BIOINFORMATICS APPLICATIONS NOTE Vol. 23 no. 12 2007, pages 1565–1567
doi:10.1093/bioinformatics/btm138

Genetics and population analysis

GENOME: a rapid coalescent-based whole genome simulator

Liming Liang*, Sebastian Zöllner and Gonçalo R. Abecasis

Center for Statistical Genetics, Department of Biostatistics, University of Michigan, 1420 Washington Heights, Ann Arbor, MI 48109, USA

Received and revised on March 14, 2007; accepted on April 3, 2007

Advance Access publication April 25, 2007

Associate Editor: Martin Bishop

ABSTRACT

Summary: GENOME proposes a rapid coalescent-based approach to simulate whole genome data. In addition to features of standard coalescent simulators, the program allows for recombination rates to vary along the genome and for flexible population histories. Within small regions, we have evaluated samples simulated by GENOME to verify that GENOME provides the expected LD patterns and frequency spectra. The program can be used to study the sampling properties of any statistic for a whole genome study.

Availability: The program and C++ source code are available online at http://www.sph.umich.edu/csg/liang/genome/

Contact: lianglim@umich.edu

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

1 INTRODUCTION

The coalescent approach (Donnelly and Tavaré, 1995; Hudson 1983, 1990; Kingman, 1982) is an efficient way to sample of sequences from a theoretical population that follows the Wright–Fisher neutral model (Ewens, 1979). Simulations based on coalescent models have also been used to study the sampling properties of interesting statistics or evaluate new methods. Applications include the inference of population history (Weiss and von Haeseler, 1998), the study of positive selection (Przeworski, 2002, Voight *et al*., 2006) and whole genome linkage disequilibrium mapping of common disease genes (Kruglyak, 1999; Zöllner and von Haeseler, 2000). Existing software packages, such as ms (Hudson, 2002) and cosi (Schaffner *et al*., 2005), implement the standard coalescent approach which simulates genealogical events backward in time. Simulated events typically include the coalescence of two sequences into a single ancestral lineage, recombination within a sequence or migration between populations. Since all these events are typically rare, coalescent simulators assume that they never occur simultaneously and assume many generations pass between consecutive events. Time between events is explicitly modeled and used to skip over generations with no genealogical events of interest. The algorithm proceeds until all sequences coalesce to their most recent common ancestor and the resulting genealogy is used to place mutation events along the various sequences.

The standard approach is extremely efficient when simulating short sequences. As sequences get longer, many more coalescent, recombination and migration events occur and the time intervals between them diminish. For longer sequences, little computational efficiency is gained by skipping over uninteresting generations and substantial computational effort is expended tracking recombination events and their positions and allocating memory to track the many ancestral fragments of each sequence as they repeatedly recombine and coalesce with each other. Overall, the standard coalescent approach which is suitable for short genomic segments (<2–3 Mb) becomes very slow for larger regions (>100 Mb).

As genome-wide studies gather more interest and become a reality, efficient tools for simulating large sequences are essential to study the sampling properties of arbitrary statistics that might be evaluated on a genome-wide association study and to compare the performance of different methods that may be applied to genome-wide scale data. For example, in an ongoing study we are evaluating the distribution of stretches of haplotype shared among a majority of individuals with disease and need an efficient coalescent framework to evaluate the null distribution of the statistic. There is great interest in developing fast coalescent simulators to address this and similar problems; one potential speedup involves making further simplifying assumptions about the genealogy (Marjoram and Wall, 2006). Here, we propose an alternative framework for the coalescent that allows efficient simulation of genealogies for long sequences and still fully captures the complexity of the genealogy. In our approach, the genealogy of sampled sequences is simulated backwards in time, one generation at a time, in a procedure that is computationally efficient and removes the bifurcate tree approximation (in the standard approach, each coalescence event involves exactly two sequences that coalesce to a common ancestor but, using our approach, multiple sequences can coalesce to a common ancestor simultaneously). When multiple sub-populations are simulated, the program allows for migration among subpopulations and for user-specified demographic events such as population bottlenecks and expansions or population merges and splits. We allow recombination rates to vary so as to mimic the pattern of hotspots along the genome. As in the standard

*To whom correspondence should be addressed.

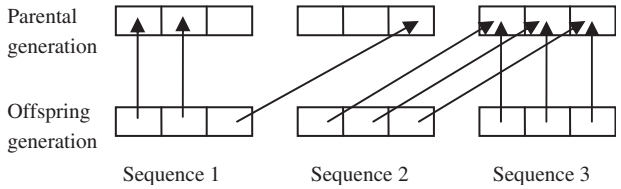

Fig. 1. Discrete generation implementation. In the example, each sequence is divided into three fragments. There is a recombination event between the second and the third fragments in sequence 1. Sequences 2 and 3 coalesce.

coalescent approach, mutations are simulated assuming an infinite-sites model.

2 DESCRIPTION

As in the standard coalescent approach, we simulate the genealogy of a sample of sequences, conditional on parameters such as the population size, the recombination rate and rates of migration between subpopulations. Instead of simulating the time to next event, we simulate the coalescent and recombination events at every generation proceeding backwards in time (Fig. 1). For each generation, the sequences are stored in a sparse matrix where rows correspond to individuals and columns correspond to short stretches of sequence. The matrix is sparse because only portions of sequence with a descendant in the final generation are tracked. We allocate two sparse matrices in memory (the current and the previous generation, which are reused) together with a separate structure summarizing coalescent events for each portion of the sequence. To allow for population stratification or other constraints on mating, we define a set of rules that can be used to relate each individual sequence (a row in one of the sparse matrices) to its ancestors in the previous generation (one or more rows in the second sparse matrix). Since we simulate all intervening generations, these rules can be quite sophisticated—to enforce multiple populations, geographic relationships between subpopulations, diploid individuals (so that each sequence has exactly two ancestors), etc. These features are commonly only found in forward simulators, which are computationally much less efficient.

Because our approach can simulate multiple coalescent and recombination events in the same generation, it naturally accommodates situations where the number of sequences sampled approximates the effective population size or where the sequences are very long. Conditional on the genealogy tree, mutations are placed on the branches. The number of mutations on each branch follows a Poisson distribution with mean equal to the product of the mutation rate and the branch length in generations. The infinite-site mutation model is assumed. As with many other coalescent simulators, we also allow the number of mutations to be fixed so that the probability that a mutation occurs on a particular branch is proportional to its length. Varying recombination rate and population histories can also be specified by parameter files. The output distinguishes ancestral state and derived alleles and is similar to the output format of the ms program. The genealogy trees for each fragment in Newick format can be output, ready for plotting with PHYLIP (Felsenstein, 2005) or for use with seq-gen, a sequence evolution program by Rambaut and Grassly (1997). Detailed instructions and examples are available on our website.

3 COMPARISON WITH EXISTING SOFTWARE PACKAGE

To evaluate our simulator, we first compared the generated allele frequency spectra with theoretical expectations. Using a goodness-of-fit test, we observed no significant differences between the expected spectra and those generated by GENOME. We have compared our simulated samples with those generated by Hudson's ms (simulating a 2 Mb region). The two simulators provide similar LD patterns and frequency spectra (detailed results available from our website). When simulating long regions, GENOME is substantially faster than ms. For example, when simulating a sample of 1200 chromosomes, each 150 Mb long, from two populations of size 10 000, GENOME requires ~66 min, compared to >12 h for Hudson's ms (using a standard 2.8 GHz Pentium CPU). The scaled rates of mutation, recombination and migration were set to $4N\mu = 60\,000$, $4Nr = 60\,000$ and $4Nm = 10$ in the simulation described, but more details together with comparisons for other settings are available on our website. As expected, GENOME also outperforms cosi, a coalescent simulator that allows for flexible recombination rates and is somewhat slower than ms.

4 IMPLEMENTATION AND COMPUTATIONAL ISSUES

The program is written in C++ and is portable to a variety of operating systems, including Windows, Linux and MacOS. The Mersenne Twister Code (Matsumoto and Nishimura, 1998) is used as the source of random deviates. In addition to a stand-alone version, our simulator is also provided as a C++ function 'genome()', that can be incorporated as a module in other programs.

ACKNOWLEDGEMENTS

This research was supported in part by research grants from the National Heart Lung and Blood Institute and the National Human Genome Research Institute. G.R.A. is a Pew Scholar for the Biomedical Sciences and is supported by the Pew Charitable Trusts.

Conflict of Interest: none declared.

REFERENCES

Donnelly,P. and Tavaré,S. (1995) Coalescents and genealogical structure under neutrality. *Annu. Rev. Genet.*, **29**, 401–421.

Ewens,W.J. (1979) *Mathematical Population Genetics*. Springer, Berlin.

Felsenstein,J. (2005) PHYLIP (Phylogeny Inference Package) version 3.6. Distributed by the author. Department of Genome Sciences, University of Washington, Seattle.

Hudson,R.R. (1983) Properties of a neutral allele model with intragenic recombination. *Theor. Popul. Biol.*, **23**, 183–201.

Hudson,R.R. (1990) Gene genealogies and the coalescent process. *Oxf. surv. evol. biol.*, **7**, 1–44.

Hudson,R.R. (2002) Generating samples under a Wright-Fisher neutral model. *Bioinformatics*, **18**, 337–378.

Kingman,J.F.C. (1982) The coalescent. *Stochastic Process. Appl.*, **13**, 235–248.

Kruglyak,L. (1999) Prospects for whole-genome linkage disequilibrium mapping of common disease genes. *Nat. Genet.*, **22**, 139–144.

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

Matsumoto,M. and Nishimura,T. (1998) Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator. *ACM Trans. on Modeling and Computer Simulation*, **8**, 3–30.

Przeworski,M. (2002) The signature of positive selection at randomly chosen loci. *Genetics*, **160**, 1179–1189.

Rambaut,A. and Grassly,N.C. (1997) Seq-gen: An application for the monte carlo simulation of dna sequence evolution along phylogenetic trees. *Comput. Appl. Biosci.*, **13**, 235–238.

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

Voight,B.F. *et al.* (2006) A map of recent positive selection in the human genome. *PLoS Biol.*, **4**, e72.

Weiss,G. and von Haeseler,A. (1998) Inference of population history using a likelihood approach. *Genetics*, **149**, 1539–1546.

Zöllner,S. and von Haeseler,A. (2000) A coalescent approach to study linkage disequilibrium between single-nucleotide polymorphisms. *Am. J. Hum. Genet.*, **66**, 615–628.