Evaluation of methods for inference of ancestral recombination graphs

Débora Y. C Brandt*, **Xinzhu Wei*** **and Rasmus Nielsen***,[3]

*Department of Integrative Biology and Department of Statistics, University of California Berkeley, Berkeley, CA 94720, USA

ABSTRACT The ancestral recombination graph (ARG) is a model that describes the genealogy of samples of DNA sequences, keeping full information of inferred coalescence, mutation, and recombination events. Recent methods have made impressive progress towards scalably estimating whole-genome genealogies. In addition to inferring the ARG, some of these methods can also provide ARGs sampled from a defined posterior distribution. Obtaining good samples of ARGs is crucial for quantifying statistical uncertainty and for the downstream processing of ARGs to estimate parameters such as effective population size, mutation rate and age of variants. Here, we use simulations to benchmark three ARG inference programs: ARGweaver, Relate and tsdate. We use neutral coalescent simulations to 1) compare the true coalescence times to the inferred times at each locus; 2) compare the distribution of coalescence times across all loci to the expected exponential distribution; 3) evaluate whether the sampled coalescence times have the properties expected of a valid posterior distribution. We found that Relate and ARGweaver estimate coalescence times at each locus that are closer to the true values than tsdate. However, all methods have biases towards overestimating small coalescence times and underestimating large ones. Bias in these methods is unsurprising since they are all Bayesian, and thus are not designed to yield unbiased estimates. Finally, ARGweaver provides a distribution of samples that is closer to the correct posterior distribution than Relate, but this accuracy comes at a substantial computational cost. We conclude that the best choice of method will depend on the number of input sequences and on the specifics of the downstream analyses.

KEYWORDS Ancestral recombination graph; ARGweaver; Relate; tsdate; simulation; calibration

[OBS: subtitles here are just for our reference, I don't plan to keep them all in the final version – Debora]

The ancestral recombination graph (ARG) is a model that represents the evolutionary history of a sample of DNA sequences as a set of coalescence, recombination and mutation events (Griffiths and Marjoram 1997). At each given site, the genealogy can be described by the coalescent model (Kingman 1982), but because recombination breaks loci apart, the local genealogies can differ between sites. Hudson (1983) first proposed a model to combine recombination and coalescence to fully describe the genealogy of a set of samples.

doi: 10.1534/genetics.XXX.XXXXXX
Manuscript compiled: Tuesday 10th August, 2021

[3]University of California, Berkeley Department of Integrative Biology
3040 Valley Life Sciences Building #3140 Berkeley, CA 94720-3140.
rasmus_nielsen@berkeley.edu

Motivation - quantify ARG uncertainty

The ARG can also be defined as a data structure. In this context, it is a lossless representation of a sample of DNA sequences. As such, the ARG fully represents all the information in a sample of DNA sequences and therefore it is necessarily more informative than any summary statistics traditionally used to learn about evolutionary processes (*e.g.* F_{ST}, π, Tajima's D, or haplotype based ones such as EHH, etc). In other words, all traditional summary statistics can be calculated from the ARG.

an you can do much more

One difficulty, however, is that it is impossible to know the true ARG underlying a sample, since many ARGs can be compatible with a given sample of DNA sequences. Thus, if one is interested in modelling evolutionary processes using ARGs, it is desirable to quantify the uncertainty around the inferred ARG by getting samples of ARGs according to their posterior probability, given some data. Another way to see this from a Bayesian perspective is that the true ARG underlying a sample of DNA sequences is only one instance of a random variable that was sampled from a distribution that depends on evolutionary parameters of interest (*e.g.* effective population size, selection, mutation and recombination rates).

Inferring full ARGs, and quantifying inference uncertainty by getting samples from their posterior distribution is a challenging problem theoretically and computationally. It requires navigating a high dimensional distribution of ARGs, which are themselves a complicated data structure. For this reason, inferring ARGs and sampling from their posterior distribution seemed like a nearly impossible endeavour some years ago, but important theoretical and methodological developments now allow us to do so. Today, many methods to estimate the ARG from a sample of DNA sequences are available. However, thus far only ARGweaver (Rasmussen *et al.* 2014) and Relate (Speidel *et al.* 2019) are able to output samples of ARGs proportionally to their posterior probability, thus accounting for inference uncertainty.

Accurate sampling from the posterior distribution is especially relevant for downstream methods that rely on importance sampling to infer evolutionary parameters from ARGs. In essence, these methods weight parameter inference under each sampled ARG by the ARG probability, and therefore require that the samples of ARGs accurately reflect their probability distribution. This type of methods can be used to infer population size history (Speidel *et al.* 2019), selection (Stern *et al.* 2019), migration, mutation and recombination rates .

more citations for methods that use ARGs as a model for evolutionary inference

Here, we first summarize the most important milestones for the development of methods for ARG inference and introduce their theoretical foundations relevant for interpreting our results. Next, we use simulations to evaluate ARG inference in three programs: ARGweaver(Rasmussen *et al.* 2014), Relate (Speidel *et al.* 2019) and tsdate (Wohns *et al.* 2021). tsdate is a more recent method that does not allow ARG sampling, but is a promising framework for it. We evaluate these programs by comparing the coalescence times between simulated and inferred ARGs.

Milestones for the development of ARG inference methods

SMC and SMC' models Within the coalescent framework, the most straightforward way to include recombination is to consider it as a process of splitting lineages as one moves backwards in time (Hudson 1983; Griffiths and Marjoram 1997). One key insight for the development of methods based on the ARG was the idea that the coalescent process with recombination can also be simulated as a process that happens along a sequence (Wiuf and Hein 1999), as opposed to the formulation as a process that occurs backwards in time. In that formulation, the ARG is constructed as a sequence of local coalescent trees along a genome, separated by recombination events. At each recombination breakpoint, a new tree is formed from the immediately preceding tree. To form the next tree, first one of the branches in the current tree is detached. Next, a branch from any of the previous trees in the sequence is randomly chosen. Finally, the detached branch coalesces to it.

To allow for more computational efficiency in simulating this process, it has been approximated as a Markovian process (*i.e.* a process where the next state only depends on the current state). In this approximation, when a lineage is detached from a tree at a recombination event, it can only coalesce back to one of the other lineages present at the current tree (McVean and Cardin 2005). This model is called the Sequentially Markovian Coalescent (SMC). The SMC model has later been improved to better approximate the full coalescent with recombination. In this improved version, known as SMC′, when a recombination happens, the detached lineage can coalesce to any branch in the current tree, including the one it was detached from. This means that some recombination events in this model do not generate a different local coalescent tree. This apparently slight modification has been shown to significantly improve the model in terms of approximation of the full coalescent (Marjoram and Wall 2006; Wilton *et al.* 2015).

Li and Stephens copy model Another model that approximated the coalescent with recombination making it more computationally tractable is the Li and Stephens haplotype copying model (Li and Stephens 2003). In this model, the genealogy is generated by a copying process in which each new haplotype is formed by copying chunks of other haplotypes present in the sample. Errors in the copy are introduced according to the mutation rate, and changes on which haplotype to copy from are made according to the recombination rate. Similar to the SMC and SMC′ models, the Li and Stephens model is not only more computationally tractable for simulating data and performing MCMC, but also reproduces relevant properties of the full model and depends on the same parameters such as recombination and mutation rates.

Methods I'm comparing

Altogether, the formulation of the coalescent with recombination as a spatial process along a sequence (Wiuf and Hein 1999), as well its simplification as a Markovian process in SMC (McVean and Cardin 2005) and SMC′ (Marjoram and Wall 2006) and as a copying process in the Li and Stephens model (Li and Stephens 2003) paved the way to methods that are able to estimate ARGs for large sample sizes. Notably, ARGweaver (Rasmussen *et al.* 2014) is based on the SMC model, and Relate (Speidel *et al.* 2019) and tsdate (Kelleher *et al.* 2019; Wohns *et al.* 2021) are based on the Li and Stephens model.

ARGweaver and time discretization + weaving In ARGweaver, two crucial novelties further make the problem of ARG sampling tractable. First, discretization of time (such that all recombination and coalescence events are only allowed to happen at a discrete set of timepoints) decreases the size of the state space ARGs, which is otherwise infinite. Second, ARGweaver uses the lineage threading approach, which is a clever way to navigate the large state space of ARGs. With these key implementations, ARGweaver uses MCMC under the SMC or SMC′ model to generate samples of ARGs from the posterior distribution.

Relate and Li and Stephens 2003 model Relate tackles the problem of ARG inference differently, dividing it in 2 steps. First, the Li and Stephens (Li and Stephens 2003) haplotype copying model is used to infer local tree topologies. Next, it uses MCMC under a coalescent prior to infer coalescence times on those local trees. Relate is able to output samples of coalescence times from the posterior distribution using this MCMC approach, but it does not generate different samples of ARG topologies.

tsinfer, tsdate and the tree sequence framework Tsdate (Wohns *et al.* 2021) is a method that dates tree sequences inferred by tsinfer (Kelleher *et al.* 2019). Like Relate, tsinfer is also based on the copying process from

Li and Stephens (Li and Stephens 2003). Another key innovation of tsinfer is using the highly efficient tree sequence data structure to store sequence data and compute statistics from them. tsinfer performs inference in two steps. First, it recreates ancestral haplotypes based on allele sharing between samples. Next, it uses an HMM to infer the closest matches between ancestral haplotypes and the sampled haplotypes, using an ancestral copying process based on the Li and Stephens model (Li and Stephens 2003). This step results in the tree topology. Next, the local tree topologies inferred by tsinfer can be dated by tsdate. tsdate uses a conditional coalescent prior, where the traditional coalescent is conditioned on the number of descendants of each node on a local tree. Like ARGweaver, tsdate also uses a discretization of time to make computations more efficient.

What we are doing here

Here, we used simulations to benchmark ARG inference in ARGweaver (Rasmussen *et al.* 2014), Relate (Speidel *et al.* 2019) and tsdate (Wohns *et al.* 2021). We focus mainly on ARGweaver and Relate because they inform about uncertainty in inference by allowing the user to output multiple samples from the posterior distribution. Sampling from the posterior is not currently implemented in tsdate, but we include it in this evaluation because it is a promising framework for very fast ARG inference and will likely include an option to output samples from the posterior distribution of ARGs soon.

We ran coalescent simulations on msprime (Kelleher *et al.* 2016) and we compared the true (simulated) ARGs to the ARGs sampled by ARGweaver and Relate and inferred by tsinfer/tsdate. We compared the ARGs based on their pairwise coalescence times, using three different types of evaluation. First, we evaluate the true pairwise coalescence time at each site to the inferred time. Second, we compared the overall distribution of pairwise coalescence times across all sites and all MCMC samples to the expected distribution. Since data was simulated under the coalescent with recombination, and Bayesian inference was done using close approximations of the coalescent with recombination to compute the likelihood, the expected posterior distribution of coalescent times is the same as the prior, *i.e.* exponential with rate 1. Third, we used simulation based calibration (SBC) (Cook *et al.* 2006; Talts *et al.* 2020) to evaluate if the posterior distributions sampled by ARGweaver and Relate are well calibrated.

Methods

Simulations with msprime

We simulated tree sequences and SNP data with msprime version 0.7.4 (Kelleher *et al.* 2016). Unless otherwise noted, simulations were done under the standard neutral coalescent (Hudson model in msprime) and using the following parameters: 4 diploid samples (*i.e.* 8 haplotypes), effective population size of 10,000 diploids ($2N = 20,000$), mutation rate and recombination rate of $2 * 10^{-8}$ per base pair per generation and a total sequence length of 100Mb.

We varied these standard simulation scenarios in several ways: using SMC and SMC′ models, different numbers of samples (4, 16, 32 and 80 haplotypes) and 10 times increased mutation to recombination ratio (both by increasing mutation rate to $2 * 10^{-7}$, and by decreasing recombination rate to $2 * 10^{-9}$).

We ran msprime simulations using the python Application Programming Interface (API), and stored data in VCF and tree sequence format.

Next, we describe how we ran ARGweaver, Relate and tsinfer/tsdate under the simulated parameters described above. In ARGweaver and Relate, we ran 1000 MCMC iterations after burn-in and output every 10th sample, for a total of 100 MCMC samples from the posterior.

ARGweaver

VCF files from msprime were converted to ARGweaver sites format using a custom python script. We ran ARGweaver's *arg-*

sample program to sample ARGs. This was done in parallel on twenty 5Mb segments, using the *–region* option. We used the same values used in the msprime simulations (*–mutrate* and *–recombrate 2e-8* and *–popsize 10000*) and except where otherwise noted, we ran ARGweaver using the SMC′ model (*–smcprime* option). We ran ARGweaver for 1200 or 2200 iterations (*–iters*) depending on how long it took to converge. Assessment of convergence is described below. We extracted 100 MCMC samples from every 10th iteration among the last 1000 iterations (default *–sample-step 10*).

We extracted all pairwise coalescence times with the program *arg-summarize* using options *–tmrca* and *–subset*, and we used bedops (version 2.4.35 (Neph *et al.* 2012)) to match the times sampled by ARGweaver to the simulated ones at each sequence segment. Finally, we used a custom Python script to calculate the ranks of simulated pairwise coalescence times on ARGweaver MCMC samples per site.

Time discretization

In ARGweaver, time is discretized such that recombination and coalescence events are only allowed to happen at a user-defined number of time points, K (default value is 20) (Rasmussen *et al.* 2014). These time points s_j (for $0 <= j <= K-1$) are given by the function

$$s_j = g(j) = \frac{1}{\delta}\left\{exp\left[\frac{j}{K-1}log(1+\delta s_{K-1})\right]-1\right\} \tag{1}$$

where δ is a parameter determining the degree of clustering of points in recent times. Small values of δ leads to a distribution of points that is closer to uniform between 0 and s_{K-1}, and higher values increase the density of points at recent times (default value is 0.01) (Hubisz and Siepel 2020). Equation 1 determines that s_0 is always 0, and s_{K-1} (or s_{max}) is user defined in ARGweaver, with the parameter *–maxtime* (default value is 200,000).

Rounding of continuous times into these K time points is given by defining breakpoints between them, such that the breakpoint between times s_j and s_{j+1} is $s_{j+\frac{1}{2}} = g(j+\frac{1}{2})$. All continuous values between $s_{j-\frac{1}{2}}$ and $s_{j+\frac{1}{2}}$ are assigned the value s_j. We note that for the first and last intervals, the values assigned (s_0 and s_{K-1}) don't correspond to a midpoint in the time interval, but rather to its minimum ($s_0 = 0$) or maximum ($s_{K-1} = s_{max}$)

Here, we use the same time discretization bins defined by ARGweaver breakpoints ($s_{j+\frac{1}{2}}$) as described above. However, we change the value assigned to times in these bins: instead of using s_j, we define t_j so that half of the probability mass on time bin j is to the left of t_j, and half is to the right of it. This step is relevant for the simulation-based calibration (see below), where we take the rank of true (simulated) coalescence times relative to the values sampled by ARGweaver. With s_j, any coalescence times in the first or last time interval from ARGweaver would not be represented by a midpoint as they are in the other intervals. We correct for that by using t_j, so that all time intervals are comparable.

Relate does not use time discretization, and tsdate uses a different discretization. Here, we apply the ARGweaver time discretization scheme to compare results among these methods.

Relate

VCF files generated with msprime were converted to Relate haps and sample files using *RelateFileFormats –mode CovertFromVcf* and Relate's *PrepareInputFiles* script. We ran Relate mode All with the same mutation rate (*-m 2e-8*) and effective population size (*-N 20000*) used in the msprime simulations, as well as a recombination map with constant recombination rate along the genome, with the same rate used in msprime (*2e-8*).

We used Relate's *SampleBranchLengths* program to get 1000 MCMC samples of coalescence times for the local trees inferred in the previous step, with anc/mut output format (*–format a*). Similarly to ARGweaver, we also performed this step in 20 sequence segments of 5Mb, and we thinned the results to keep only every 10th MCMC sample. Finally, we concatenated *anc* and *mut* files from all chunks and used a custom c++ script to obtain all pairwise coalescence times for the local trees and branch lengths inferred in the previous step, and to calculate the ranks of true pairwise coalescence times relative to the 100 MCMC samples.

tsinfer and tsdate

VCF files generated by msprime were provided as input to the python API using *cyvcf2.VCF* and converted to tsinfer *samples* input object using the *add_diploid_sites* function described in the tsinfer tutorial (https://tsinfer.readthedocs.io/en/latest/tutorial.html#reading-a-vcf). Genealogies were inferred with tsinfer (version 0.2.0 (Kelleher *et al.* 2019)) with default settings and dated with tsdate (version 0.1.3 (Wohns *et al.* 2021)) using the same parameter values as in the simulations (*Ne=10000*, *mutation_rate=2e-8*), with a prior grid with 20 timepoints.

Pairwise coalescence times were extracted from the tree sequences using functions *get_interval()* and *tmrca()* from tskit (version 0.3.4 (Kelleher *et al.* 2018)), and coalescence times at each site were matched to the simulated ones using bedops (Neph *et al.* 2012).

Convergence of MCMC chains

ARGweaver In ARGweaver, we monitored convergence by plotting likelihood per iteration and autocorrelation between consecutive samples. We chose a burn in period of 200 iterations for most scenarios, which was enough for the convergence of most ARGweaver runs, based on the visual inspection of autocorrelation and traces plots. Traces of the output statistics and autocorrelation plots from both programs are provided in Supplemental Materials .

include argw convergence in supp

Relate Relate does not output MCMC traces. Therefore, we evaluated convergence based on the repeatability of our results when running Relate multiple times for the same data, as well as running one longer chain (10 times as long, with 10 thousand iterations). Since results didn't change between 1000 or 10,000 iterations, and neither among five repeated runs with 1000 iterations, we chose to use 1000 MCMC iterations, and, similarly to ARGweaver, we thinned down samples, taking every tenth.

include convergence diagnostic stats analyses (CODA)

could explain here that Relate already converges in the topology inference part and burnin is not needed to sample branch lengths?

Point estimates of pairwise coalescence times

We estimated pairwise coalescence times from the MCMC samples from Relate and ARGweaver by taking the mean value of 100 samples at each site. Since tsdate does not have an option to output multiple samples of node times, we use its single estimate of pairwise coalescence times directly.

Mean squared error of point estimates was calculated per bin of size 0.1 of the simulated coalescence times (in units if 2N generations). In each bin, we calculated the mean of the point estimates at all sites, and the mean squared error was calculated relative to that mean.

Simulation-based calibration

To go one step further than comparing MCMC point estimates to the true simulated values, we have used a method proposed by Cook *et al.* (2006) and Talts *et al.* (2020) to assess if Bayesian methods are sampling well from the true posterior distribution. Cook *et al.* (2006) proposed testing this using simulations. The essential idea is that when you simulate data according to parameters sampled from the prior, and use the same data sampling distribution in the MCMC inference, then the posterior distribution should recover the prior distribution. In other words, under this condition, the posterior distribution is well calibrated by construction. Therefore, we can evaluate MCMC software by checking if their posterior distributions are well calibrated.

More specifcally, Cook *et al.* (2006) and Talts *et al.* (2020) proposed that we i) generate samples of parameter values θ (in our case, coalescence times), from their prior distribution $P(\theta)$ (in our case, the prior comes from the full coalescent with recombination process) and ii) generate data y (sequences) from the data sampling distribution $P(y|\theta)$, which in our case comes from adding mutations according to a certain rate on the genealogies from the coalescent with recombination. Here, all these steps were done by the simulation program msprime. Next, the simulated data and parameters are provided to the MCMC sampler (ARGweaver and Relate) for inference of the posterior distribution using the same data generating model. In our case, ARGweaver and Relate use approximations of the model (coalescent with recombination) used in the data simulations: the SMC′ and the Li and Stephens models, respectively. The final step for simulation based calibration is calculating the rank of each simulated parameter value on the values sampled from the posterior by MCMC. In our case, we took 100 MCMC samples from ARGweaver and Relate thus our ranks take values from 0 to 100. Cook *et al.* (2006) and Talts *et al.* (2020) showed that if the MCMC methods are sampling from a well calibrated posterior as expected, these ranks will follow a uniform distribution.

Deviations from the uniform distribution of ranks show how the posterior is biased. For example, an excess of low and high ranks indicate that the inferred posterior distribution is underdispersed relative to the true posterior.

Results

Comparison of simulated to estimated coalescence time per site

We compared coalescence times estimated by ARGweaver, Relate and tsdate to the true values known from msprime simulations. In all three methods, estimates of coalescence time per site are biased and have low precision (Fig. 1A, 2A, 3A and S2).

Small values of coalescence times are generally overestimated, while large values tend to be underestimated (Fig S2). In ARGweaver, the first issue (overestimation of small coalescence times) seems to be more prevalent (Fig. 1A, mean square error (MSE)=0.397). In Relate, there are more sites where coalescence times are underestimated (Fig. 2A, MSE=0.845). In tsdate, point estimates are apparently bounded to a narrow range (Fig. 3A, MSE=0.812).

For ARGweaver and Relate, these point estimates of coalescence times are the means of the samples from the posterior. As Bayesian estimates, they are not designed to be unbiased. The concept of bias itself is based on the idea that there is a true parameter value. In a Bayesian point of view, there is no one true parameter value, but a prior distribution. Therefore, it's important to keep in mind that a Bayesian estimator given by the mean of the posterior distribution has been "weighted" by the prior distribution.

needs better wording

In a Bayesian context, a given ARG (G_0) can be seen as one instance of a random variable, and one could be interested in the parameters (θ) of the distribution from which that ARG was drawn ($P(G|\theta)$). For example, one could be interested in effective population size, which is a parameter that affects the distribution of coalescence times in ARGs. In that scenario, bias in certain estimates would not be necessarily problematic if the correct distribution of coalescence times was recovered. To sum up, a Bayesian method that generates biased estimates of some parameter values may still accurately infer the posterior distribution. In the next sections, we evaluate the distribution of coalescence times from ARGweaver, Relate and tsdate.

Posterior distribution of coalescence times

We simulated data under the standard coalescent model, therefore the distribution of pairwise coalescence times (in units of $2N$ generations, where N is the diploid effective population size) follows an exponential distribution with rate parameter 1 (Fig. S3).

We compared the expected exponential distribution of coalescence times to the observed distribution of coalescence times across all sites inferred by ARGweaver, Relate and tsdate (Fig. 1D, 2D and 3D). For ARGweaver and Relate, we output 100

MCMC samples from the posterior distribution, and and plot the distribution of pairwise coalescence times across all sites and MCMC samples.

I could also take the mean or median of all MCMC samples for each site

To facilitate visual comparison of the distributions between methods, we discretized Relate and tsdate coalescence times into the same bins as ARGweaver (Fig. 2D and 3D, see distributions without discretization in Fig. S4 and see Methods for a description of ARGweaver time discretization). Because time discretization breakpoints are regularly spaced in a log scale, we use a log scale on the x axis for better visualization.

Distributions of coalescence times from ARGweaver (Fig. 1D) and Relate (Fig. 2D) show an excess at low values, when compared to the expected exponential distribution. However, that bias is more pronounced in Relate than ARGweaver. In tsdate, the distribution is truncated at ??? and there is an excess of coalescence times around the mean (Fig. 3D). We note that the plots from ARGweaver and Relate are not directly comparable to those of tsdate, since there are 100 coalescence time samples at each site from the former two programs, but only one from tsdate.

max coal time inferred by tsdate

Simulation based calibration

In this section, we use simulation-based calibration to evaluate whether ARGweaver and Relate are generating samples from a valid posterior distribution of ARGs (see Methods). To that end, we simulated coalescence times at multiple sites following the standard coalescent prior distribution, and we generated 100 MCMC samples from the posterior distribution using both ARGweaver and Relate. Finally, we analyse the distribution of the ranks of the simulated coalescence times relative to the 100 sampled values, at each site.

In the previous section, we showed that the posterior distributions of ARGweaver and Relate are similar to the theoretically expected exponential. However, in that analysis we have not evaluated the distribution of MCMC samples relative to each simulated value. The results of simulation based calibration are informative about that, and can reveal if the posterior distribution is well calibrated around each simulated value.

The distribution of ranks from ARGweaver (Fig. 1G) is closer to uniform than that of Relate (Fig. 2G). Both show an excess of low and high ranks, but this excess is more pronounced in Relate. The excess of low and high ranks indicates that the sampled posterior distribution is underdispersed (Talts *et al.* 2020).

Increased mutation to recombination ratio

When inferring an ARG from sequence data, what provides information for inference is the set of mutations that cause variable sites in the sequence data. The lower the recombination rate, the longer will be the span of local trees, and more mutations will be available to provide information about each local tree. More generally, an increased mutation to recombination ratio is expected to increase the amount of information available to infer the ARG. In our standard simulations presented so far, the mutation to recombination ratio is one ($\mu = \rho = 2 * 10^{-8}$), which is similar to what is observed in humans .

REF

Here, we increased the simulated mutation to recombination ratio to 10, both by decreasing the recombination rate (ρ) ten fold and also by increasing the mutation rate (μ) ten fold. We expected that these scenarios would improve inference of ARGs. However, for Relate the coalescence times distribution (Fig. 2E,F) showed larger skew towards lower values with increased $\mu/\rho = 10$ than with $\mu/\rho = 1$ (Fig 2D). The simulation based calibration remained qualitatively similar (Fig. 2G-I).

The results from ARGweaver with $\mu/\rho = 10$ were more surprising, with the simulation based calibration showing a more pronounced underdispersion of the posterior distribution (Fig. 1H,I). The overall distribution of coalescence times, however, showed little change (Fig. 1E,F), and point estimates improved with increased mutation to recombination ratio (Fig. 1B,C).

In tsdate, both the point estimates and

posterior distribution improved with increasing mutation to recombiation ratio (Fig. 3), particularly with increased mutation rate, as expected.

Number of samples

include 200 samples for Relate and tsdate

Next, we evaluate ARG inference with simulations with larger sample sizes. Our standard sample size used so far was 8 haplotypes, and here we change it to 4, 16, 32 and 80.

With a smaller sample size (n=4 haplotypes), the coalescence time distribution from Relate showed an excess around the mean value (coalescence time of 1) (Fig 5F). With increasing sample sizes, it was skewed towards lower values) (Fig 5G-J). Calibration of the posterior distribution remains similar with increasing sample sizes (Fig. 5K-O).

For ARGweaver, increasing sample sizes had the effect of slightly increasing the underdispersion of the posterior distribution (Fig. 4F-J,K-O). This could be caused by an MCMC mixing problem, causing the ARGweaver MCMC to not converge well with an increasing number of samples, due to an increasing number of states to explore by the algorithm.

plots with convergence of argweaver with n>8

Both the point estimates and posterior distribution of tsdate improved with increasing sample sizes (Fig. 6). This is expected from inference using the Li and Stephens copy algorithm (Li and Stephens 2003), which tends to .

Run time

Standard simulations

- ARGweaver: 640.8 total hours for all chunks (34.5h per 5Mb chunk)
- Relate: 17 total hours
- tsinfer+tsdate: 4.9min (not comparable because it is not doing MCMC sampling)

figure out a reasonably way to compare those numbers

Discussion

ARG inference is both an interesting theoretical problem per se and also a useful tool for downstream inference of particular aspects of evolutionary history, such as natural selection or population size changes. It is a hard computational problem, which requires implementing approximations of the full coalescent with recombination model. We have compared methods that use different approaches to this problem, and evaluated their accuracy using simulated data and comparisons of three aspects: 1) individual point estimates of each pairwise coalescence time; 2) the overall distribution of coalescence times across all sites; 3) the sampling of ARGs from the posterior distribution.

We found that there is a strong speed-accuracy trade-off in ARG inference. ARGweaver is performing best in our three tests: point estimates, the overall distribution of coalescence times, and the quality of sampling from the posterior. Importantly, it is also the only method that we compared that resamples both topologies and node times. This probably leads to a better exploration of ARG space and the reason why it provides better samples from the posterior. On the other hand, is also contributes to making ARGweaver much slower than the other methods and not scalable for genomewide inference of 100 or more samples, taking X days for n=32 and l=100Mb.

Despite not performing as well as ARGweaver in our evaluation criteria, Relate and tsdate can be good enough for comparisons of average trees among different regions in the genome. Additionally, we showed that their inference is better with more samples, particularly in tsdate (Fig. 5,6). Because these programs are fast enough even for thousands of samples, this is an ideal scenario to use those programs.

simulations with 200 samples

Increasing the mutation to recombination ratio in simulations improved point estimates from ARGweaver, but did not improve posterior calibration (Fig. 1). This lack of improvement of the posterior sampling is not explained by lack of convergence (Fig. S5-S10).

not sure how to discuss this

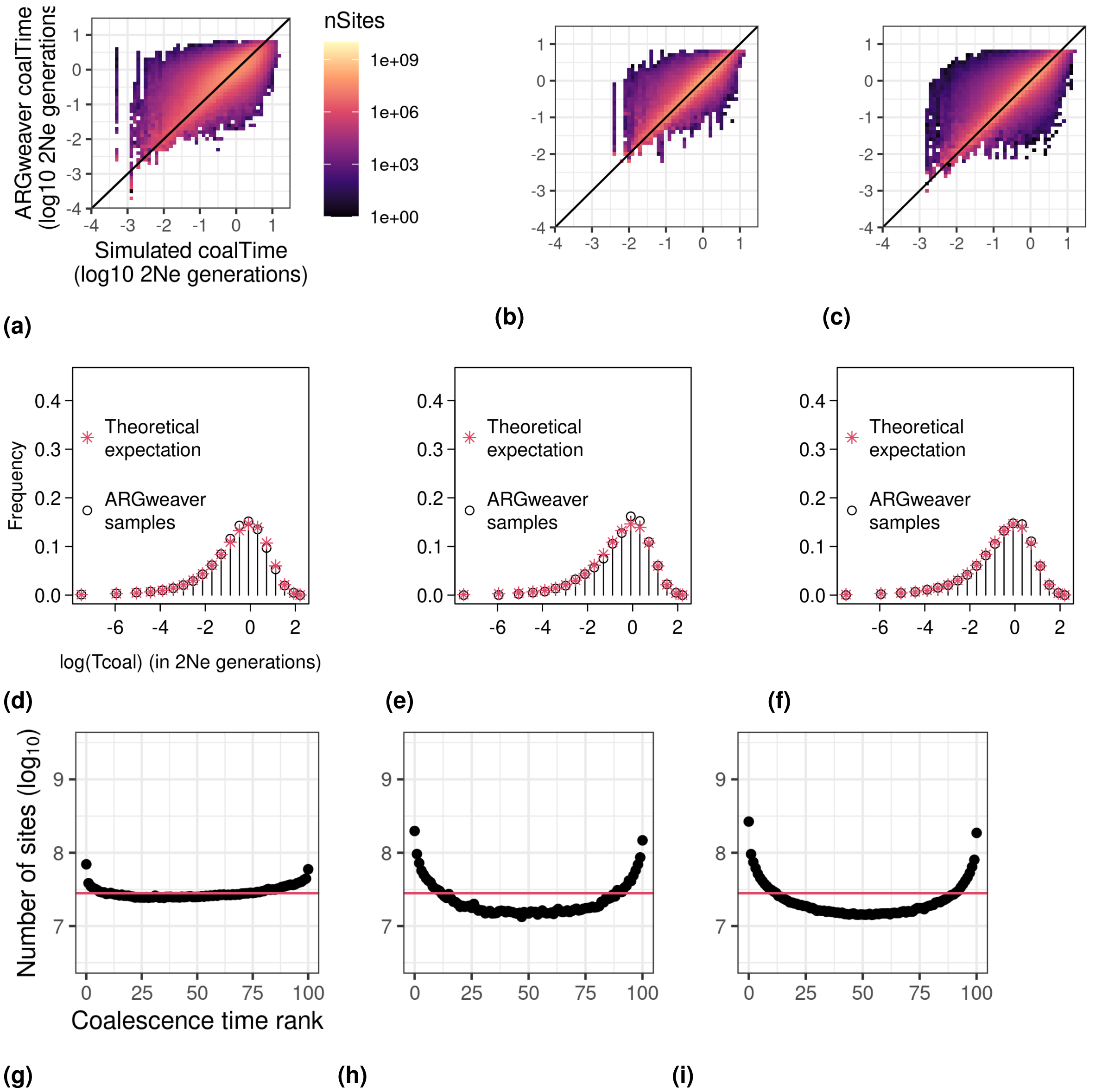


Figure 1 ARGweaver increased mutation to recombination rate. Left column (A,D,G): $\mu = \rho = 2 * 10^{-8}$; middle column (B,E,H): $\mu/\rho = 10$ decreasing recombination rate ($\rho = 2 * 10^{-9}$); right column (C,F,I): $\mu/\rho = 10$ increasing mutation rate ($\mu = 2 * 10^{-7}$). (A-C) point estimates (mean of 100 MCMC iterations). Note that axes are in log scale. Mean squared error: A) 0.397 B) 0.120 C) 0.117. See Fig. S1 for this data plotted in linear axes. Diagonal line shows x=y, points show the mean inferred coalescence time across all sites within the same true coalescence time bin. (D-F) Distribution of coalescence times inferred by ARGweaver, combining all sites and MCMC samples. (G-I) Counts of ranks from simulation-based calibration. Horizontal line shows expected uniform distribution.

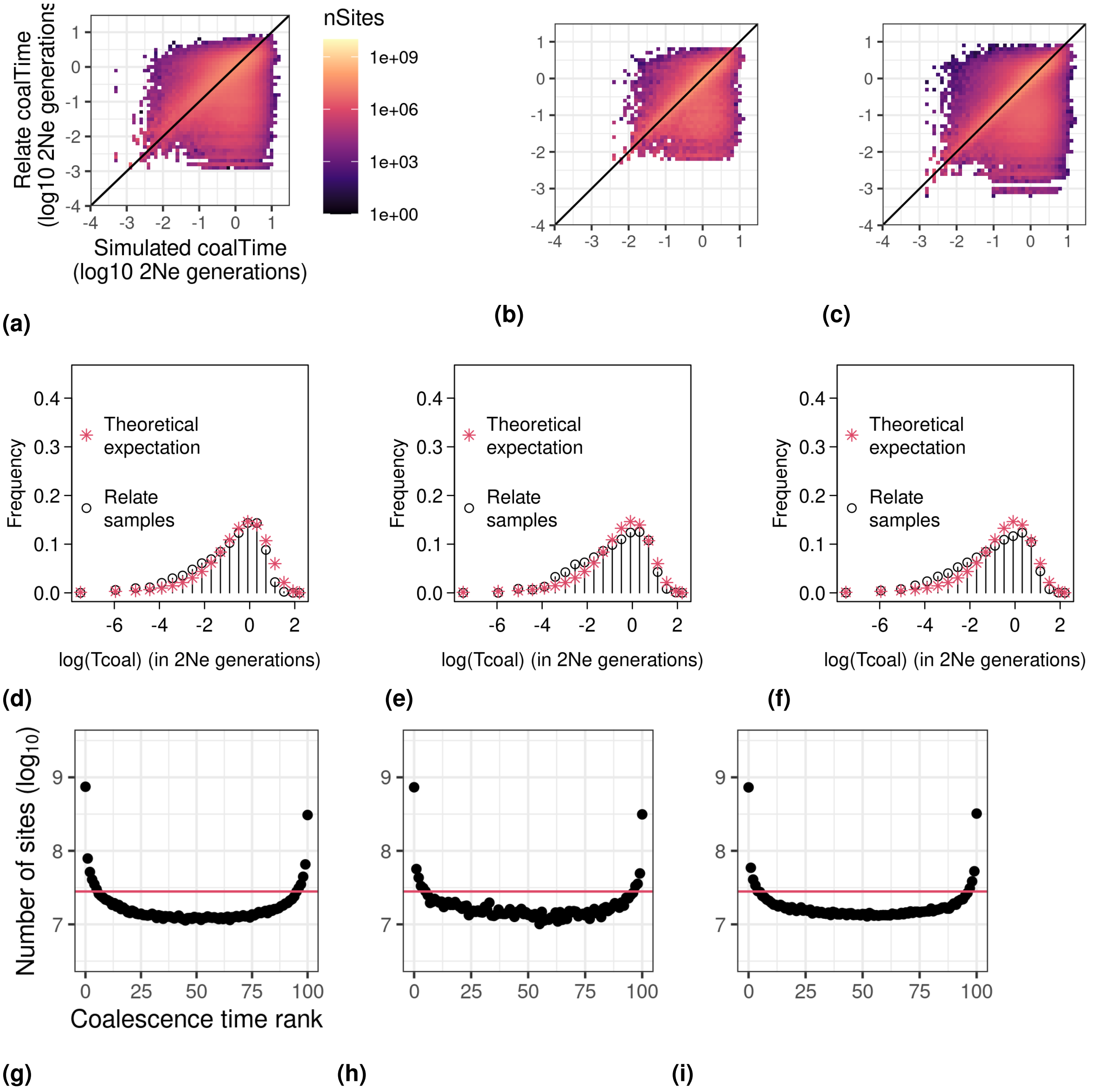


Figure 2 Relate increased mutation to recombination rate. Left column: $\mu = \rho = 2 * 10^{-8}$; middle column: $\mu/\rho = 10$ decreasing recombination rate ($\rho = 2 * 10^{-9}$); right column: $\mu/\rho = 10$ increasing mutation rate ($\mu = 2 * 10^{-7}$). (A-C) point estimates (mean of 100 MCMC iterations). Mean squared error calculated per simulated bin: A) 0.845, B) 0.629, C) 0.600. (D-F) Distribution of coalescence times inferred by Relate. Plots D shows the same data as in Figure S4A, using different binning. (G-I) Counts of ranks from simulation-based calibration. Horizontal line shows expected uniform distribution.

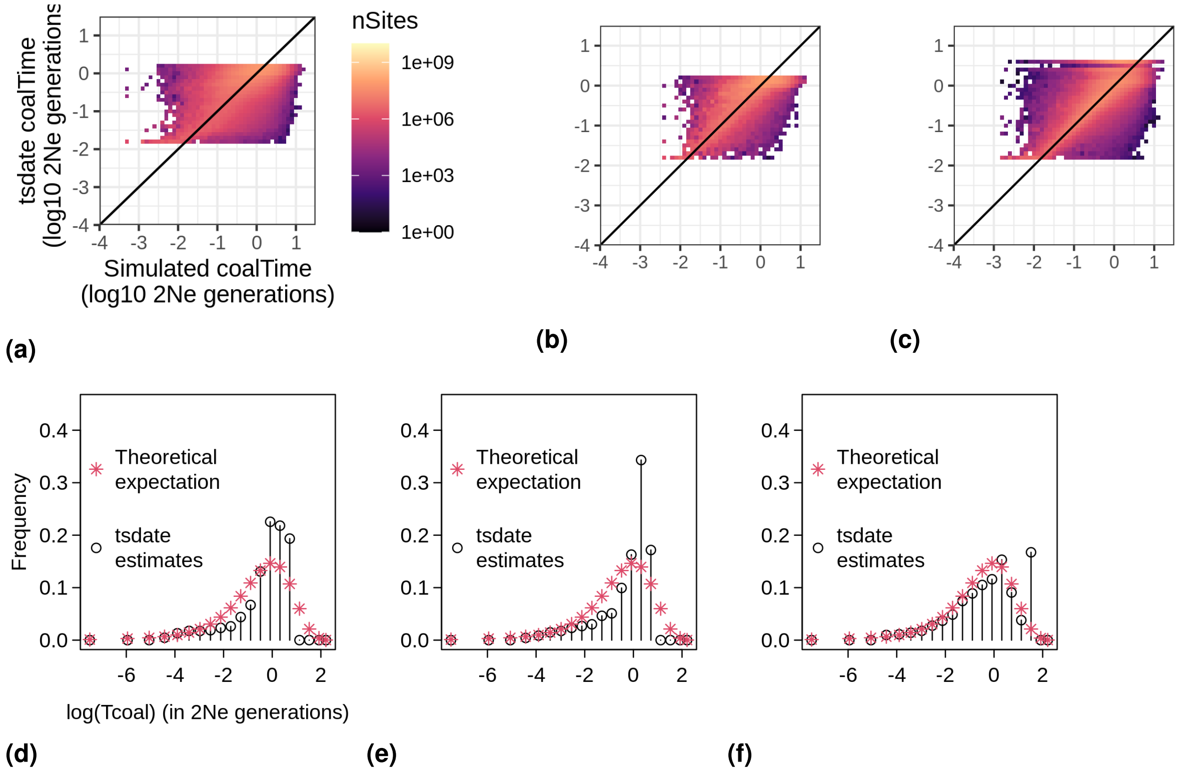


Figure 3 tsdate increased mutation to recombination rate. Left column: $\mu = \rho = 2 * 10^{-8}$; middle column: $\mu/\rho = 10$ decreasing recombination rate ($\rho = 2 * 10^{-9}$); right column: $\mu/\rho = 10$ increasing mutation rate ($\mu = 2 * 10^{-7}$). (A-C) point estimates (mean of 100 MCMC iterations). Mean squared error: A)0.812 B) 0.679 C) 1.399 (D-F) Distribution of coalescence times inferred by tsdate. Plots D shows the same data as in Figure S4B, using different binning.

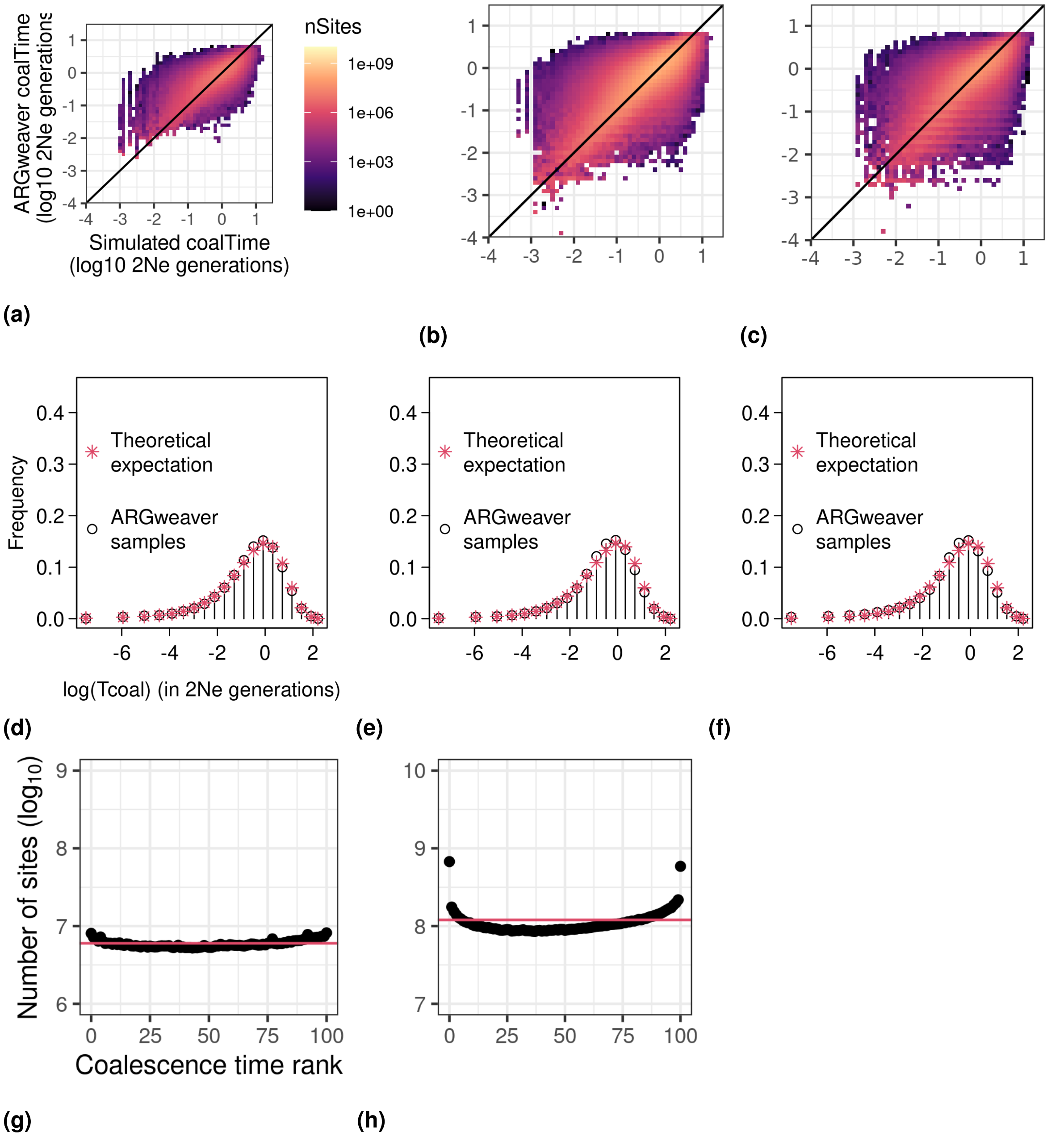


Figure 4 ARGweaver (A-E) Point estimates. Mean squared error: A)0.419 B) 0.384 C) 0.382. (F-J) Distribution of coalescence times, (K-O) Simulation-based calibration. Note that the y axis is centralized on different values but always has the same length. (A,D,G) nsamples = 4 haplotypes (B,E,H) nsamples = 16 haplotypes (C,F,I) nsamples=32 haplotypes

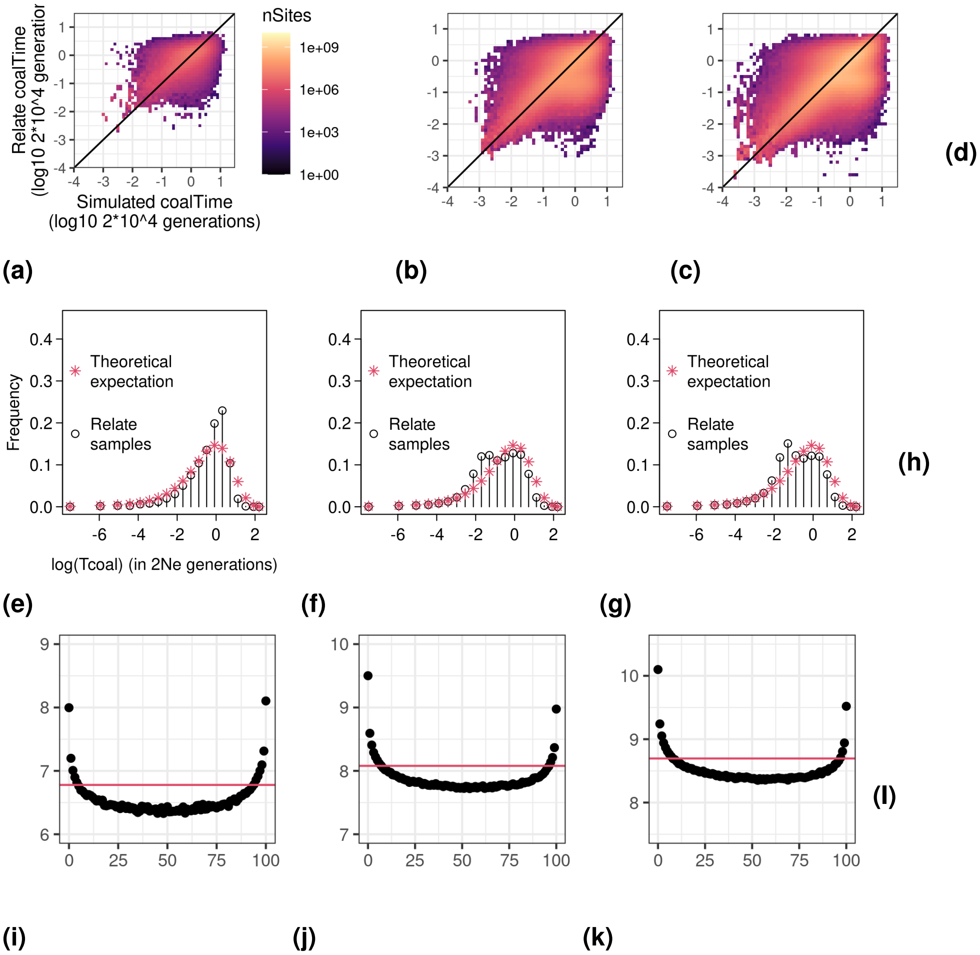


Figure 5 Relate (A-D) Point estimates. Mean squared error: A) 0.676, B) 0.844, C) . (E-H) Distribution of coalescence times, (I-L) Simulation-based calibration.(A,E,I) nsamples = 4 haplotypes (B,F,J) nsamples = 16 haplotypes (C,G,K) nsamples = 32 haplotypes (D,H,L) 80 haplotypes

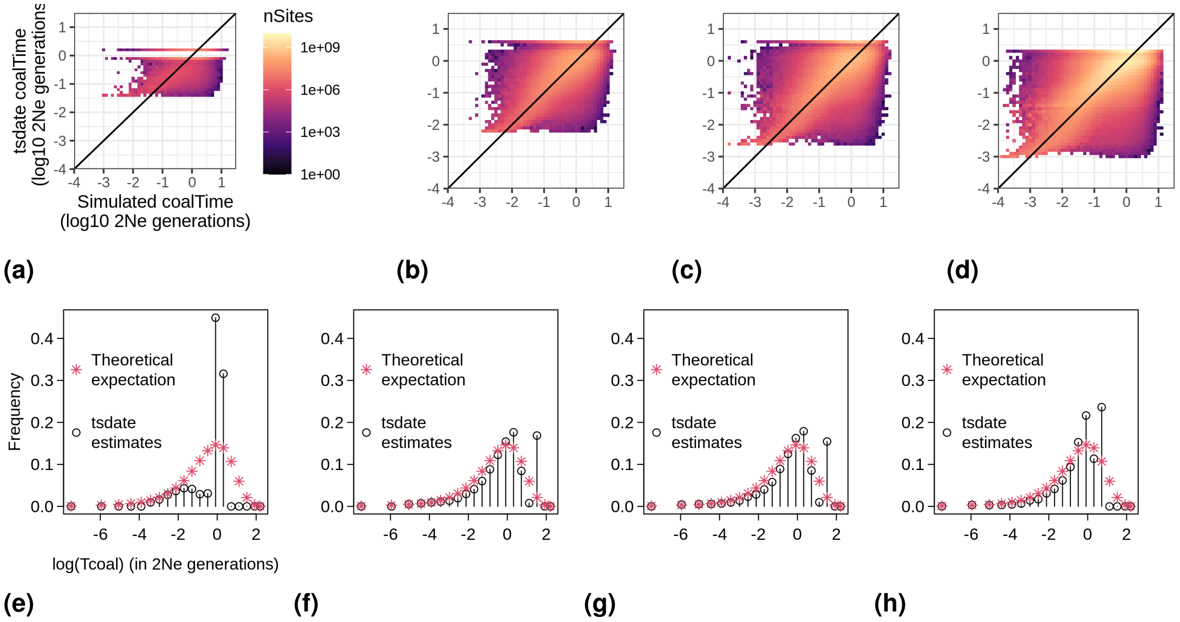


Figure 6 tsdate (A-D) Point estimates. Mean squared error: A) 0.877, B)1.608, C)1.606, D)0.845. (E-H) Distribution of coalescence times.(A,E) nsamples = 4 haplotypes (B,F) nsamples = 16 haplotypes (C,G) 32 haplotypes (D,H) 80 haplotypes

Other points for discussion

Underestimation of coalescence times In **ARGweaver** the excess of underestimated coalescence times (high ranks) could be explained by an effect of the MCMC sampler being stuck with short branches. Once one coalescence time is underestimated, all its subtrees' coalescence times will also tend to be underestimated. In short branches such as those, recombination will be less likely to occur, thus it will be relatively more difficult to sample a tree that changes a branch that was underestimated than to change a branch that was overestimated.

Relate only samples new local trees when there is enough discrepancy with the current local tree. Therefore, short branches, which have fewer mutations, will be sampled less frequently and this may cause biases in estimation. Similarly for old coalescence times, recombinations happening when there are few lineages left will have a lower chance of changing tree topology and will not be well sampled. (Deng *et al.* 2021)

Other features of methods that we didn't explore

ARGweaver can take into account local variation in coverage, mapping quality, sequencing errors, and incorporate gt probabilities. Our simulations assume known genotypes, and no error.

We also assume the data is phased correctly.

Differences in mutation hotspots can be informed to ARGweaver and Relate. We're not using any of that.

In our standard simulations we use mutation rate equal to recombination rate, which is believed to be approximately true for humans. In reality, the average recombination rate is not distributed equally along the genome in humans and other mammals, but is concentrated in recombination hotspots. Therefore, it is possible that inference would be easier with real data, where local trees probably span longer sequences separated by recombination hotspots.

In supplement but not yet mentioned

- point estimate plots in linear scale
- Relate sample order effect

- SMC vs. SMC′ modes in ARGweaver

Literature Cited

Cook, S. R., A. Gelman, and D. B. Rubin, 2006 Validation of software for Bayesian models using posterior quantiles. Journal of Computational and Graphical Statistics **15**: 675–692.

Deng, Y., Y. S. Song, and R. Nielsen, 2021 The distribution of waiting distances in ancestral recombination graphs. Theoretical Population Biology .

Griffiths, R. C. and P. Marjoram, 1997 An Ancestral Recombination Graph. In *Progress in Population Genetics and Human Evolution, IMA Volumes in Mathematics and its Applications, vol. 87*, edited by P. Donnelly and S. Tavare, pp. 257–270, Springer.

Hubisz, M. and A. Siepel, 2020 Inference of Ancestral Recombination Graphs Using ARGweaver. In *Statistical Population Genomics*, edited by Julien Y. Dutheil, volume 2090, chapter 10, pp. 231–266.

Hudson, R. R., 1983 Properties of a neutral allele model with intragenic recombination. Theoretical Population Biology **23**: 183–201.

Kelleher, J., A. M. Etheridge, and G. McVean, 2016 Efficient Coalescent Simulation and Genealogical Analysis for Large Sample Sizes. PLoS Computational Biology **12**: 1–22.

Kelleher, J., K. R. Thornton, J. Ashander, and P. L. Ralph, 2018 Efficient pedigree recording for fast population genetics simulation. PLOS Computational Biology **14**: 1–21.

Kelleher, J., Y. Wong, A. W. Wohns, C. Fadil, P. K. Albers, *et al.*, 2019 Inferring whole-genome histories in large population datasets. Nature Genetics **51**: 1330–1338.

Kingman, J. F. C., 1982 On the Genealogy of Large Populations. Technical report.

Li, N. and M. Stephens, 2003 Modelling Linkage Disequilibrium using Single Nucleotide Polymorphism Data **2233**: 2213–2233.

Marjoram, P. and J. D. Wall, 2006 Fast "coalescent" simulation. BMC Genetics **7**: 16.

McVean, G. A. T. and N. J. Cardin, 2005 Approximating the coalescent with recombination. Philosophical transactions of the Royal Society of London. Series B, Biological sciences **360**: 1387–93.

Neph, S., M. S. Kuehn, A. P. Reynolds, E. Haugen, R. E. Thurman, *et al.*, 2012 BEDOPS: high-performance genomic feature operations. Bioinformatics **28**: 1919–1920.

Rasmussen, M. D., M. J. Hubisz, I. Gronau, and A. Siepel, 2014 Genome-Wide Inference of Ancestral Recombination Graphs. PLoS Genetics **10**.

Speidel, L., M. Forest, S. Shi, and S. R. Myers, 2019 A method for genome-wide genealogy estimation for thousands of samples. Nature Genetics **51**: 1321–1329.

Stern, A. J., P. R. Wilton, and R. Nielsen, 2019 *An approximate full-likelihood method for inferring selection and allele frequency trajectories from DNA sequence data*, volume 15.

Talts, S., M. Betancourt, D. Simpson, A. Vehtari, and A. Gelman, 2020 Validating Bayesian Inference Algorithms with Simulation-Based Calibration. arXiv pp. 1–19.

Wilton, P. R., S. Carmi, and A. Hobolth, 2015 The SMC′ is a highly accurate approximation to the ancestral recombination graph. Genetics **200**: 343–355.

Wiuf, C. and J. Hein, 1999 Recombination as a point process along sequences. Theoretical Population Biology **55**: 248–259.

Wohns, A. W., Y. Wong, B. Jeffery, A. Akbari, S. Mallick, *et al.*, 2021 A unified genealogy of modern and ancient genomes. bioRxiv .

Supplementary Materials

Figures

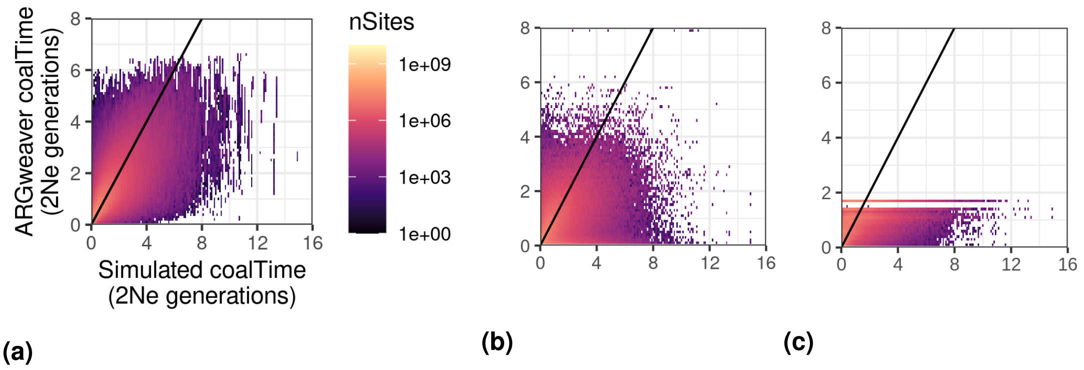
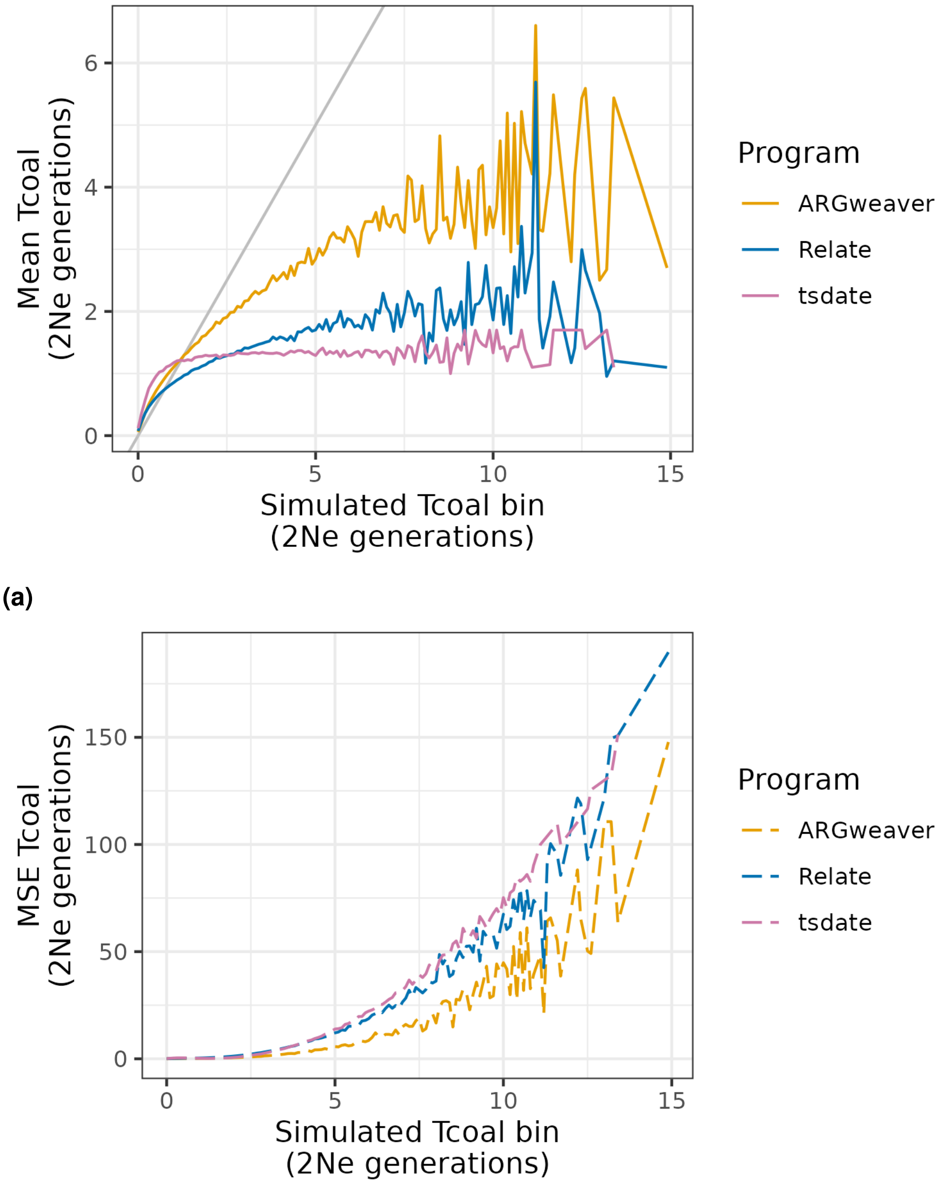


Figure S1 True pairwise coalescence time from msprime simulations compared to inferred coalescence time from (A) ARGweaver (B) Relate (C) tsdate. Note that axes are in linear scale. See Fig. **??** for this data plotted in logarithmic axes. Standard simulations with mu=rec and 8 haploid samples. Diagonal line shows x=y, points show the mean inferred coalescence time across all sites within the same true coalescence time bin.



(a)

(b)

Figure S2 Mean (a) and mean squared error (b) of estimates by ARGweaver, Relate and tsdate in each bin of size 0.1 of simulated coalescence times. These results are for standard simulations with n=8 samples, mutation and recombination rates of $2 * 10^{-8}$.

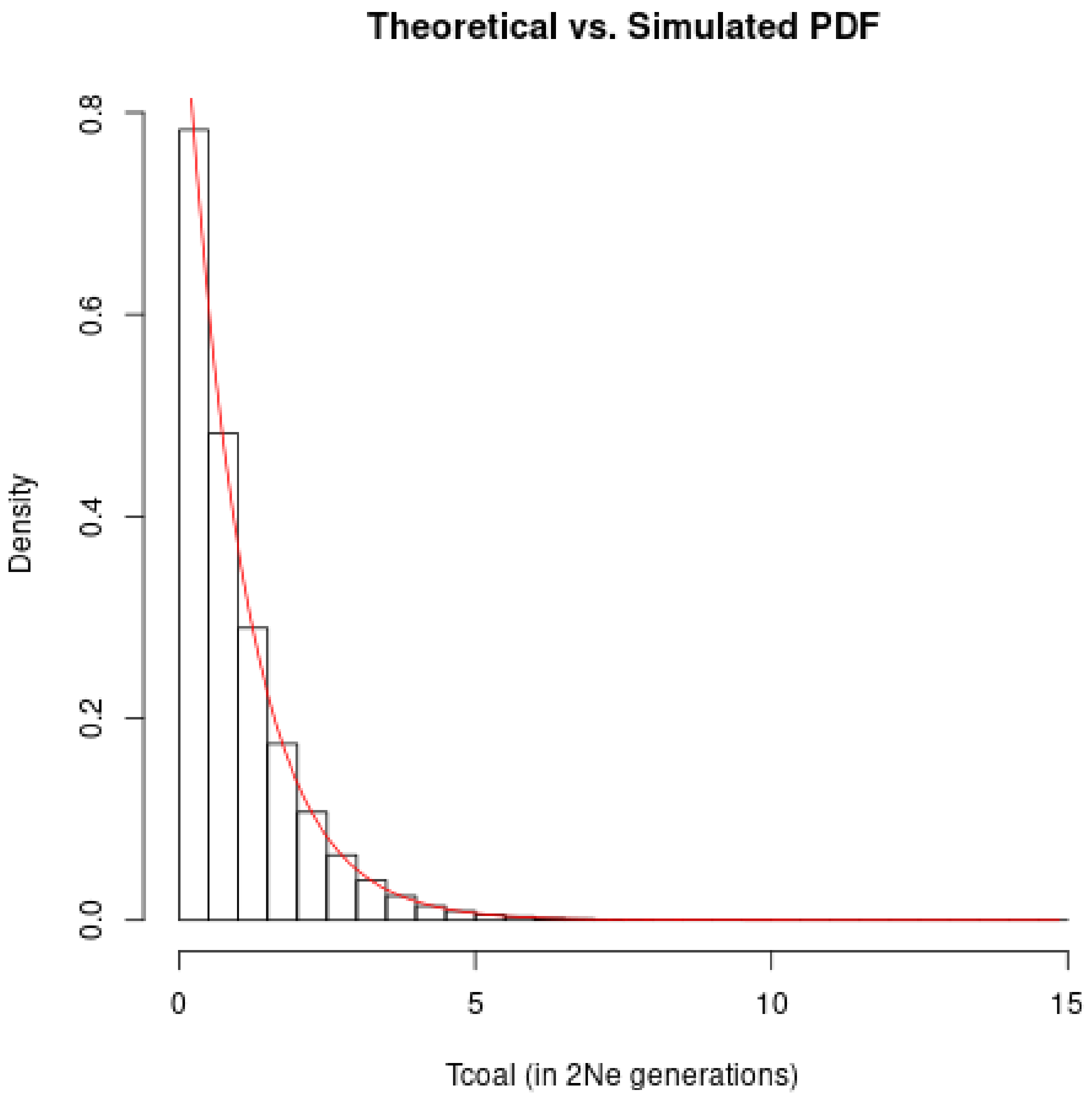
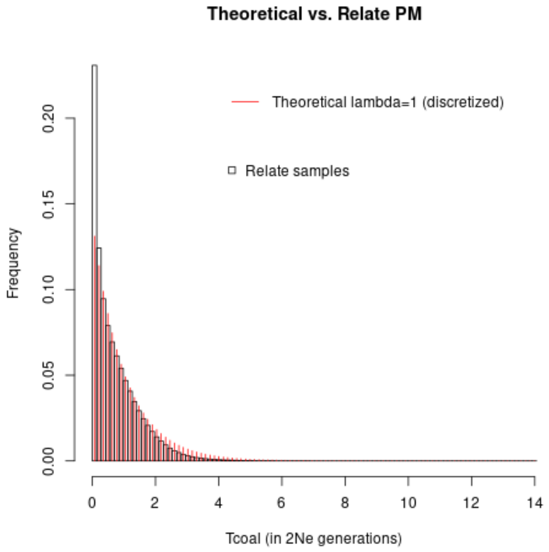


Figure S3 Histogram of the distribution of coalescence times in msprime simulations. Red line show expected exponential distribution with rate 1.



(a) tsdate pmf

Figure S4 Posterior distributions of pairwise coalescence times in Relate and tsdate without ARGweaver time discretization. (A) Relate with n equal size bins (B) tsdate with n equal size bins

Evaluating MCMC Convergence

ARGweaver

MCMC traces were obtained from ARGweaver .stats output file and plotted in R (Fig. S5, S6, S7). Autocorrelation plots were generated with the R function *acf*. Gelman-Rubin convergence statistics were generated with R package *coda* for 5 chains starting at different random ARGs for the first 5Mb of simulated sequence (Table S1, Fig. S8, S9, S10).

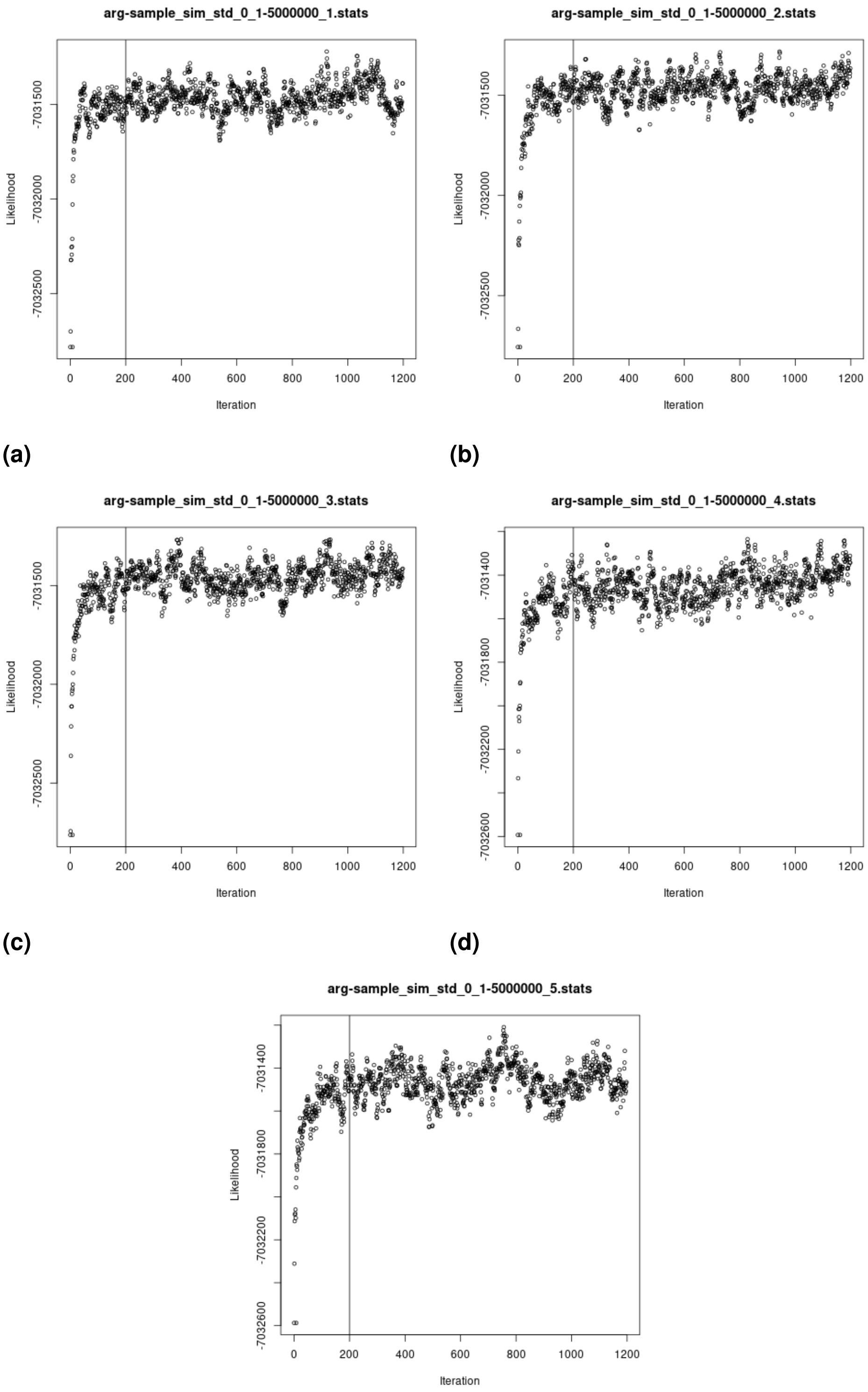
We verified that ARGweaver converged in all three simulation scenarios.

Table S1 Gelman-Rubin convergence diagnostic statistics

	Standard		Lower recombination		Higher mutation	
ARGweaver stats	Point est.	Upper C.I.	Point est.	Upper C.I.	Point est.	Upper C.I.
prior	1.06	1.15	1.01	1.03	1.00	1.01
prior2	1.06	1.17	1.01	1.03	1.00	1.01
likelihood	1.02	1.05	1.04	1.11	1.01	1.02
joint	1.06	1.16	1.01	1.02	1.01	1.02
recombs	1.04	1.1	1.01	1.03	1.00	1.01
noncompats	1.02	1.04	1.01	1.03	1.01	1.04
arglen	1.06	1.16	1.08	1.21	1.05	1.12
Multivatiate psrf	1.15		1.1		1.05	

Relate

Relate does not output traces files like ARGweaver, so we evaulated convergence by running it 5 times and verifying that the posterior distributions of all runs were similar. We also ran Relate for 10x as many iterations, and we verifyed that the posterior distribution of coalescence times didn't change much.



(a)

(b)

(c)

(d)

(e)

Figure S5 Convergence of likelihood across iterations of ARGweaver MCMC for data simulated under standard conditions. We used a burn in of 200 iterations (indicated by vertical line), and ran them for 1200 iterations in total, sampling every 10th iteration.

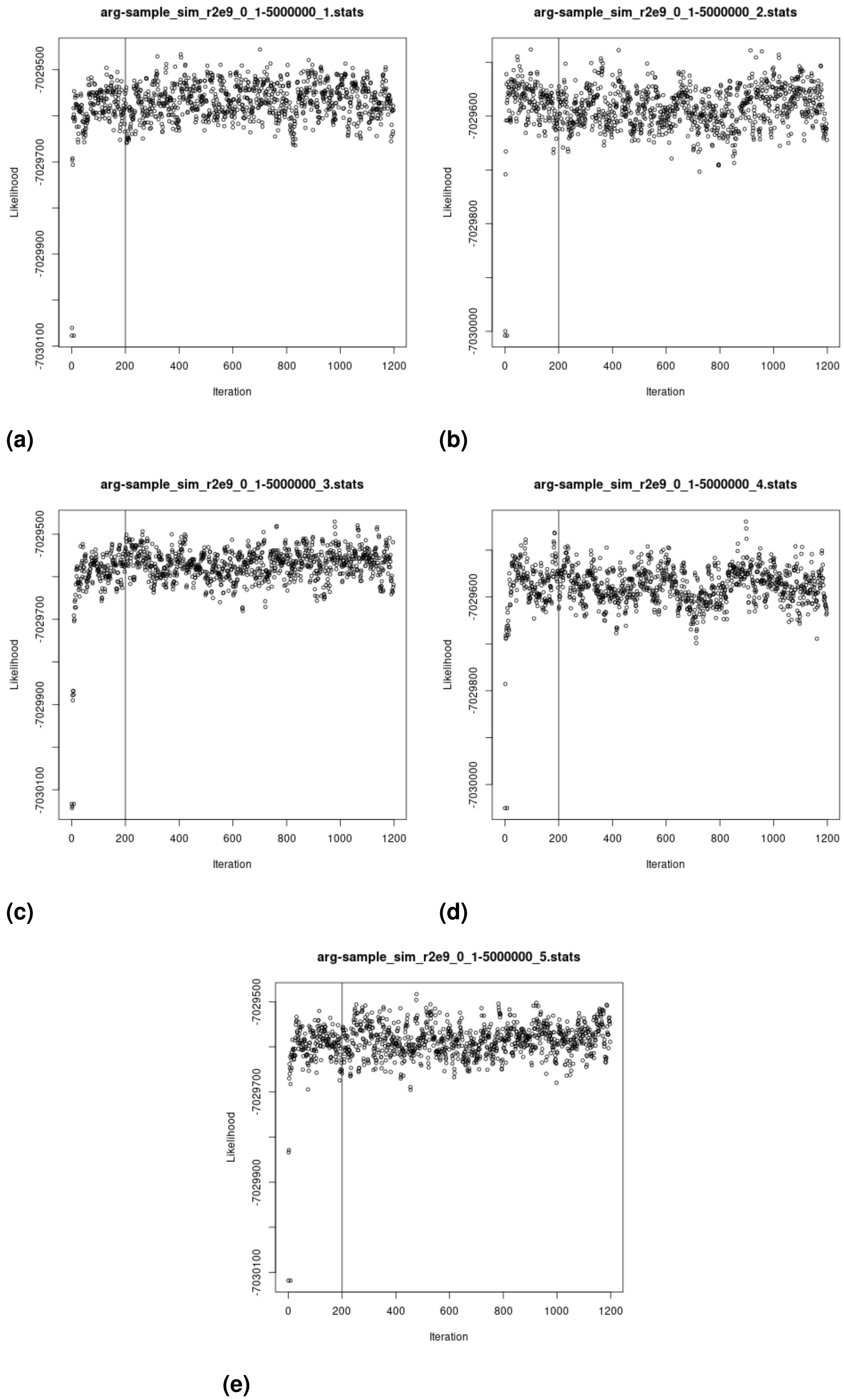
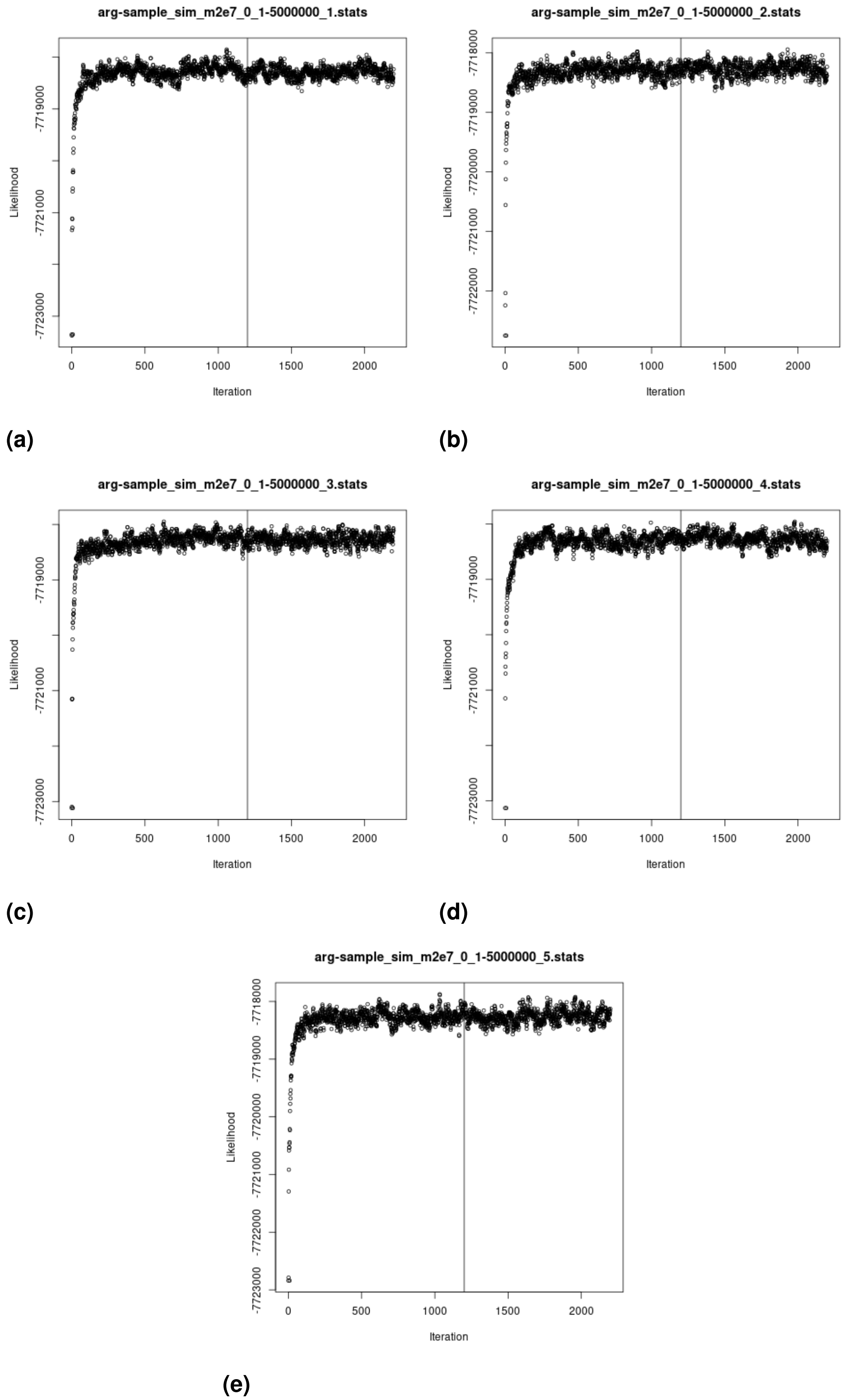


Figure S6 Convergence of likelihood across iterations of ARGweaver MCMC for data simulated with ten times lower recombination rate ($2 * 10^{-9}$) than standard conditions. We used a burn in of 200 iterations (indicated by vertical line), and ran them for 1200 iterations in total, sampling every 10th iteration.



(a)

(b)

(c)

(d)

(e)

Figure S7 Convergence of likelihood across iterations of ARGweaver MCMC for data simulated with ten times higher mutation rate ($2 * 10^{-7}$) than standard conditions. These chains took longer to converge, and we used a burn in of 1200 iterations (indicated by vertical line), and ran them for 2200 iterations in total, sampling every 10th iteration.

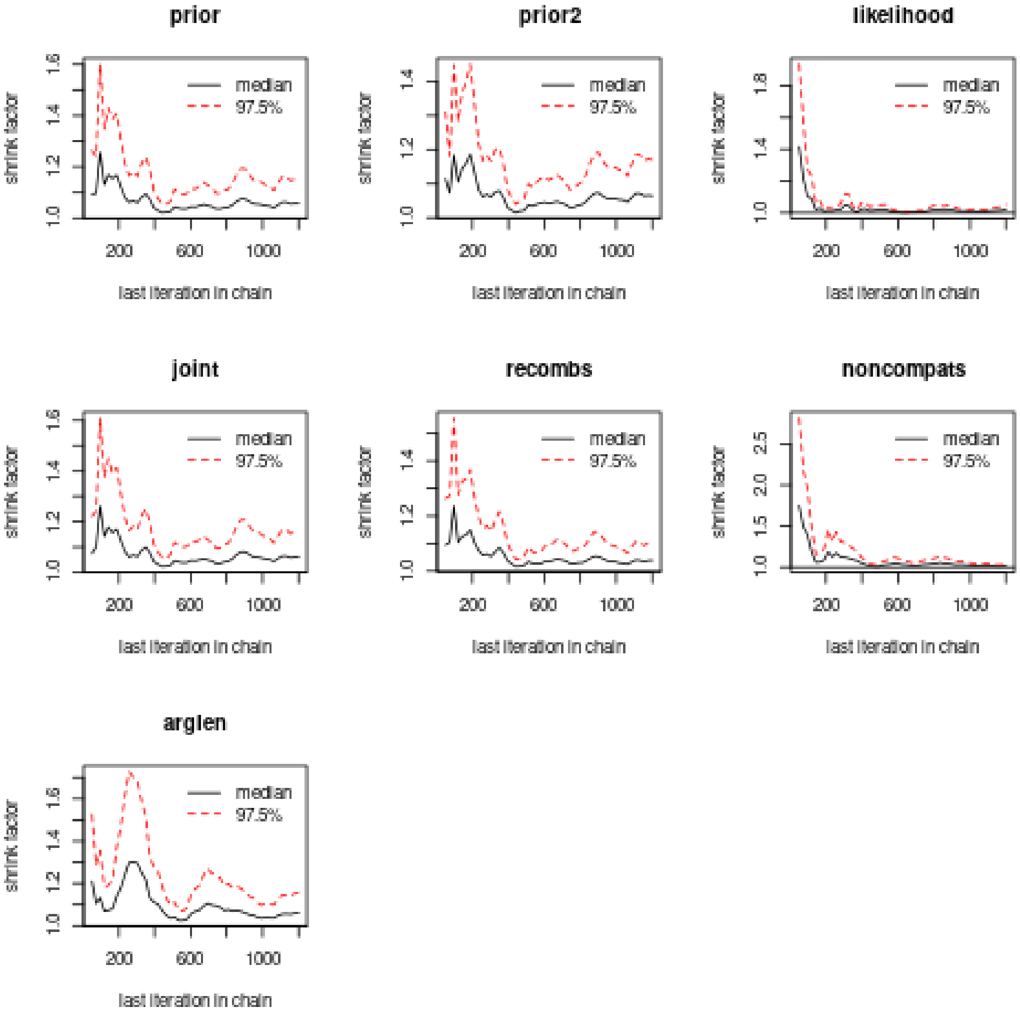


Figure S8 Gelman-Rubin convergence diagnostic for data simulated under standard conditions.

Relate sample order effect

There seems to be an effect of sample order, where coalescence times between the first samples (e.g. pairs 0-1, 0-2, 1-2 in Fig S12 and Fig **??**) show a more dramatic underestimation. This sample order effect showed up in independent simulations.

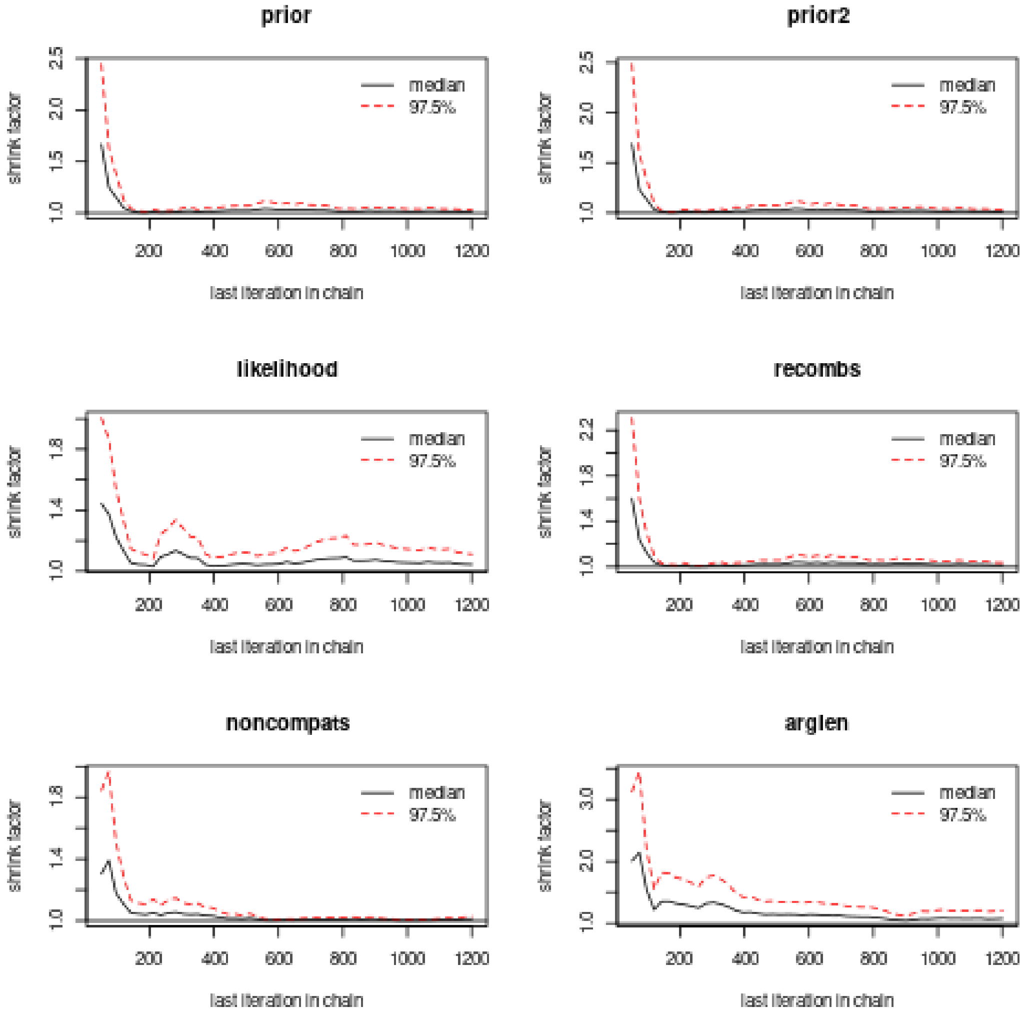


Figure S9 Gelman-Rubin convergence diagnostic for data simulated with ten times lower recombination rate ($2 * 10^{-9}$) than standard conditions.

SMC and SMC' modes in ARGweaver

In previous ARGweaver results, I simulated under standard Hudson coalescent, and did inference in ARGweaver under SMC' (Figs S13A, S14A). Inference looks better when simulating under SMCprime and inferring under SMCprime (Figs S13C, S14C). It is unexpected that simulating and inferring under SMC (Figs S13B, S14B) is not better than simulating under Hudson and inferring under SMC. This could be due to a problem in the implementation of the SMC model in either ARGweaver or msprime.

Zooming in, the residual bias of ARGweaver showing an excess of high ranks (*i.e.* underestimating coalescence times, Fig **??**A) disappears in simulations under SMC or SMCprime. The posterior distributions are still underdispersed in these scenarios, as shown by the U shape of the ranks plots (Fig **??**B,C).

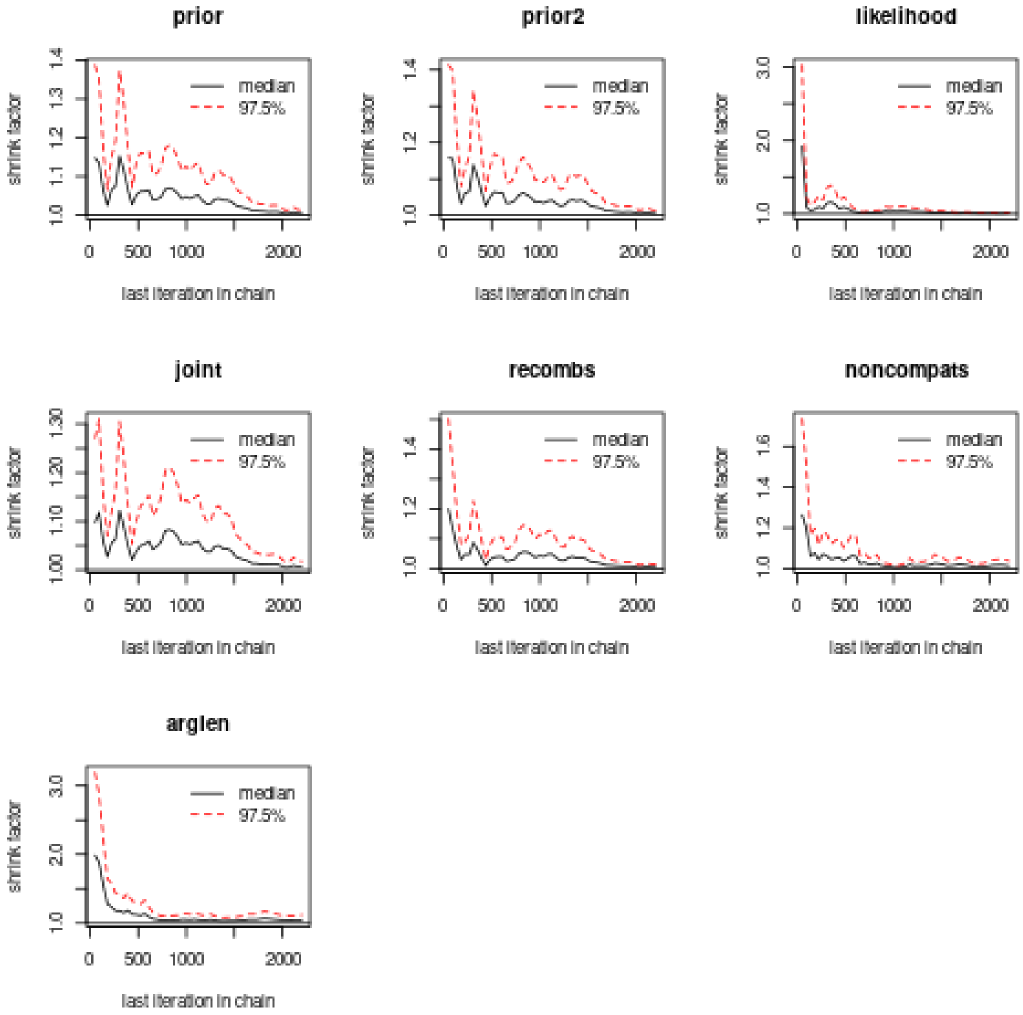


Figure S10 Gelman-Rubin convergence diagnostic for data simulated with ten times higher mutation rate ($2 * 10^{-7}$) than standard conditions.

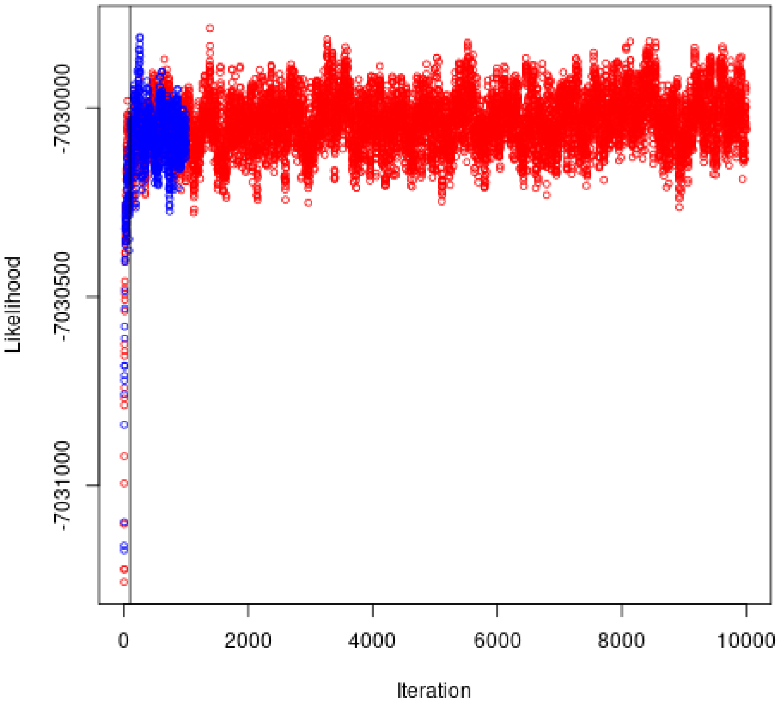


Figure S11 Convergence of the likelihood two ARGweaver runs for 1 chunk of 5Mb. In blue, 1000 iterations; in red, 10,000 iterations

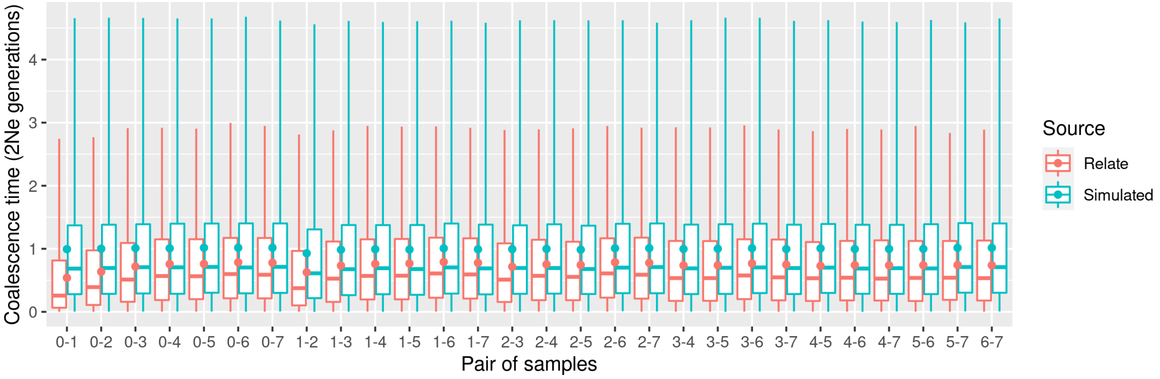
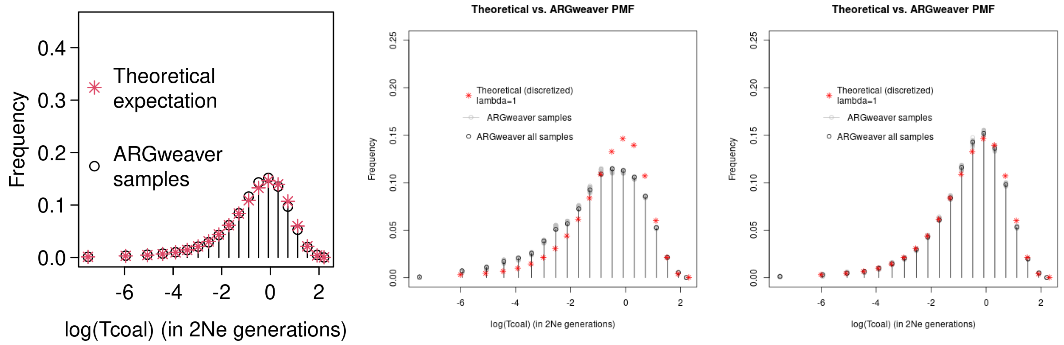


Figure S12 Relate sample order effect. Boxplots show quartiles and dots show means. "Simulated" is the distribution of coal times from the ARGs simulated with msprime. "Relate" is the distribution of coalescence times from Relate output (mean of 1000 MCMC samples for each site).

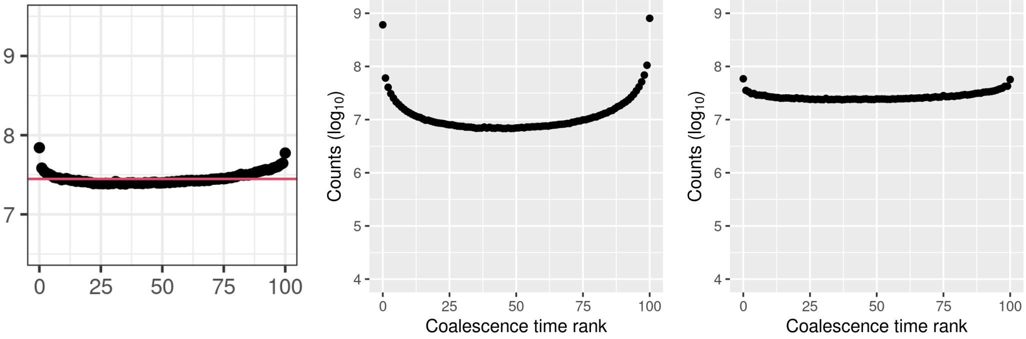


(a) Simulate Hudson, ARGweaver SMC

(b) Simulate SMC, ARGweaver SMC

(c) Simulate SMCprime, ARGweaver SMCprime

Figure S13 Distribution of coal times



(a) Simulate Hudson, ARGweaver SMCprime

(b) Simulate SMC, ARGweaver SMC

(c) Simulate SMCprime, ARGweaver SMCprime

Figure S14 Ranks