggest using the automatic scale adjustment (1.12) for setting the proposal step-size σ, with $P^*_{\text{jump}} = 0.3$.

Finally, to assess whether the efficiency ordering of the kernels depends on the specific function estimated, we considered estimating a tail probability of the normal target $N(0,1)$. For estimating the probability $\mathbf{P}(x > 2.3263) = 0.01$, we obtained the same ordering of the kernels as for estimation of the target mean (Figure 3.7a). The highest efficiency was $E \approx 0.4$, achieved by the two Mirror kernels. Similar results were obtained for estimating the probability $\mathbf{P}(x > 1.2815) = 0.1$ (Figure 3.7b), but with a generally narrower range of σ achieving the maximum efficiency. The MirrorU kernel was the most efficient, with $E \approx 0.5$. For both cases, the efficiency and optimal σ are generally comparable with those for the mean estimation (Table 3.1), except for the two Mirror kernels where the highest efficiency drops significantly from $E \approx 1.8$ (Table 3.1), and the optimal σ increases to about 1.3-1.5. This reduced efficiency improvement could be due to the fact that the Mirror kernels are designed to be efficient for estimating location statistics such as mean or mode of the target distribution. The effect of 'moving to the other side' of the target should be less beneficial for estimating other properties of the target such as a tail probability. The sensitivity of the Mirror kernels to different estimation quantities as well as its tuning parameter μ^* remains to be explored.

3.2 Multidimensional target distributions

3.2.1 Two-dimensional Gaussian targets

We considered two bivariate Gaussian targets $N_2(0, I)$ and $N_2(0, \Sigma)$ with $\Sigma = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$ with $\rho = 0.9$. For the $N_2(0, I)$ target, we compared several proposal kernels that were either a two-dimensional distribution, or a cycle of two one-dimensional distributions. For the $N_2(0, \Sigma)$ target, we also used a variable transformation to deal with the correlation between target components.

The proposal kernels considered are as follows. Let $x = (x_1, x_2)$ be the current value and $x' = (x'_1, x'_2)$ be the proposed value from $q(x'_1, x'_2|x_1, x_2)$.

- Two-dimensional proposals on $\mathbf{R}^2$:
  - K1. Gaussian:
    $$q(x'_1, x'_2|x_1, x_2) = N(x'_1, x'_2|(x_1, x_2), \sigma^2 I_2).$$
    This is a symmetric kernel, with $q(x'_1, x'_2|x_1, x_2) = q(x_1, x_2|x'_1, x'_2)$. The proposal ratio is 1.
  - K2. Square:
    $$q(x'_1, x'_2|x_1, x_2) = \frac{1}{12\sigma^2} 1_S(x_1, x_2)$$
    where $S := \{(u, v) \in \mathbf{R}^2 : x_1 - \sqrt{3}\sigma \le u \le x_1 + \sqrt{3}\sigma, x_2 - \sqrt{3}\sigma \le v \le x_2 + \sqrt{3}\sigma\}$ is the square of side length $2\sqrt{3}\sigma$, centred at (x_1, x_2), and $1_S(x_1, x_2) := 1$ if $(x_1, x_2) \in S$, and 0 otherwise; this has mean (x_1, x_2) and covariance I_2. To generate (x'_1, x'_2) from this kernel, draw $x'_i \sim U(x_i - \sqrt{3}\sigma, x_i + \sqrt{3}\sigma)$ independently for $i = 1, 2$. The proposal ratio is 1.
  - K3. Disc:
    $$q(x'_1, x'_2|x_1, x_2) = \frac{1}{4\pi\sigma^2} 1_S(x_1, x_2)$$
    where $S := \{(u, v) \in \mathbf{R}^2 : (x_1 - u)^2 + (x_2 - v)^2 \le 4\sigma^2\}$ is the disc of radius 2σ, centred at (x_1, x_2); this has mean (x_1, x_2) and covariance I_2. To generate (x'_1, x'_2) from this kernel, first draw $r \sim U(0, 1)$ and $\theta \sim U(0, 2\pi)$, then set $x'_1 := x_1 + 2\sigma\sqrt{r}\cos\theta$ and $x'_2 := x_2 + 2\sigma\sqrt{r}\sin\theta$. The proposal ratio is 1.
- Two one-dimensional proposals (one for each coordinate):
  - K4. Two 1D uniform proposals:
    1. First, draw $u \sim U(x_1 - \sqrt{3}\sigma, x_1 + \sqrt{3}\sigma)$ and set $(x'_1, x_2) = (u, x_2)$ with probability $\min\left(1, \frac{\pi(u, x_2)}{\pi(x_1, x_2)}\right)$, otherwise set $(x'_1, x_2) = (x_1, x_2)$. The proposal ratio is 1.
    2. Then, draw $v \sim U(x_2 - \sqrt{3}\sigma, x_2 + \sqrt{3}\sigma)$ and set $(x'_1, x'_2) = (x'_1, v)$ with probability $\min\left(1, \frac{\pi(x'_1, v)}{\pi(x'_1, x_2)}\right)$, otherwise set $(x'_1, x'_2) = (x'_1, x_2)$. The proposal ratio is 1.

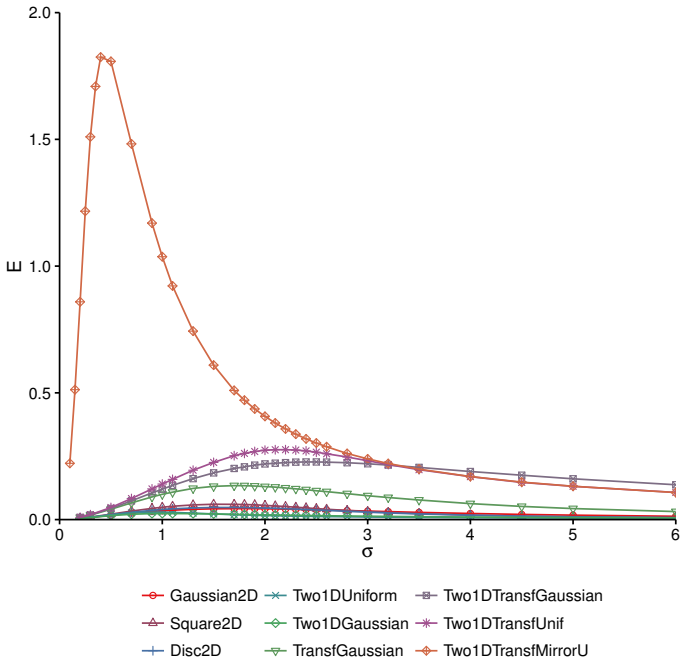


Figure 3.8: Efficiency of proposal kernels for the $N_2(0, \Sigma)$ target, with $\Sigma = \left(\begin{smallmatrix} 1 & 0.9 \\ 0.9 & 1 \end{smallmatrix}\right)$.

- K5. Two 1D Gaussian proposals. This is similar to the uniform one (K4), but with $u \sim N(x, \sigma^2)$ and $v \sim N(y, \sigma^2)$ instead.

For the $N_2(0, \Sigma)$ target, we considered four additional proposal kernels, based on the whitening transformation

$$y = \Sigma^{-1/2} x \tag{3.3}$$

to remove the correlation between the two components and rescale all the components to have variance 1.

- K6. 2D Gaussian with transformation. To propose a new point, generate $y' \sim N(y, \sigma^2 I)$ and set $x' = \Sigma^{1/2} y'$. The proposal ratio is 1.
- K7. Two 1D Uniform with transformation. This is similar to K4, but for the transformed variable y.
- K8. Two 1D Gaussian with transformation. This is similar to K7, but with Gaussian proposal instead of uniform.
- K9. Two 1D MirrorU with transformation. This is similar to K7, but with MirrorU proposal instead of uniform, with $\mu^* = 0.1$ for both components.

For the $N_2(0, I)$ target, the two two-dimensional versions of the uniform kernel, Square2D (K2) and Disc2D (K3), were more efficient than Gaussian2D (K1) as expected (Table 3.2). The efficiency was almost doubled when two one-dimensional proposals were used instead of a single two-dimensional move (compare Two1DUniform (K4) with Square2D and Disc2D, and Two1DGaussian (K5) with Gaussian2D in Table 3.2). The optimal σ and efficiency for these kernels agreed with those for the $N(0, 1)$ target in Table 3.1 as

Table 3.2: Efficiency for estimating the mean of two-dimensional Gaussian targets.

	Kernel	$N_2(0, I)$					$N_2(0, \Sigma)$				
		optimal σ	P_{jump}	E	E_π^2	ρ_1	optimal σ	P_{jump}	E	E_π^2	ρ_1
K1	Gaussian2D	1.7	0.352	0.134	0.544	0.762	1.8	0.159	0.043	0.174	0.913
K2	Square2D	1.7	0.300	0.155	0.475	0.728	1.5	0.167	0.060	0.236	0.882
K3	Disc2D	1.7	0.294	0.156	0.552	0.724	1.5	0.153	0.048	0.192	0.904
K4	Two1DUniform	2.2	0.407	0.276	0.880	0.560	1.0	0.393	0.030	0.166	0.917
K5	Two1DGaussian	2.5	0.430	0.228	0.744	0.628	1.0	0.456	0.024	0.141	0.930
K6	TransfGaussian2D			n/a			1.7	0.352	0.134	0.475	0.762
K7	Two1DTransfUnif			n/a			2.2	0.407	0.276	0.880	0.560
K8	Two1DTransfGaussian			n/a			2.5	0.430	0.228	0.743	0.629
K9	Two1DTransfMirrorU ($\mu^* = 0.1$)			n/a			0.4	0.852	1.825	2.987	−0.494

Note.—The $N_2(0, \Sigma)$ target has covariance $\Sigma = \left(\begin{smallmatrix} 1 & 0.9 \\ 0.9 & 1 \end{smallmatrix}\right)$. Efficiency ($E$) is for estimating the mean of the first component $\mu_1 = \mathrm{E}x_1$.

expected. Note, however, that this improvement in statistical efficiency comes with an extra cost in computing time that scales with the target's dimensionality. If the target is d-dimensional, it would require d evaluations of target density and d MH acceptance steps instead of just one.

For the $N_2(0, \Sigma)$ target, applying kernels K1-K5 directly gave poor results, with efficiency of only $2 - 6\%$, compared with over 10% efficiency for the $N_2(0, I)$ target (Table 3.2 and Figure 3.8). This inefficiency was because these proposals failed to account for the high correlation ($\rho = 0.9$) between the variables in the target. When the correlation was removed via whitening transformation (3.3) in TransfGaussian2D (K6), we recovered the same efficiency of 0.134 as achieved by Gaussian2D (K1) on the $N_2(0, I)$ target. The same was true when combining one-dimensional moves with the transformation (3.3); compare Two1DTransfUnif (K7) with Two1DUnif (K4) on $N_2(0, I)$ ($E = 0.276$), and Two1DTransfGaussian (K8) with Two1DGaussian (K5) on $N_2(0, I)$ ($E = 0.228$). Note that simply using one-dimensional proposals without transformation can yield worse performance than the corresponding two-dimensional moves as correlations make it more difficult to make a large move along the axis-aligned directions. Finally, the Two1DTransfMirrorU (K9) kernel was several times more efficient than all other kernels considered.

3.2.2 Multivariate Gaussian target using multidimensional uniform and Mirror kernels

We extended the one-dimensional uniform and MirrorN kernels to multiple dimensions for the $N_d(0, I)$ target, obtaining optimal scaling, optimal efficiency and P_{jump} (Table 3.3). For the uniform kernel, we considered the Cube and Sphere extensions in multi-dimensions. For MirrorN, we considered two variants, MirrorN1 with $x'|x \sim N(x^*, \hat{\Sigma})$ and MirrorN$\frac{1}{2}$ with $x'|x \sim N(x^*, \frac{1}{4}\hat{\Sigma})$, where $x^* = 2\mu^* - x$, with μ^* and $\hat{\Sigma}$ being the estimated target mean and variance. The efficiency was calculated by averaging over 10 replicates.

We found that the Cube and Sphere kernels were more efficient than the Gaussian kernel for $d = 1, 2, 3, 4$, but both were similar to the Gaussian for $d > 4$. The MirrorN1 and MirrorN$\frac{1}{2}$ kernels were several times more efficient than the Gaussian, Cube and Sphere kernels for $d \leq 10$, with MirrorN$\frac{1}{2}$ being over twice more efficient than MirrorN1. Note that these MirrorN moves evaluated in Table 3.3 are d-dimensional moves. In comparison, the efficiency of one-dimensional MirrorU and MirrorN is higher than 100% whatever the dimension of the target is (Table 3.1).

3.2.3 Hundred-dimensional Gaussian target

To demonstrate the scalability of our approach to high dimensions, we considered the $N_{100}(0, \Sigma)$ target where Σ^{-1} was generated from a Wishart distribution with identity scale matrix and 100 degrees of freedom. The target distribution used had many strong correlations, with 1627 out of 4950 pairs of variables

Table 3.3: Optimal step-size (σ) and asymptotic efficiency (E) for the Gaussian target $N_d(0, I)$ and five proposal kernels. The results for the Gaussian kernel are from Table 1 in Gelman et al. (1996b). For MirrorN1 and MirrorN$\frac{1}{2}$, the proposal covariance was $\hat{\Sigma}$ and $\frac{1}{4}\hat{\Sigma}$, respectively, where $\hat{\Sigma}$ is the estimated target covariance from the burn-in. The results were averaged over 10 replications.

d	Gaussian kernel			Cube kernel			Sphere kernel			MirrorN1 kernel		MirrorN$\frac{1}{2}$ kernel	
	optimal σ	E	P_{jump}	optimal σ	E	P_{jump}	optimal σ	E	P_{jump}	E	P_{jump}	E	P_{jump}
1	2.40	0.233	0.441	2.20	0.276	0.416	2.20	0.276	0.416	1.290	0.706	2.815	0.846
2	1.70	0.136	0.352	1.64	0.155	0.317	1.62	0.157	0.315	0.869	0.552	2.118	0.756
3	1.39	0.098	0.316	1.36	0.107	0.284	1.36	0.109	0.279	0.643	0.453	1.727	0.694
4	1.25	0.076	0.279	1.20	0.081	0.266	1.18	0.083	0.262	0.501	0.382	1.418	0.635
5	1.10	0.062	0.275	1.07	0.065	0.259	1.07	0.067	0.252	0.375	0.314	1.259	0.601
6	1.00	0.053	0.266	0.98	0.055	0.252	0.98	0.056	0.245	0.297	0.266	1.100	0.567
7	0.93	0.047	0.261	0.91	0.047	0.249	0.91	0.048	0.242	0.251	0.233	0.970	0.533
8	0.87	0.041	0.255	0.85	0.041	0.245	0.85	0.042	0.240	0.190	0.191	0.861	0.503
9	0.80	0.037	0.261	0.80	0.037	0.245	0.80	0.037	0.237	0.160	0.170	0.749	0.469
10	0.74	0.034	0.267	0.76	0.033	0.243	0.76	0.034	0.236	0.129	0.145	0.683	0.444

Table 3.4: Efficiency for estimating the mean of the first component of the target $N_{100}(0, \Sigma)$.

Kernel	Proposal σ	P_{jump}	E	ρ_1
1DTransfGaussian (true Σ)	Automatic	0.392	0.225	0.631
1DTransfGaussian (estimated Σ)	Automatic	0.401	0.228	0.624
1DTransfMirrorU1 (true Σ)	$\sigma = s$	0.674	1.072	−0.034
1DTransfMirrorU1 (estimated Σ)	$\sigma = \hat{s}$	0.675	1.031	−0.016
1DTransfMirrorU$\frac{1}{2}$ (true Σ)	$\sigma = \frac{1}{2}s$	0.830	2.433	−0.423
1DTransfMirrorU$\frac{1}{2}$ (estimated Σ)	$\sigma = \frac{1}{2}\hat{s}$	0.821	2.319	−0.397
HMC (Stan)	Automatic	0.894	0.00682	0.983

having correlations with magnitude greater than 0.99.

We compared the one-dimensional Gaussian and MirrorU kernels. For the MirrorU, the parameter μ^* was set to the target mean estimated during the burn-in, and the component-specific proposal step-size σ was set to either $\hat{s}$ or $\frac{1}{2}\hat{s}$, where $\hat{s}$ is the estimated standard deviation of the component in the target. These two proposals are referred to as MirrorU1 and MirrorU$\frac{1}{2}$, respectively. We used the whitening transformation (3.3) to remove correlations among the components and rescale all the components to have variance 1. This transformation requires the target's covariance matrix Σ, which was estimated during the burn-in. For comparison, we also included the popular Stan algorithm (version 2.15.1) (Carpenter et al., 2017), which implements HMC with automatic tuning.

We used 10^5 iterations of burn-in and 10^7 iterations of the main chain. If estimation of Σ was required, we initialised Σ to the identity matrix and updated it every 10^4 iterations (thus ten rounds of update in total). The final covariance matrix used by the sampler was based on the last 10^4 burn-in samples. For the Gaussian kernel, we used automatic tuning of proposal step-size (1.12) with optimal $P_{\text{jump}} = 0.4$.

For this problem, the MirrorU1 and MirrorU$\frac{1}{2}$ kernels gave about four-fold and ten-fold increase in efficiency, respectively, compared with the Gaussian kernel (Table 3.4). Efficiency was similar whether the true or estimated variances were used, illustrating that the approach of estimating the variance is practical. The Stan algorithm did not perform well and took about 100 times longer than the Gaussian and MirrorU kernels.

3.2.4 Bayesian logistic regression

Next, we applied the MirrorU kernel to a Bayesian logistic regression analysis of the German credit dataset. The same dataset was used by Girolami and Calderhead (2011) to demonstrate several state-of-the-art MCMC algorithms, namely MALA, HMC and their Riemannian manifold versions. We also included the Stan algorithm (Carpenter et al., 2017) for comparison. Note that MALA and HMC require the first derivatives, while their manifold versions additionally require the Fisher information matrix as well as its

Table 3.5: Efficiency for estimating the mean of the posterior distribution for the logistic regression problem. Running time (in seconds) was for 10^6 iterations for all kernels. The 1D Gaussian kernel was implemented in both C and Matlab, and indicated a 2-fold difference in running time between the two languages.

Parameter	Kernel							
	1DUniform	1DMirrorU$\frac{1}{2}$	1DGaussian	MALA	HMC	Manifold MALA	Manifold HMC	HMC (Stan)
1	0.255	0.678	0.143	0.075	0.144	0.710	1.099	1.402
2	0.260	1.245	0.166	0.088	0.149	0.732	1.120	1.532
3	0.253	1.153	0.088	0.049	0.153	0.752	1.093	1.346
4	0.244	1.572	0.144	0.074	0.151	0.743	1.098	1.459
5	0.239	1.071	0.082	0.045	0.153	0.762	1.092	1.325
6	0.260	1.196	0.175	0.082	0.149	0.746	1.090	1.549
7	0.259	1.634	0.149	0.080	0.156	0.739	1.090	1.505
8	0.253	1.236	0.195	0.118	0.150	0.745	1.094	1.634
9	0.250	1.320	0.147	0.082	0.155	0.758	1.071	1.566
10	0.280	1.270	0.076	0.038	0.155	0.737	1.099	1.303
11	0.269	1.086	0.114	0.059	0.154	0.731	1.089	1.454
12	0.241	1.802	0.193	0.125	0.154	0.756	1.102	1.596
13	0.252	1.266	0.141	0.073	0.151	0.743	1.098	1.458
14	0.266	1.625	0.184	0.105	0.151	0.738	1.088	1.608
15	0.261	1.244	0.115	0.059	0.153	0.739	1.084	1.420
16	0.245	0.161	0.184	0.051	0.131	0.712	1.143	1.433
17	0.260	1.548	0.181	0.109	0.153	0.741	1.092	1.566
18	0.253	1.314	0.155	0.062	0.149	0.717	1.103	1.534
19	0.263	0.473	0.070	0.034	0.147	0.700	1.105	1.289
20	0.266	0.601	0.070	0.036	0.149	0.713	1.092	1.292
21	0.273	1.293	0.039	0.020	0.158	0.736	1.103	1.102
22	0.269	1.535	0.038	0.020	0.153	0.735	1.102	1.105
23	0.250	1.134	0.119	0.065	0.150	0.737	1.109	1.351
24	0.266	1.455	0.054	0.026	0.153	0.725	1.087	1.155
25	0.273	1.006	0.054	0.026	0.152	0.723	1.087	1.149
Running time (s), C	936	936	937	n/a	n/a	n/a	n/a	2263
Running time (s), Matlab	n/a	n/a	1981	299	4649	7069	25039	n/a

derivatives. The target distribution is

$$p(\theta|x,y) \propto p(\theta)\prod_{n=1}^{N} p(y_n|x_n,\theta)$$
$$\propto \exp\left(-\frac{1}{2\alpha}\theta^\top\theta + \sum_{n=1}^{N} y_n(\theta^\top x_n) - \sum_{n=1}^{N}\log(1+e^{\theta^\top x_n})\right),$$

where θ is a vector of an intercept term and 24 regression coefficients, x_n is a vector of 24 normalised predictors (with zero mean and unit variance), $y_n \in \{0,1\}$ is an indicator for a good credit risk, and $N = 1000$. We gave each component of θ an independent Gaussian prior $N(0,\alpha)$ with $\alpha = 100$, following Girolami and Calderhead (2011). Each chain was run for 10^7 iterations after 10^4 burn-in iterations. For MALA, MCMC and the manifold versions, we used the Matlab implementation of Girolami and Calderhead (2011) and ran for 10^6 iterations.

From Table 3.5, the multidimensional MALA and HMC proposals were worse than the simple one-dimensional 1DUniform and were comparable to the 1DGaussian kernel. The manifold versions of MALA and HMC were much better than all those four, and Stan performed the best. The Mirror$\mathrm{U}\frac{1}{2}$ kernel had comparable efficiency to manifold HMC and Stan, achieving super-efficiency ($E > 1$) for most of the 25 parameters, while taking less time. We note that the Mirror kernel requires estimation of the target mean and variance, but is otherwise very simple to implement, and does not require any fine-tuning. MALA and HMC require estimation of the target variance, and the manifold versions in addition need higher derivatives or Fisher information. In complex models where analytic expressions of the required derivatives are not available, automatic differentiation may be used to evaluate derivatives at machine precision. However, this comes at the cost of increased running time, especially for higher derivatives, as well as making the implementation considerably more complex as a specialised library is required. In addition, MALA, HMC and their manifold versions all have at least one parameter that requires tuning. The Mirror kernel appeared to strike a good balance between efficiency and simplicity. However, manifold MALA, HMC and manifold HMC gave consistent efficiency across dimensions, while for the Mirror kernel, some components had much lower efficiency than the rest (Table 3.5).

3.2.5 Molecular clock dating in phylogenetics

We applied the proposal kernels studied above to a Bayesian inference problem of estimating species divergence time and evolutionary rate using molecular sequence data from two species. The dataset was the 12S rRNA gene from the mitochondrial genome of human and orangutan from Horai et al. (1995), summarised as $x = 90$ differences out of $n = 948$ sites.

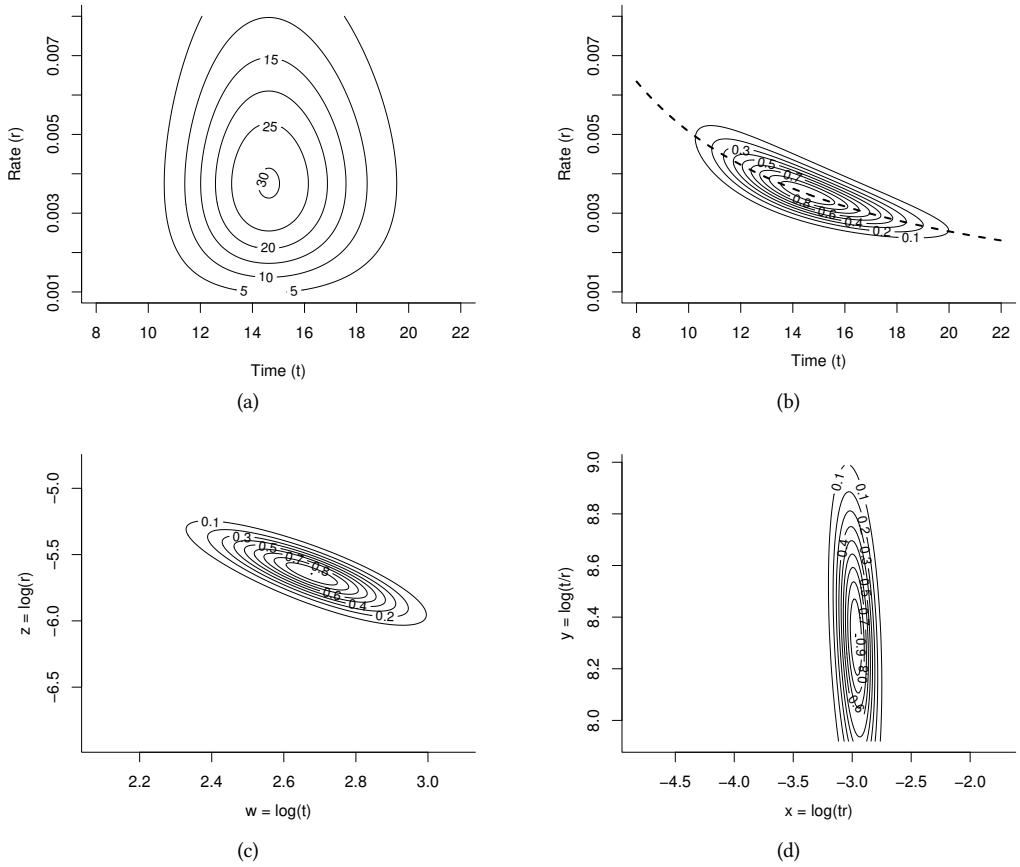


Figure 3.9: Prior $p(t, r)$ (a) and posterior $p(t, r|x)$ (b) distributions for the molecular clock dating problem. The dashed curve in the posterior (b) indicates the values of (t, r) for which $2tr = \theta = 0.1015$ (see text). (c) and (d) are different transformations of (b). All plots are based on the same ranges of values of t and r.

3.2.5.1 Model

The evolutionary process at each site is modelled as a continuous-time Markov process on the four nucleotides (T, C, A, and G) with the transition rate matrix $Q = \{q_{ij}\}$, with $q_{ij} = \lambda$ for any $i \neq j$ (Jukes and Cantor, 1969). The substitution rate for each nucleotide is thus $r = 3\lambda$ per time unit, which is one million years here. The transition probability matrix is $P_t = \{P_t(i, j)\}$, with

$$P_t(i,j) = \begin{cases} \frac{1}{4} + \frac{3}{4}e^{-\frac{4}{3}tr} & \text{if } i = j, \\ \frac{1}{4} - \frac{1}{4}e^{-\frac{4}{3}tr} & \text{if } i \neq j. \end{cases}$$

Given the data of x differences at n sites, the likelihood is

$$p(x|t,r) = \left(\frac{1}{16} + \frac{3}{16}e^{-\frac{8}{3}tr}\right)^{n-x} \left(\frac{1}{16} - \frac{1}{16}e^{-\frac{8}{3}tr}\right)^{x}.$$

This is a function of the genetic distance $\theta := 2tr$, but not of t and r individually. From our data, the maximum likelihood estimate of θ was $\hat{\theta} = \frac{3}{4}\log(\frac{3n}{3n-4x}) = 0.1015$, with the 95% likelihood interval

(0.0817, 0.1245).

We assigned gamma priors $t \sim G(40, 40/15)$, with mean 15 Myrs and 95% interval (10.7, 20.0), and $r \sim G(4, 800)$, with mean 0.005 substitutions per million years, and 95% interval (0.0014, 0.0110) (Figure 3.9a). The posterior distribution is

$$\begin{aligned} p(t, r|x) &\propto p(y|t, r)p(t)p(r) \\ &\propto \left(\frac{1}{16} + \frac{3}{16}e^{-\frac{8}{3}tr}\right)^{n-x} \left(\frac{1}{16} - \frac{1}{16}e^{-\frac{8}{3}tr}\right)^{x} t^{39}e^{-(40/15)t}r^{3}e^{-800r}. \end{aligned} \tag{3.4}$$

We sampled from this posterior $p(t, r|x)$ (Figure 3.9b) using MCMC algorithms with different proposal schemes, and compared their efficiencies for estimating the posterior means of t and r.

3.2.5.2 MCMC algorithms for posterior inference

Since the uniform proposal is generally more efficient than the Gaussian proposal, we considered seven proposal kernels (A1-7) based on the uniform and MirrorU kernels and five state-of-the-art MCMC algorithms: MALA, HMC, HMC (Stan), manifold MALA, manifold HMC (A8-A12), which are based on a multivariate Gaussian proposal. The derivatives and Fisher information matrices required by algorithms A8-A12 were derived using the unnormalised posterior (3.4); these quantities were analytically tractable but tedious to derive. We used variable transformations to deal with correlations and/or scale differences of the target variables (Figure 3.9b). Depending on the transformation used, each algorithm has component-specific scaling parameters. Specifically, σ_t and σ_r are standard deviations of proposals on t and r; σ_w and σ_z are for $w \coloneqq \log t$ and $z \coloneqq \log r$ (Figure 3.9c); σ_x and σ_y are for $x \coloneqq \log(tr)$ and $y \coloneqq \log(t/r)$ (Figure 3.9d). The details for tuning these step-size parameters are summarised in Table 3.6.

Algorithm A1 1D Uniform on t, r. The algorithm consists of two MH steps.

1. Draw $t'|t \sim U(t - \sigma_t\sqrt{3}, t + \sigma_t\sqrt{3})$. If $t' < 0$, set $t' \leftarrow -t'$. The proposal ratio is 1.
2. Draw $r'|r \sim U(t - \sigma_r\sqrt{3}, t + \sigma_r\sqrt{3})$. If $r' < 0$, set $r' \leftarrow -r'$. The proposal ratio is 1.

The step-size parameters σ_t and σ_r were automatically tuned to achieve $P_{\text{jump}} = 0.4$ (see Section 1.3.4.2).

Algorithm A2 1D Uniform on w, z. The algorithm consists of two MH steps.

1. Draw $u \sim U(-\sqrt{3}, \sqrt{3})$ and set $t' \leftarrow te^{\sigma_w u}$. The proposal ratio is $\frac{t'}{t}$.
2. Draw $v \sim U(-\sqrt{3}, \sqrt{3})$ and set $r' \leftarrow re^{\sigma_z v}$. The proposal ratio is $\frac{r'}{r}$.

The step-size parameters σ_w and σ_z were automatically tuned to achieve $P_{\text{jump}} = 0.4$.

Algorithm A3 2D Uniform on w, z. The algorithm uses a single two-dimensional proposal.

1. First, draw $u \sim U(-\sqrt{3}, \sqrt{3})$ and set $t' \leftarrow te^{\sigma_w u}$. Then draw $v \sim U(-\sqrt{3}, \sqrt{3})$ and set $r' \leftarrow re^{\sigma_z v}$. The proposal ratio is $\frac{t'r'}{tr}$.

It is hard to adjust two step-sizes σ_w and σ_z in one proposal. We used $\sigma_w = s_w \times 2.2 \times \frac{1.7}{2.4}$ and $\sigma_z = s_z \times 2.2 \times \frac{1.7}{2.4}$, where s_w and s_z are the standard deviations of $w = \log t$ and $z = \log r$, estimated during the burn-in. Here, 2.4 and 1.7 are optimal scales in 1D and 2D for the Gaussian kernel (Table 3.3), while 2.2 is the optimal scale for the uniform kernel in 1D (Tables 3.1 and 3.2).

Algorithm A4 1D Uniform on w, z with whitening transformation (3.3). Let $w \coloneqq \log t$ and $z \coloneqq \log r$. Let $\widehat{\Sigma}$ denote the estimated covariance matrix of (w, z) during burn-in. The algorithm consists of two MH steps.

1. Set $\left(\begin{smallmatrix} w \\ z \end{smallmatrix}\right) \leftarrow \left(\begin{smallmatrix} \log t \\ \log r \end{smallmatrix}\right)$ and $\left(\begin{smallmatrix} \tilde{w} \\ \tilde{z} \end{smallmatrix}\right) \leftarrow \widehat{\Sigma}^{-1/2} \left(\begin{smallmatrix} w \\ z \end{smallmatrix}\right)$. Draw $u \sim U(-\sqrt{3}, \sqrt{3})$ and set $\tilde{w}' \leftarrow \tilde{w} + \sigma_w u$ and $\tilde{z}' \leftarrow \tilde{z}$. Then set $\left(\begin{smallmatrix} w' \\ z' \end{smallmatrix}\right) \leftarrow \widehat{\Sigma}^{1/2} \left(\begin{smallmatrix} \tilde{w}' \\ \tilde{z}' \end{smallmatrix}\right)$ and $\left(\begin{smallmatrix} t \\ r \end{smallmatrix}\right) \leftarrow \left(\begin{smallmatrix} e^w \\ e^z \end{smallmatrix}\right)$. The proposal ratio is $\frac{t'r'}{tr}$.
2. Set $\left(\begin{smallmatrix} w \\ z \end{smallmatrix}\right) \leftarrow \left(\begin{smallmatrix} \log t \\ \log r \end{smallmatrix}\right)$ and $\left(\begin{smallmatrix} \tilde{w} \\ \tilde{z} \end{smallmatrix}\right) \leftarrow \widehat{\Sigma}^{-1/2} \left(\begin{smallmatrix} w \\ z \end{smallmatrix}\right)$. Draw $v \sim U(-\sqrt{3}, \sqrt{3})$ and set $\tilde{w}' \leftarrow \tilde{w}$ and $\tilde{z}' \leftarrow \tilde{z} + \sigma_z v$. Then set $\left(\begin{smallmatrix} w' \\ z' \end{smallmatrix}\right) \leftarrow \widehat{\Sigma}^{1/2} \left(\begin{smallmatrix} \tilde{w}' \\ \tilde{z}' \end{smallmatrix}\right)$ and $\left(\begin{smallmatrix} t \\ r \end{smallmatrix}\right) \leftarrow \left(\begin{smallmatrix} e^w \\ e^z \end{smallmatrix}\right)$. The proposal ratio is $\frac{t'r'}{tr}$.

The step-size parameters σ_w and σ_z were automatically tuned to achieve $P_{\text{jump}} = 0.4$.

Algorithm A5 1D Uniform on x, y. The algorithm consists of two one-dimensional MH steps on x and y.

1. Set $x \leftarrow \log(tr)$ and $y \leftarrow \log(t/r)$. Draw $u \sim U(-\sqrt{3}, \sqrt{3})$ and set $x' \leftarrow x + \sigma_x u$ and $y' \leftarrow y$. Then set $t' \leftarrow e^{\frac{x'+y'}{2}}$ and $r' \leftarrow e^{\frac{x'-y'}{2}}$. The proposal ratio is $\frac{t'r'}{tr}$.
2. Set $x \leftarrow \log(tr)$ and $y \leftarrow \log(t/r)$. Draw $v \sim U(-\sqrt{3}, \sqrt{3})$ and set $x' \leftarrow x$ and $y' \leftarrow y + \sigma_y v$. Then set $t' \leftarrow e^{\frac{x'+y'}{2}}$ and $r' \leftarrow e^{\frac{x'-y'}{2}}$. The proposal ratio is $\frac{t'r'}{tr} = 1$.

The step-size parameters σ_x and σ_y were automatically tuned to achieve $P_{\text{jump}} = 0.4$.

Algorithm A6 1D MirrorU on x, y. **A6a** 1D MirrorU1 on x, y. **A6b** 1D MirrorU$\frac{1}{2}$ on x, y. The algorithm consists of two one-dimensional MH steps on x and y.

1. Set $x \leftarrow \log(tr)$ and $y \leftarrow \log(t/r)$. Draw $x'|x \sim U(2\mu_x^* - x - \sigma_x\sqrt{3}, 2\mu_x^* - x + \sigma_x\sqrt{3})$ and set $y' \leftarrow y$. Then set $t' \leftarrow e^{\frac{x'+y'}{2}}$ and $r' \leftarrow e^{\frac{x'-y'}{2}}$. The proposal ratio is $\frac{t'r'}{tr}$.
2. Set $x \leftarrow \log(tr)$ and $y \leftarrow \log(t/r)$. Draw $y'|y \sim U(2\mu_y^* - y - \sigma_y\sqrt{3}, 2\mu_y^* - y + \sigma_y\sqrt{3})$ and set $x' \leftarrow x$. Then set $t' \leftarrow e^{\frac{x'+y'}{2}}$ and $r' \leftarrow e^{\frac{x'-y'}{2}}$. The proposal ratio is $\frac{t'r'}{tr} = 1$.

Here, μ_x^*, μ_y^* were set to the estimated means $\hat{\mu}_x, \hat{\mu}_y$ of x and y, respectively, and σ_x, σ_y were set to either $\hat{s}_x, \hat{s}_y$ (A6a) or $\frac{1}{2}\hat{s}_x, \frac{1}{2}\hat{s}_y$ (A6b), where $\hat{s}_x$ and $\hat{s}_y$ are the estimated standard deviations of x and y from the burn-in.

Algorithm A7 1D MirrorU on w, z with whitening transformation (3.3). Let $\widehat{\Sigma}$ denote the estimated covariance matrix of (w, z) during burn-in. The algorithm consists of two MH steps.

1. Set $\left(\begin{smallmatrix} w \\ z \end{smallmatrix}\right) \leftarrow \left(\begin{smallmatrix} \log t \\ \log r \end{smallmatrix}\right)$ and $\left(\begin{smallmatrix} \tilde{w} \\ \tilde{z} \end{smallmatrix}\right) \leftarrow \widehat{\Sigma}^{-1/2} \left(\left(\begin{smallmatrix} w \\ z \end{smallmatrix}\right) - \left(\begin{smallmatrix} \hat{\mu}_w \\ \hat{\mu}_z \end{smallmatrix}\right)\right)$. Draw $u \sim U(-\sqrt{3}, \sqrt{3})$ and set $\tilde{w}' \leftarrow -\tilde{w} + \sigma_w u$ and $\tilde{z}' \leftarrow \tilde{z}$. Then set $\left(\begin{smallmatrix} w' \\ z' \end{smallmatrix}\right) \leftarrow \widehat{\Sigma}^{1/2} \left(\begin{smallmatrix} \tilde{w}' \\ \tilde{z}' \end{smallmatrix}\right) + \left(\begin{smallmatrix} \hat{\mu}_w \\ \hat{\mu}_z \end{smallmatrix}\right)$ and $\left(\begin{smallmatrix} t \\ r \end{smallmatrix}\right) \leftarrow \left(\begin{smallmatrix} e^w \\ e^z \end{smallmatrix}\right)$. The proposal ratio is $\frac{t'r'}{tr}$.

2. Set $\binom{w}{z} \leftarrow \begin{pmatrix}\log t\\ \log r\end{pmatrix}$ and $\binom{\tilde{w}}{\tilde{z}} \leftarrow \widehat{\Sigma}^{-1/2}\left(\binom{w}{z} - \begin{pmatrix}\hat{\mu}_w\\ \hat{\mu}_z\end{pmatrix}\right)$. Draw $v \sim U(-\sqrt{3}, \sqrt{3})$ and set $\tilde{w}' \leftarrow \tilde{w}$ and $\tilde{z}' \leftarrow -\tilde{z} + \sigma_z v$. Then set $\binom{w'}{z'} \leftarrow \widehat{\Sigma}^{1/2}\binom{\tilde{w}'}{\tilde{z}'} + \begin{pmatrix}\hat{\mu}_w\\ \hat{\mu}_z\end{pmatrix}$ and $\binom{t}{r} \leftarrow \binom{e^w}{e^z}$. The proposal ratio is $\frac{t'r'}{tr}$.

The step-size parameters σ_w and σ_z were set to $\frac{1}{2}$.

Algorithm A8 MALA with preconditioning on w, z. The algorithm uses a single two-dimensional proposal.

1. Set $(w, z) \leftarrow (\log t, \log r)$. Draw $(w', z') \sim N(m(w, z), \varepsilon^2 A)$ where

$$m(w, z) \coloneqq (w, z) + \frac{\varepsilon}{2} A \nabla \log p(w, z|x).$$

   Set $(t', r') \leftarrow (e^{w'}, e^{z'})$. The proposal ratio is $\frac{N((w,z)|(w',z') + \frac{\varepsilon^2}{2} A \nabla \log p(w',z'|x), \varepsilon^2 A)}{N((w',z')|(w,z) + \frac{\varepsilon^2}{2} A \nabla \log p(w,z|x), \varepsilon^2 A)} \frac{t'r'}{tr}$.

The scalar step-size parameter ε was tuned manually to achieve the highest efficiency. The matrix A was set to $\frac{1}{(\det \widehat{\Sigma})^{1/2}} \widehat{\Sigma}$, where $\widehat{\Sigma}$ denotes an estimate of the target covariance from the burn-in, following Marshall and Roberts (2012). The proposal step-size is $\sigma_{w,z} = \varepsilon \mathrm{diag}(A^{1/2}) = \frac{\varepsilon}{(\det \widehat{\Sigma})^{1/4}} \mathrm{diag}(\widehat{\Sigma}^{1/2})$.

Algorithm A9 HMC on w, z. Let $L_{\max}$ be the upper bound on the number of leapfrog steps and let ε be the leapfrog step-size. Let $\widehat{\Sigma}$ denote the estimated covariance matrix of (w, z) during burn-in.

1. Set $(w, z) \leftarrow (\log t, \log r)$. Draw an auxiliary variable $\phi \sim N(0, \widehat{\Sigma})$. Set $(w', z') \leftarrow (w, z)$ and $\phi' \leftarrow \phi$. Draw $L \sim U\{1, \ldots, L_{\max}\}$. For $\ell = 1, \ldots, L$, (a) set $\phi' \leftarrow \phi' + \frac{\varepsilon}{2} \nabla \log p(w', z')$, (b) set $(w', z') \leftarrow (w', z') + \varepsilon \widehat{\Sigma}^{-1} \phi'$, and (c) set $\phi' \leftarrow \phi' + \frac{\varepsilon}{2} \nabla \log p(w', z')$. Then set $(t', r') \leftarrow (e^{w'}, e^{z'})$. The proposal ratio is $\frac{N(\phi'|0, \widehat{\Sigma})}{N(\phi|0, \widehat{\Sigma})} \frac{t'r'}{tr}$.

The parameters $L_{\max}$ and ε were tuned manually to achieve the highest efficiency.

Algorithm A10 HMC (Stan) on w, z. NUTS algorithm. See Hoffman and Gelman (2014) for description.

Algorithm A11 Manifold MALA on w, z. This algorithm uses a single two-dimensional proposal.

1. Set $(w, z) \leftarrow (\log t, \log r)$. Draw $(w', z') \sim N(m(w, z), \varepsilon^2 G^{-1})$ where

$$m(w, z) \coloneqq (w, z) + \varepsilon^2 \left(\frac{1}{2} G^{-1}(w, z) \nabla \log p(w, z|x) + \Omega(w, z)\right),$$

$$G(w, z) \coloneqq -\mathbf{E}_{p(x|w,z)} \nabla^2_{(w,z)} \log p(x|w, z) - \nabla^2_{(w,z)} \log p(w, z)$$

   (the Fisher information matrix for the likelihood plus the negative Hessian of the log prior density), and

$$\Omega(w, z) \coloneqq \frac{1}{2} G^{-1} \begin{pmatrix} \mathrm{tr}\big(G^{-1} \partial_w G\big) \\ \mathrm{tr}\big(G^{-1} \partial_z G\big) \end{pmatrix} - \sum_{j=w,z} (G^{-1} \partial_j G) G^{-1}_{\cdot,j}.$$

   Set $(t', r') \leftarrow (e^{w'}, e^{z'})$. The proposal ratio is $\frac{N((w,z)|m(w',z'), \varepsilon^2 G^{-1}(w',z'))}{N((w',z')|m(w,z), \varepsilon^2 G^{-1}(w,z))} \frac{t'r'}{tr}$.

The step-size parameter ε was tuned manually to achieve the highest efficiency.

Algorithm A12 Manifold HMC on w, z. Let $L_{\max}$ be the upper bound on the number of leapfrog steps and let ε be the leapfrog step-size. Let M be the number of fixed point iterations for the generalised leapfrog integrator from Girolami and Calderhead (2011). This algorithm uses a single two-dimensional proposal.

1. Set $(w, z) \leftarrow (\log t, \log r)$. Draw an auxiliary variable $\phi \sim N(0, G)$. Set $(w', z') \leftarrow (w, z)$ and $\phi' \leftarrow \phi$. Draw $L \sim U\{1, \ldots, L_{\max}\}$. For $\ell = 1, \ldots, L$,

   a) Set $\tilde{\phi} \leftarrow \phi'$. For $m = 1, \ldots, M$, set

   $$\tilde{\phi} \leftarrow \phi' + \frac{\varepsilon}{2}\left(\nabla \log p(w', z'|x) - \frac{1}{2}\mathrm{tr}(G^{-1}\nabla G) + \frac{1}{2}\tilde{\phi}^{\top}G^{-1}(\nabla G)G^{-1}\tilde{\phi}\right).$$

   Then set $\phi' \leftarrow \tilde{\phi}$.

   b) Set $(\tilde{w}, \tilde{z}) \leftarrow (w', z')$. For $m = 1, \ldots, M$, set

   $$(\tilde{w}, \tilde{z}) \leftarrow (w', z') + \frac{\varepsilon}{2}\left(G^{-1}(w', z') + G^{-1}(\tilde{w}, \tilde{z})\right)\phi'.$$

   Then set $(w', z') \leftarrow (\tilde{w}, \tilde{z})$.

   c) Set

   $$\phi' \leftarrow \phi' + \frac{\varepsilon}{2}\left(\nabla \log p(w', z'|x) - \frac{1}{2}\mathrm{tr}(G^{-1}\nabla G) + \frac{1}{2}\phi'^{\top}G^{-1}(\nabla G)G^{-1}\phi'\right).$$

   Set $(t', r') \leftarrow (e^{w'}, e^{z'})$. The proposal ratio is $\frac{N(\phi'|0, G(w', z'))}{N(\phi|0, G(w, z))}\frac{t'r'}{tr}$.

Parameters $L_{\max}$ and ε were tuned manually to achieve the highest efficiency, and M was fixed to 3.

Note that the parameters of the model are t and r, as are the state of the Markov chain. The transformed variables w and z or x and y are used to design efficient moves in the t-r space. Also note that A4 and A7 use generic logarithm and whitening transformations to deal with correlations and scale differences, while A5 and A6 use certain features of the model (namely the fact that the likelihood depends on tr only) to design efficient transformations or search direction.

3.2.5.3 Results

For each kernel, we simulated a Markov chain for 5×10^7 iterations, after a burn-in of 8×10^4 iterations. The estimates of the two marginal posterior means (and the 2.5th and 97.5th percentiles) were identical for all algorithms: 14.58 (10.5, 19.4) for t and 0.00361 (0.0025, 0.0051) for r, while the efficiency of the algorithms varied by nearly 40 folds (Table 3.6).

When the target's covariance structure was not taken into account, the efficiency achieved was less than 10%. The one-dimensional uniform proposals on t and r and on $\log t$ and $\log r$ (A1 and A2, respectively) were very inefficient, with $E \approx 5\%$, even less efficient than the two-dimensional uniform kernel (A3). This

Table 3.6: Efficiency of twelve kernels for the molecular clock dating problem. The scaling factor $c = \sigma/s$ is the ratio of the proposal standard deviation σ over the target standard deviation s.

	Kernel	Proposal step-size (σ)	Running time (s)	Time (t)			Rate (r)		
				c	P_{jump}	E	c	P_{jump}	E
A1	1D Uniform on t, r	Automatic	26	1.29	0.396	0.054	1.24	0.405	0.052
A2	1D Uniform on w, z	Automatic	28	1.46	0.403	0.055	1.33	0.388	0.054
A3	2D Uniform on w, z	$\sigma_w \leftarrow \hat{s}_w \times 2.2 \times \frac{1.7}{2.4}$, $\sigma_z \leftarrow \hat{s}_z \times 2.2 \times \frac{1.7}{2.4}$	22	1.76	0.206	0.079	1.55	0.206	0.078
A4	1D Uniform on w, z with whitening	Automatic	24	2.18	0.412	0.265	2.22	0.395	0.263
A5	1D Uniform on x, y	Automatic	26	2.15	0.401	0.284	2.16	0.399	0.211
A6a	1D MirrorU1 on x, y	$\sigma_x \leftarrow \hat{s}_x, \sigma_y \leftarrow \hat{s}_y$	38	1.07	0.621	0.970	0.96	0.646	0.621
A6b	1D MirrorU$\frac{1}{2}$ on x, y	$\sigma_x \leftarrow \frac{1}{2}\hat{s}_x, \sigma_y \leftarrow \frac{1}{2}\hat{s}_y$	38	0.48	0.762	1.168	0.46	0.766	0.411
A7	1D MirrorU$\frac{1}{2}$ on w, z with whitening	$\sigma_w \leftarrow \frac{1}{2}, \sigma_z \leftarrow \frac{1}{2}$	27	0.48	0.829	2.308	0.49	0.823	1.802
A8	MALA	Manual	30	1.48	0.617	0.600	1.47	0.617	0.591
A9	HMC	Manual	55	3.44	0.882	1.143	2.46	0.882	1.117
A10	HMC (Stan)	Automatic	1056	2.60	0.949	0.298	2.61	0.949	0.291
A11	Manifold MALA	Manual	162	n/a	0.631	0.571	n/a	0.631	0.568
A12	Manifold HMC	Manual	2727	n/a	0.939	1.715	n/a	0.939	1.670

Note.—s_w and s_z are the standard deviations of $w := \log t$ and $z := \log r$; s_x and s_y are the standard deviations of $x = \log(tr)$ and $y = \log(t/r)$, $\hat{\mu}$ denotes the estimate of the true mean μ, and $\hat{s}$ denotes the estimate of the true standard deviation s. The running time was averaged over 10 replications. The scaling factors for A6a and A6b were not exactly 1 and 0.5 because the variances estimated during burn-in involve inaccuracies. For manifold MALA and manifold HMC, the scaling factor depends on the current position of the Markov chain.

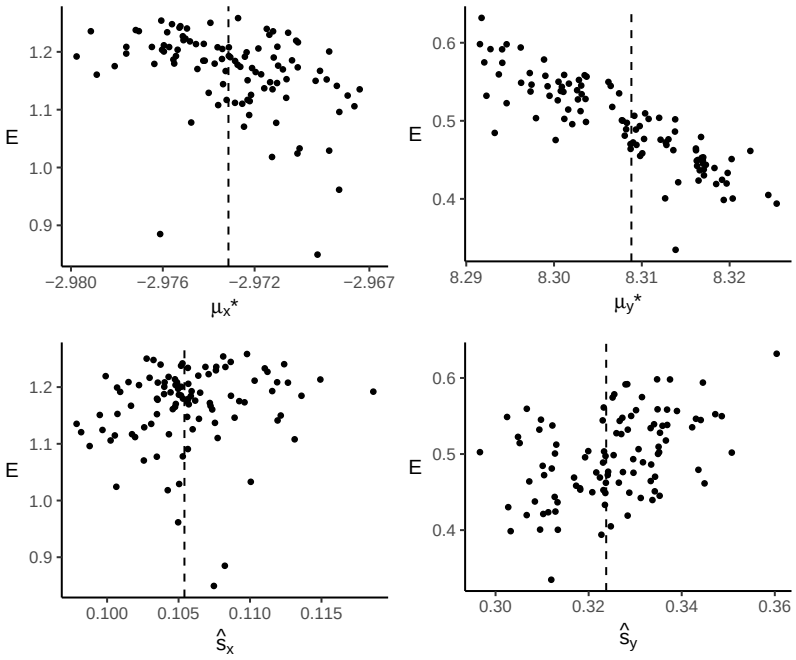


Figure 3.10: Efficiency (E) for estimating t (left column) and r (right column) over 100 replicate runs of the algorithm A6b (1D MirrorU$\frac{1}{2}$ on x, y) in the phylogenetic example, plotted as a function of $\mu_x^*, \mu_y^*, \hat{s}_x$ and $\hat{s}_y$ estimates obtained from the burn-in. The means (μ_x, μ_y) and standard deviations (s_x, s_y) were estimated using four rounds during the burn-in of 8×10^4 iterations, with each round consisting of 2×10^4 iterations. The estimates were then used to construct the Mirror move.

was not surprising as both pairs (t, r) and $(\log t, \log r)$ were highly correlated (correlation about -0.8), as expected from the fact that the likelihood depends on the product tr only. Removing the correlation and adjusting for the scale differences between the target variables via the whitening transformation (3.3) (A4) improved the efficiency significantly. An alternative and computationally cheaper way to reduce the correlation is to use the transformation $x = \log(tr)$ and $y = \log(t/r)$ (A5), based on our knowledge of the model. This reduced the correlation to -0.28, and yielded a similar efficiency boost as A4.

The MirrorU kernels A6 and A7 had a superior performance to the uniform kernel using the same transformation (A4 and A5) with no extra computational cost. However, the efficiency for these Mirror kernels depended on the estimated means and variances of the target from the burn-in. Independent simulations with different estimates of the target mean and variance suggested that the efficiency estimates were stable, with mean efficiency 1.165 for t and 0.497 for r, which were comparable to those in Table 3.6 (Figure 3.10). Note that for these Mirror kernels (A6 and A7), the efficiency for estimating the mean of t was always considerably higher than that of r. Moreover, from the replicated runs in Figure 3.10, the efficiency appeared to depend on the values of μ_y^* and $\hat{s}_y$. In particular, a decreasing trend in the efficiency as μ_y^* becomes larger was observed. Further investigation will be required to better understand these effects.

Both MALA (A8) and manifold MALA (A11) performed better than the uniform kernel (A4) ($E \approx 60\%$), but did not beat the MirrorU kernel. HMC (A9) and manifold HMC (A12) also gave super-efficient estimates, but at considerably greater computational and implementation cost. Stan (A10) did not perform as well as

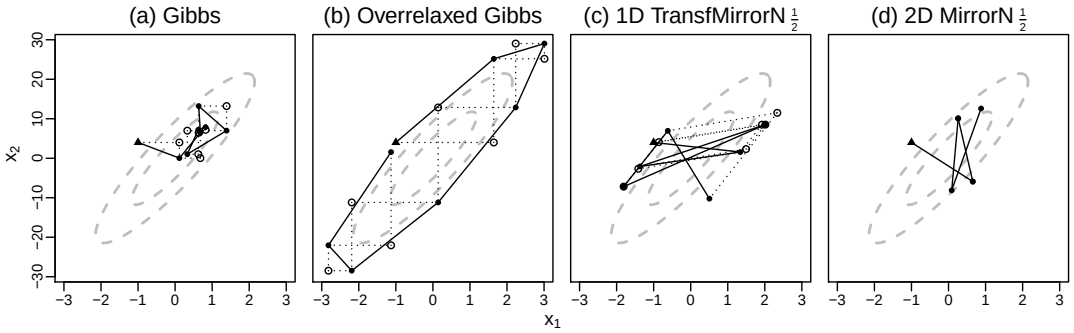


Figure 3.11: Sample path from a few steps of four algorithms for sampling from $N_2(0, \Sigma)$, with $\Sigma = \left(\begin{smallmatrix} 1 & 9 \\ 9 & 100 \end{smallmatrix}\right)$: (a) standard Gibbs sampler, (b) overrelaxed Gibbs sampler ($\alpha = -0.98$), (c) MH using 1D TransfMirrorN$\frac{1}{2}$ kernel, and (d) MH using 2D MirrorN$\frac{1}{2}$ kernel. The first three (a-c) consist of a sequence of two one-dimensional moves, while the last one (d) is a single two-dimensional move. The 1D TransfMirrorN$\frac{1}{2}$ kernel applies the MirrorN kernel $y_i'|y_i \sim N(2(\hat{\Sigma}^{-1/2}\mu^*)_i - y_i, \frac{1}{4})$, $i = 1, 2$, on $y = \hat{\Sigma}^{-1/2}x$, where $x = (x_1, x_2)$ is the target variable, and μ^* and $\hat{\Sigma}$ are estimated mean and covariance matrix of the target from the burn-in as described in Section 3.2.3. The 2D MirrorN$\frac{1}{2}$ kernel proposes $x'|x \sim N(2\mu^* - x, \frac{1}{4}\hat{\Sigma})$. Triangle = starting point $(-1, 4)$; filled circle = state of the Markov chain; empty circle = intermediate step (for the one-dimensional moves). Two ellipses enclose the 50% and 90% probability mass of the target.

other variants of HMC (A9, A12). In terms of efficiency per second, all variants of the MirrorU kernel outperformed manifold MALA, manifold HMC and Stan by a substantial margin (Table 3.6). Finally, although well-tuned MALA and HMC also gave good efficiency-per-time results, the need for high-order derivatives and manual tuning of the step-size parameters make them challenging to implement for general targets.

3.3 Discussion

3.3.1 Measures of performance

We have compared the mixing efficiency of different MH proposals as measured by the asymptotic variance for estimating a function of the target distribution (such as the mean or tail probability). Since the efficiency of the kernel may depend on the function or target (Mira, 2001), we have included several targets in our evaluation. We note that the ranking of the proposal kernels stays largely the same across all targets we evaluated, suggesting the existence of some general principles that may apply to fairly arbitrary targets.

Besides the mixing efficiency, another useful measure is the rate of convergence of Markov chains to the stationary distribution, such as δ_8 and $|\lambda|_2$ in Table 3.1. The convergence rate should affect the desired length of the burn-in. We consider the convergence rate to be less important than the mixing efficiency because the burn-in is typically a small fraction of the MCMC run, and because a kernel efficient for mixing tends to also be good for convergence. For example, the uniform kernel converges faster and mixes more efficiently than the Gaussian kernel (Table 3.1). It is also cheaper to simulate than the Gaussian kernel.

For the Mirror kernel, a small step-size gives estimates with lower asymptotic variance, but with slower convergence. It is thus preferable to use large steps during the burn-in for fast convergence, and small steps afterwards for fast mixing.

In practical MCMC applications, the computational and implementational costs are of major concern. We note that the computational cost may depend on hardware and software implementation details, as well as the specific inference problem. For example, certain one-dimensional moves may not change the likelihood and are thus more computationally efficient, such as the change to $y = t/r$ when $x = tr$ is fixed in the molecular clocking dating example. Our analyses of the logistic regression and the molecular clock dating examples suggest that the Mirror moves are simpler to implement and run faster than the manifold MALA and HMC kernels. We leave it to the algorithm developer to assess the computational cost of different proposals in their specific applications.

3.3.2 Comparison with other MCMC algorithms

Several MCMC algorithms have been proposed to improve mixing by suppressing the diffusive behaviour of the random walk MH proposals in which every iteration tends to take a small step in a random direction. We discuss a few that are related to our work.

The idea of proposing values on the other side of the distribution has appeared in the literature before. For instance, the overrelaxation method (Adler, 1981; Barone and Frigessi, 1990) is a Gibbs sampler for Gaussian conditionals that makes a move to the other side of each component's full conditional. The update for the component i is

$$x_i' = \mu_{i|-i} + \alpha(x_i - \mu_{i|-i}) + \sqrt{\sigma^2_{i|-i}(1-\alpha^2)}z, \qquad z \sim N(0,1),$$

where $\mu_{i|-i}$ and $\sigma^2_{i|-i}$ are conditional mean and variance of x_i given all other variables x_{-i}, and $\alpha \in (-1, 1)$ is a user-specified parameter. Choosing $\alpha \in (-1, 0)$ will make a move to the other side of the full conditional distribution of x_i. The Markov chain does not move to the other side of the target in one step, but instead moves along the density contour (Figure 3.11b), with higher-order autocorrelations oscillating between positive and negative signs (Figure 3.12). This results in cancellations of autocorrelations in (1.9), yielding a lower asymptotic variance than the standard Gibbs sampler in certain cases. By contrast, the Mirror kernel is a general MH proposal kernel that moves to the other side of the target in one step, giving a negative first-order autocorrelation (Figure 3.12). In addition, its implementation does not require the knowledge of the full conditionals. The mirror reflection of the current state through a centre point as an MH proposal kernel to induce negative correlations has been suggested by Tierney (1994, Section 4.3.3), who referred to it as an antithetic variate method, but theoretical analysis and empirical comparisons have been lacking.

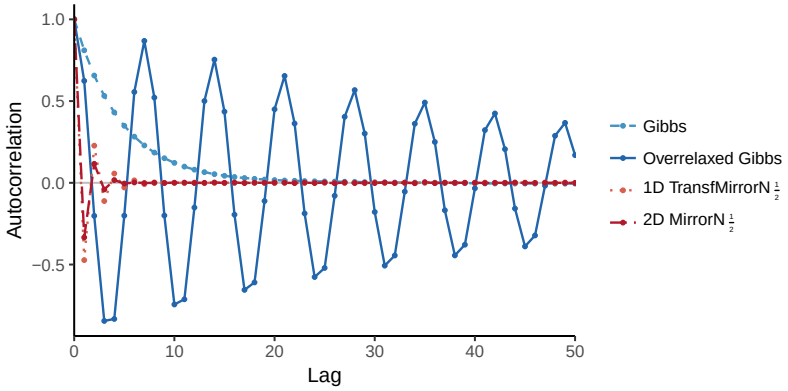


Figure 3.12: Autocorrelation function for the four proposal kernels of Figure 3.11, calculated using 10^6 iterations after a burn-in of 8,000 iterations. The efficiency for the four kernels is 0.104, 11.127, 2.784 and 2.122.

In the antithetic coupling method (Hammersley and Morton, 1956; Frigessi et al., 2000), two Markov chains are constructed with one to be the mirror reflection of the other. Combining the two chains yields a low-variance estimate. By contrast, the Mirror kernel introduces negative correlations within a chain rather than between chains.

HMC is another method that aims to propose a value away from the current position, in the direction of the peak of the target. A proposal is generated by simulating a trajectory of the so-called Hamiltonian dynamics. It requires computation of the first derivative of the log target density, and its parameters are difficult to tune to achieve good mixing. Automatic parameter tuning in HMC is currently a topic of research (Neal, 2011; Wang et al., 2013; Hoffman and Gelman, 2014). MALA is an MH algorithm that uses the Langevin proposal and can be viewed as a special case of HMC (Section 1.3.4.1). For the $N(\mu, s^2)$ target, choosing the step-size $\sigma = 2s$ gives the MALA update $x'|x \sim N(2\mu - x, 4s^2)$, which is equivalent to the MirrorN kernel using a fixed scaling factor of 2.

3.3.3 Parametrisation, variable transformation and efficiency for estimating different functions

Parametrisation of the target distribution or variable transformation is a useful approach for designing efficient MCMC samplers. We have illustrated this with several transformations that deal with correlations and/or scales of the target variables. We note that using different functions f in the Monte Carlo estimator (1.3) to evaluate MCMC mixing efficiency for the same target π is equivalent to using different target densities but the same function (such as the mean). Given that the ranking of kernels does not appear to be sensitive to the target used or the function to be estimated, a useful approach is to transform the target distribution into one for which efficient proposal kernels are known, and design proposals for the target variables accordingly.

To find a good proposal $q(x'|x)$ for the target $\pi_X(x)$, we may use a one-to-one transformation $y = T(x)$ so

Table 3.7: Efficiency for estimating the mean of three distributions. The step-size σ_x was adjusted to achieve $P^*_{\text{jump}} = 0.4$. The transformation $y = e^{-x}$ was used for Exp(1) and folded Gaussian, and $y = \Phi(x)$ was used for $N(0,1)$, with σ_y fixed at the optimal value for the $U(0,1)$ target (Table 3.1 and Yang and Rodríguez (2013, Table S1)).

Kernel	P_{jump}	E	E^2_π	ρ_1
Exp(1) target, Exp(1)-CDF transform				
Uniform ($\sigma_x = 2.5$)	0.408	0.161	0.589	0.705
TransfUniform ($\sigma_y = 2.8$)	1.000	1.298	2.283	−0.142
TransfBactrianTriangle ($m = 0.95, \sigma_y = 3.2$)	1.000	2.014	2.820	−0.410
TransfStrawHat ($a = 1, \sigma_y = 3.2$)	1.000	2.026	2.950	−0.474
folded Gaussian $N_+(0,1)$ target, Exp(1)-CDF transform				
Uniform ($\sigma_x = 2.3$)	0.392	0.213	0.259	0.643
TransfUniform ($\sigma_y = 2.8$)	0.839	1.075	0.755	−0.039
TransfBactrianTriangle ($m = 0.95, \sigma_y = 3.2$)	0.834	1.919	0.961	−0.322
TransfStrawHat ($a = 1, \sigma_y = 3.2$)	0.847	2.224	1.013	−0.394
$N(0,1)$ target, t_2-CDF transform				
Uniform ($\sigma_x = 2.2$)	0.405	0.275	0.879	0.561
TransfUniform ($\sigma_y = 2.8$)	0.832	0.959	1.961	0.020
TransfBactrianTriangle ($m = 0.95, \sigma_y = 3.2$)	0.836	1.592	2.471	−0.236
TransfStrawHat ($a = 1, \sigma_y = 3.2$)	0.846	1.680	2.548	−0.274
$N(0,1)$ target, logistic-CDF transform				
TransfUniform ($\sigma_y = 2.8$)	0.739	0.875	1.880	0.060
TransfBactrianTriangle ($m = 0.95, \sigma_y = 3.2$)	0.710	1.292	2.268	−0.134
TransfStrawHat ($a = 1, \sigma_y = 3.2$)	0.752	1.459	2.382	−0.191

that the resulting density $\pi_Y(y)$ resembles a simple density for which an efficient proposal $q(y'|y)$ is known. The X- and Y-chains are then coupled in the sense that if the initial states are the same with $y_0 = T(x_0)$ and if the same sequence of random numbers is used to run the two chains, then $y_n = T(x_n)$ for all $n \geq 1$. Estimating $\mathbf{E}_{\pi_X}(f(x))$ using the X-chain samples $x_{1:N}$ is then the same as estimating $\mathbf{E}_{\pi_Y}(f(T^{-1}(y)))$ using the Y-chain samples $y_{1:N}$. Thus finding an efficient proposal kernel for a given target is equivalent to finding a good variable transformation or parametrisation. It is then profitable to study the mixing efficiency for estimating various functions for simple targets such as the uniform distribution, where several highly efficient kernels are available (Table 3.1). Viewed in this light, our early observation that different proposal kernels with the same general shape have similar performances is equivalent to the observation that the ranking of proposals is insensitive to the target or function used.

As an example, consider the target $x \sim \text{Exp}(1/\mu)$ with mean μ. Then $y = e^{-x/\mu} \sim U(0,1)$. From Table 3.1, the uniform kernel[1] $y'|y \sim U(y - \frac{\sigma}{2}, y + \frac{\sigma}{2})$ with reflection and with $\sigma = 2.8$ achieves $E = 1.537$ for estimating $\mathbf{E}(y)$. Transformed onto the original variable x, the move is as follows. Set $y = e^{-x/\mu}$, sample $y'|y \sim U(y - \frac{\sigma}{2}, y + \frac{\sigma}{2})$ and reflect so that $y' \in (0,1)$. Then set $x' = -\mu \log y'$. The acceptance probability

[1] $U(0,1)$ has standard deviation $s = \frac{1}{2\sqrt{3}}$, so the kernel is $y' = y + \frac{\sigma}{s}u$ where $u \sim U(-\sqrt{3}, \sqrt{3})$, which is equivalent to $y' \sim U(y - \frac{\sigma}{2\sqrt{3}}\sqrt{3}, y + \frac{\sigma}{2\sqrt{3}}\sqrt{3}) \equiv U(y - \frac{\sigma}{2}, y + \frac{\sigma}{2})$.

is

$$\alpha(x, x') = \min\left(1, \mathrm{e}^{(x'-x)/\mu} \times \frac{\pi(x')}{\pi(x)}\right), \tag{3.5}$$

which equals 1. This algorithm gives $E = 1.298$ for estimating $\mathbf{E}(x) = \mathbf{E}(-\mu \log y)$ (Table 3.7). This is good performance since w was optimised for estimating $\mathbf{E}(y)$ instead of $\mathbf{E}(x)$. Even higher efficiency is achieved by using bimodal kernels such as BactrianTriangle or the new StrawHat on y (Table 3.7).

Next, we use the same transformation $y = e^{-x/\mu}$ to sample from the folded Gaussian $\pi(x) \propto \exp(-\frac{1}{2}x^2)$, $x > 0$, to estimate $\mathbf{E}(x) = 0.7979$. The acceptance probability is given by (3.5) although this does not equal 1. The uniform kernel on y gives $E = 1.075$ (Table 3.7). This is good because Exp(1) has only a passing resemblance to the folded Gaussian. Again bimodal kernels such as BactrianTriangle and StrawHat give even higher efficiency (Table 3.7).

Lastly, we consider two generic transformations for targets with support on the real line. We sample from $x \sim N(0, 1)$ using uniform, BactrianTriangle or StrawHat kernel on $y = h((x - \hat{\mu})/\hat{s})$ where h is the cumulative distribution function (CDF) of the t_2 or logistic distribution, and $\hat{\mu}$ and $\hat{s}$ are empirical estimates of the target's mean and standard deviation from the burn-in. For both transformations, the uniform kernel gives E close to 1 for estimating $\mathbf{E}(x) = 0$, whereas the BactrianTriangle and StrawHat kernels give $E > 1$ (Table 3.7).

Chapter 4

Species tree inference in the *Anopheles gambiae* mosquito species complex

Deep coalescence and introgression make it challenging to infer phylogenetic relationships among closely related species that arose through radiative speciation events. Despite numerous phylogenetic analyses and the availability of whole genomes, the phylogeny in the *Anopheles gambiae* species complex has not been confidently resolved. In this chapter, we performed Bayesian inference of the species tree under the multispecies coalescent (MSC) model (reviewed in Section 2.2), using over 80,000 coding and noncoding short segments (loci) extracted from the whole genome data of six members of this species complex from recent studies (Neafsey et al., 2015; Fontaine et al., 2015). The MSC model takes into account genealogical heterogeneity across the genome as well as uncertainty in the locus-specific gene trees. We obtained a robust estimate of the species tree that provides a more parsimonious interpretation of inversion and introgression events than the previously suggested species tree from Fontaine et al. (2015) (Sections 4.3.1 and 4.3.7). The concatenation approach used by Fontaine et al. (2015) was shown to produce artefactual species trees (Section 4.3.2). These findings were confirmed by simulation informed by the real data (Section 4.3.3). To infer gene flow between pairs of species, we analysed various data subsets of species triplets using the MSC model together with the fact that introgression reduces the species divergence times (Section 4.3.5). We also explicitly estimated the gene flow rates for different chromosomal regions under the isolation-with-migration (IM) model (reviewed in Section 2.3). Our results highlight the importance of accommodating incomplete lineage sorting and introgression in phylogenomic analyses of species that arose through recent radiative speciation events.

4.1 Introduction

The *Anopheles gambiae* species complex is a group of sub-Saharan African mosquito species that is comprised of at least eight recognised species and includes major malaria vectors in Africa. These species

are morphologically nearly indistinguishable but are genetically distinct, and have different ecological traits and reproductive behaviours such as range, habitats, resting and feeding preferences and vectorial capacity (Coluzzi et al., 1979; White et al., 2011). Three members, *A. gambiae*, *A. coluzzii* and *A. arabiensis*, are ecologically most similar, with large overlapping geographical ranges across sub-Saharan Africa and are principal vectors of the *Plasmodium* parasites (Wiebe et al., 2017). *A. gambiae* and *A. coluzzii* are closely related sibling species that are highly anthropophilic and are responsible for the majority of malaria transmission in Africa, while *A. arabiensis* is a less dominant vector (Takken and Verhulst, 2013). Other species in the complex have much more restricted geographical distributions (Sinka et al., 2012; Wiebe et al., 2017). *A. melas* and *A. merus* are salt-tolerant species that breed in brackish coastal waters of eastern and western Africa, respectively. They have similar ecological and morphological characteristics and are minor vectors (Coluzzi et al., 1979). *A. quadriannulatus* is zoophilic and plays no role in malaria transmission despite its vector competence for *Plasmodium falciparum* (Takken et al., 1999).

Inference of the evolutionary relationships among the members of the *A. gambiae* species complex is a fundamental step towards identifying genomic changes associated with epidemiologically important traits and developing effective malaria control strategies. However, this task has been extremely challenging. First, rapid succession of speciation events in the species complex combined with large population sizes of ancestral species has caused widespread genealogical heterogeneity, or incomplete lineage sorting (ILS), across the genome (Ayala and Coluzzi, 2005; Fontaine et al., 2015). Second, introgression is prevalent in autosomal regions of the genome, particularly among the three major vector species *A. gambiae*, *A. coluzzii* and *A. arabiensis* (Besansky et al., 2003; Wang-Sattler et al., 2007; O'Loughlin et al., 2014). Third, different genomic regions, such as the X chromosome, the autosomes and inversion regions on chromosomes 2L and 3L, show systematically different phylogenetic relationships, possibly due to complex effects of chromosomal inversion, introgression and natural selection (Slotman et al., 2005; Ayala et al., 2017). Inversions, both fixed and polymorphic, are prevalent across the genome (Coluzzi et al., 2002) and are shown to be associated with adaptation in different ecological habitats (Ayala et al., 2017). As a result of those complicating factors, different types of molecular data support different species phylogenies. For instance, the close relationship between *A. arabiensis* and *A. gambiae*+*A. coluzzii* is supported by sequence data in the autosomal regions, the Y chromosome (Hall et al., 2016) and the mitochondrial genome (Fontaine et al., 2015), but not by chromosomal inversions (Coluzzi et al., 1979, 2002) or the X chromosome (Fontaine et al., 2015). Similarly ecology and morphology group *A. merus* with *A. melas*, but this sister relationship is not supported by genomic sequences and chromosomal inversions (Coluzzi et al., 1979, 2002).

Fontaine et al. (2015) provided the first phylogenomic analysis of the species complex using complete nuclear and mitochondrial genomes of six members: *A. gambiae*, *A. coluzzii*, *A. arabiensis*, *A. merus*, *A. melas* and *A. quadriannulatus*. The maximum likelihood (ML) phylogenies from 50-kb non-overlapping windows sliding along the genome showed widespread heterogeneity in the genealogical history across the genome.

In particular, the X chromosome and autosomes produced drastically different phylogenies. The authors provided evidence that the majority tree for the X chromosome represents the true species branching order, while extensive introgressions have altered the autosomal phylogeny. However, their sliding-window approach fits one tree to all sites in the large window and ignores the ILS. We also refer to this approach as concatenation. For closely related species formed through radiative speciations, concatenation is well-known to be unreliable (Edwards et al., 2016): it may be inconsistent and converge to a wrong species tree when the amount of data increases (Kubatko and Degnan, 2007; Roch and Steel, 2015). The same genomic data were analysed using a phylogenetic network model with coalescent that captures both ILS and gene flow between species (Wen et al., 2016a,b), producing different phylogenies. Nevertheless, those analyses treated inferred gene trees as input data and ignore information in gene-tree branch lengths. As a result, they may lack power and fail to account for uncertainty in the gene trees due to limited phylogenetic information at each locus (Xu and Yang, 2016).

Here, we compiled datasets consisting of loosely linked short genomic segments (100-1,000 bases in length, at least 2kb apart), referred to as *loci*, from the genomes of the six members of the *A. gambiae* species complex (Neafsey et al., 2015; Fontaine et al., 2015) and performed Bayesian species tree analysis using the program BPP (Yang, 2015), which implements the multispecies coalescent (MSC) model (Rannala and Yang, 2003; Yang and Rannala, 2014; Rannala and Yang, 2017). Our approach explicitly accommodates gene-tree heterogeneity across loci and makes full use of information in the sequence data, including information about coalescent times or gene tree branch lengths, and fully accounts for uncertainty in the gene trees. We compiled and analysed separate datasets for the coding and noncoding regions of the genome. We also performed concatenation analysis using RAxML (Stamatakis, 2014) on these datasets as in Fontaine et al. (2015). We used simulation to understand the different estimates of the species tree from the coalescent and concatenation analyses. Since BPP does not account for gene flow between species, we used the ML program 3s (Zhu and Yang, 2012; Dalquen et al., 2017) to test for migration between species and to explicitly estimate the migration rates. The program implements ML inference under the isolation-with-migration (IM) model (Hey and Nielsen, 2004; Hey, 2010), which extends the MSC model by allowing gene flow between the two ingroup species, with a third species used as an outgroup. However, 3s assumes a fixed species tree and is currently limited to three sequences from at most three species. Unlike previous studies, our analyses of genome-wide data lead to a robust conclusion about the species phylogeny of this species complex, thus providing a framework for studying the evolution of ecological and epidemiological traits in this medically relevant group of mosquitoes. As an example, we discuss an implication of our species tree on the evolution of 2La inversion polymorphism, which is an important epidemiological trait associated with susceptibility to *Plasmodium* infection in natural mosquito populations (Riehle et al., 2017).

4.2 Methods

4.2.1 Datasets

We obtained the whole genome alignment from Fontaine et al. (2015) (doi:10.5061/dryad.f4114) for six species in the *A. gambiae* species complex: *A. gambiae* (G), *A. coluzzii* (C), *A. arabiensis* (A), *A. melas* (L), *A. merus* (R) and *A. quadriannulatus* (Q), as well as the *A. gambiae* PEST reference genome and two Pyretophorus outgroup species (*A. christyi* and *A. epiroticus*). For each of the six ingroup species, there are two genomes, one from a laboratory colony (reference genome) and another from field-collected individuals (non-reference genome). We used twelve whole genomes for the six ingroup species, and *A. christyi* (O) genome as an outgroup, excluding *A. gambiae* PEST and *A. epiroticus* genomes from our analysis. There are thus 12 sequences per locus, or 13 if the outgroup is included. The original alignment was partitioned into 2L, 2R, 3L, 3R and X chromosomal arms. We further separated out three main inversion regions 2La, 3La, Xag (in chromosomes 2L, 3L and X, respectively) using breakpoint coordinates from Table S11 in Fontaine et al. (2015), resulting in ten chromosomal regions: 2L1, 2La (the inversion region on 2L with coordinates 20.5-42.1 Mb), 2L2, 2R, 3L1, 3La (the inversion region on 3L with coordinates 14.5-35.6 Mb), 3L2, 3R, Xag (the inversion region about 14.8 Mb on the distal end of the X chromosome) and X2 (the pericentromeric region of the X chromosome with coordinates 14.8-24 Mb) (Table 4.1). The distal end of the X chromosome contains a small region of about 21 kb outside of the Xag inversion, which may not be very informative about the species tree, and was combined into the Xag region here.

The MSC model implemented in BPP and 3S assumes free recombination among loci and no recombination within a locus. Thus ideal loci for this kind of analysis are short genomic segments that are far apart so that recombination within a locus can be ignored while recombination between loci is so common that the different loci have nearly independent histories (Burgess and Yang, 2008; Lohse et al., 2011).

We used the gene set annotation of *A. gambiae* PEST strain (AgamP3 assembly) from VectorBase to split the alignment into coding and noncoding regions. For the noncoding regions, we split the alignment for each chromosomal region into smaller segments, referred to as loci, using the ambiguous nucleotide character (N) as breakpoints. Each locus was between 100 and 1,000 bases and had fewer than 50% gaps, and two consecutive loci were at least 2 kb apart. In a preliminary analysis, we also compiled data with a minimum gap of 10 kb between loci and the results were very similar. In addition, linkage disequilibrium for populations of *A. gambiae* and *A. coluzzii* is estimated to decay to <5% within 1 kb (The *Anopheles gambiae* 1000 Genomes Consortium, 2017). Thus we used 2 kb, which also preserved more loci. There were 57,592 noncoding loci in total (Table 4.1). Manual inspection revealed a considerable number of misaligned regions in the original whole genome alignment from Fontaine et al. (2015). We thus realigned all loci using MAFFT (Katoh and Standley, 2013), using the iterative refinement method (L-INS-i option). This appeared to fix the alignment errors. After removing gaps, each locus had between 11 to 973 sites

Table 4.1: Number of loci in each chromosome region in non-coding and coding datasets.

Dataset	Chromosome region										Total
	2L1	2La	2L2	2R	3L1	3La	3L2	3R	Xag	X2	
Non-coding	4134	6732	2330	17027	2496	6280	1823	14323	1825	622	57592
Coding	2223	2776	1362	6849	983	1998	764	4977	1179	394	23505

(median 195). The number of parsimony-informative sites ranged from 0 to 229 (median 15). For the coding regions, we also required each locus, which is a part of an exon, to have length at least 100, and contain fewer than 50% gaps. But unlike the noncoding loci, we did not constrain the maximum length of each locus or the minimum distance between loci. There were 23,505 coding loci in total (Table 4.1). Each locus ranged from 52 to 6,541 sites (median 210). The number of parsimony-informative sites ranged from 0 to 403 (median 6). All processing of the original genome alignment data was done using custom python scripts.

4.2.2 Species tree estimation using BPP and concatenation

We inferred the species tree among the six ingroup species using two methods: (1) Bayesian MSC-based method implemented in BPP v.4.0 (Rannala and Yang, 2003; Yang and Rannala, 2014; Yang, 2015; Rannala and Yang, 2017) using the JC model (Jukes and Cantor, 1969) for sequence likelihood given the gene tree (as this this is the only model currently implemented in BPP) and (2) concatenation and ML under the GTR+Γ_4 model using RAxML v.8.2 (Stamatakis, 2014). To reduce the computation cost and to explore the heterogeneity in the species relationships across the genome, we partitioned the data into blocks of 100 loci in each chromosomal region, resulting in 582 blocks for the noncoding data, and 239 blocks for the coding data. Each block was analysed separately, treated as 100 loci with independent genealogical histories by BPP and as one super-sequence by RAxML (Figure 4.1).

The MSC model accommodates ancestral polymorphism and deep coalescence, and the likelihood implementation in BPP accounts for phylogenetic uncertainties at each locus (Section 2.2.1). The parameters under the MSC include $\Theta = (\tau_i, \theta_i)$, where τ_i is the species divergence time, θ_i is the population size parameter. Inverse gamma priors were assigned on τ and θ parameters. For the noncoding data, we used $\theta \sim \text{InvG}(3, 0.04)$ for all populations, which has mean 0.02, and the root age $\tau_0 \sim \text{InvG}(3, 0.2)$, which has mean 0.1. Given τ_0, species divergence times for non-root nodes were given a uniform distribution on the

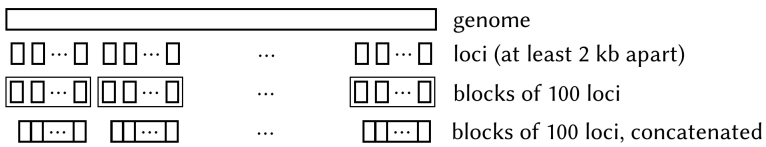


Figure 4.1: Diagram illustrating how the loci extracted from the whole-genome data were grouped into blocks of 100. Each block was analysed separately, and one species tree was inferred for each block.

interval $(0, \tau_0)$, generated from the symmetric Dirichlet distribution (Yang and Rannala, 2010). These θ and τ parameters are in the units of the expected number of mutations per site. To convert these parameters to actual times (before present) and actual population sizes, we used the mutation rate estimates for *Drosophila*: 2.8×10^{-9} (Keightley et al., 2014) and 5.5×10^{-9} (Schrider et al., 2013) mutations per site per generation, with 11 generations per year (The *Anopheles gambiae* 1000 Genomes Consortium, 2017). Thus the population sizes have prior mean of about 0.91 or 1.79 million individuals, and the root divergence time has prior mean of about 1.65 or 3.25 million years. For the coding data, we used $\theta_i \sim \text{InvG}(3, 0.008)$ and $\tau_0 \sim \text{InvG}(3, 0.04)$, which have five times smaller means than for the noncoding data. The species tree prior was the uniform distribution over rooted trees (Yang and Rannala, 2014). We initially estimated the species tree with θ integrated out analytically as this improved the mixing property of the algorithm (A01 analysis in Yang (2015)). We then estimated the population size parameters and species divergence times for each of the most likely species trees (A00 analysis in Yang (2015)).

For A01 analysis, MCMC was run for 2×10^6 iterations after 4×10^4 iterations of burn-in. Samples were recorded every 20 iterations. For each block of loci, two independent runs were performed using different starting trees. Convergence was assessed mostly by checking for consistency between runs in posterior probabilities for species trees. If the MAP trees from the two runs were the same, we required their posterior probabilities to differ by ≤ 0.3, while if the MAP trees were different, we required the mean absolute difference between the two posterior distributions of the species tree probabilities to be ≤ 0.3. We then combined the samples from the two runs to produce a posterior summary. Otherwise, we repeated the two runs until convergence was achieved.

For parameter estimation on a fixed species tree (A00 analysis), we also included the outgroup species (*A. christyi*) in the data since the estimated parameters will be used later in simulation experiments. We performed ten independent runs of MCMC, each with 10^6 iterations after a burn-in of 4×10^4 iterations.

For the concatenation analysis, we merged each block of 100 loci into a single alignment and then ran RAxML. We also split each alignment into two subsets, containing only either reference genomes or genomes from resequencing natural population samples. We used the GTR+Γ_4 model and performed 100 independent runs with random starting trees (option -N 100) to infer the ML tree. The number of bootstrap replicates was 100.

4.2.3 Generation and analysis of simulated datasets

Our BPP analysis suggested tree ii for the autosomes and tree xi for the Xag region of the X chromosome (Figures 4.2 and 4.3, Table 4.2), while the sliding-window analysis of Fontaine et al. (2015) favoured trees i and ix, respectively. To investigate those differences, we used the MCCOAL program in BPP v.4.0 to simulate two datasets using trees ii and xi, each with ten replicates. We performed BPP and concatenation/RAxML analyses on the simulated datasets in the same way as for the real datasets. Note that our goal is to

understand the effect of ILS on the performance of the MSC and concatenation approaches, using the MSC model to simulate data seemed appropriate. Although other types of models such as forward-time frequency-based models may be used instead of the MSC model, we expect the choice of the simulator to have a minor effect on the results as long as they can generate data with ancestral polymorphisms and ILS.

For tree ii, we simulated 6,464 loci under the GTR+Γ_4 model, each of length 200. We used the posterior means of τs and θs in the MSC model obtained from the real 6,464 loci from chromosome 2L, exclusive of 2La region (Figure 4.9). The parameters of the GTR+Γ_4 model for evolving sequences given the gene tree were allowed to vary among loci. For each locus, the base frequencies $\pi = (\pi_T, \pi_C, \pi_A, \pi_G)$ were generated from a Dirichlet distribution $\pi \sim \text{Dirichlet}(\alpha_T, \alpha_C, \alpha_A, \alpha_G)$ with parameters $(\alpha_T, \alpha_C, \alpha_A, \alpha_G) = (20.49, 21.22, 20.46, 20.97)$. These were the ML estimates obtained when the Dirichlet distribution was fitted to the observed base frequencies. The exchangeability parameters $q = (a, b, c, d, e, f)$ for the GTR model (Yang, 1994a) were also generated from a Dirichlet distribution $q \sim \text{Dirichlet}(\alpha_a, \alpha_b, \alpha_c, \alpha_d, \alpha_e, \alpha_f)$ with parameters $(\alpha_a, \alpha_b, \alpha_c, \alpha_d, \alpha_e, \alpha_f) = (7.59, 3.23, 2.95, 2.93, 2.93, 7.57)$, estimated by fitting the Dirichlet distribution to the RAxML estimates of q for the data at each locus. The overall rate for each locus was generated from G(5, 5). The shape parameter $\alpha = 5$ was based on fitting the gamma distribution G(α, β) to locus-wise estimates of the tree lengths from RAxML. The alignment of sequences at each locus is not always informative enough to estimate the α parameter in the GTR+Γ_4 model. Instead, we generated α for each locus from G(20, 4), with mean 5. Similarly for tree xi, we simulated 1,825 loci each of length 200, where the species-tree parameters (θs and τs) were estimated using the real 1,825 loci from the Xag region (Figure 4.9). We used $\pi \sim \text{Dirichlet}(20.49, 21.22, 20.46, 20.97)$ and $q \sim \text{Dirichlet}(7.72, 3.16, 3.24, 3.18, 2.69, 7.45)$. Other parameters were the same as for tree ii.

4.2.4 Likelihood ratio test of gene flow and ML estimation of migration rates

Since BPP currently does not allow gene flow between species, we performed a separate analysis using an isolation-with-migration (IM) model implemented in the maximum likelihood program 3s (Zhu and Yang, 2012; Dalquen et al., 2017), reviewed in Section 2.3.1. The current implementation works with three species (1, 2 and 3) assuming the species tree ((1, 2), 3), and only allows gene flow between the two ingroup species (1 and 2) with migration rates M_{12} and M_{21}, while species 3 is used as an outgroup. Here, $M_{ij} = N_j m_{ij}$ is the expected number of individuals migrating from species i to j per generation. We tested for gene flow between *A. gambiae* and *A. arabiensis*, and between *A. merus* and *A. quadriannulatus*, as suggested by conflicting species tree estimates from the BPP analysis. We used *A. christyi* (O) as the outgroup (species 3). Thus we analysed two species triplets, GAO and RQO, with a fixed species tree ((GA)O) and ((RQ)O), respectively. As *A. christyi* is a very distant outgroup, we additionally analysed GAL, GAR and RQL triplets, using *A. melas* or *A. merus* as the outgroup. The noncoding loci were used,

and different chromosomal arms and inversion regions were analysed separately, as well as the entire chromosome arms and all autosomal regions as a whole. For each locus, we chose one of the following three data configurations uniformly at random: 123, 113 and 223. Here, 123 means one sequence from each of the three species, and 113 means two sequences from species 1 and one sequence from species 3, etc. When one sequence from a species was used, it was always from the reference genome. We estimated parameters under two models, M0 (no gene flow) and M2 (gene flow), and compared them using a likelihood ratio test (LRT). Model M0 is the MSC model (Section 2.2.1). It has two species divergence times (τ_0 for the root, and τ_1 for the two ingroup species) and four effective population sizes: $\theta_1, \theta_2, \theta_4, \theta_5$ (for the two extant populations, 1 and 2, and for two ancestral populations, 4 for the root and 5 for the ingroup species); see Figure 2.3. There is no population size parameter for the outgroup (θ_3) since we always used one outgroup sequence. Model M2 is the IM model and has two additional parameters M_{12} and M_{21}. Integration over the two coalescent times in the gene trees in the likelihood calculation used Gaussian quadrature with 32 points (Yang, 2002). We ran the program twice for each analysis, and the run with a higher log-likelihood value was used.

4.3 Results

4.3.1 Species branching order varies systematically among different parts of the genome

We compiled 57,592 noncoding and 23,505 coding loci using the whole genome alignment from Fontaine et al. (2015) for six species in the *A. gambiae* species complex: *A. gambiae* (G), *A. coluzzii* (C), *A. arabiensis* (A), *A. melas* (L), *A. merus* (R) and *A. quadriannulatus* (Q). Our analysis assumed the molecular clock so that rooted trees can be inferred without the outgroup but we also constructed datasets that include *A. christyi* (O) as the outgroup. The genome was partitioned into ten chromosomal regions (Table 4.1). For computational tractability of BPP and to explore the heterogeneity in the species relationships across the genome, we split each dataset into blocks of 100 loci, so that there are 582 noncoding blocks and 238 coding blocks. Each block was analysed using BPP to calculate the posterior probabilities for species trees (A01 analysis in Yang (2015)).

Systematically different species trees were inferred from BPP for different genomic regions, for the dataset without the outgroup (Figure 4.2, Table 4.2) and with the outgroup (Figure 4.3). As in Fontaine et al. (2015), we recognise four regions of the genome with distinct phylogenetic relationships: (1) the majority of the autosomes and the pericentromeric region of the X chromosome, (2) the 2La inversion region, (3) the 3La inversion region and (4) the Xag inversion region. In most parts of the autosomal genome (2L1, 2L2, 2R, 3L1, 3L2 and 3R), the maximum posterior probability (MAP) species tree was tree ii: (R(L(Q(A(GC))))), and less commonly, tree iii: (L(R(Q(A(GC))))) (Figure 4.2). The results were highly consistent between the

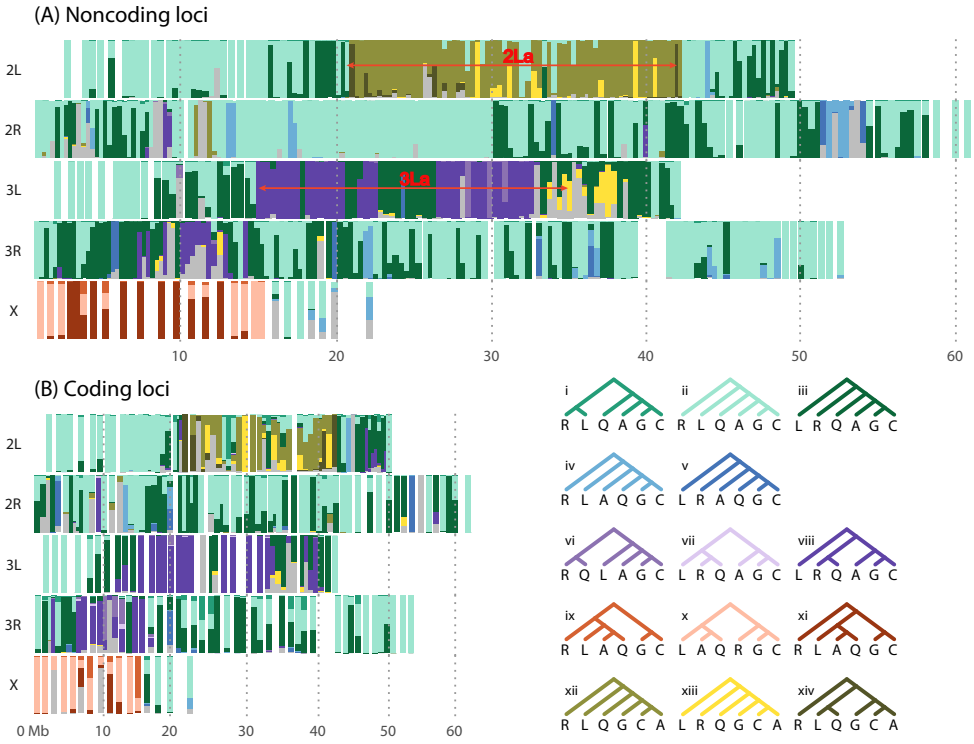


Figure 4.2: Posterior probabilities of species trees inferred using BPP for 100-locus blocks of (A) noncoding and (B) coding loci. The y-axis scales from 0 to 1. The x-axis provides approximate chromosomal coordinates of blocks, where the position for each block was taken to be the average of the starting positions in the AgamP3 coordinates over all loci within the block.

noncoding and coding data. Tree ii was also the autosomal tree obtained in studies using coalescent-based methods (Wen et al., 2016a,b). By contrast, the most common ML tree on the autosomes in the sliding-window analysis of Fontaine et al. (2015) was tree i, with the (RL) clade. This tree had near-zero posterior for almost all blocks in our analysis (Figure 4.2). Note that species trees i, ii, and iii are three phylogenetic resolutions for R, L and the clade (Q(A(GC))) around a very short branch at the root (Figure 4.9A&C). We demonstrated in Section 4.3.3 that tree i, which is more balanced than tree ii, may be an artefact of the sliding-window approach used in Fontaine et al. (2015).

For the Xag region, the MAP tree from BPP for the noncoding data was predominantly tree xi: (R((L(AQ))(GC))). For the coding data, it was mostly tree x: ((L(AQ))(R(GC))), while tree xi was the MAP tree for only two blocks (Figure 4.2 and Table 4.2). By contrast, previous studies (Fontaine et al., 2015; Wen et al., 2016a,b) inferred tree ix: ((R(L(AQ)))(GC)) for the Xag region, with (GC) branching first. This tree was rarely supported in the BPP analysis of either noncoding or coding dataset, and we argued below that this was a result of a bias in the sliding-window analysis. Note that trees ix, x and xi are three resolutions of the clades R, (GC) and (L(AQ)) around a very short internal branch at the root of the species complex (Figure 4.9B&D). The noncoding loci are more divergent and more informative about the species phylogeny, and are less affected by natural selection than the coding loci. Later, we reviewed other lines of evidence supporting tree xi, instead of trees ix or x, as the true species tree (Section 4.3.4).

Table 4.2: Proportions of inferred species trees (with minimum, median and maximum support values for each inferred tree in parentheses) for noncoding and coding loci from BPP and RAxML by chromosomal regions.

Tree	Non-coding loci			Coding loci		
	BPP	RAxML (reference)	RAxML (non-reference)	BPP	RAxML (reference)	RAxML (non-reference)
Autosomes (excluding 2La and 3La)						
i	0	0.4118 (0.32, 0.76, 0.99)	0.4965 (0.25, 0.73, 1.00)	0.0057 (0.42, 0.42, 0.42)	0.1667 (0.35, 0.63, 0.91)	0.2069 (0.25, 0.57, 0.87)
ii	0.5882 (0.48, 1.00, 1.00)	0.2188 (0.33, 0.69, 1.00)	0.2400 (0.41, 0.73, 1.00)	0.4655 (0.43, 0.97, 1.00)	0.4655 (0.26, 0.77, 1.00)	0.5000 (0.36, 0.79, 1.00)
iii	0.2776 (0.31, 0.92, 1.00)	0.0141 (0.44, 0.62, 0.73)	0.0282 (0.32, 0.58, 0.77)	0.3678 (0.39, 0.82, 1.00)	0.0402 (0.52, 0.62, 0.72)	0.0632 (0.39, 0.58, 0.85)
iv	0.0306 (0.49, 0.94, 1.00)	0.0141 (0.62, 0.71, 0.91)	0	0.0115 (0.82, 0.89, 0.95)	0.0345 (0.43, 0.63, 0.77)	0
v	0.0188 (0.54, 0.80, 0.99)	0	0	0.0230 (0.38, 0.69, 0.99)	0.0057 (0.56, 0.56, 0.56)	0
vi	0	0.0965 (0.34, 0.57, 0.99)	0.1388 (0.31, 0.64, 1.00)	0.0115 (0.44, 0.60, 0.76)	0.0460 (0.40, 0.60, 0.89)	0.0632 (0.30, 0.44, 0.92)
vii	0	0.0212 (0.44, 0.62, 0.88)	0.0259 (0.29, 0.57, 0.83)	0	0	0
viii	0.0353 (0.38, 0.76, 1.00)	0.0024 (0.28, 0.28, 0.28)	0.0094 (0.38, 0.49, 0.68)	0.0690 (0.35, 0.93, 0.97)	0.0402 (0.27, 0.48, 0.69)	0.0690 (0.32, 0.61, 0.85)
ix	0	0	0	0	0	0
x	0	0	0	0	0	0
xi	0	0	0	0	0	0
xii	0.0047 (0.89, 0.94, 1.00)	0.0024 (0.45, 0.45, 0.45)	0	0	0.0115 (0.55, 0.63, 0.71)	0
xiii	0.0094 (0.40, 0.56, 0.91)	0.0024 (0.58, 0.58, 0.58)	0	0	0	0.0057 (0.79, 0.79, 0.79)
xiv	0	0	0	0	0	0
2La						
i	0	0	0	0.0357 (0.77, 0.77, 0.77)	0.1071 (0.43, 0.63, 0.72)	0
ii	0.0588 (0.53, 0.81, 0.99)	0	0	0.1071 (0.29, 0.44, 0.77)	0.1071 (0.62, 0.66, 0.69)	0
iii	0.0147 (0.53, 0.53, 0.53)	0	0	0.0357 (0.66, 0.66, 0.66)	0	0
iv	0	0	0	0	0	0
v	0	0	0	0	0	0
vi	0	0	0	0	0	0
vii	0	0.0147 (0.75, 0.75, 0.75)	0	0	0	0
viii	0	0	0	0	0.0357 (0.76, 0.76, 0.76)	0
ix	0	0	0	0	0	0
x	0	0	0	0	0	0
xi	0	0	0	0	0	0
xii	0.8088 (0.51, 0.97, 1.00)	0.0441 (0.47, 0.49, 0.67)	0.1471 (0.28, 0.60, 0.85)	0.4643 (0.15, 0.71, 1.00)	0.1429 (0.28, 0.47, 0.72)	0.3571 (0.41, 0.69, 0.91)
xiii	0.0588 (0.57, 0.83, 0.98)	0	0.0147 (0.29, 0.29, 0.29)	0.1429 (0.43, 0.55, 0.99)	0	0.0357 (0.49, 0.49, 0.49)
xiv	0.0441 (0.55, 0.92, 0.94)	0.1029 (0.36, 0.60, 0.86)	0.0735 (0.24, 0.65, 0.76)	0.1429 (0.63, 0.79, 1.00)	0.1429 (0.35, 0.50, 0.54)	0.1071 (0.46, 0.59, 0.61)

Table 4.2: Continued.

Tree	Non-coding loci			Coding loci		
	BPP	RAxML (reference)	RAxML (non-reference)	BPP	RAxML (reference)	RAxML (non-reference)
			3La			
i	0	0	0.0164 (0.78, 0.78, 0.78)	0	0	0
ii	0	0.0164 (0.33, 0.33, 0.33)	0.0164 (0.84, 0.84, 0.84)	0	0	0
iii	0.3540 (0.36, 0.99, 1.00)	0.0328 (0.50, 0.62, 0.73)	0.0164 (0.79, 0.79, 0.79)	0.2000 (0.62, 0.76, 1.00)	0.1000 (0.66, 0.69, 0.72)	0.0500 (0.78, 0.78, 0.78)
iv	0	0	0	0	0	0
v	0.0159 (0.77, 0.77, 0.77)	0	0	0	0	0
vi	0.0476 (0.70, 0.97, 1.00)	0.2787 (0.35, 0.68, 0.86)	0.3279 (0.49, 0.78, 0.92)	0	0.1000 (0.83, 0.87, 0.90)	0.1500 (0.60, 0.62, 0.91)
vii	0	0.0328 (0.52, 0.53, 0.54)	0.1148 (0.47, 0.71, 0.91)	0	0	0
viii	0.5714 (0.69, 1.00, 1.00)	0.2623 (0.40, 0.59, 0.75)	0.4754 (0.47, 0.75, 0.95)	0.7000 (0.51, 1.00, 1.00)	0.4500 (0.56, 0.66, 0.93)	0.6000 (0.54, 0.85, 0.98)
ix	0	0	0	0	0	0
x	0	0	0	0	0	0
xi	0	0	0	0	0	0
xii	0	0	0	0	0	0
xiii	0.0159 (0.54, 0.54, 0.54)	0.0164 (0.69, 0.69, 0.69)	0	0	0.0500 (0.70, 0.70, 0.70)	0
xiv	0	0	0	0	0	0
			Xag			
i	0	0	0	0	0	0
ii	0	0	0	0	0	0
iii	0	0	0	0	0	0
iv	0	0	0	0	0	0
v	0	0	0	0	0	0
vi	0	0	0	0	0	0
vii	0	0	0	0	0	0
viii	0	0	0	0	0	0
ix	0	0.0526 (0.51, 0.51, 0.51)	0.1053 (0.57, 0.61, 0.65)	0.0833 (0.73, 0.73, 0.73)	0.2500 (0.46, 0.48, 0.65)	0.1667 (0.50, 0.55, 0.59)
x	0.3684 (0.83, 0.92, 1.00)	0.5789 (0.45, 0.69, 0.97)	0.5263 (0.38, 0.62, 0.95)	0.5833 (0.57, 0.97, 1.00)	0.0833 (0.97, 0.97, 0.97)	0.0833 (0.93, 0.93, 0.93)
xi	0.6316 (0.43, 0.99, 1.00)	0.3684 (0.53, 0.91, 0.94)	0.3684 (0.65, 0.78, 0.94)	0.1667 (0.35, 0.52, 0.69)	0.6667 (0.56, 0.74, 1.00)	0.7500 (0.58, 0.71, 0.99)
xii	0	0	0	0	0	0
xiii	0	0	0	0	0	0
xiv	0	0	0	0	0	0

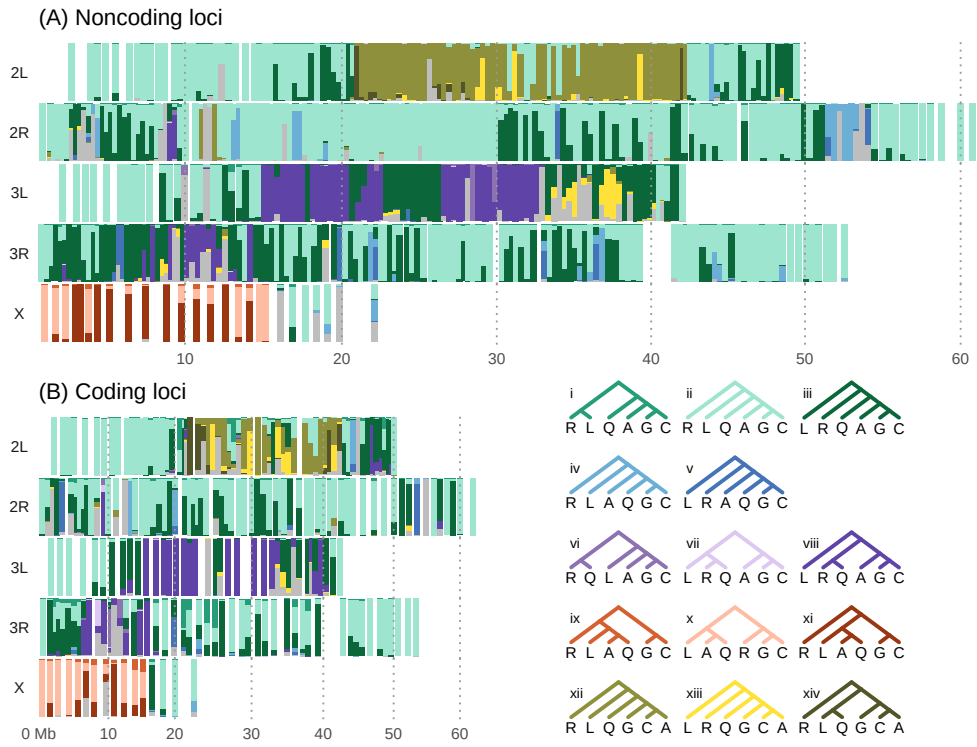


Figure 4.3: Posterior probabilities of species trees inferred using BPP when the outgroup species *A. christyi* was included. The outgroup is always the earliest branching species in the MAP trees and is omitted in the tree diagrams. See legend to Figure 4.2.

The two major autosomal inversion regions, 2La and 3La, were dominated by slightly different trees from the rest of the autosomes. In the 2La region, tree xii: (R(L(Q(G(CA))))) was the MAP tree in almost all blocks, and was almost exclusive to this part of the genome. The phylogeny of the 2La region is discussed in Section 4.3.6. The 3La region was dominated by tree viii: (L((RQ)(A(GC)))), with the (A(GC)) clade as in most parts of the autosomes, and the (RQ) clade, which suggests introgression between *A. merus* and *A. quadriannulatus*.

4.3.2 Concatenation produces different phylogenies from coalescent-based methods

While the sliding-window analysis of Fontaine et al. (2015) inferred tree i for the autosomes and tree ix for the Xag, our BPP analysis inferred trees ii and xi, respectively. Both the data and the analysis methods differ between the two studies. Instead of a 50-kb contiguous block in each sliding window of Fontaine et al. (2015), we used 100 widely spaced loci in each block. Also we separated the noncoding and coding loci and realigned the sequences at each locus. To identify the factors that account for the different inferred trees, we used RAxML to infer one ML tree for each block, with all 100 loci in the block concatenated into a single alignment (of about 20 kb). *A. christyi* was used as the outgroup to root the tree.

For the autosomal noncoding data, the most common ML tree was tree i (with frequency 46%, vs. 23% for tree ii) (Table 4.2 and Figure 4.4A). This was consistent with Fontaine et al. (2015) and different from the

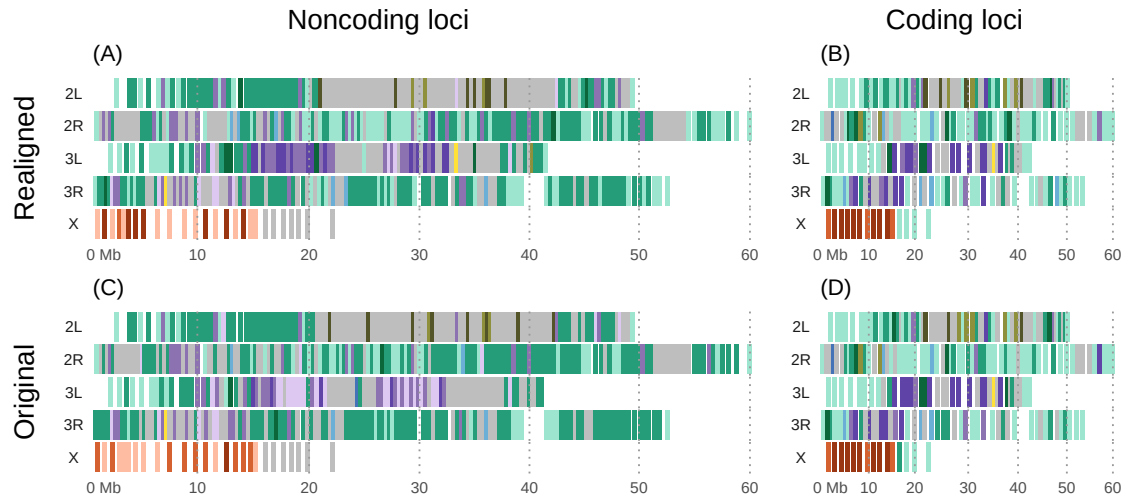


Figure 4.4: ML concatenation trees inferred using RAxML from blocks of 100 loci. The reference genomes were used for each ingroup species, and the results for the non-reference genomes were virtually identical (not shown). Colours represent different trees defined in Figure 4.2.

BPP analysis. For the autosomal coding data, the most common ML tree was tree ii (with frequency 48%) although it had a tendency to infer tree i as well (frequency 19%) (Table 4.2). This was more consistent with the BPP analysis. Whether the sequences were realigned or not did not impact the results (Figure 4.4).

For the noncoding data from the Xag region, the most common ML tree for the realigned data was tree x (with frequency 53%, vs. 37% for tree xi) (Table 4.2, Figure 4.4A). For the coding Xag data, the most common ML tree for the realigned data was tree xi (with frequency 71%, vs. 21% for tree ix). When the original genome alignments were used, the most common ML trees were trees ix and x for the noncoding data, and tree xi for the coding data (Figure 4.4C-D). The sliding-window analysis of the noncoding data in the Xag region in Fontaine et al. (2015), which should be most similar to our concatenation analysis of the original alignments, indeed favoured tree ix. The results suggest that the most important factor accounting for the different trees between the two studies is the method used: the BPP coalescent method on one hand and the sliding-window/concatenation on the other. Concatenation fitted a single tree to all sites in the alignment, ignoring the genealogical heterogeneity across the genome. Furthermore, alignment errors appeared to affect the ML analysis of the Xag data in Fontaine et al. (2015).

4.3.3 Simulation suggests systematic errors in concatenation analysis

To understand the differences between the MSC approach using BPP and concatenation using RAxML, we analysed two sets of data simulated under the MSC model (Rannala and Yang, 2003) and GTR+Γ_4 (Yang, 1994a,b). The first set was generated using tree ii, the autosomal BPP tree, with parameters estimated from the noncoding loci on 2L (2L1+2L2) under the MSC (A00 analysis in Yang (2015)) (Figure 4.9A). The second set was generated using tree xi, the Xag BPP tree, with parameters estimated from the Xag noncoding loci

Table 4.3: Proportions of inferred trees for simulated datasets analysed in blocks of 100 loci (with minimum, median and maximum support values for the inferred tree in parentheses), averaged over 10 replicates.

Tree	BPP	RAxML (Subset 1)	RAxML (Subset 2)
	2L data (6,464 loci, 10 replicates)		
i	0.0062 (0.43, 0.65, 1.00)	0.4308 (0.34, 0.76, 1.00)	0.4492 (0.33, 0.76, 1.00)
ii*	0.9877 (0.47, 0.99, 1.00)	0.5139 (0.29, 0.77, 1.00)	0.5062 (0.31, 0.78, 1.00)
iii	0.0062 (0.48, 0.53, 0.81)	0.0385 (0.51, 0.56, 0.59)	0.0354 (0.36, 0.59, 0.94)
	Xag data (1,825 loci, 10 replicates)		
ix	0.1000 (0.42, 0.52, 0.78)	0.1105 (0.36, 0.61, 0.99)	0.1316 (0.38, 0.57, 0.96)
x	0.0474 (0.41, 0.67, 1.00)	0.4790 (0.46, 0.83, 1.00)	0.4632 (0.36, 0.83, 1.00)
xi*	0.8526 (0.38, 0.84, 1.00)	0.4105 (0.33, 0.75, 1.00)	0.4053 (0.38, 0.74, 1.00)

Note.—For BPP the inferred tree is the MAP tree and the support value is the posterior probability, while for RAxML the inferred tree is the ML tree and the support value is the minimum bootstrap support value for clades. RAxML also inferred other trees in a small fraction (about 1%) of 2L datasets. Trees are given in Figure 4.2. The correct tree (indicated by *) is tree ii for 2L data and tree xi for Xag data.

(Figure 4.9B). For each set, the same number of loci were simulated as in the real data, and were analysed in blocks of 100 loci.

The MAP species tree from BPP matched the correct tree ii in about 99% of the replicate data blocks for the 2L data, and about 85% for the Xag data (Table 4.3). The posterior probability for the MAP tree was high when the tree was correct (median 0.99 for 2La and 0.84 for Xag) and was low when the MAP tree was wrong (median 0.65 for 2L and 0.67 for Xag). Even though BPP assumed the simple JC model (Jukes and Cantor, 1969) while the data were simulated under the far more complex GTR+Γ_4 model, BPP performed well. Also 100 loci from the 2L region appeared to be enough for BPP to infer the species tree with high confidence and high accuracy, but not for the Xag region, apparently because the Xag tree has an extremely short branch (Figure 4.9B).

Concatenation/ML performed far more poorly than BPP. For the 2L data, the ML tree was the true tree ii about 51% of the time and the incorrect tree i about 44% of the time (Table 4.3). Note that tree i has a more balanced shape, and concatenation is known to favour the incorrect, more balanced, tree when the true species tree is unbalanced with very short internal branches Yang (2014, pp.333-335). Tree i was the most common ML tree for the autosomes in the sliding-window analysis of Fontaine et al. (2015) and in our concatenation analysis of the noncoding data (Figure 4.4A&C and Table 4.2). For the Xag data, the ML tree was the correct tree xi about 41% of the time, and the incorrect tree x about 47% of the time (Table 4.3). Again tree x was the most common ML tree, as in the analysis of real Xag noncoding data (Figure 4.4A&C and Table 4.2). The bootstrap support values for the ML trees were mostly moderate, and did not appear to depend on whether the ML tree was correct or not (Table 4.3). This was probably due to the presence of conflicting phylogenetic signals from different loci while the method attempted to fit one tree to all loci. The simulation results closely mimicked the analysis of the real data, providing strong evidence that sliding-window/concatenation is unreliable for inferring the species tree in the *A. gambiae* species complex, and that tree i for the autosomes and tree ix for the Xag inferred in Fontaine et al. (2015) were

methodological artefacts.

4.3.4 The X chromosome represents the true species phylogeny, with *A. merus* diverging first

Fontaine et al. (2015) observed that the X chromosome (or more precisely, the Xag region) and the autosomes support drastically different species trees and argued that the majority tree for Xag represents the true species branching order while the autosome trees were a consequence of introgression between species. Our analysis supports this assertion, consistent with the long-standing view that differentially fixed inversions on the X chromosome act as a reproductive barrier between species while the autosomes may be more easily mixed among the three species *A. arabiensis*, *A. gambiae* and *A. coluzzii* (Besansky et al., 2003; Wang-Sattler et al., 2007; Neafsey et al., 2010; O'Loughlin et al., 2014; Crawford et al., 2015). Nevertheless, our BPP analysis inferred different trees for both the autosomes and the Xag from those of Fontaine et al. (2015). Here, we first summarise evidence in favour of the Xag trees as opposed to the autosomal trees. We then discuss evidence supporting tree xi (the BPP Xag tree) in particular as the true species tree.

There are two major pieces of evidence that support of the Xag trees as the true species tree, rather than the autosomal trees (tree ii from BPP or tree i from Fontaine et al. (2015)). First, the Xag trees are compatible with evidence on cross-species introgression. Note that the major difference between the Xag and autosomal trees concerns the relationships of *A. arabiensis* and *A. gambiae*+*A. coluzzii*. Introgression of *A. arabiensis* into the common ancestor of *A. gambiae*+*A. coluzzii* in any of trees ix, x and xi (the three alternative trees for the Xag region) yields tree ii, the most common BPP tree for the autosomes. Introgression between *A. arabiensis* and *A. gambiae*+*A. coluzzii* has long been suggested (Besansky et al., 1994; García et al., 1996). This introgression is analysed in Section 4.3.5 through its impact on divergence times and through direct estimation of migration rates. By contrast, while tree x for the Xag could be explained by introgression of *A. merus* into the common ancestor of *A. gambiae*+*A. coluzzii* in tree ii if tree ii were the true species tree, evidence for such introgression has never been reported in the literature.

Second, chromosomal inversions support the Xag trees and contradict the autosomal trees i and ii (Kamali et al., 2012; Fontaine et al., 2015). Ten fixed inversions have been identified in the *A. gambiae* complex, of which five are on the X chromosome. Comparison with outgroup species revealed that the chromosomal orientations of *A. gambiae* and *A. merus* closely resemble the ancestral karyotype (Kamali et al., 2012; Fontaine et al., 2015), suggesting early divergences of *A. merus* and *A. gambiae*+*A. coluzzii*, consistent with the Xag trees. By contrast, *A. arabiensis* and *A. gambiae*+*A. coluzzii* differ by at least five overlapping inversions on the X chromosome, with an intermediate orientation (X+) found in *A. melas* and *A. quadriannulatus* (Coluzzi et al., 1979, 2002; Fontaine et al., 2015). Moreover, *A. gambiae*+*A. coluzzii* and *A. merus* share the ancestral Xag orientation. As a result, explaining the data using tree ii would require a reversal from the derived orientation X+ to the ancestral Xag in the lineage leading to *A. gambiae*+*A. coluzzii*. It

is thus highly unlikely that *A. arabiensis* and *A. gambiae+A. coluzzii* form a clade as suggested by the autosomal trees.

Consideration of the statistical properties of the methods suggests that tree ix for the Xag and tree i for the autosomes inferred in Fontaine et al. (2015) are artifactual. Our simulation has highlighted the systematic bias of the concatenated/ML method, which behaves similarly to the sliding-window approach of Fontaine et al. (2015): when the true tree is ii or xi, concatenation/ML tends to infer trees i and ix (or x), respectively (Table 4.3). Consistent with this, we note that the neighbour-joining method applied to the average sequence divergences for the Xag region inferred tree xi, even though ML inferred tree ix (Fontaine et al., 2015, Fig. 1D). While ML applied to concatenated data may be inconsistent, the average coalescent times or sequence divergences track species divergences, so that neighbour-joining (or UPGMA in the case where the molecular clock holds) is a coalescent-aware and statistically consistent method (Liu and Edwards, 2009). Here we summarise further evidence against tree ix for the Xag and tree i for the autosomes.

First, tree i for the autosomes cannot be explained by introgression from *A. arabiensis* to *A. gambiae+A. coluzzii* alone, whereas such introgression in any of the three alternative trees for the Xag (ix, x and xi) leads to tree ii, the BPP tree for the autosomes. Second, while chromosomal inversions favour the Xag trees over the autosomal trees, as discussed above, they support tree xi (the BPP Xag tree) far more strongly than tree ix or tree x (Kamali et al., 2012, Fig. 8C). Indeed the most parsimonious tree for the fixed inversion data (Fontaine et al., 2015, Fig. S27A) has the relationship (((QA)G)R), which is consistent with tree xi and requires no independent fixations of the same inversions in different lineages. By contrast, the inversion phylogeny that is consistent with tree ix, (((QA)R)G), is not parsimonious and requires independent fixations in two lineages and introgression of 2La from *A. gambiae* to *A. arabiensis*, as well as ancient polymorphisms of the 2La and 2Ro inversions that likely predate speciation in the species complex (Fontaine et al., 2015, Fig. S27B). Note that *A. merus* is the only species in the complex that has an ancestral 2Ro inversion and a derived 2Rp inversion on chromosome 2R, while the other species in the complex are fixed for the derived $2R+^{o}$ and ancestral $2R+^{p}$ orientations (Kamali et al., 2012). As a result, tree xi requires only one fixation event for each of those two inversions, whereas the other two trees (ix and x) require two independent fixations of $2R+^{o}$ in two lineages, one leading to (GC) and another leading to (L(AQ)). The phylogeny for the 2La region is discussed in Section 4.3.6. Our suggested species tree, tree xi, provides a much simpler interpretation of the chromosomal inversion data, compared with tree ix as suggested by Fontaine et al. (2015).

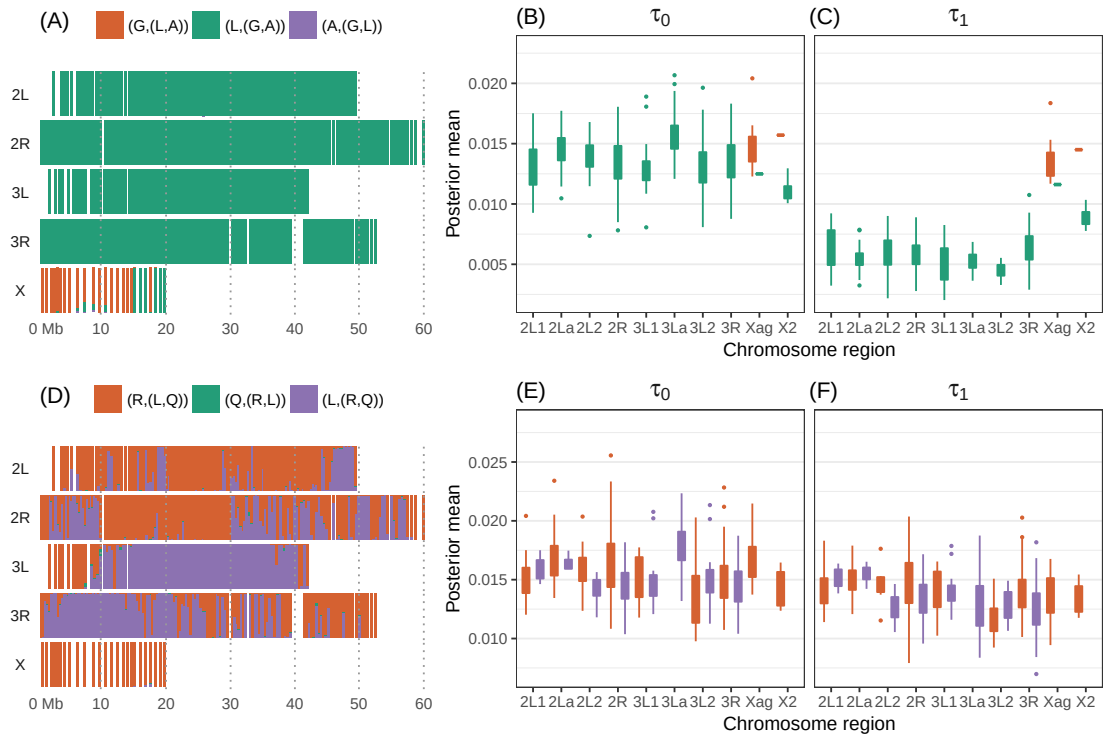


Figure 4.5: BPP analysis of GAL and RQL triplets. Left panel: posterior probabilities of species trees. Middle and right panels: posterior means of the two divergence times in the MAP species tree across different regions of the genome.

4.3.5 Divergence times and migration rates suggest A-to-G introgression in autosomes and R-to-Q introgression in chromosome 3L

Our analysis of introgression has two components. First, following Fontaine et al. (2015), we used BPP to estimate the species divergence times as introgression has the effect of reducing divergence times between species (Figure 4.5). Second we used the program 3S to explicitly estimate the migration rates between pairs of species under the MSC model with migration (Table 4.4).

Since the autosomes and X chromosome support different trees for *A. gambiae*, *A. arabiensis* and *A. melas* (Figure 4.2), we analysed the GAL triplet data using BPP. The most common MAP tree was (L(GA)) for

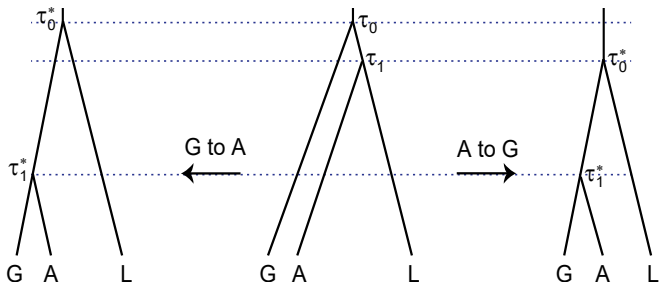


Figure 4.6: Introgression changes species relationships and reduces divergence times (Fontaine et al., 2015, Fig. S16). For the GAL triplet, A-to-G introgression leads to the tree ((GA)L), with divergence times $\tau_0^* = \tau_1 < \tau_0$ and $\tau_1^* < \tau_1$, while G-to-A introgression leads to the tree ((GA)L), with $\tau_0^* = \tau_0$ and $\tau_1^* < \tau_1$.

the autosomes and (G(LA)) for Xag (Figure 4.5A), consistent with the BPP analysis of the full data of six species (Figure 4.2 and Table 4.2). Fontaine et al. (2015) fitted ML trees to 10-kb non-overlapping windows across the autosomes, and found that the divergence times in the tree (G(LA)) were greater than those in (L(GA)), suggesting that the reduced divergence is a consequence of autosomal introgression, and that the Xag tree represents the true species relationship, but the direction of the introgression was inconclusive (Fontaine et al., 2015, Fig. 3). The same pattern was observed here. In the Xag tree (G(LA)), the two node ages were nearly identical, with $\tau_0 \approx \tau_1$, while in the autosomal tree (L(GA)), the root age τ_0 was close to the root age in the Xag tree, but τ_1 was much smaller (Figure 4.5B-C). This was so even if we took into consideration the mutation rate variation among genomic regions (Table 4.5). This provides strong evidence that the Xag region is not affected by introgression and represents the true species relationship, while there is gene flow between *A. arabiensis* and *A. gambiae*+*A. coluzzii* for the autosomes (Figure 4.6).

The A→GC introgression should lead to a reduction of both τ_0 and τ_1 while the GC→A introgression should reduce τ_1 only (Figure 4.6). However, there may be little power to use this prediction to infer the direction of introgression because (1) the gene trees are a mixture generated from the original species tree as well as the introgressed species tree, (2) the original species tree is star-like (Figure 4.9) with $\tau_0 \approx \tau_1$ (Figure 4.6) so that the two hypotheses make nearly identical predictions, and (3) the mutation rate varies among genomic regions (Table 4.5), complicating the comparison of τ estimates. Similarly the sliding-window analysis and the D statistic used in Fontaine et al. (2015) are uninformative about the direction of migration.

To explicitly estimate the migration rates between *A. arabiensis* and *A. gambiae*, we used the 3s program to analyse GAO, GAR and GAL triplets, using *A. christyi* (O), *A. merus* (R) and *A. melas* (L) as the outgroup, respectively. For the GAO triplet, the estimates suggest G→A introgression, but the evidence is not significant except for 2L and 3L (Table 4.4). No gene flow was detected in the opposite direction, nor on the X chromosome. However, since *A. christyi* is a very distant outgroup (Figure 4.9), our data in effect consisted of species pairs and may not be informative (Dalquen et al., 2017). We thus further analysed the GAL and GAR triplets (Table 4.4). While *A. melas* is not a correct outgroup, both the correct species tree (G(LA)) and the incorrect tree (L(GA)) are close to the star tree (Figure 4.9), so that estimates from the wrong tree (L(GA)) may still be informative. Indeed the results for the GAL and GAR triplets were highly similar, suggesting strong evidence of introgression from *A. arabiensis* to *A. gambiae* affecting the autosomes but not the X chromosome. We found no evidence for gene flow from *A. gambiae* to *A. arabiensis*. The migration rate estimates varied considerably among chromosomal regions, which may reflect different strengths of natural selection removing immigrants, besides random sampling errors. The estimates from the GAR triplet may be the most reliable. The average rate for the autosomes from *A. arabiensis* to *A. gambiae* was $Nm = 0.22$ immigrants per generation (Table 4.4). For any plausible value of N, the migration proportion m must be orders of magnitude smaller than the recorded frequencies of hybridisation

Table 4.4: MLEs ($\times 10^{-2}$) from 3s analysis of triplet data under models M0 (no gene flow) and M2 (with gene flow).

Chr	Model	τ_1	τ_0	θ_4	θ_5	θ_1	θ_2	M_{12}	M_{21}	$2\Delta\ell$
GAO, species tree ((GA)O)										
2L12	M0	0.47	7.28	11.38	0.93	2.93	0.99			
	M2	0.50	7.27	11.38	0.92	2.63	0.98	0.00	11.13	3.70
2La	M0	0.50	8.28	10.53	1.17	49.41	1.04			
	M2	0.50	8.28	10.53	1.17	52.04	1.03	0.00	20.69	2.21
2L	M0	0.47	7.69	11.12	1.07	6.24	1.01			
	M2	0.56	7.69	11.12	1.04	4.16	0.98	0.00	60.12	43.10
2R	M0	0.42	7.55	11.29	0.86	3.67	1.50			
	M2	0.42	7.55	11.29	0.86	3.67	1.50	0.00	0.00	0.00
3L12	M0	0.37	7.82	10.44	0.90	1.79	1.14			
	M2	0.40	7.82	10.44	0.88	1.58	1.13	0.00	12.14	3.79
3La	M0	0.47	8.12	9.51	0.82	10.18	1.80			
	M2	0.47	8.12	9.51	0.82	9.49	1.79	0.00	12.63	0.21
3L	M0	0.42	7.97	9.95	0.87	3.40	1.44			
	M2	0.46	7.97	9.96	0.84	2.73	1.39	0.00	29.39	13.56
3R	M0	0.47	7.39	10.81	0.98	3.34	1.67			
	M2	0.47	7.39	10.81	0.98	3.34	1.67	0.00	0.00	0.00
auto	M0	0.43	7.51	11.00	0.92	3.17	1.42			
	M2	0.44	7.51	11.00	0.92	3.12	1.42	0.00	2.08	1.10
Xag	M0	1.08	7.44	13.62	1.74	0.71	0.34			
	M2	1.12	7.44	13.63	1.70	0.71	0.33	0.17	0.00	1.39
X2	M0	0.75	9.32	10.97	0.95	0.35	0.38			
	M2	0.75	9.32	10.97	0.95	0.35	0.38	0.00	0.00	0.00
GAR, species tree ((GA)R)										
2L12	M0	0.64	1.21	1.49	0.62	2.97	1.11			
	M2	0.74	1.20	1.51	0.51	2.36	1.07	0.00	17.07	30.13
2La	M0	0.64	1.37	1.50	0.88	32.18	1.11			
	M2	1.28	1.34	1.56	0.10	12.76	1.01	0.00	365.67	197.87
2L	M0	0.62	1.28	1.51	0.77	6.30	1.11			
	M2	0.85	1.26	1.54	0.52	3.76	1.05	0.00	60.43	146.54
2R	M0	0.59	1.22	1.53	0.55	3.92	1.58			
	M2	0.61	1.21	1.54	0.53	3.55	1.55	0.00	11.68	9.20
3L12	M0	0.46	1.12	1.55	0.73	2.12	1.06			
	M2	0.61	1.11	1.58	0.56	1.49	1.03	0.00	25.85	26.47
3La	M0	0.60	1.25	1.40	0.60	13.39	1.91			
	M2	0.67	1.25	1.41	0.54	8.55	1.80	0.00	90.19	9.95
3L	M0	0.54	1.19	1.46	0.66	4.23	1.51			
	M2	0.72	1.18	1.49	0.47	2.57	1.39	0.00	53.68	78.85
3R	M0	0.70	1.12	1.55	0.50	3.78	1.81			
	M2	0.81	1.11	1.57	0.36	2.89	1.72	0.00	24.74	59.11
auto	M0	0.61	1.17	1.54	0.58	3.43	1.49			
	M2	0.69	1.16	1.56	0.49	2.70	1.43	0.00	21.93	124.48
Xag	M0	1.37	1.37	1.92	0.01	0.77	0.37			
	M2	1.37	1.37	1.91	0.01	0.77	0.37	0.00	0.05	0.90
X2	M0	0.81	1.19	1.50	0.58	0.37	0.36			
	M2	0.81	1.19	1.50	0.58	0.37	0.36	0.00	0.00	0.00

Note.—Chr, chromosomal regions: 2L12 = 2L1 + 2L2 = 2L without 2La, 3L12 = 3L1 + 3L2 = 3L without 3La, and auto = 2L12 + 2R + 3L12 + 3R (autosomes without 2La and 3La). The likelihood ratio test statistic ($2\Delta\ell$) for testing models M0 (no gene flow) against M2 (gene flow) is compared with the critical values 4.61 at 10% level, 5.99 at 5% level, and 9.21 at 1% level.

Table 4.4: Continued.

Chr	Model	τ_1	τ_0	θ_4	θ_5	θ_1	θ_2	M_{12}	M_{21}	$2\Delta\ell$
GAL, species tree ((GA)L)										
2L12	M0	0.59	1.16	1.40	0.66	3.49	1.17			
	M2	0.74	1.15	1.43	0.48	2.48	1.11	0.00	25.80	40.54
2La	M0	0.65	1.31	1.38	0.91	44.09	1.08			
	M2	1.29	1.29	1.42	0.00	16.42	1.00	0.00	467.54	175.21
2L	M0	0.60	1.22	1.41	0.81	7.70	1.13			
	M2	0.89	1.21	1.45	0.45	4.03	1.06	0.00	80.37	154.76
2R	M0	0.60	1.15	1.38	0.54	3.94	1.58			
	M2	0.64	1.15	1.38	0.49	3.35	1.53	0.00	18.46	19.36
3L12	M0	0.50	1.13	1.52	0.65	2.10	1.25			
	M2	0.77	1.10	1.58	0.33	1.29	1.11	2.27	30.75	61.61
3La	M0	0.59	1.36	1.63	0.62	11.80	2.12			
	M2	0.65	1.36	1.64	0.57	7.63	1.97	0.00	99.92	14.16
3L	M0	0.57	1.23	1.62	0.61	3.78	1.63			
	M2	0.75	1.21	1.66	0.42	2.26	1.46	0.00	52.54	107.60
3R	M0	0.70	1.14	1.43	0.50	3.85	1.89			
	M2	0.75	1.13	1.45	0.42	3.21	1.82	0.00	18.53	29.59
auto	M0	0.62	1.15	1.41	0.57	3.57	1.57			
	M2	0.70	1.14	1.43	0.47	2.77	1.50	0.00	24.38	138.59
Xag	M0	1.13	1.14	1.80	1.87	0.72	0.37			
	M2	1.13	1.14	1.80	21.74	0.72	0.37	0.00	0.00	0.09
X2	M0	0.96	1.10	1.31	0.27	0.35	0.44			
	M2	1.06	1.09	1.32	0.06	0.34	0.43	0.00	0.25	4.20
RQO, species tree ((RQ)O)										
2L12	M0	1.13	7.38	11.49	1.35	0.61	1.09			
	M2	1.14	7.38	11.49	1.34	0.60	1.09	0.00	0.09	0.56
2La	M0	1.30	8.35	10.60	1.28	0.76	1.22			
	M2	1.30	8.35	10.60	1.28	0.76	1.22	0.00	0.00	0.00
2L	M0	1.20	7.78	11.21	1.32	0.68	1.15			
	M2	1.21	7.78	11.21	1.32	0.68	1.15	0.00	0.06	0.48
2R	M0	1.12	7.63	11.47	1.39	0.60	1.24			
	M2	1.13	7.63	11.48	1.39	0.60	1.23	0.11	0.00	0.88
3L12	M0	1.02	7.88	10.61	1.50	0.63	0.87			
	M2	1.06	7.88	10.61	1.47	0.61	0.87	0.00	0.44	4.20
3La	M0	1.07	8.22	9.64	0.97	0.94	1.83			
	M2	1.08	8.22	9.64	0.96	0.94	1.81	0.45	0.00	1.53
3L	M0	1.04	8.05	10.11	1.21	0.79	1.30			
	M2	1.07	8.05	10.11	1.20	0.79	1.28	0.59	0.00	5.67
3R	M0	1.02	7.46	10.94	1.29	0.70	1.45			
	M2	1.03	7.46	10.94	1.28	0.70	1.44	0.17	0.00	0.64
auto	M0	1.08	7.58	11.15	1.36	0.64	1.24			
	M2	1.09	7.58	11.15	1.36	0.64	1.23	0.18	0.00	3.79
Xag	M0	1.15	7.47	13.78	2.01	0.48	0.53			
	M2	1.38	7.46	13.81	1.77	0.46	0.54	0.00	0.67	14.48
X2	M0	1.07	9.25	11.47	1.51	0.20	0.26			
	M2	1.08	9.25	11.47	1.51	0.20	0.26	0.00	0.00	0.00

Table 4.4: Continued.

Chr	Model	τ_1	τ_0	θ_4	θ_5	θ_1	θ_2	M_{12}	M_{21}	$2\Delta\ell$
RQL, species tree ((RQ)L)										
2L12	M0	1.30	1.30	1.57	0.01	0.62	1.20			
	M2	1.30	1.30	1.57	0.01	0.62	1.20	0.00	0.00	0.00
2La	M0	1.38	1.38	1.49	0.03	0.76	1.40			
	M2	1.38	1.38	1.49	0.03	0.76	1.40	0.00	0.00	0.00
2L	M0	1.34	1.34	1.54	0.02	0.69	1.29			
	M2	1.34	1.34	1.54	0.02	0.69	1.29	0.00	0.06	0.00
2R	M0	1.29	1.29	1.62	0.02	0.62	1.32			
	M2	1.29	1.29	1.62	0.02	0.62	1.32	0.00	0.00	0.00
3L12	M0	1.22	1.22	1.61	0.01	0.63	0.92			
	M2	1.22	1.22	1.61	0.01	0.63	0.92	0.00	0.00	0.00
3La	M0	1.52	1.52	1.64	0.00	1.01	1.87			
	M2	1.51	1.51	1.79	0.00	1.02	1.79	0.67	0.00	44.40
3L	M0	1.37	1.37	1.76	0.00	0.84	1.40			
	M2	1.38	1.38	1.75	0.00	0.84	1.38	0.12	0.00	2.59
3R	M0	1.24	1.25	1.62	0.01	0.73	1.55			
	M2	1.24	1.25	1.62	0.01	0.73	1.55	0.00	0.00	0.00
auto	M0	1.27	1.27	1.61	0.01	0.66	1.32			
	M2	1.27	1.27	1.61	0.01	0.66	1.32	0.00	0.00	0.00
Xag	M0	1.15	1.15	2.01	1.53	0.52	0.64			
	M2	1.15	1.15	2.01	56.82	0.52	0.64	0.00	0.00	0.32
X2	M0	1.18	1.18	1.50	1.01	0.20	0.28			
	M2	1.18	1.18	1.50	17.05	0.20	0.28	0.00	0.00	0.01

(which is $< 0.1\%$ but perhaps not much lower) (Coluzzi et al., 2002). The migration rate Nm represents the expected number of 'successful' migrants, which are those that have contributed DNA in the recipient population after natural selection has removed unfit introgressed alleles. With migration, the estimates of τ_1 in model M2 (gene flow) were greater than those under M0 (no gene flow). Thus ignoring migration underestimates the species divergence time τ_1, as also seen in the BPP analysis above (Figure 4.5). Strong positive correlations between the migration rate M and τ_1 are thus expected. The estimates of τ_0 (the age of the *A. gambiae* complex) were very similar between the two models (M0 and M2) and were also consistent with the estimates from BPP (discussed later in Section 4.3.7), in the range 0.012-0.014.

We also analysed the RQL triplet since there was evidence for R-Q introgression in chromosome 3L (Figure 4.2). As expected, the MAP species tree was predominantly (R(LQ)) throughout the genome and in particular in the Xag region, but was (L(RQ)) in most of 3L and in large parts 3R (Figure 4.5D). On average, (R(LQ)) had older divergence times than the other two trees in most regions on the genome (Figure 4.5E-F).

The likelihood ratio test (LRT) applied to the RQO triplet data detected evidence for gene flow from R to Q in the autosomes, particularly in chromosome 3L, but the evidence was not significant (Table 4.4). The estimates of τ_0 ranged between 0.0738 and 0.0925, comparable to those from the GAO triplet. The

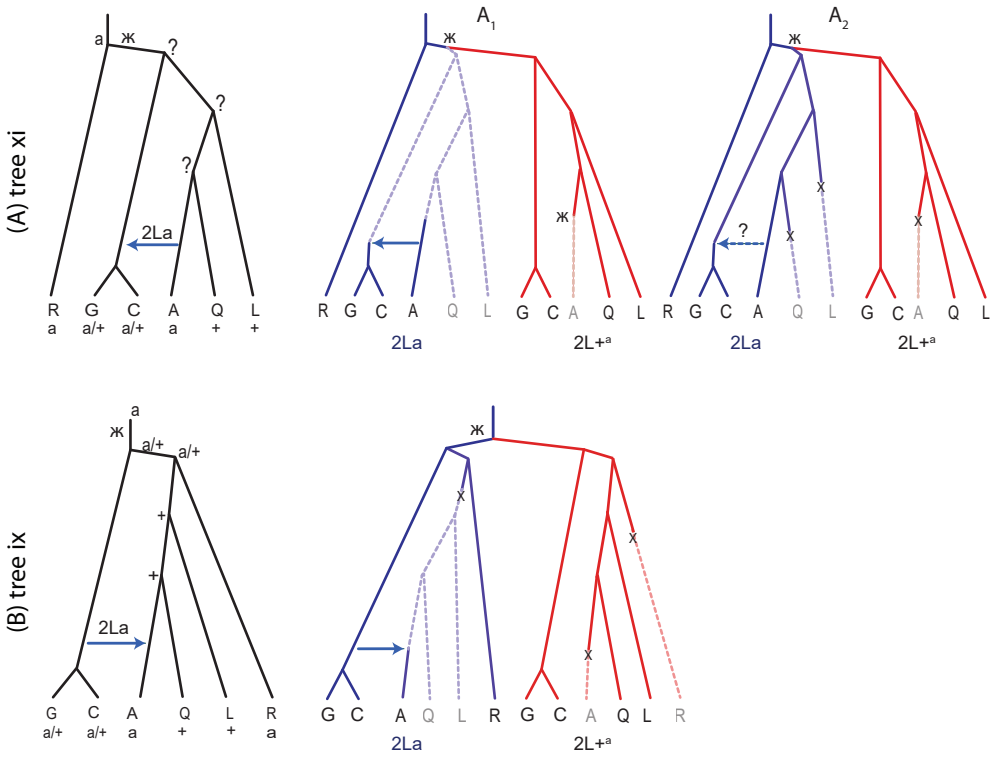


Figure 4.7: Species trees A (*top*) and B (*bottom*) for the 2La region (Fontaine et al., 2015, Fig. S27A-B), based on the assumed species tree xi and ix, respectively. The inversion orientations in the extant and ancestral species are given as 'a': fixed for the 2La orientation, '+': fixed for the 2L+^{a} orientation, and 'a/+': polymorphic for both orientations.

estimates of τ_1 were in the range 0.0102–0.0138, consistent with the BPP estimates on tree xi (0.0156 for Xag and 0.0138 for 2L) (Figure 4.9). Since the LRT suffered from a lack of power due to the use of the distant outgroup, we also analysed the triplet RQL, treating L as the outgroup. Given that the species tree is star-like (Figure 4.9), we expect the estimates to be similar to those if the correct species tree were used. The result suggested gene flow from *A. merus* to *A. quadriannulatus* affecting the 3La region exclusively (Table 4.4). This conclusion was consistent with the previous work based on the *D* statistic using the tree (((L,Q),R),O) (Fontaine et al., 2015), while our analysis additionally provided the direction of introgression.

4.3.6 The evolutionary history of the 2La inversion region

The 2La inversion is a trans-species paracentric chromosomal inversion in *Anopheles* mosquitoes. It is polymorphic in *A. gambiae* and *A. coluzzii*, fixed for the ancestral 2La orientation in *A. arabiensis* and *A. merus*, and fixed for the derived 2L+^{a} orientation in *A. quadriannulatus* and *A. melas* (Coluzzi et al., 2002; Sharakhov et al., 2006). This inversion region has been shown to be associated with malaria vectorial efficiency, adaptation to ecological habitats (in particular, aridity) (Coulibaly et al., 2016; Ayala et al., 2017) and susceptibility to *Plasmodium* infection (Riehle et al., 2017). Sequence divergences in the 2La region are known to be greater between the karyotypes in *A. gambiae* and *A. coluzzii* (2La/2La, 2La/2L+^{a}, 2L+^{a}/2L+^{a}) than between species (Neafsey et al., 2010; O'Loughlin et al., 2014; Weetman et al., 2014; Fontaine et al.,

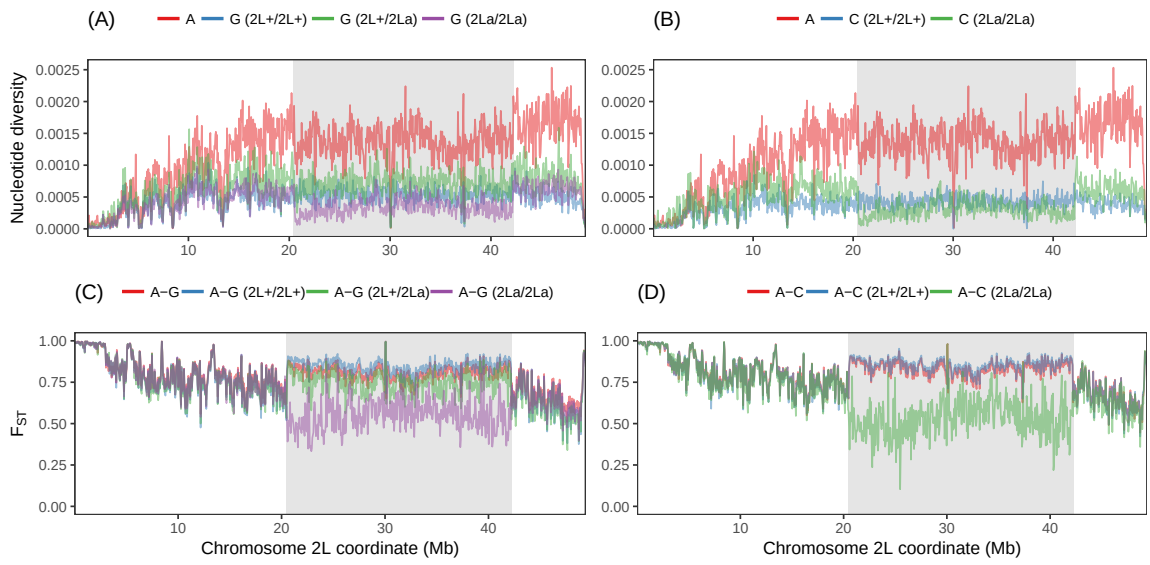


Figure 4.8: (A and B) Nucleotide diversity and (C and D) pairwise F_{ST} statistic between *A. arabiensis* (A) and different 2La karyotypes of *A. gambiae* (G) and *A. coluzzii* (C) calculated from the genome-wide SNP data of natural populations from Fontaine et al. (2015). The 2La region is shaded. Sample sizes are $n = 23$ for *A. gambiae* (35% 2L+^{a}/2L+^{a}, 22% 2L+^{a}/2La, 43% 2La/2La), $n = 11$ for *A. coluzzii* (73% 2L+^{a}/2L+^{a}, 27% 2La/2La, no 2L+^{a}/2La) and $n = 12$ for *A. arabiensis*.

2015; Riehle et al., 2017).

The species tree ix implies an evolutionary history for the 2La region, referred to as tree B here (Figure 4.7B). This was suggested by Fontaine et al. (2015, Figs. 5A & S27B) and posits the existence of ancestral polymorphism of the two orientations prior to the radiation of the species complex, two independent losses of the 2L+^{a} orientation (in *A. merus* and *A. gambiae*), one loss of 2La, as well as GC→A introgression of the 2La orientation and complete replacement of the 2L+^{a} orientation in *A. arabiensis*. Our likelihood ratio test detected no such introgression, but the same data provided overwhelming evidence for introgression in the opposite direction (Table 4.4). Similarly, a model of GC→A introgression was found to be incompatible with the data in a simulation-based analysis of site-frequency spectrum data (He and Knowles, 2016). Moreover, crossing experiments found evidence of introgression of 2La region from *A. arabiensis* into *A. gambiae* but not in the opposite direction (della Torre et al., 1997; Slotman et al., 2005).

Our proposed species tree (tree xi) suggests an alternative history for the 2La region, referred to as tree A (Figure 4.7A), which is more parsimonious (Fontaine et al., 2015, Fig. S27A). This posits the origin of the derived 2L+^{a} form after *A. merus* branched off and A→GC introgression. This introgression may predict less polymorphic 2La orientation than 2L+^{a} in *A. gambiae* and *A. coluzzii*. Indeed, a reduced nucleotide diversity in the 2La region in the 2La/2La karyotype of *A. gambiae* and *A. coluzzii* was observed, but not in 2La/2L+^{a} and 2L+^{a}/2L+^{a} (Figure 4.8A-B). Such differences in nucleotide diversity among the different karyotypes may not be predicted by tree B since all three karyotypes in *A. gambiae* and *A. coluzzii* are old under tree B. Also there was no clear reduction in the nucleotide diversity in the 2La region in *A. arabiensis*, as may be predicted by tree B. Genetic differentiation measured by F_{ST} was reduced between *A. arabiensis*

and the 2La/2La karyotype of *A. gambiae* and *A. coluzzii*, but not for the other pairs, relative to the rest of chromosome 2L (Figure 4.8C-D), although this pattern is predicted by both trees as a consequence of 2La introgression. Moreover, the frequency of the 2La orientation in *A. gambiae* is higher in geographical ranges where *A. gambiae* overlaps with *A. arabiensis* (He and Knowles, 2016), consistent with tree A, but not expected under tree B.

We consider two variations of tree A: A1 and A2. Tree A1 requires one inversion of 2La into $2L+^{a}$ after *A. merus* branched off and one reversal of $2L+^{a}$ to 2La in the lineage leading to *A. arabiensis*, with the polymorphism of 2La region in *A. gambiae*+*A. coluzzii* explained by the A→GC introgression. Since the reversal to 2La occurred in a homogeneous background in *A. arabiensis*, the breakpoints of 2La in *A. arabiensis* are expected to differ from those in *A. gambiae*+*A. coluzzii* and *A. merus*. However, the organisation of genes and noncoding elements around the 2La breakpoints in those species appear to be identical (Sharakhov et al., 2006), suggesting that the multiple origins scenario of the 2La orientation in this species complex is highly unlikely.

Tree A2 assumes an extensive period of ancestral polymorphism of both orientations, and independent losses of the 2La orientation in *A. melas* and *A. quadriannulatus*, and a loss of the $2L+^{a}$ orientation in *A. arabiensis*. This allows A→GC introgression of the 2La orientation but does not require it. The introgression in a polymorphic background should result in two distinct haplotypes of the 2La orientation in *A. gambiae* and *A. coluzzii* (the original and introgressed). However, no such heterogeneity has been found in the 2La heterozygotes in analysis of genome-wide SNP data from hundreds of field-caught mosquitoes of *A. gambiae* and *A. coluzzii*, apart from clustering by geographical origins (Riehle et al., 2017; The *Anopheles gambiae* 1000 Genomes Consortium, 2017), which supports the scenario of no A→GC introgression of 2La. A variant of tree A2 is to allow an additional loss of the 2La orientation in the *A. gambiae*+*A. coluzzii* ancestor followed by introgression of 2La from *A. arabiensis*. Note that all losses occur in the shared polymorphic background, thus preserving the breakpoint structure. This is consistent with the single origin scenario of the 2La orientation.

These hypotheses make different predictions about sequence divergences between species and between the different karyotypes of *A. gambiae* and *A. coluzzii*, which may be useful to distinguish between them. However, the whole-genome data of Fontaine et al. (2015) are in the form of haploid consensus sequences generated from the diploid samples and may not have such resolution.

4.3.7 Estimation of species divergence parameters

The BPP estimates of τ were proportional between the coding and noncoding data, with the regression coefficient of 0.524 (with $r^2 = 0.998$) for tree ii for the 2L region, and 0.323 ($r^2 = 0.994$) for tree xi for the Xag (Figure 4.10A-B). The role of purifying selection in removing nonsynonymous mutations is

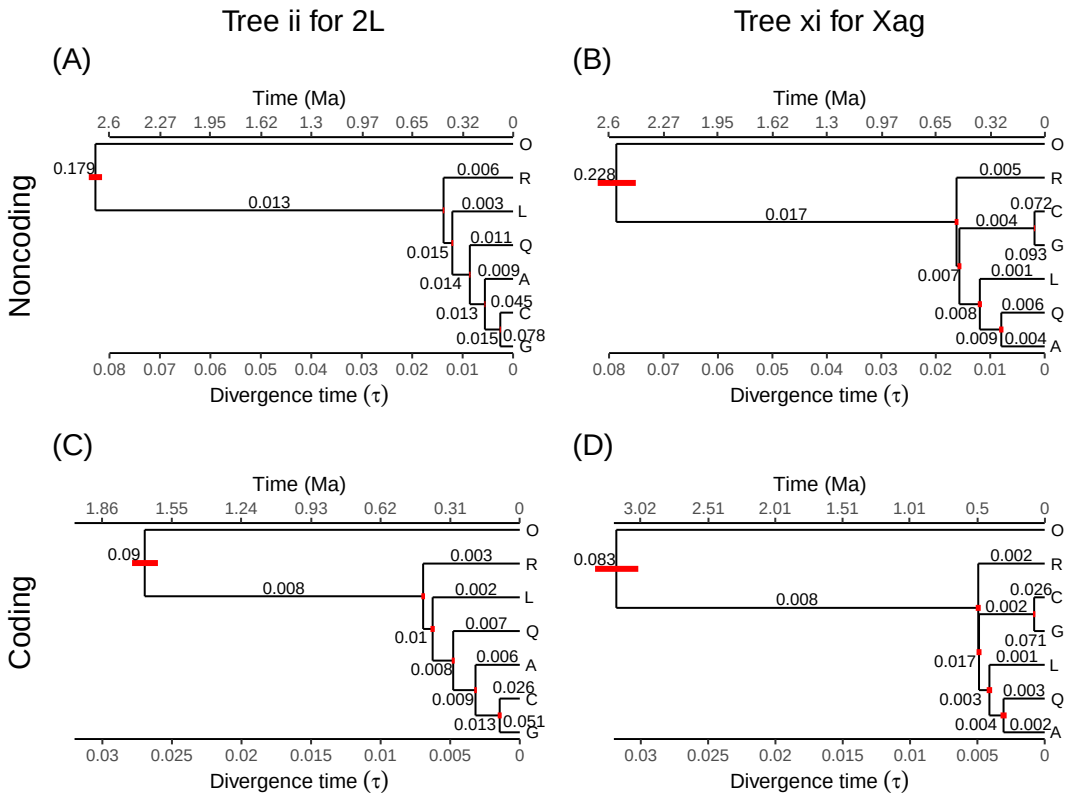


Figure 4.9: Trees ii and xi with the posterior estimates of population sizes (θs, numbers on the branches) and species divergence times (τs, the bottom horizontal axis; bars represent 95% HPD intervals) from BPP. Parameters for tree ii were estimated from all loci in chromosome 2L excluding 2La region, while those for tree xi were estimated from all loci in the Xag region of the X chromosome. Divergence times were calculated assuming the mutation rate 2.8×10^{-9} per site per generation for autosomal noncoding loci (A), with 11 generations per year, and 0.524 and 0.323 times (Figure 4.10) as large for the coding autosomes (C) and coding Xag loci (D), respectively. Ma, million years ago.

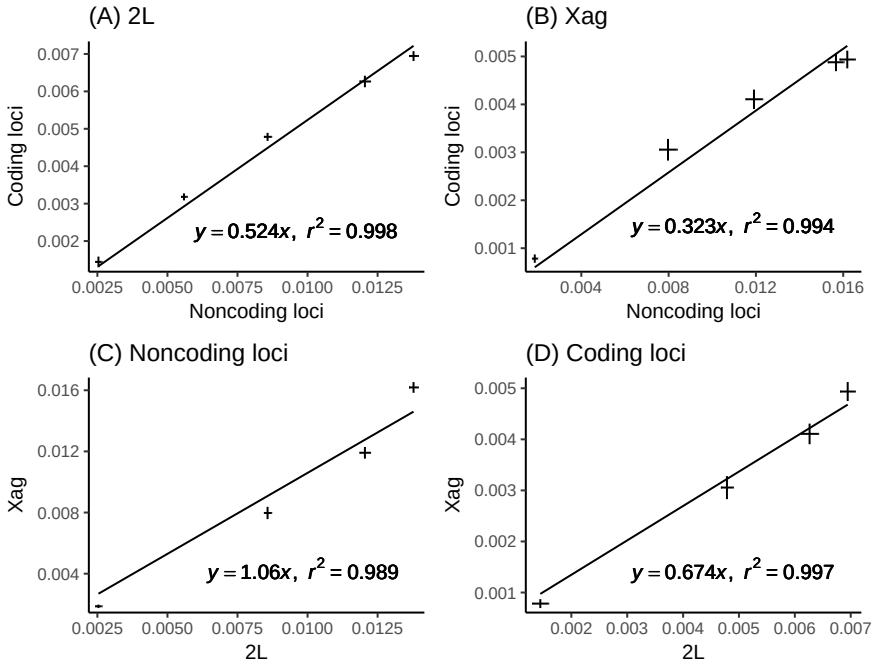


Figure 4.10: Posterior means of species divergence times (τ) from different datasets (see Figure 4.9). Error bars represent the 95% HPD intervals.

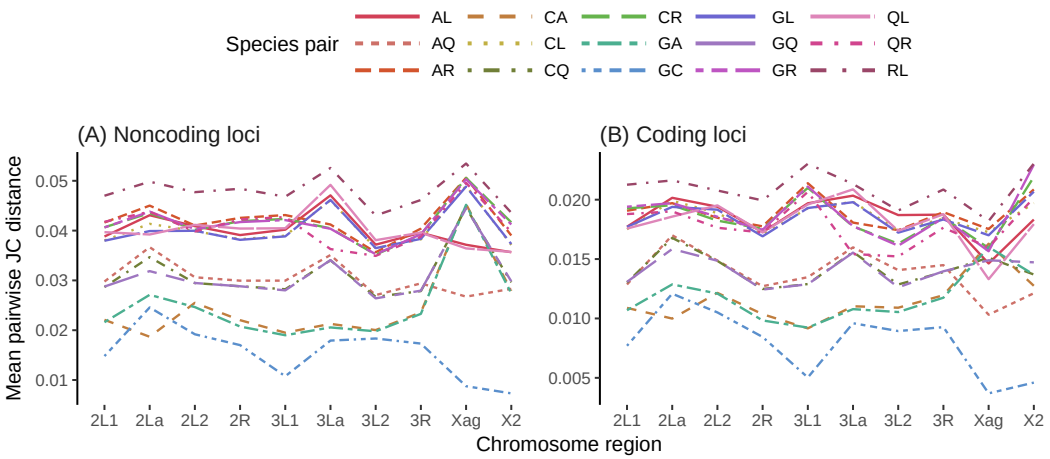


Figure 4.11: Pairwise JC distance from the whole-genome data, averaged over loci in each chromosomal region.

Table 4.5: Relative mutation rates for noncoding loci in different chromosomal regions

chr	τ_1 (LRO)	θ_5 (LRO)	d_{JC} (RL)
2L12	1.055	0.977	1.018
2La	1.113	0.983	1.071
2R	1.025	1.019	1.040
3L12	0.966	0.987	0.966
3La	1.116	1.077	1.130
3R	0.964	0.978	0.992
auto	1	1	1
Xag	1.015	1.260	1.149
X2	0.873	1.161	0.937

Note.—The relative rates were calculated using the MLEs of τ_1 or θ_5 in the 3s analysis of the LRO triplet data or using the JC distance (Figure 4.11) between *A. merus* (R) and *A. melas* (L), rescaled relative to the autosomes (auto). While θs for modern species may be used, the data from Fontaine et al. (2015) are haploid consensus sequences generated from diploid samples, so that information concerning nucleotide diversity may be partially lost. Estimates based on the ancestral θ_5 may be affected by different population sizes for the autosomes and the X chromosome, while the JC distance between species may be similarly affected since it consists of one component after the species split and another component from the coalescent time in the ancestral species. Thus among the different relative-rate estimates, those based on τ_1 may be preferable.

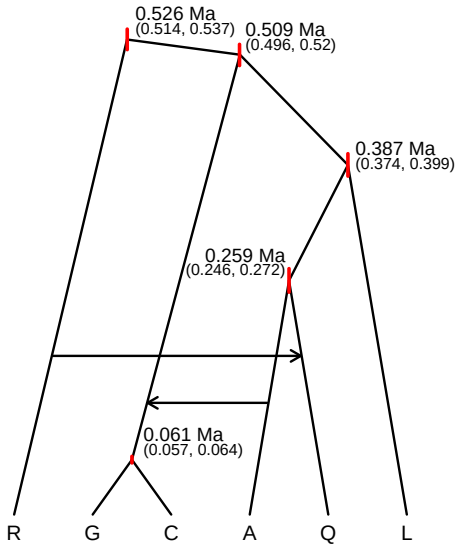


Figure 4.12: Estimated species phylogeny with introgression for the *A. gambiae* species complex. Divergence times are based on the divergence time estimates (τs) from the Xag data (Figure 4.9B). Arrows indicate that introgression occurred between species pairs only, without timing information. The 95% HPD intervals are in parentheses, also shown as vertical bars.

predominantly the reduction of neutral mutation rate in the coding regions, highlighting the utility of coding loci in MSC-based analysis (Shi and Yang, 2018).

The estimates of τs were also largely proportional between tree ii for the 2L and tree xi for the Xag (after the GC clade was removed to make the two trees equivalent). The slope was 1.060 for the noncoding loci ($r^2 = 0.989$ using four pairs of τ) (Figure 4.10C), suggesting that the X chromosome has a slightly higher mutation rate than those on the autosomes (Table 4.5). For the coding loci, the τ estimates were also nearly proportional but the slope was 0.674 (with $r^2 = 0.997$) (Figure 4.10D), indicating that the coding regions in the Xag region are more conserved than those in the autosomes.

To translate the estimates of τs and θs into geological times and population sizes, a mutation rate has to be assumed. Since no mutation rate estimates were available for the *Anopheles*, we used the *Drosophila* rate of 2.8×10^{-9} mutations per site per generation (Keightley et al., 2014) and 11 generations per year (The *Anopheles gambiae* 1000 Genomes Consortium, 2017). This placed the root of the *A. gambiae* species complex at 0.526 (0.514, 0.537) Ma (with the 95% HPD interval), the divergence of the GC clade from the (L(AQ)) clade at 0.509 (0.496, 0.520) Ma, and the divergence of *A. gambiae* and *A. coluzzii* at 0.061 (0.057, 0.064) Ma (Figure 4.12). The G-C divergence time is expected to be a serious underestimate because BPP analysis ignored the gene flow between *A. gambiae* and *A. coluzzii*, which should cause those sister species to be preferentially grouped together and their divergence time to be underestimated (Leaché et al., 2014). If we instead used the mutation rate of 5.5×10^{-9} (Schrider et al., 2013), the ages would be younger by about a half. Our age estimates were much younger than those reported in Fontaine et al. (2015), where the age of the clade was estimated to be 1.85 (0.93, 2.77) Ma (with ±1.96 standard deviation). Using the mutation rate of 1.1×10^{-9} and 10 generations per year as in Fontaine et al. (2015) would increase our estimate for

the age of the clade to 1.472 (1.440, 1.503) Ma, which was still 20% younger than their date. One reason for this difference could be attributed to the method: the concatenation analysis misinterprets sequence divergence as species divergence by ignoring the distinction between gene trees and species trees, and may thus be expected to overestimate node ages. Furthermore, concatenation produces systematically biased estimates of species divergence times and population sizes because it incorrectly attributed the genealogical heterogeneity across the genome as variation in the evolutionary rate among sites (Ogilvie et al., 2017). However, other factors could also affect the estimates in geological times. For instance, the mutation rate of *Anopheles* could differ from that of *Drosophila*. The effect of population size dynamics was also ignored in our analysis.

4.4 Discussion

4.4.1 The species phylogeny provides a framework for studying the evolution of ecological and epidemiological characters

While we support the major conclusion of Fontaine et al. (2015) that the Xag region of the X chromosome represents the species branching order, and the conflicting autosome phylogenies are a result of extensive introgression, the inferred species trees are different: Fontaine et al. (2015) inferred tree ix (defined in Figure 4.2) for the Xag, while we inferred tree xi, with *A. merus* diverging first. We simulated data of 10,000 loci using tree xi for the Xag as well as the parameter estimates on tree xi, but with migration from *A. arabiensis* to the common ancestor of *A. gambiae* and *A. coluzzii* at the rate of 0.22 migrants per generation. We then analysed the data using BPP and concatenation/ML. Both methods generated tree ii as the best estimate in every block of 100 loci. This result suggests that the level of introgression estimated from the autosomes is indeed sufficient to mislead phylogenetic and species-tree methods to infer an incorrect phylogeny, as discussed in Mallet et al. (2016).

The knowledge of the species tree (Figure 4.12) provides a necessary foundation for studying the evolution of ecologically and epidemiologically important traits in this group of species. For instance, physiological adaptations to saltwater breeding in *A. merus* and *A. melas* must have evolved independently, as postulated earlier based on the pattern of chromosomal inversions (Coluzzi and Sabatini, 1969; Coluzzi et al., 1979; Kamali et al., 2012). Another example, discussed in Section 4.3.6, is the evolution of 2La region, which has been shown to be associated with susceptibility to *Plasmodium* infection in natural populations (Riehle et al., 2017). In this case, our species tree provided a simpler and more parsimonious interpretation of chromosome inversions than the species tree from Fontaine et al. (2015), although more data and analysis will be required in order to reach a definitive conclusion, for examples, breakpoint analysis of high-quality fully-phased whole genomes from natural populations.

4.4.2 Implications for the evolution of vectorial capacity

Species in the *A. gambiae* complex exhibit different levels of vector status for human malaria. Among the six species considered here, all but *A. quadriannulatus* are considered dominant vectors, with *A. arabiensis* and *A. gambiae+A. coluzzii* being major vectors (Sinka et al., 2010). We say that vector species have vectorial capacity for malaria, while nonvectors do not. Here, vectorial capacity refers to the ability of a mosquito to serve as a disease vector by supporting parasite development and reproduction as well as transmitting the disease among the human host. This epidemiologically important trait is likely to be a product of complex interactions of many physiological and ecological factors such as host preference, feeding behaviour, longevity, population density, susceptibility to parasite infection and vector competence (Cohuet et al., 2010; White et al., 2011), mediated by environmental factors such as ambient temperature, humidity and mosquito microbiome (Lefèvre et al., 2013) as well as parasite genetic factors. Effects of these individual factors and their interactions on vectorial capacity, particularly on malaria transmission, are largely unknown (Lefèvre et al., 2018).

Based on the inferred species tree (Figure 4.12), it is possible that the acquisition of malaria vectorial capacity could have occurred once in the common ancestor of the species complex and then lost in *A. quadriannulatus*. We note, however, that phylogenetic reconstruction of single complex characters such as vectorial capacity may involve high uncertainties. Identifying specific genes or gene families and amino acid changes in protein-coding sequences associated with differences in vectorial capacity as well as estimating the timing of speciation and introgression events (e.g. by obtaining more relevant mutation rate estimates for these species) may lead to a more definitive conclusion about the origin and evolution of this epidemiologically important trait. Furthermore, analysing genomes of other dominant malaria vectors outside of the *A. gambiae* complex in Africa as well as other parts of the world will likely to shed more light on the evolution of vectorial capacity (Neafsey et al., 2015).

Given high degrees of anthropophily in *A. gambiae* and *A. coluzzii*, it is possible that their evolution has become tightly linked with human evolution. Their divergence time was estimated to be about 61 ka (thousand years ago), with 95% HPD interval (57 ka, 64 ka) (Figure 4.12). This broadly agrees with a recent estimate of the divergence time between the human *Plasmodium falciparum* and the gorilla *P. praefalciparum* at about 50 ka (Otto et al., 2018), with 95% confidence interval (40 ka, 60 ka)[1]. This date range is considerably more recent than the presence of first modern humans in sub-Saharan Africa >100 ka (Schlebusch et al., 2012), but does overlap with several major migration events in Africa such as the migration of click-language-speaking hunter-gatherers and the out-of-Africa dispersal (reviewed in Nielsen et al. (2017)).

It has been speculated that speciation and geographical expansions of *A. gambiae* and *A. coluzzii* could

[1]This date estimate was based on fitting an approximate IM model implemented in the G-PhoCS (Gronau et al., 2011) program to noncoding regions without untranslated (UTR) sequences.

be associated with the onset of the so-called Bantu expansion. This refers a series of human population expansions in Central and South Africa driven by the development of agriculture, which would facilitate the lifestyle of *A. gambiae* and *A. coluzzii* as we observe today (e.g. The *Anopheles gambiae* 1000 Genomes Consortium (2017)). However, the Bantu expansion only occurred around 5.6 ka (Li et al., 2014). This recent date suggests that the breeding habitats of the mosquitoes associated with human agriculture, and possibly the ability to effectively transmit human malaria, might be a relatively recent traits in the species complex (Coluzzi et al., 2002; Ayala and Coluzzi, 2005). Alternatively, the *Anopheles* mutation rate would have to be at least an order of magnitude higher to make the divergence time between *A. gambiae* and *A. coluzzii* more recent.

4.4.3 The importance of coalescent-based methods to inferring challenging species trees resulting from radiative speciations

While theoretical studies have suggested that concatenation may be unreliable and even inconsistent when the species tree contains short internal branches and large ancestral populations (Kubatko and Degnan, 2007; Roch and Steel, 2015), real data examples are relatively rare (Giarla and Esselstyn, 2015; Shi and Yang, 2018). The *A. gambiae* species complex appears to be such a case and serves to illustrate the importance of properly accounting for ILS in such analysis and the power of full-likelihood coalescent-based methods in resolving such difficult species phylogenies. Our analysis of both the real and simulated data suggests that the JC mutation model assumed in 3s and BPP is adequate for capturing multiple substitutions and recovers the true species tree even in datasets simulated under a more complex GTR+Γ_4 model. Note that sequence divergence between the species in the complex is within 5% (Figure 4.11). The molecular clock assumption also approximately holds as the species are closely related. The impacts of various factors in the inference of shallow species trees, including sequence divergence, model assumptions, recombination and noncoding versus coding data partitioning have been discussed in detail in (Shi and Yang, 2018). Accommodating the gene tree/species tree conflicts and the impact of introgression in a proper statistical framework was found to have the greatest impact on the analysis. While Fontaine et al. (2015) emphasised incomplete lineage sorting, their methods ignored it. We expect such methodological differences to be important in other similar challenging species tree problems.

The data we analysed, which consist of widely separated loci from the genome, constitute only a small fraction of the genome (21.6 Mb / 278 Mb = 7.8%). However, with so many loci, the species trees can be resolved with high confidence and accuracy. The sliding-window analysis of Fontaine et al. (2015) used more data in terms of base pairs, but it ignored the gene-tree heterogeneity across the genome and may be misled by ILS.

The agreement of our inferred species tree, tree xi based on the genomic sequences from the Xag region of the X chromosome, with the chromosomal inversion phylogeny of Kamali et al. (2012) highlights the

utility of both chromosomal rearrangements and genomic sequence data in resolving the challenging phylogeny of the *A. gambiae* complex. As pointed out before (White et al., 2011; Kamali et al., 2012), sequence data tend to have weak phylogenetic information when the species are closely related and the sequences are highly similar. However the number of characters is huge. Chromosomal rearrangements represent rare or even unique events, which make each character highly informative. However they may often be compatible with multiple interpretations. It is common to assume that species sharing inversions form a clade or are sister taxa (White et al., 2011; Kamali et al., 2012), but such inference is not safe when the original and derived states of the inversion are unknown and the inferred tree is unrooted. For a long time *A. quadriannulatus* was considered the closest to the ancestral lineage because it has a large number of hosts, feeds on animal blood, tolerates temperate climates and possesses a 'standard' karyotype (Coluzzi et al., 1979, 2002). However, this was based on misinterpretations of the unrooted phylogeny.

Chapter 5

Summary

In this thesis, we address two aspects of Bayesian data analysis, namely, inference computation (Chapter 3) and real data analysis under complex phylogenetic models (Chapter 4), with relevant background reviewed in Chapters 1 and 2, respectively.

In Chapter 3, we illustrated three design principles for boosting efficiency of the MH algorithm, a popular MCMC algorithm for posterior inference. High efficiency translates to a reduction in the running time required to achieve a specified level of accuracy of the posterior quantities of interest. First, we proposed several new proposal kernels for the MH algorithm based on the idea of reducing autocorrelations. In particular, the Mirror kernels (Section 3.1.2) directly inject negative autocorrelations into the Markov chain. A similar idea of introducing negative correlations to improve the efficiency of an estimator is known as antithetics in the statistics literature, but its application to Markov chain kernels has not been explored until now. The Mirror kernels provide a first example where the antithetic principle is directly applied to the Markov chain samples. Moreover, we empirically demonstrated that the Mirror kernels can achieve efficiency >100% in many cases. This is a relatively rare example of MCMC algorithms that give super-efficient estimators. Second, our examples showed that a sequence of well-designed one-dimensional proposals can be more efficient than a single d-dimensional proposal. Third, we suggest variable transformations such as whitening (3.3) and CDF-based transformations (Section 3.3.3) as a generic tool for boosting efficiency of MCMC. Compared with many state-of-the art MCMC algorithms such as MALA and HMC, our approach appears to be much simpler to implement and computationally less expensive. However, our examples were mostly unimodal. The proposed design principles may not work well for high dimensional targets with multiple modes or when the target variables have highly non-linear dependencies and correlation structures.

In Chapter 4, we performed Bayesian inference of the species tree of the *Anopheles gambiae* mosquito species complex using whole-genome sequence data. Unlike previous studies, our approach is based on a proper probabilistic data model of genomic sequences and a coalescent-based model that explicitly captures gene-tree heterogeneity across genomic loci. Our analyses of real and simulated data lead to a ro-

bust conclusion about the species phylogeny and introgression events that are more consistent with other sources of evidence such as chromosome inversions than previous work, providing a basis for studying the evolution of ecological and epidemiological traits. Of particular interest are traits associated with the mosquito's ability to support and transmit malaria parasites, and traits associated with insecticide resistance. For instance, our species tree suggests several hypotheses about the evolutionary history of the 2La region, where inversion polymorphism in natural populations has been shown to be associated with differential susceptibility to *Plasmodium* infection (Sections 4.3.6 and 4.4.2). However, differentiating among these scenarios would require more complete population genomic data that are representative of diverse natural mosquito populations across geographical distributions as well as further analyses such as functional annotations, gene synteny, chromosome evolution (including fine-scale characterisation of inversion breakpoint structures) and adaptive evolution of introgressed loci. Ultimately, to fully appreciate the evolution of vectorial capacity in these vector species, the species tree of the *A. gambiae* complex obtained here should be put into a wider phylogeographic context that includes other *Anopheles* species outside of this species complex, some of which are also major human malaria vectors in Africa or other regions of the world, and the evolution of mosquito vectors should be understood in the context of co-evolution with both the *Plasmodium* parasites and the human host.

Bibliography

Adler, S. L. (1981). Over-relaxation method for the Monte Carlo evaluation of the partition function for multiquadratic actions. *Phys. Rev. D*, 23:2901–2904. page 76

Andersen, L. N., Mailund, T., and Hobolth, A. (2014). Efficient computation in the IM model. *J. Math. Biol.*, 68(6):1423–1451. pages 43, 44, and 49

Andrieu, C., de Freitas, N., Doucet, A., and Jordan, M. I. (2003). An introduction to MCMC for machine learning. *Mach. Learn.*, 50(1-2):5–43. page 25

Ayala, D., Acevedo, P., Pombi, M., Dia, I., Boccolini, D., Costantini, C., Simard, F., and Fontenille, D. (2017). Chromosome inversions and ecological plasticity in the main African malaria mosquitoes. *Evolution*, 71(3):686–701. pages 82 and 102

Ayala, F. J. and Coluzzi, M. (2005). Chromosome speciation: Humans, *Drosophila*, and mosquitoes. *Proc. Natl. Acad. Sci. USA*, 102(suppl 1):6535–6542. pages 82 and 110

Bai, Y., Craiu, R. V., and Di Narzo, A. F. (2011). Divide and conquer: a mixture-based approach to regional adaptation for MCMC. *J. Comput. Graph. Statist.*, 20(1):63–79. page 27

Barone, P. and Frigessi, A. (1990). Improving stochastic relaxation for Gaussian random fields. *Probab. Eng. Inf. Sci.*, 4(3):369–389. page 76

Becquet, C. and Przeworski, M. (2009). Learning about modes of speciation by computational approaches. *Evolution*, 63(10):2547–2562. page 49

Bédard, M., Douc, R., and Moulines, E. (2012). Scaling analysis of multiple-try MCMC methods. *Stochastic Process. Appl.*, 122(3):758–786. page 31

Bédard, M., Douc, R., and Moulines, E. (2014). Scaling analysis of delayed rejection MCMC methods. *Methodol. Comput. Appl. Probab.*, 16(4):811–838. page 31

Beerli, P. and Felsenstein, J. (1999). Maximum-likelihood estimation of migration rates and effective population numbers in two populations using a coalescent approach. *Genetics*, 152(2):763–773. page 43

Besansky, N. J., Krzywinski, J., Lehmann, T., Simard, F., Kern, M., Mukabayire, O., Fontenille, D., Touré, Y., and Sagnon, N. (2003). Semipermeable species boundaries between *Anopheles gambiae* and *Anopheles*

arabiensis: Evidence from multilocus DNA sequence variation. *Proc. Natl. Acad. Sci. USA*, 100(19):10818–10823. pages 82 and 95

Besansky, N. J., Powell, J. R., Caccone, A., Hamm, D. M., Scott, J. A., and Collins, F. H. (1994). Molecular phylogeny of the *Anopheles gambiae* complex suggests genetic introgression between principal malaria vectors. *Proc. Natl. Acad. Sci. USA*, 91(15):6885–6888. page 95

Beskos, A., Pillai, N., Roberts, G., Sanz-Serna, J.-M., and Stuart, A. (2013). Optimal tuning of the hybrid Monte Carlo algorithm. *Bernoulli*, 19(5A):1501–1534. page 31

Beskos, A., Roberts, G., and Stuart, A. (2009). Optimal scalings for local Metropolis-Hastings chains on nonproduct targets in high dimensions. *Ann. Appl. Probab.*, 19(3):863–898. page 31

Bierkens, J., Fearnhead, P., and Roberts, G. (2016). The zig-zag process and super-efficient sampling for Bayesian analysis of big data. *ArXiv e-prints*, 1607.03188. page 25

Bouchard-Côté, A., Vollmer, S. J., and Doucet, A. (2018). The bouncy particle sampler: a nonreversible rejection-free Markov chain Monte Carlo method. *J. Amer. Statist. Assoc.*, 113(522):855–867. page 25

Box, G. E. P. (1980). Sampling and Bayes' inference in scientific modelling and robustness. *J. Roy. Statist. Soc. Ser. A*, 143(4):383–430. page 21

Brooks, S., Gelman, A., Jones, G. L., and Meng, X.-L., editors (2011). *Handbook of Markov chain Monte Carlo*. Chapman & Hall/CRC Handbooks of Modern Statistical Methods. CRC Press. page 25

Browne, W. J. (2006). MCMC algorithms for constrained variance matrices. *Comput. Statist. Data Anal.*, 50(7):1655–1677. page 56

Burgess, R. and Yang, Z. (2008). Estimation of hominoid ancestral population sizes under bayesian coalescent models incorporating mutation rate variation and sequencing errors. *Mol. Biol. Evol.*, 25(9):1979–1994. page 84

Carpenter, B., Gelman, A., Hoffman, M., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., and Riddell, A. (2017). Stan: a probabilistic programming language. *J. Stat. Softw.*, 76(1):1–32. page 65

Chen, M.-H. and Shao, Q.-M. (1999). Monte Carlo estimation of Bayesian credible and HPD intervals. *J. Comput. Graph. Statist.*, 8(1):69–92. page 33

Cohuet, A., Harris, C., Robert, V., and Fontenille, D. (2010). Evolutionary forces on *Anopheles*: what makes a malaria vector? *Trends Parasitol.*, 26(3):130–136. page 109

Coluzzi, M. and Sabatini, A. (1969). Cytogenetic observations on the salt water species, *Anopheles merus* and *Anopheles melas*, of the *gambiae* complex. *Parassitologia*, 11:177–186. page 108

Coluzzi, M., Sabatini, A., della Torre, A., Di Deco, M. A., and Petrarca, V. (2002). A polytene chromosome analysis of the *Anopheles gambiae* species complex. *Science*, 298(5597):1415–1418. pages 82, 95, 101, 102, 110, and 111

Coluzzi, M., Sabatini, A., Petrarca, V., and Deco, M. D. (1979). Chromosomal differentiation and adaptation to human environments in the *Anopheles gambiae* complex. *Trans. R. Soc. Trop. Med. Hyg.*, 73(5):483–497. pages 82, 95, 108, and 111

Costa, R. J. and Wilkinson-Herbots, H. (2017). Inference of gene flow in the process of speciation: An efficient maximum-likelihood method for the isolation-with-initial-migration model. *Genetics*, 205(4):1597–1618. page 49

Coulibaly, B., Kone, R., Barry, M. S., Emerson, B., Coulibaly, M. B., Niare, O., Beavogui, A. H., Traore, S. F., Vernick, K. D., and Riehle, M. M. (2016). Malaria vector populations across ecological zones in Guinea Conakry and Mali, West Africa. *Malar. J.*, 15(1):191. page 102

Craiu, R. V. and Rosenthal, J. S. (2014). Bayesian computation via Markov chain Monte Carlo. *Annu. Rev. Stat. Appl.*, 1(1):179–201. page 25

Crawford, J. E., Riehle, M. M., Guelbeogo, W. M., Gneme, A., Sagnon, N., Vernick, K. D., Nielsen, R., and Lazzaro, B. P. (2015). Reticulate speciation and barriers to introgression in the *Anopheles gambiae* species complex. *Genome Biol. Evol.*, 7(11):3116–3131. page 95

Dalquen, D. A., Zhu, T., and Yang, Z. (2017). Maximum likelihood implementation of an isolation-with-migration model for three species. *Syst. Biol.*, 66(3):379–398. pages 13, 44, 45, 47, 49, 83, 87, and 98

De Maio, N., Schlötterer, C., and Kosiol, C. (2013). Linking great apes genome evolution across time scales using polymorphism-aware phylogenetic models. *Mol. Biol. Evol.*, 30(10):2249–2262. page 37

De Maio, N., Schrempf, D., and Kosiol, C. (2015). PoMo: An allele frequency-based approach for species tree estimation. *Syst. Biol.*, 64(6):1018–1031. page 37

Degnan, J. H. and Rosenberg, N. A. (2009). Gene tree discordance, phylogenetic inference and the multispecies coalescent. *Trends Ecol. Evol.*, 24(6):332–340. page 37

della Torre, A., Merzagora, L., Powell, J. R., and Coluzzi, M. (1997). Selective introgression of paracentric inversions between two sibling species of the *Anopheles gambiae* complex. *Genetics*, 146(1):239–244. page 103

Devroye, L. (1986). *Non-uniform random variate generation*. Springer-Verlag New York. pages 24 and 56

Diaconis, P. (2009). The Markov chain Monte Carlo revolution. *Bull. Amer. Math. Soc. (N.S.)*, 46(2):179–205. page 25

Edwards, S. V., Cloutier, A., and Baker, A. J. (2017). Conserved nonexonic elements: A novel class of marker for phylogenomics. *Syst. Biol.*, 66(6):1028–1044. page 35

Edwards, S. V., Xi, Z., Janke, A., Faircloth, B. C., McCormack, J. E., Glenn, T. C., Zhong, B., Wu, S., Lemmon, E. M., Lemmon, A. R., Leaché, A. D., Liu, L., and Davis, C. C. (2016). Implementing and testing the multispecies coalescent model: A valuable paradigm for phylogenomics. *Mol. Phylogenet. Evol.*, 94:447–462. pages 42 and 83

Eriksson, J. S., de Sousa, F., Bertrand, Y. J. K., Antonelli, A., Oxelman, B., and Pfeil, B. E. (2018). Allele phasing is critical to revealing a shared allopolyploid origin of *Medicago arborea* and *M. strasseri* (Fabaceae). *BMC Evol. Biol.*, 18(1):9. page 36

Folk, R. A., Soltis, P. S., Soltis, D. E., and Guralnick, R. (2018). New prospects in the detection and comparative analysis of hybridization in the tree of life. *Am. J. Bot.*, 105(3):364–375. page 42

Fontaine, M. C., Pease, J. B., Steele, A., Waterhouse, R. M., Neafsey, D. E., Sharakhov, I. V., Jiang, X., Hall, A. B., Catteruccia, F., Kakani, E., Mitchell, S. N., Wu, Y.-C., Smith, H. A., Love, R. R., Lawniczak, M. K., Slotman, M. A., Emrich, S. J., Hahn, M. W., and Besansky, N. J. (2015). Extensive introgression in a malaria vector species complex revealed by phylogenomics. *Science*, 347(6217):1258524. pages 15, 36, 81, 82, 83, 84, 86, 88, 89, 92, 93, 94, 95, 96, 97, 98, 102, 103, 104, 106, 107, 108, and 110

Frigessi, A., Gåsemyr, J., and Rue, H. (2000). Antithetic coupling of two Gibbs sampler chains. *Ann. Statist.*, 28(4):1128–1149. page 77

García, B. A., Caccone, A., Mathiopoulos, K. D., and Powell, J. R. (1996). Inversion monophyly in african anopheline malaria vectors. *Genetics*, 143(3):1313–1320. page 95

Gelfand, A. E., Smith, A. F. M., and Lee, T.-M. (1992). Bayesian analysis of constrained parameter and truncated data problems using Gibbs sampling. *J. Amer. Statist. Assoc.*, 87(418):523–532. page 56

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2014). *Bayesian data analysis*. CRC Press, third edition. pages 21 and 28

Gelman, A., Meng, X.-L., and Stern, H. (1996a). Posterior predictive assessment of model fitness via realized discrepancies. *Statist. Sinica*, 6(4):733–760. page 22

Gelman, A., Roberts, G. O., and Gilks, W. R. (1996b). Efficient Metropolis jumping rules. In *Bayesian statistics, 5*, pages 599–607. pages 17, 29, 30, 31, 32, and 64

Gelman, A. and Shalizi, C. R. (2013). Philosophy and the practice of Bayesian statistics. *Br. J. Math. Stat. Psychol.*, 66(1):8–38. page 22

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 6:721–741. page 26

Geyer, C. J. (1992). Practical Markov chain Monte Carlo. *Statist. Sci.*, 7(4):473–483. page 29

Giarla, T. C. and Esselstyn, J. A. (2015). The challenges of resolving a rapid, recent radiation: Empirical and simulated phylogenomics of philippine shrews. *Syst. Biol.*, 64(5):727–740. page 110

Gillespie, J. H. and Langley, C. H. (1979). Are evolutionary rates really variable? *J. Mol. Evol.*, 13(1):27–34. page 39

Girolami, M. and Calderhead, B. (2011). Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 73(2):123–214. pages 65, 67, and 72

Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732. page 27

Gronau, I., Hubisz, M. J., Gulko, B., Danko, C. G., and Siepel, A. (2011). Bayesian inference of ancient human demography from individual genome sequences. *Nat Genet*, 43(10):1031–1034. page 109

Gruenstaeudl, M., Reid, N. M., Wheeler, G. L., and Carstens, B. C. (2016). Posterior predictive checks of coalescent models: P2C2M, an R package. *Mol. Ecol. Resour.*, 16(1):193–205. page 42

Guan, Y. and Krone, S. M. (2007). Small-world MCMC and convergence to multi-modal distributions: from slow mixing to fast mixing. *Ann. Appl. Probab.*, 17(1):284–304. page 27

Hall, A. B., Papathanos, P.-A., Sharma, A., Cheng, C., Akbari, O. S., Assour, L., Bergman, N. H., Cagnetti, A., Crisanti, A., Dottorini, T., Fiorentini, E., Galizi, R., Hnath, J., Jiang, X., Koren, S., Nolan, T., Radune, D., Sharakhova, M. V., Steele, A., Timoshevskiy, V. A., Windbichler, N., Zhang, S., Hahn, M. W., Phillippy, A. M., Emrich, S. J., Sharakhov, I. V., Tu, Z. J., and Besansky, N. J. (2016). Radical remodeling of the Y chromosome in a recent radiation of malaria mosquitoes. *Proc. Natl. Acad. Sci. USA*, 113(15):E2114–E2123. page 82

Hammersley, J. M. and Morton, K. W. (1954). Poor man's Monte Carlo. *J. Roy. Statist. Soc. Ser. B.*, 16:23–38; discussion 61–75. page 24

Hammersley, J. M. and Morton, K. W. (1956). A new Monte Carlo technique: antithetic variates. *Proc. Cambridge Philos. Soc.*, 52:449–475. pages 29 and 77

Harrison, R. G. and Larson, E. L. (2014). Hybridization, introgression, and the nature of species boundaries. *J. Hered.*, 105(S1):795–809. page 42

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109. page 26

He, Q. and Knowles, L. L. (2016). Identifying targets of selection in mosaic genomes with machine learning: applications in *Anopheles gambiae* for detecting sites within locally adapted chromosomal inversions. *Mol. Ecol.*, 25(10):2226–2243. pages 103 and 104

Heled, J. and Bouckaert, R. (2013). Looking for trees in the forest: summary tree from posterior samples. *BMC Evol. Biol.*, 13(1):221. page 33

Hey, J. (2010). Isolation with migration models for more than two populations. *Mol. Biol. Evol.*, 27(4):905–920. pages 43, 44, and 83

Hey, J. and Nielsen, R. (2004). Multilocus methods for estimating population sizes, migration rates and divergence time, with applications to the divergence of *Drosophila pseudoobscura* and *D. persimilis*. *Genetics*, 167(2):747–760. pages 43, 44, and 83

Hey, J. and Nielsen, R. (2007). Integration within the felsenstein equation for improved markov chain monte carlo methods in population genetics. *Proc. Natl. Acad. Sci. USA*, 104(8):2785–2790. pages 43 and 44

Hobolth, A., Andersen, L. N., and Mailund, T. (2011). On computing the coalescence time density in an isolation-with-migration model with few samples. *Genetics*, 187(4):1241–1243. pages 43 and 44

Hoffman, M. D. and Gelman, A. (2014). The No-U-Turn Sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *J. Mach. Learn. Res.*, 15:1593–1623. pages 71 and 77

Holden, L., Hauge, R., and Holden, M. (2009). Adaptive independent Metropolis-Hastings. *Ann. Appl. Probab.*, 19(1):395–413. page 27

Horai, S., Hayasaka, K., Kondo, R., Tsugane, K., and Takahata, N. (1995). Recent African origin of modern humans revealed by complete sequences of hominoid mitochondrial DNAs. *Proc. Natl. Acad. Sci. USA*, 92(2):532–536. page 67

Huber, M. L. (2016). *Perfect simulation*. CRC Press. page 25

Hudson, R. R. (1983). Testing the constant-rate neutral allele model with protein sequence data. *Evolution*, 37(1):203–217. page 39

Innan, H. and Watanabe, H. (2006). The effect of gene flow on the coalescent time in the human-chimpanzee ancestral population. *Mol. Biol. Evol.*, 23(5):1040–1047. page 49

Jones, G. L. (2004). On the Markov chain central limit theorem. *Probab. Surv.*, 1:299–320. page 28

Jukes, T. H. and Cantor, C. R. (1969). Evolution of protein molecules. In Munro, H. N., editor, *Mammalian Protein Metabolism*, volume 3, pages 21–132. pages 40, 41, 68, 85, and 94

Kamali, M., Xia, A., Tu, Z., and Sharakhov, I. V. (2012). A new chromosomal phylogeny supports the repeated origin of vectorial capacity in malaria mosquitoes of the *Anopheles gambiae* complex. *PLOS Pathog.*, 8(10):e1002960. pages 95, 96, 108, 110, and 111

Katoh, K. and Standley, D. M. (2013). MAFFT multiple sequence alignment software version 7: Improvements in performance and usability. *Mol. Biol. Evol.*, 30(4):772–780. page 84

Keightley, P. D., Ness, R. W., Halligan, D. L., and Haddrill, P. R. (2014). Estimation of the spontaneous mutation rate per nucleotide site in a *Drosophila melanogaster* full-sib family. *Genetics*, 196(1):313–320. pages 86 and 107

Kemeny, J. G. and Snell, J. L. (1960). *Finite Markov chains*. Springer-Verlag New York. page 29

Kimura, M. and Weiss, G. H. (1964). The stepping stone model of population structure and the decrease of genetic correlation with distance. *Genetics*, 49(4):561–576. page 43

Kubatko, L. S. and Degnan, J. H. (2007). Inconsistency of phylogenetic estimates from concatenated data under coalescence. *Syst. Biol.*, 56(1):17–24. pages 36, 83, and 110

Lanier, H. C., Huang, H., and Knowles, L. L. (2014). How low can you go? the effects of mutation rate on the accuracy of species-tree estimation. *Mol. Phylogenet. Evol.*, 70:112–119. page 35

Lanier, H. C. and Knowles, L. L. (2012). Is recombination a problem for species-tree analyses? *Syst. Biol.*, 61(4):691–701. page 42

Leaché, A. D., Harris, R. B., Rannala, B., and Yang, Z. (2014). The influence of gene flow on species tree estimation: A simulation study. *Syst. Biol.*, 63(1):17–30. pages 42 and 107

Lefèvre, T., Ohm, J., Dabiré, K. R., Cohuet, A., Choisy, M., Thomas, M. B., and Cator, L. (2018). Transmission traits of malaria parasites within the mosquito: Genetic variation, phenotypic plasticity, and consequences for control. *Evol. Appl.*, 11(4):456–469. page 109

Lefèvre, T., Vantaux, A., Dabiré, K. R., Mouline, K., and Cohuet, A. (2013). Non-genetic determinants of mosquito competence for malaria parasites. *PLOS Pathog.*, 9(6):e1003365. page 109

Lemmon, E. M. and Lemmon, A. R. (2013). High-throughput genomic data in systematics and phylogenetics. *Annu. Rev. Ecol. Evol. Syst.*, 44(1):99–121. pages 35 and 36

Li, S., Schlebusch, C., and Jakobsson, M. (2014). Genetic variation reveals large-scale population expansion and migration during the expansion of Bantu-speaking peoples. *Proc. Royal Soc. B*, 281(1793):20141448. page 110

Li, W.-H. (1976). Distribution of nucleotide differences between two randomly chosen cistrons in a subdivided population: The finite island model. *Theor. Popul. Biol.*, 10:303–308. page 43

Liu, J. S. (2008). *Monte Carlo strategies in scientific computing*. Springer-Verlag New York. page 25

Liu, L. and Edwards, S. V. (2009). Phylogenetic analysis in the anomaly zone. *Syst. Biol.*, 58(4):452–460. pages 36 and 96

Liu, L., Xi, Z., Wu, S., Davis, C. C., and Edwards, S. V. (2015a). Estimating phylogenetic trees from genome-scale data. *Ann. N. Y. Acad. Sci.*, 1360(1):36–53. page 42

Liu, Y., Gelman, A., and Zheng, T. (2015b). Simulation-efficient shortest probability intervals. *Stat. Comput.*, 25(4):809–819. page 33

Lohse, K., Chmelik, M., Martin, S. H., and Barton, N. H. (2016). Efficient strategies for calculating blockwise likelihoods under the coalescent. *Genetics*, 202(2):775–786. page 44

Lohse, K., Harrison, R. J., and Barton, N. H. (2011). A general method for calculating likelihoods under the coalescent process. *Genetics*, 189(3):977–987. pages 44 and 84

Long, C. and Kubatko, L. (2018). The effect of gene flow on coalescent-based species-tree inference. *Syst. Biol.*, 67(1):770–785. page 42

Maddison, W. P. (1997). Gene trees in species trees. *Syst. Biol.*, 46(3):523–536. page 36

Mallet, J., Besansky, N., and Hahn, M. W. (2016). How reticulated are species? *BioEssays*, 38(2):140–149. pages 42 and 108

Marshall, T. and Roberts, G. (2012). An adaptive approach to Langevin MCMC. *Stat. Comput.*, 22(5):1041–1057. page 71

Maruyama, T. (1970). Analysis of population structure: I. One-dimensional stepping-stone models of finite length. *Ann. Hum. Genet.*, 34(2):201–219. page 43

Mau, B. and Newton, M. A. (1997). Phylogenetic inference for binary data on dendograms using Markov chain Monte Carlo. *J. Comput. Graph. Statist.*, 6(1):122–131. page 33

McCormack, J. E., Faircloth, B. C., Crawford, N. G., Gowaty, P. A., Brumfield, R. T., and Glenn, T. C. (2012). Ultraconserved elements are novel phylogenomic markers that resolve placental mammal phylogeny when combined with species-tree analysis. *Genome Res.*, 22(4):746–754. page 35

Mendes, F. K. and Hahn, M. W. (2016). Gene tree discordance causes apparent substitution rate variation. *Syst. Biol.*, 65(4):711–721. page 36

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *J. Chem. Phys.*, 21(6):1087–1092. page 26

Mira, A. (2001). Efficiency of finite state space Monte Carlo Markov chains. *Statist. Probab. Lett.*, 54(4):405–411. page 75

Molloy, E. K. and Warnow, T. (2018). To include or not to include: The impact of gene filtering on species tree estimation methods. *Syst. Biol.*, 67(2):285–303. page 35

Nachman, M. W. and Payseur, B. A. (2012). Recombination rate variation and speciation: theoretical predictions and empirical results from rabbits and mice. *Philos. Trans. Royal Soc. B*, 367(1587):409–421. page 49

Nath, H. and Griffiths, R. (1993). The coalescent in two colonies with symmetric migration. *J. Math. Biol.*, 31(8):841–851. page 43

Neafsey, D. E., Lawniczak, M. K. N., Park, D. J., Redmond, S. N., Coulibaly, M. B., Traoré, S. F., Sagnon, N., Costantini, C., Johnson, C., Wiegand, R. C., Collins, F. H., Lander, E. S., Wirth, D. F., Kafatos, F. C., Besansky, N. J., Christophides, G. K., and Muskavitch, M. A. T. (2010). SNP genotyping defines complex gene-flow boundaries among African malaria vector mosquitoes. *Science*, 330(6003):514–517. pages 95 and 102

Neafsey, D. E., Waterhouse, R. M., Abai, M. R., Aganezov, S. S., Alekseyev, M. A., Allen, J. E., Amon, J., Arcà, B., Arensburger, P., Artemov, G., Assour, L. A., Basseri, H., Berlin, A., Birren, B. W., Blandin, S. A., Brockman, A. I., Burkot, T. R., Burt, A., Chan, C. S., Chauve, C., Chiu, J. C., Christensen, M., Costantini, C., Davidson, V. L. M., Deligianni, E., Dottorini, T., Dritsou, V., Gabriel, S. B., Guelbeogo, W. M., Hall, A. B., Han, M. V., Hlaing, T., Hughes, D. S. T., Jenkins, A. M., Jiang, X., Jungreis, I., Kakani, E. G., Kamali, M., Kemppainen, P., Kennedy, R. C., Kirmitzoglou, I. K., Koekemoer, L. L., Laban, N., Langridge, N., Lawniczak, M. K. N., Lirakis, M., Lobo, N. F., Lowy, E., MacCallum, R. M., Mao, C., Maslen, G., Mbogo, C., McCarthy, J., Michel, K., Mitchell, S. N., Moore, W., Murphy, K. A., Naumenko, A. N., Nolan, T., Novoa, E. M., O'Loughlin, S., Oringanje, C., Oshaghi, M. A., Pakpour, N., Papathanos, P. A., Peery, A. N., Povelones, M., Prakash, A., Price, D. P., Rajaraman, A., Reimer, L. J., Rinker, D. C., Rokas, A., Russell, T. L., Sagnon, N., Sharakhova, M. V., Shea, T., Simão, F. A., Simard, F., Slotman, M. A., Somboon, P., Stegniy, V., Struchiner, C. J., Thomas, G. W. C., Tojo, M., Topalis, P., Tubio, J. M. C., Unger, M. F., Vontas, J., Walton, C., Wilding, C. S., Willis, J. H., Wu, Y.-C., Yan, G., Zdobnov, E. M., Zhou, X., Catteruccia, F., Christophides, G. K., Collins, F. H., Cornman, R. S., Crisanti, A., Donnelly, M. J., Emrich, S. J., Fontaine, M. C., Gelbart, W., Hahn, M. W., Hansen, I. A., Howell, P. I., Kafatos, F. C., Kellis, M., Lawson, D., Louis, C., Luckhart, S., Muskavitch, M. A. T., Ribeiro, J. M., Riehle, M. A., Sharakhov, I. V., Tu, Z., Zwiebel, L. J., and Besansky, N. J. (2015). Highly evolvable malaria vectors: The genomes of 16 *Anopheles* mosquitoes. *Science*, 347(6217):1258522. pages 36, 81, 83, and 109

Neal, P. and Roberts, G. (2011). Optimal scaling of random walk Metropolis algorithms with non-Gaussian proposals. *Methodol. Comput. Appl. Probab.*, 13(3):583–601. page 31

Neal, R. M. (2011). MCMC using Hamiltonian dynamics. In *Handbook of Markov chain Monte Carlo*, Chapman & Hall/CRC Handb. Mod. Stat. Methods, pages 113–162. CRC Press. pages 31 and 77

Nichols, R. (2001). Gene trees and species trees are not the same. *Trends Ecol. Evol.*, 16(7):358–364. page 36

Nielsen, R., Akey, J. M., Jakobsson, M., Pritchard, J. K., Tishkoff, S., and Willerslev, E. (2017). Tracing the peopling of the world through genomics. *Nature*, 541(7637):302–310. page 109

Nielsen, R. and Wakeley, J. (2001). Distinguishing migration from isolation: A Markov chain Monte Carlo approach. *Genetics*, 158(2):885–896. page 43

Nieto Feliner, G., Álvarez, I., Fuertes-Aguilar, J., Heuertz, M., Marques, I., Moharrek, F., Piñeiro, R., Riina,

R., Rosselló, J. A., Soltis, P. S., and Villa-Machío, I. (2017). Is homoploid hybrid speciation that rare? An empiricist's view. *Heredity*, 118:513–516. page 42

Notohara, M. (1990). The coalescent and the genealogical process in geographically structured population. *J. Math. Biol.*, 29(1):59–75. page 43

Ogilvie, H. A., Bouckaert, R. R., and Drummond, A. J. (2017). StarBEAST2 brings faster species tree inference and accurate estimates of substitution rates. *Mol. Biol. Evol.*, 34(8):2101–2114. pages 37, 41, and 108

Ogilvie, H. A., Heled, J., Xie, D., and Drummond, A. J. (2016). Computational performance and statistical accuracy of *BEAST and comparisons with other methods. *Syst. Biol.*, 65(3):381–396. page 37

O'Loughlin, S. M., Magesa, S., Mbogo, C., Mosha, F., Midega, J., Lomas, S., and Burt, A. (2014). Genomic analyses of three malaria vectors reveals extensive shared polymorphism but contrasting population histories. *Mol. Biol. Evol.*, 31(4):889–902. pages 82, 95, and 102

Otto, T. D., Gilabert, A., Crellen, T., Böhme, U., Arnathau, C., Sanders, M., Oyola, S. O., Okouga, A. P., Boundenga, L., Willaume, E., Ngoubangoye, B., Moukodoum, N. D., Paupy, C., Durand, P., Rougeron, V., Ollomo, B., Renaud, F., Newbold, C., Berriman, M., and Prugnolle, F. (2018). Genomes of all known members of a plasmodium subgenus reveal paths to virulent human malaria. *Nat. Microbiol.*, 3(6):687–697. page 109

Pamilo, P. and Nei, M. (1988). Relationships between gene trees and species trees. *Mol. Biol. Evol.*, 5(5):568–583. page 39

Papaspiliopoulos, O., Roberts, G. O., and Sköld, M. (2007). A general framework for the parametrization of hierarchical models. *Statist. Sci.*, 22(1):59–73. page 27

Payseur, B. A. and Rieseberg, L. H. (2016). A genomic perspective on hybridization and speciation. *Mol. Ecol.*, 25(11):2337–2360. page 42

Peskun, P. H. (1973). Optimum Monte-Carlo sampling using Markov chains. *Biometrika*, 60:607–612. pages 29 and 30

Pillai, N. S., Stuart, A. M., and Thiéry, A. H. (2012). Optimal scaling and diffusion limits for the Langevin algorithm in high dimensions. *Ann. Appl. Probab.*, 22(6):2320–2356. page 31

Rannala, B. and Yang, Z. (1996). Probability distribution of molecular evolutionary trees: A new method of phylogenetic inference. *J. Mol. Evol.*, 43(3):304–311. page 33

Rannala, B. and Yang, Z. (2003). Bayes estimation of species divergence times and ancestral population sizes using DNA sequences from multiple loci. *Genetics*, 164(4):1645–1656. pages 37, 39, 83, 85, and 93

Rannala, B. and Yang, Z. (2017). Efficient Bayesian species tree inference under the multispecies coalescent. *Syst. Biol.*, 66(5):823–842. pages 37, 41, 83, and 85

Riehle, M. M., Bukhari, T., Gneme, A., Guelbeogo, W. M., Coulibaly, B., Fofana, A., Pain, A., Bischoff, E., Renaud, F., Beavogui, A. H., Traore, S. F., Sagnon, N., and Vernick, K. D. (2017). The *Anopheles gambiae* 2La chromosome inversion is associated with susceptibility to *Plasmodium falciparum* in Africa. *eLife*, 6:e25813. pages 83, 102, 103, 104, and 108

Robert, C. and Casella, G. (2011). A short history of Markov chain Monte Carlo: subjective recollections from incomplete data. *Statist. Sci.*, 26(1):102–115. page 25

Robert, C. P. and Casella, G. (2004). *Monte Carlo statistical methods*. Springer-Verlag New York, second edition. page 25

Roberts, G. O., Gelman, A., and Gilks, W. R. (1997). Weak convergence and optimal scaling of random walk Metropolis algorithms. *Ann. Appl. Probab.*, 7(1):110–120. pages 13, 31, and 32

Roberts, G. O. and Rosenthal, J. S. (1998). Optimal scaling of discrete approximations to Langevin diffusions. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 60(1):255–268. pages 13, 31, and 32

Roberts, G. O. and Rosenthal, J. S. (2004). General state space Markov chains and MCMC algorithms. *Probab. Surv.*, 1:20–71. pages 25 and 26

Roberts, G. O. and Sahu, S. K. (1997). Updating schemes, correlation structure, blocking and parameterization for the Gibbs sampler. *J. Roy. Statist. Soc. Ser. B*, 59(2):291–317. page 27

Roch, S. and Steel, M. (2015). Likelihood-based tree reconstruction on a concatenation of aligned sequence data sets can be statistically inconsistent. *Theor. Popul. Biol.*, 100:56–62. pages 36, 83, and 110

Rosenbluth, M. N. and Rosenbluth, A. W. (1955). Monte Carlo calculation of the average extension of molecular chains. *J. Chem. Phys.*, 23(2):356–359. page 24

Sayyari, E., Whitfield, J. B., and Mirarab, S. (2017). Fragmentary gene sequences negatively impact gene tree and species tree reconstruction. *Mol. Biol. Evol.*, 34(12):3279–3291. page 35

Schlebusch, C. M., Skoglund, P., Sjödin, P., Gattepaille, L. M., Hernandez, D., Jay, F., Li, S., De Jongh, M., Singleton, A., Blum, M. G. B., Soodyall, H., and Jakobsson, M. (2012). Genomic variation in seven Khoe-San groups reveals adaptation and complex African history. *Science*, 338(6105):374–379. page 109

Schrempf, D., Minh, B. Q., De Maio, N., von Haeseler, A., and Kosiol, C. (2016). Reversible polymorphism-aware phylogenetic models and their application to tree inference. *J. Theor. Biol.*, 407:362–370. page 37

Schrider, D. R., Houle, D., Lynch, M., and Hahn, M. W. (2013). Rates and genomic consequences of spontaneous mutational events in *Drosophila melanogaster*. *Genetics*, 194(4):937–954. pages 86 and 107

Sethuraman, A. and Hey, J. (2016). IMa2p – parallel MCMC and inference of ancient demography under the isolation with migration (IM) model. *Mol. Ecol. Resour.*, 16(1):206–215. page 43

Sharakhov, I. V., White, B. J., Sharakhova, M. V., Kayondo, J., Lobo, N. F., Santolamazza, F., della Torre, A., Simard, F., Collins, F. H., and Besansky, N. J. (2006). Breakpoint structure reveals the unique origin of an interspecific chromosomal inversion (2La) in the *Anopheles gambiae* complex. *Proc. Natl. Acad. Sci. USA*, 103(16):6258–6262. pages 102 and 104

Shi, C.-M. and Yang, Z. (2018). Coalescent-based analyses of genomic sequence data provide a robust resolution of phylogenetic relationships among major groups of gibbons. *Mol. Biol. Evol.*, 35(1):159–179. pages 37, 42, 107, and 110

Sinka, M. E., Bangs, M. J., Manguin, S., Coetzee, M., Mbogo, C. M., Hemingway, J., Patil, A. P., Temperley, W. H., Gething, P. W., Kabaria, C. W., Okara, R. M., Van Boeckel, T., Godfray, H. C. J., Harbach, R. E., and Hay, S. I. (2010). The dominant *Anopheles* vectors of human malaria in Africa, Europe and the Middle East: occurrence data, distribution maps and bionomic précis. *Parasites Vectors*, 3(1):117. page 109

Sinka, M. E., Bangs, M. J., Manguin, S., Rubio-Palis, Y., Chareonviriyaphap, T., Coetzee, M., Mbogo, C. M., Hemingway, J., Patil, A. P., Temperley, W. H., Gething, P. W., Kabaria, C. W., Burkot, T. R., Harbach, R. E., and Hay, S. I. (2012). A global map of dominant malaria vectors. *Parasites Vectors*, 5(1):69. page 82

Slotman, M. A., Della Torre, A., Calzetta, M., and Powell, J. R. (2005). Differential introgression of chromsomal regions between *Anopheles gambiae* and *An. arabiensis*. *Am. J. Trop. Med. Hyg.*, 73(2):326–335. pages 82 and 103

Solís-Lemus, C., Yang, M., and Ané, C. (2016). Inconsistency of species tree methods under gene flow. *Syst. Biol.*, 65(5):843–851. page 42

Sousa, V. and Hey, J. (2013). Understanding the origin of species with genome-scale data: modelling gene flow. *Nat. Rev. Genet.*, 14:404–414. page 42

Stamatakis, A. (2014). RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics*, 30(9):1312–1313. pages 83 and 85

Strasburg, J. L. and Rieseberg, L. H. (2010). How robust are "isolation with migration" analyses to violations of the IM model? A simulation study. *Mol. Biol. Evol.*, 27(2):297–310. page 49

Strobeck, C. (1987). Average number of nucleotide differences in a sample from a single subpopulation: A test for population subdivision. *Genetics*, 117(1):149–153. page 43

Szöllősi, G. J., Tannier, E., Daubin, V., and Boussau, B. (2015). The inference of gene trees with species trees. *Syst. Biol.*, 64(1):e42–e62. pages 36 and 37

Tajima, F. (1983). Evolutionary relationship of DNA sequences in finite populations. *Genetics*, 105(2):437–460. page 39

Takahata, N. (1988). The coalescent in two partially isolated diffusion populations. *Genetical Research*, 52(3):213–222. page 43

Takahata, N. (1989). Gene genealogy in three related populations: consistency probability between gene and population trees. *Genetics*, 122(4):957–966. page 39

Takahata, N. and Nei, M. (1985). Gene genealogy and variance of interpopulational nucleotide differences. *Genetics*, 110(2):325–344. pages 39 and 43

Takahata, N., Satta, Y., and Klein, J. (1995). Divergence time and population size in the lineage leading to modern humans. *Theor. Popul. Biol.*, 48(2):198–221. pages 40 and 48

Takken, W., Eling, W., Hooghof, J., Dekker, T., Hunt, R., and Coetzee, M. (1999). Susceptibility of *Anopheles quadriannulatus* theobald (Diptera: Culicidae) to *Plasmodium falciparum*. *Trans. R. Soc. Trop. Med. Hyg.*, 93(6):578–580. page 82

Takken, W. and Verhulst, N. O. (2013). Host preferences of blood-feeding mosquitoes. *Annu. Rev. Entomol.*, 58(1):433–453. page 82

The *Anopheles gambiae* 1000 Genomes Consortium (2017). Genetic diversity of the African malaria vector *Anopheles gambiae*. *Nature*, 552:96–100. pages 84, 86, 104, 107, and 110

Tierney, L. (1994). Markov chains for exploring posterior distributions. *Ann. Statist.*, 22(4):1701–1762. pages 26, 27, 28, and 76

Tierney, L. (1998). A note on Metropolis-Hastings kernels for general state spaces. *Ann. Appl. Probab.*, 8(1):1–9. page 30

Tigano, A. and Friesen, V. L. (2016). Genomics of local adaptation with gene flow. *Mol. Ecol.*, 25(10):2144–2164. page 42

von Neumann, J. (1951). Various techniques used in connection with random digits. In *Monte Carlo Method, National Bureau of Standards Applied Math*, volume 12, pages 36–38. page 24

Wakeley, J. (1996). Pairwise differences under a general model of population subdivision. *J. Genet.*, 75(1):81–89. page 43

Wang, Y. and Hey, J. (2010). Estimating divergence parameters with small samples from a large number of loci. *Genetics*, 184(2):363–379. pages 43 and 44

Wang, Z., Mohamed, S., and de Freitas, N. (2013). Adaptive Hamiltonian and Riemann manifold Monte Carlo. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, pages 1462–1470. page 77

Wang-Sattler, R., Blandin, S., Ning, Y., Blass, C., Dolo, G., Touré, Y. T., Torre, A. d., Lanzaro, G. C., Steinmetz, L. M., Kafatos, F. C., and Zheng, L. (2007). Mosaic genome architecture of the *Anopheles gambiae* species complex. *PLOS ONE*, 2(11):e1249. pages 82 and 95

Weetman, D., Steen, K., Rippon, E. J., Mawejje, H. D., Donnelly, M. J., and Wilding, C. S. (2014). Contemporary gene flow between wild *An. gambiae s.s.* and *An. arabiensis*. *Parasites Vectors*, 7(1):345. page 102

Wen, D. and Nakhleh, L. (2018). Coestimating reticulate phylogenies and gene trees from multilocus sequence data. *Syst. Biol.*, 67(3):439–457. page 44

Wen, D., Yu, Y., Hahn, M. W., and Nakhleh, L. (2016a). Reticulate evolutionary history and extensive introgression in mosquito species revealed by phylogenetic network analysis. *Mol. Ecol.*, 25(11):2361–2372. pages 83 and 89

Wen, D., Yu, Y., and Nakhleh, L. (2016b). Bayesian inference of reticulate phylogenies under the multispecies network coalescent. *PLOS Genet.*, 12(5):e1006006. pages 83 and 89

White, B. J., Collins, F. H., and Besansky, N. J. (2011). Evolution of *Anopheles gambiae* in relation to humans and malaria. *Annu. Rev. Ecol. Evol. Syst.*, 42(1):111–132. pages 82, 109, and 111

Wiebe, A., Longbottom, J., Gleave, K., Shearer, F. M., Sinka, M. E., Massey, N. C., Cameron, E., Bhatt, S., Gething, P. W., Hemingway, J., Smith, D. L., Coleman, M., and Moyes, C. L. (2017). Geographical distributions of African malaria vector sibling species and evidence for insecticide resistance. *Malar. J.*, 16(1):85. page 82

Wilkinson-Herbots, H. M. (1998). Genealogy and subpopulation differentiation under various models of population structure. *J. Math. Biol.*, 37(6):535–585. page 43

Wilkinson-Herbots, H. M. (2008). The distribution of the coalescence time and the number of pairwise nucleotide differences in the "isolation with migration" model. *Theor. Popul. Biol.*, 73(2):277–288. page 43

Wilkinson-Herbots, H. M. (2012). The distribution of the coalescence time and the number of pairwise nucleotide differences in a model of population divergence or speciation with an initial period of gene flow. *Theor. Popul. Biol.*, 82(2):92–108. page 49

World Health Organization (2017). *World malaria report 2017*. page 5

Wright, S. (1931). Evolution in mendelian populations. *Genetics*, 16(2):97–159. page 43

Wright, S. (1943). Isolation by distance. *Genetics*, 28(2):114–138. page 43

Xu, B. and Yang, Z. (2016). Challenges in species tree estimation under the multispecies coalescent model. *Genetics*, 204(4):1353–1368. pages 37 and 83

Xu, X., Meng, X.-L., and Yu, Y. (2013). Thank God that regressing Y on X is not the same as regressing X on Y: direct and indirect residual augmentations. *J. Comput. Graph. Statist.*, 22(3):598–622. page 27

Yang, Z. (1994a). Estimating the pattern of nucleotide substitution. *J. Mol. Evol.*, 39:105–111. pages 87 and 93

Yang, Z. (1994b). Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: Approximate methods. *J. Mol. Evol.*, 39:306–314. page 93

Yang, Z. (2002). Likelihood and Bayes estimation of ancestral population sizes in hominoids using data from multiple loci. *Genetics*, 162(4):1811–1823. pages 40, 48, and 88

Yang, Z. (2014). *Molecular Evolution: A Statistical Approach*. Oxford University Press. page 94

Yang, Z. (2015). The BPP program for species tree estimation and species delimitation. *Curr. Zool.*, 61(5):854–865. pages 41, 83, 85, 86, 88, and 93

Yang, Z. and Rannala, B. (2010). Bayesian species delimitation using multilocus sequence data. *Proc. Natl. Acad. Sci. USA*, 107(20):9264–9269. page 86

Yang, Z. and Rannala, B. (2014). Unguided species delimitation using DNA sequence data from multiple loci. *Mol. Biol. Evol.*, 31(12):3125–3135. pages 41, 83, 85, and 86

Yang, Z. and Rodríguez, C. E. (2013). Searching for efficient Markov chain Monte Carlo proposal kernels. *Proc. Natl. Acad. Sci. USA*, 110(48):19307–19312. pages 17, 30, 32, 51, 52, 56, and 78

Yu, Y. and Meng, X.-L. (2011). To center or not to center: that is not the question—an ancillarity-sufficiency interweaving strategy (ASIS) for boosting MCMC efficiency. *J. Comput. Graph. Statist.*, 20(3):531–570. page 27

Zhang, C., Ogilvie, H. A., Drummond, A. J., and Stadler, T. (2018). Bayesian inference of species networks from multilocus sequence data. *Mol. Biol. Evol.*, 35(2):504–517. page 44

Zhu, T. and Yang, Z. (2012). Maximum likelihood implementation of an isolation-with-migration model with three species for testing speciation with gene flow. *Mol. Biol. Evol.*, 29(10):3131–3142. pages 44, 83, and 87