Bioinformatics, 32(7), 2016, 1048–1056
doi: 10.1093/bioinformatics/btv716
Advance Access Publication Date: 7 December 2015
Original Paper

Genetics and population analysis

Sampling ARG of multiple populations under complex configurations of subdivision and admixture

Anna Paola Carrieri,[1] Filippo Utro[2] and Laxmi Parida[2,]*

[1]Dipartimento Di Informatica Sistemistica E Comunicazione, Università Degli Studi Di Milano-Bicocca, Viale Sarca 336, Milano, Italy and [2]Computational Genomics, IBM T. J. Watson Research, Yorktown Heights, NY 10598, USA

*To whom correspondence should be addressed.

Associate Editor: Alfonso Valencia

Received on 14 August 2015; revised on 3 November 2015; accepted on 28 November 2015

Abstract

Motivation: Simulating complex evolution scenarios of multiple populations is an important task for answering many basic questions relating to population genomics. Apart from the population samples, the underlying Ancestral Recombinations Graph (ARG) is an additional important means in hypothesis checking and reconstruction studies. Furthermore, complex simulations require a plethora of interdependent parameters making even the scenario-specification highly non-trivial.

Results: We present an algorithm SimRA that simulates generic multiple population evolution model with admixture. It is based on random graphs that improve dramatically in time and space requirements of the classical algorithm of single populations.

Using the underlying random graphs model, we also derive closed forms of expected values of the ARG characteristics i.e., height of the graph, number of recombinations, number of mutations and population diversity in terms of its defining parameters. This is crucial in aiding the user to specify meaningful parameters for the complex scenario simulations, not through trial-and-error based on raw compute power but intelligent parameter estimation. To the best of our knowledge this is the first time closed form expressions have been computed for the ARG properties. We show that the expected values closely match the empirical values through simulations.

Finally, we demonstrate that SimRA produces the ARG in compact forms without compromising any accuracy. We demonstrate the compactness and accuracy through extensive experiments.

Availability and implementation: SimRA (*Sim*ulation based on *R*andom graph *A*lgorithms) source, executable, user manual and sample input-output sets are available for downloading at: https://github.com/ComputationalGenomics/SimRA

Contact: parida@us.ibm.com

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

We address the task of modeling and simulating complex scenarios of related multiple populations with subdivision and admixture. These scenarios can be used to study the effect on the genetic profiles of extant populations as well as for testing complex hypotheses. The aim of simulations is to not only capture the resulting populations but also the relevant evolutionary history (for possible reconstruction studies). In literature, most admixture models are based on rather simplistic hypothesis of their possible inter-evolution history. One of the bottlenecks has been the sheer size of the monolithic common history of multi-populations, each of realistic size. Under these conditions simulators of even simple scenarios of just three populations often do not terminate in reasonable time in spite of meaningful parameter settings (sometimes up to 10–12 hours, for instance with

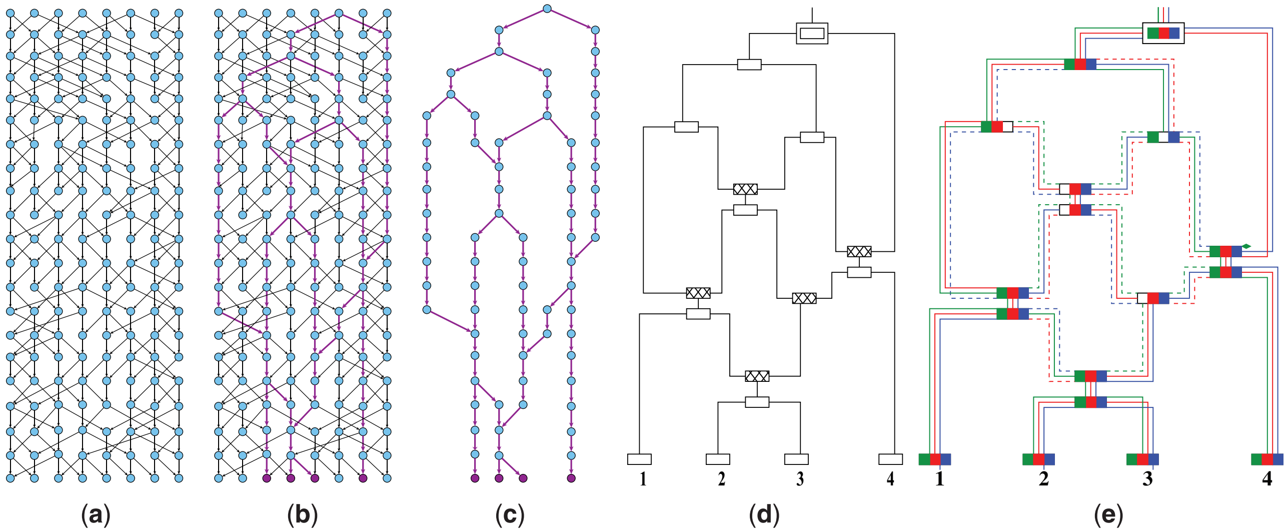


Fig. 1. (**a**) shows the complete genealogical (pedigree) graph of a Wright Fisher population of 8 at each generation. Every individual has exactly two parents. (**b**) shows the substructure of (a) based on tracking some chromosomal segment from four extant samples, marked in dark purple. The bold edges mark the flow of the genetic segments of interest to the four extant units. (**c**) shows the relevant part by removing the extraneous parts of the network of (b). Note that a forward simulator (moving in time from past to present) may have to simulate the network in (a) or (b), whereas the network in (c) is adequate for a backward simulator (moving in time from present to past). In fact a backward simulator may only construct the network in (**d**) where every node has either multiple descendants or multiple ascendants (termed the ARG). For visual clarity, the time scale has been adjusted. (**e**) The possible flow of 3 non-mixing segments in the ARG is shown in three distinct colors, green, red and blue (from left to right on the chromosomal segment). The dashed edges imply that these do not affect any of the four extant samples, due to a recombination node in their path

COSI (Schaffner *et al.*, 2005)). We observed a similar abort-and-rerun requirement in our experiments even with the classical algorithm (called *Hudson* in the paper).

We present a framework for modeling complex evolutionary scenarios and an algorithm named SimRA that is both time and space efficient enough to be practical. SimRA makes it possible to run hundreds of experiments in very short time (in minutes) enabling a very effective means of carrying out complex studies, such as in Parida *et al.* (2015). We demonstrate that the algorithm does not compromise the accuracy of the resulting simulations, all the while being very compact in its description.

1.1 Background

ARG is a directed acyclic graph (DAG) that captures the common evolutionary history of extant samples (Griffiths and Marjoram, 1997). SimRA is based on backward simulation of the ARG. Backward simulations begin in the present and move in time through the past generations and are usually more efficient than forward simulations due to the elimination of many (obvious) redundant paths in the evolution process. The big picture showing the relationship between a complete genealogical network and an ARG highlighting the backward trace of history is illustrated in Figure 1.

Complex simulation of scenarios results in complex interplay of parameters. For instance, what should be the sample size m of a population such that the expected number of active lineages in t generations is more than one. We present analytic forms of the expected ARG characteristics of a population in terms of the input parameters. These derivations use the graph-theoretic results of the random graph model presented in Parida (2010b). To the best of our knowledge, this is the first time that such closed forms of the ARG characteristics estimates have been computed. The estimated expected values can be effectively utilized by the user to design appropriate input regimens, removing time consuming trial and error iterations (see case study in Supplementary Section S5).

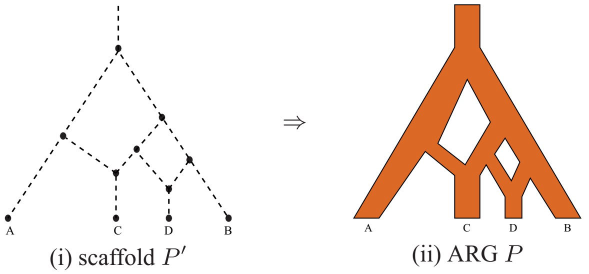
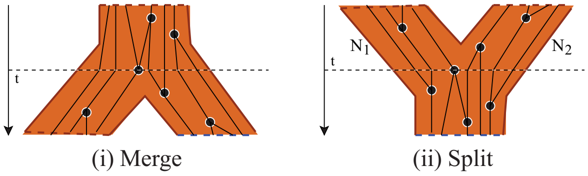


Fig. 2. An example with four populations A, B, C, D. (i) shows the scaffold P'. (ii) shows a corresponding ARG P. Note that in general the structure of P' is not apparent from P and the ARG P simply looks like the ARG in Figure 1. See text for more details

2 Modeling multiple populations

We model the relationship between m populations by a DAG P' with m leaf nodes, and call it a *scaffold*. An example is shown in Figure 2 (i). The progress of time is assumed to be from top to bottom and the m leaf nodes are annotated with the population labels. Further, each edge e in P' has three characteristics: the incubation length len(e), the number of lineages at the bottom of the edge, $l_b(e)$ and the number of lineages at the top of the edge, $l_t(e)$. The length is a time parameter defined in generations. Note that two parameters, an effective population size and a recombination rate, determine the number of lineages $l_t(e)$ for a fixed pair of values of $l_b(e)$ and len(e). We assume that the scaffold P' is binary (i.e., each internal node in P' has exactly two ascendants or two descendants, but not both). For each internal node, the *junction constraints* are defined as



(i) Merge (ii) Split

Fig. 3. The two types of nodes in a scaffold P': Merge and split nodes are marked by the horizontal dashed lines at time t. The tiny black disc nodes and the thin black edges are part of the underlying ARG P

follows (see also Figure 3). A node v in P' that has two incoming edges e_1 and e_2 and an outgoing edge e_3 (termed a *split node*) the following relationship holds: $l_t(e_3) \le l_b(e_1) + l_b(e_2)$, i.e., the lineages at v is the union of the lineages of the two incoming edges. Similarly if node v has two outgoing edges e_1 and e_2 with one incoming edge e_3 (termed a *merge* node), then $l_b(e_3) \le l_t(e_1) + l_t(e_2)$, i.e., the lineages at v is the union of the lineages of the two outgoing edges. Each edge e of P' represents the evolution of a Wright Fisher population captured in a DAG say P_e. The union of each of these DAGs by appropriately gluing the ends of the edges corresponding to the nodes of P' gives the ARG P that can be written as: $P = \cup_{e \in P'} P_e$. Such a P is shown in Figure 2 (ii) where the leaf nodes correspond to extant units of each population of P'.

Figure 4 shows an example of parameters that define the scaffold P'. Additionally a recombination rate (r) and effective population size for each edge, ultimately decides the topology of the resulting ARG. Further, the mutation rates and the short tandem repeats (STR) details define the polymorphism in the samples of the individuals of the populations.

Consider the scaffold specified in Figure 4. Each edge is simulated as a single population (Section 3) and assume that the effective populations size N_{e_j} is specified for each edge e_j. For instance, the edge e_1 labeled with population D is simulated with 12 extant samples, i.e., $l_b(e_1) = 12$, $len(e_1) = t_1$. The resulting surviving lineages $l_t(e_1)$ is split in the ratio 1:1 as shown in Figure 3. Similarly, the edge e_2 is labeled with population C with 18 extant samples, $l_b(e_2) = 18$, $len(e_2) = t_2$. The resulting surviving lineages $l_t(e_2)$ is split in the ratio 1:1. Next, $l_t(e_1)/2$ lineages are simulated until a time depth of t_3 on edge e_3 to give $l_t(e_3)$ lineages. Population B is simulated with 14 extant samples on edge e_4 until a time depth of t_3, i.e., $l_b(e_4) = 14$, $len(e_4) = t_4$. The $l_t(e_3)$ lineages are combined with $l_t(e_4)$ lineages. This node in the scaffold is a merge node as shown in Figure 3 and the population is simulated to a time depth of t_6 (i.e., for time $t_6 - t_3$). Similarly all the edges are simulated until a total time depth of t_7.

3 ARG network sampling algorithm

We now address the problem of simulating each edge of the scaffold P' which can also be viewed as the ARG sampling of a single population. This is a well-studied problem as discussed in (Hein et al., 2004; Hudson, 2002). In the remainder of the paper, we refer to this classical backward algorithm, based on Kingman coalescence, as Hudson (Kingman, 1982). The reader is directed to (Hein et al., 2004; Hudson, 2002) and citations therein for a comprehensive description of Hudson. However, we found that even Hudson algorithm was not efficient enough to admit complex scaffold simulations: it was too time consuming and in many instances failed to terminate in reasonable time, forcing to abort and re-run. A single scaffold requires multiple runs (corresponding to each edge) thus making the algorithm prohibitively expensive. Here we present our algorithm for simulating a single (neutral) population.

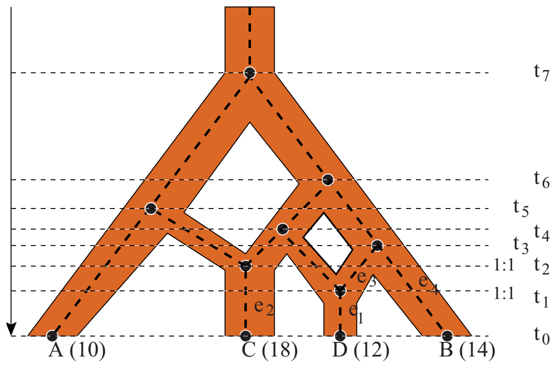


Fig. 4. Specifying the family of 4 populations, A, B, C and D with sample sizes 10, 14, 18 and 12 respectively. The horizontal dashed lines correspond to time $t_0 = 0 < t_1 < t_2 < ... < t_7$. At times t_1 and t_2 the surviving lineages are split in the ratio 1:1 along the diverging lines of the scaffold at the split nodes

3.1 Overview of the approach

The algorithm is based on Kingman coalescence and is along the lines discussed in (Hein et al., 2004) and is very similar to the Hudson algorithm. Again, to keep the discussion self-contained we give the complete description here, along with the changes specific to SimRA in the text of the description below. The algorithm works back-in-time starting from the present (time 0), moving back into the past. Further, the ARG is incrementally constructed by identifying the *event* nodes in the graph. An event node either has multiple incoming or multiple outgoing edges. For example a chain node is not an event node. An important assumption, that considerably simplifies the algorithm, is made: The probability of multiple events in the same epoch (generation) is extremely low, hence the algorithm assumes there is at most one event per generation. The design of the overall algorithm is affected by this and at each step the algorithm simply seeks the closest generation from the current where an event node occurs.

3.1.1 The closest event node from the current state in the ARG

In the interest of brevity, some basic definitions are presented in Supplementary Section S1 and use here. Let L lineages be active at time T. Let t^{coal}_{ab} denote the time to the coalescence of lineages l_a and l_b. Let t^{rcmb}_{l} denote the time to the closest (to T) recombination event of lineage l. Eq. 23 (in Supplementary) shows that each of the $\binom{L}{2}$ coalescent events, generically written as t^{coal}_{ab}, can be approximated by an exponential distribution with parameter $\lambda = 1$. Recall the following observation from (Parida, 2010b):

Observation 1. (**Ancestor Without Ancestry Paradox**) *The edges (and nodes) of an ARG must be annotated with the chromosomal segment that flows through the edge.*

Based on the above observation, Eq. 26 (in Supplementary) can be approximated by an exponential distribution with parameter r'_l where $r'_l = Nr_l$ and r_l is the recombination rate of the segment flowing through lineage l. These approximations to the exponential distributions are based on two assumptions (Wright Fisher population): the population at each generation is N and a unit picks its parent randomly from the previous generation (non-overlapping generations and panmictic mating population). Also, note that the factor of N in r'_l is due to the approximation of the distributions, and not due to the underlying population evolution model. The task is to find t, the time to the closest event node in the past. This event could either be a coalescent event (merging of two lineages) or a recombination event (splitting of a lineage). Since all the events are independent, then we seek overall minimum. Thus

$$t = \min\Big(\overbrace{\min_{1\le a<b\le L}(t_{ab}^{\text{coal}})}, \underbrace{\min_{1\le l\le L}(t_l^{\text{rcmb}})}\Big)$$
$$= \operatorname{Exp}\Big(\overbrace{1+1+\cdots+1}+\underbrace{r'_1+r'_2+\cdots+r'_L}\Big) \qquad (1)$$

using Property 1 (Supplementary Section S1). The overbraces capture the $\binom{L}{2}$ coalescent events and the underbraces capture the L recombination events. t computes the time to the closest event back in time from the current time T, but, is the closest event coalescent or recombination? The answer to this comes from Property 2 (Supplementary Section S1). The event is a coalescent event with probability

$$\frac{\binom{L}{2}}{\binom{L}{2}+\sum_l {r_l}'} \qquad (2)$$

and a recombination at lineage $1 \le k \le L$with probability

$$\frac{{r_k}'}{\binom{L}{2}+\sum_l {r_l}'} \qquad (3)$$

In the implementation of the algorithm, both Eqs. 2 and 3 are used in a single draw of a random number. Imagine a unit interval $[0,1]$ is broken up into $1+L$ sub intervals of lengths in the following ratio $\binom{L}{2}$: $r'_1 : r'_2 : \ldots : r'_l : \ldots : r'_L$. Thus a random number drawn from the interval $[0,1]$ belongs to one of these $1+L$ sub-intervals and is appropriately interpreted: the first interval implies coalescent event and kth ($k>1$) interval implies a recombination at the lineage l_{k-1}. Since the events are random, t is estimated first and then the lineages are picked at random from the L active lineages.

3.1.2 Genetic material flowing through the ARG

The chromosomal segment whose evolution history is captured by the ARG is represented as the real interval $[0,1]$, without loss of generality. Every node in the ARG is annotated with union of one or more sub-intervals of $[0,1]$. Thus genetic material, I, carried by a node is: $I = \{[\ell_1,u_1],[\ell_2,u_2],\ldots,[\ell_s,u_s]\}$, where $0 \le \ell_1 < u_1 < \ell_2 < u_2 < \ldots < \ell_s < u_s \le 1$. The closed intervals $[\ell_i,u_i] \in I$ are termed *solid* and the open intervals (u_i,ℓ_{i+1}) are termed *gaps* where $1 \le i < i+1 \le s$. The length (len) of I is defined as the total span of I, irrespective of the gaps, while density (den) of I is defined as the total span of the solid intervals only. The definitions are summarized as (see also Fig. 3):

$$\text{solid}(I) = [\ell_1,u_1]\cup[\ell_2,u_2]\cup\ldots\cup[\ell_s,u_s],$$
$$\text{gaps}(I) = (u_1,\ell_2)\cup(u_2,\ell_3)\cup\ldots\cup(u_{s-1},\ell_s),$$
$$\text{len}(I) = u_s-\ell_1,$$
$$\text{den}(I) = \sum_{i=1}^{s} u_i-\ell_i,$$

$$[x,y]\subset\text{solid}(I) \iff [x,y]\subset[\ell_i,u_i],\ \text{for some } 1\le i\le s.$$

The union operation on segments, $I_a \cup I_b = I_{a\cup b}$, has the natural interpretation:

$$[\ell,u]\in I_{a\cup b} \iff [\ell,u]\subset\text{solid}(I_a)\ \text{OR}\ [\ell,u]\subset\ \text{solid}\,(I_b). \qquad (4)$$

The splitting of I at x ($\ell_1 \le x \le u_s$) into I_a and I_b is defined as:

$$I \overset{\text{split}}{\rightarrow} \begin{cases} I_a = I_b = I, \text{when } x=\ell_1 \text{ or } x=u_s, \\ \left.\begin{array}{l} I_a = \{[\ell_1,u_1],[\ell_2,u_2],\ldots,[\ell_j,x]\}, \\ I_b = \{[x,u_j],[\ell_{j+1},u_{j+1}],\ldots,[\ell_s,u_s]\}, \end{array}\right\} \text{when } \ell_j < x < u_j, \\ \left.\begin{array}{l} I_a = \{[\ell_1,u_1],[\ell_2,u_2],\ldots,[\ell_j,u_j]\}, \\ I_b = \{[\ell_{j+1},u_{j+1}],\ldots,[\ell_s,u_s]\}, \end{array}\right\} \text{when } u_j \le x \le \ell_{j+1}. \end{cases} \qquad (5)$$

3.2 On the uniqueness of GMRCA

A founding ancestor of the extant units is termed GMRCA (grand most recent common ancestor). Let Ω denote the set of all (infinite) graphs, with nodes partitioned into distinct levels, or generations, with N nodes at each level, and each node having no more than two parents. For each $X \in \Omega$, and any subset V of the nodes at level 0, there is an induced subgraph of X, namely the ARG induced by V and we call this the ARG associated with X.

Following Parida (2010a) we introduce a probability measure on Ω as follows. For $X \in \Omega$ and $h > 0$, we denote by X_h the *truncation* of X to depth h, i.e., X_h is the finite induced graph from X on the set of vertices of level $\le h$. Similarly, for a subset $E \subset \Omega$, and $h > 0$, we denote $E_h = \{X_h | X \in E\}$. We say that E is *finitely determined* if there exists some h_0, such that $X \in E \iff X_{h_0} \in E_{h_0}$, and in this case we denote $\mu(E) = \frac{|E_{h_0}|}{|\Omega_{h_0}|}$. The family, $\mathcal{F}$, of finitely determined subsets $E \in 2^{\Omega}$ clearly forms a field, and thus by the Caratheodory extension theorem (see for example Varadhan (2001), Theorem 1.1, pp. 4), μ can be uniquely extended to the σ-field generated $\mathcal{B}$ by this family. We denote this measure also by μ and consider Ω as a probability space with measure μ.

Let $E^{\text{unq}} \subset \Omega$ be the set of graphs $X \in \Omega$, such that the ARG associated to X has a unique GMRCA. The following theorem (whose proof is presented in the Supplementary due to space constraints) follows from the definition of the measure μ. It assures us that almost every ARG has a unique GMRCA. In fact, in over ten thousand simulations, of which about three thousand are reported in this paper, SimRA terminated in every instance with a unique GMRCA.

Theorem 1. *The subset E^{unq} is measurable and $\mu(E^{\text{unq}}) = 1$.*

Corollary 1. *The measure of the space of all ARGs with no unique GMRCA is zero.*

3.3 Algorithm to generate the topology

INPUT: Due to historical reasons, the unit of recombination rate is specified in centiMorgans per megabase per generation and the mutation rate is specified in number of mutations per base pair per generation ($\times 10^{-8}$). The input parameters and some typical parameter values for a human chromosomal segment are given below.

	Parameters	user-specified units	units in bp for algorithm	example values
g	seqment length	Kb	$\times 10^3$ bp	75×10^3
V	STR locations	–	–	$[0.3, 0.7] \times g$
m	extant units	–	–	100×1
N	population size	–	–	10×10^3
		rates/generation		
r	recombination	cM/Mb/gen	$\left\{ \times\left(\frac{0.01}{10^6} = 10^{-8}\right)\right.$ Morgan/bp/gen	0.1×10^{-8}
μ	SNP mutation	mut/bp/gen $\times\ 10^{-8}$	$\times 1$ mut/bp/gen	1.5×10^{-8}
μ^{str}	STR mutation	mut/locus/gen $\times\ 10^{-4}$	$\times 1$ mut/locu/gen	6.9×10^{-4}

ASSUMPTION: Not more than one event, coalescent or recombination, occurs at a generation. Also, no back mutations, i.e., a position undergoes no more than one mutation in the entire ARG. The mutation rate and recombination rate are uniform over the segment being simulated.

OUTPUT: ARG; L is the number of GMRCAs.

```
Algorithm

1. Initialization.
   a. The genetic material, $I_v$, of each of the $m$ leaf nodes, $v$,
      is set to $I_v = \{[0,1]\}$.
   b. The number of live lineages $L$ is initialized to $m$.
   c. For lineage $l$, incident on leaf node $v$, the recombination
      rate (based on Eq. 26 in Supplementary Section S1) is:

         $r_l' = N r_l \text{ where } r_l = g r \text{len}(I_v).$        (6)

      Since, len $(I_v) = 1$ for the leaf nodes, for each $l$, $r_l' = \alpha$
      where

         $\alpha = Ngr.$        (7)

   d. Time $T$ is set to 0 and iteration $i$ to 1.

2. Loop. Iterate until $L$ is one (or $T$ crosses a pre-defined
   threshold). Iteration $i$ is defined as follows.
   a. Compute the recombination rate $r_l'$ of each lineage $l$
      (the outgoing edge on node $v$) using Eqs. 6 and 7 as
      $r_l' = \alpha \times \text{len}(I_v)$. Then compute the time $t_i$ to the next
      event using the exponential distribution (Eq. 1):

         $t_i = \text{Exp}\left(\binom{L}{2} + \sum_{l=1}^{L} r_l'\right).$        (8)

      In other words, draw a random number from the
      above exponential distribution.
   b. Based on Eqs. 2 and 3, if coalescent event, then pick
      two lineages, $l_a$ and $l_b$ (with genetic material $I_a$ and
      $I_b$ respectively) at random and coalesce them to one
      and update the genetic material of this new node
      and lineage $I_a \cup I_b$ (as defined in Eq. 4). Update $L$ to
      $L - 1$.
   c. If recombination at lineage $k$, then randomly pick a
      point $x$ on the segment being carried by lineage $k$, split-
      ting the lineage into two, as defined in Eq. 5. Update
      the genetic material of the two lineages based on this
      splitting point. Update $L$ to $L + 1$.
   d. $T$ is updated as $T + t_i$ and iteration as $i + 1$.
```

3.4 Painting ARG edges with SNP & STR mutations

Each time t of Step 2(a) can be scaled to generation as $j = Nt$. Let time t be associated with an incoming edge on node v. At this stage, each edge is annotated with the mutation events, which is appropriately reflected in I_v, the segment carried by node v.

3.4.1 SNP mutations

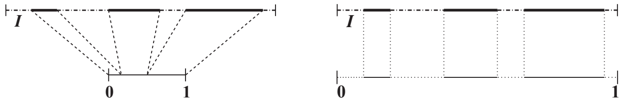
Since number of generations is Nt and the span of the segment I_v has been normalized in the initialization step, let

$$p = \mu N t \text{ and } n = g\text{den}(I_v). \quad (9)$$

Each edge of the ARG, incoming on node v, is annotated with number of mutations based on Eq. 9 as follows. X, the random draw from a Poisson distribution with parameter np:

$$X = \text{Poisson}\,(np). \quad (10)$$

Then the X mutations are placed at random in segment I_v (excluding the gaps, see Fig. 5).



(a) Using den(I) in Eq 9. (b) Using len(I) in Eq 9.

Fig. 5. The top line represents a chromosomal segment I carried by an edge in both (a) and (b): gaps(I) are shown as dashed lines and solid(I) as solid segments. I is mapped to a normalized line segment, say $[0,1]$, shown in the bottom in both (a) and (b). In (a) the gaps are skipped, and the lengths of each element in solid(I) is proportionally represented in $[0,1]$. Thus any element in $[0,1]$ can be mapped back to a unique location in solid(I). In (b) the gaps are not skipped, and the lengths of each element in solid(I) and in gaps(I), is proportionally represented in $[0,1]$. Any element only in the solid section in $[0,1]$ can be mapped back to a unique location in solid(I); any other element maps to a gap in I

3.4.2 STR mutations

Note that the number and positions of the STR loci are fixed by the input specification. For each STR locus, k, carried by I_v, we compute the following. The number of STR mutations at locus k on each edge of the ARG, incoming on node v, is X_v, the random draw from a Poisson distribution with parameter $Nt\mu^{\text{str}}$:

$$X_k = \text{Poisson}\,(Nt\mu^{\text{str}}). \quad (11)$$

Let p_+ be the probability of the mutation that increases the number of copies (by 1 in one generation) and p_- be the probability of the mutation that decreases the number of copies (by 1 in one generation). Then, X_{k_+}, the number of times the STR mutation results in an increase in the number of copies of the repeat follows a binomial distribution, hence is the random draw from a binomial distribution with parameter X_k and p_+ $X_{k_+} = \text{Binomial}\,(X_k, p_+)$. Thus the remainder, i.e., $X_k - X_{k_+}$ must be the number of events that result in decrease of the number of copies. Thus Δ_k the net change in the number of copies at locus k is: $\Delta_k = X_{k_+} - (X_k - X_{k_+}) = 2X_{k_+} - X_k$. If unspecified, we use the default value of $p_+ = \frac{1}{2}$, assuming $p_+ = p_- = \frac{1}{2}$.

4 Four quantitative hallmarks of ARG

The ARG is a random object defined by parameters, m, the extant sample size; N, the population size; g, the length of the genomic segment whose common history is being tracked; r the recombination rate; μ, SNP mutation rate. In fact, other polymorphisms, such as STR, can be incorporated just as the SNP mutations are. Note that the unique founding ancestor, GMRCA, is attained in an ARG with probability 1 (see Section 3.2). We consider the following four quantities as the hallmark of the random object ARG with parameters m, N, g, r, μ:

1. Depth of the ARG (H).
2. Number of non-mixing segments in the sample population (Z).
3. Number of polymorphic sites in the sample population (Y).
4. Diversity in the sample population (D).

5 Closed-form approximations of the expected hallmark values

We derive approximations of the expected hallmark values as closed-form functions of the ARG parameters. We did not find analytic or closed-forms of the expected values for the general scenario in literature, except some very specialized cases such as depth of

GMRCA in the absence of recombinations (Hein et al., 2004). Our derivations are based on the theorems and observations in Parida (2010a).

In fact, we found that if we required a single population only to study the hallmark expected values, but not the sample population, then the closed form approximations were tight enough to make the actual simulation redundant.

Overview of the derivations. We use two notions: *depth* of a node and *girth* of an edge. An edge length, as well as depth of a node, is defined to be in time units. The unit of time is measured in generations. The depth of each node is measured from the leafnodes and the depth of a leafnode is defined to be 0. The *girth* of an edge is defined to be the product of the edge length and the size of the genomic segment the edge transmits.

1. The ARG network is decomposed into overlapping trees (see Thm 2; also Figs 6 and 7 and Fig. 11 in Supplementary).
2. For each tree, we compute the depth of each node and the girth of each edge, using Kingman coalescence. The depth of a tree is simply the depth of its root node. The girth of the tree is the sum of the girth of each edge of the tree.
3. The depth and girth of each tree are used for approximating the ARG hallmark values. However, the interdependence of the trees makes these computations non-trivial.

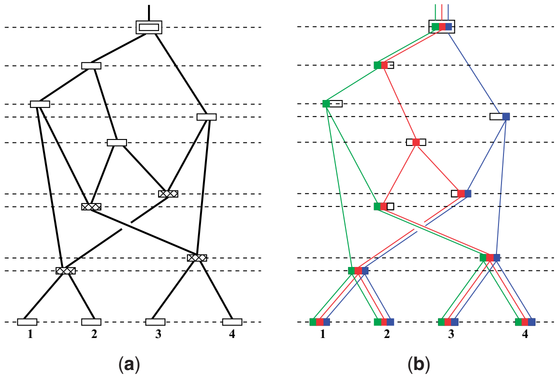


Fig. 6. The horizontal lines denote the time at which an event (coalescence or recombination) occurs. (**a**) An example of an ARG. (**b**) depicts the flow of the 3 non-mixing segments shown in three distinct colors: green, red and blue (from left to right on the chromosomal segment)

5.1 Mathematical details

To keep this section self-contained, we recall the following basic identities. Let $0 < m' < m$.

$$\sum_{i=m'+1}^{m}\left(\frac{1}{i-1}-\frac{1}{i}\right)=\frac{1}{m'}-\frac{1}{m}, \tag{12}$$

$$\sum_{i=2}^{m}\frac{1}{i-1}=1+\frac{1}{2}+\ldots+\frac{1}{m-1}\approx \log m, \tag{13}$$

$$\sum_{i=m'+1}^{m}\frac{1}{i-1}\approx \log m-\log m'=\log\frac{m}{m'}. \tag{14}$$

Consider a tree with m leafnodes. Using Kingman coalescence, all the non-leaf nodes of the tree can be written in increasing depth (from the leafnodes) as $v_1, v_2, \ldots, v_{m-1}$, with the active lineages decreasing by one at each node. Let t_i denote the depth of v_i from v_{i-1} where depth of v_0 is defined to be 0. Then the tree truncated at a depth that has m' active lineages, is written as $T_{m,m'}$. Let $H_{T_{m,m'}}$ be the depth of this tree. Then using Property 3 (Supplementary Section S1), linearity of expectations, and the above identities we get:

$$\begin{aligned}\mathbb{E}(H_{T_{m,m'}}) &= \sum_{i=1+m'}^{m}\mathbb{E}(t_i)=\sum_{i=1+m'}^{m}\frac{1}{\binom{i}{2}}\\ &=2\sum_{i=1+m'}^{m}\left(\frac{1}{i-1}-\frac{1}{i}\right)=2\left(\frac{1}{m'}-\frac{1}{m}\right).\end{aligned} \tag{15}$$

Let g the length of the genomic segment carried by each edge in the tree and the girth of the tree be $wt_{T_{m,m'}}$. Then

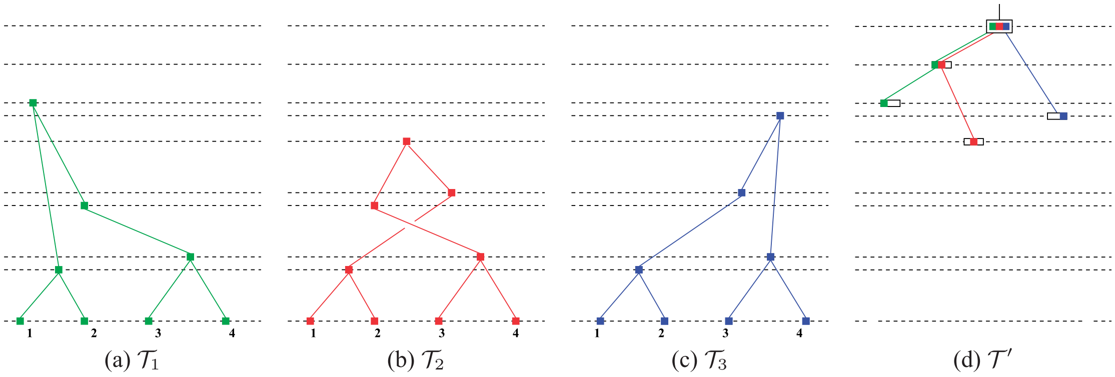
$$\begin{aligned}\mathbb{E}(wt_{T_{m,m'}}) &= \sum_{i=1+m'}^{m} i\mathbb{E}(t_i)g\\ &= g\sum_{i=1+m'}^{m}\frac{i}{\binom{i}{2}}=2g\sum_{i=1+m'}^{m}\frac{1}{i-1}\end{aligned} \tag{16}$$

$$\approx 2g\log\frac{m}{m'}. \tag{17}$$

The complete tree with a single root node is written as $T_{m,1}$ and

$$\mathbb{E}(H_{T_{m,1}})=2\left(1-\frac{1}{m}\right), \tag{18}$$

$$\mathbb{E}(wt_{T_{m,1}})=2g\log m. \tag{19}$$



(a) $\mathcal{T}_1$ (b) $\mathcal{T}_2$ (c) $\mathcal{T}_3$ (d) $\mathcal{T}'$

Fig. 7. The four trees embedded in the ARG network of Figure 6 (as captured in Eq. 20). The three marginal trees, $\mathcal{T}_1$ to $\mathcal{T}_3$ correspond to the three non-mixing segments shown in green, red and blue

We recall the following from Parida (2010a) relating population genetics entities with graph entities like least common ancestor (LCA). A non-mixing genetic segment does not have any recombination event in the common history of the m samples.

Theorem 2 *Let $\mathcal{G}$ be an ARG with some $K \geq 1$ non-mixing segments. Then K marginal trees are embedded in $\mathcal{G}$ and the GMRCA of $\mathcal{G}$ is the LCA of the K LCAs of the K marginal trees.*

Figure 6 gives a simple illustration on an ARG on four samples with three non-mixing segments. An alternative view of the theorem is as follows: Let $\mathcal{T}(l, b)$ denote a tree defined on l leafnodes each carrying the segment of length b. Then for some partition of genome segment g into K non-overlapping segments, where $g = g_1 \cup g_2 \cup \ldots \cup g_K$,

$$\mathcal{G} \approx \left(\bigcup_{k=1}^{K} \mathcal{T}_k(m, g_k) \right) \cup \mathcal{T}'(K, g), \tag{20}$$

where the roots of the $\mathcal{T}_k$'s are the leaves of $\mathcal{T}'$. Two examples illustrate the embedded trees, one in Figure 7 and the other, due to space constraints, in the Supplementary in Figure S11. In the former the number of nodes in $\mathcal{T}'$ is of the order of K while the latter has the smallest possible size of just one node. In both, $\mathcal{T}_1$-$\mathcal{T}_3$ are the marginal trees of usual shape and size.

Corollary 2. *If H_1 is the maximum of the depths of $T_k(m, g_k)$and H_2 is the depth of $\mathcal{T}'(K, g)$and H is the depth of the GMRCA of the $\mathcal{G}$, then*

$$H = H_1 + H_2. \tag{21}$$

Corollary 3. *The girth of ARG $\mathcal{G}$ is the sum of the girth of each $\mathcal{T}_k(m, g_k)$ and $\mathcal{T}'(K, g)$.*

5.2 Summary of closed-form formulations

Due to space constraints, the derivations have been presented in Supplementary Section S2 and only the results are summarized here.

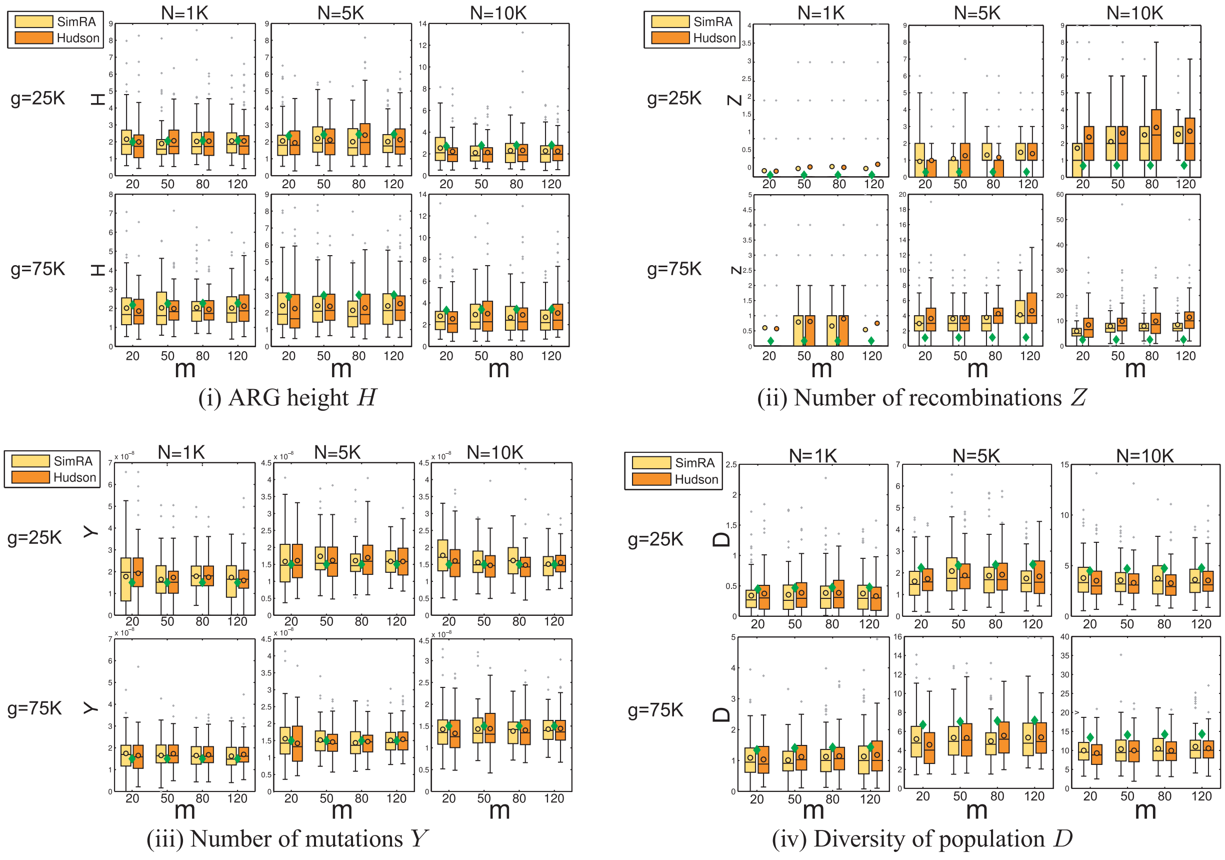
Let

$$H_1 = 2\left(1 - \frac{1}{m}\right),$$

$$\alpha = Nrg, \beta = \alpha H_1 + 1, \gamma = 2\alpha - \beta,$$

$$H_2 = \frac{\gamma + \sqrt{\gamma^2 - 8\alpha(1-\beta)}}{2\alpha}, \text{ when } r > 0,$$

$$K = (H_1 + H_2)Nrg + 1.$$



(i) ARG height H

(ii) Number of recombinations Z

(iii) Number of mutations Y

(iv) Diversity of population D

Fig. 8. The closed form expected values are compared against empirical values for different parameter values of m, g and N. The recombination rate used is $r = 0.1 \times 10^{-8}$ Morgan/bp/generation, $\mu = 1.5 \times 10^{-8}$ mutations/bp/generation, and $m' = 1$. For the results with $m' > 1$, see Figure S12 in the Supplementary. Note that the mutation rate affects only (iii) and (iv). Each experiment was run 100 times, using both SimRA and Hudson. The box-and-whisker diagram summarizes the result for each. On each box, the central mark is the median, the circle is the mean, the edges of the box are the 25th and 75th percentiles, the whiskers extend to the most extreme data points not considered outliers, and outliers are plotted individually. In each, the green diamond is the expected value as computed by the closed form, while the hollow circles are the observed empirical values by SimRA and Hudson. Notice that not only do the two algorithms give similar values; the closed form is also a tight approximation

Then the four (expected) hallmark values are:

$$\mathbb{E}(\mathrm{H}) \approx H_1 + H_2,$$

$$\mathbb{E}(Z) \approx K - 1,$$

$$\mathbb{E}(Y) \approx 2\mu g N(\log m + \log K),$$

$$\mathbb{E}(D) \approx 2gN\mu \sum_{i=2}^{m} \frac{1}{i^2}.$$

Expected Height of Truncated ARG. H_1, α and β values are as above and m' is the number of surviving lineages of the truncated ARG. Then

$$\gamma = 2\alpha - m'\beta,$$

$$H_2 = \frac{\gamma \pm \sqrt{\gamma^2 - 8\alpha m'(m' - \beta)}}{2\alpha m'}, \text{ when } r > 0,$$

$$\mathbb{E}(\mathrm{H}) \approx H_1 + H_2.$$

6 Comparison study

For a comprehensive survey of sampling algorithms and simulators the reader is directed to (Hoban *et al.*, 2012), which discusses both backward and forward simulators.

Many simulators in literature address the issue of redundancy in the simulations and they can also be classified on the basis of the extent of non-redundancy (see Parida, 2012, for instance).

The underlying mathematics of a backward simulator is nontrivial and the classical Hudson algorithm captures the essence of backward simulations.

SimRA simulates multiple populations under admixture and subdivision, while other simulators incorporate other demographic models, making it difficult for a nose-to-nose comparison. However, the core engine of SimRA can be compared with the Hudson algorithm, which forms the basis in all backward simulators. Hence in the comparative study here, we use only the single population of SimRA and the classical Hudson algorithm using exactly the same input parameters. Furthermore, to keep the comparisons agnostic to other possible extraneous factors, we use identical implementation for the common parts of SimRA and Hudson.

6.1 Differences from Hudson algorithm

Recall that Hudson algorithm uses a single scaled recombination rate Nr, while SimRA uses the L segmented versions $r'_l = Nr_l$, $l = 1, .., L$. This is reflected in Eqs. 1–3. Eq. 1 suggests that in our algorithm, to account for recombination event, time t takes into account not just the number of active lineages but also the size of the segments carried by each of them ($r_l = gr\mathrm{len}(I_v)$ of Eq. 6). Note that $\sum_v \mathrm{len}(I_v) \neq 1$ at each iteration making the two computations distinct; hence distinct algorithms. Thus if p_c is the probability of coalescence (Eq. 2), then the probability of recombination is 1 $-p_c$ with equal probability over all the lineages in Hudson algorithm. But SimRA uses Eq. 3 to pick a lineage for the recombination event. Thus Eq. 3 has no counterpart in the classical Hudson algorithm. The accuracy of the two algorithms are comparable while SimRA outperforms Hudson in time, space and non-redundancy factor, as seen below. We performed extensive comparative analysis between the two algorithms SimRA and Hudson, to measure various outcomes. In particular, we carried out hundred runs for each parameter set up, for both the algorithms.

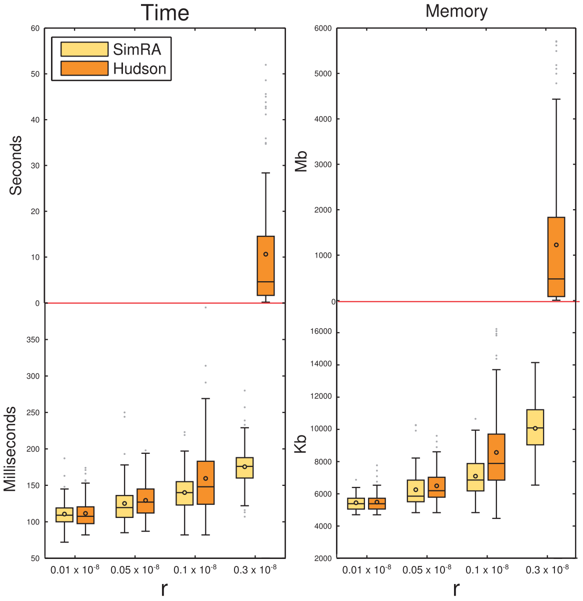


Fig. 9. The box-and-whisker diagrams of the time and memory performance of SimRA and Hudson computed for 100 runs with N = 10K, g = 150K, $\mu = 1.5 \times 10^{-8}$ and different value of recombination rates (shown in the x-axis) and for each parameter setting both the algorithms were run 100 times. The red line demarcates the time and space requirement for Hudson for the rightmost value of *r*. While all values of *r* the time and space requirement of SimRA is better than that of Hudson. In particular for large *r* SimRA is nearly two orders of magnitude better than Hudson

6.2 Accuracy measures

We demonstrate the accuracy of the SimRA algorithm by comparing the four hallmark values to the ones computed by Hudson. The results are shown in Figure 8. Notice that the SimRA and the Hudson estimates are very close to each other, over 100 runs for each configuration. We use the same set-up to compare the closed form expected values of the last section to the observed empirical values. Again, note the tightness of the approximations. In addition, Figure 12 in the Supplementary shows the expected height of truncated ARG compared against the observed values using both SimRA and Hudson, with similar accuracy.

6.3 Time and space performance

Figure 9 shows the results of the comparative time and space performances. SimRA shows consistently superior performance in both time and space, and, the difference is particularly accentuated with increasing values of recombination rate *r*. For higher values of *r*, the time and space requirement is nearly two orders of magnitude higher for Hudson.

In particular, for the study summarized in Figure 10, the Hudson algorithm had to be aborted and re-run several times and it took over six months just to complete, while SimRA was done with the four hundred runs in less than half a day.

6.4 Non-redundant ARGs of SimRA

How redundant is the ARG network sampled by an algorithm? If we assume that all the marginal trees and the resulting samples are the *essential content* of a simulation, then it is meaningful to ask what portion of the ARG resulting from a simulation has no contribution to the essential content. This is formally studied as the minimal descriptor in Parida *et al.* (2011) and other details pertaining to

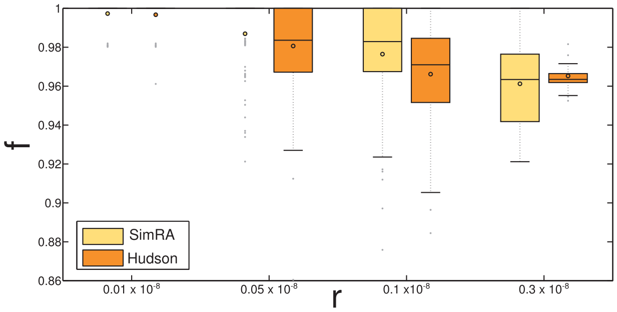


Fig. 10. The box-and-whisker diagrams of the compaction factor *f*, for different values of *r*. The other parameter values are the same as used in Figure 9 and for each parameter setting both the algorithms were run 100 times. ARGs produced by SimRA are consistently more compact (or less redundant) than that of Hudson

the execution of the experiments are presented in Supplementary Section S4.

Let the compaction factor *f* be defined as the ratio of the number of nodes in the minimal descriptor to the number of nodes in the original ARG as in (Utro *et al.*, 2013). Thus the closer the value of *f* to 1, the less redundant is the ARG and thus more compact. Figure 10 shows that the ARGs produced by SimRA are systematically more compact than the ones produced by the Hudson algorithm. This is particularly accentuated for higher values of *r*.

7 Conclusion

The design of the SimRA algorithm was influenced by the implications of the *Ancestor Without Ancestry Paradox*, which also paved the way for computing the closed forms of the expected values of the ARG characteristics. To the best of our knowledge this is the first time analytic formulae have been given for an ARG.

Such closed-forms, apart from mathematical completeness, also serve multiple practical purposes.

Ironically, it obviates single population simulations in many situations where the interest is only in the characteristic estimates. In others it provides a framework for evaluating correctness of the ARG sampling algorithms. Also, for complex scenarios such as the ones with population scaffold architectures it aids in parameter specification (this is illustrated in Supplementary Section S5).

Through extensive comparison studies, we demonstrated that the ARGs produced by SimRA are more compact, more efficient in time and space, without compromising accuracy. Currently we are looking into extending SimRA to incorporate other demographic models including selection.

Acknowledgments

We are thankful to Saugata Basu for his insights on measure theory. We are also grateful to Daniel Platt for many useful discussions, particularly on STR variations in populations. Part of the SimRA implementation was carried out by APC when she was visiting IBM Watson Research Center.

Conflict of Interest: none declared.

References

Griffiths,R.C. and Marjoram,P. (1997) An ancestral recombinations graph. In: Donnelly,P. and Tavare,S. (eds.) *Progress in Population Genetics and Human Evolution. IMA Vols in Mathematics and Its Applications*, Springer, New York, USA, **87**, pp. 257–270.

Hein,J. *et al*. (2004) *Gene Genealogies, Variation and Evolution: A Primer in Coalescent Theory*. Oxford University Press, USA.

Hoban,S. *et al*. (2012) Computer simulations: tools for population and evolutionary genetics. *Nat. Rev. Genet*., **13**, 110–122.

Hudson,R. (2002) Generating samples under a WrightFisher neutral model of genetic variation. *Bioinformatics*, **18**, 337–338.

Kingman,J. (1982) On the genealogy of large populations. *J. Appl. Probab*., **19**, 27–43.

Parida,L. (2010a) Ancestral recombinations graph: a reconstructability perspective using random-graphs framework. *J. Comput. Biol*., **17**, 1345–1350.

Parida,L. (2010b). Graph model of coalescence with recombinations. In: Heath,L. and Ramakrishnan,N. (eds.) *Problem Solving Handbook in Computational Biology and Bioinformatics*, pp. 85–100.

Parida,L. (2012). Non-redundant representation of ancestral recombinations graphs. In: Anisimova,M. (ed.) *Evolutionary Genomics: Statistical and Computational Methods: Volume 2*.

Parida,L. *et al*. (2011) A minimal descriptor of an ancestral recombinations graph. *BMC Bioinformatics*, **12**, S6

Parida,L. *et al*. (2015) Topological signatures for population admixture. *RECOMB LNBI*, **9029**, 261–275.

Schaffner,S. *et al*. (2005) Calibrating a coalescent simulation of human genome sequence variation. *Gen. Res*., **15**, 1576–1583.

Utro,F. *et al*. (2013) Sum of parts is greater than the whole: inference of common genetic history of populations. *BMC Genomics*, **14**, S10

Varadhan,S.R.S. (2001). *Probability Theory, Volume 7 of Courant Lecture Notes in Mathematics*. New York University, Courant Institute of Mathematical Sciences/American Mathematical Society: New York/ Providence, RI, USA.