Systematic Biology (2020), **0**, 0, *pp.* 1–38
doi:10.1093/sysbio/SnapperFinal

Bayesian Inference of Species Trees using Diffusion Models

Marnus Stoltz[1], Boris Baeumer[1], Remco Bouckaert[2], Colin Fox[3], Gordon Hiscott[1], and David Bryant[1,*]

[1] *Department of Mathematics and Statistics, University of Otago, Dunedin 9054, New Zealand*
[2] *Centre for Computational Evolution, University of Auckland, Auckland 1142, New Zealand*
[3] *Department of Physics, University of Otago, Dunedin 9054, New Zealand*

**Department of Mathematics and Statistics, University of Otago, Dunedin 9054, New Zealand, phone*

+64 3 479 7889. david.bryant@otago.ac.nz

Abstract

We describe a new and computationally efficient Bayesian methodology for inferring species trees and demographics from unlinked binary markers. Likelihood calculations are carried out using diffusion models of allele frequency dynamics combined with novel numerical algorithms. The diffusion approach allows for analysis of datasets containing hundreds or thousands of individuals. The method, which we call Snapper, has been implemented as part of the BEAST2 package. We conducted simulation experiments to assess numerical error, computational requirements and accuracy recovering known model parameters. A re-analysis of soybean SNP data demonstrates that the models implemented in Snapp and Snapper can be difficult to distinguish in practice, a characteristic which we tested with further simulations. We demonstrate the scale of analysis possible using a SNP dataset sampled from 399 fresh water turtles in 41 populations.

Key words: Bayesian inference; species trees; diffusion models; multi-species coalescent; spectral methods; SNP data.

Recent years have witnessed a proliferation in the number of methods for inferring species trees from whole genomes (Bryant and Hahn, 2020). Some have gone so far as to herald a paradigm shift in phylogenetics (Edwards, 2009). It is now widely accepted that (i) phylogenetic analysis needs to take account of the varying evolutionary histories of different parts of the genome, and (ii) estimation of evolutionary relationships between populations (or species) should take account of evolutionary dynamics *within* populations (or species).

In the systematics community, taking account of population dynamics in a phylogenetic context has meant implementing some version of the *multispecies coalescent* (e.g. Liu et al., 2008; Degnan and Rosenberg, 2009; Liu et al., 2010; Song et al., 2012; Bryant et al., 2012; Rannala and Yang, 2017). The multispecies coalescent is a model of the distribution of gene trees for samples from multiple species or populations. One of the key advantages of the multispecies coalescent is that, for neutral mutations, the gene tree can be decoupled from the mutation process, a feature which forms the basis of many implementations. Nevertheless, coalescent-based methods have their limitations. It is difficult to incorporate selection and the running time increases rapidly with the number of individuals being sampled and the number of independent loci.

Bryant et al. (2012) showed that explicit sampling of gene trees could be avoided in the case of unlinked binary markers. This is appropriate for the estimation of species trees from unlinked single nucleotide polymorphisms (SNPs). Their algorithm was implemented in Snapp and can analyse data sets with hundreds of thousands of loci and up to 200 individuals (depending on computing resources and sampling difficulties). Unfortunately the likelihood algorithm in Snapp does not scale that well as the number of individuals increases, limiting the method in practice.

Rather than tinker with the Snapp algorithm, we have taken a completely new approach with a different kind of model. Like Gutenkunst et al. (2010), Sirén et al. (2010) and Lukic and Hey (2012) we use diffusion models, though we apply them in a new way.

The end result is that we can carry out SNAPP-type analyses but with essentially no limits on the number of individuals being sampled.

Diffusion models, like the coalescent, are a convenient approximation of the standard Wright-Fisher discrete models. In fact, diffusion models pre-date coalescent models by a good fifty years (Wright, 1931; Kingman, 1982). Whereas coalescent models describe the distribution of the genealogy or gene tree for a sample of individuals, diffusion models describe the frequency of an allele in the population as a whole. There are straightforward extensions incorporating selection.

There are three challenges to overcome working with a model of allele frequencies. First, we do not actually observe the population frequencies, we just observe a sample drawn from that population. This problem is easily solved by incorporating the sampling step explicitly into our likelihood.

Second, gene frequencies are continuous traits, so it is not possible to sum over ancestral trait values as in Felsenstein's pruning algorithm (Felsenstein, 1981). For this, we follow a numerical approach developed in Hiscott et al. (2016), though with some new twists to improve efficiency and accuracy.

Third, diffusion models do not give explicit transition probabilities or densities: these are only available via partial differential equations (PDEs). We solve these differential equations numerically, extending a standard spectral approach from a single population to the entire species tree. We note that there is significant potential for adapting our methods to other diffusion-based models.

Models and Methods

Our starting point is the Wright-Fisher discrete model of drift and mutation. Suppose we have a population of N diploid individuals, giving $2N$ copies of each autosomal gene. We consider binary markers with two alleles (say 'red' and 'green') and a model with non-overlapping generations. Let the sequence $X_1, X_2, X_3, \ldots$ denote the

number of red alleles in each generation. Under random mating, X_{n+1} has a binomial distribution with parameters $2N$ and $(1-u)\frac{X_n}{2N} + v\left(1-\frac{X_n}{2N}\right)$, where u is the probability of mutating from red to green and v the probability of mutating from green to red. In this way the numbers $X_1, X_2, X_3, \ldots$ of red alleles will follow a random walk with discrete time steps (generations) and states (from 0 to $2N$).

The idea behind diffusion models is to approximate this discrete random walk by a continuous random walk that is easier to work with analytically. We set up the approximation so that the larger N gets, the better the approximation fits. We describe the *proportion* of gene copies having the red allele, rather than the total count. The state space of the process will therefore be the interval $[0, 1]$. Instead of considering discrete generations, we construct a random walk which is continuous in time.

Diffusion models for allele frequencies involve a rescaling of time. There are simple, practical, reasons for this. As the population size N gets larger, the rate of genetic drift decreases, ultimately approaching zero drift in the limit. However the effect of drift is something that we would like to model. For this reason we change the units of time so that the rate of drift remains approximately the same as N increases, eventually converging to some non-zero amount. The convention for diploid populations is to use a scale where one unit of time corresponds to $2N$ generations (one coalescent unit).

If we are to change the units of time, we need to adjust the rate of mutation accordingly. With these time units, the rate of change due to mutations from green alleles to red alleles is $2Nu$. We adopt the standard notation and define $\beta_1 = 2Nu$ and $\beta_2 = 2Nv$.

For each value of t we let $f(x, t)$ denote the density of the allele proportion at time t, noting that this allele proportion is a continuous random variable with state space $[0, 1]$. Surprisingly, there is very little choice over what the function $f(x, t)$ can be, after a few basic assumptions are made. See Etheridge (2011) for technical details and McKane and Waxman (2007) for a discussion about how we need to augment f with point masses at 0 and 1.

As is often the case in mathematical modelling, we work with the function f indirectly using a PDE. Stochastic process theory (Øksendal, 2003, chap. 5) tells us that the function f satisfies the PDE

$$\frac{\partial f(x,t)}{\partial t} = -\frac{\partial}{\partial x}\left(\beta_1(1-x) - \beta_2 x\right) f(x,t) + \frac{1}{2}\frac{\partial^2}{\partial x^2} x(1-x) f(x,t). \tag{0.1}$$

This equation by itself is not enough to uniquely determine f. We also need to specify what f looks like at the boundaries. To specify that the distribution of the initial state is given by some density π we add the initial condition

$$f(x,0) = \pi(x) \text{ for } x \in [0,1]. \tag{0.2}$$

Note that to specify that the initial value equals a specific value, say x_0, we need π to be a Dirac-delta function $\pi(x) = \delta(x - x_0)$ which is essentially an infinitely thin spike on one value x_0.

Even with initial conditions, the PDE (0.1) does not uniquely determine the function f. We also need to add boundary conditions at $x = 0$ and $x = 1$ to guarantee that the probability of going outside the interval $[0, 1]$ is always zero. Writing this conservation of probability as a condition on integrals of f and then plugging in the PDE eventually leads to what in physics is known as a *zero-flux condition*

$$-(\beta_1(1-x) - \beta_2 x) f(t,x) + \frac{1}{2}\frac{\partial}{\partial x} x(1-x) f(t,x) = 0, \tag{0.3}$$

when $x = 0$ or $x = 1$. See McKane and Waxman (2007) for details.

The combination of PDE (0.1) and the boundary conditions (0.2) and (0.3) are just a mathematically convenient, if not particularly transparent, way to describe the function f. The function f, in turn, is just a way to approximate the probabilities in the original discrete process. Ethier and Norman (1977) proved that the error with the diffusion approximation decreases like $O(u + v + 1/N)$. If u or v are large (compared to $\frac{1}{N}$) then the diffusion approximation will fail miserably. In a recent study, Tataru et al. (2017) used simulations to quantify the error from the diffusion approximation in small populations,

and found that diffusion models gave reasonable approximations even when the population has fewer than 100 individuals.

Modelling allele frequencies on a species tree

The diffusion model describes how allele frequencies change over time in a single population. The model extends directly to multiple populations in a species tree (Sirén et al., 2010). As in Bryant et al. (2012) we think of the root of the species tree as at the top of the tree with leaves at the bottom. The model describes evolution of the allele frequencies from the top of the tree to the bottom.

The allele frequency at the root has a distribution given by the stationary density of the diffusion model (in this case, a beta distribution). The allele frequency at the bottom of a branch has a distribution given by the diffusion model with an initial value equal to the allele frequency at the top of the branch. At a speciation, the two daughter populations have the same allele frequency as the parent population.

We now formalise these ideas. Suppose that the branches in the species tree are numbered $i = 1, 2, \ldots, 2s-2$ where, for convenience, the branches adjacent to the leaves are numbered $1, 2, \ldots, s$. Let X_i^T denote the allele frequency immediately below the top of branch i. Let X_i^B denote the allele frequency immediately above the bottom of branch i. Let $X_\rho = X_\rho^B$ denote the allele frequency at the root.

At the root the proportion of red alleles in the ancestral population has a beta distribution, which is the stationary distribution of the diffusion model (Ewens, 2004). The density of X_ρ is

$$\pi_\rho(x|\beta_1, \beta_2) = \frac{\Gamma\{2\beta_1 + 2\beta_2\}}{\Gamma\{2\beta_1\}\Gamma\{2\beta_2\}} x^{2\beta_1 - 1}(1-x)^{2\beta_2 - 1}, \qquad 0 < x < 1. \tag{0.4}$$

Along any branch in the species tree, the changes in allele frequencies are modelled using the diffusion process. The allele frequency at the start of the branch gives the initial density, $f(x, 0)$, for $x \in [0, 1]$. The distribution of the allele frequency y at the end of the branch is then given by $f(y, t)$, where t is the length of the branch in units of $2N$

generations and $y \in [0, 1]$.

At a speciation, we make the assumption that there is no correlation between allele type and speciation. The two descendant species are assumed to have identical allele frequencies to the parent node.

We have, therefore, a model for allele frequencies over the whole tree. However allele frequencies at the leaves are not directly observed. Instead we have a sample of n_i individuals, giving $2n_i$ allele copies sampled at random from each species i. If x_i is the red allele frequency for the population then the observed number r_i of red alleles in the sample for this marker has binomial distribution with parameters $2n_i$ and x_i, so that

$$P[R_i = r|x_i] = \binom{2n_i}{r} x_i^r (1 - x_i)^{2n_i - r},$$

see Sirén et al. (2010).

Algorithm AlleleSim in Figure 1simulates allele counts under this model. We used the algorithm of Jenkins et al. (2017) to simulate diffusions along each branch, thereby avoiding numerical issues at the boundaries which cause problems in alternative methods (Williamson et al., 2005; Bollback et al., 2008; Gutenkunst et al., 2009; Song and Steinrücken, 2012; Steinrücken et al., 2014).

Analytical formula for partial likelihoods

We now describe how to express the likelihood functions analytically. For each node i in the species tree and each ancestral state x, which in our case is an allele frequency, we define two partial likelihoods, $\ell_i^B(x)$ and $\ell_i^T(x)$. The first function gives the probability of observing all allele counts for taxa in the subtree rooted at i, conditional on the state $X_i^B = x$. The second function is defined in the same way, but is instead conditioned on the state X_i^T at the top of the branch.

Partial likelihoods at the leaves — Suppose that node i is a leaf and that we have sampled n_i diploid individuals from this population. Let r_i denote the number of observed allele copies carrying the red allele for the site. If x is the proportion of red alleles in the

population then r_i has a binomial distribution with parameters $2n_i$ and x. Hence the partial likelihood at a leaf i is given by

$$\ell_i^B(x) = \binom{2n_i}{r_i} x^{r_i}(1-x)^{2n_i - r_i}. \tag{0.5}$$

See Figure 2for a depiction of partial likelihoods at the leaves.

Partial likelihoods at a speciation— Let j and k be the children of node i. We assume that the allele frequencies for daughter populations after a speciation are exactly those of the parent population before speciation. The partial likelihoods at the bottom of the branch above i is then the product of partial likelihoods at the tops of the branches immediately below i, so

$$\ell_i^B(x) = \ell_j^T(x)\ell_k^T(x) \tag{0.6}$$

for all $x \in [0,1]$. See Figure 2for depiction of partial likelihoods at a speciation.

Partial likelihoods along a branch Let N_i be the effective population size for the branch directly above node i, and let τ_i denote the length of the branch above node i, measured in numbers of generations. Then $t_i = \frac{\tau_i}{2N_i}$ equals the length of the branch in coalescent units. Suppose that we already know the partial likelihood values at the bottom of the branch. That is, for each value of $y \in [0,1]$ the value $\ell_i^B(y)$ equals the probability of observing everything below branch i, given that the allele frequency at the bottom of branch i equals y. We would like to determine the corresponding partial likelihood at the top of the branch as a function of the ancestral frequency x.

For each $x \in [0,1]$ and $t \in [0, t_i]$ we let $g(x,t)$ denote the probability of observing all allele counts at or below node i, given that the ancestral frequency at time t above the node i is t. Then for all x we have

$$g(x,0) = \ell_i^B(x), \text{ for } x \in [0,1]. \tag{0.7}$$

and

$$\ell_i^T(x) = g(x,t_i), \text{ for } x \in [0,1]. \tag{0.8}$$

The function $g(x,t)$ satisfies a PDE which can be obtained from (0.1) and (0.3) using integration by parts multiple times (see Øksendal, 2003)

$$\frac{\partial g(x,t)}{\partial t} = (\beta_1(1-x) - \beta_2 x)\frac{\partial}{\partial x}g(x,t) + \frac{1}{2}x(1-x)\frac{\partial^2}{\partial x^2}g(x,t). \tag{0.9}$$

This is known as the *backwards equation* whereas the original PDE (0.1) is called the *forwards equation*. We note that this PDE together with the initial conditions (0.7) are enough to determine the function g and hence the partial likelihood function ℓ_i^T at the top of the branch (Epstein and Mazzeo, 2010). We will solve this PDE numerically, as outlined below. Solutions to this backward diffusion are smooth (infinitely differentiable) functions (Epstein and Mazzeo, 2010, Thm 9.2 pg.595), whenever the initial conditions are smooth. From (0.5) we see that the partial likelihoods smooth, and since in (0.6) the product of smooth functions is smooth we have that partial likelihoods throughout the entire tree are smooth. We can therefore avoid many of the numerical headaches (such as solutions that tend toward infinite spikes) encountered by those using the forward equation directly (Lukic and Hey, 2012).

Likelihood of the tree Equations (0.5), (0.6) and (0.9) can be applied to the entire tree, starting with the leaves and moving upwards towards the root. They define the partial likelihood at the root $\ell_\rho^B(x)$, which is the probability of observing all allele counts for all populations, conditioning on the allele frequency $X_\rho = X_\rho^B$ being equal to x.

To compute the probability of a site k we integrate the product of the partial likelihood at the root and with the stationary density of the diffusion model (as given in (0.4)):

$$L_k = \int_0^1 \pi(x|\beta_1,\beta_2)\ell_\rho^B(x)dx. \tag{0.10}$$

The log-likelihood of the tree is then

$$\log L = \sum_k \log(L_k). \tag{0.11}$$

When working with SNP data we replace the likelihood for a site with the likelihood

conditional on the site being non-constant, given by $L_k/(1 - L_0)$, where L_0 is the probability of observing a constant site (see Felsenstein 1992; Bryant et al. 2012).

The algorithm LOG-LIKELIHOOD in Figure 3provides a high-level overview of likelihood computation. Details of the numerics are found in the Appendix 1.

Translation of parameters

One of the more confusing aspects of working across population genetics and phylogenetics is the different ways that different communities parameterize different variables. In this section we summarise the parameters and outputs for our model, and show how they might be converted into other formulations.

There are four groups of input variables that could conceivably be inferred from data using this model. They are

1. The species tree.
2. The branch lengths in the species tree. In our model, we measure the length of a branch in *number of generations*. Let τ_i denote the length of the branch i, which is the branch immediately above node i.
3. The mutation rates u and v giving the probabilities of mutating from a red to a green allele or a green to a red allele, *each generation*.
4. The effective population size N_i for each branch i, the branch above node i.

Different methods for inferring species trees describe these parameters in different ways. One convention in phylogenetics is to express branch lengths in terms of *expected mutations per site*. Under our model, the rate of mutations per site, per generation, is

$$\begin{aligned}\mu &= P(\text{ lineage has red allele })u + P(\text{ lineage has green allele })v \\ &= \frac{v}{u+v}u + \frac{u}{u+v}v, \end{aligned} \tag{0.12}$$

so a branch of length τ_i generations corresponds to a branch length of $\mu\tau_i$ expected mutations per site.

Methods which ignore the branch lengths in gene trees are not able to infer both population sizes and branch lengths, and instead use a single parameter per branch, typically measured in coalescent units (e.g. Vachaspati and Warnow, 2015; Liu, Yu, and Edwards, 2010). If τ_i is the length of a branch in numbers of generations, the length of the branch in coalescent units is given by $t_i = \frac{\tau_i}{2N_i}$.

In our model, we parameterize effective population size for branch i directly as N_i. SNAPP (Bryant et al., 2012), Best (Liu, 2008), and BPP (Yang, 2015) instead use $\theta_i = 4N_i\mu$ as the parameter for effective population size. Under an infinite sites model, $4N_i\mu$ equals the expected proportion of sequence differences between two individuals from the same population. In a finite sites model, $4N_i\mu$ is an overestimate of this expectation, due to backward mutations.

An important issue with the diffusion model as we have described it, and indeed with many multispecies coalescent models, is that there is an identifiability problem with rates. If the mutation rates are multiplied by some constant c and at the same time branch lengths and population sizes are multiplied by $\frac{1}{c}$, the probability of the data remains the same.

One solution is to estimate the mutation rate μ beforehand and use this value, or a prior distribution around that value, to include μ as part of our model. The prior distribution for u and v is then reformulated so that (0.12) is satisfied. This strategy is used by StarBeast (Heled and Drummond, 2009), where the average substitution (mutation) rate r_μ is fixed ahead of time.

An alternative strategy is to express branch lengths in terms of expected substitutions (mutations) per site and population sizes in terms of the θ parameter, as both of these are invariant to the choice of μ (Yang, 2015; Bryant et al., 2012). As a consequence, the effective population sizes N_i cannot be inferred without additional

information. This approach adapts well to phylogenetics where it is customary to describe mutation rates using a normalised rate matrix, such as the Jukes-Cantor and HKY85 matrices

$$\mathbf{Q} = \begin{bmatrix} - & 1/3 & 1/3 & 1/3 \\ 1/3 & - & 1/3 & 1/3 \\ 1/3 & 1/3 & - & 1/3 \\ 1/3 & 1/3 & 1/3 & - \end{bmatrix} \text{ and } \mathbf{Q} = \frac{1}{r} \begin{bmatrix} - & \pi_C & \kappa\pi_G & \pi_T \\ \pi_A & - & \pi_G & \kappa\pi_T \\ \kappa\pi_A & \pi_C & - & \pi_T \\ \pi_A & \kappa\pi_C & \pi_G & - \end{bmatrix}.$$

In this matrix, the stationary probabilities for the nucleotides are $\pi_A, \pi_C, \pi_G, \pi_T$, the parameter κ controls the ratio of transitions to transversions, and the diagonal elements are chosen to make rows sum to zero. The Jukes-Cantor matrix on the left is already normalised, while in the HKY85 model r would be chosen to make the overall substitution rate equal to 1.

Rate matrices can be incorporated directly into SNAPPER, either using prior information for μ, or when using the same branch length and population parameter scheme as BPP. For any site, if X and Y are the two nucleotides present and we (arbitrarily) assign red to X and green to Y then the appropriate choices for u and v are simply $\mathbf{Q}_{XY}$ and $\mathbf{Q}_{YX}$. For example, under the HKY85 model, if the red allele is nucleotide A and green allele is nucleotide C then the rates can be approximated by $u = \mu\mathbf{Q}_{AC}$ and $v = \mu\mathbf{Q}_{CA}$.

Comparison with allele frequency spectrum methods

We note that diffusion models are already used in population genetics to approximate changes in the *allele frequency spectrum* (AFS) (Gutenkunst et al., 2009; Lukic and Hey, 2012; Racimo et al., 2016). For each value of x between 0 and 1, the AFS gives the proportion of sites for which the derived allele appears in the proportion x of the sample. For example, we might count up all sites where the derived allele appears in 30 of the 100 sampled genomes. If 5% of all sites fall into this category then the observed AFS value for 0.3 will be 5%.

Methods based on the AFS to infer genetic parameters for a single population are widespread in the literature (Boitard et al., 2016; Lapierre et al., 2017; Nielsen, 2000). In

all of these analyses, it is assumed that the population dynamics have achieved stationarity when the AFS predicted by the model is used.

The *joint-AFS* is the extension of the AFS to multiple populations. Suppose that there are s populations. For every vector $(x_1, x_2, \ldots, x_s)$ of numbers between 0 and 1, the joint-AFS gives the proportion of sites where the derived allele appears in a proportion x_1 of the individuals in the first population, x_2 of the individuals in the second population, and so on. While the AFS for each population by itself will be the stationary AFS, the joint-AFS for multiple populations depend on the time since separation. Fortunately the PDE determining the predicted joint-AFS is a straight-forward extension of the PDE for a single population (Gutenkunst et al., 2009; Lukic and Hey, 2012; Racimo et al., 2016).

There are two main advantages of an approach based on the joint-AFS when compared to the method we describe here. First, the PDE for the joint-AFS only needs to be solved once for all sites, whereas in our approach we end up solving the PDEs once for each (distinct) site. Second, migration between populations can be incorporated quite simply into the expected AFS calculation, whereas it would break down the dynamical programming strategy that we use.

The main disadvantage of the joint-AFS approach is that the PDE has as many dimensions as the number of species and so suffers from the *curse of dimensionality* (Gutenkunst et al., 2009) and an exponential growth in the size of grid used by standard numerical methods. This factor severely limits the number of species which can be considered concurrently, whereas in our approach the running time scales linearly with the number of species.

Numerical calculation

So far, we have only discussed how the model and likelihood are defined, not how they are computed. Making the algorithm efficient for large-scale analysis required development of a large suite of numerical algorithms, many of which are novel. Details of

these methods can be found in Appendix 1, with some of the key ideas outlined here.

The partial likelihood functions ℓ_i^B and ℓ_i^T are central to our method. These are continuous functions on $[0, 1]$, so we cannot computer or store them exactly on a computer. Instead we approximate them using a collection of judiciously chosen *basis functions*, in our case the shifted Chebyshev functions (details in Appendix 1). Each partial likelihood function is approximated using a weighted sum of the basis functions, the weights (or coefficients) being determined by the algorithm as we move up the tree. As the number of basis functions increases, the accuracy improves, though so does the computation time. The approach using basis functions differs from that of Hiscott et al. (2016), where partial likelihood functions were encoded using the values of the functions at a fixed mesh of points. The new approach is more flexible and has greatly improved accuracy, as we demonstrate below. In fact it is possible to bound the error introduced in approximation and show that this error decreases extremely rapidly as the number of basis functions increases.

To compute the approximations of the likelihood and partial likelihood functions we substitute the approximation formulas into equations (0.5), (0.6) and (0.9), as detailed in Appendix 1. A lot more algorithmic work took place to obtain sufficient accuracy and speed, including

- Special recurrence formulas for stable evaluation of partial likelihoods at the tips (0.5).
- An efficient algorithm for computing the product of two approximations, based on fast Fourier transforms.
- Efficient algorithms which take advantage of structure in the PDE in equation (0.9) to find rapid and accurate numerical solutions.
- Dynamic caching algorithms to share partial likelihood calculations between sites.

See Appendix 1 for details.

With all of the efficiency gains, the numerical algorithm takes only $O(sKlog(K))$ time per site where s is the number of species, K the number of basis functions. We assume that the data is pre-processed so that for each site we have the frequencies of individuals with each type of allele. This pre-processing takes linear time in the size of the data set, and only needs to be carried out once. In comparison, SNAPP takes $O(sn^2 \log n)$ time per site with same amount of pre-processing. The algorithms scale quite differently; the running time of the new algorithm depends on the number of basis functions K but not on the number of individuals.

*S*NAPPER

The likelihood algorithm forms the core of a Bayesian inference software package, SNAPPER. The software is open-source and available to download at `https://github.com/rbouckaert/snapper`. Like SNAPP, it takes biallelic data at multiple loci for multiple individuals in a set of species and returns samples from the joint posterior distribution of (i) species phylogenies, (ii) species divergence times and (iii) effective population sizes. As in SNAPP we implemented multithreading to take advantage of parallel computation on multiple core machines or graphics processing units. The range of prior distributions, and flexible prior specification, remains essentially unchanged.

Like *S*NAPP we use Markov chain Monte Carlo (MCMC) to sample from the posterior distribution of species trees and parameters. The proposals we implemented are essentially the same as those used in SNAPP. We added one new proposal that selects a subtree of the species tree and scales all population sizes within that subtree simultaneously, a refinement of an existing proposal. We also added a new prior based on the CIR process rate variation model (Lepage et al., 2006) that is designed to model correlation in effective population sizes across the species tree, details below. The package includes simulators and python scripts for integrating SNAPPER with the iPyRad data pipeline to assist with streamlining data analysis.

Simulation protocol

We conducted a number of simulation experiments to assess the performance of the algorithms and their equivalence with approaches based on the coalescent.

The first experiment assesses the impact of numerical error. The accuracy and running time of SNAPPER both depend heavily on the number of Chebyshev basis functions (K) used in approximations. An important question is how many basis functions are required and the rate at which the numerical approximation converges to the true values.

We selected ten 4-taxa species trees with a caterpillar topology and ten 4-taxa species tree with a balanced topology and branch lengths. Half of the trees in each group were scaled to have very short branches (average of 0.005 coalescent units). The other half of each group was left with very long branches (average of 0.5 coalescent units). We followed the same procedure for ten 16-taxa species trees with a caterpillar topology and ten 16-taxa species tree with a balanced topology. Population size (θ) parameters for each tree were generated from a Gamma distribution with mean 0.1 and variance 0.0001. We assumed equal forward and backwards mutation rates throughout. For each tree we simulated data under the diffusion model for a single site drawing the total number of individuals at each tip from a uniform distribution between 5 and 30. We then computed the log-likelihood for different numbers of basis function, $K = 5, \ldots, 50$. Since there is no analytical expression for the likelihood we assess convergence by comparing values calculated to those with a large ($K = 200$) number of basis functions.

The second experiment examined the differences between the coalescent model and the diffusion model. For this experiment we simulated data according to the multi-species coalescent but analysed them using SNAPPER and a diffusion model. Theory suggests that the coalescent and diffusion models are approximations of each other (and of the underlying Wright-Fisher model).

We generated 300 species trees using the Yule distribution (Yule, 1925) with speciation rate of 10. For one third of those species trees we generated population size (θ)

parameters from a Gamma distribution with mean 0.01 and variance 0.0001; for another third we generated population size (θ) parameters from a Gamma distribution with mean 0.01 and variance 0.00001; for another third we generated population size (θ) parameters from a Gamma distribution with mean 0.01 and variance 0.000001. We then simulated 1000 SNPs for each tree (with 32 total individuals) under the coalescent model using the program simSnapp (freely available at `https://www.beast2.org/snapp/`). For each simulated SNP dataset we ran two MCMC chains for 100,000 iterations using Snapper. One of the chains was specified with 'correct' priors, i.e distributions used to simulate the SNPs. For the other chain we specified incorrect priors. The 'incorrect' prior for the tree was a Yule distribution (Yule, 1925) with speciation rate of 1. The 'incorrect' prior used for the population size (θ) parameters was a Gamma distribution with mean 0.1 and variance 0.0001.

In the third experiment we looked at how computational time for Snapp and Snapper scales in terms of number of individuals. We simulated data according to the multi-species coalescent and compared computational time of the log-likelihood between Snapp and Snapper. We generated six 4-taxa species trees from a Yule distribution with speciation rate 10. We started with 10 individuals (in total for all species) for the first tree and incrementally increased the number by 5 up to a total number of 35 individuals. For each tree we simulated 1000 SNPs under the coalescent model. We follow the same procedure for six 8-taxa species trees. We used $K = 33$ Chebyshev basis functions (K) for Snapper. Log-likelihood computation was done without caching.

All simulations and inference were run on a laptop with an Intel i7-8565U CPU.

Analysis of wild and cultivated soybeans

Theory predicts that the multispecies coalescent model underlying Snapp and the diffusion model behind Snapper are approximations of one another (Griffiths and Spanó, 2010), with differences decreasing as effective population size increases. To test this

hypothesis on natural data we reanalyzed the soybean dataset in Chifman and Kubatko (2014) using both approaches. The data consists of 1,027,026 SNPs from a total of 20 individuals in 10 populations. We assumed a prior Yule distribution (Yule, 1925) on the species tree with expected tree height of 0.5 coalescent units, with equal forward and backwards mutation rates throughout.

Chifman and Kubatko (2014) reported that SNAPP was not sampling well from the posterior distribution. Upon reanalysis it quickly became apparent the main cause of difficulty was the inference of population sizes on branches with close to zero length. One solution is to add correlation of branch lengths to the model. We did this through use of a prior based loosely on the CIR process.

The CIR process is widely used in finance to model interest rate fluctuations (Cox et al., 1985). Lepage et al. (2006) use the process to model fluctuations of evolutionary rate along a tree. The CIR process is one of the few mathematically tractable random processes which are continuous, positive, and have a stationary (long term) distribution. At any point in time, the process has a gamma distribution. If we condition on current value then the distribution at some time t in the future is non-central Chi-squared. The equations are quite messy - see Lepage et al. (2006) for further details.

We use the process to model the prior distribution of population parameters θ over the tree. The value of θ in the root population is assumed to have a gamma distribution (as in BPP (Yang, 2015)). We then assume that the θ values evolve down each branch following the CIR process. However rather than integrate out the variation along each branch we just sample the values at the nodes themselves.

The tree-based CIR model has three parameters. Two are the standard parameters α, β for the gamma distribution, and the third, κ controls the rate at which correlation between θ values disappears over time. The correlation between the values at either end of a branch of length t is given by $e^{-\kappa t}$. If κ is zero then all branches will have the same θ value; if κ is infinite then all branches will all have independent θ values. We used a

correlated CIR prior for population sizes with hyperparameters $\alpha = 2$, $\beta = 200$ and $\kappa = 10$, giving an expected θ value of 0.01.

To improve mixing with correlated θ values we also implemented an MCMC proposal which scales all population sizes within a randomly selected subtree by a random factor.

We ran ten MCMC chains with 1,000,000 iterations each for both SNAPP and SNAPPER. This analysis was carried out using 8 cores from the NESI high performance computing server (https://www.nesi.org.nz/).

Analysis of freshwater turtle Emydura macquarii

To illustrate the scale of data set that can be handled using SNAPPER we reanalysed unlinked SNP data of Georges et al. (2018) from a group of freshwater turtles known collectively as Emydura. The range of Emydura extends almost the full length of the Australian continent from north to south. The group is currently recognized as a complex of closely related and morphologically distinct allopatric forms, variously regarded as species, subspecies or distinct morphological lineages (Georges et al., 2018).

The dataset consists of SNP data from 399 individuals divided into 41 populations, mostly from the subspecies *E. macquarii*, sampled from 57 distinct water bodies. The geographic sampling covers the coastal drainages of eastern Australia, from the Hunter River in the south (New South Wales) to the Normanby River (Queensland) in the north; the rivers of the Murray-Darling Basin, including the Paroo drainage, and the Lake Eyre Basin, and the intervening Bulloo Basin (Georges et al., 2018). The analyzed dataset contains 5,186 unlinked SNPs after sites with missing data was removed.

We assumed a prior Yule distribution (Yule, 1925) on the species tree with expected tree height of 1 coalescent unit. We used a correlated CIR prior for population sizes with hyperparameters $\alpha = 2$, $\beta = 200$ and $r = 10$, which corresponds to expected θ of 0.01. We assumed equal forward and backwards mutation rates throughout.

We ran the SNAPPER sampler for 2,000,000 iterations with convergence assessed in Tracer (Rambaut et al., 2018). This analysis was carried out on a laptop with 2 cores and Intel Coffee Lake i3-7100 CPU.

RESULTS

Simulation results

We summarize the convergence results of the first simulation in Figure 4(4 species trees) and in Figure 5(16 species trees). We see that for all trees the error of the log-likelihood decreases exponentially with the number of Chebyshev basis functions (K), that is, it decreases like α^K for some $\alpha < 1$. Numerical approximation error is larger for very short branch lengths.There are good reasons for this. Firstly, when the partial likelihood function is spiked, the approximate solutions are high degree polynomials requiring more basis functions. Secondly, population sizes for short branches are intrinsically more difficult to estimate, no matter which model or method is used. The only information we have about population sizes comes from the distribution of coalescent events, and on short branches there are simply insufficient coalescent events to make sound inference. Later, we address this by adopting a prior distribution on population sizes which introduces correlation between neighbouring branches. Note that, apart from branch length distribution, tree shape does not seem to have a noticeable effect on the rate of error convergence.

Sim	Priors	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_ρ	height	length	top.
Γ(2,200)	C	0.91	0.91	0.94	0.93	0.94	0.97	0.95	1.00	1.00	1.00
Γ(2,200)	I	0.84	0.88	0.87	0.84	0.75	0.71	0.06	0.99	0.99	1.00
Γ(4,400)	C	0.93	0.97	0.92	0.97	0.9	0.98	0.95	1.00	1.00	1.00
Γ(4,400)	I	0.91	0.95	0.93	0.94	0.78	0.68	0.32	0.99	0.99	1.00
Γ(80,8000)	C	0.98	0.97	0.99	0.99	0.97	0.98	0.97	1.00	1.00	1.00
Γ(80,8000)	I	0.94	0.95	0.96	0.93	0.84	0.83	0.51	1.00	1.00	1.00

Table 1. We summarize the frequency that parameters fall within the 95% highest posterior density for simulations from the second experiment. In the "Sim" column we indicate the Gamma distributions used for simulating SNP data. In the "Prior" column "C" indicates correct priors used simulating MCMCs and "I" indicate incorrect priors used for MCMC simulation. Height is the height of the tree; length is the total sum of the branch lengths; top. is the topology of the tree.

For the second experiment we report the frequencies for which the 95% highest posterior density of the MCMC chains generated under the diffusion model were able to recover the known parameters, see Table 1. We note that in both cases of correct and incorrect priors tree height, tree length and topology were recovered. However we see that the SNP data generated from Gamma distributions with high variance made it harder to recover population size (θ) parameters We can also see a slight influence of the priors specified when looking at populations size (θ) parameter recovery rates. Most notably when incorrect priors are used recovery rates for population sizes on branches near the root decrease. It is not surprising that recovery rates of populations sizes (θ) parameters are low close to the root. This is not due to the different models that were used for simulating data (coalescent model) and inferring parameters (diffusion model). Rather it is because posterior variance on population size rapidly increases as we move up the tree. Equivalently we can say that there is very little information in the SNP data for population sizes close to the root. In contrast we see that SNP data contains a lot of information about the height, tree length and topology since recovery rates are unaffected by priors. The MCMC chains took on average approximately 200 seconds to run.

Figure 6 plots computational times of Snapp and Snapper in the third experiment. The increase in running time of Snapp as the number n of individuals increases leads to a significant difference in efficiency between Snapp and Snapper.

Analysis of wild and cultivated soybeans SNP data

Both *S*NAPP and *S*NAPPER took a bit of a week to complete ten MCMC chains of length $1,000,000$. Note that due to small number of individuals per population SNAPPER has little advantage over SNAPP in terms of runtime for this particular dataset. Tracer (Rambaut et al., 2018) was used to assess convergence by looking at trace plots and effective sample size for each parameter. We give the summary statistics in Appendix 2.

Figure 7(SNAPPER analysis) and Figure 8(SNAPP analysis) summarize the posterior distribution of the species tree using Densitree (Bouckaert, 2010). In both cases there is only one topology in the 95% highest posterior density. Posterior distributions of branch lengths are mostly indistinguishable. There are some small differences in posterior distributions of populations sizes. However in all cases population sizes follow the same apparent distributions.

Analysis of SNP data from freshwater turtles *Emydura macquarii*

It took a total of ~ 500 hours for the sampler to run on a laptop with two cores and an Intel i3-7100 CPU. We provide a complete list of inferred parameter values in the Appendix 2 and a DensiTree in Figure 9.The shape of the tree agrees with the genetic distances and SVDQuartets trees computed in Georges et al. (2018). We extend the analysis in Georges et al. (2018) by quantifying uncertainty around branch lengths and inferring population size (θ) parameters. There is some uncertainty surrounding topology in the Fitzroy clade (samples 35-43). The Cooper Creek clade (samples 96-102) is sister to the North-east Coast clade (samples 10-19) in all three analyses, as is the sister relationship between the Hunter R population (sample 92) and the Murray-Darling basin populations (samples 112-131). The East Coast clade (samples 29-60) and South-east Coast clade (samples 62-84) is sister to the Hunter-Murray-Darling basin clade (samples 92; 112-131) in the SNAPPER tree, in agreement with the Fitch-Margoliash distance tree. The Kolan-Burnett-Mary clade (samples 48-58) is sister to the Fitzroy clade in the

SNAPPER tree. *E. subglobosa* and *E. victoriae* populations, are both considered as outgroups to the remaining populations. The SNAPPER analysis supports relationships within the North-east Coast clade that reflect drainage proximity better than does the SVDQuartets analysis (Arthur George, personal communication).

Figure 10gives a close-up of the posterior distribution of *E. macquarii* populations restricted to the Murray-Darling basin. The figure is an example of the extent of uncertainty surrounding the tree due to short branch lengths. As we discuss above, population size estimates here will be mostly dependent on the prior due to little information available on such short branch lengths.

Discussion

In this paper we present Snapper, a computationally more efficient method to supersede Snapp. Like Snapp the method takes biallelic markers sampled from individuals in multiple populations and computes the likelihood of the species tree topology together with branch lengths and population sizes. We achieved this computational efficiency by computing the likelihood using diffusion models rather than the multispecies coalescent. The Wright-Fisher diffusion and coalescent are dual processes and it is therefore not surprising that the same inference can be made under these distinct but closely-related model frameworks. The application of diffusion models is made possible by a new computational framework for evaluating diffusion-based likelihoods on trees, one which does not limit the number of species, and which avoids many of the numerical head-aches faced by existing approaches.

The Snapper sampler is based on the Snapp sampler and uses the same move proposals, which are standard in the Beast2 software package. We implemented a new proposal which simultaneously updates parameter values for a subtree, noting significant improvement in sampling. There is considerable scope to develop improved sampling strategies, something that will become easier with efficient likelihood calculation.

Finally, we note that our approach could be extended to other diffusion-based models. These models could include additional parameters such as selection, migration and linkage, and could potentially be handled within the same computational and statistical framework.

Acknowledgements

We thank Arthur Georges for making the *E, macquarii* SNP dataset of available and for his help with the comparative and biogeographic interpretation inferred species phylogeny. This work was support by the NZ Marsden Fund (UOO1411).

References

Boitard, S., W. Rodríguez, F. Jay, S. Mona, and F. Austerlitz. 2016. Inferring population size history from large samples of genome-wide molecular data-an approximate Bayesian computation approach. PLoS Genetics 12:e1005877.

Bollback, J. P., T. L. York, and R. Nielsen. 2008. Estimation of $2N_e$s from temporal allele frequency data. Genetics 179:497–502.

Bouckaert, R. R. 2010. Densitree: making sense of sets of phylogenetic trees. Bioinformatics 26:1372–1373.

Bryant, D., R. Bouckaert, J. Felsenstein, N. A. Rosenberg, and A. RoyChoudhury. 2012. Inferring Species Trees Directly from Biallelic Genetic Markers: Bypassing Gene Trees in a Full Coalescent Analysis. Molecular Biology and Evolution 29:1917–1932.

Bryant, D. and M. Hahn. 2020. The concatenation question. *in* Phylogenetics in the Genomics Era (F. Delsuc, N. Galtier, and C. Scornavacca, eds.). (No commercial publisher).

Chifman, J. and L. Kubatko. 2014. Quartet inference from SNP data under the coalescent model. Bioinformatics 30:3317–3324.

Cox, J. C., J. E. Ingersoll Jr, and S. A. Ross. 1985. An intertemporal general equilibrium model of asset prices. Econometrica: Journal of the Econometric Society Pages 363–384.

Degnan, J. H. and N. A. Rosenberg. 2009. Gene tree discordance, phylogenetic inference and the multispecies coalescent. Trends in Ecology & Evolution 24:332–340.

Edwards, S. V. 2009. Is a new and general theory of molecular systematics emerging? Evolution: International Journal of Organic Evolution 63:1–19.

Epstein, C. L. and R. Mazzeo. 2010. Wright–Fisher diffusion in one dimension. SIAM Journal on Mathematical Analysis 42:568–608.

Etheridge, A. 2011. Some mathematical models from population genetics : École d'éte de probabilités De Saint-flour XXXIX-2009. Springer.

Ethier, S. N. and M. F. Norman. 1977. Error estimate for the diffusion approximation of the Wright–Fisher model. Proceedings of the National Academy of Sciences 74:5096–5098.

Ewens, W. 2004. Mathematial population genetics. I. Theoretical introduction. Interdisciplinary Applied Mathematics 2 ed. Springer, New York.

Felsenstein, J. 1981. Evolutionary trees from DNA sequences: a maximum likelihood approach. Journal of Molecular Evolution 17:368–376.

Felsenstein, J. 1992. Phylogenies from restriction sites: a maximum-likelihood approach. Evolution 46:159–173.

Fox, L. and I. Parker. 1968. Chebyshev Polynomials in Numerical Analysis. Oxford Mathematical Handbooks Oxford University Press, London.

Georges, A., B. Gruber, G. B. Pauly, D. White, M. Adams, M. J. Young, A. Kilian, X. Zhang, H. B. Shaffer, and P. J. Unmack. 2018. Genomewide SNP markers breathe new life into phylogeography and species delimitation for the problematic short-necked turtles (Chelidae: Emydura) of eastern Australia. Molecular Ecology 27:5195–5213.

Griffiths, R. C. and D. Spanó. 2010. Diffusion processes and the coalescent. Pages 358–379 *in* Probability and Mathematical Genetics: Papers in Honour of Sir John Kingman (N. H. Bingham and C. M. Goldie, eds.) London Mathematical Society Lecture Note Series. Cambridge University Press.

Gutenkunst, R., R. Hernandez, S. Williamson, and C. Bustamante. 2010. Diffusion approximations for demographic inference: ∂a∂i. Nature Precedings 5:1–1.

Gutenkunst, R. N., R. D. Hernandez, S. H. Williamson, and C. D. Bustamante. 2009.

Inferring the joint demographic history of multiple populations from multidimensional SNP frequency data. PLoS Genetics 5.

Heled, J. and A. J. Drummond. 2009. Bayesian inference of species trees from multilocus data. Molecular biology and evolution 27:570–580.

Hiscott, G., C. Fox, M. Parry, and D. Bryant. 2016. Efficient Recycled Algorithms for Quantitative Trait Models on Phylogenies. Genome Biology and Evolution 8:1338–1350.

Jenkins, P. A., D. Spano, et al. 2017. Exact simulation of the Wright–Fisher diffusion. Annals of Applied Probability 27:1478–1509.

Kingman, J. 1982. The coalescent. Stochastic Processes and their Applications 13:235–248.

Lapierre, M., A. Lambert, and G. Achaz. 2017. Accuracy of demographic inferences from the site frequency spectrum: the case of the Yoruba population. Genetics 206:439–449.

Lepage, T., S. Lawi, P. Tupper, and D. Bryant. 2006. Continuous and tractable models for the variation of evolutionary rates. Mathematical Biosciences 199:216–233.

Liu, L. 2008. Best: Bayesian estimation of species trees under the coalescent model. Bioinformatics 24:2542–2543.

Liu, L., D. K. Pearl, R. T. Brumfield, and S. V. Edwards. 2008. Estimating species trees using multiple-allele DNA sequence data. Evolution: International Journal of Organic Evolution 62:2080–2091.

Liu, L., L. Yu, and S. V. Edwards. 2010. A maximum pseudo-likelihood approach for estimating species trees under the coalescent model. BMC Evolutionary Biology 10:302.

Lukic, S. and J. Hey. 2012. Demographic inference using spectral methods on SNP data, with an analysis of the human out-of-Africa expansion. Genetics 192:619–639.

Mason, J. C. and D. C. Handscomb. 2002. Chebyshev polynomials. Chapman and Hall/CRC.

McKane, A. J. and D. Waxman. 2007. Singular solutions of the diffusion equation of population genetics. Journal of Theoretical Biology 247:849–858.

Nielsen, R. 2000. Estimation of population parameters and recombination rates from single nucleotide polymorphisms. Genetics 154:931–942.

Øksendal, B. 2003. Stochastic differential equations, an Introduction with Applications. Universitext Springer.

Racimo, F., G. Renaud, and M. Slatkin. 2016. Joint estimation of contamination, error and demography for nuclear dna from ancient humans. PLoS genetics 12:e1005972.

Rambaut, A., A. J. Drummond, D. Xie, G. Baele, and M. A. Suchard. 2018. Posterior summarization in Bayesian phylogenetics using tracer 1.7. Systematic Biology 67:901–904.

Rannala, B. and Z. Yang. 2017. Efficient Bayesian species tree inference under the multispecies coalescent. Systematic Biology 66:823–842.

Schmelzer, T. and L. N. Trefethen. 2007. Evaluating matrix functions for exponential integrators via Carathéodory-Fejér approximation and contour integrals. Electronic Transactions on Numerical Analysis 29:1–18.

Sirén, J., P. Marttinen, and J. Corander. 2010. Reconstructing population histories from single nucleotide polymorphism data. Molecular Biology and Evolution 28:673–683.

Song, S., L. Liu, S. V. Edwards, and S. Wu. 2012. Resolving conflict in eutherian mammal phylogeny using phylogenomics and the multispecies coalescent model. Proceedings of the National Academy of Sciences 109:14942–14947.

Song, Y. S. and M. Steinrücken. 2012. A simple method for finding explicit analytic transition densities of diffusion processes with general diploid selection. Genetics 190:1117–29.

Steinrücken, M., A. Bhaskar, and Y. S. Song. 2014. A novel spectral method for inferring general diploid selection from time series genetic data. The Annals of Applied Statistics 8:2203–2222.

Tataru, P., M. Simonsen, T. Bataillon, and A. Hobolth. 2017. Statistical inference in the Wright–Fisher model using allele frequency data. Systematic Biology 66:e30–e46.

Trefethen, L. N. 2013. Approximation theory and approximation practice. SIAM, Philadelphia.

Vachaspati, P. and T. Warnow. 2015. ASTRID: accurate species trees from internode distances. BMC Genomics 16:S3.

Waldvogel, J. 2006. Fast construction of the Fejér and Clenshaw–Curtis quadrature rules. BIT Numerical Mathematics 46:195–202.

Williamson, S. H., R. Hernandez, A. Fledel-Alon, L. Zhu, R. Nielsen, and C. D. Bustamante. 2005. Simultaneous inference of selection and population growth from patterns of variation in the human genome. Proceedings of the National Academy of Sciences 102:7882–7887.

Wright, S. 1931. Evolution in Mendelian populations. Genetics 16:97.

Yang, Z. 2015. The BPP program for species tree estimation and species delimitation. Current Zoology 61:854–865.

Yule, G. U. 1925. A mathematical theory of evolution, based on the conclusions of Dr. J.C. Willis, F.R.S. Philosophical transactions of the Royal Society of London. Series B, containing papers of a biological character 213:21–87.

APPENDIX

Computing the likelihood efficiently

In the main text, we showed how the recursions for the partial likelihoods can be derived analytically. We did not show how to actually compute those likelihoods. Computing the likelihoods amounts to an extremely high-dimensional integration problem. Our approach is to use numerical techniques combined with dynamic programming. While the likelihoods we compute are approximate, the extent of the error decreases extremely rapidly as the number of basis functions increases.

Shifted Chebyshev polynomials

Hiscott et al. (2016) describe a general strategy for computing likelihoods numerically whereby partial likelihoods are evaluated on a mesh of values at each node, and accurate quadrature methods used to carry out the actual computation. We will extend the same strategy by using a set of basis functions to express approximate partial likelihoods instead of a mesh of values. The basis functions are cleverly chosen to help solve equation (0.9) efficiently and accurately.

The basis functions we use are called the *shifted Chebyshev polynomials of the first kind*, denoted

$$T_0^*(x), T_1^*(x), T_2^*(x), \ldots.$$

Shifted Chebyshev polynomials are defined on $[0, 1]$ and have particularly nice properties (Fox and Parker, 1968). They also have a lot of equivalent definitions, but the simplest uses the following recursion,

$$\begin{aligned} T_0^*(x) &= 1 \\ T_1^*(x) &= 2x - 1 \\ T_k^*(x) &= 2(2x-1)T_{k-1}^*(x) - T_{k-2}^*(x). \end{aligned}$$

The shifted Chebyshev polynomials are related to the better-known Chebyshev

polynomials $T_0, T_1, T_2, \ldots$ by the identity: $T_k^*(x) = T_k(2x-1)$. That is, they are obtained by shifting and scaling the Chebyshev polynomials to have domain $[0,1]$, see Mason and Handscomb (2002).

There are two main ways of using shifted Chebyshev polynomials to approximate ℓ_i^B and ℓ_i^T as functions of x. The first is to approximate the function directly as a linear combination of the shifted Chebyshev polynomials, that is, finding sets of coefficients $\lambda_{i,0}^B, \lambda_{i,1}^B, \lambda_{i,2}^B, \ldots$ and $\lambda_{i,0}^T, \lambda_{i,1}^T, \lambda_{i,2}^T, \ldots$ so that for all x in $[0,1]$ we have

$$\ell_i^B(x) \approx \sum_{k=0}^{K} \lambda_{i,k}^B T_k^*(x) \quad \text{and} \quad \ell_i^T(x) \approx \sum_{k=0}^{K} \lambda_{i,k}^T T_k^*(x).$$

It can be shown that the error in this approximation drops exponentially quickly as the number K of basis functions increases. We therefore only need to store a few coefficients in order to evaluate the partial likelihood function at any x, with small error.

The second way of obtaining an approximation is by determining the values $\ell_i^B(x)$ and $\ell_i^T(x)$ at a pre-specified set of points $x_0, x_1, \ldots, x_K$ and then finding the unique combination of coefficients $\lambda_{i,0}^B, \ldots, \lambda_{i,K}^B$ such that

$$\sum_{k=0}^{K} \lambda_{i,k}^B(x_j) = \ell_i^B(x_j)$$

for all $j = 0, 1, \ldots, K$. This is called polynomial interpolation. As it happens, there is a particular choice of points $x_0, \ldots, x_K$ for which we can switch back and forth between function values

$$\ell_i^B(x_0), \ldots, \ell_i^B(x_K)$$

and interpolation coefficients

$$\lambda_{i,0}^B, \ldots, \lambda_{i,K}^B$$

and back with little numerical error and in $O(K \log K)$ time (Waldvogel, 2006; Trefethen, 2013). These are called the Chebyshev-Lobatto points, defined by the rather opaque formula

$$x_j = \left(1 - \cos\left(\frac{2\pi j}{K}\right)\right)/2, \text{ for } j = 0, \ldots, K. \tag{A.1}$$

These points are all in the interval $[0, 1]$ with a denser packing of points nearer 0 and 1. They equal the x-coordinates of points spaced regularly around a semi-circle.

We use both coefficient values $\lambda^B_{i,k}, \lambda^T_{i,k}$ and function values $\ell^B_i(x_j), \ell^T_i(x_j)$ when approximating the partial likelihoods ℓ^B_i and ℓ^T_i.

Approximate partial likelihoods at a leaf

We compute our approximate likelihood at the bottom of a leaf branch by simply evaluating the function

$$\ell^B_i(x_j) = \binom{2n_i}{r_i} x_j^{r_i}(1 - x_j)^{2n_i - r_i} \text{ for j} = 1, \ldots, \text{K} \tag{A.2}$$

at the Chebyshev-Lobatto points. We then compute corresponding coefficients $\lambda^B_{i,0}, \lambda^B_{i,1}, \ldots, \lambda^B_{i,K}$ using the FFT algorithm (Trefethen, 2013).

Solving the diffusion model numerically

Shifted Chebyshev polynomials provide the foundation for the numerical methods we use to solve the PDE (0.9). Note that, unlike the forward diffusion (0.1), solutions to this backward diffusion are smooth (infinitely differentiable) functions , meaning that we can avoid infinite spikes and other numerical headaches encountered by those using the forward diffusion directly (Lukic and Hey, 2012).

We approximate $g(x, t)$ in terms of shifted Chebyshev polynomials

$$g(x, t) \approx \sum_{k=0}^{K} \lambda_k(t) T^*_k(x). \tag{A.3}$$

Then

$$\frac{\partial g(x, t)}{\partial t} \approx \sum_{k=0}^{K} \frac{\partial \lambda_k(t)}{\partial t} T^*_k(x), \tag{A.4}$$

$$\frac{\partial g(x, t)}{\partial x} \approx \sum_{k=0}^{K} \lambda_k(t) \frac{\partial T^*_k(x)}{\partial x} \tag{A.5}$$

and

$$\frac{\partial^2 g(x,t)}{\partial x^2} \approx \sum_{k=0}^{K} \lambda_k(t) \frac{\partial^2 T_k^*(x)}{\partial x^2}. \tag{A.6}$$

Let $\boldsymbol{\lambda}(t)$ denote the vector of values $\lambda_0(t), \ldots, \lambda_K(t)$. Formulas for the derivative of shifted Chebyshev polynomials lead eventually to a $(K+1) \times (K+1)$ matrix $\boldsymbol{D}$ such that

$$\sum_{k=0}^{K} \lambda_k(t) \frac{\partial T_k^*(x)}{\partial x} = \sum_{k=0}^{K} (\boldsymbol{D}\boldsymbol{\lambda}(t))_k T_k^*(x),$$

and

$$\sum_{k=0}^{K} \lambda_k(t) \frac{\partial^2 T_k^*(x)}{\partial x^2} = \sum_{k=0}^{K} (\boldsymbol{D}^2\boldsymbol{\lambda}(t))_k T_k^*(x).$$

After some tedious algebra we can derive a $(K+1) \times (K+1)$ $\boldsymbol{Q}$ matrix so that

$$(\beta_1(1-x) - \beta_2 x)\frac{\partial}{\partial x}\sum_{k=0}^{K} \lambda_k(t) T_k^*(x) + \frac{1}{2}x(1-x)\frac{\partial^2}{\partial x^2}\sum_{k=0}^{K} \lambda_k(t) T_k^*(x) = \sum_{k=0}^{K} (\boldsymbol{Q}\boldsymbol{\lambda}(t))_k T_k^*(x)$$

for all $x \in [0,1]$. Plugging this and (A.4) into (0.9) we obtain an equation

$$\sum_{k=0}^{K} \frac{\partial \lambda_k(t)}{\partial t} T_k^*(x) = \sum_{k=0}^{K} (\boldsymbol{Q}\boldsymbol{\lambda}(t))_k T_k^*(x)$$

for an approximate solution to the PDE, giving the system

$$\frac{\partial \boldsymbol{\lambda}(t)}{\partial t} = \boldsymbol{Q}\boldsymbol{\lambda}(t). \tag{A.7}$$

with initial condition $\boldsymbol{\lambda}(0)$ given as the Chebyshev coefficients of $g(x,0)$.

Like Bryant et al. (2012) we use rational approximations to $\exp(\mathbf{Q}t)\boldsymbol{\lambda}(0)$ using a Carathéodory-Fejér approximation (Schmelzer and Trefethen, 2007). Additionally, we use techniques adapted from Fox and Parker (1968) to take advantage of structure in the matrix $\mathbf{Q}$, allowing an implementation of the Carathéodory-Fejér approximation which runs in $O(K)$ time, outlined below.

To summarise, consider a node i and let t_i denote the length (in coalescent units) of the branch connecting i to its parent. Suppose ℓ_i^B with corresponding coefficients $\lambda_{i,0}^B, \ldots, \lambda_{i,K}^B$ is already computed at the bottom of the branch. We then compute partial likelihood ℓ_i^T with corresponding coefficients $\lambda_{i,0}^T, \ldots, \lambda_{i,K}^T$ at the top of the branch by

1. Setting $\boldsymbol{\lambda}(0) = \lambda^B_{i,0}, \ldots, \lambda^B_{i,K}$
2. Computing a numerical approximation for $\exp(\boldsymbol{Q}t_i)\boldsymbol{\lambda}(0)$.
3. Setting $\lambda^T_{i,0}, \ldots, \lambda^T_{i,K} = \boldsymbol{\lambda}(t_i)_0, \ldots, \boldsymbol{\lambda}(t_i)_K$.

Approximate partial likelihoods at a speciation

Consider a node i with two children j and k. We suppose that the approximations for the partial likelihoods $\ell^T_j(x_0), \ell^T_j(x_1), \ldots \ell^T_j(x_K)$ and $\ell^T_k(x_0), \ell^T_k(x_1), \ldots \ell^T_k(x_K)$. We then have

$$\ell^B_i(x_p) = \ell^T_j(x_p)\ell^T_k(x_p) \text{ for all } p = 0, 1, \ldots, K. \tag{A.8}$$

This computation takes $O(K)$ time.

Approximate likelihoods at the root

The final integration to carry out is over the partial likelihood at the root

$$\int_0^1 \pi(x|\beta_1, \beta_2)\ell^B_\rho(x)dx,$$

see (0.10). The density for a beta distribution (0.4) can be infinite at the boundaries, meaning that numerical integration techniques such as Clenshaw-Curtis quadrature can give poor approximations. The solution is to separate out those parts which are difficult to integrate numerically and determine them analytically. Suppose we have an approximation of $\ell^B_\rho(x)$ as a polynomial

$$f(x) = \sum_{k=0}^{K} \lambda^B_{\rho,k} T^*_k(x) \approx \ell^B_\rho(x).$$

We factor this polynomial as

$$f(x) = x(1-x)g(x) + f(0) + (f(1) - f(0))x$$

and then compute

$$\int_0^1 f(x)\pi(x|\beta_1, \beta_2)dx = \int_0^1 g(x)x(1-x)\pi(x|\beta_1, \beta_2)dx + \int_0^1 (f(0)+(f(1)-f(0))x)\pi(x|\beta_1, \beta_2)dx.$$

Noting that the first integral is now well-behaved while the second integral evaluates to $f(0) + (f(1) - f(0))\frac{\beta_1}{\beta_1+\beta_2}$ by properties of the Beta distribution.

Computing the log-likelihood of a species trees

Algorithm NUMERICAL-LOG-LIKELIHOOD in Figure 11 summarises the numerical calculation of the likelihood. This algorithm takes $O(sKlog(K))$ time per site with $O(ns)$ pre-processing, where s is the number of species, K the number of Chebyshev basis functions and n is the number of individuals at a site. The calculations require $O(sK)$ memory. The pre-processing step involves counting the frequencies of allele types in each population. In practice this step could be carried out once per data set, rather than once per tree evaluated.

The conversion to and from coefficients to function values in the Chebyshev expansion in lines 7 and 9 each takes $O(K \log K)$ time, using the FFT algorithm reviewed above. Solution of the PDE in line 10 takes $O(K)$ time.

The $O(K)$ algorithm for solving diffusions

Previously, we introduced a matrix $\mathbf{Q}$ which plays a key role in the numerical solution of diffusions. The matrix is chosen so that if

$$g(x,t) = \sum_{k=0}^{K} \boldsymbol{\lambda}(t)_k T_k^*(x) \tag{A.9}$$

then

$$(\beta_1(1-x) - \beta_2 x)\frac{d}{dx}g(x,t) + \tfrac{1}{2}x(1-x)\frac{d^2}{dx^2}g(x,t) \approx \sum_{k=0}^{K}(\mathbf{Q}\boldsymbol{\lambda}(t))_k T_k^*(x). \tag{A.10}$$

The bottleneck in the numerical method we use is the repeated solution of linear systems that look like

$$(\mathbf{Q} - z\mathbf{I})\mathbf{x} = \mathbf{v}$$

for difference complex values z and vectors $\mathbf{v}$. Using a direct method, these take $O(K^2)$ time each as $\mathbf{Q}$ is lower triangular. However we can do better, using a trick described in

Fox and Parker (1968). Here we give a very high level description of the approach.

The key idea is to apply integration twice to the LHS of (A.10). Using integration by parts multiple times we have

$$\begin{aligned}\int_0^x \int_0^y \left[(\beta_1(1-z) - \beta_2 z)\frac{d}{dz}g(z,t) + \tfrac{1}{2}z(1-z)\frac{d^2}{dz^2}g(z,t)\right] dz\, dy \\ = -\int_0^x \int_0^y g(z,t)dz + (\beta_1(1-x) - \beta_2 x - 1 + 2x)\int_0^x g(z,t)dz \\ + \tfrac{1}{2}x(1-x)g(x,t) + (\tfrac{1}{2} - \beta_1)xg(0,t). \qquad \text{(A.11)}\end{aligned}$$

We introduce two new matrices $\mathbf{X}$ and $\mathbf{Y}$. The matrix $\mathbf{X}$ is derived from properties of shifted Chebyshev polynomials, and is defined so that if g is expanded as in (A.9) then

$$\int_0^x \int_0^y g(z)dz\, dy \approx \sum_{k=0}^{K} (\mathbf{X}\boldsymbol{\lambda}(t))_k T_k^*(x)$$

The matrix $\mathbf{Y}$ comes from (A.11) and has the property that if g is expanded as in (A.9) then the RHS of (A.11) equals

$$\sum_{k=0}^{K} (\mathbf{Y}\boldsymbol{\lambda}(t))_k T_k^*(x).$$

We then obtain

$$\mathbf{XQ} \approx \mathbf{Y}.$$

The usefulness of this follows from that fact that, with the exception of two rows, all the non-zero entries in $\mathbf{X}$ and $\mathbf{Y}$ are on or near the digaonal: both matrices are almost banded. To solve $(\mathbf{Q} - z\mathbf{I})\mathbf{x} = \mathbf{v}$ we multiply both sides by $\mathbf{X}$ and solve the sparse system that results. Overall, this now takes $O(K)$ time.

795

Figure captions

Fig. 1. Simulation algorithm for allele counts under the diffusion model. Suppose S is the set of tree parameters that are needed as input for the algorithm. Branches are numbered $i = 1, 2, \ldots, 2n-2$ where, for convenience, the branches adjacent to the leaves are numbered $1, 2, \ldots, n$. The root branch is denoted by ρ. As in the text, we let X_i^T denote the allele frequency immediately below the top of branch i; X_i^B denote the allele frequency immediately above the bottom of branch i; X_ρ^B denotes the allele frequency at the root. The values $r_1, r_2, \ldots, r_n$ are the simulated red allele counts for each species. Note that in a *pre-order traversal* we visit every node in order so that the children of a node are always visited *after* the node itself.

Fig. 2. Partial likelihoods on a species trees. We show the partial likelihood $\ell_j^B(x)$ at the bottom of a leaf branch j. Moving along the branch we also show the partial partial likelihood $\ell_j^T, \ell_k^T(x)$ at the top of the leaf branches j and k as well as the partial likelihood $\ell_i^B(x) = \ell_j^T(x)\ell_k^T(x)$ after a speciation at the bottom of the parent branch i.

Fig. 3. Overview of likelihood calculation under the diffusion model. The details of the actual numerical computation can be found in Appendix 1. Suppose X represent the SNP data as input for the algorithm and let Θ be all the tree parameters as input for the algorithm. Branches are numbered $i = 1, 2, \ldots, 2n-2$ where, for convenience, the branches adjacent to the leaves are numbered $1, 2, \ldots, n$. The root branch is denoted by ρ. Note that in a *post-order traversal* we visit every node in order so that the children of a node are always visited *before* the node itself.

Fig. 4. Log scale plot of relative error for basis functions $K = 5, 6, \ldots, 50$ of 4-taxa trees: (a) balanced short; (b) caterpillar short; (c) balanced tall; (d) caterpillar tall. The sub-linear decrease in log-error corresponds to an exponential decrease in error, until the limit of machine precision is reached and no further improvements in error are possible.

Fig. 5. Log scale plot of relative error for basis functions $K = 5, 6, \ldots, 50$ of 16-taxa trees: (a) balanced short; (b) caterpillar short; (c) balanced tall; (d) caterpillar tall. The sub-linear decrease in log-error corresponds to an exponential decrease in error, until the limit of machine precision is reached and no further improvements in error are possible.

Fig. 6. Average times (in seconds) to compute the likelihood of an alignment with 1000 sites on 4- and 8-taxa trees, for Snapp and Snapper with $s = 4, 8$ (number of taxa) and $n = 10, 15, \ldots, 35$ (number of individuals).

Fig. 7. A Snapper inference of soybean species trees displayed using Densitree (Bouckaert, 2010). Branch thickness is related to relative population sizes, tree height is reported in expected number of mutations and population sizes is printed below each corresponding branch, as $\mu \pm \sigma$.

Fig. 8. A Snapp inference of soybean species trees displayed using Densitree (Bouckaert, 2010). Branch thickness is related to relative population sizes, tree height is reported in expected number of mutations and population sizes is printed below each corresponding branch, as $\mu \pm \sigma$.

Fig. 9. A 41 population Densitree of freshwater turtle *E. macquarii*. On the x-axis, variation in the tree represents uncertainty of branch lengths. Thickness of the branches represent posterior mean of population sizes. Timescale grid at the top is given in expected number of mutations per lineage per site. A finer detail plot for the Murray-Darling basin populations can be found in Figure 10.

Fig. 10. An 11 population densitree of freshwater turtle *E. macquarii* restricted to Murray-Darling basin to provide better resolution of branch lenghts and topology. We show the fraction of trees in the tree set that contain a clade as text on the graph. We also display the mean of a node as a marker on the tree. Timescale grid is given in expected number of mutations per lineage per site.

Fig. 11. Numerical computation of the likelihood in SNAPPER. The values x_j are the Chebyshev-Lobatto points (A.1) and L_k it the probability of site k given the species tree and parameters. The algorithm runs in $O(sKlog(K))$ time per site with $O(ns)$ pre-processing and $O(sK)$ memory.