VIRUS EVOLUTION

Virus Evolution, 2019, 5(1): vez003

doi: 10.1093/ve/vez003
Resources

SANTA-SIM: simulating viral sequence evolution dynamics under selection and recombination

Abbas Jariani,[1,2] Christopher Warth,[3] Koen Deforche,[4] Pieter Libin,[5,6] Alexei J Drummond,[7,8] Andrew Rambaut,[9,†] Frederick A. Matsen IV,[3] and Kristof Theys[5,*,‡]

[1]Laboratory for Genetics and Genomics, Center of Microbial and Plant Genetics, KU Leuven, 3001 Leuven, Belgium, [2]Laboratory for Systems Biology, VIB, 3001 Leuven, Belgium, [3]Program in Computational Biology, Fred Hutchinson Cancer Research Center, Seattle, WA, USA, [4]Emweb, 3020 Herent, Belgium, [5]Laboratory Clinical and Evolutionary Virology, Rega Institute for Medical Research – KU Leuven, 3000 Leuven, Belgium, [6]Artifical Intelligence Lab, Department of Computer Science, Vrije Universiteit Brussel, 1000 Brussels, Belgium, [7]Centre for Computational Evolution, University of Auckland, Auckland 1010, New Zealand, [8]Department of Biosystems Science and Engineering, Eidgenössische Technische Hochschule Zurich, 4058 Basel, Switzerland and [9]Ashworth Laboratories, Institute of Evolutionary Biology, University of Edinburgh, Edinburgh EH9 3FL, UK

*Corresponding author: E-mail: kristof.theys@kuleuven.be

†https://orcid.org/0000-0003-4337-3707

‡https://orcid.org/0000-0003-0242-7000

Abstract

Simulations are widely used to provide expectations and predictive distributions under known conditions against which to compare empirical data. Such simulations are also invaluable for testing and comparing the behaviour and power of inference methods. We describe SANTA-SIM, a software package to simulate the evolution of a population of gene sequences forwards through time. It models the underlying biological processes as discrete components: replication, recombination, point mutations, insertion–deletions, and selection under various fitness models and population size dynamics. The software is designed to be intuitive to work with for a wide range of users and executable in a cross-platform manner.

Key words: mutation; selection; fitness; simulation; recombination.

1. Introduction

Simulating population dynamics is a popular and effective strategy to model the outcome of molecular genetic processes (e.g. selection and recombination) and to verify evolutionary hypotheses against experimental observations. Simulations of evolutionary histories in population genetics can be categorized either as forward-in-time or backwards-in-time (coalescent) genealogical models. Coalescent models have been historically the leading simulation method and are used for the inference of genetic variation in populations through progressively coalescing lineages according to a stochastic process until only the most recent common ancestor of the sample population is reached (Kingman, 1982; Hudson, 1983). This process is appreciated for its time and memory efficiency as it only considers a sample of observed individuals, irrespective of how large the population is, and is widely used to simulate changes in population size, recombination and sub-populations with migration (Laval and Excoffier, 2004; Mailund *et al.*, 2005). In contrast, forward-in-time evolution simulations are computationally more intensive as the evolutionary history of the entire population is modelled through time. However, these models allow for more complexity and scenarios can include population processes (e.g. natural selection) that are difficult to incorporate in the backwards-time approach (Gillespie, 2001). A large collection of software tools for forward-time genetic data simulation has been developed in the past decades, varying in objectives targeted and compromising between ease of configuration and model complexity. Additionally, resulting from the software design decisions driven by the necessary trade-offs, the tools differ significantly with respect to performance and platform portability. Extensive comparisons of available forward simulators show that a wide but incomplete range of genetic and population processes are considered by these tools (Hoban, Bertorelle, and Gaggiotti, 2012; Yuan *et al.*, 2012; Peng *et al.*, 2013, 2015).

2. Approach

We present the SANTA-SIM software package, which implements an individual-based, discrete-generation, and forwards-time simulator for molecular evolution of genetic data in a finite population. Across the heterogeneous landscape of available simulators, and despite recent advances in simulation models (Zanini and Neher, 2012; Petitjean and Vanet, 2014; Haller and Messer, 2017), SANTA-SIM offers a valuable addition as SANTA-SIM addresses a desired balance between the complexity of the underlying framework and the different evolutionary scenarios that can be modelled. SANTA-SIM is directed towards haploid organisms, and particularly useful to study rapidly evolving pathogens such as RNA viruses that can experience diverse selection pressures and recombination events. SANTA-SIM is distinguished from previous simulators (of virus evolution) by its modularity, flexibility and extensibility of simulation components. Discrete components reflect the different underlying biological processes, which can be configured separately and combined to simulate complex evolutionary processes. Unlike many of previously published tools designed to study population genetics (Balloux, 2001; Peng and Kimmel, 2005; Guillaume and Rougemont, 2006; Hernandez, 2008; Carvajal-Rodriguez, 2008; Ritchie and Bush, 2010), SANTA-SIM is primarily focussed on the dynamics of selection in the face of a changing fitness landscape. For instance, a simulation could have consecutive epochs to model environmental changes. Moreover, rather than only modelling selection by predefined fitness values for different alleles, fitness of each allele could be affected by context-dependent effects such as the population size or the duration that the allele has been present in the population, which can be modelled by various fitness functions included in SANTA-SIM. Samplers can be adapted to extract statistics, sequence alignments, and genealogy trees from the simulation. Furthermore, SANTA-SIM offers a novel experimental framework for simulation of insertion–deletion mutations (indels) and allows changing dynamics of population size.

3. Materials and methods

SANTA-SIM was written in the Java programming language and is available as an open-source project (https://github.com/santa-dev/santa-sim) under the Apache License Version 2.0 (APLv2). A simulation run only requires the pre-built cross-platform executable file (available at the Releases section of the project website) and an XML configuration file (examples are provided at the project website), where the configurable aspects of the simulation process are detailed. The SANTA-SIM framework was developed in a modular manner so that different biological processes are implemented in separate components. This design allows users with basic programming experience to understand how the software functions, and to easily adapt existing or implement new features.

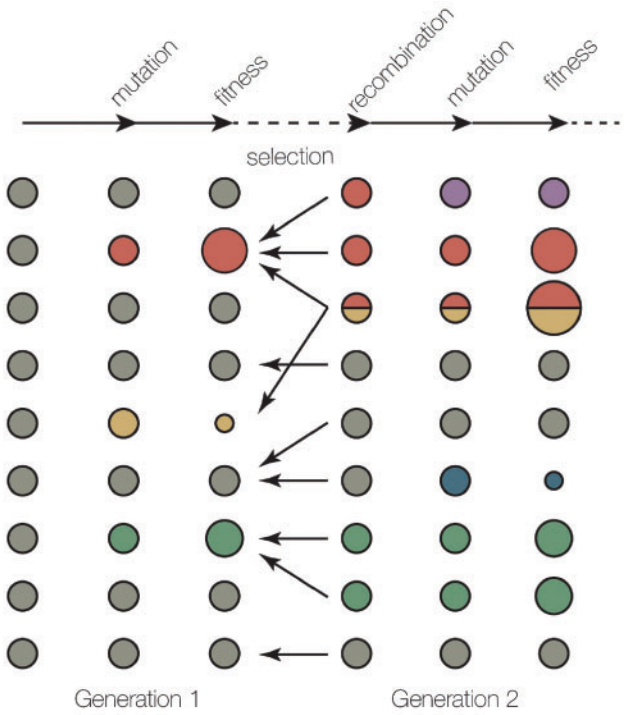

Figure 1. Overview of simulation process in SANTA-SIM. A cycle of two generations in SANTA-SIM simulation, consisting of mutation, recombination, fitness evaluation, and selection. The circles on the left and right, respectively represent the individuals from the first population (parents) and the second generation (progenies). The size of the circles represents the fitness while the colour represents the genotype. Parents with higher fitness are more likely to be selected to generate a progeny, shown by the number of arrows. Each progeny could be generated from one parent (clonal replication) or two parents (recombinant replication).

An overview of the evolution simulation process is demonstrated in Fig. 1 for a cycle of two consecutive generations. After the calculation of fitness for the individuals within a population, the next generation is selected from these parents according to their fitness. Recombination can occur between two parents to generate a progeny with a genome inherited from two different parents. Subsequently, mutations are introduced into the new generation and the simulation will proceed for the following generations.

In order to enable SANTA-SIM to be easily extended, the most fundamental simulation components are presented to the developer as Java interfaces. An overview of the interfaces and their default implementations within SANTA-SIM can be found in Table 1.

Using these model components, the entire simulation can be organized as a sequence of epochs, with each epoch having different selection functions, replication operators, and mutation operators. The transition of the population to a new epoch thus reflects a deterministic environmental change for the

Table 1. An overview of the Java interfaces that correspond to the basic simulation components.

Interface	Component description
FitnessFactor	Enables developers to define a custom fitness function. Default implementations of this interface implement fitness functions that take into account genome sequence and the size of the virus population.
Mutator	Enables developers to control how a molecular sequence is to be modified when a mutational event takes place. SANTA-SIM ships with a default implementation that allows mutations to happen on the nucleotide level.
Replicator	Enables developers to control how a virus can be replicated. Default implementations of this interface implement clonal replication, recombinant replication and recombinant replication with hotspots.
Sampler	Enables developers to implement different ways to sample from the virus population. Default implementations of this interface implement an alignment sampler, an allele frequency sampler, a genome description sampler a general statistics sampler and a tree sampler.
Selector	Enables developers to implement different ways to apply evolutionary selection on individuals. SANTA-SIM ships with a binary search selector and a roulette wheel selector for constant population size. For simulating the dynamics of population size under logistic growth model, a selector is implemented that takes into account the growth rate (r), carrying population (K), and population size (P) to determine the expected number of progeny for each individual, where population dynamics are determined by $r^*P^*(1-P/K)$, also known as the Verhulst model.
PopulationGrowth	PopulationGrowth Facilitates further development of the software to specify the growth processes custom-tailored for specific cases which are not predicted in the selector interface implementations.

population. The software can also be configured to report insights into the operation of the simulation. These outputs can either be at the population level by describing allele frequencies through time, population fitness, diversity, and divergence or at the individual level by providing nucleotide or amino acid sequence alignments at chosen loci and phylogenies.

3.1 Population, individuals, genomes, and features

The population in SANTA-SIM consists of individual organisms, each of which contains a single genome. The genome is a linear sequence of nucleotides organized into *features*. Features reflect the genome organization into genes and open reading frames. Each feature may be composed of one or more fragments of the genome, read in either a forward or reverse direction, and may overlap with each other. Different modes and degrees of selection on either nucleotides or amino acid sites can be specified on each feature.

3.2 Evolutionary process

The evolutionary process in SANTA-SIM assumes discrete generations, and each generation consists of a concatenation of discrete components. The population is subjected to mutations, recombination, and various models for fitness assignment based on the genotype of the genome features. The size of the population can change depending on the overall fitness of the population. The interaction between these components supports the configuration of complex evolutionary scenarios.

The simulation begins with an initial population of individuals which is seeded from a single sequence or a pool of different sequences. At each generation, evolution is simulated in four sequential steps of replication (with optionally recombination), mutation, fitness, and selection.

3.2.1 Replication

Together with the mutation component, below, the replication component is analogous to the actions of a polymerase complex and produces the genetic material for a new individual from one or more parents. The simplest replicator is clonal and the descendant inherits the genome of exactly one parent. We also provide a recombinant replicator that models a 'template-switching' polymerase. For this replicator, two probabilities are defined: a probability that a recombinant is used instead of the clonal replicator, and a probability of the polymerase switching between the parents' templates as replication proceeds along the genome.

3.2.2 Mutation

Mutation is implemented as an independent process after replication. The user specifies a per-site and per-generation probability of mutation and the mutator component then applies mutations to the genome accordingly. For efficiency, the default mutator draws the number of mutations from a Poisson distribution with an expectation given by the number of nucleotides and the mutation rate. These are then distributed uniformly across the sites. A bias towards transition-type mutations can be specified to reflect the action of specific polymerases.

In addition to substitution mutations, SANTA-SIM also offers an indel mutation model that can be useful to more closely mimic the evolutionary behaviour of retroviruses like HIV-1. The user can specify a per-generation probability of indel, together with an independent distribution of indel length. Frameshifting a genome is assumed to be nearly always fatal so only whole-codon indels are permitted. Indels not only change the content and length of a simulated genome, but they can also affect regions subject to fitness constraints. Depending on the position relative to the boundaries of fitness-constrained regions, indels can either shrink or lengthen a feature and therefore have an impact in fitness scores. Changes to fitness-constrained regions will affect subsequent generations of the lineage, but will not affect sibling lineages. No fitness value is directly assigned to an indel at this moment, given the complexity of inferring genotype–phenotype relationships in this context, but the experimental implementation of the current indel model provides a framework for ongoing research. Details on the indel mutation model and illustrations of the implications for fitness calculations can be found at the project webpage.

3.2.3 Fitness calculation

The fitness of each genome is calculated using one or more fitness functions (Table 2). By shuffling selection coefficients among states over time, non-stationary random positive selection can be implemented. Distinct fitness functions can be

Table 2. Overview of the fitness functions that can be implemented in the simulations by their respective specification in the XML file.

Fitness function	Description
Neutral	Does not lead to a selective constraint.
Empirical	Manual specification of fitness values for each state (for instance amino-acid) at one or more positions.
Purifying	Assigning negative selection coefficients based on observed frequencies in an alignment, or using bio-chemical properties compared with the most frequent state in an alignment.
Age dependent	Records the age of alleles, and assigns higher penalties to older alleles.
Frequency dependent	Assigns higher penalties to more frequent alleles.
Exposure dependent	Penalizes the exposure of an allele, by integrating frequency over time.
Population size dependent	Enables to define fitness as a function of total population size.

defined for the nucleotide sequence and the corresponding amino acid translation. Furthermore, different regions of the genome can be assigned different fitness functions (e.g. most sites under purifying selection but with a few sites under diversifying selection). Such fitness functions are introduced in the XML configuration file.

3.2.4 *Selection*

The next generation of individuals then selects their parents from the previous generation where each parent is selected with replacement with a probability proportional to its genome's fitness. The number of parents that are selected for each new individual depends on the mode of replication, which is described next.

3.3 Sampling sequences, phylogenies, and statistics

Given the large scale of many typical simulations with biologically relevant parameters, we have made every effort to use memory efficiently. For example, genome sequences are stored in a central 'gene-pool' so that only unique genomes are stored with the individuals having only an index for the genome they currently carry. Individuals that replicate without any mutations thus inherit this index. This also makes calculations of the population genetic diversity more efficient. In addition, where applicable, fitness may be computed incrementally from the parent's fitness in the previous generations and incidental mutations. We also implemented an optional framework where genomes are stored as differences from a central 'master' sequence. This master sequence can be recalculated occasionally to release memory.

At predefined time intervals or specific times during the simulation, SANTA-SIM can report statistics about the current population, including average fitness, genetic diversity and number of unique genomes. A random sample of individuals of a specified size can also be generated and the genomic sequences recorded as a nucleotide or amino acid alignment (FASTA or NEXUS format) for use in other software applications.

Finally, it is possible for SANTA-SIM to keep track of the genealogy of the entire population and then provide the tree of the individuals sampled. A variety of events can be recorded, such as the prevalence of different states in selected sites, or shuffling events in the purifying fitness function, to be able to investigate the effect of these events on the population.

4. Results

4.1 Simulation examples

We describe three example runs that demonstrate some of the functionalities of SANTA-SIM and the fitness functions that can be applied. A first example simulates the consecutive selection of a set of deleterious mutations that each gives a fitness advantage in a new environment. A second example demonstrates how the initial frequency of a beneficial mutation during a selective bottleneck impacts diversity and phylogeny, which is related to the discussion of soft and hard selective sweeps (Hermisson and Penningsa, 2005). A third example shows the interplay between the fitness of a mutation and its frequency, as in the case of a host–pathogen interaction. The configuration files and additional details of these simulations are available as Supplementary Material.

4.1.1 *Directional selection driven by changing selective advantage of mutations*

In the face of environmental changes, certain mutations which were previously neutral or deleterious can confer a selective advantage in the new environment. Such a scenario occurs when a viral population, for example, HIV-1, is subjected to suppressive action of antiviral treatment and demonstrates adaptive evolution. We can simulate this example using two epochs, with no selective pressure present in the first epoch. A population of 10,000 sequences was created by evolving an initial sequence (609 nucleotides in length) under mutation and moderate level of purifying selection for a duration of 3,000 generations. A set of four mutations across this sequence were associated with a selective disadvantage, making their fixation in the population unlikely. The second epoch started after generation 3,000, with a strong beneficial impact of these four mutations in the new environment, and their subsequent selection. Figure 2 illustrates the dynamics of population diversity and the frequency of the beneficial alleles while Fig. 3 shows the phylogeny reconstructed from sequences sampled through the course of the simulation. Genetic diversity was calculated as the mean pairwise distance, based on percent identity, within a random sample of 1,000 individuals from the population.

4.1.2 *Fraction of viruses with beneficial alleles affects the trajectory of a selective sweep*

Directional selection, as shown in the previous section, results in a decrease of genetic variation in the viral population, but the extent of reduction depends on the initial frequency of the beneficial mutation and the strength of selection (Smith and Haigh, 1974). When a strongly beneficial but rare mutation increases in frequency, the genetic background of this adaptive mutation

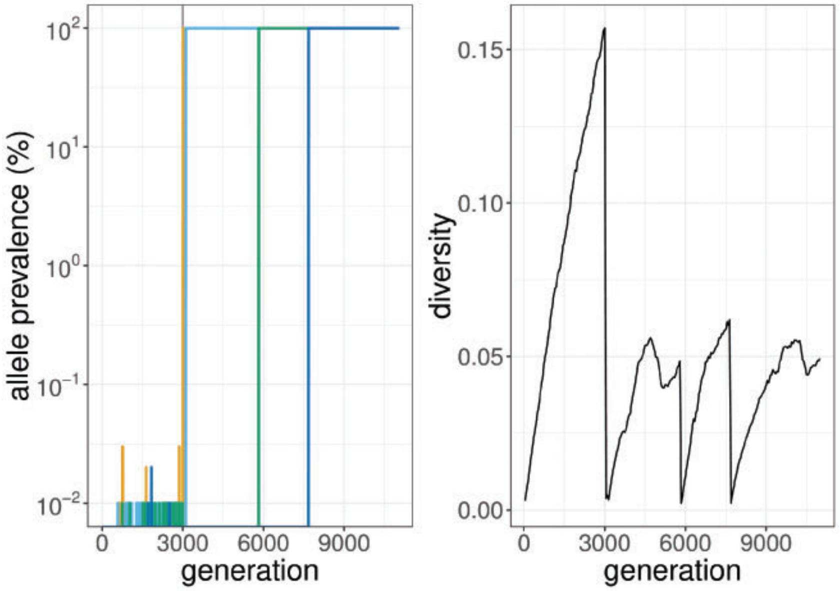

Figure 2. The simulation has two phases. In the first 3,000 generations, the only selective force is purifying selection. After this initial phase (vertical grey line), four particular mutations become beneficial: 50 T (yellow), 100 K (light blue), 150 A (green), and 200 G (dark blue). Mutation 100 K has been present in the initial population at low prevalence (%). Prevalence on the y-axis is shown as log10 transformed. Diversity drops through each wave of selective sweep where a beneficial mutation appears and takes over. The simulation starts from a population with only one sequence at the first generation. Diversity was defined as the mean pairwise identity percentage between all sequences. For a given nucleotide position between two sequences, two non-identical bases will result in a score of one for that position while identical bases give a zero score. The distance of the two sequences was calculated as the mean of such identity scores across all nucleotide positions. In this simulation, the alleles reaching fixation have primarily appeared de novo or selected from standing variation (100 K), as no recombination events were simulated here.

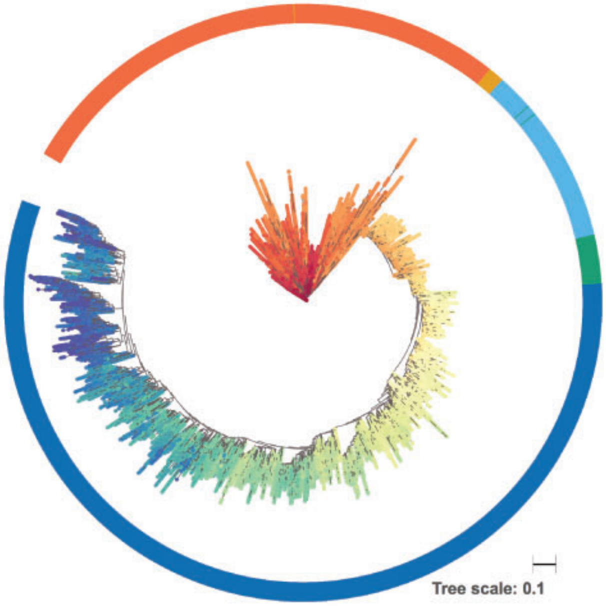

Figure 3. Phylogenetic tree from the sampled sequences through multiple waves of selective sweep. From the simulation run of 10,000 generations, a sample of 500 sequences was collected at every 100th generation, and a tree was made with FastTree using default parameters (Price, Dehal, and Arkin, 2009). The tree is coloured by increasing generations (from red to blue) and the outer band denotes the consecutive selection of beneficial mutations (see Fig. 2 for mutation colours, the red section denotes absence of a mutation).

will dominate the population and have a strong impact on diversity, known as a hard selective sweep. In contrast, rapid adaptation to a novel selection force by a mutation either highly prevalent or newly arising simultaneously in different

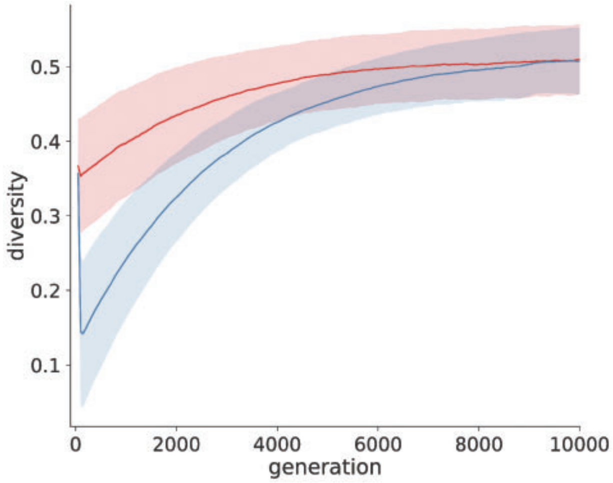

Figure 4. Diversity trajectory through selective sweeps for different initial frequencies of the beneficial alleles. Two modes of selective sweep are simulated for a population of size 10,000: In one case a small fraction of the initial population carries the beneficial allele (2 in 10,000; blue line), whereas in the other case a higher fraction carries this allele (55 in 10,000; red line). The diversity is calculated as the mean pairwise distance between 1,000 sampled sequences from the population, similar to the previous simulation. The simulation is repeated 1,000 times. The mean and standard deviation of these replicates are shown as the thick line and shaded area, respectively.

individuals will lead to a less drastic reduction in genetic variation of the population, known as a soft sweep (Wilson, Pennings, and Petrov, 2017). This example investigates how a starting mutation frequency affects a selective sweep. Two independent simulations were carried out for a starting population of size 10,000 individuals, but where one population had 55 individuals with the beneficial mutation (here amino acid lysine

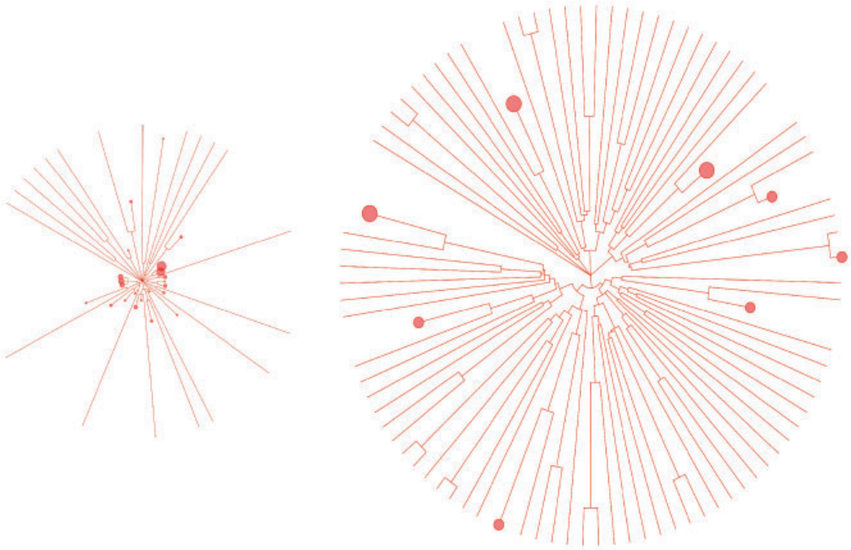

Figure 5. Phylogenetic trees after fixation of the beneficial allele grouped for two levels of initial prevalence of allele. A population with two levels of initial prevalence of a beneficial allele is subjected to selective sweeps. Phylogenetic unrooted trees are sampled (100 sequences), using tree sampling capability of SANTA-SIM, after the fixation of the beneficial allele. The tree on the left corresponds to the case where the starting population had a lower number of individuals with the beneficial allele (2 in 10,000), while the tree on the right corresponds to the case where the starting population had a higher number of individuals with the beneficial allele (55 in 10,000). Clades were collapsed (red dots) when average branch length distance to the taxa were below an illustrative threshold.

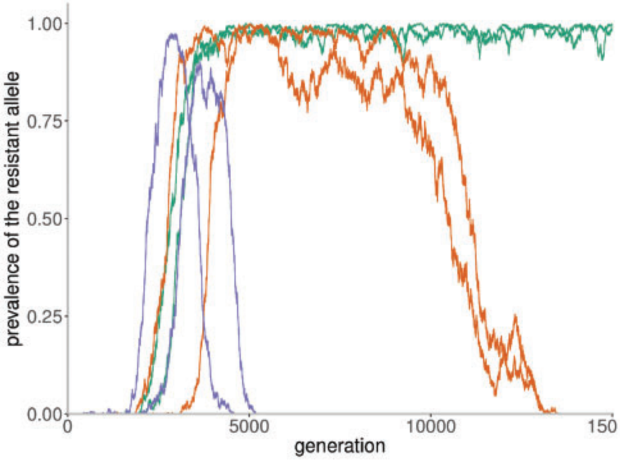

Figure 6. Simulation of selection dynamics in host pathogen co-evolution. Simulation of the interplay between the appearance of an escape mutation in a pathogen and host adaptation to the resistance. The selection coefficient of the beneficial mutation was set to 0.05. The exposure dependent fitness function was used to simulate the gradual decrease in fitness of the beneficial resistance allele in a pathogen as the hosts immune system adapts. Three parameters for exposure penalty were used. The penalty parameter for the green curves is set to 10^{-7}, for the orange curves it is 10^{-6} and for the purple curves it is 10^{-5}. There are two replicates shown for each simulation.

at position 100) before the onset of the selective sweep, while the other population only had two individuals with the beneficial mutation.

In order to carry out this simulation, initially a seed heterogeneous population was created by evolving a starting population with only one genome sequence (609 nucleotides in length) for 50,000 generation in presence of moderate purifying selection. Afterwards, the target mutation position of this seed sequence was edited so that in one case there are only two individuals with the beneficial allele, while in another case there are fifty-five individuals with the beneficial allele. These seed populations were subjected to the second phase of simulation were selective sweeps occurred, giving a selective advantage to the mentioned allele.

Figure 4 illustrates that the magnitude of diversity reduction upon a sweep is dependent on the fraction of the beneficial mutation in the initial population. The scenario of only two individuals with the beneficial allele being present and dominating the population immediately upon the sweep, sequence diversity drops significantly since the growing population originated from the two individuals with the increased fitness advantage. However, when fifty-five individuals carry the beneficial allele, a smaller reduction in sequence diversity is observed due to the more heterogeneous origin of the growing population. Diversity remains at a quasi-constant level after the sweep since the span of the number of generations here is rather short.

In Figure 5, phylogenetic trees were constructed after fixation of the beneficial allele in the population and demonstrate distinguishably different branching patterns at the root of the trees for the two modes of the selective sweep and resulting different levels of diversity, which resulted in the pattern in Fig. 4. A limited number of branching patterns is associated with reduced genetic diversity, resulting in shorter branches within the tree and subsequent increased collapsing of clades, compared with a more diverse virus population, when using a fixed distance threshold for collapsing.

4.1.3 Simulation of dynamics of host–pathogen co-evolution: a complex selection scenario

Pathogen escape from the host immune response can be transient when adaptation of the host immune system to the pathogen variant occurs. In this example, we simulated a scenario where the pathogen has the capability to develop escape from the host's immune system and increase its fitness by acquiring a particular beneficial mutation. At the same time, the host gradually develops more protection against the alleles of the pathogen exposed to the host.

SANTA-SIM can model this kind of adaptive dynamics using an 'exposure dependent fitness function' which assigns fitness

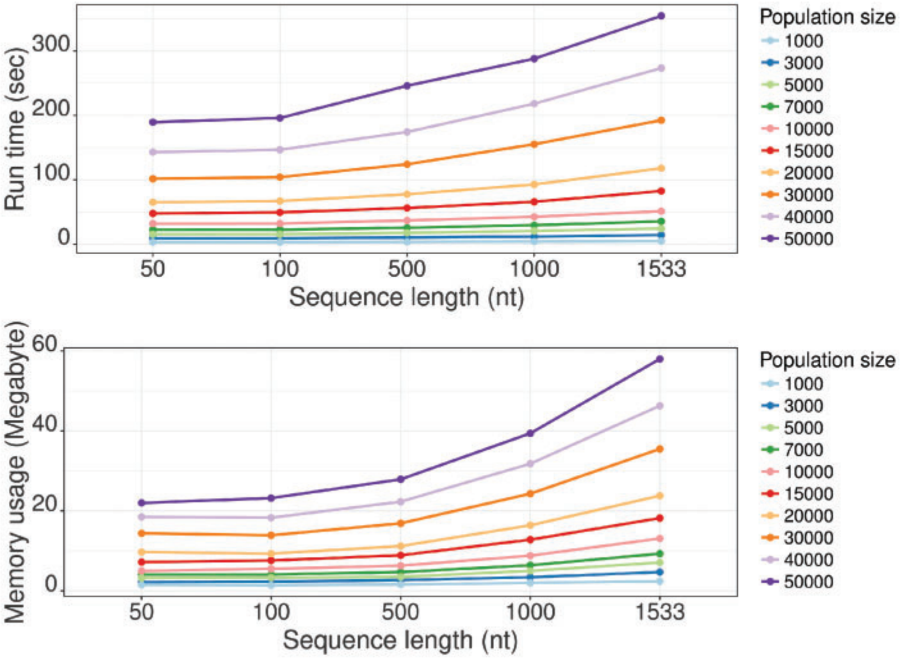

Figure 7. Performance benchmarking. Memory and run time of simulations with purifying selection for 10,000 generations with different population sizes and genome lengths.

values to pathogen variants based on the exposure of their allele in the population since it last appeared. Such fitness is assigned as e^{-E*p} where p is the penalty parameter and E is the integrated prevalence of the allele over time since its last appearance. Thus the variants are penalized when they have been present for a longer period at a higher prevalence in the population. The severity of how much the exposed alleles are punished also depends on a penalty parameter. Overall, the fitness of each individual is determined by two factors: presence of the beneficial mutation and the exposure penalty which models adaptive evolution of host's immune system.

Figure 6 demonstrates the effect of the exposure penalty parameter. A low value (shown in green) is similar to the absence of exposure-dependent fitness function and the beneficial mutation reaches and remains a high prevalence. The minor fluctuation in prevalence results from the marginal increase relative to other alleles with selection coefficient of 0.05. For higher values of the penalty parameter for allele exposure, it is observed that mutations that increase in prevalence are penalized for their exposure and affecting the duration of high prevalence. Simulation for each parameter is shown for duplicates.

4.2 Memory and runtime profiling

Multiple simulation configurations were defined with a range of population sizes and genome lengths. A memory footprint and elapsed wall-clock time were measured for each simulation as shown in Fig. 7. A computer with 3.4 GHz Intel Core i7 CPU and 32 GB of RAM was used to run the simulations (Ubuntu 16.04, Java openjdk version 1.8). All simulations are configured to run for 10,000 generations under purifying selection and without sampling the simulated sequences. More information on this analysis is available at the project website. We also compared the performance of SANTA-SIM against three established simulators using a common evolutionary scenario: simuPOP (Peng and Kimmel, 2005), SFS CODE (Hernandez, 2008), and VIRAPOPS (Petitjean and Vanet, 2014). These benchmarking results, together with the experiment files, are provided as Supplementary Material and demonstrate that SANTA-SIM is competitive in terms of memory and runtime statistics.

5. Discussion

SANTA-SIM is a forward-time discrete-generation gene sequence simulator, designed to scale to large population sizes and micro-organism genome lengths while implementing complex selection and recombination scenarios. SANTA-SIM is open-source software and written in an extremely modular fashion to facilitate a wide range of additional components being implemented to accommodate varying simulation environments and organisms.

Acknowledgements

The authors would like to thank Beth Shapiro for testing the software. They also thank the anonymous reviewers for their constructive suggestions that greatly improved the quality of this article.

Funding

C.W. and F.M. were partially supported by the Bill and Melinda Gates Foundation (Award Number OPP1110049). F.M. was supported in part by National Institutes of Health R01-GM113246 a Faculty Scholar grant from the Howard Hughes Medical Institute and the Simons Foundation. P.L. was supported by a Fonds Wetenschappelijk Onderzoek PhD grant. K.T. was supported by a Fonds Wetenschappelijk Onderzoek postdoctoral grant. A.R. acknowledges the support of the Wellcome Trust through project 206298/Z/17/Z, European Union Seventh Framework Programme under Grant Agreement no. 278433-PREDEMICS and no. 725422-

RESERVOIRDOCS and the Bill and Melinda Gates Foundation (grant number OPP1175094).

Data availability

SANTA-SIM is freely available at https://github.com/santa-dev/santa-sim as an open source software project.

Supplementary data

Supplementary data are available at *Virus Evolution* online.

Conflict of interest: None declared.

References

Balloux, F. (2001) 'EASYPOP (version 1.7): A Computer Program for Population Genetics Simulations', *The Journal of Heredity*, 92: 301–2.

Carvajal-Rodriguez, A. (2008) 'GENOMEPOP: A Program to Simulate Genomes in Populations', *BMC Bioinformatics*, 9: 223.

Gillespie, J. H. (2001) 'Is the Population Size of a Species Relevant to Its Evolution?', *Evolution; International Journal of Organic Evolution*, 55: 2161–9.

Guillaume, F., and Rougemont, J. (2006) 'Nemo: An Evolutionary and Population Genetics Programming Framework', *Bioinformatics*, 22: 2556–7.

Haller, B. C., and Messer, P. W. (2017) 'SLiM 2: Flexible, Interactive Forward Genetic Simulations', *Molecular Biology and Evolution*, 34: 230–40.

Hermisson, J., and Penningsa, P. (2005) 'Soft Sweeps: Molecular Population Genetics of Adaptation from Standing Genetic Variations', *Genetics*, 169: 2335–52.

Hernandez, R. D. (2008) 'A Flexible Forward Simulator for Populations Subject to Selection and Demography', *Bioinformatics*, 24: 2786–7.

Hoban, S., Bertorelle, G., and Gaggiotti, O. E. (2012) 'Computer Simulations: Tools for Population and Evolutionary Genetics', *Nature Reviews. Genetics*, 13: 110–22.

Hudson, R. R. (1983) 'Properties of a Neutral Allele Model with Intragenic Recombination', *Theoretical Population Biology*, 23: 183–201.

Kingman, J. F. C. (1982) 'The Coalescent', *Stochastic Processes and Their Applications*, 13: 235–48.

Laval, G., and Excoffier, L. (2004) 'SIMCOAL 2.0: A Program to Simulate Genomic Diversity over Large Recombining Regions in a Subdivided Population with a Complex History', *Bioinformatics*, 20: 2485–7.

Mailund, T. et al. (2005) 'CoaSim: A Flexible Environment for Simulating Genetic Data under Coalescent Models', *BMC Bioinformatics*, 6: 252.

Peng, B., and Kimmel, M. (2005) 'simuPOP: A Forward-time Population Genetics Simulation Environment', *Bioinformatics*, 21: 3686–7.

—— et al. (2013) 'Genetic Simulation Resources: A Website for the Registration and Discovery of Genetic Data Simulators', *Bioinformatics*, 29: 1101–2.

—— et al. (2015) 'Genetic Data Simulators and Their Applications: An Overview', *Genetic Epidemiology*, 39: 2–10.

Petitjean, M., and Vanet, A. (2014) 'VIRAPOPS: A Forward Simulator Dedicated to Rapidly Evolved Viral Populations', *Bioinformatics*, 30: 578–80.

Price, M. N., Dehal, P. S., and Arkin, A. P. (2009) 'FastTree: Computing Large Minimum Evolution Trees with Profiles instead of a Distance Matrix', *Molecular Biology and Evolution*, 26: 1641–50.

Ritchie, M., and Bush, W. (2010) 'Genome Simulation Approaches for Synthesizing in Silico Datasets for Human Genomics', *Advances in Genetics*, 72: 1–24.

Smith, J. M., and Haigh, J. (1974) 'The Hitch-hiking Effect of a Favourable Gene', *Genetical Research*, 23: 23–35.

Wilson, B. A., Pennings, P. S., and Petrov, D. A. (2017) 'Soft Selective Sweeps in Evolutionary Rescue', *Genetics*, 205: 1573–86.

Yuan, X. et al. (2012) 'An Overview of Population Genetic Data Simulation', *Journal of Computational Biology: A Journal of Computational Molecular Cell Biology*, 19: 42–54.

Zanini, F., and Neher, R. A. (2012) 'FFPopSim: An Efficient Forward Simulation Package for the Evolution of Large Populations', *Bioinformatics*, 28: 3332–3.