Coalescent: an Open-Science framework for Importance Sampling in Coalescent theory

Importance sampling is widely used in coalescent theory to compute data likelihood. Efficient importance sampling requires a trial distribution close to the target distribution of the genealogies conditioned on the data. Moreover, an efficient proposal requires intuition about how the data influence the target distribution. Different proposals might work under similar conditions, and sometimes the corresponding concepts overlap extensively. Currently, there is no framework available for coalescent theory that evaluates proposals in an integrated manner. Typically, problems are not modeled, optimization is performed vigorously on limited datasets, user interaction requires thorough knowledge, and programs are not aligned with the current demands of open science. We have designed a general framework (http://coalescent.sourceforge.net) for importance sampling, to compute data likelihood under the infinite sites model of mutation. The framework models the necessary core concepts, comes integrated with several data sets of varying size, implements the standard competing proposals, and integrates tightly with our previous framework for calculating exact probabilities. The framework computes the data likelihood and provides maximum likelihood estimates of the mutation parameter. Well-known benchmarks in the coalescent literature validate the framework's accuracy. We evaluate several proposals in the coalescent literature, to discover that the order of efficiency among three standard proposals changes when running time is considered along with the effective sample size. The framework provides an intuitive user interface with minimal clutter. For speed, the framework switches automatically to modern multicore hardware, if available. It runs on three major platforms (Windows, Mac and Linux). Extensive tests and coverage make the framework accessible to a large community.

PeerJ PrePrints | http://dx.doi.org/10.7287/peerj.preprints.395v1 | CC-BY 4.0 Open Access | received: 25 May 2014, published: 25 May 2014

Susanta Tewari[1§], John L Spouge[1]

[1] National Center for Biotechnology Information, Bethesda, MD 20894

[§]Corresponding author

Email addresses:

ST: tewaris@ncbi.nlm.nih.gov

JLS: spouge@ncbi.nlm.nih.gov

Overview

Infinite-Sites Model (K69)

Excellent overviews of various coalescent models are already available (e.g., Hein, Schierup & Wiuf, 2005; Wakeley, 2009). Here for the sake of completeness, we briefly describe the infinite-sites model (denoted "*K69*", after Kimura, 1969). Most of our notation follows Wakeley (2009).

Consider an aligned sample of DNA sequences, and note that alignment columns can contain gaps. If an alignment column lacks gaps, call it a "site". Model K69 considers only sites. Under Model K69, the sample evolves its from most recent common ancestor (MRCA) by through reproduction and mutation at the sites. Model K69 is most suitable for long DNA sequences with low mutation rates, because it permits at most one mutation at each site during the evolution of the sampled sequences. The state of each site in a sampled sequence (its "character") can therefore be summarized by a binary digit: 0, if the corresponding DNA letter agrees with the MRCA; and 1, otherwise. A site is segregating if some sequences in the relevant sample contain the character 1. Thus, the segregating sites comprise the essential data in the sample. The sample data can be represented as $D = [X, \nu]$, where X is a binary matrix (i.e., $X_{i,j} \in \{0,1\}$) with distinct rows X_i ("haplotypes") and ν is an column vector such that ν_i counts of multiplicity of the haplotype X_i among the sampled sequences. Thus,

the character of haplotype X_i at site j is $X_{i,j} \in \{0,1\}$. See Table 1 for a sample data set $D = [X, \nu]$ similar to Figure 8.6 in Wakeley (2009).

As an aid to visualization, data conforming to Model K69 always have a unique *gene tree*. Figure 1, e.g., shows the gene tree corresponding to Table 1. Within a gene tree, the order of mutations on any edge is arbitrary, and permutation of the column order in the haplotype matrix X does not affect the gene tree for $D = (X, \nu)$. Gusfield (1991) gives an efficient algorithm for constructing gene trees.

Under Model K69, the MRCA (represented by a matrix with a single row of 0s) evolves into the sample data $D = [X, \nu]$ by passing stepwise through a sequence of ancestral configurations, which have a form $C = [X, \nu]$ similar to the sample data. In the following, a "singleton row i" is a row with count $\nu_i = 1$. Backwards in time, starting from sample back to MRCA, each step corresponds to one of three possible evolutionary operations on the current ancestral configuration: (1) coalescence (deleting one of some identical rows); (2) removing a mutation of type I (changing the only 1 in some column j into 0, leaving the corresponding singleton row i unique in the ancestral configuration); and (3) removing a mutation of type II (changing the only 1 in some column j into 0, to make the corresponding singleton row i the same as some other row(s) in the ancestral configuration). Removal in both mutational types I and II is restricted to "the only 1 in some column" and "a singleton row", because Model K69 permits at most one mutation at each site. Once a mutation is removed, the corresponding site is no longer a segregating site (i.e., the corresponding column in the new binary matrix has only 0s). Thus, a computer can efficiently represent the removal of a mutation simply by removing the corresponding column from X, a representation we now use. Under the representation, the MRCA becomes an empty matrix with count 1.

PeerJ PrePrints | http://dx.doi.org/10.7287/peerj.preprints.395v1 | CC-BY 4.0 Open Access | received: 25 May 2014, published: 25 May 2014

To represent the three evolutionary operations mathematically, consider an ancestral configuration $C=[X,\nu]$, let e_i denote a column vector with 1 in the i-th position, and 0 elsewhere Given C, let A denote the set of singleton rows i, and $A(i)$ denote column j with the smallest index in row i, so that row i and column $A(i)$ satisfy the restrictions on mutations of type I. (In the following, the arbitrary choice of the smallest column index $A(i)$ is feasible, because column order is irrelevant to the gene tree.) Let δ_i be the corresponding evolutionary operator that deletes column $A(i)$ from X, creating the new ancestral configuration $[\delta_i X,\nu]$. Similarly, let B denote the set of singleton rows i, each with a single column j satisfying the restrictions on mutations of type II. For each i, let $B(i)$ be the row index of the "merge haplotype", the haplotype that row i becomes when the 1 in column j is changed to 0. Let R_i be the corresponding evolutionary operator, which deletes row i and column j from X, creating the haplotype matrix $R_i X$, and which also deletes the i-th row of the column vector ν, so the new ancestral configuration is $\left[R_i X, R_i\left(\nu+e_{B(i)}\right)\right]$.

Having defined the sample space of ancestral configurations $[X,\nu]$ and the steps that Model K69 permits, we now determine the corresponding probability measure $p[X,\nu]$, which implicitly depends on a population mutation parameter θ. The MRCA probability is $p([\ ],\ [1])\ =\ 1.0$, and the probabilities $p[X,\nu]$ satisfy the recursion

- $$\begin{aligned} n(n-1+\theta)\,p[X,\nu] = & \sum_{i:\nu_i\geq 2}\nu_i(\nu_i-1)\,p[X,\nu-e_i]+ \\ & \theta\sum_{i:i\in A}p[\delta_i X,\nu]+ \\ & \theta\sum_{i:i\in B}\left(\nu_{B(i)}+1\right)p\left[R_i X, R_i\left(\nu+e_{B(i)}\right)\right] \end{aligned} \quad , \qquad (1.1)$$

For introductory examples see (Hein, Schierup & Wiuf, 2005; Wakeley, 2009).

Importance Sampling for Computing Likelihood

Wakeley (2009) gives an overview of importance sampling for computing likelihood in coalescent theory. Briefly, here are the key concepts. To make the dependence of the probability on the mutational parameter θ explicit, let $p(D;\theta) = p[X,\nu]$ for $D = (X,\nu)$. The data probability in (1.1) can also be written as the following:

- $$p(D;\theta) = \sum_G p(D \mid G;\theta)\, p(G) \qquad (1.2)$$

where the sum is over all genealogies G consistent with the data D. Let $q(.)$ be any probability measure, and $E_q[.]$ be its expectation, and define the likelihood ratio $w(G) = p(G)/q(G)$. If $p(.)$ is absolutely continuous with respect to $q(.)$, i.e., if $q(G) > 0$ wherever $p(G) > 0$, then

- $$\begin{aligned} p(D;\theta) &= \sum_G p(D \mid G;\theta) \frac{p(G)}{q(G)} q(G) \\ &= E_q\left[p(D \mid G;\theta) \frac{p(G)}{q(G)} \right] \\ &= E_q\left[p(D \mid G;\theta)\, w(G) \right] \end{aligned} \qquad (1.3)$$

Usually, in the context of importance sampling, $p(.)$ is called the target distribution; $q(.)$, the trial distribution; and $w(.)$, the importance sampling weight (Hammersley & Handscomb, 1964; Liu, 2001). Given R realizations G_r ($r = 1,\ldots,R$) of the genealogy G *independently* sampled from the trial distribution $q(.)$, then the strong law of large numbers implies that with probability *1*,

- $$p(D;\theta) = \lim_{n\to\infty} R^{-1} \sum_{r=1}^{R} p(D \mid G_r;\theta)\, w(G_r). \qquad (1.3)$$

Thus, importance sampling provides a likelihood estimator

- $$\hat{p}_{IS}(D;\theta) \approx R^{-1} \sum_{r=1}^{R} p(D \mid G_r;\theta)\, w(G_r). \qquad (1.4)$$

PeerJ PrePrints | http://dx.doi.org/10.7287/peerj.preprints.395v1 | CC-BY 4.0 Open Access | received: 25 May 2014, published: 25 May 2014

Equation (1.1) provides a sequence of steps from a population sample to its most recent common ancestor (MRCA), each step corresponding to a single ancestral coalescence or the loss of a single mutation. A Monte Carlo simulation can therefore assign trial probabilities $q(.)$ to these time-steps, to create a sequential importance-sampling scheme (Liu, 2001). Many coalescent processes are Markovian, making sequential importance sampling (SIS) a natural choice for simulating them, because events occurring at different time-steps are independent. The coalescence literature often uses the terminology "proposals" for the probability assignments, but "proposals" is not the standard usage in the Monte Carlo literature. (The non-standard usage might be derived from the Metropolis method, which accepts or rejects "proposals".) In any case, this paper tries to adhere to the standard terminology (Liu, 2001; Robert, C. P. & Casella, G., 2004) in the Monte Carlo literature.

Standard Sequential Samplers

Sequential samplers choose among the evolutionary operations corresponding to the different terms in equation (1.1). Because each operation is determined once the corresponding haplotype X_i in the ancestral configuration $C=(X,\nu)$ is known, we let $q(i\,|\,C)$ with various subscripts denote corresponding trial probability.

The Ethier-Griffiths-Tavare (EGT) Sequential Sampler

The EGT recursion in equation (1.1) directly suggests a sequential sampling scheme (Griffiths & Tavare, 1994):

- $$q_{GT}(i\,|\,C) \propto \left\{ \begin{array}{cc} \nu_i - 1 & \nu_i \geq 2 \\ \dfrac{\theta}{n} & i \in A \\ \dfrac{\theta\left(\nu_{B(i)}+1\right)}{n} & i \in B \\ 0 & \text{otherwise} \end{array} \right\} \qquad (1.5)$$

The Stephens-Donnelly (SD) Sequential Sampler
Stephens & Donnelly (2000) developed a sampling scheme by characterizing the target distribution and then approximating it with

- $$q_{SD}(i \mid C) \propto \begin{Bmatrix} v_i & v_i \geq 2 \text{ or } i \in A \text{ or } i \in B \\ 0 & \text{otherwise} \end{Bmatrix} \quad (1.6)$$

The Hobolth-Uyenoyamay-Wiuf (HUW) Proposal
Hobolth, Uyenoyamay & Wiuf (2008) approximated the effects of all mutations on the probabilities for the next step from the sample to the MRCA, to derive

- $$q_{HUW}(i \mid C) \propto \begin{Bmatrix} \sum_m u_{i,m}(\theta_0) & v_i \geq 2 \text{ or } i \in A \text{ or } i \in B \\ 0 & \text{otherwise} \end{Bmatrix}, \quad (1.7)$$

where θ_0 is a fixed value of θ,

- $$u_{i,m}(\theta) = \begin{Bmatrix} p_\theta(d_m)\dfrac{v_i}{d_m} & X_{i,m} = 1 \\ \left[1 - p_\theta(d_m)\right]\dfrac{v_i}{(n - d_m)} & X_{i,m} = 0 \end{Bmatrix}$$

$$d_m = \sum_m X_{i,m} v_i$$

and

- $$p_\theta(d_m) = \frac{\sum_{k=2}^{n-d_m+1} \frac{d_m - 1}{n-k} \frac{1}{k-1+\theta} \binom{n-d_m-1}{k-2} \binom{n-1}{k-1}^{-1}}{\sum_{k_0=2}^{n-d_m+1} \frac{1}{k_0-1+\theta} \binom{n-d_m-1}{k_0-2} \binom{n-1}{k_0-1}},$$

$$p_\theta(1) = \frac{\frac{1}{n-1+\theta}}{\sum_{k_0=2}^{n} \frac{1}{k_0-1+\theta} \frac{k_0-1}{n-1}}$$

where $p_\theta(d_m)$ is the probability that the next evolutionary operation (coalescence or mutation) involves a row X_i where $X_{i,m} = 1$ (i.e., haplotype i bears mutation m), and $u_{i,m}$

denotes the probability of involving row X_i in the next mutation event m. The proposal probability in equation (1.7) sums $u_{i,m}$ over all mutations m for row X_i.

Implementation

Motivation

Felsenstein *et al.* (1999) summarizes the main problems posed by computational inference in population genetics. To be useful to a broad community in population genetics, new theoretical methods must yield computations linear in time-and space-complexity. To compare theoretical methods, computer implementation of the corresponding equations is insufficient. Particularly for Monte Carlo methods, an integrative analysis must model whole problems, to reuse results, reduce the cost of maintenace, and maintain reliability. We therefore followed a systems approach, where the current framework for importance sampling mirrors our approach to computing exact coalescent probabilities (Tewari & Spouge, 2012).

We now describe the architecture of the framework, diagramming the key classes and interfaces with the unified modelling language (UML), while displaying the various connections and assumptions. The framework consists of several packages, which progressively narrow the most general concepts down to the specifics of *K69*, the infinite-sites model of mutation.

Core framework

The core framework models the concepts for any domain of sampling. It corresponds to the package *commons.is*. Figure 3 displays the key classes: *Sampler*, *Proposal,* and *Factor*.

Sampler

Sampler generalizes equation (1.4):as

- $$\hat{E}_q[X] = \sum_{r=1}^{R} h(X_r)\, w(X_r), \qquad (1.8)$$

where $h(X)$ is a "mean function", and $w(X_r) = p(X_r)/q(X_r)$, $p(.)$ and $q(.)$ being the target and trial distributions, respectively. For coalescent models, X corresponds to the genealogy G, and $h(X)$ denotes the conditional probability of observing data D given the genealogy X. Note that for coalescent models, X represents the relevant events in the entire genealogical history of the sample, including coalescent events. For the standard sequential sampling schemes above, $h(X) = 1$ identically for all X.

Proposal & Factor

Proposal draws an independent sample X each time *sample()* is called and *Factor* computes $w(X)$ in equation (1.8). *Factor* can be created by computing the weight $w(X)$ directly (by implementing sub-interface *Proposal_w_Prob*) or by implementing an analytical expression for the ratio, if available (e.g., the so-called "functional path" F_j in equation (12) of Griffiths & Tavare, 1994).

Coalescent Models

The following subsection describes SIS schemes for coalescent genealogies. Our framework for exact probabilities (Tewari & Spouge, 2012) already contains the general concepts for coalescent models, so it implements sequential schemes for specific models readily, using the key class *GProposal* within the package *coalescent.is*. Figure 3 sketches the implementation of key classes and their interactions.

GProposal

GProposal implements SIS via *Proposal_w_Prob*. It builds the sample and its probability recursively, from the alternatives that equation (1.1) presents for each step, using probabilities from the framework for exact probabilities. Figure 4 illustrates SIS in a coalescent process. Although the framework does not currently implement *partial-weight* based judgements (Liu, 2001) in its SIS, it can easily accommodate them.

PeerJ PrePrints | http://dx.doi.org/10.7287/peerj.preprints.395v1 | CC-BY 4.0 Open Access | received: 25 May 2014, published: 25 May 2014

Genealogy & AC

Genealogy and *AC* ("Ancestral Configuration") come from the framework for exact probabilities. *Genealogy* defines the chain of events from the sample to the MRCA. *AC* denotes sample configuration in a generic coalescent model and given the allele and event types, specifies the recursion.

Method: proposalWeight

GProposal specifies the SIS of all coalescent models based on equation (1.1). Specific proposals (e.g., equations (1.5), (1.6) and (1.7)) need only implement the abstract method *proposalWeight*. This design localizes errors, limiting the scope of problems associated with a specific proposal; a useful feature for Monte Carlo. It also stabilizes results when comparing sequential sampling schemes: although general optimizations might improve the performance of several schemes, the implementation of each scheme would share the gain, thereby maintaining relative efficiencies.

The Infinite-Sites Model *K69*

For the infinite-sites model of mutation (*K69*), we implemented three standard proposals (equations (1.5), (1.6) and (1.7)) with the abstract method *proposalWeight*, as described above. The proposals are collected in *GProposals_K69* (see Figure 6), which follows the factory design pattern (Gamma et.al., 1995). The framework for exact probabilities already specifies the interface *K69_AC* of the ancestral configuration under infinite-sites model of mutation, so we used it to specify the three proposals. Figure 7 illustrates the implementation of SD Proposal, which demonstrates that proposals can be written compactly from the corresponding equations, leaving the framework to encapsulate the details.

Multiple Parameters

When computing a likelihood for a range of parameter values, one can generate several realizations to compute a likelihood for each value, or even more efficiently, one can use each realization to compute likelihoods for several values (Griffiths & Tavare, 1994). Ideally,

PeerJ PrePrints | http://dx.doi.org/10.7287/peerj.preprints.395v1 | CC-BY 4.0 Open Access | received: 25 May 2014, published: 25 May 2014

sweeping across several values should be pluggable, i.e., the sweep should become automatic, once the scheme for the model is given. Our framework is unusual, in that its careful design incorporates foundations permitting automatic sweeps, i.e., no additional implementation is required to sweep across a range of parameter values, when a standard proposal is written against a single model.

Parallel Sequential Sampling Schemes

The object-oriented design of the framework naturally promotes the use of multiple computer cores. The framework design generally prefers modularity to optimizing running times, but platform-independence and the possibility of running the framework on common multi-core machines justifies the compromise. Multiple schemes can run on separate *threads*, reducing computing time and providing direct visual feedback on their running times. Some schemes both increase statistical power and reduce computing time relative to other schemes, but more typically, a nuanced trade-off occurs.

Tests and Coverage

The user application is only one part of the framework: extensibility to future solutions also puts constraints on it. Its expansion must not break existing features and design contracts (Freeman & Pryce, 2009). Debugging must remain manageable and coverage of *checked exceptions* (known causes of disruption) must grow over time. *Automated unit tests* along with their *coverage* can satisfy these constraints. *Coverage* measures how well the tests are doing their job. Table 2 provides the number of tests and coverage for the packages in the framework. Typically, more than 70% coverage should inspire confidence.

Results

In the literature, the data set of Griffiths & Tavaré (Griffiths & Tavaré, 1994) has become a standard benchmark for proposals. Figure 8 displays the gene tree for the dataset in a format similar to Figure 3 of (Hobolth, Uyenoyamay & Wiuf, 2008). We computed the likelihood

PeerJ PrePrints | http://dx.doi.org/10.7287/peerj.preprints.395v1 | CC-BY 4.0 Open Access | received: 25 May 2014, published: 25 May 2014

curve with the framework and validated the maximum likelihood estimates (MLE) with the published values shown in Table 3. Table 4 displays the maximum likelihood values for various proposals. The published values were estimated from figures in (Wu, 2009), because point estimates were not available. The framework's estimates are consistent with the literature. The effective sample size (ESS) for proposal $Q(.)$ is defined (Liu, 2001, p.35) as

- $$ESS = \frac{R}{1 + \mathrm{var}_q\left(w(X)\right)}, \qquad (1.9)$$

where R is the number of realizations (samples), and w is the corresponding importance weight. Hobolth *et al.* (Hobolth, Uyenoyamay & Wiuf, 2008) compared various proposals by estimating the ESS. Loosely, ESS quantifies the vicinity of the target distribution to the trial distribution, so SIS improves as the ESS increases.

Hobolth *et al.* (Hobolth, Uyenoyamay & Wiuf, 2008) investigated the performance of the three proposals by comparing ESSs as mutation rates and numbers of realizations varied. The relative efficiency of the three proposals (EGT < SD < HUW) was stable in their Figure 6. Our simulation study has two aims: (1) to confirm the proposal ranking; and (2) to consider the effect running times have on the ranking. Figure 8 plots the ratio of ESSs to the ESS for SD, for various mutation rates and numbers of realizations. For each cell, the tool *ms* (Hudson, 2002) or *msms* (Ewing, G. & Hermisson, J., 2010, acting as a cross-platform fallback) simulated three independent sets of samples for the corresponding mutation rate and number of sample size. Within each cell, there are 2 plots: (1) one for a fixed number of realizations; and (2) one for fixed computer running time, the maximum time taken by any proposal in the first plot for the fixed number of realizations. The number of realizations was 108,000, close to 100,000, as in Hobolth et al. (Hobolth, Uyenoyamay & Wiuf, 2008).

PeerJ PrePrints | http://dx.doi.org/10.7287/peerj.preprints.395v1 | CC-BY 4.0 Open Access | received: 25 May 2014, published: 25 May 2014

Discussion

Table 4 verifies (by both standard error and ESS) the results in (Hobolth, Uyenoyamay & Wiuf 2008): the SIS performance order is EGT < SD < HUW. Note, however, that the running times for SD and EGT are nearly equal, and half of the running time for HUW. Within the framework, all proposals share the same runtime infrastructure, so the accuracy in HUW comes at a price: about double the running time per realization.

Like Table 4, Figure 9 verifies the results in (Hobolth, Uyenoyamay & Wiuf, 2008), but shows that when the figure of merit *is ESS per running time*, EGT < HUW < SD, where SD is only slightly better than HUW. Although EGT is noticeably faster than HUW, HUW compensates with the accuracy of its estimates. The computational expense of HUW derives mostly from $p_\theta(d)$ in Equation (1.7), despite its being computed only once per realization. However, when HUW computes the MLE of the mutation rate θ, it does not need any scaling as occurs in Equation (12) in (Griffiths & Tavare, 1994), partly because of the asymptotics in Equation (12) & (13) in (Hobolth, Uyenoyamay & Wiuf, 2008). Thus, HUW computes the MLE noticeably faster than EGT, especially when many mutation rates θ are examined.

Conclusions

Running time can be a significant consideration when comparing the efficiency of different importance sampling schemes. If ESS per running time replaces ESS as a figure of merit, then order of efficiency among the three proposals considered (equations (1.5), (1.6) and (1.7)) changes from EGT < SD < HUW to EGT < HUW < SD. HUW is noticeably slower than both SD & EGT, but its ESS is only slightly inferior to SD, because its accuracy compensates for the increase in running time. (Hobolth, Uyenoyamay, & Wiuf, 2008) have also indicated that data patterns inherent in coalescent models with mutations could be

PeerJ PrePrints | http://dx.doi.org/10.7287/peerj.preprints.395v1 | CC-BY 4.0 Open Access | received: 25 May 2014, published: 25 May 2014

exploited to improve importance sampling schemes further. Our work might be useful in benchmarking these and other improvements to importance sampling schemes.

We have followed a systems approach, in the spirit of open science (Stodden 2013a; Stodden 2013b; Stodden 2013c). Running our software verifies claims made here, and for verification purposes, the supplementary materials include an illustrated stepwise instructions manual. All downloads are available at the project website: http://coalescent.sourceforge.net. The user interface is intuitive, so it requires only a basic familiarity with the theory. The framework is open source, scalable and comes with several test cases and is backed by a large test coverage. The present framework augments our earlier framework for exact algorithms (Tewari & Spouge, 2012) with the same approach, adding another tool to the likelihood analysis for population genetics data under the infinite sites model of mutation.

Acknowledgements

It is our pleasure to acknowledge helpful conversations with Dr. Sergey Sheetlin.

References

Felsenstein, J., Kuhner, M. K., Yamato, J., & Beerli, P. (1999). Likelihoods On Coalescents: A Monte Carlo Sampling Approach To Inferring Parameters From Population Samples Of Molecular Data. *Statistics in molecular biology and genetics.*

Freeman, S., & Pryce, N. (2009, October). *Growing Object-Oriented Software, Guided by Tests.* Addison-Wesley Professional.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). *Design Patterns Elements of Reusable Object-Oriented Software.* Addison-Wesley.

Griffiths, R. C., & Tavare, S. (1994). Ancestral inference in population genetics. *Statistical Science*, 307-319.

Hammersley, J. M., & Handscomb, D. C. (1964). *Monte Carlo Methods.* Methuen, London.

Hein, J., Schierup, M. H., & Wiuf, C. (2005). *Gene Genealogies, Variation and Evolution A Primer in Coalescent Theory.* Oxford University Press.

Hobolth, A., Uyenoyamay, M. K., & Wiuf, C. (2008). Importance Sampling for the Infinite Sites Model. *Statistical Applications in Genetics and Molecular Biology, 7*(1).

Hudson, R. R. (2002). Generating samples under a Wright-Fisher neutral model. *Bioinformatics, 18*, 337-338.

Kimura, M. (1969, April). The Number Of Heterozygous Nucleotide Sites Maintained In A Finite Population Due To Steady Flux Of Mutations. *Genetics, 61*(4), 893-903. Retrieved from http://www.genetics.org/cgi/reprint/61/4/893

Liu, J. S. (2001). *Monte Carlo Strategies in Scientific Computing.* Springer.

Robert, C. P., & Casella, G. (2004). *Monte Carlo statistical methods.* New York : Springer.

Stephens, M., & Donnelly, P. (2000). Inference in molecular population genetics. *J. R. Statist. Soc. B*, 605-635.

Stodden, V. (2013, June 3). "Setting the Default to Reproducible" in Computational Science Research. *"Setting the Default to Reproducible" in Computational Science Research*. Retrieved from http://www.siam.org/news/news.php?id=2078

Stodden, V. (2013, September). Changes in the Research Process Must Come From the Scientific Community, not Federal Regulation. *Changes in the Research Process Must Come From the Scientific Community, not Federal Regulation*. Retrieved from http://blog.stodden.net/2013/09/24/changes-in-the-research-process-must-come-from-the-scientific-community-not-federal-regulation/

Stodden, V. a. (2013, 06). Toward Reproducible Computational Research: An Empirical Analysis of Data and Code Policy Adoption by Journals. *PLoS ONE, 8*(6), e67111. Retrieved from http://dx.doi.org/10.1371%2Fjournal.pone.0067111

Tewari, S., & Spouge, J. L. (2012). Coalescent: an open-source and scalable framework for exact calculations in coalescent theory. *BMC Bioinformatics, 13*, 257.

Wakeley, J. (2009). *Coalescent Theory. An Introduction.* Roberts and Company Publishers.

Wu, Y. (2009). Exact Computation of Coalescent Likelihood for Panmictic and Subdivided Populations Under the Infinite Sites Model. *IEEE Transactions On Computational Biology And Bioinformatics*.

Author Instructions

https://peerj.com/about/author-instructions/

Figures

Fig 1

Title: Gene tree for data in Table 1.

Legend: Gene tree for data in Table 1 drawn using the framework.

Snapshot:

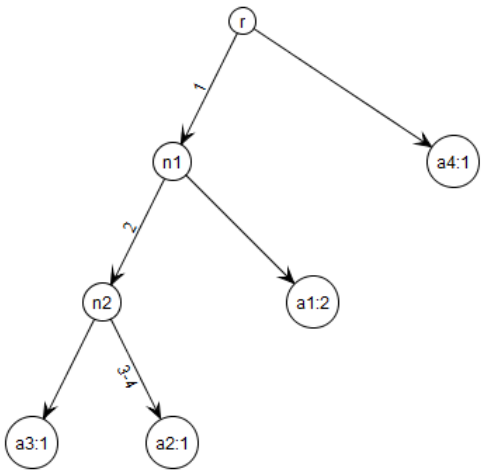


Fig 2

Title: Tree operations

Legend: Tree operations in the recursion for the infinite sites model.

Snapshot:

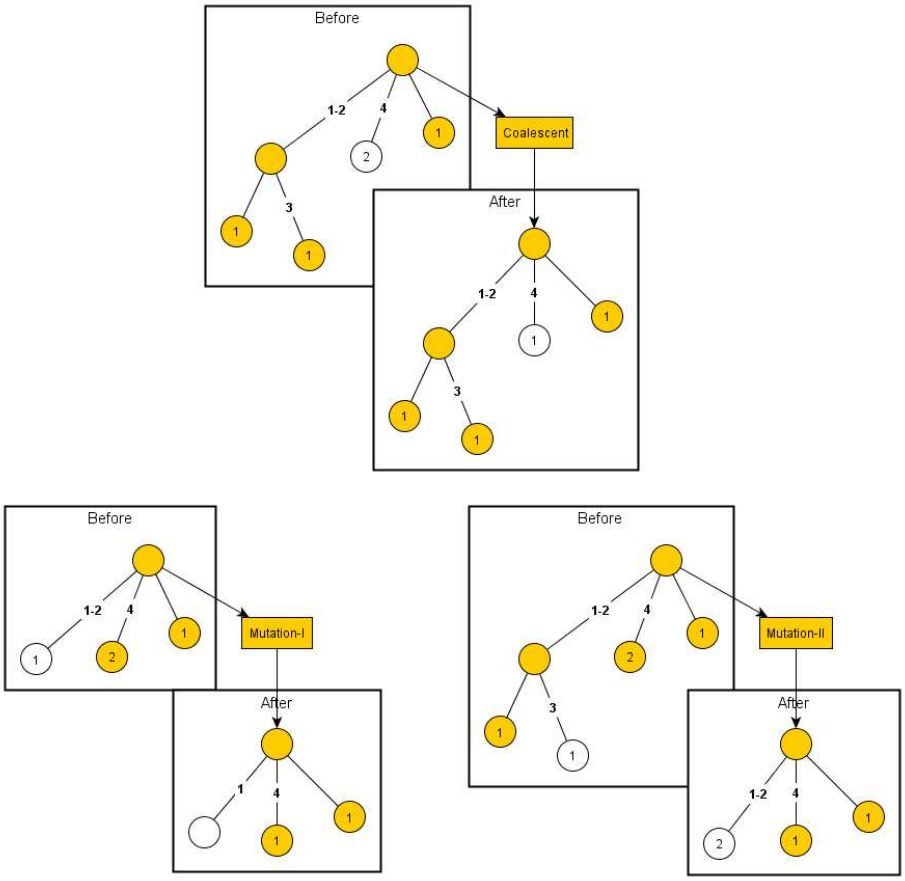


Fig 3

Title: UML diagram of the core framework.

Legend: UML diagram of the core framework. The diagram shows key classes in the package *commons.is*. These classes would apply to any domain of sampling.

Snapshot:

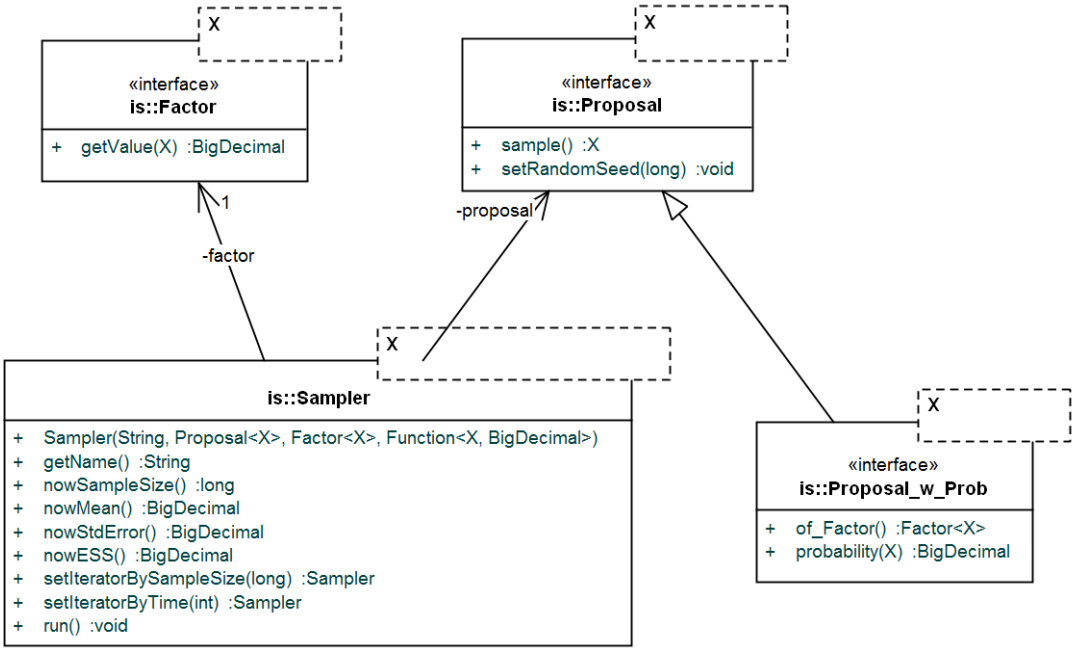


Fig 4

Title: UML diagram of the IS framework specific to infinite-sites model.

Legend: UML diagram of the IS framework specific to infinite-sites model. The diagram shows relations for the key class *GProposal* in the package *coalescent.is*.

Snapshot:

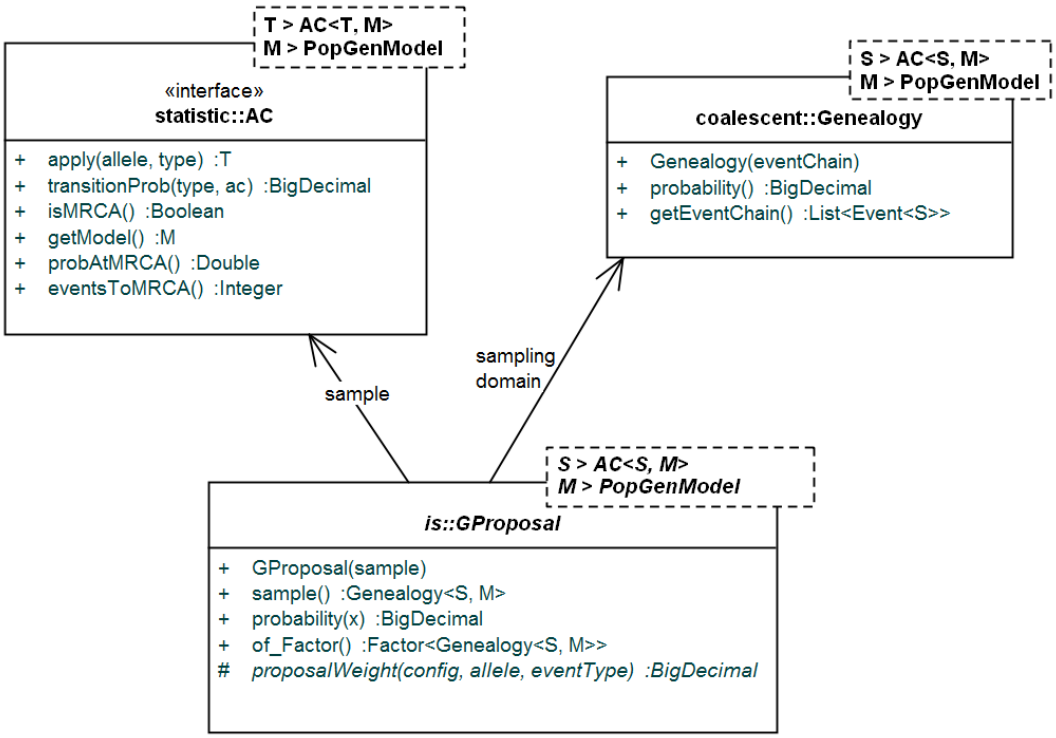


Fig 5

Title: Genealogy, as a domain of sampling.

Legend: Genealogy, as a domain of sampling. The diagram shows the space of genealogies compatible with data (not shown). A sample from this space is marked. Each node is labelled in the format (I, J): i corresponds to `statistic.eventstoMRCA()` in Fig 3 (total number of events, coalescent or mutation, before the configuration reaches MRCA) and j denotes a counter at that level.

Snapshot:

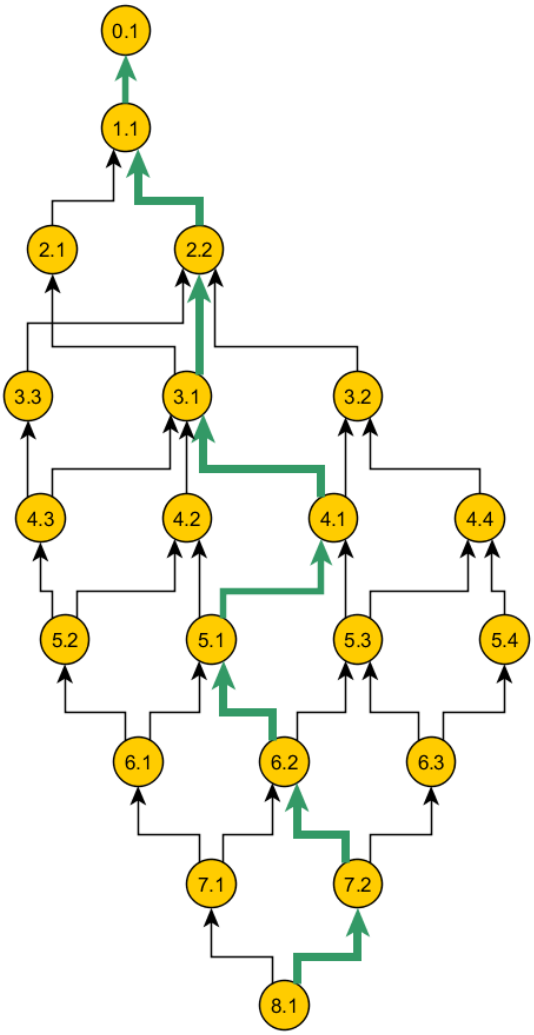


Fig 6

Title: API of factory methods for all implemented proposals.

Legend: API of factory methods for all implemented proposals for the infinite-sites model.

API for ancestral configuration (AC) is also shown.

Snapshot:

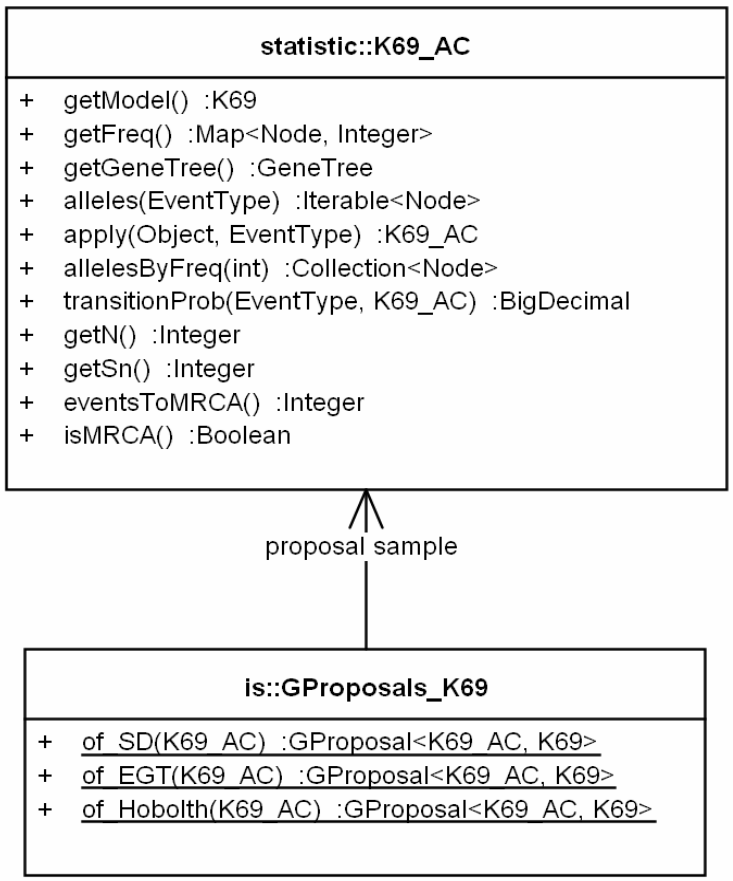


Fig 7

Title: Demonstrates writing of a new proposal.

Legend: Demonstrates writing of a new proposal using the SD proposal. Note that the implementation is a close translation of the corresponding equation.

Snapshot:

```
public static GProposal<K69_AC, K69> of_SD(final K69_AC sample) {

    return new GProposal<K69_AC, K69>(sample) {

        @Override
        protected BigDecimal proposalWeight(final K69_AC config,
                final Object allele,
                final EventType eventType) {

            final Node actual_allele = (Node) allele;
            final int allele_freq    = config.getGeneTree().getFreq(actual_allele);

            return new BigDecimal(allele_freq);
        }
    };
}
```

Fig 8

Title: Gene tree corresponding to the benchmark data set.

Legend: Gene tree corresponding to the benchmark data set. The figure was drawn using the framework.

Snapshot:

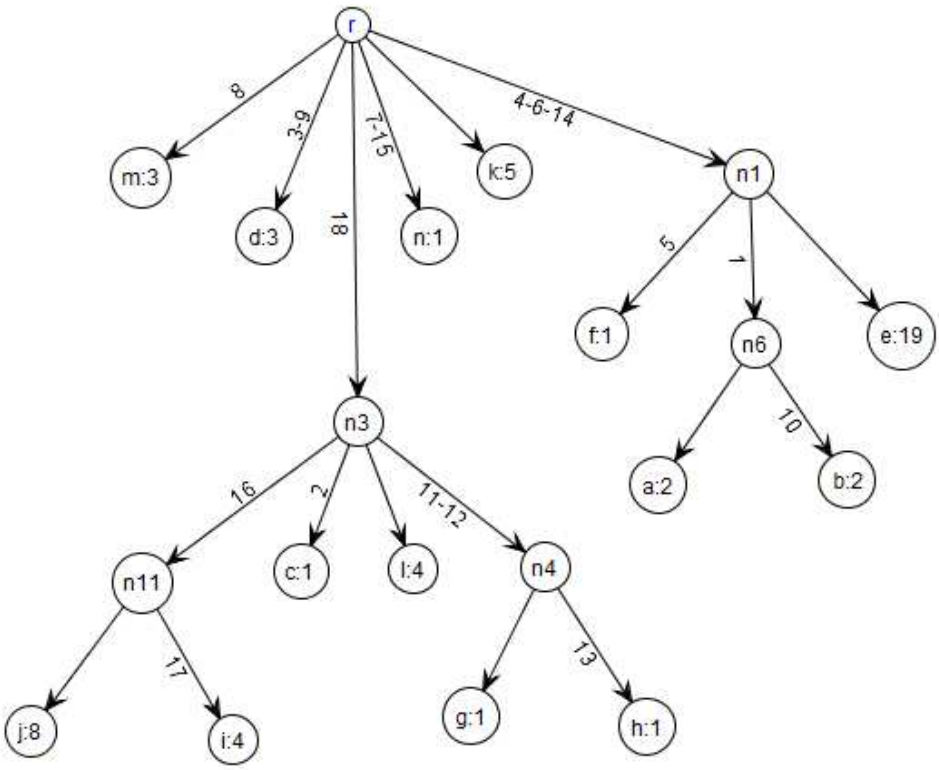


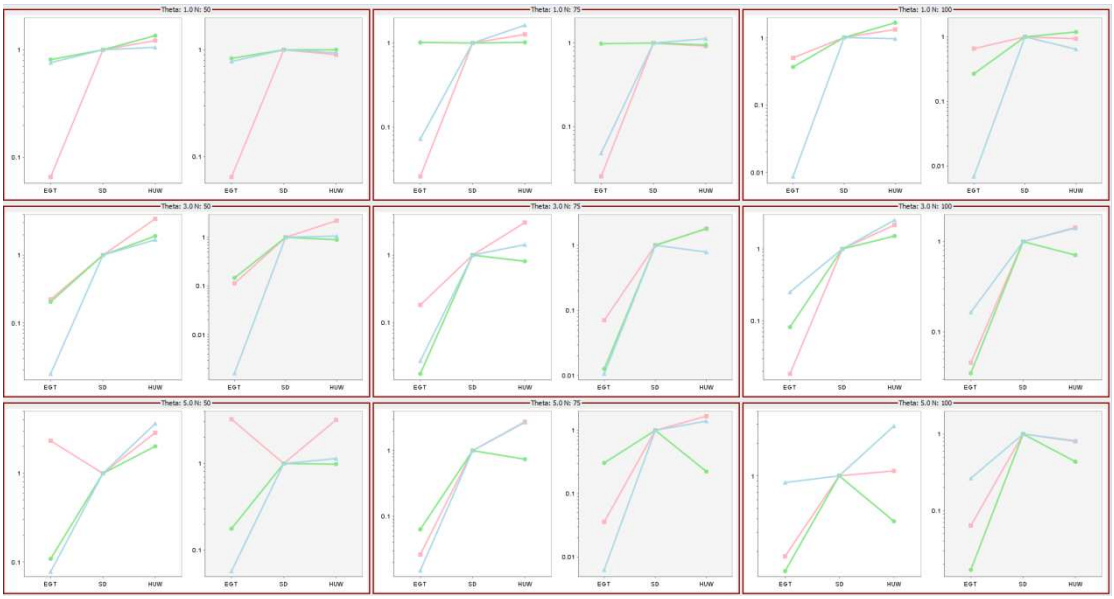
Fig 9

Title: Simulation results showing significance of time in proposal efficiency.

Legend: Simulation results showing significance of time in proposal efficiency.

It also validates the claims made by HUW for fixed sample size (see text).

Snapshot:



Tables

Table 1

Title: A sample data set for Model K69 (the infinite-sites model).

Legend: Data set for Model K69 similar to Figure 8.6 in Wakeley (2009) The characters are encoded as 0 and 1. Mutations are encoded as numbers from 1 to 4. The bolded cells are the haplotype matrix X, whose rows X_i give the characters in each haplotype. The column vector $v = (2,1,1,1)^T$ to the right of X counts each haplotype in the sample data set, so the total number of genetic samples is $n = 2+1+1+1 = 5$.

Snapshot:

Allele	Mutation				Count
	1	*2*	*3*	*4*	
a1	1	0	0	0	2
a2	1	1	1	1	1
a3	1	1	0	0	1
a4	0	0	0	0	1

Table 2

Title: Metric for the Tests and Coverage of the framework.

Legend: Metric for the Tests and Coverage of the framework. Typically, 70% coverage is considered stable.

Snapshot:

Test Area	Number of Tests	Coverage (Line)
Common	47	70%
Model	10	75%
Data	31	92%
Phylogeny	11	86%
Recursion	23	62%
Statistic	27	84%
Providers	30	61%
Importance Sampling	*62*	76%
	241 (total)	*75.75%* (avg.)

Table 3

Title: Computing MLE using multiple proposals.

Legend: MLE of the mutation rate in the range [1.0, 10.0] with an increment of 0.1 using multiple proposals; corresponding published values are included. Likelihood curve and the associated data are included in the supplementary material. See text for the reference corresponding the published values.

Snapshot:

Proposals	MLE		Sample Size	
	Published	Framework	Published	Framework
EGT	4.8	4.8	200,000	100,000
SD	[4.5, 5.0]	4.9	100,000	100,000
HUW	[4.5, 5.0]	4.9	100,000	100,000

PeerJ PrePrints | http://dx.doi.org/10.7287/peerj.preprints.395v1 | CC-BY 4.0 Open Access | received: 25 May 2014, published: 25 May 2014

Table 4

Title: Estimating likelihood at the MLE by multiple proposals.

Legend: Estimating likelihood at the MLE of population mutation rate 4.8 by multiple proposals. Corresponding published values are included. Sample size is 100,000. Exact probability is 8.71E-20 (Wu, 2009).

Snapshot:

Proposals	Published	Framework			
	Likelihood	Likelihood	Std. Error	ESS	Time(s)
EGT	7.76E-20	7.57E-20	8.31E-21	82	1,347
SD	9.33E-20	9.14E-20	5.41E-21	283	1,046
HUW	8.70E-20	9.01E-20	3.75E-21	572	2,160

PeerJ PrePrints | http://dx.doi.org/10.7287/peerj.preprints.395v1 | CC-BY 4.0 Open Access | received: 25 May 2014, published: 25 May 2014