Bioinformatics, 33(14), 2017, 2221–2223
doi: 10.1093/bioinformatics/btx146
Advance Access Publication Date: 16 March 2017
Applications Note

Genetics and population analysis

SIApopr: a computational method to simulate evolutionary branching trees for analysis of tumor clonal evolution

Thomas O. McDonald and Franziska Michor*

Department of Biostatistics and Computational Biology, Center for Cancer Evolution, Dana-Farber Cancer Institute, and Department of Biostatistics, Harvard School of Public Health, Boston, MA 02115, USA

*To whom correspondence should be addressed.

Associate Editor: Oliver Stegle

Received on October 11, 2016; revised on February 17, 2017; editorial decision on March 11, 2017; accepted on March 15, 2017

Abstract

Summary: SIApopr (Simulating Infinite-Allele populations) is an R package to simulate time-homogeneous and inhomogeneous stochastic branching processes under a very flexible set of assumptions using the speed of C++. The software simulates clonal evolution with the emergence of driver and passenger mutations under the infinite-allele assumption. The software is an application of the Gillespie Stochastic Simulation Algorithm expanded to a large number of cell types and scenarios, with the intention of allowing users to easily modify existing models or create their own.
Availability and Implementation: SIApopr is available as an R library on Github (https://github.com/olliemcdonald/siapopr).
Supplementary information: Supplementary data are available at *Bioinformatics* online.
Contact: michor@jimmy.harvard.edu

1 Introduction

Branching processes are a class of stochastic processes used to model the growth and composition of reproducing populations (Haccou *et al.*, 2005). These processes are convenient for modelling clonal evolution of cancer cells, where new (epi)genetic alterations may appear and give rise to subclones with different fitness than their parents (Nowell, 1976), leading to a potentially large number of clones in an exponentially growing population. Simulation of these complex processes is usually done with the Gillespie Stochastic Simulation Algorithm (SSA) (Gillespie, 1977), which is typically used to model chemical reaction systems, but is also applicable to birth–death processes where individuals have exponentially distributed lifetimes before undergoing an event such as birth or death.

Branching process models can be used to count cell population sizes and determine proportions of alleles for multiple subpopulations. These models enjoy a rich history in cancer research (Haccou *et al.*, 2005; Kimmel and Axelrod, 2002). However, branching processes quickly become intractable when trying to estimate parameters with likelihood-based methods even in simple multi-type models. Furthermore, researchers often face a lack of data that is of sufficiently high resolution for model predictions and parameter estimation. Some data types, such as data from barcoding experiments, provide a large number of observations representing a nearly complete distribution of cell types at a single time point (Bhang *et al.*, 2015). Simulation then becomes a useful tool in these scenarios, where multiple instantiations of the underlying process attempt to reconstruct the distribution for a given set of parameters. Branching process simulations are used in these studies to identify parameters and models that can likely explain the underlying biological mechanism; they also can be used to study alternative scenarios, where a set of parameters or models is incapable of explaining biological phenomena (Gao *et al.*, 2016). Despite these previous studies, a flexible and generalizable software package for branching process simulations remains to be presented.

SIApop adapts the SSA to simulate an infinite-allele branching process where mutant cells are of a unique type each and have random variables representing their birth and death rates. Our software extends the processes formally described in Foo *et al.* (2015) and Gao *et al.* (2016) to a large number of flexible scenarios. The base process initiates with an individual ancestor with a given birth rate,

b_0, and death rate, d_0. Upon a reproduction event, one daughter cell may harbor a new mutation, giving rise to a new cell type, with probability u_0. If an individual gives birth to a new mutant, then the latter has birth rate $b_1 = b_0 + s_1$, where s_1 is chosen from a probability distribution $F(s)$. Our software includes the ability for cells to have time-dependent rates, $b_i(t)$ and $d_i(t)$, and mutation probabilities differing from those of parent cells.

This class of branching processes is useful for modeling the effects of driver mutations while still accounting for genetic drift from passenger mutations since the fitness of new mutants is chosen from a distribution, removing the need for distinguishing between mutation types in the formulation of the branching process. The implementation of a fitness landscape also allows different mutations to have different fitness effects instead of requiring the same change with each new mutation (Foo *et al.*, 2015).

Other forward-time population simulation software exists, including OncoSimulR (Diaz-Uriarte, 2017), simuPOP (Peng and Kimmel, 2005) and fwdpp (Thornton, 2014), but Simulating Infinite-Allele populations (SIApopr) is unique in that is designed to generate branching processes with an infinite number of driver mutations. The most similar software, OncoSimulR, has a limit to the number of driver mutations. We also include the ability to customize the ancestor file so that a single simulation can achieve what would normally require multiple simulations and merging of the data thereafter.

2 Description

SIApopr uses the direct SSA to advance the simulation by first determining the time until the next event (birth, death, mutation etc.) and then choosing the clone that undergoes the event and determining which type of event takes place. When a birth takes place, the new individual can be a mutant with probability u_i where i is the new type of individual. Under this condition, a new clone is formed that is of a unique type not yet present in the population, and the parameters for this clone are randomly chosen from their respective distributions. The code is written in a modular manner to simplify customizing the process and distributions, so the user is able to easily change the underlying distributions in the code and recompile to achieve different results.

SIApopr simulates birth-death-mutation processes in C++ with flexibility in model selection, including multiple (even infinite) types of cells. The program is capable of generating a large number of individuals across multiple clones that can appear due to the given mutation probability or can be predefined as ancestors. Throughout the process, the number of cells within each type is stored and the final count along with information about each type is output at the end of the simulation. Clones are treated as nodes in a double linked list along with a link to the parent clone. The birth, death, mutation rate, parent node, and time of appearance are stored. Each clone is given a labelled value equal to the order of its appearance in the process, and the identity of the clone contains the labels of all ancestors of the clone. This approach allows the user to trace the lineage of any clone in the process. Ancestors can be input into the model with their own labels as well.

As a particular example, we allow the new types to have birth and death rates drawn from a double exponential distribution as in (Foo *et al.*, 2015) with a point mass at 0 with mass equal to the probability that the mutation is a passenger and has no effect on fitness. The mutation probability of a daughter can also be determined from a random distribution, $u_1 = u_0 + v_1$ where $v_1 \sim \text{Beta}(\alpha, \beta)$ with a maximum value of 1. A modification of the code to include any other distribution is also possible for users.

The model inputs determine the type of process run, with the most general process allowing for mutations with random fitness contributions. Setting the mutation probability to zero yields a branching process where no new clones appear, so the branching process is single type or has a type space based on the ancestor list provided as a separate file. Allowing for a mutation probability, but setting the passenger probability to 1 or removing information about the fitness distribution, simulates the infinite-allele branching process from Pakes (1989).

2.1 Ancestor information and input

A separate input file may contain information about ancestors, allowing the user to begin a process with as many ancestors as desired, or restart a process that previously ended. The ancestor information is a tab-delimited file where each row describes a clone and must contain at least the number of ancestor cells associated with a clone, but may also contain the birth rate, death rate, mutation probability and clone identifier. The output of a process can serve as the ancestor input, so the time of appearance and information about a clone's descendants may also be included but are only for record-keeping. Given a list of cell counts without any other information, the simulator uses values coming from the input file that contains global parameters. The input file allows the user to designate parameters and options for simulating and outputting the distribution. The various parameters are described in the Supplementary Material. Default values are given so that the process may be run without any input and a single-type birth–death process will be simulated.

2.2 Time-dependent processes

A separate program is included that extends the birth–death-mutation process simulation to time-inhomogeneous processes where the birth and death rates vary as a function of time. We implement the SSA using adaptive thinning to choose the next time of an event, and the state is updated at each time step by integrating over the propensity function (Lewis and Shedler, 1979). Adaptive thinning requires simulating a random variable from an exponential distribution with a constant propensity function that bounds the true propensity function; thus we find the maximum propensity of each clone and use the sum of these values over all individuals to determine our constant exponential distribution parameter. The propensity function is a function describing the probability that a birth or death event occurs within a time interval.

2.3 Sampling

Motivated by single-cell sequencing studies, an option to sample from the final population is included where the user specifies the sample size, both in terms of the number of patients and the number of single cells per patient (Gao *et al.*, 2016; Wang *et al.*, 2014). The output is a list of identifiers for the cells sampled, which represent the mutations present in each cell. Sampling proceeds without replacement within each sample. Selecting this option allows the user to observe differences in heterogeneity and phylogenies between the simulated and observed populations. Single cell sequencing is currently limited to a small number of cells, and therefore the cells sampled would most likely be from dominant clones. Sampling can give a glimpse of the expected evolutionary structure of these dominant clones, but rarely includes subclones with smaller cell counts. We also include functions in R to sample from the entire cell population and provide allele frequencies, representing results from bulk sequencing studies.

3 Conclusion

SIApopr is designed to simulate birth–death-mutation processes under the infinite-allele assumption and tracks all clones in the process. Estimation methods for multi-type branching processes quickly become intractable as the model complexity increases. Simulation methods such as SIApopr provide a method for comparing evolutionary models, and can be performed quickly on a cluster. A basic branching process run with a mutation rate of 0 and no death up to 1 000 000 cells takes an average of 257 milliseconds on a personal 2.7 GHz Intel Core i5 CPU with 8 GB RAM running RStudio in OS X. Including a mutation probability of 0.01 results in an average runtime of 995 ms. Running with a mutation rate of 1 up to 10 000 cells takes an average of 284 ms on the same computer.

Acknowledgements

We would like to thank Giovanni Parmigiani and the Michor lab for insightful discussions and testing the code.

Funding

This work has been supported by the Dana-Farber Cancer Institute Physical Science-Oncology Center; grant number NIH [U54CA193461].

Conflict of Interest: none declared.

References

Bhang,H.C. *et al*. (2015) Studying clonal dynamics in response to cancer therapy using high-complexity barcoding. *Nat. Med*., **21**, 440–448.

Diaz-Uriarte,R. (2017) OncoSimulR: genetic simulation with arbitrary epistasis and mutator genes in asexual populations. *Bioinformatics*, **33**, 1898–1899.

Foo,J. *et al*. (2015) An evolutionary approach for identifying driver mutations in colorectal cancer. *PLoS Comput. Biol*., **11**.

Gao,R. *et al*. (2016) Punctuated copy number evolution and clonal stasis in triple-negative breast cancer. *Nat. Genet*., **48**, 1119–1130.

Gillespie,D.T. (1977) Exact stochastic simulation of coupled chemical reactions. *J. Phys. Chem. US*, **81**, 2340–2361.

Haccou,P. *et al*. (2005) *Branching Processes: variation, Growth, and Extinction of Populations. No. 5*. Cambridge university press, Cambridge.

Kimmel,M. and Axelrod,D. (2002) Branching Processes in Biology. *Interdisciplinary Applied Mathematics*. Vol. 19. Springer, New York.

Lewis,P.A. and Shedler,G.S. (1979) Simulation of nonhomogeneous Poisson processes by thinning. *Nav. Res. Logist. Q*, **26**, 403–413.

Nowell,P.C. (1976) The clonal evolution of tumor cell populations. *Science*, **194**, 23–28.

Pakes,A.G. (1989) An infinite alleles version of the Markov branching process. *J Aust Math Soc A*, **46**, 146–169.

Peng,B., and Kimmel,M. (2005) simuPOPL a forward-time population genetics simulation environment. *Bioinformatics*, **21**, 3686–3687.

Thornton,K.R. (2014) A C++ template library for efficient forward-time population genetic simulation of large populations. *Genetics*, **198**, 157–166.

Wang,Y. *et al*. (2014) Clonal evolution in breast cancer revealed by single nucleus genome sequencing. *Nature*, **512**, 155–160.