BIOINFORMATICS APPLICATIONS NOTE Vol. 27 no. 9 2011, pages 1332–1334
doi:10.1093/bioinformatics/btr124

Genetics and population analysis Advance Access publication March 12, 2011

fastsimcoal: a continuous-time coalescent simulator of genomic diversity under arbitrarily complex evolutionary scenarios

Laurent Excoffier[1,2,*] and Matthieu Foll[1,2]

[1]Institute of Ecology and Evolution, University of Berne, 3012 Berne and [2]Swiss Institute of Bioinformatics, 1015 Lausanne, Switzerland

Associate Editor: Jeffrey Barrett

ABSTRACT

Motivation: Genetic studies focus on increasingly larger genomic regions of both extant and ancient DNA, and there is a need for simulation software to match these technological advances. We present here a new coalescent-based simulation program fastsimcoal, which is able to quickly simulate a variety of genetic markers scattered over very long genomic regions with arbitrary recombination patterns under complex evolutionary scenarios.

Availability and Implementation: fastsimcoal is a C++ program compiled for Windows, MacOsX and Linux platforms. It is freely available at cmpg.unibe.ch/software/fastsimcoal/, together with its detailed user manual and example input files.

Contact: laurent.excoffier@iee.unibe.ch

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

Received on December 28, 2010; revised on February 23, 2011; accepted on March, 2011

1 INTRODUCTION

Coalescent theory has revolutionized the field of population genetics by providing a very powerful and intuitive framework for the study of molecular diversity within and between species (Hudson, 1990). It has also provided very efficient ways to simulate diversity under complex evolutionary scenarios (Wakeley, 2009).

A variety of coalescent-based simulation programs have been developed in the past few years (e.g. Chen *et al*., 2009; Ewing and Hermisson, 2010; Hudson, 2002; Laval and Excoffier, 2004), allowing one to generate genetic data much more quickly than with classical forward approaches [see e.g. Carvajal-Rodriguez (2008) for a review of recent coalescent and forward genetic simulators]. However, many standard simulation programs now have problems in generating polymorphisms over long genomic regions, like those now produced by new generation sequencing technologies (e.g. Durbin *et al*., 2010).

In this article, we present fastsimcoal, a completely rewritten continuous-time implementation of simcoal2 (Laval and Excoffier, 2004), with a fast sequential Markovian coalescent (SMC) model for recombining DNA sequences (Marjoram and Wall, 2006). fastsimcoal retains simcoal2 full flexibility in setting up complex demographic scenarios, arbitrary recombination distances between markers, arbitrary migration rates between populations and historical events. In addition, fastsimcoal allows users to specify the sampling times of different population samples making it suitable for serial sampling

*To whom correspondence should be addressed.

2 METHODS AND IMPLEMENTATION

2.1 Continuous-time coalescent

The slow generation-by-generation approach implemented in simcoal2 has been replaced by a faster continuous-time coalescent framework similar to that implemented in ms (Hudson, 2002). Despite this change, we have kept the same parameterization of evolutionary scenarios as in simcoal2. Times of demographic events are still defined in generations, population sizes are defined in number of haploid individuals and mutation and recombination rates are defined per base pair per generation.

2.2 Sequential Markov coalescent

In recent years, some new approximations of the recombination process have been developed (Chen *et al*., 2009; Marjoram and Wall, 2006; McVean and Cardin, 2005) that allow one to simulate DNA sequences over hundreds of megabytes much faster than under the traditional ancestral recombination graph (ARG) model. The new SMC (McVean and Cardin, 2005) has been shown to produce patterns of polymorphisms and linkage disequilibrium extremely similar to those generated under a classical ARG model (Chen *et al*., 2009; Eriksson *et al*., 2009; Marjoram and Wall, 2006), while being much faster and demanding less memory resources. Under SMC, one generates a tree on the left end of the sequence under study, and computes the position of a recombination event on the right-hand side assuming an exponential distribution of recombination positions along the sequence. A recombination event is then implemented at random along the current tree, and the detached recombining lineage is then free to coalesce with the other remaining lineages, leading to a new tree with a potentially different topology and most recent common ancestor (MRCA). This procedure is continued until one reaches the end of the sequence to be generated. Some variations of this SMC' algorithm have been proposed to allow for coalescent events to occur between the detached lineage and the exact same branch that was cut by the recombination event [SMC' algorithm, Marjoram and Wall (2006)], or with trees that are further than one recombination away on the left-hand side (Chen *et al*., 2009).

We have implemented here the SMC' algorithm in a structured population with demes connected by migrations. For each tree, we thus record all migration events having occurred in addition to all coalescent events. These events are then replayed to generate the next tree, such that the detached recombinant lineage can migrate in any deme and potentially coalesce with lineages from the left tree that were present there at the same time.

2.3 Arbitrary recombination rates between markers

As in simcoal2, we have retained the possibility to simulate genetic markers at arbitrary recombination distances, such as to be able to model an array of markers at fixed locations on the genome or hot spots of recombination,

which is not directly possible with ms. We have thus implemented the possibility to have multiple recombination events between adjacent markers (Supplementary Fig. S1). This strategy results in patterns of linkage disequilibrium between pairs of markers that are virtually identical to those obtained under an ARG model (see Supplementary Fig. S3).

2.4 Serial sampling

Ancient DNA is becoming increasingly easy to extract and analyze, and when combined with present samples, it can shed new light on ancient demographic processes [see e.g. Ramakrishnan *et al*. (2005)]. Moreover, ignoring sampling heterochrony can lead to severe biases in demographic inferences and statistical tests (Depaulis *et al*., 2009). It therefore appears important to take into account uneven sampling time when using simulations to generate expected diversity. Like serial simcoal (Anderson *et al*., 2005), fastsimcoal now offers the possibility to define the age of each sample. Input file compatibility is preserved with simcoal2, by assuming present sampling when this age is omitted.

2.5 Sampling simulation parameters

Coalescent simulation programs are now central to several Approximate Bayesian Computation [ABC,Beaumont *et al*. (2002)] procedures to estimate demographic, mutation or recombination parameters from genetic data (e.g. Cornuet *et al*., 2008; Lopes *et al*., 2009; Wegmann *et al*., 2010). In order to facilitate this integration and efficiently perform simulations for a series of different parameter values, fastsimcoal can directly draw parameters from predefined prior distributions, or it can read parameter values directly from a definition file. These parameter values are then substituted to specific keywords present in a template input file, in a way similar to what is implemented in ABCToolBox (Wegmann *et al*., 2010). This integration is useful here since a single continuous-time coalescent simulation can be much faster than the calling of an external program that generates random parameter values. fastsimcoal now outputs site frequency spectra from DNA sequence data, which can be directly used as a summary statistic in ABC.

2.6 Input and output files

We have retained full compatibility with simcoal2 parameter files, and former input files should run without modifications with fastsimcoal. Additional options can now be explicitly given on the command line, according to the linux command line style. For instance, typing './fastsimcoal –i test.par –n 100' will launch 100 simulations of the model defined in input file *test.par*. './fastsimcoal –h' will list all possible command line options. Output files are similar to those generated by simcoal2, except that only polymorphic sites are now output for DNA sequences, with an indication of their position on the chromosome.

2.7 Benchmarks

fastsimcoal has been extensively tested against ms (Hudson, 2002), MaCS (Chen *et al*., 2009) and simcoal2, with and without recombination (Supplementary Tables S2–S4 and Figs S2 and S3). In summary, fastsimcoal leads to patterns of diversity and linkage disequilibrium identical to those produced by MaCS, while always being faster (up to nine times for small sample sizes). Both MaCS and fastsimcoal give patterns of genetic diversity very similar to ms, and are much faster with recombination. Additional comparisons between the SMC and forward approaches can be found in Chen *et al*. (2009). Other comparisons between forward simulations and continuous-time coalescent approaches (e.g. ms) can be found in Davies *et al*. (2007) or in Padhukasahasram *et al*. (2008).

3 DISCUSSION

While preserving all the simulation flexibility of simcoal2, fastsimcoal is based on a faster continuous-time SMC, similar to that implemented in MaCS (Chen *et al*., 2009), while being always faster than MaCS, especially in case of migration between populations and for small sample sizes ($n < 100$). In addition to MaCS, fastsimcoal allows one to (i) generate genetic data other than just DNA sequences [e.g. short tandem repeats (STRs), single nucleotide polymorphisms (SNPs)], (ii) simulate markers at arbitrary chromosome positions (e.g. by using a recombination map), (iii) simulate samples at different time periods (e.g. ancient DNA, or different viral strains) and it includes a parameter sampler allowing its integration into Bayesian parameter estimation procedures. Note however, that the use of the continuous-time coalescent approximation has been criticized for simulating diversity over long chromosomal segments, as it neglects the possibility of multiple and simultaneous coalescent events, the probability of which increases with recombination rate (Davies *et al*., 2007), and as it ignores overlapping recombinations in case of selfing (Padhukasahasram *et al*., 2008). However, despite these simplifying assumptions, patterns of diversities and linkage disequilibrium generated by continuous-time coalescent simulators are found very similar to those produced by forward and more realistic simulators (e.g. Davies *et al*., 2007; Padhukasahasram *et al*., 2008). For this reason, SMC-based simulators should thus be clearly favored for generating long stretches of DNA when computing time is an issue. Other ARG-based coalescent approaches like *ms* (Hudson, 2002), or those allowing for multiple and simultaneous coalescent events (e.g. Laval and Excoffier, 2004; Liang *et al*., 2007) may be preferred otherwise.

Funding: Swiss SNF (grant no. 3100A0-126074 to L.E.).

Conflict of Interest: none declared.

REFERENCES

Anderson,C.N. *et al.* (2005) Serial SimCoal: a population genetics model for data from multiple populations and points in time. *Bioinformatics*, **21**, 1733–1734.

Beaumont,M.A. *et al*. (2002) Approximate Bayesian computation in population genetics. *Genetics*, **162**, 2025–2035.

Carvajal-Rodriguez,A. (2008) Simulation of genomes: a review. *Curr. Genomics*, **9**, 155–159.

Chen,G.K. *et al*. (2009) Fast and flexible simulation of DNA sequence data. *Genome Res.*, **19**, 136–142.

Cornuet,J.M. *et al.* (2008) Inferring population history with DIY ABC: a user-friendly approach to approximate Bayesian computation. *Bioinformatics*, **24**, 2713–2719.

Davies,J.L. *et al.* (2007) On recombination-induced multiple and simultaneous coalescent events. *Genetics*, **177**, 2151–2160.

Depaulis,F. *et al*. (2009) Using classical population genetics tools with heterochroneous data: time matters! *PLoS ONE*, **4**, e5541.

Durbin,R.M. *et al.* (2010) A map of human genome variation from population-scale sequencing. *Nature*, **467**, 1061–1073.

Eriksson,A. *et al*. (2009) Sequential Markov coalescent algorithms for population models with demographic structure. *Theor. Popul. Biol.*, **76**, 84–91.

Ewing,G. and Hermisson,J. (2010) MSMS: a coalescent simulation program including recombination, demographic structure and selection at a single locus. *Bioinformatics*, **26**, 2064–2065.

Hudson,R.R. (1990) Gene genealogies and the coalescent process. In Futuyma, D.J. and Antonovics, J.D. (eds), *Oxford Surveys in Evolutionary Biology*. Oxford University Press, New York, pp. 1–44.

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

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**, 2485–2487.

Liang,L. *et al*. (2007) GENOME: a rapid coalescent-based whole genome simulator. *Bioinformatics*, **23**, 1565–1567.

Lopes,J.S. *et al*. (2009) PopABC: a program to infer historical demographic parameters. *Bioinformatics*, **25**, 2747–2749.

Marjoram,P. and Wall,J.D. (2006) Fast 'coalescent' simulation. *BMC Genet.*, **7**, 16.

McVean,G.A. and Cardin,N.J. (2005) Approximating the coalescent with recombination. *Philos. Trans. R Soc. Lond. B Biol. Sci.*, **360**, 1387–1393.

Padhukasahasram,B. *et al.* (2008) Exploring population genetic models with recombination using efficient forward-time simulations. *Genetics*, **178**, 2417–2427.

Ramakrishnan,U. *et al*. (2005) Detecting past population bottlenecks using temporal genetic data. *Mol. Ecol.*, **14**, 2915–2922.

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

Wegmann,D. *et al.* (2010) ABCtoolbox: a versatile toolkit for approximate Bayesian computations. *BMC Bioinformatics*, **11**, 116.