Variational Inference Using Approximate Likelihood Under the Coalescent With Recombination

Xinhao Liu,[1] Huw A. Ogilvie,[1] and Luay Nakhleh[*,1,2]
[1]Department of Computer Science
[2]Department of Biosciences, Rice University, 6100 Main Street, Houston, TX 77005, USA
[*]**Corresponding author:** E-mail: nakhleh@rice.edu.
Associate Editor:

Abstract

Coalescent methods are proven and powerful tools for population genetics, phylogenetics, epidemiology, and other fields. A promising avenue for the analysis of large genomic alignments, which are increasingly common, are coalescent hidden Markov model (coalHMM) methods, but these methods have lacked general usability and flexibility. We introduce a novel method for automatically learning a coalHMM and inferring the posterior distributions of evolutionary parameters using black-box variational inference, with the transition rates between local genealogies derived empirically by simulation. This derivation enables our method to work directly with three or four taxa and through a divide-and-conquer approach with more taxa. Using a simulated dataset resembling a human-chimp-gorilla scenario, we show that our method has comparable accuracy to previous coalHMM methods. Both species divergence times and population sizes were accurately inferred. We were also able to infer the topology of the local genealogies. Furthermore, we illustrate how to scale the method to larger data sets through a divide-and-conquer approach. This accuracy means our method is useful now, and by deriving transition rates by simulation it is flexible enough to enable future implementations of all kinds of population models.

Key words: Coalescent with recombination, recombination, species tree, local genealogies, hidden Markov models, variational inference.

Introduction

A powerful and widely accepted and employed mathematical framework for capturing the evolution of genomes and their individual loci is the *theory of coalescence* (Kingman, 1982). This framework, applied to the increasingly available genomic data, has "turned theoretical population genetics on its head" (Hartl and Clark, 2007) and propelled population and phylogenomic inferences to successful applications that span several fields of biology and biomedicine (Rogers and Gibbs, 2014; Siepel, 2009). Coalescent-based models allow for estimating the values of parameters, including population divergence times, mutation and recombination rates, ancestral population sizes, population structure, etc., from patterns of site frequencies and local genealogies (Hartl and Clark, 2007; Nielsen and Slatkin, 2013).

To account for varying levels of complexities in evolutionary histories, the standard coalescent has been extended in various directions to accommodate processes such as recombination, population structure and migration, and selection (Hudson, 1990; Wakeley,

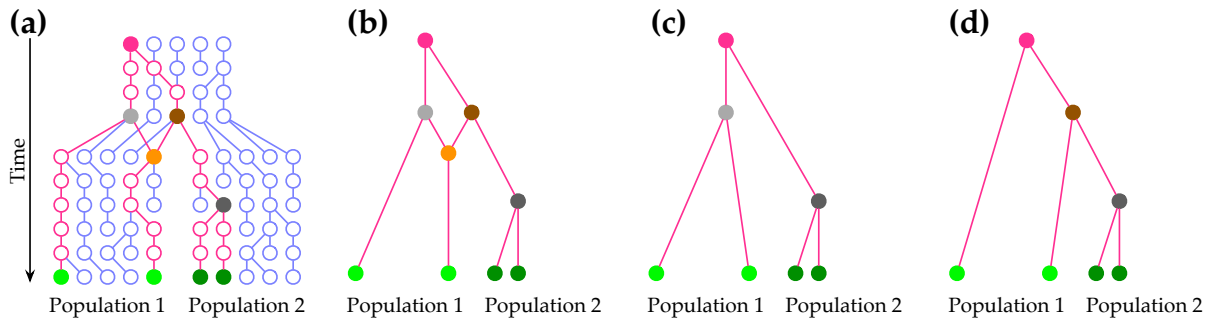

FIG. 1. The multispecies coalescent with recombination (a) The evolutionary history of a sample of four extant individuals in two divergent populations from their MRCA (solid red circle). The recombination node (solid orange circle) results in an ancestral recombination graph (ARG), shown in (b). (c) The genealogy of genomic regions that trace their evolution back from the recombination node to the grey ancestral node. (d) The genealogy of genomic regions that trace their evolution back from the recombination node to the brown ancestral node.

2008). In this work, we focus on the coalescent with recombination, illustrated in figure 1a.

As McVean and Cardin (2005) noted, the *coalescent with recombination is very difficult to estimate likelihoods under* due to, at least, three important issues: (1) the state-space of recombining genealogies (also known as *ancestral recombination graphs*, or ARGs, illustrated in figure 1b) is huge; (2) the data are generally not very informative about the actual ARG; and (3) likelihood estimation is a missing-data problem with highly redundant augmentation.

The consequence of these issues is seen in BACTER (Vaughan *et al.*, 2017), a Bayesian method that uses MCMC to infer the ARG posterior distribution down to the coalescent and recombination times and genomic boundaries of recombinant segments. While this kind of joint inference yields the most detailed posterior distribution, calculating the likelihood of an ARG scales poorly as the number of recombinations increases, and BACTER is limited to analysis of an unstructed population (Vaughan *et al.*, 2017).

Coalescent inference can be scaled up using multilocus methods that assume each locus is spaced far enough apart so that there is effectively no linkage between loci, and each locus is short enough so that no recombination has occured within it. However, the posterior distribution of local geneologies will be diffuse and incomplete, and the assumption of no recombination within loci has been called into question (Springer and Gatesy, 2016).

To strike a balance between the scalability of multilocus methods and the power of ARG inference, the coalescent with recombination can be approximated as a sequential process operating across the genome, rather than operating in time along the branches of the phylogeny (Hein *et al.*, 2005). Using this view, the coalHMM (for "coalescent hidden Markov Model") was introduced (Hobolth *et al.*, 2007). In this model, an HMM is built such that every coalescent history (gene history) given the species tree is modeled by a state, the transition probabilities are derived based on the recombination rate and the given genealogies, and the emission probabilities are given by the likelihood of the gene trees (Felsenstein, 1981). Figure 1c,d show two gene histories that are embedded inside the ARG shown in figure 1b.

In the work of (Hobolth *et al.*, 2007), the authors determined the transition probabilities by careful inspection of recombination scenarios given the species tree. Later, Dutheil *et al.* (2009) provided a detailed mathematical derivations under the coalescent with recombination of the model of Hobolth *et al.* (2007). Such a manual approach to deriving transition probabilities has limited coalHMM-based inference of evolutionary parameters to three genomes.

In this work, we propose a new approach for automatic derivation of the coalHMM and parameter estimation under it. In this approach, the likelihood of a candidate model (species tree, divergence times, and population sizes) is computed by simulating data under the coalescent with recombination using the candidate model, using this data to automatically construct an HMM, and then computing the likelihood by means of the Forward algorithm (Baum *et al.*, 1972, 1970; Chang and Hancock, 1966). Using this automated procedure for likelihood calculations, we introduce a novel application of variational inference for parameter estimation (the species tree topology is assumed to be known and fixed). We demonstrate the utility and accuracy of our method on both simulated and biological data. Furthermore, we discuss and provide preliminary results for how to scale the method to larger numbers of taxa using a divide-and-conquer approach. The automated nature of our method provides a step towards wider applicability of the coalescent with recombination.

New Approaches

Model parameters

Given the topology of a species tree, we seek to estimate its continuous parameters from the genomic data under the (multispecies) coalescent with recombination.

As we stated above, maximum likelihood estimation of the topology's parameters under the exact complex model of the coalescent with recombination is intractable. We introduce a novel variational Bayesian method for accomplishing this estimation by using a simulation-based likelihood kernel. We derive an empirical, simulation-based coalHMM and perform the likelihood computations on the HMM, which can be done in polynomial time in the number of states (Durbin *et al.*, 1998).

Let Ψ be a species tree on set $\mathscr{X}$ of taxa, $\mathbf{N}$ be a vector of the effective population sizes associated with Ψ's internal and root branches, and $\mathbf{T}$ be a vector of the divergence times of the internal nodes of Ψ. Let $\Theta = [\mathbf{N};\mathbf{T}]$. We fix the hyperparameters of the prior distributions on the parameters Θ. We also assume a fixed mutation rate μ and recombination rate ρ. Our method assumes one sequence from each extant population so the posterior distribution of tip branch population sizes will be identical to the prior and, hence, not estimated by our method.

Simulation-based likelihood approximation

For a fixed species tree topology Ψ, given a specific Θ and a sequence alignment $\mathcal{S}$, we seek to compute the likelihood of Θ given by $P(\mathcal{S}|\Theta)$. Note that this likelihood marginalizes over the local geneology at each

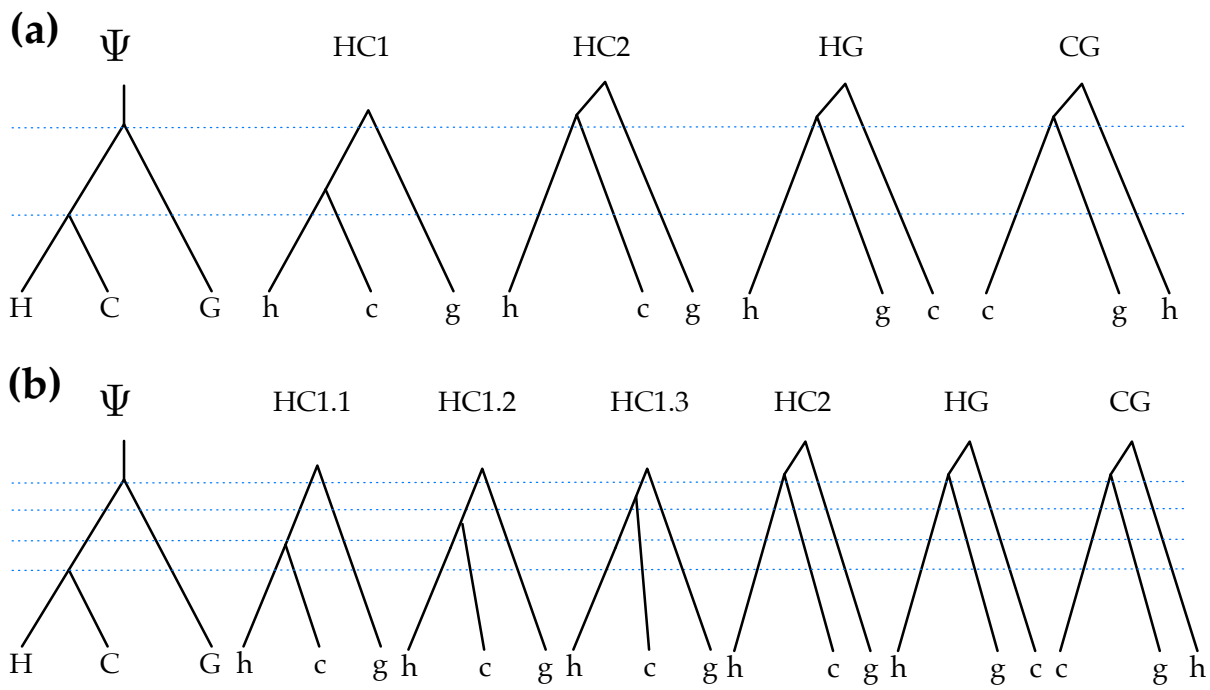

FIG. 2. The coalHMM model. (a) The four states of a standard coalHMM that corresponds to the species tree Ψ. (b) The six states of a refined coalHMM that corresponds to the species tree Ψ when its internal branch is broken into three sub-branches.

site. Algorithm 1 gives the procedure for approximate likelihood computation.

Algorithm 1: ApproximateLikelihood.

Input: Species tree topology Ψ. Sequence alignment $\mathcal{S}$. Continous parameters Θ. Number of sub-branches nb. Simulation length ℓ.
Output: Approximate likelihood $P(\mathcal{S}|\Theta)$.

$\mathscr{G} = \langle g_1, g_2, \ldots, g_k \rangle \leftarrow$ **CR**$(\Psi, \Theta, \rho, \ell)$;
$\mathcal{M} \leftarrow$ **BuildcoalHMM**$(\mathscr{G}, nb, \mu)$;
$L \leftarrow$ **Forward**$(\mathcal{S}, \mathcal{M})$;
return L;

CR (for **C**oalescent with **R**ecombination) runs a coalescent-with-recombination simulator to generate a sequence $\mathscr{G}$ of k local genealogies corresponding to ℓ sites under the model specified by Ψ and Θ. Here, each of the k genealogies correspond to a contiguous genomic region of one or more sites, the genomic regions of the genealogies are pairwise disjoint, and the concatenation of the k genomic regions yields a region of ℓ sites. Each of the k genomic regions is recombination-free, and every two consecutive regions are separated by at least one recombination event. In the implementation, we use `msprime` (Kelleher *et al.*, 2016), a reimplementation of Hudson's classical `ms` simulator (Hudson, 2002) for efficient coalescent simulations. There is clearly a trade-off between computational requirements and accuracy when setting the value of ℓ, which we discuss in the Results section below.

After the sequence $\mathscr{G}$ of gene trees is produced, **BuildcoalHMM** empirically builds a coalHMM as follows. In its basic version, **BuildcoalHMM** builds a coalHMM with one state per coalescent history (Degnan and Salter, 2005) given the species tree. For example, for the species tree Ψ in figure 2a, the basic coalHMM would have four states corresponding to the four coalescent histories HC1, HC2, HG, and CG. This is precisely the model used in (Hobolth *et al.*, 2007). However, as discussed elsewhere (Dutheil *et al.*,

2009), having one state per coalescent history could result in unidentifiability of some of the parameters. To ameliorate this problem, **BuildcoalHMM** in our method can refine the states further by segmenting branches in the species tree into contiguous non-overlapping sub-branches, and refining individual coalescent histories based on this segmentation. This concept is illustrated in figure 2b, where the internal branch separating (H,C) from the root of the tree is segmented into three sub-branches. Now, coalescent history HC1 of figure 2a is refined into three coalescent histories, HC1.1, HC1.2 and HC1.3, each corresponding to a unique mapping of the coalescent history of h and c to a sub-branch. The number of sub-branches is controlled by the parameter *nb* in Algorithm 1. We explore the impact of *nb* on the accuracy and computational requirements in the Results section below. Finally, the transition probabilities are derived empirically from the simulated coalescent histories, estimating the rate of transition from one history to another by simple counting of the number of transitions in the simulation. The emission probabilities for each state at each alignment column of $\mathcal{S}$ are computed by Felsenstein's pruning algorithm (Felsenstein, 1981). In the implementation, we use the BEAGLE library (Ayres *et al.*, 2012) for efficient implementation of Felsenstein's algorithm.

Finally, once the coalHMM is built, **Forward** runs the Forward algorithm (Durbin *et al.*, 1998) to compute $p(\mathcal{S})$ as an approximation of the likelihood $P(\mathcal{S}|\Theta)$.

Bayesian formulation and variational inference

As noted above, the data in our case is a sequence alignment $\mathcal{S}$ on the set of taxa of the species tree. We are interested in the posterior $p(\Theta|\mathcal{S}) \propto P(\mathcal{S}|\Theta)p(\Theta)$, where we assume a prior distribution on Θ. Exact computation of the pesterior is intractable, so we use variational inference to find an approximate distribution to $p(\Theta|\mathcal{S})$. In variational inference, we posit a simple family of distributions over Θ and try to find the member of the family closest in terms of the Kullback-Leibler (KL) divergence to the true posterior $p(\Theta|\mathcal{S})$ (Bishop, 2006). Denote the variational distribution we posit on Θ as $Q(\Theta|\lambda)$, governed by a set of free parameters λ, our goal is to approximate $p(\Theta|\mathcal{S})$ by optimizing λ to make $Q(\Theta|\lambda)$ as close in KL divergence to $p(\Theta|\mathcal{S})$ as possible. In variational inference, we optimize the Evidence Lower BOund (ELBO), given by

$$\begin{aligned}\mathcal{L}(\lambda) &:= E_{Q(\Theta|\lambda)}[\log p(\mathcal{S},\Theta) - \log Q(\Theta|\lambda)] \\ &= E_{Q(\Theta|\lambda)}[\log P(\mathcal{S}|\Theta) + \log p(\Theta) - \log Q(\Theta|\lambda)].\end{aligned} \tag{1}$$

Maximizing $\mathcal{L}(\lambda)$ amounts to minimizing the KL divergence from Q to p. Due to the nature of the coalHMM likelihood, it is not possible to derive a closed-form coordinate-ascent optimization algorithm as most traditional variational methods do. Therefore, we use Black Box Variational Inference (BBVI) (Ranganath *et al.*, 2014). BBVI is a stochastic optimization algorithm using noisy estimates of the gradient to maximize the ELBO, without the need for model-specific derivations (hence the "black box"). The gradient of the ELBO (Eq. (1)) with respect to λ can be written (Ranganath *et al.*, 2014) as

$$\begin{aligned}\nabla_\lambda \mathcal{L} = E_{Q(\Theta|\lambda)}[&\nabla_\lambda \log Q(\Theta|\lambda)\cdot \\ &(\log P(\mathcal{S}|\Theta) + \log p(\Theta) - \log Q(\Theta|\lambda))],\end{aligned} \tag{2}$$

and its noisy unbiased Monte Carlo estimate is

$$\nabla_\lambda \mathcal{L} \approx \frac{1}{N}\sum_{n=1}^{N}[\nabla_\lambda \log Q(\Theta^{(n)}|\lambda)\cdot$$
$$(\log P(\mathcal{S}|\Theta^{(n)}) + \log p(\Theta^{(n)}) - \log Q(\Theta^{(n)}|\lambda))], \quad (3)$$

where $\Theta^{(n)} \sim Q(\Theta|\lambda)$ is the n-th of N samples from the current variational distribution. All parts of the equations are known: $\nabla_\lambda \log Q(\Theta^{(n)}|\lambda)$ is the score function (Cox and Hinkley, 1979) of the current variational distribution; an approximation of $\log P(\mathcal{S}|\Theta^{(n)})$ is computed by Algorithm 1 above; $\log p(\Theta^{(n)})$ is the prior; and $\log Q(\Theta^{(n)}|\lambda)$ is computation about the variational distribution itself. Using Eq. (3) we can compute noisy gradients of $\mathcal{L}(\lambda)$ from samples of the variational posterior, and therefore we are able to do stochastic gradient ascent in the space of $\mathcal{L}(\lambda)$ to optimize λ.

Factorized approximation

As the variational family of $Q(\Theta)$, we assume the variational distribution to be a factorized Gaussian. Each parameter in Θ has a univariate Gaussian with a mean and a standard deviation. That is, each population size and node height of the species tree that we are interested in is independent and has a Gaussian variational posterior with its own mean and standard deviation. We have

$$Q(\Theta|\lambda) = \prod_{i=1}^{M} Q_i(\Theta_i|\lambda_i) = \prod_{i=1}^{M} \mathcal{N}(\Theta_i|\mu_i, \sigma_i), \quad (4)$$

where $M = |\Theta|$ is the number of continuous parameters (divergence times and population sizes) associated with Ψ.

The per-component gradient of the ELBO with respect to each component of λ then becomes

$$\nabla_{\lambda_d} \mathcal{L} = E_{Q(\Theta|\lambda)}[\nabla_{\lambda_d} \log Q_i(\Theta_i|\lambda_i)\cdot$$
$$(\log P(\mathcal{S}|\Theta) + \log p(\Theta) - \log Q_i(\Theta_i|\lambda_i))], \quad (5)$$

where λ_d belongs to the i-th factor of the factorized variational distribution. For factorized Gaussian, each factor has two λ components, mean and standard deviation. Taking all together, Algorithm 2 gives the general BBVI framework for inferring the variational posterior of the continuous parameters on the species tree of interest under the multispecies coalescent with recombination.

Variance reduction and adaptive learning rate

Algorithm 2 gives the basic structure for Bayesian inference under coalescent with recombination with a simulation-based likelihood kernel. However, a few more challenges remain to be addressed to make it useful. In particular, the variance of the Monte Carlo estimator of the gradient given in Eq. (3) can be too large to be useful. To reduce the variance of the sampled estimator, we use control variate (Ranganath *et al.*, 2014; Ross, 1997). A control variate is a family of functions with equivalent expectation but smaller variance than the function being approximated by Monte Carlo. Details of the application of control variates to BBVI can be found in (Ranganath *et al.*, 2014).

Another crucial challenge is setting the learning rate schedule. A large learning rate might overshoot the optimum, but a small learning rate might never converge. Moreover, the variational distribution in our problem has different scales (the scale of the population sizes

Algorithm 2: Black Box Variational Inference.

Input: Species tree topology Ψ. Sequence alignment $\mathcal{S}$. Number of sub-branches nb. Simulation length ℓ. Number of iterations T. Number of samples per iteration N. Learning rate α.

Output: $\lambda_{1:D}$ of the optimized variational posterior $Q(\Theta|\lambda)$.

Initialize λ randomly;
for $t \leftarrow 1$ *to* T **do**
  for $n \leftarrow 1$ *to* N **do**
    $\Theta^{(n)} \sim Q(\Theta|\lambda)$;
  for $d \leftarrow 1$ *to* D **do**
    $\hat{\nabla}_{\lambda_d}\mathcal{L} = \frac{1}{N}\sum_{n=1}^{N}[\nabla_{\lambda_d}\log Q_i(\Theta_i^{(n)}|\lambda_i)\cdot(\textbf{ApproximateLikelihood}(\Psi,\mathcal{S},\Theta^{(n)},nb,\ell)+\textbf{Prior}(\Theta^{(n)})-\log Q_i(\Theta_i^{(n)}|\lambda_i))]$;
    $\lambda_d \leftarrow \lambda_d + \alpha\hat{\nabla}_{\lambda_d}\mathcal{L}$;
return λ;

and the node heights are different), so we would like our learning rate to be able to handle the smallest scale while not too small for the largest scale. As a result, we implement the AdaGrad (Duchi *et al.*, 2011) optimizer to adaptively set the learning rate. AdaGrad adapts each learning rate by scaling it inversely proportional to the square root of the sum of all the squared past values of the gradient, resulting in greater progress in the more smoothly sloped direction of the parameter space and smaller progress otherwise. AdaGrad is a per-parameter updater, meaning it has a different adaptive learning rate for each parameter, addressing the multi-scale problem of our distribution. Other off-the-shelf optimizers like RMSProp (Tieleman and Hinton, 2012) and Adam (Kingma and Ba, 2014) can also be easily implemented within our framework.

A divide-and-conquer approach to larger numbers of genomes

Algorithm 2 is able to handle an arbitrary number of taxa at increasing computational costs. In practice, however, running it on more than three taxa takes a long time because the number of states grows super-exponentially with the number of taxa. We propose a divide-and-conquer approach to ameliorate this problem. The method first divides the set of taxa into overlapping three-taxon subsets whose subtree parameters cover all the parameters of the full species tree, and then infers the parameters of each subtree using Algorithm 2.

For the divide step, in order to cover all the continuous parameters of the full species tree Ψ, we only need to cover all the internal edges. Using three leaves to cover an internal branch $e = (u,v)$ in Ψ, we need one leaf from the left child clade of v, one leaf from the right child clade of v, and one leaf from the right child clade of u. The process is illustrated in figure 3. The full species tree on taxa A, B, C, and D has two internal branches, each of which is covered by one of the two subtrees shown in the figure. There are a total of six continuous parameters for the full tree: node heights T_1, T_2, and T_3, population sizes N_{12}, N_{23}, and root population size N_1. Analysis of data from taxa A, B, and C allows for inferring the parameters T_2, T_3 and N_{23}. Analysis of data from taxa A, C, and D allows for inferring the parameters T_1, T_2, N_{12}, and N_1.

While in this work we consider each subset independently when running Algorithm 2, a future direction involves implementing an algorithm that is

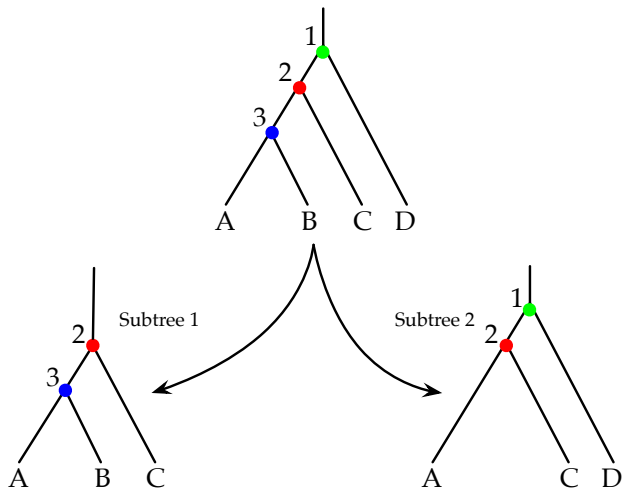

FIG. 3. Divide-and-conquer inference on a four-taxon data set. The set of taxa $\{A,B,C,D\}$ is divided into two sets $\{A,B,C\}$ and $\{A,C,D\}$ along with their respective species trees. Each of the two data sets are analyzed and the results are merged.

aware of the parameter overlap. For example, when the two data sets are analyzed, the algorithm is made aware of the fact that T_2 is the same parameter in both data sets. This should scale linearly with the number of taxa, as we would need to infer the parameters for one triplet per internal branch.

Methods

Simulated data

We simulated 100 datasets with 500,000 sites each, intended to resemble human-chimp-gorilla (HCG). We refer to the three extant species as human (H), chimp (C) and gorilla (G), and the ancestral species as the human-chimp ancestor (HC) and the human-chimp-gorilla ancester (HCG). The continuous parameters used in simulation are population sizes $N_{HC} = N_{HCG} = 40,000$, $N_H = N_C = N_G = 30,000$, speciation time $T_{HC} = 160,000$ generations (or 4 Myr assuming a generation time of 25 years), speciation time $T_{HCG} = 220,000$ generations (or 5.5 Myr assuming a generation time of 25 years). The recombination rate is $r = 1.5 \times 10^{-8}$ per site per generation, corresponding to a genetic recombination frequency of 1.5 cM per Mb. The mutation rate is 2.5×10^{-8} per site per generation, corresponding to 0.1% change per million years assuming a generation time of 25 years. The parameters are the same as used in the other two human-chimp-gorilla simulation studies of coalescent HMM (Dutheil *et al.*, 2009; Hobolth *et al.*, 2007).

As in Algorithm 1, we used the program `msprime` (Kelleher *et al.*, 2016) as the simulator for coalescent with recombination process, and `INDELible` (Fletcher and Yang, 2009) as the sequence evolution simulator.

Results and Discussion

Simulation study on human-chimp-gorilla alignment

We conducted a simulation study on an alignment resembling human-chimp-gorilla to demonstrate our method's ability to accurately infer continuous parameters of a three-taxon species tree. The simulation setup consist of two steps. In the first step, we took the demographic parameters of a species tree and simulated under the coalescent with recombination process. This step gave us a set of segments of the sequence, where each segment had a corresponding coalescent tree. The second step used standard evolutionary simulators to generate sequence alignments at each segment at the given substitution rate under the coalescent tree at that segment. The result of the simulation was a sequence alignment for the set of taxa, where different sites in the alignment had potentially different genealogies.

For each dataset, black box variational inference with simulation-based likelihood was used to find

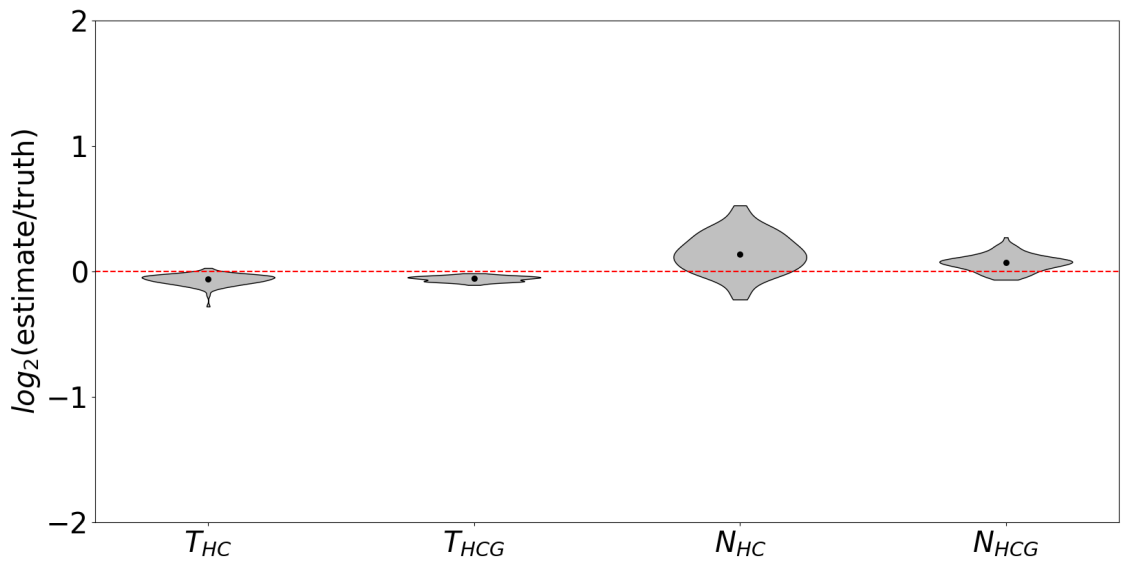

FIG. 4. Accuracy results of our method on simulated human-chimp-gorilla datasets. The violin plot shows the base-2 logarithm of the relative error (estimate/truth) for the analysis of 100 datasets. A value of 0 (the red dashed line) represents an exact estimate.

the variational posterior of each continuous parameter. The configuration for constructing simulation-based coalHMM as in Algorithm 1 is $nb = 2$ and `-r` $= 1000$. The simulation length ℓ is determined automatically using a scaling formula described below. For stochastic variational inference search, we used 50 samples per iteration with 200 iterations. We used uniform prior on node heights and gamma prior on population sizes. For each data set, the inference took 10.08 hours on a Macbook Pro with 2.4 GHz Intel Core i5 CPU. 3.93 hours were used to build coalHMM by simulation, and 6.15 hours were used to compute likelihood of samples by forward algorithm.

Figure 4 shows the accuracy of the maximum a posteriori (MAP) estimates of each parameter as a violin plot. All parameters are estimated with very good accuracy and little variability, indicating the correctness and consistency of our method. Generally, population sizes are estimated with larger variability than node heights. Since our method is a Bayesian approach, we are also interested in the 95% credible interval of the estimation. Figure 5 shows, for each parameter, the number of inferences where the true value is within 95% credible interval of the estimation. For all parameters, the true value falls within 95% credible interval most of the time. The method infers node heights T_{HC} and T_{HCG} correctly almost always, with only three exceptions for T_{HC}. Population sizes fall outside the 95% credible interval more times than node heights, agreeing with our observation that population sizes are estimated with

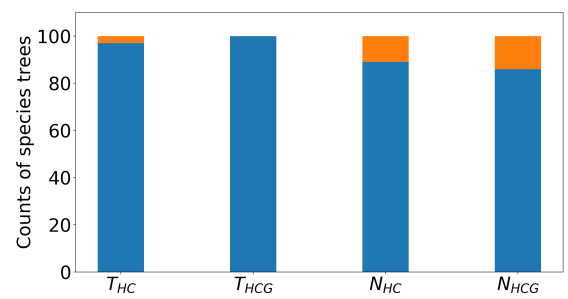

FIG. 5. Accuracy of variational posteriors. For each parameter, the number of inferences with the true value within 95% credible interval of the estimation (blue) and outside 95% credible interval of the estimation (orange) are shown.

larger variability than node heights. We note that N_{HCG} falls outside the 95% credible interval more times than N_{HC}, due to the fact that the Gaussian posterior on N_{HCG} is usually estimated to have lower standard deviation than N_{HC}.

Analysis of a biological dataset

We reanalyzed the empirical human-chimp-gorilla sequences from Hobolth *et al.* (2007) for comparison with previous models. We reanalyzed target 106 (Chromosome 20) of Hobolth *et al.* (2007) and compared our result with that of Hobolth *et al.* (2007) and Dutheil *et al.* (2009) done on the same data. The coalHMM derivation configuration and black box variational inference settings were the same as in the simulation study above. We used a recombination rate $r = 2 \times 10^{-9}$ per site per generation, and a mutation rate $\mu = 2.35 \times 10^{-8}$ per site per geneation, as they were estimated from pedigree data and reported in Dutheil *et al.* (2009) for this target. The result is shown in figure 6. Generally, our method infers comparable results with both methods. For T_{HC}, all three methods infer about the same value. For T_{HCG}, our method is closer to the estimation of the 2009 model than the 2007 one. It should suggest our estimate is accurate because the 2009 model result is achieved after bias correction is applied to the 2009 model. Our estimation of population sizes are also reasonable and on the same scale as the other two methods.

Local genealogy inference

Other than inferring continuous parameters, an important capability of coalHMMs is to infer the local genealogy of each site along the genome. Since the hidden states of a coalescent hidden Markov model are coalescent histories (genealogies), local genealogy inference can be done by posterior decoding of the HMM along the sequence data, which gives us the posterior probability of each genealogy at each site. In this section, we study the efficacy of our simulation-based HMM in local genealogy inference.

We used the same 100 simulated datasets as in the simulation study above. Since we used `msprime` to simulate under the coalescent with recombination process when generating data, we have the true coalescent tree of each site. We used $nb = 2$ and `-r` $= 1000$ to build our HMM by simulation. For the human-chimp-gorilla species tree, there are four types of genealogies: HC1, HC2, HG, and CG, as shown in figure 2a. Since we fine-grained each branch into two sub-branches, our HMM has a higher granularity than four genealogies. The total number of states in our HMM is actually 13. However, for the purpose of local genealogy inference, we only consider the four basic types as they are the most meaningful categorization for determining the shared ancestry of molecular characters or traits. Therefore, after posterior decoding on the 13-state HMM, we merged hidden states of the same type togther and took the type with the highest posterior probability as the infered genealogy at each site. We also discretized the true coalescent tree at each site into one of the four genealogies. We then compared the inferred genealogy with the true one. Table 1 shows the confusion matrix of the classification task, as well as the precision and recall measures for each type of genealogy. Figure 7 shows a graphical comparison of the posterior probabilities of each genealogy at each site with the

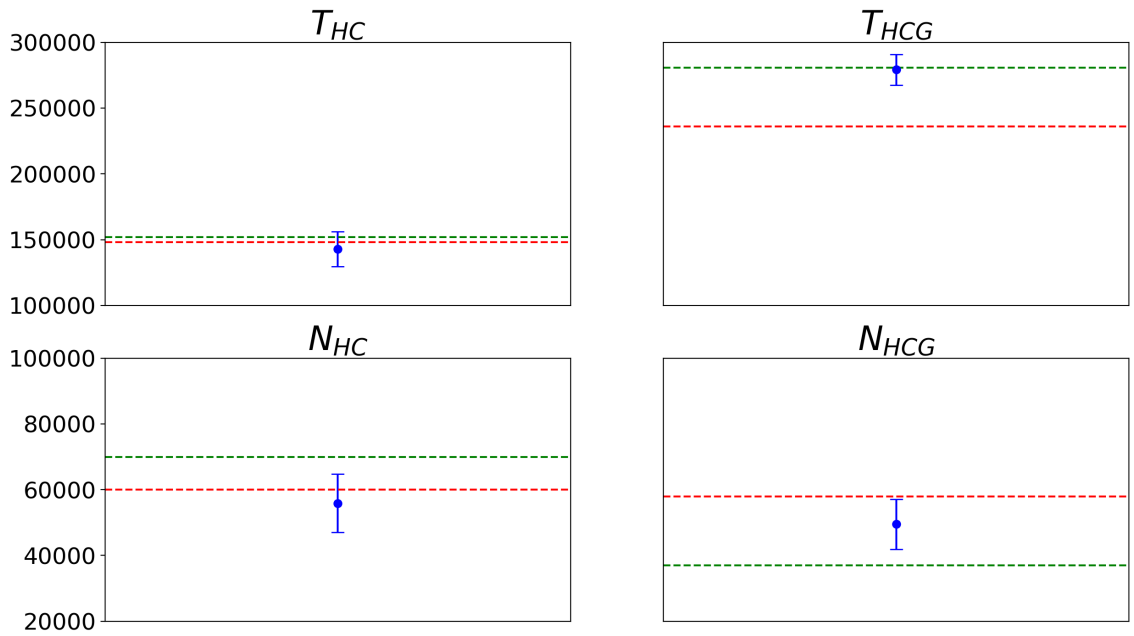

FIG. 6. Inference results on target 106. The factorized normal variational posterior distribution of each parameter inferred by our method is shown in blue. The dashed red line shows the maximum likelihood solution obtained by Hobolth *et al.* (2007). The dashed green line shows the maximum likelihood solution obtained by Dutheil *et al.* (2009). For the 2009 model, we took the result after bias correction.

Table 1. Classification accuracy of local genalogies on 100 simulated datasets. (a) A confusion matrix of the genealogy classification task based on posterior decoding. Sums over rows give the frequencies of true genealogies, and sums over columns give the frequencies of inferred genealogies. The diagonal corresponds to correctly inferred cases. (b) The precision measure for each genealogy, as defined by the number of true positives over the number of positives. (c) The recall measure for each genealogy, as defined by the number of true positives over the true number of sites with that genealogy.

	(a) Posterior Decoding				(b) Precision	(c) Recall
Inferred genealogy / True genealogy	HC1	HC2	HG	CG		
HC1	47.56%	0.52%	2.23%	2.32%	66.73%	90.38%
HC2	11.32%	0.70%	1.89%	1.93%	38.44%	4.44%
HG	6.17%	0.30%	7.12%	2.02%	53.27%	45.60%
CG	6.21%	0.31%	2.12%	7.26%	53.67%	45.64%

true genealogy along the sequence from a segment of 100,000 sites from one of the datasets.

The results show that the accuracy of local genealogy inference is very good: 90% of true HC1 sites are inferred to have HC1 genealogy (Table 1). This number is 46% for HG and CG, with an overall classification accuracy of 63%. This number is significantly higher than random expectation (Dutheil *et al.*, 2009). We observe the same good perforamance in figure 7, where there is a good correspondence between true genealogy and posterior distribution. However, note that the recall measure of HC2 is only 4.44%, meaning only 4.44%

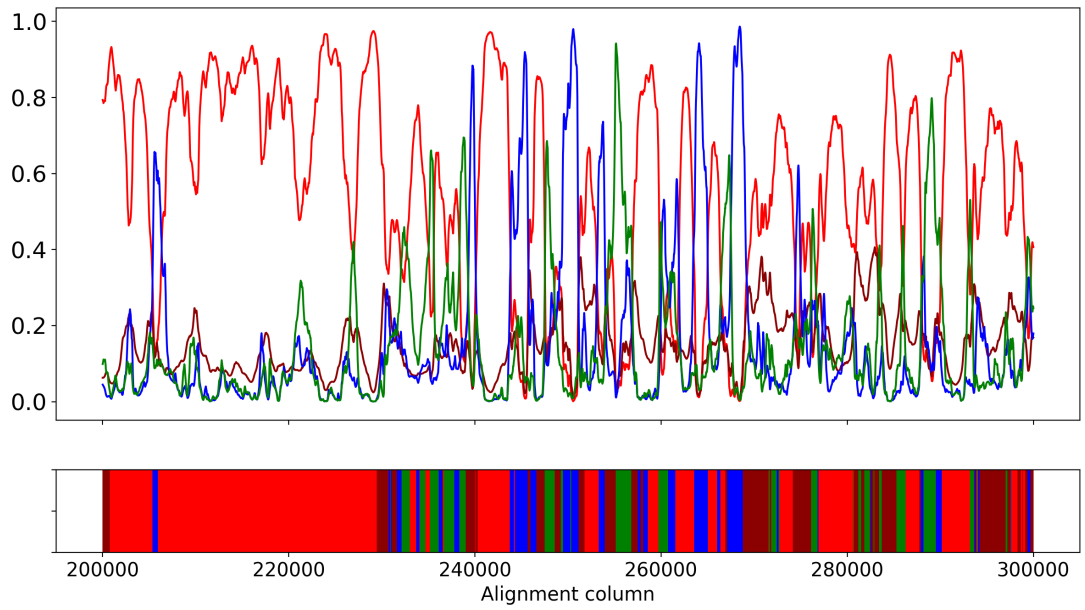

FIG. 7. True genealogy and posterior distribution along the sequence. The upper panel shows posterior probability of each genealogy at each site. The lower panel shows the true genealogy of each site. Coloring corresponds to different genealogies: genealogy HC1 is in red, HC2 is in dark red, HG is in blue, and CG is in green.

of all true HC2 genealogies are actually estimated to be HC2. Dutheil *et al.* (2009) reported the same poor performance on HC2. Many sites with HC2 as true genealogy are assigned to another type, mostly HC1 (Dutheil *et al.*, 2009). This is likely a model artifact of the HMM approximation to the coalescent with recombination process. We already know that the HMM approximation would underestiamte the recombination rate (Dutheil *et al.*, 2009; Mailund *et al.*, 2011), which means it would underestimate state transitions, leading to a global underestimation of incomplete lineage sorting. For most of the true HC2 sites, it is unsurprising that these sites are misclassified as HC1, since the stationary frequency of HC1 is so much higher and given that the site patterns of true HC1 sites and true HC2 sites should be similar.

Relationship between inference accuracy, number of sub-branches, simulation length, and branch length

The accuracy of inferences depends on the quality of the approximate likelihood, which in turn depends on two aspects: the accuracy of the simulation-based coalHMM approximation of the coalescent with recombination process, and the quality of the trained coalHMM itself. The accuracy of the coalHMM approximation of the coalescent with recombination process is determined by the refinement of coalHMM state space, *i.e.,* the number of sub-branches on each branch of the species tree when building the HMM. If the number of sub-branches is small, the resulting coalHMM has a state space of coarse coalescent histories not enough to capture the detailed coalescent distribution, leading to biased likelihood (Dutheil *et al.*, 2009; Mailund *et al.*, 2011). The quality of the coalHMM itself (*i.e.,* the quality of the transition

matrix) is determined by the length of simulation used to derive the HMM since our coalHMM is trained directly from labeled sequence data. The more sub-branches we use to refine a branch, the more accurate approximation we obtain, but more sub-branches incur a larger state space, necessitating a longer simulation length in order to train a high-quality transition matrix. If we use a large number of sub-branches but a small simulation length, the resultant HMM will be unreliable because of limited training data. Moreover, depending on the branch length, we may not need a large number of sub-branches to approximate the coalescent process on that branch, but a number of sub-branches too small may introduce bias. In this section, we study the relationship between accuracy of inference result, number of sub-branches used to refine HMM state space, length of simulation used to build HMM, and branch length of the species tree we do inference on. We derive some empirical suggestions on what number of sub-branches and simulation length to use for any specific inference problem.

There are two hyper-parameters controlling the HMM building process in Algorithm 1: *nb*, the number of sub-branches on each branch of the species tree, and l, the length of simulated HMM training data. In our implementation, they are user-defined inputs `NUM_BIN` and `CROSS_OVER_RATE`. `NUM_BIN` is the number of sub-branches used to approximate each branch of the species tree. `CROSS_OVER_RATE` is the `-r` switch of `ms` (Hudson, 2002), which is defined as $4N_0r$, where N_0 is a customized population size and r is the probability of recombination per generation between the ends of the locus being simulated, which is the probability of recombination per site per generation times the number of sites to simulate. The value of N_0 does not matter and our implementation assumes 10000, and it can be changed according to the scale of population sizes. For example, if the recombination rate is 1.5×10^{-7}/site/generation and we seek to simulate 500000 sites, we use $4 \times 10000 \times 1.5 \times 10^{-7} \times 500000 = 3000$ as the value for the `-r` option.

We simulated sequence alignments under three different evolutionary scenarios, the difference between which is the internal branch length. All three scenarios have the same three-taxon tree topology $((A,B),C)$;. Internal node height T_{AB} is fixed at 100,000 generations. The root node height varies across scenarios. Scenario 1 has root node height 150,000 generations. Scenario 2 has root node height 200,000 generations. Scenario 3 has root node height 400,000 generations. Population sizes of all branches are fixed at 40,000. Hence, the branch length of the internal branch for scenarios 1, 2, and 3 in coalescent units are 0.625, 1.25, and 3.75, respectively. The recombination rate is $r = 1.5 \times 10^{7}$/site/generation. The mutation rate is 1.25×10^{-6}/site/generation. The length of the sequence is 100,000 bp.

We inferred the continuous parameters of each scenario under various configurations of the hyper-parameters. The number of sub-branches explored were 1, 2, 3, 4, and 5. The values for the `-r` parameter explored were 500, 1000, 3000, and 5000, which correspond to simulation lengths of 83333, 166666, 500000, and 833333, respectively. For each combination of number of sub-branches and `-r` parameter value (simulation length), we conducted inference on each scenario using

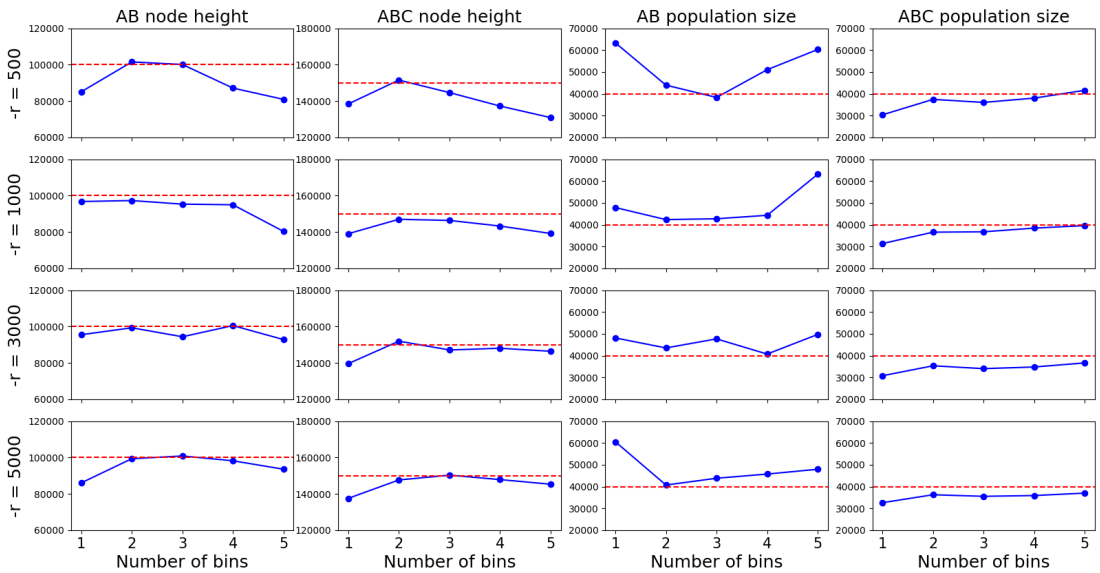

FIG. 8. Results on scenario 1 (internal branch length 0.625). Dashed red lines are true values. Blue lines are inferred MAP values. Rows correspond to different simulation lengths. Columns corresponds to different continuous parameters. x-axes are number of sub-branches ranging from 1 to 5.

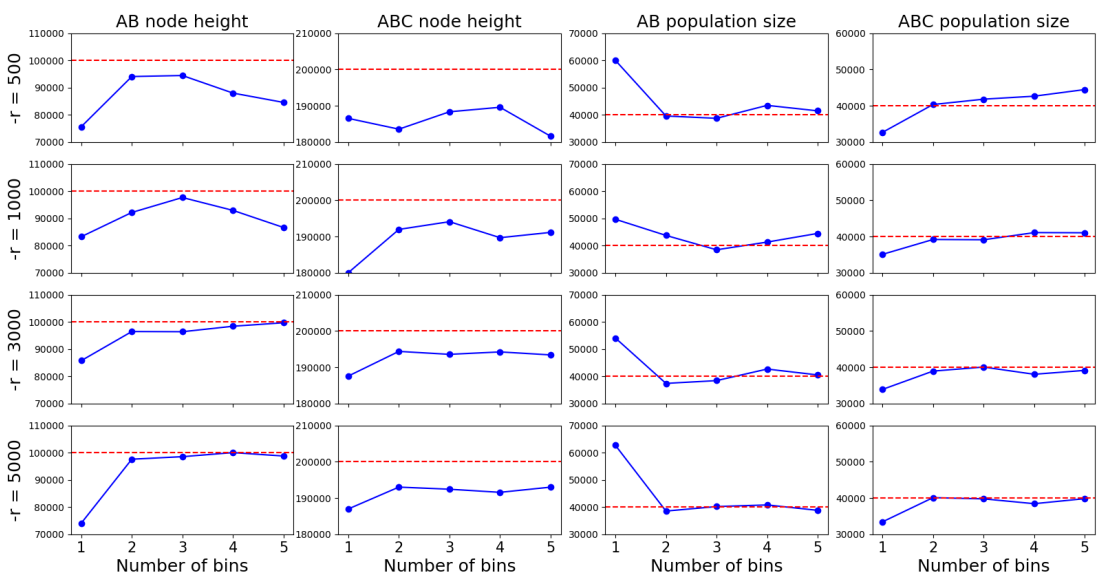

FIG. 9. Results on scenario 2 (internal branch length 1.25). Dashed red lines are true values. Blue lines are inferred MAP values. Rows correspond to different simulation lengths. Columns corresponds to different continuous parameters. x-axes are number of sub-branches ranging from 1 to 5.

the combination for coalHMM construction to find the MAP solution, and inspected the accuracy. In total, we conducted $3 \times 4 \times 5 = 60$ inferences.

Figures 8, 9, and 10 show inference results for scenario 1, 2, and 3, respectively. The message is clearest in figure 10. Looking at each individual plot, for a fixed simulation length, increasing the number of sub-branches increases the accuracy of inference, until it flattens out. But figure 8 suggests that the accuracy does not stay on a plateau after reaching a certain number of sub-branches. Rather, if the simulation length

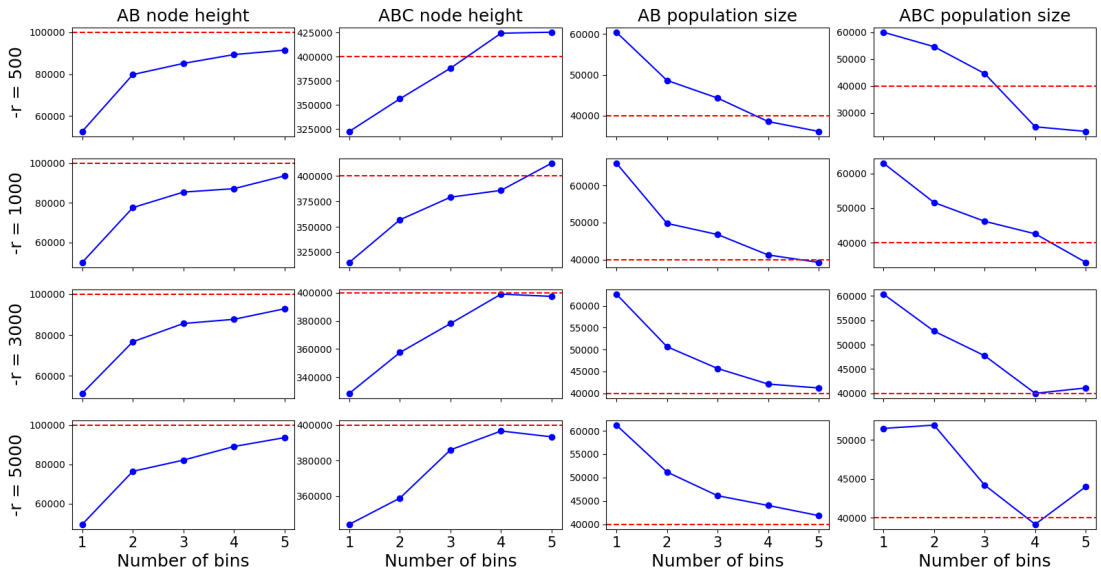

FIG. 10. Results on scenario 3 (internal branch length 3.75). Dashed red lines are true values. Blue lines are inferred MAP values. Rows correspond to different simulation lengths. Columns corresponds to different continuous parameters. x-axes are number of sub-branches ranging from 1 to 5.

is not long enough, increasing the number of sub-branches might decrease inference accuracy. The reason is that the transition rate matrix of a large state space cannot be sufficiently trained from a short length of simulation. Each column of the plots show that generally, for a fixed number of sub-branches, increasing the simulation length increases the accuracy, but the gain is smaller than increasing the number of sub-branches. Simulation length only determines how close is the transition matrix of a coalHMM trained from simulated data to the true transition rate matrix calculated from a strict mathematical model. However, the bias in approximate likelihood comes from a restricted state space, not a poor transition matrix. If the state space is restricted, increasing simulation length does not solve the bias problem because the bias still exists when the transition rate matrix is analytically derived (Dutheil *et al.*, 2009; Mailund *et al.*, 2011). Put simply, increasing the simulation length is not going to address any biases resulting from discretization the state space of coalescent histories, or from using a Markov chain to approximate the ARG.

The appropriate number of sub-branches and simulation length to use depends on the internal branch length of the species tree. For example, using two sub-branches and `-r` = 1000 infers a very good result on scenario 1, but does not work well on scenario 3. Based on the plot, for a short internal branch (about one coalescent unit), two or three sub-branches with a `-r` of around 1000 are sufficient. For a larger branch length (one to three coalescent units), three or four sub-branches with a `-r` of about 3000 would suffice. For branches longer than three coalescent units, more than four sub-branches and a `-r` value higher than 3000 would be needed.

Running time is also a consideration when choosing hyper-parameters. We now study the impact of these hyper-parameters on running time. Almost all the

time taken by the inference attributes to Algorithm 1. The algorithm has two time-consuming sub-procedures: building the coalHMM by simulation, and calculating the likelihood by the Forward algorithm. The running times of the two sub-procedures depend on the number of sub-branches, the length of the simulation, and the scale of the species tree. Figure 11 shows the relationship between time taken by the Forward algorithm and the number of sub-branches for refining the species tree, while keeping all other variables fixed, when evaluating the approximate likelihood of one model. Clearly, the number of sub-branches has a huge impact

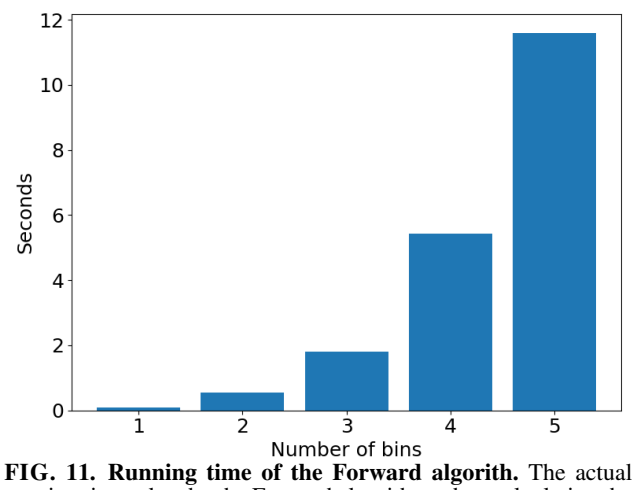

FIG. 11. Running time of the Forward algorith. The actual running time taken by the Forward algorithm when calculating the approximate likelihood of Scenario 3, with different *nb* parameter values in Algorithm 1.

on the running time, as the forward algorithm takes time exponential in the number of sub-branches. The reason is that when more sub-branches are used to refine the species tree, the resulting coalHMM has exponentially more hidden states, which results in significant increase in the Forward algorithm running time since it is quadratic in the HMM size. Figure 12 shows the relationship between the HMM building time, the Forward algorithm running time, and the length of simulation used to build coalHMM, while keeping all other variables fixed. We observe that the time taken for

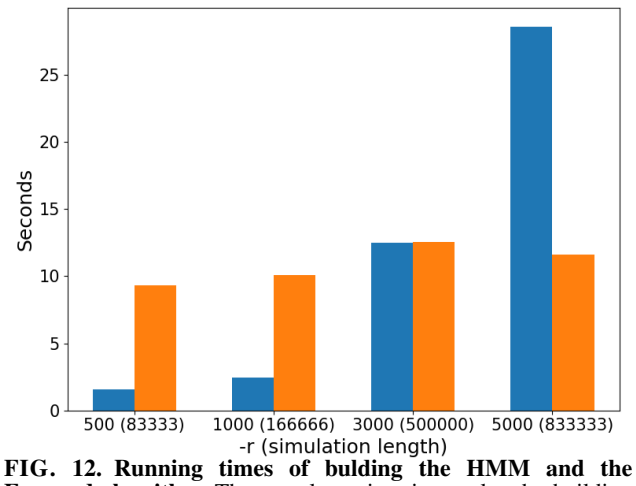

FIG. 12. Running times of bulding the HMM and the Forward algorithm. The actual running times taken by building the HMM by simulation (blue bars) and running the Forward algorithm (orange bars) when calculating the approximate likelihood of Scenario 3, with different `-r` parameter values (the *l* parameter in Algorithm 1).

building the HMM grows exponentially with simulation length, which has to do with the complexity of the underlying simulator, which in our case is `msprime`. The Forward algorithm running time remains roughly unchanged since the number of hidden states of the coalHMM does not depend on simulation length.

Figure 13 shows the relationship between the HMM building running time, the Forward algorithm running time, and the branch length of the species tree on which the approximate likelihood is being evaluated. As before,

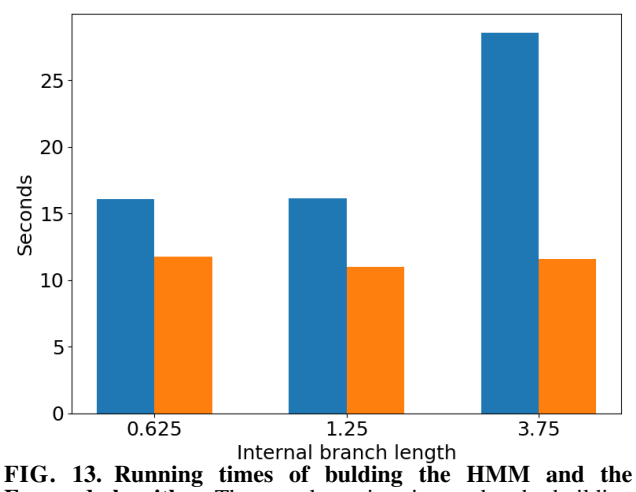

FIG. 13. Running times of bulding the HMM and the Forward algorithm. The actual running times taken by building the HMM by simulation (blue bars) and running the Forward algorithm (orange bars) when calculating the approximate likelihood of Scenarios 1, 2, and 3.

the time taken for building the HMM grows, though not exponentially, with the simulation length, while the Forward algorithm running time remains unchanged. These results highlight the need for developing scalable simulators under the coalescent with recombination, which we identify as a future research direction.

Simulation study on a four-taxa dataset

To demonstrate the scalability of our method, and the efficiency of the divide-and-conquer approach, we simulated a four-taxon sequence dataset on the species tree whose topology is (((A,B),C),D). The demographic parameters are T_{AB} = 100,000, T_{ABC} = 160,000, T_{ABCD} = 450,000. All branches have population size 40,000. The recombination rate is $r = 1.5 \times 10^{-7}$/site/generation. The mutation rate is 1.25×10^{-6}/site/generation. The length of the sequence is 200,000 bp. We first ran a full inference on all taxa, and then ran a divide-and-conquer inference to compare the results.

Full inference on four taxa

The configuration of the coalHMM likelihood kernel for the full inference is $nb = 2$ and `-r` $= 1000$. Black box variational inference is set to run for 200 iterations with 50 samples per iteration. We used uniform prior on node heights and gamma prior on population sizes. The inference took 83.09 hours. 7.62 hours were used to build the coalHMM, and 75.46 hours were used to compute likelihood by the Forward algorithm. It is worth pointing out that, due to the increased number of taxa, the number of hidden states of the coalHMM is significantly increased, resulting in a very long running time for the Forward algorithm, which is quadratic in the number of hidden states. This is a major reason why coalHMM methods are limited to a few taxa. The inference results are shown in Table 2. The true values of most of the parameters are within the 95% credible intervals of the estimates. We used two sub-branches per branch, which, based on our results for three taxa, is suboptimal for accuracy, but inference with only two sub-branches took over 80 hours on a Macbook Pro with 2.4 GHz Intel Core i5 CPU. Results using more sub-branches may be more accurate but we stuck with two sub-branches to limit running time.

Table 2. Result of full inference on the four-taxon dataset. The means and standard deviations of the parameter estimates as well as the true parameter values are shown.

Parameter	Mean	Standard Deviation	True Value
T_{AB}	81770.21479178488	12729.791261759352	100000
T_{ABC}	145883.25694148958	11567.852419117551	160000
T_{ABCD}	422377.26013317733	16254.369793927155	450000
N_{AB}	51343.119852312404	6134.68571483791	40000
N_{ABC}	37787.87413469022	3430.4473740293065	40000
N_{ABCD}	55340.045966958125	6623.18933433634	40000

Divide-and-conquer inference

To reduce the running time and improve the accuracy, we used a divide-and-conquer approach on this dataset. As shown in figure 3 above, two three-taxon inferences were run to cover all the parameters of the four-taxon tree. The ((A,B),C) subtree covers T_{AB}, T_{ABC}, and N_{AB}, while the ((A,C),D) subtree covers T_{ABC}, T_{ABCD}, N_{ABC}, and N_{ABCD}. The parameter T_{ABC} is covered by both datasets.

To infer parameters of the ((A,B),C) tree, we used $nb = 3$ and `-r` $= 1000$ for building the coalHMM. Black box variational inference settings and prior settings are the same as the full inference. The inference took 5.20 hours. 3.01 hours were used to build the coalHMM, and 2.19 hours were used to compute the likelihood by the

Forward algorithm. The parameter estimate results are shown in Table 3.

Table 3. Inference results on the ((A,B),C) tree.

Parameter	Mean	Standard Deviation	True Value
T_{AB}	99025.81222875975	12902.551894013623	100000
T_{ABC}	160494.53209910868	13153.931610727426	160000
N_{AB}	39223.13327409828	7055.689181906025	40000

To infer parameters of the ((A,C),D) tree, we used $nb = 4$ and `-r` $= 1000$ for building the coalHMM. We used one more sub-branch for each branch since the ((A,C),D) tree has a longer internal branch resulting from not sampling taxon B. This further illustrates the flexibility of the divide-and-conquer approach where coalHMM settings can be adjusted according to the specifics of different sub-instances of the problem, saving computational resources overall. The inference took 26.92 hours. 5.41 hours were used to build the coalHMM, and 21.50 hours were used to compute the likelihood by the Forward algorithm. The results are shown in Table 4.

Table 4. Inference results on the ((A,C),D) tree.

Parameter	Mean	Standard Deviation	True Value
T_{ABC}	149775.75047387235	11253.520914091265	160000
T_{ABCD}	445402.1137388789	14241.673883209456	450000
N_{ABC}	43287.90254150772	4514.9114549946	40000
N_{ABCD}	42950.05831226692	4531.445207063785	40000

As the results in both tables demonstrate, all the parameters are recovered with good accuracy in both subtrees, a significant improvement on the direct full inference in Table 2. A another significant improvement is the running time. While the full inference took 83.09 hours, the two subtree inferences only took 5.20 and 26.92 hours, respectively. Since the two subtree inferences are independent, they can be run in parallel. Hence the divide-and-conquer method reduces the runnning time of the inference from over 80 hours to a little bit over 25 hours, while achieving a higher accuracy on all parameters, proving that the divide-and-conquer technique is a promising approach towards large-scale population history inference.

Concluding Remarks

Coalescent methods are a fundamental tool of population genetics and increasingly standard in phylogenetics. As whole genomes become more affordable and widely available, coalHMM is a promising approach for the analysis of large genomic alignments. However, coalHMM methods have been difficult to implement, or to generalize beyond a simple three-taxon ultrametric tree. With our simulation-based approach to constructing the HMM and variational inference of a demographic history's parameters based on it, we achieve accuracy matching previous methods with much greater flexibility. This flexibility is demonstrated by extending coalHMM to four taxa, although in principle all kinds of generalizations are possible as long as they can be simulated. Examples would be non-constant demographic functions such as linear, stepwise or exponential changes in population sizes, and horizontal gene flow such as ancient introgression or secondary contact. Our new approach will be immediately useful to researchers working at the intersection of population genetics and phylogenetics, but also represents a major step forward in terms of applying coalHMMs to biological systems beyond the relatively simple human-chimp-gorilla tree.

Data Availability

The dataset from Hobolth *et al.* (2007) was used in our study for comparison with previous models. We used target 106 (Chromosome 20), which contains 258 thousand base pairs. The data is publicly available. The code is available through the publicly available software package PhyloNet (Than *et al.*, 2008).

Acknowledgments

This work was supported by grants DBI-2030604, CCF-1514177, CCF-1800723, and DMS-1547433 from the National Science Foundation provided to L.N.

References

Ayres, D. L., Darling, A., Zwickl, D. J., Beerli, P., Holder, M. T., Lewis, P. O., Huelsenbeck, J. P., Ronquist, F., Swofford, D. L., Cummings, M. P., *et al.* 2012. Beagle: an application programming interface and high-performance computing library for statistical phylogenetics. *Systematic biology*, 61(1): 170–173.

Baum, L. E. *et al.* 1972. An inequality and associated maximization technique in statistical estimation for probabilistic functions of markov processes. *Inequalities*, 3(1): 1–8.

Baum, L. E., Petrie, T., Soules, G., and Weiss, N. 1970. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The annals of mathematical statistics*, 41(1): 164–171.

Bishop, C. M. 2006. *Pattern recognition and machine learning*. Springer.

Chang, R. and Hancock, J. 1966. On receiver structures for channels having memory. *IEEE Transactions on Information Theory*, 12(4): 463–468.

Cox, D. and Hinkley, D. 1979. *Theoretical Statistics*. Taylor & Francis.

Degnan, J. H. and Salter, L. A. 2005. Gene tree distributions under the coalescent process. *Evolution*, 59: 24–37.

Duchi, J., Hazan, E., and Singer, Y. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7).

Durbin, R., Eddy, S., Krogh, A., and Mitchison, G. 1998. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, Cambridge/London/New York.

Dutheil, J. Y., Ganapathy, G., Hobolth, A., Mailund, T., Uyenoyama, M. K., and Schierup, M. H. 2009. Ancestral population genomics: the coalescent hidden Markov model approach. *Genetics*, 183: 259–274.

Felsenstein, J. 1981. Evolutionary trees from dna sequences: a maximum likelihood approach. *Journal of Molecular Evolution*, 17(6): 368–376.

Fletcher, W. and Yang, Z. 2009. INDELible: A Flexible Simulator of Biological Sequence Evolution. *Molecular Biology and Evolution*, 26(8): 1879–1888.

Hartl, D. L. and Clark, A. G. 2007. *Principles of Population Genetics*. Sinauer, Sunderland, MA, 4th edition.

Hein, J., Schierup, M. H., and Wiuf, C. 2005. *Gene Genealogies, Variation and Evolution*. Oxford University Press, Oxford.

Hobolth, A., Christensen, O., Mailund, T., and Schierup, M. 2007. Genomic relationships and speciation times of human, chimpanzee, and gorilla from a coalescent hidden Markov model. *PLoS Genetics*, 3(2): e7. doi:10.1371/journal.pgen.0030007.

Hudson, R. R. 1990. Gene genealogies and the coalescent process. *Oxford surveys in evolutionary biology*, 7(1): 44.

Hudson, R. R. 2002. Generating samples under a Wright-Fisher neutral model of genetic variation. *Bioinformatics*, 18: 337–338.

Kelleher, J., Etheridge, A. M., and McVean, G. 2016. Efficient coalescent simulation and genealogical analysis for large sample sizes. *PLoS Comput Biol*, 12(5): 1–22.

Kingma, D. P. and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kingman, J. F. C. 1982. The coalescent. *Stochastic Processes and Their Applications*, 13: 235–248.

Mailund, T., Dutheil, J. Y., Hobolth, A., Lunter, G., and Schierup, M. H. 2011. Estimating divergence time and ancestral effective population size of Bornean and Sumatran orangutan subspecies using a coalescent hidden Markov model. *PLoS Genet*, 7(3): e1001319.

McVean, G. A. and Cardin, N. J. 2005. Approximating the coalescent with recombination. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 360(1459): 1387–1393.

Nielsen, R. and Slatkin, M. 2013. *An introduction to population genetics: theory and applications*. Sinauer Associates Sunderland, MA.

Ranganath, R., Gerrish, S., and Blei, D. 2014. Black box variational inference. In *Artificial Intelligence and Statistics*, pages 814–822.

Rogers, J. and Gibbs, R. A. 2014. Comparative primate genomics: emerging patterns of genome content and dynamics. *Nature Reviews Genetics*, 15(5): 347–359.

Ross, S. 1997. *Simulation*. Statistical modeling and decision science. Academic Press.

Siepel, A. 2009. Phylogenomics of primates and their ancestral populations. *Genome Research*, 19: 1929–1941.

Springer, M. S. and Gatesy, J. 2016. The gene tree delusion. *Molecular Phylogenetics and Evolution*, 94: 1–33.

Than, C., Ruths, D., and Nakhleh, L. 2008. PhyloNet: a software package for analyzing and reconstructing reticulate evolutionary relationships. *BMC Bioinformatics*, 9(1): 322.

Tieleman, T. and Hinton, G. 2012. Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning.

Vaughan, T. G., Welch, D., Drummond, A. J., Biggs, P. J., George, T., and French, N. P. 2017. Inferring ancestral recombination graphs from bacterial genomic data. *Genetics*, 205(2): 857–870.

Wakeley, J. 2008. *Coalescent Theory*. Roberts & Company, Greenwood Village, CO.