Syst. Biol. 64(5):741–751, 2015

DOI:10.1093/sysbio/syv030
Advance Access publication May 25, 2015

Monte Carlo Strategies for Selecting Parameter Values in Simulation Experiments

JESSICA W. LEIGH* AND DAVID BRYANT

Department of Mathematics and Statistics, University of Otago, P.O. Box 56, Dunedin 9054, New Zealand
**Correspondence to be sent to: Department of Mathematics and Statistics, University of Otago, P.O. Box 56, Dunedin 9054, New Zealand; E-mail: jleigh@maths.otago.ac.nz.*

Received 27 July 2014; reviews returned 21 October 2014; accepted 12 May 2015
Associate Editor: Cécile Ané

Abstract.—Simulation experiments are used widely throughout evolutionary biology and bioinformatics to compare models, promote methods, and test hypotheses. The biggest practical constraint on simulation experiments is the computational demand, particularly as the number of parameters increases. Given the extraordinary success of Monte Carlo methods for conducting inference in phylogenetics, and indeed throughout the sciences, we investigate ways in which Monte Carlo framework can be used to carry out simulation experiments more efficiently. The key idea is to *sample* parameter values for the experiments, rather than iterate through them exhaustively. Exhaustive analyses become completely infeasible when the number of parameters gets too large, whereas sampled approaches can fare better in higher dimensions. We illustrate the framework with applications to phylogenetics and genetic archaeology. [Importance sampling; Markov chain Monte Carlo; phylogenetic analysis; plant domestication; simulation.]

Simulation experiments based on random, synthetic data are now a fundamental tool throughout the sciences. In evolutionary and systematic biology, simulations are used extensively to compare and promote, different methods and models (e.g. FitzJohn 2010; Guindon et al. 2010; Huang et al. 2010; Pigot et al. 2010; Anisimova et al. 2011; Leaché and Rannala 2011; Wiens and Morrill 2011; Zhang et al. 2011; Grummer et al. 2014; Leaché et al. 2014). Indeed, some of the most influential papers in the area have been completely based on simulations (e.g. Kuhner and Felsenstein 1994; Huelsenbeck 1995) and it has become difficult to publish a methodology paper which does not include some type of simulation experiment.

Simulations typically involve models with parameters that are either unknown or only roughly estimated. One of the biggest challenges when implementing a simulation is deciding which values, or range of values, to use for these parameters. The range of values needs to be broad enough that the experimental results have some level of generality, dense enough that important detail is not missed, and yet not so extensive that the experiment cannot be completed within the limits imposed by computational resources and time. The goal of this article is to propose a new framework for carrying out these simulation experiments, one which differs from existing approaches by the way model parameters are selected.

The standard approach to selecting parameter values is to use what we call a *grid-based strategy*. Some parameters are fixed at single values. For the other parameters, a set of discrete values is chosen in advance, most usually a regularly spaced set of points. The simulation experiment is then conducted by considering all possible tuples of parameter values, carrying out multiple replicates for each tuple.

A grid-based strategy has many advantages, perhaps the most important being that it is intuitive and easy to implement. However, the strategy also suffers from critical limitations (Fig. 1). First and foremost is the curse of dimensionality: As the number of variable parameters grows, the number of grid points increases exponentially (Liu 2008). In practice, this means that researchers need to either be highly selective about which parameters to vary, or to use only a restricted set of possible values, or to gain access to a vast number of fast computers.

Second, when choosing a discrete set of possible values in advance, experimenters run the risk of missing important features which fall between the grid points (Fig. 1b). This is particularly critical when it is not completely clear, in advance, what scale a parameter is operating in. While we might select 20 grid points ranging from 0.1 to 2.0 spaced at equal intervals, the interesting behavior might well be when the parameter is between 10^{-2} and 10^{-3}.

These limitations are not unique to grid-based strategies, and are definitely not unique to the problem of parameter selection in simulations. In fields like numerical integration, enormous progress has been made dealing with these kinds of limitations through the introduction of Monte Carlo techniques, where points are selected randomly according to a carefully tailored sampling scheme. Indeed, Monte Carlo strategies now dominate the range of solution techniques for high-dimensional problems (Liu 2008). It is only natural, then, to explore how these strategies might help with parameter selection in simulation experiments.

There are several theoretical obstacles to overcome if we are to adapt Monte Carlo machinery for parameter choice in simulation experiments. First, it is not immediately obvious which distribution is the appropriate one to sample parameters from. Second, most Monte Carlo algorithms require the value of a distribution to be known, at least up to a scalar constant. Simulation experiments are often based on complex models, and the probabilities of interest are rarely known or computable, even up to a scalar constant.

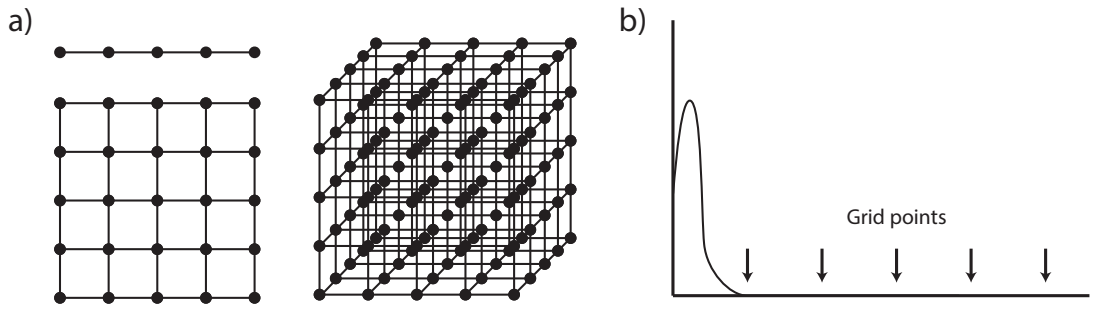

FIGURE 1. Limitations of the grid-based strategy. a) As the number of dimensions increase, the number of combinations of parameter values increases exponentially. b) Selecting a discrete set of sample values runs the risk of missing important features of the problem, particularly if the behavior in question operates at a different scale than was used to set up the experiment.

We address the first obstacle by describing a Bayesian-style framework for conducting simulation experiments in which the distribution of interest is the "posterior" distribution of the parameters conditional on a particular simulation outcome. We say "Bayesian-style" because we never work directly with actual data and the choice of prior is eventually factored out of the final output. We address the second obstacle using the clever algorithms of Marjoram et al. (2003) and Didelot et al. (2011) for conducting *Markov chain Monte Carlo* (MCMC) and *estimating normalizing constants* in the absence of explicit likelihoods. Our methodology, therefore, overlaps with that used for *approximate Bayesian computing* (ABC), though we do not require the reduction of data to summary statistics.

Suppose, for example, we want to determine the probability that a method (say parsimony) correctly infers the phylogenetic tree. The parameters of interest might include the number of taxa, the amount of mutation, and features of the model of sequence evolution. The first step in our framework produces a distribution on these parameters with higher values in the combinations of parameters for which the method produces the correct tree. The second step gives an estimate, as a function of the parameters, of the probability of inferring the correct tree.

The algorithm of Didelot et al. (2011) which we use in the second step is based on *importance sampling*. Importance sampling is a technique for estimating properties of a distribution when we only have samples from an approximation to that distribution. Working with samples from an approximate distribution could easily introduce a lot of bias. To avoid this, we use carefully computed weights, called importance weights, to correct the estimates produced from the approximate distribution; see below for the important weight formulas which we use. Surprisingly, this approach can often produce more accurate estimates than sampling from the original distribution: It can be advantageous to work with an approximation which, for example, puts more weight on 'interesting' parts of the distribution (Liu 2008).

Importance sampling has a major advantage over many Monte Carlo approaches in that it produces a collection of *independent* estimates of the parameter or parameters, meaning that we can use textbook statistics to quantify the uncertainty in those estimates. In our case, the MCMC algorithm and kernel density estimation (KDE) steps provide an approximation of the posterior distribution of parameters, given the simulation outcome of interest, and the degree to which this distribution is incorrect determines the variance of the importance samples. There will, naturally, be a level of tradeoff between the effort spent on obtaining a better approximation and the effort spent on importance sampling.

In some respects, our work is related to the *Bayesian melding* technique, as introduced by Raftery et al. (1995) and Poole and Raftery (2000), and developed further by Sevcíková et al. (2007, 2011). Bayesian melding is a full Bayesian approach to analyzing simulation models, in the sense that it requires priors for the model parameters and uses sampling algorithms to study a posterior distribution of model features. The importance sampling techniques used in these articles, however, all require an explicitly computable likelihood. In contrast, the algorithm we use does not require an explicit likelihood. Our objective is also different: We make use of Bayesian methodology but the end goal is to reconstruct the otherwise intractable likelihood function.

There is also a natural overlap between our work and experimental design. In a classical factorial experiment, the influence of multiple factors on the outcome are explored within a regression framework. Partial factorial designs can be used to efficiently explore interactions between a larger number of factors (Clarke and Kempson 1997). Clearly, we can treat the simulations as experiments. If the assumptions of a conventional factorial experiment are satisfied (notably the model describing how factors interact) then conventional factorial designs will likely provide a more efficient framework for conducting a simulation experiment.

Our framework does not make strong assumptions about the ways different parameters or factors interact. This improves robustness, but may well reduce efficiency. Other differences between our framework and a conventional experimental design are that we choose parameter values randomly, and that we make use of

simulation outcomes when selecting the next set of parameter values.

To illustrate our new framework we conduct two case studies: First, we compare two workhorses of phylogenetic inference, Unweighted Pair Group Method with Arithmetic Mean (UPGMA, Sokal and Sneath 1963) and the neighbor-joining (NJ) algorithm (Saitou and Nei 1987). Both algorithms construct trees (or hierarchies) from distance data, but while UPGMA assumes that the mutations accumulate at a constant rate, NJ accommodates a variable evolutionary rate, albeit at the cost of a slightly higher sampling variance. We will use a simple simulation to identify situations where UPGMA performs at least as well as NJ, primarily to illustrate aspects of our MCMC framework.

Second, we reexamine an important simulation experiment conducted by Allaby et al. (2008a). A key question of prehistoric agriculture is whether domesticated crops have single or multiple origins. Allaby et al. (2008a) used simulations to demonstrate that a standard phylogenetics-based method could infer a single origin even given data from (simulated) crops with multiple origins. Aspects of Allaby et al.'s model, particularly the parameters used, have been contested (Ross-Ibarra and Gaut 2008). Here, we consider the choice of parameter values and ask whether the results of Allaby et al.'s study are robust to error in the model parameters. We conclude that they are indeed robust. We use this example to illustrate how our MCMC approach can be used to explore multidimensional parameter spaces in a way that grid-based strategies cannot.

Bayesian Framework

There are three principal ingredients of any simulation experiment: the simulator itself, the hypotheses being examined, and the scheme used to select the parameter values and replicate counts. The *simulator* takes a vector of parameter values θ and generates synthetic data D according to a distribution $P(D|\theta)$ determined by the underlying model. Generally, the relationship between D and θ is specified only implicitly and actual values of $P(D|\theta)$ are unavailable.

As a starting point, we assume that the goal of the simulation experiment is to study a particular *outcome*, which corresponds to a particular event. For example, if we are using simulations to assess a given hypothesis test, the "outcome" of interest might be that the test gives a false positive and the goal might be to identify parameter values for which the probability of this outcome, a false positive, is large. We let $\mathcal{R}$ denote the outcome of interest, so $P(\mathcal{R}|\theta)$ is the probability of that outcome for a given set of parameter values θ.

The probability $P(\mathcal{R}|\theta)$ for a particular choice of values of θ can be estimated by conducting multiple simulations with the same parameter values θ and recording the proportion of times the outcome of interest $\mathcal{R}$ occurs. This is used in the standard grid-based strategies for simulations:

G1. For all combinations of parameter values θ:

G2. Carry out r simulations using parameters θ (e.g., $r = 100$);

G3. Estimate $P(\mathcal{R}|\theta)$ by the proportion of simulations for which $\mathcal{R}$ occurs.

Our approach is to turn the conventional methodology on its head and follow a Bayesian strategy. We ask: "If the simulation returns the specified outcome, what are the probable values of θ?" In other words, "What is $P(\theta|\mathcal{R})$?" For this probability to make sense, we need to specify a prior distribution $P(\theta)$ on θ. Bayes' rule then gives:

$$P(\theta|\mathcal{R}) = P(\mathcal{R}|\theta)\frac{P(\theta)}{P(\mathcal{R})}. \quad (1)$$

Note that the "likelihood" $P(\mathcal{R}|\theta)$ can (usually) not be directly evaluated.

The samples from $P(\theta|\mathcal{R})$ are of interest in themselves. They indicate the part of parameter space where a particular outcome is most likely to occur. For example, if the simulation experiment is designed to investigate when a particular method or test fails, the values of θ sampled from $P(\theta|$ "method fails") will be those values where the method or test is most vulnerable.

We will, however, make further use of the samples from $P(\theta|\mathcal{R})$. We show how to use the output of the sampling algorithm to estimate the probabilities $P(\mathcal{R}|\theta)$ over the whole range of θ. For example, below we examine when UPGMA finds a tree which is just as accurate as NJ. The samples from $P(\theta|\mathcal{R})$ indicate the areas of parameter space where UPGMA tended to do just as well as NJ; the values $P(\mathcal{R}|\theta)$ tell us the probability that this occurs for a given value of θ.

The standard approach (G1—G3) for estimating $P(\mathcal{R}|\theta)$ would iterate systematically over many different choices for θ, conducting multiple replicates for each θ. Our approach uses the samples from $P(\theta|\mathcal{R})$ to fit a smoothed density estimate, and then an importance sampling technique to estimate the normalization constant $P(\mathcal{R})$.

For simplicity, we focus here on the posterior distribution of the input parameters θ conditional on a simple outcome $\mathcal{R}$. We could similarly examine the conditional distribution $P(f(\theta, D)|\mathcal{R})$ of some function f of θ and the data. For example, if $f(\theta, D)$ was a measure of saturation in the data, then the conditional distribution of $P(f(\theta, D)|\mathcal{R})$ could tell us whether UPGMA outperforms NJ mainly in the presence of saturation. Alternatively, $f(\theta, D)$ might indicate the difference in error for the UPGMA and NJ estimates which could be useful in determining whether UPGMA ever performs substantially better than NJ.

Sampling Algorithm

Our method for sampling from $P(\theta|\mathcal{R})$ is based on the *MCMC without likelihoods* algorithm of Marjoram et al. (2003). Their algorithm takes a data set $\mathcal{D}$ and generates samples from the posterior density $P(\theta|\mathcal{D})$. It does not use the likelihood function $P(\mathcal{D}|\theta)$. Instead, it uses a simulator to generate data sets $\mathcal{D}'$ for varying choices of parameters θ:

F1. If now at θ, propose a move to θ' as sampled from a proposal distribution $q(\theta \to \theta')$.

F2. Generate $\mathcal{D}'$ using model $\mathcal{M}$ with parameters θ'.

F3. If $\mathcal{D}=\mathcal{D}'$, go to F4; otherwise stay at θ and return to F1.

F4. Calculate $h=h(\theta,\theta')=\min\left(1, \frac{P(\theta')q(\theta'\to\theta)}{P(\theta)q(\theta\to\theta')}\right)$.

F5. Accept θ' with probability h; otherwise stay at θ; then return to F1.

Marjoram and colleagues proved that the chain produced has the required stationary distribution, given an appropriate transition kernel q.

The event $\mathcal{D}=\mathcal{D}'$ is an outcome, and this algorithm produces samples of θ conditional on this outcome. To apply this algorithm within our context, we replace the statement $\mathcal{D}=\mathcal{D}'$ with the outcome of interest $\mathcal{R}$ (i.e., the result of a simulation experiment). It can be readily shown that the algorithm will then produce samples from $P(\theta|\mathcal{R})$. To improve efficiency, we also switch the order of steps so that a simulation is carried out only if the move has not been rejected by the prior and the transition kernel. Our implementation of this algorithm is available as Supplementary Material on Dryad at http://dx.doi.org/10.5061/dryad.366j4.

S1. If now at θ, propose a move to θ' according to a transition kernel $q(\theta \to \theta')$.

S2. Calculate $h=h(\theta,\theta')=\min\left(1, \frac{P(\theta')q(\theta'\to\theta)}{P(\theta)q(\theta\to\theta')}\right)$.

S3. With probability h, go to S4; otherwise stay at θ and return to S1.

S4. Generate $\mathcal{D}$ with distribution $P(\cdot|\theta')$ using the simulator and assess $\mathcal{R}$ using $\mathcal{D}$.

S5. If $\mathcal{R}$ occurred, accept θ'; otherwise stay at θ.

S6. Return to S1.

Note that if the transition kernel is symmetric and the prior distribution is uniform then S2 and S3 can be deleted.

With mild conditions on the proposal distribution q, the chain produced by the algorithm converges to the required stationary distribution $P(\theta|\mathcal{R})$, even though knowledge of the likelihood $P(\mathcal{R}|\theta)$ is never required. This property allows us to bypass the need for multiple replicates at each θ value, though multiple replicates are conducted if the proposed moves are rejected. It also provides a way of sampling through the space of θ values without the need to resort to grid sampling.

If the chain can reach every state θ and the simulation outcome is not completely deterministic, the MCMC algorithm is guaranteed to converge. The actual rate of convergence can be assessed using the same suite of tools available for standard MCMC (Brooks et al. 2011; Liu 2008). In situations where $\mathcal{R}$ is especially unlikely, the rejection rate for proposed parameters may be unacceptably high, leading to slow convergence of the chain. This is a scenario often encountered in ABC, and has been at least partly addressed using a variety of strategies (see Csillery et al. 2010). One option is to use a tempering strategy, where we start the chain with an alternative version of $\mathcal{R}$ which is more likely, later switching or converging to the desired choice of $\mathcal{R}$. An alternative strategy is to consider $\mathcal{R}^c$, the outcome that $\mathcal{R}$ does *not* occur, and switch to sampling from $P(\theta|\mathcal{R}^c)$ rather than $P(\theta|\mathcal{R})$. One of these chains will have high acceptance rates where the other does not. The change in distribution can be accommodated using a small modification of the importance sampling step, described below.

Samples from $P(\theta|\mathcal{R})$ tell us which parameter values were likely, given that the simulation resulted in a specified outcome. For example, if a simulation was used to examine when a statistical test gave false positives, we could identify which parameter values were most likely to lead to a false positive result. However, the samples do *not* tell us how likely the false positives are for those parameter values. That is, samples from $P(\theta|\mathcal{R})$ do not provide direct information about $P(\mathcal{R}|\theta)$.

From Equation (1) we have:

$$P(\mathcal{R}|\theta)=\frac{P(\theta|\mathcal{R})P(\mathcal{R})}{P(\theta)}. \quad (2)$$

The prior $P(\theta)$ is known in advance. We estimate $P(\theta|\mathcal{R})$ using a sample produced by the MCMC algorithm described above. The missing ingredient is $P(\mathcal{R})$, the overall probability of the outcome of interest when parameter values are sampled from the prior density. In general, the estimation of the normalizing constant:

$$P(\mathcal{R})=\int_\theta P(\mathcal{R}|\theta)P(\theta)d\theta \quad (3)$$

is a challenging statistical and computational problem, even when the likelihood function $P(\mathcal{R}|\theta)$ is available (Gelman and Meng 1998). Our approach, which is based on Algorithm 4 of Didelot et al. (2011), is a computationally efficient approximation based on importance sampling.

Following (Zhang 1996), we use a KDE approximation to apply importance sampling. KDE is a technique which takes a discrete set of samples (in our case, the output from the MCMC algorithm) and estimates the density or distribution of that the samples were generated from. It is often described as the "smooth" equivalent of a histogram.

The KDE is constructed by replacing each point with a copy of a smooth "kernel" function centered at that point, typically a multidimensional normal (Gaussian) density. The effect is a bit like replacing each point with a pile of sand. Doing this for all points creates a smooth surface filling the gaps between points so that the surface will be higher in the areas where there are more sample points. Importantly, it is relatively easy to evaluate the density produced by a KDE method, and to sample random points from that density.

One important technical issue which arises with kernel density estimates is what to do at a boundary to compensate for the absence of sample points outside the boundary. There is an extensive literature on this problem: In our work we implemented the correction described by Diggle (1985).

The KDE is built using normal kernels, with output from our MCMC sampler. Let $\theta_1, \theta_2, \ldots, \theta_n$ be values generated from $P(\theta|\mathcal{R})$ by our MCMC algorithm, with deletion of initial burn-in and subsampling. Let $K(\theta)$ denote the KDE computed using these samples, so that $K(\theta)$ is an approximation of $P(\theta|\mathcal{R})$. Note that this approximation need not introduce bias in what follows, though a better approximation will lead to lower variance estimates.

We sample M new values $\theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(M)}$ from $K(\cdot)$ and compute the importance weights:

$$w_i = \frac{P(\theta_i)}{K(\theta_i)} \tag{4}$$

following the scheme detailed by (Didelot et al. 2011).

For each $\theta^{(i)}$ value we carry out a simulation with parameters $\theta^{(i)}$. If the outcome $\mathcal{R}$ occurs then we let $h_i = 1$, otherwise we let $h_i = 0$. We then estimate $P(\mathcal{R})$ using:

$$P(\mathcal{R}) \approx \frac{\sum_{i=1}^{M} h_i w_i}{\sum_{i=1}^{M} w_i}. \tag{5}$$

Note that we only need to be able to compute w_i up to a scalar constant. A (rough) estimate for the variance of this estimator is given by:

$$\widehat{\sigma^2} = \frac{1}{M} \frac{\frac{1}{M}\sum_{i=1}^{M} w_i^2 (h_i - P(\mathcal{R}))^2}{\left(\frac{1}{M}\sum_{i=1}^{M} w_i\right)^2}. \tag{6}$$

Note that if $\theta_1, \theta_2, \ldots, \theta_n$ are samples from $P(\theta|\mathcal{R}^c)$ rather than $P(\theta|\mathcal{R})$ then we can still obtain an estimate of $P(\mathcal{R})$. As before, we draw M samples from the KDE $K(\theta)$ constructed from $\theta_1, \ldots, \theta_n$ and compute the importance weights according to Equation (4). For each new sample $\theta^{(i)}$, carry out a simulation using parameters $\theta^{(i)}$. Let $h_i^c = 1$ if $\mathcal{R}$ does *not occur* and $h_i^c = 0$ if it does occur. We then obtain an estimate for $P(\mathcal{R})$ of

$$P(\mathcal{R}) \approx 1 - \frac{\sum_{i=1}^{M} h_i^c w_i}{\sum_{i=1}^{M} w_i}. \tag{7}$$

This observation is particularly useful as a check for the quality of estimation, or when $\mathcal{R}$ is sufficiently unlikely that sampling from $P(\theta|\mathcal{R})$ becomes difficult.

Complex Simulations

So far, the framework we have developed handles only one kind of simulation experiment: We have a specific, binary outcome and wish to know how the probability of that outcome varies across the parameter space. In practice, simulation experiments can rarely be reduced to the study of single, binary outcomes. It is not uncommon to use simulations to explore the behavior of multiple variables, even in the absence of a fixed hypothesis.

As an example, consider the UPGMA versus NJ experiment introduced above and discussed in more detail below. In our case study, we consider a binary outcome $\mathcal{R}$ corresponding to the UPGMA tree being as good as the NJ tree. We might want to know, however, *how much better* UPGMA does when it out-performs NJ. For this, we evaluate how much closer, on average, the UPGMA tree is to the NJ tree in the times that it does as least as well as the NJ tree.

Let Y denote the quantity that we are interested in, say the difference in accuracy between the UPGMA and NJ trees. We can estimate the conditional expectation $E[Y|\mathcal{R}]$ using a slight modification to the earlier algorithm. As before, we draw values $\theta_1, \ldots, \theta_n$ from the KDE $K(\theta)$ approximating $P(\theta|\mathcal{R})$. For each θ_i we simulate data $\mathcal{D}_i$, compute the corresponding value Y_i for Y and let w_i be the importance weight given earlier. An estimate of $E[Y|\mathcal{R}]$, the expected value of Y given that the outcome occurred is then given by:

$$E[Y|\mathcal{R}] \approx \frac{1}{P(\mathcal{R})} \frac{\sum_{i=1}^{M} h_i Y_i w_i}{\sum_{i=1}^{M} w_i}. \tag{8}$$

See below for an application of this idea in Case study 1.

Case Studies

Case Study I: Comparing Two Phylogenetic Algorithms

UPGMA and NJ are two popular methods for inferring phylogenetic trees from matrices of pairwise distances between sequences. While NJ is still considered a relatively reliable method of quickly inferring phylogeny, UPGMA passed out of favor long ago due to its inability to accommodate variation in evolutionary rates across lineages. To demonstrate the application of our framework, we designed simulations over a parameter space in which UPGMA was expected to perform relatively poorly in a known region. Details of the simulation methodology are described in Supplementary Material (available on Dryad at http://dx.doi.org/10.5061/dryad.366j4).

For each simulation, a *true tree* of 30 taxa was generated and used to evolve synthetic genetic data. The

distribution of the tree was governed by two parameters: the *scale*, which controlled the height of the tree (and thereby the level of variability in the data) and the *skew*, which controlled the variation in the mutation rate in different parts of the tree. We used the sequence simulator implemented in the Python package P4 (Foster 2004) to evolve nucleotide sequences along this tree and then produced distance estimates from these. Trees were then inferred using each of the two inference methods. For this comparison, the "outcome of interest" $\mathcal{R}$ corresponded to the event that the tree produced by UPGMA was at least as close to the true tree as was the tree produced by NJ. We measured closeness using the Robinson–Foulds (RF) (Robinson and Foulds 1981) distance.

The MCMC algorithm described above was used to sample values for the parameters $\theta = (\text{scale}, \text{skew})$ from the posterior distribution $P(\theta|\mathcal{R})$. We ran the algorithm for 2,000,000 iterations so as to minimize error due to incomplete convergence. Standard convergence and autocorrelation statistics were computed using our own MATLAB scripts. A KDE approximation for $P(\theta|\mathcal{R})$ was computed from the MCMC sample, with the edge-correction renormalization procedure of (Diggle 1985) applied to compensate for underestimation at the boundaries. The resulting edge-corrected KDE was then used in the importance sampling algorithm to estimate $P(\mathcal{R})$, $P(\mathcal{R}|\theta)$ as a function of θ, and the expected difference in RF distances when UPGMA performs at least as well as NJ.

In practice, we would not want to run such a long MCMC chain to approximate $P(\theta|\mathcal{R})$. We extracted chains of different lengths from the single long chain, and examined the impact of different sample sizes on the estimation of $P(\mathcal{R})$.

We do not have an analytical formula for $P(\mathcal{R}|\theta)$, making it difficult to assess the quality of our estimates directly. Instead we ran further simulations. In one validation experiment, we randomly selected 400-parameter combinations and carried out 100 replicates of the UPGMA–NJ simulation for each combination of skew and scale. In a second validation experiment, this whole procedure was repeated for 400-parameter combinations arranged on a 2D grid. For each parameter combination θ, whether selected randomly or from a grid, we used the simulation to test the hypothesis that true value for $P(\mathcal{R}|\theta)$ equalled the value for $P(\mathcal{R}|\theta)$ estimated by our method.

The experiments run using a grid of parameter values was also used to assess relative efficiency of our approach versus a grid-based approach. We computed variance estimates for the grid-based approach to variance estimates for our approach. We also implemented a third estimator whereby 40,000 combinations of parameter values were selected at random from the (uniform) prior and a single simulation performed for each. The variance for this third approach is given by the standard binomial formula $\frac{P(\mathcal{R})(1-P(\mathcal{R}))}{N}$ where $N = 40,000$.

Finally, we used the modified importance sampling techniques to examine the distribution of the difference in RF scores in the cases that UPGMA did just as well. We used the samples from $P(\theta|\mathcal{R})$ and $P(\theta|\mathcal{R}^c)$ obtained in the previous step, but conducted a new set of importance sample simulations.

Case Study II: Assessing the Robustness of a Published Simulation Experiment

The origin of agriculture was a defining moment of human history, yet there is still debate over the nature and timing of the domestication process (Allaby et al. 2008a,b; Ross-Ibarra and Gaut 2008; Brown et al. 2009; Gross and Olsen 2010; Molina et al. 2011). Whereas archaeobotanical evidence suggests a protracted and complex period of domestication, at least of Fertile Crescent cereals, phylogenetic evidence has suggested a single domestication origin for many crop plants. Allaby and colleagues used a series of simulation experiments to demonstrate that the phylogenetic techniques used could be misleading: Under a particular model of plant domestication, the admixture of two populations that emerged in independent domestication events can appear monophyletic (see Allaby et al. 2008a,b; see also Ross-Ibarra and Gaut 2008).

The model of domestication and admixture used in the simulations of Allaby et al. (2008a) is governed by seven parameters controlling recombination, population sizes, and durations of key intervals in the populations' histories. Four of the seven parameter values (wild population size n_w, bottleneck size n_b, bottleneck duration t_b, and domestication interval t_d) were fixed in advance using estimates from the literature, while the remaining three were set to a limited number of different values. The experiment of Allaby et al. therefore demonstrates that the phylogenetic method is misleading for a *particular choice* of parameter values. We used our simulation framework to investigate how robust their conclusion was to different choices of parameter values.

The model used for these simulations is described in detail in (Allaby et al. 2008b), see also Figure S1 in Supplementary Material (available on Dryad at http://dx.doi.org/10.5061/dryad.366j4). Briefly, frequencies for 20 chromosomes were generated for each of two wild-type plant populations of size n_w. To simulate a bottleneck, n_b individuals were then drawn (with replacement) from each of the wild-type populations, and this small population size was maintained (for each of the two populations) for t_b generations. In each new generation, when a new genome was created, each pair of chromosomes underwent a recombination event with probability p_r. Each population was then allowed to expand to n_d individuals over the course of a single generation, and this new domesticated population size was maintained for t_d generations. The two domesticated populations were then merged and the simulation was continued for

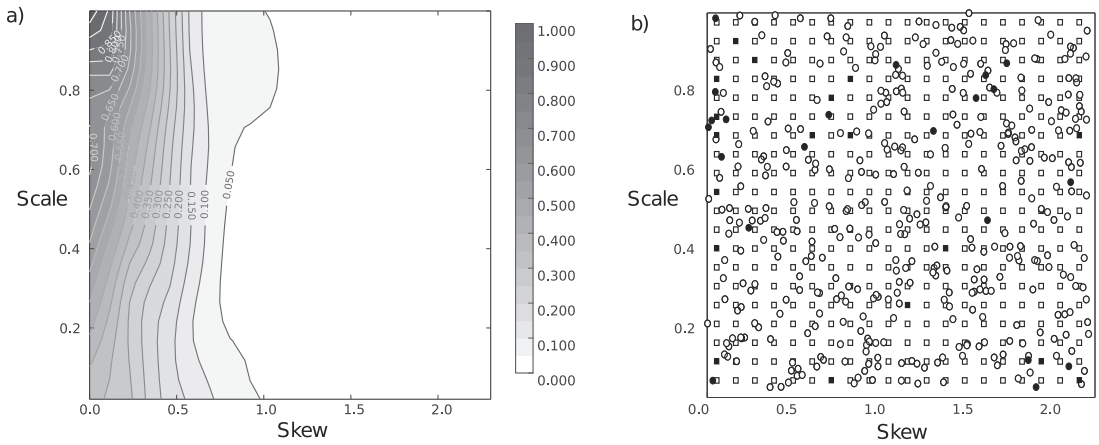

FIGURE 2. Comparison of UPGMA and NJ. a) Our estimate of the likelihood function $P(\mathcal{R}|\Theta)$ giving the probability that UPGMA performs at least as well as NJ; b) Validation of our estimate of $P(\mathcal{R}|\Theta)$ using direct simulations: squares mark parameter choices selected on a grid; Circles mark parameter choices selected uniformly; dark circles or squares mark to parameter choices where the number of replicates for which UPGMA is as close as NJ has probability <0.05, given the estimated value for $P(\mathcal{R}|\Theta)$. A total of 100 replicates were carried out for each choice of parameters.

another t_h generations with this admixed population (of size n_d). At the end of the simulation, an NJ tree was constructed from a distance matrix estimated using the method of Dice (1945) from two individuals sampled from each of the wild-type populations, the domesticated populations, and the admixed population (a total of 10 individuals). If the inferred tree contained a split separating the two members of the admixed population from the others, they were considered to appear (erroneously) monophyletic.

We performed simulations using the seven parameters described above ($\theta=[n_w, n_b, t_b, n_d, t_d, t_h, p_r]$) and an "outcome of interest" $\mathcal{R}$ corresponding to the event that the phylogenetic method falsely inferred monophyly. Prior densities for all parameters were uniform on intervals centered on the values used in Allaby et al. (2008a). We used MCMC to sample parameter values θ from $P(\theta|\mathcal{R})$, and then used KDE and importance sampling techniques to estimate $P(\mathcal{R}|\theta)$ as a function of θ. See the Supplementary Material (available on on Dryad at http://dx.doi.org/10.5061/dryad.366j4) for precise details about the simulation experiment.

We were unable to compare our approach with a grid-based strategy because of a practical, and very important, obstacle. The model has seven parameters so a grid with 20 divisions in each direction gives a total of $20^7 = 1.28 \times 10^9$ grid points. A simulation takes around 2 seconds to run (in our implementation), so running a single simulation at each grid point would have taken up approximately 80 years of computing time.

RESULTS

Case Study I: Comparing Two Phylogenetic Algorithms

Figure 2(a) gives a contour map for our estimate of $P(\mathcal{R}|\theta)$, the probability that UPGMA performs as well as NJ as a function of scale and skew. The simulations indicate that UPGMA tends to perform at least as well as NJ when the skew is small (clocklike data), particularly when the scale is large (data closer to saturation). This makes sense: UPGMA is known to perform better when the substitution rate does not vary across the tree (i.e., the tree is clocklike). It is also a lower variance method than NJ and so will perform better as the data become noisier. Overall, the estimated probability that UPGMA does just as well as NJ was 0.112, with uncertainty $\pm 2\sigma = \pm 0.005$ given by the importance sampling algorithm.

Figure 2(b) compares our estimates to those obtained using a grid-based strategy or those obtained by selecting parameter values uniformly at random. Each square or circle marks a parameter combination used in 100 replicates of the simulation. We calculated 95% confidence intervals for the probability of success at each parameter combination. If the value obtained using our procedures fell outside that confidence interval, we plotted a dark square or circle. Of the 400-parameter combinations on the grid, only 18 of our estimates ($\approx$ 5%, as expected for the 95% confidence interval) fell outside of the confidence interval. Of the 400-parameter combinations selected randomly, only 21 ($\approx$5%) of our estimates fell outside the confidence interval.

In Table 1, we compare the efficiencies of the different approaches, where we consider a method to be more efficient if it requires fewer simulations to obtain an equivalent sampling variance. For this comparison we ignore the fact that the estimates of $P(\mathcal{R})$ obtained by a grid-based strategy are not expected to converge to the true estimate of $P(\mathcal{R})$. While the grid-based strategy will produce unbiased estimates of $P(\mathcal{R}|\theta)$ when θ corresponds to a grid point, the average of these will not necessarily be an unbiased estimate of $P(\mathcal{R})$ as it could miss important features between grid points. In contrast, sampling from the prior and our MCMC+IS approach will produce estimates which converge to the correct value.

TABLE 1. Varying efficiency of different methods of estimating $P(\mathcal{R})$, covering different chain lengths and different numbers of importance samples.

Method	Total sims.	$P(\mathcal{R})$	Variance	E[Diff$\|\mathcal{R}$]	Variance
MCMC (100) +IS (400)	10,400	0.145	7.96×10^{-4}	1.74	0.34
MCMC (100) +IS (4,000)	14,000	0.163	2.69×10^{-4}	1.23	0.073
MCMC (100) +IS (40,000)	50,000	0.115	1.05×10^{-4}	1.43	0.092
MCMC (1000) +IS (400)	100,400	0.102	1.98×10^{-3}	1.92	0.761
MCMC (1000) +IS (4,000)	104,000	0.116	1.76×10^{-4}	1.50	0.129
MCMC (1000) +IS (40,000)	140,000	0.126	2.67×10^{-5}	1.48	0.049
Complementary (1000 + 40,000)	140,000	0.117	1.77×10^{-5}	-	-
Grid strategy	40,000	0.108	1.43×10^{-6}	-	-
Random uniform samples	40,000	0.100	8.41×10^{-5}	-	-

Note: The third column gives the estimate for $P(\mathcal{R})$ obtained using each method, the fourth column gives the estimated sample variances. The final two columns give an estimate for the expected difference in RF distances (with sampling variance) conditional on UPGMA performing at least as well as NJ.

For each approach we list the number of simulations performed, the estimated $P(\mathcal{R})$, and the sampling variance.

At first glance, the importance sampling approach appears extremely efficient: 4000 importance sample replicates gives an (estimated) sample variance roughly equivalent to 40,000 samples from the prior. The catch is that the importance sampling step is preceded by an expensive MCMC step: Each single MCMC sample required 100 simulation experiments. Nevertheless, using only 100 MCMC samples and 4000 importance samples we still obtained a lower variance estimate than that obtained sampling from the prior. The benefit of carrying out a more extensive MCMC analysis appears minimal, though it would be premature to generalize this beyond the single study.

The estimate based on a grid strategy has the smallest sample variance. It is, however, a biased estimate, as only a pre-specified subset of the parameter space (the grid points) is ever sampled. The difference between the grid strategy estimate and the (unbiased) importance sampling estimate is small, but highly *statistically* significant when compared to the sampling variances.

We repeated the analysis using the complementary outcome $\mathcal{R}^c$ in place of $\mathcal{R}$, modifying the importance sampling as described above. From 1000 MCMC samples (subsampling once every 100 iterations) and 40,000 importance samples we obtained an estimate of $P(\mathcal{R}^c) = 0.883$ (or $P(\mathcal{R}) = 0.117$) with sampling variance of 1.77×10^{-5}. The difference between this estimate and those obtained directly is not statistically significant.

The appropriate interpretation of $P(\mathcal{R}) = 0.116$ is that UPGMA does as well as NJ roughly 10% of the time. The next logical question is to ask *how much* better does UPGMA do when it equals NJ? Are there regions of the parameter space where UPGMA might be a better choice of method, even if it is not in general? To this end we used the methodology outlined above to compute the expected difference in the distance between the UPGMA tree and the true tree and the distance between the NJ tree and the true tree. Our 'best' estimate was that E[diff$|\mathcal{R}$], the expected difference conditional on $\mathcal{R}$, was around 1.48. Although this appears small compared to the maximum possible RF distance of 54, note that $\mathcal{R}$ applies even when UPGMA performs *exactly as well as* NJ, and that if UPGMA never performed strictly better than NJ we would have E[diff$|\mathcal{R}] = 0$. We conclude that, for a select few combinations of parameter values, UPGMA can not only perform as well as NJ, but also a little better.

Note that it would be unusual to use an MCMC algorithm to compute an integral in just two dimensions: Numerical integration (quadrature) algorithms are known to be far more efficient when there are only a few variables. We suspect that the same might apply to the problem of parameter selection, particularly if we have some *a priori* knowledge of the shape of the likelihood function itself, whether locally or globally. Of course, incorrect assumptions might improve efficiency but introduce bias. Our MCMC approach, while a little inefficient, is provably unbiased in the limit. As we shall see in the second study, the MCMC approach has the critical advantage that it can be extended to higher dimensional problems.

Case Study II: Assessing the Sensitivity of a Simulation Experiment

Using the MCMC algorithm, KDE, and the importance sampling method, we obtained an estimate of the likelihood function $P(\mathcal{R}|\theta)$, which gives the probability of obtaining a (false) monophyly result for every choice of the seven parameters. Figure 3 gives the average likelihood values for every choice of single parameter and every pair of parameters.

From Figure 3, we see that variation in only two out of the six variables has a significant impact on the probability of monophyly: n_d, the size of the domesticated populations and t_h, the amount of time elapsing following admixture of the two domesticated

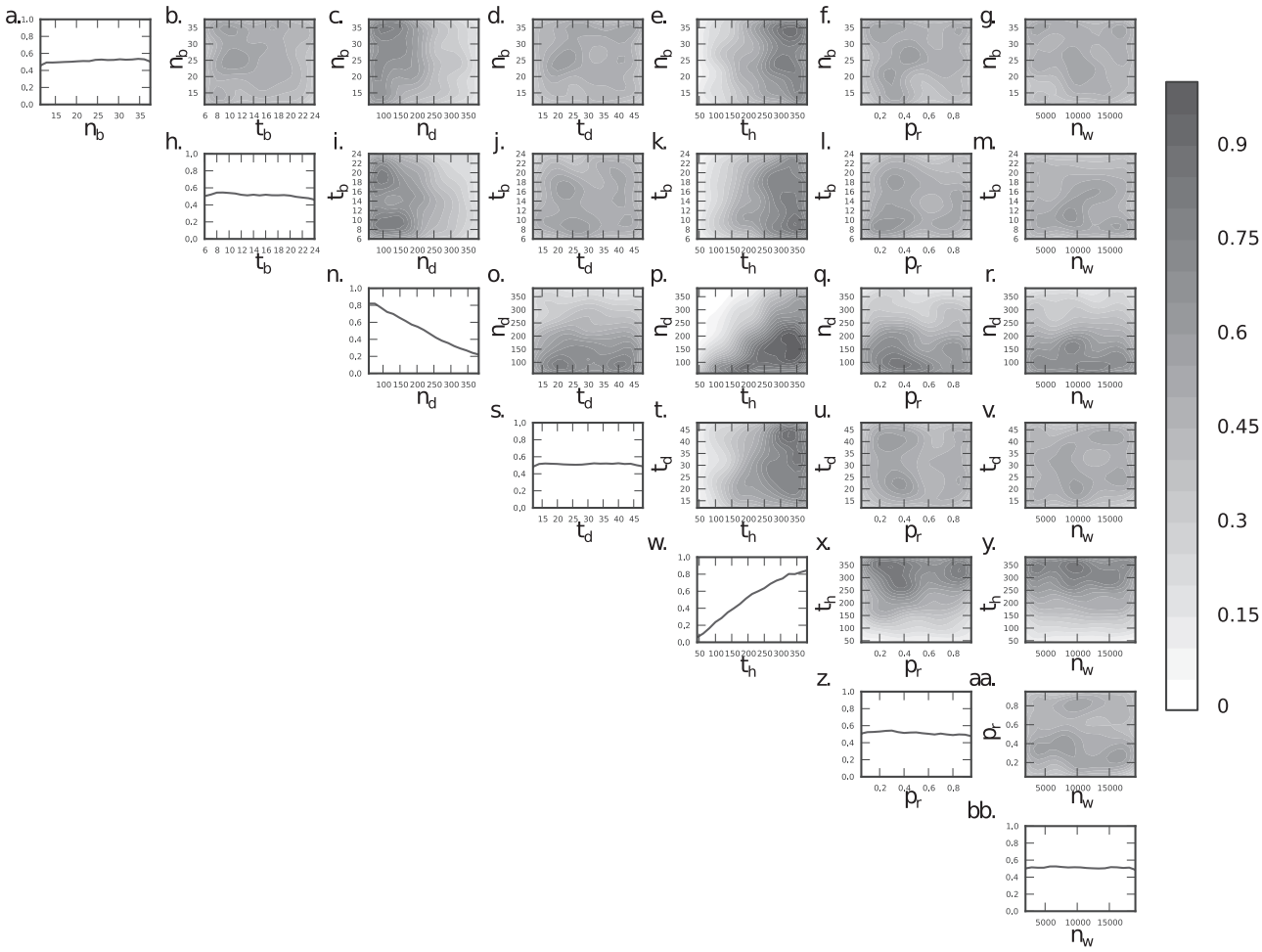

FIGURE 3. Likelihood that a hybrid population erroneously appears to have a monophyletic origin. Likelihood functions showing the probability of incorrectly inferring a monophyletic origin over single and pairs of parameters, with darker shades corresponding to a higher probability of falsely assuming monophyly. n_b: size of the bottleneck populations; t_b: duration of the bottleneck (in generations); n_d: size of the domesticated populations (following post-bottleneck expansion as well as following hybridization); t_d: duration of domestication (prior to hybridization, in generations); t_h: time after hybridization (in generations); p_r: probability of a recombination event; n_w: wild-type population size. See text for a description of how these parameters were used.

populations. The probability of the test returning a false positive increases as the population size decreases or when the number of generations increases. Both trends are consistent with theory, as both decreasing the population size and increasing the number of generations increases the chance that lineages from one of the domesticated lines have become fixed in the admixed population.

Figure 4 gives a more detailed representation of the probability of monophyly as a function of domesticated population size and bottleneck duration. Here, we have fixed the other parameters at the same values used in the simulation of Allaby et al. (2008a), and marked the various parameter values they used for the domesticated population size and the number of generations since admixture ($n_b = 20, t_b = 10, t_d = 20, n_w = 10,000$). We fixed p_r, the recombination probability, at 0.1, one of four values for this parameter used by Allaby and colleagues.

We see immediately that over the range of domesticated population size values considered by Allaby et al. (2008a), the probability of a false positive is generally high (> 0.5), but declines sharply for larger populations. Ross-Ibarra and Gaut (2008) suggest that the small effective population sizes used in Allaby et al. (2008a) are partially responsible for the results described. In response, Allaby et al. (2008b) and colleagues stated that the bottleneck parameters used in their model reflect established dimensions from biological populations .

DISCUSSION

Monte Carlo methodology has made a huge impact on numerical integration and statistical inference (Liu 2008). Here, we have presented a framework for conducting simulation experiments which uses MCMC and importance sampling to consider different combinations of parameter values. Our framework provides an alternative to a conventional "grid-based" strategy,

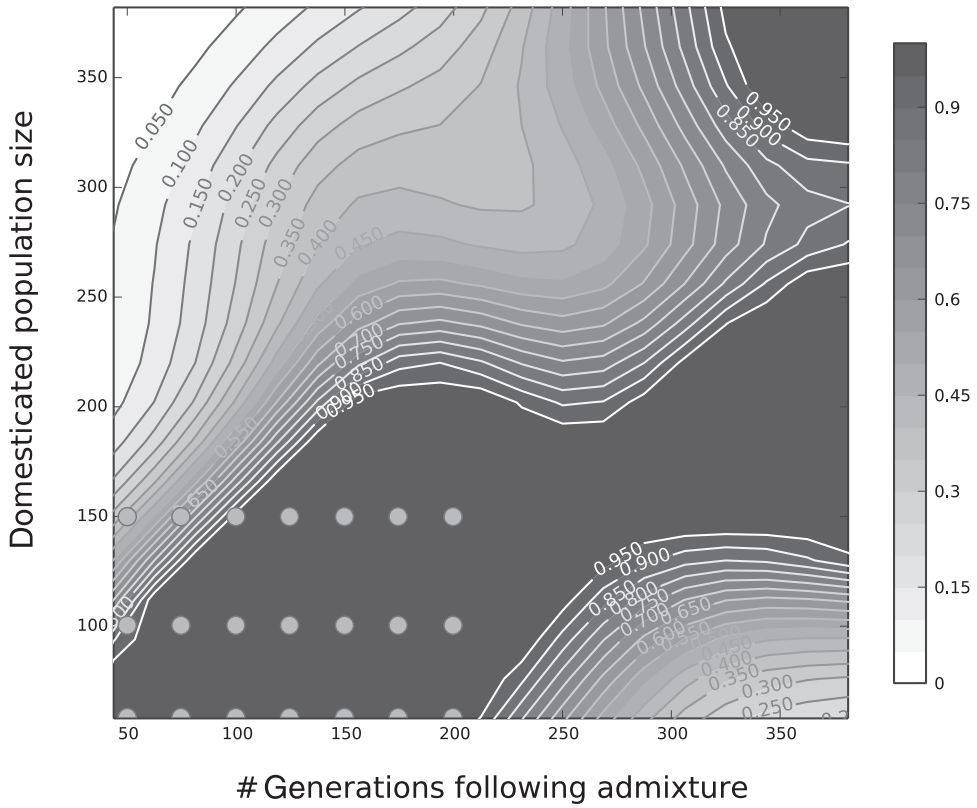

FIGURE 4. Likelihood that a hybrid population appears monophyletic: domesticated population size and post-admixture time. Likelihood function showing the probability of inferring a monophyletic origin estimated from the seven-parameter KDE by setting the remaining five parameters to the values used by Allaby et al. (2008a), as described in the text. Values used by Allaby and colleagues for domesticated population size and number of generations following admixture are indicated by small gray circles.

where the simulation experiment iterates through all combinations of a fixed set of parameter values.

The advantages (and disadvantages) of using a sampling-based strategy to select parameter values are similar to the advantages (and disadvantages) of using Monte Carlo methods for numerical integration. Our approach is less vulnerable to the "curse" of dimensionality and, importantly, does not rely on *a priori* selection of fixed grid values, values which could completely miss the regions of interest.

There are two main stages in the approach we describe. First, we use an MCMC algorithm to sample from a "posterior" distribution of the parameters conditional on a certain simulation outcome. For example, when assessing the robustness of the crop domestication model of Allaby et al. (2008a), we conditioned on the outcome that the domesticated individuals were monophyletic, and then sampled from the distribution of seven parameters describing population sizes, recombination rate, and the duration of protracted events in the history of the population. The resulting density indicates for which areas of the parameter space the domesticated population was most likely to (erroneously) appear monophyletic.

The second stage takes the output of the MCMC and uses importance sampling to estimate summary values as well the "likelihood" surface giving the probability of different outcomes for each parameter value. For example, in the domesticated crop example, the likelihood surface describes the probability that the polyphyletic domesticated population appeared monophyletic, whereas the output of the MCMC only indicates *where* the admixed population appeared monophyletic.

We present the results from two case studies: a comparison of phylogenetic methods and a test of robustness for a simulation experiment in genetic archaeology. In both cases, our approach explored the parameter space without having to specify a fixed set of values. We obtained detailed estimates of the effect that different parameter combinations had on the probability of the outcomes, estimates which could be used for further studies or to test hypotheses.

The MCMC framework which we have introduced is straightforward to implement. The basic sampler requires no more code than a grid-based strategy, and the estimation of the likelihood function uses standard tools. The framework is completely general and, while the applications explored here are all biological in nature, there is nothing to prevent the framework being applied in any context where simulation experiments are carried out.

While we have demonstrated that the framework we introduce is both feasible and practical, it is clear

that considerable advances can be made to improve the computational and statistical efficiency. The MCMC without likelihoods algorithm of Marjoram et al. (2003) which we use here is also used in ABC. There has been a great deal of work improving the efficiency of ABC sampling algorithms (Beaumont et al. 2009; Blum and François 2010; Del Moral et al. 2011), and many of these ideas could be applied here. Our application of density estimation could likewise be improved and refined. Indeed, it is conceivable that in many contexts we could construct an efficient importance distribution directly, without having to resort to the relatively efficient MCMC algorithms.

Supplementary Material

Data available from the Dryad Digital Repository: http://dx.doi.org/10.5061/dryad.366j4.

Funding

This work was supported by a postdoctoral fellowship from the Natural Science and Engineering Research Council of Canada (grant number PDF-373001-2009) the Allan Wilson Centre; and the Department of Mathematics and Statistics, University of Otago.

Acknowlegments

We thank Cécile Ané, Andy Anderson, Tilman Davies, and Peter Green for valuable comments and discussion.

References

Allaby R., Fuller D., Brown T. 2008a. Reply to Ross-Ibarra and Gaut: multiple domestications do appear monophyletic if an appropriate model is used. Proc. Natl. Acad. Sci. USA 105:E106.

Allaby R., Fuller D., Brown T. 2008b. The genetic expectations of a protracted model for the origins of domesticated crops. Proc. Natl. Acad. Sci. USA 105:13982–13986.

Anisimova M., Gil M., Dufayard J.-F., Dessimoz C., Gascuel O. 2011. Survey of branch support methods demonstrates accuracy, power, and robustness of fast likelihood-based approximation schemes. Syst. Biol. 60:685–699.

Beaumont M., Cornuet J., Marin J., Robert C. 2009. Adaptive approximate Bayesian computation. Biometrika 86:983–990.

Blum M., François O. 2010. Non-linear regression models for approximate Bayesian computation. Stat. Comput. 20:63–73.

Brooks S., Gelman A., Jones G., Meng X., editors. 2011. Handbook of Markov chain Monte Carlo. Boca Raton, FL: Chapman & Hall/CRC Handbooks of Modern Statistical Methods CRC Press.

Brown T., Jones M., Powell W., Allaby R. 2009. The complex origins of domesticated crops in the Fertile Crescent. Trends Ecol. Evol. 24: 103–109.

Clarke G., Kempson R. 1997. Introduction to the design and analysis of experiments. London: Arnold.

Csillery K., Blum M.G., Gaggiotti O.E., Francois O. 2010. Approximate Bayesian computation (ABC) in practice. Trends Ecol. Evol. 25: 410–418.

Del Moral P., Doucet A., Jasra A. 2011. An adaptive sequential Monte Carlo method for approximate Bayesian computation. Stat. Comput. 22:1–12.

Dice L. 1945. Measures of the amount of ecologic association between species. Ecology 26:297–302.

Didelot X., Everitt R., Johansen A., Lawson D. 2011. Likelihood-free estimation of model evidence. Bayesian Anal. 6:49–76.

Diggle P. 1985. A kernel method for smoothing point process data. Appl. Statist. 34(2):138–147.

FitzJohn R.G. 2010. Quantitative traits and diversification. Syst. Biol. 59:619–633.

Foster P. 2004. Modeling compositional heterogeneity. Syst. Biol. 53:485–495.

Gelman A., Meng X. 1998. Simulating normalizing constants: from importance sampling to bridge sampling to path sampling. Statist. Sci. 13:163–185.

Gross B., Olsen K. 2010. Genetic perspectives on crop domestication. Trends Plant Sci. 15:529–537.

Grummer J.A., Bryson R.W., Reeder T.W. 2014. Species delimitation using Bayes factors: simulations and application to the sceloporus scalaris species group (squamata: Phrynosomatidae). Syst. Biol. 63(2):119–133.

Guindon S., Dufayard J.-F., Lefort V., Anisimova M., Hordijk W., Gascuel O. 2010. New algorithms and methods to estimate maximum-likelihood phylogenies: assessing the performance of phyml 3.0. Syst. Biol. 59:307–321.

Huang H., He Q., Kubatko L.S., Knowles L.L. 2010. Sources of error inherent in species-tree estimation: impact of mutational and coalescent effects on accuracy and implications for choosing among different methods. Syst. Biol. 59:573–583.

Huelsenbeck J. 1995. Performance of phylogenetic methods in simulation. Syst. Biol. 44:17.

Kuhner M.K., Felsenstein J. 1994. A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates. Mol. Biol. Evol. 11:459–68.

Leaché A.D., Harris R.B., Rannala B., Yang Z. 2014. The influence of gene flow on species tree estimation: a simulation study. Syst. Biol. 63:17–30.

Leaché A.D., Rannala B. 2011. The accuracy of species tree estimation under simulation: a comparison of methods. Syst. Biol. 60:126–137.

Liu J. 2008. Monte Carlo strategies in scientific computing. Springer Series in Statistics. New York: Springer.

Marjoram P., Molitor J., Plagnol V., Tavare S. 2003. Markov chain Monte Carlo without likelihoods. Proc. Natl. Acad. Sci. USA 100: 15324–15328.

Molina J., Sikora M., Garud N., Flowers J., Rubinstein S., Reynolds A., Huang P., Jackson S., Schaal B., Bustamante C., Boyko A., Purugganan M. 2011. Molecular evidence for a single evolutionary origin of domesticated rice. Proc. Natl. Acad. Sci. USA 108: 8351–8356.

Pigot A.L., Phillimore A.B., Owens I.P., Orme C.D.L. 2010. The shape and temporal dynamics of phylogenetic trees arising from geographic speciation. Syst. Biol. 59:660–673.

Poole D., Raftery A. 2000. Inference for deterministic simulation models: the Bayesian melding approach. JASA 95:1244–1255.

Raftery A., Givens G., Zeh J. 1995. Inference from a deterministic population dynamics model for bowhead whales. JASA 90:402–416.

Robinson D., Foulds L. 1981. Comparison of phylogenetic trees. Math. Biosci. 53:131–147.

Ross-Ibarra J., Gaut B. 2008. Multiple domestications do not appear monophyletic. Proc. Natl. Acad. Sci. USA 105:E105.

Saitou N., Nei M. 1987. The neighbor-joining method: a new method for reconstructing phylogenetic trees. Mol. Biol. Evol. 4:406–425.

Sevcíková H., Raftery A., Waddell P. 2007. Assessing uncertainty in urban simulations using Bayesian melding. Transport Res. B–Meth. 41:652–669.

Sevcíková H., Raftery A., Waddell P. 2011. Uncertain benefits: application of Bayesian melding to the Alaskan Way Viaduct in Seattle. Transport Res. A 45:540–553.

Sokal R., Sneath P. 1963. Principles of numerical taxonomy. Series of books in biology. San Francisco: W.H. Freeman.

Wiens J.J., Morrill M.C. 2011. Missing data in phylogenetic analysis: reconciling results from simulations and empirical data. Syst. Biol. 60:719–731.

Zhang C., Zhang D.-X., Zhu T., Yang Z. 2011. Evaluation of a Bayesian coalescent method of species delimitation. Syst. Biol. 60:747–761.

Zhang P. 1996. Nonparametric importance sampling. JASA 91: 1245–1253.