Supplementary information for "Variational inference using approximate likelihood under the coalescent with recombination"

Xinhao Liu[1], Huw A. Ogilvie[1], and Luay Nakhleh[1,*]

[1]Department of Computer Science, Rice University, Houston, TX 77005
[*]Corresponding author: Luay Nakhleh (nakhleh@rice.edu)

Contents

S1 A Divide-and-conquer Approach to Larger Numbers of Genomes

While VICAR is general enough so as to handle an arbitrary number of taxa, its running time could increase super-exponentially in the number of taxa. It is important to note, though, that, in practice, this number could grow much slower, depending on the evolutionary parameters, which control the number of distinct states generated by the simulation underlying **ApproximateLikelihood**.

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 Fig. S1. The full species tree on

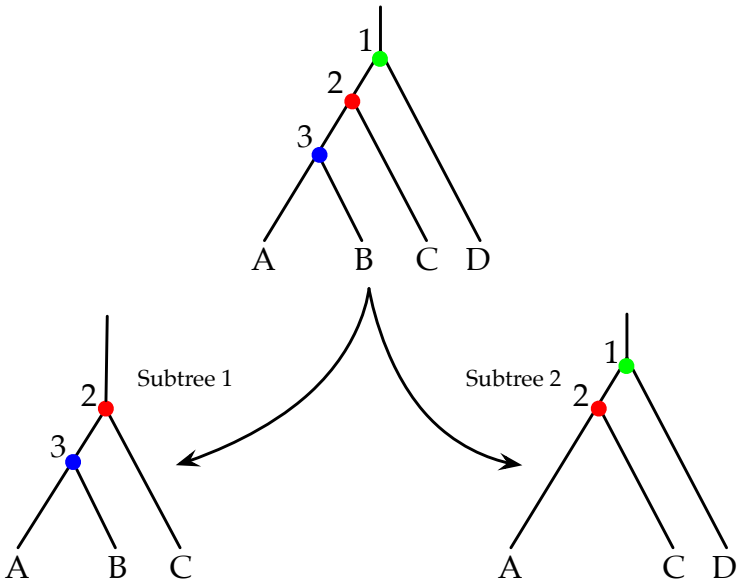

Figure S1: **Divide-and-conquer inference on a four-taxon data set**. The set of taxa $\{A, B, C, D\}$ is divided into two sets $\{A, B, C\}$ and $\{A, C, D\}$ along with their respective species trees. Each of the two data sets are analyzed and the results are merged.

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 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. The divide-and-conquer approach also has similarities to the method of composite likelihoods (Varin, Reid, and Firth, 2011; Larribe and Fearnhead, 2011), which has been used in previous coalHMM frameworks (Cheng and Mailund, 2020; Steinrücken et al., 2019; Schiffels and Wang, 2020), as well as other phylogenetic methods (Liu, Yu, and Edwards, 2010). The major difference between the divide-and-conquer method and other composite likelihood methods is that our approach performs inference on each subproblem, while composite likelihood methods combine likelihoods for a collection of the subsets of the data into one likelihood that can be used for full inference.

S2 Simulation Study on a Four-taxon Data Set

To test the practicability of the divide-and-conquer idea, we simulated a four-taxon sequence data set 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.

S2.1 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 was set to run for 200 iterations with 50 samples per iteration. We

used an improper uniform prior $U(0, \text{inf})$ on node heights, and a gamma prior on population sizes with a shape parameter of 2 and a scale parameter of 25,000. 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 S1. The true values of most of the parameters are within the 95% credible intervals of the estimates,

Table S1: **Result of full inference on the four-taxon data set.** 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	12730	100000
T_{ABC}	145883	11568	160000
T_{ABCD}	422377	16254	450000
N_{AB}	51343	6135	40000
N_{ABC}	37788	3430	40000
N_{ABCD}	55340	6623	40000

but the results are not very accurate. The reason is that 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.

S2.2 Divide-and-conquer Inference

We tried the divide-and-conquer approach on this data set to reduce the running time and improve the accuracy. As shown in Fig. S1, 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 data sets. The root population size of the ((A,B),C) subtree is inferred but omitted when reporting the results, because it is not clear whether the value should be compared with N_{ABC} or N_{ABCD}.

To infer parameters of the ((A,B),C) tree, we used $nb = 2$ 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 S2.

Table S2: **Inference results on the ((A,B),C) tree.**

Parameter	Mean	Standard Deviation	True Value
T_{AB}	99026	12903	100000
T_{ABC}	160495	13154	160000
N_{AB}	39223	7056	40000

To infer parameters of the ((A,C),D) tree, we used $nb = 4$ and `-r` $= 1000$ for building the coalHMM. We used two more sub-branches for each branch since the ((A,C),D) tree has a longer internal branch resulting from not sampling taxon B. This 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 S3.

Table S3: **Inference results on the ((A,C),D) tree.**

Parameter	Mean	Standard Deviation	True Value
T_{ABC}	149776	11254	160000
T_{ABCD}	445402	14242	450000
N_{ABC}	43288	4515	40000
N_{ABCD}	42950	4531	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 S1. Another 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 running time of the inference from over 80 hours to a little bit over 25 hours, while achieving a higher accuracy on all parameters. While the results are in preliminary form and still take a long time to run, the divide-and-conquer

technique shows promising progress towards large-scale population history inference.

S3 Impact of Parameters on Running Times

We now study the impact of nb, `-r`, and other 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. Fig. S2 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 on the running time, as the

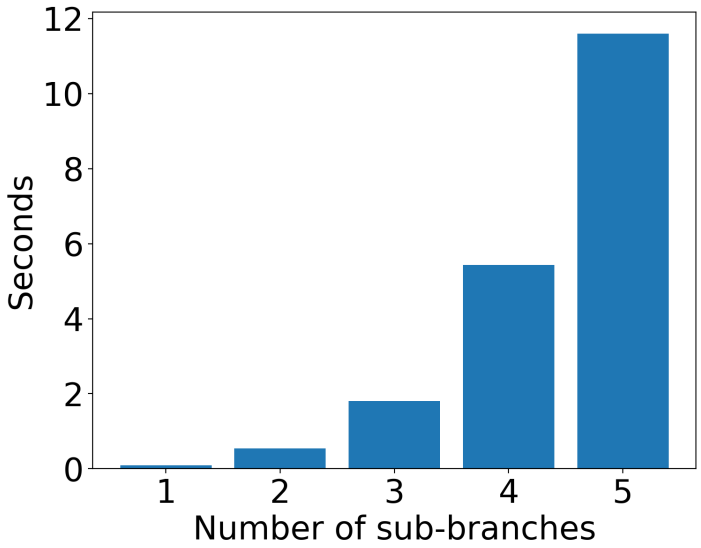

Figure S2: **Running time of the Forward algorithm.** The actual running time taken by the Forward algorithm when calculating the approximate likelihood of Scenario 3, with different nb parameter values in Algorithm 1.

Forward algorithm time significantly increases with the number of sub-branches. The reason is that the number of hidden states of the resulting coalHMM is a high-degree polynomial in the number of sub-branches, which results in a large increase to the Forward algorithm running time since it is quadratic in the HMM size. Fig. S3 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.

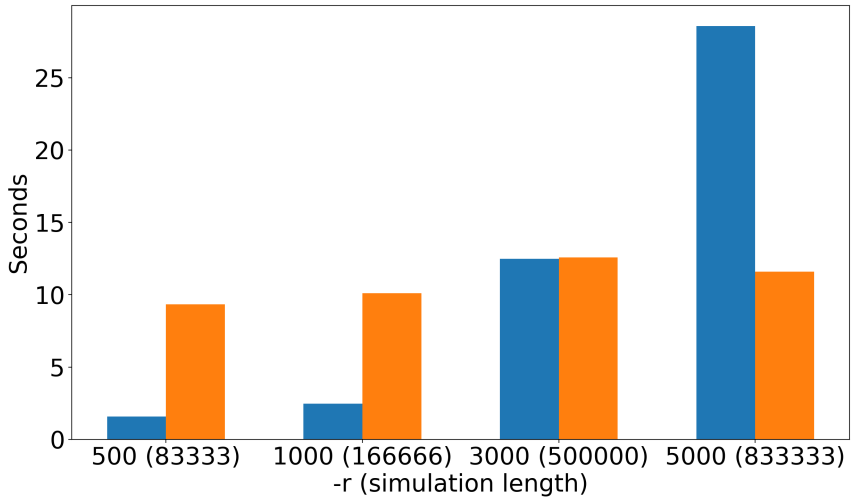

Figure S3: **Running times of building the HMM and the Forward algorithm.** The actual running times taken by building the HMM by simulation (blue bars) and running the Forward algorithm (orange bars) when calculating the approximate likelihood of Scenario 3, with different `-r` parameter values (the l parameter in Algorithm 1).

We observe that the time taken for building the HMM grows with the simulation length, while the Forward algorithm running time remains roughly unchanged since the number of hidden states of the coalHMM does not depend on the simulation length. Note that the HMM building time grows super-linearly with the simulation length, which has to do with the complexity of the underlying simulator, in our case `msprime`. Since the transition matrix of the HMM only depends on the number of transitions between pairs of adjacent sites, simulating multiple independent short regions instead of a long sequence might achieve a near linear growth in simulation time. We implemented an option to simulate several regions of short length whose sum of lengths add up to the total simulation length when building the coalHMM. Fig. S4 shows a comparison of the time taken by simulating a long region versus several short regions, with each short region of length 5,000 sites, for the same scenario as Fig. S3.

We observe a significant reduction in HMM building time when simulating multiple independent short regions, and the `msprime` simulation time indeed scales linearly with simulation length. The

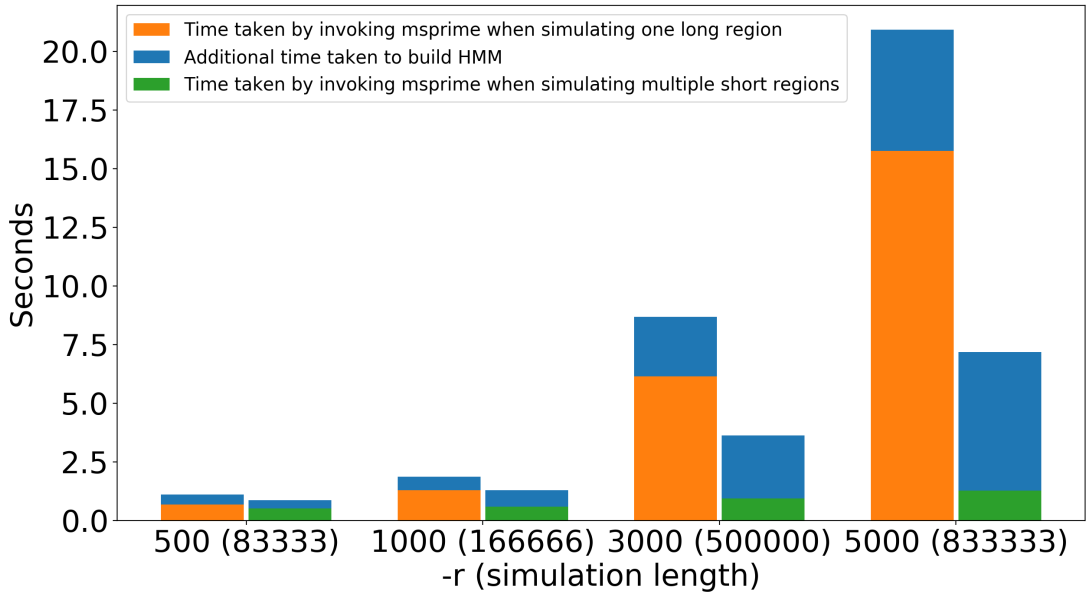

Figure S4: **Comparison of simulating a long region versus multiple short regions.** The HMM building times taken by simulating a long sequence versus multiple short sequences whose lengths add up to the long sequence length. The left bars at each x axis point are the times taken to build the HMM when simulating one long region. The right bars are the times taken to build the HMM when simulating multiple independent short regions (of length 5,000). The orange parts and the green parts are the time taken solely by invoking `msprime` for the two cases, and the blue parts are the time taken by VICAR code to parse the simulation output and actually build one HMM.

time taken by VICAR code to parse the simulation output (i.e., summarize the coalescent histories) and to build one HMM remains unchanged. We inferred the continuous parameters on scenario 3 using both long region simulation and short region simulation for building the coalHMM, and achieved the same level of accuracy, which indicates that simulating independent short regions when building the HMM for likelihood computation does not affect the accuracy of inference.

Fig. S5 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, 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

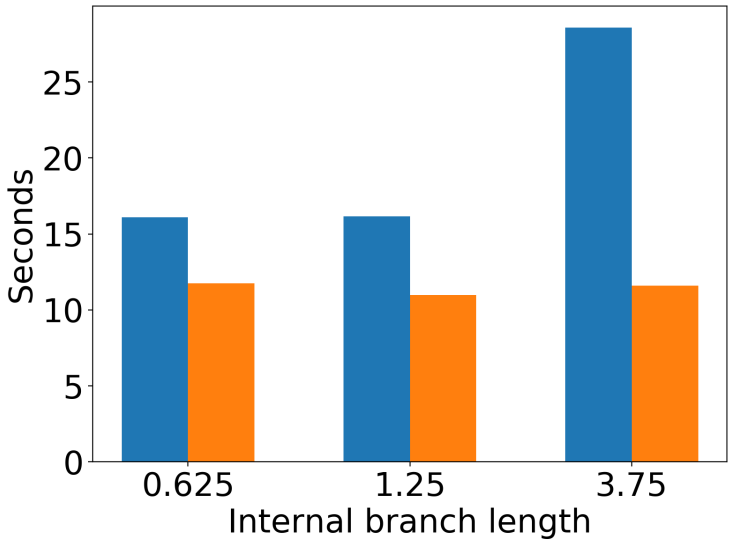

Figure S5: **Running times of building the HMM and the Forward algorithm.** The actual running times taken by building the HMM by simulation (blue bars) and running the Forward algorithm (orange bars) when calculating the approximate likelihood of Scenarios 1, 2, and 3.

we identify as a future research direction.

References

Cheng, Jade Yu and Thomas Mailund (2020). "Ancestral Population Genomics with Jocx, a Coalescent Hidden Markov Model". In: *Statistical Population Genomics*. Humana, New York, NY, pp. 167–189.

Larribe, Fabrice and Paul Fearnhead (2011). "On composite likelihoods in statistical genetics". In: *Statistica Sinica*, pp. 43–69.

Liu, Liang, Lili Yu, and Scott V Edwards (2010). "A maximum pseudo-likelihood approach for estimating species trees under the coalescent model". In: *BMC evolutionary biology* 10.1, pp. 1–18.

Schiffels, Stephan and Ke Wang (2020). "MSMC and MSMC2: the multiple sequentially markovian coalescent". In: *Statistical population genomics*. Humana, New York, NY, pp. 147–166.

Steinrücken, Matthias et al. (2019). "Inference of complex population histories using whole-genome sequences from multiple populations". In: *Proceedings of the National Academy of Sciences* 116.34, pp. 17115–17120.

Varin, C., N. Reid, and D. Firth (2011). "An overview of composite likelihood methods". In: *Statistica Sinica* 21, pp. 5–42.