A survey of genetic simulation software for population and epidemiological studies

Youfang Liu,[1] Georgios Athanasiadis[2] and Michael E. Weale[3]

[1]Bioinformatics Research Center, North Carolina State University, Campus Box 7566, Raleigh, NC 27695-7566, USA
[2]Facultat de Biologia, Departament de Biologia Animal, Facultat de Biologia, Universitat de Barcelona, Av. Diagonal 645, 08028, Barcelona, Spain
[3]Department of Medical and Molecular Genetics, King's College London, 8th Floor, Tower Wing, Guy's Hospital, London, SE1 9RT, UK
**Correspondence to:* Tel: +44 (0)20 7188 2601; Fax: +44 (0)20 7188 2585; E-mail: michael.weale@kcl.ac.uk

Date received: 17th June, 2008

Abstract

A number of programs have been developed for simulating population genetic and genetic epidemiological data conforming to one of three main algorithmic approaches: 'forwards', 'backwards' and 'sideways'. This review aims to make the reader aware of the range of options currently available to them. While no one program emerges as the best choice in all circumstances, we nominate a set of those which currently appear most promising.

Keywords: *Population genetics, genetic epidemiology, simulation software*

Introduction

The two main reasons for wanting to simulate genetic data are, first, to gain insight into the effects that underlying demographic and mutational parameters may have on the genetic data one sees, and, secondly, to create test datasets for assessing the power of alternative genetic analysis methods. Ways of tackling the first goal range from informal approaches, which aim at getting a 'feel' for how altering different parameters affects the output data, to more formal methods based on matching many simulated datasets to an observed dataset (eg approximate Bayesian computation[1]). To tackle the second goal (and particularly for genetic epidemiology methods), an additional 'ascertainment' modelling element is often required to allow the simulation of disease-affecting loci within the context of a given study design (such as a case-control study).

The key challenges that all simulation algorithms face are: (1) speed — typically one wants to do lots of simulations, so they need to be fast; (2) scalability — with the advent of genome-wide genotyping and large-scale sequencing, there is a need for simulation programs to match; and (3) flexibility — can the program cope with different demographic histories, population structure, recombination, selection, mutation models and disease models?

There are three main approaches to dealing with these challenges, here termed 'backwards', 'forwards' and 'sideways'. 'Backwards' (or coalescent) simulations start with the sample of individuals that will form your simulated dataset, then work backwards in time to construct the ancestral tree or graph of genealogical relationships that connects them all. Neutral mutations can subsequently be placed on this structure to create the simulated dataset. The simulation algorithm does not actually have to work backwards in time to achieve this, but this is a technical detail. The important point is that by restricting attention just to the genealogical

structure relevant to the sample in question, a large computational saving is generally achieved relative to the 'forwards-in-time' approach. Still greater efficiency is afforded by the classic coalescent approach, which employs a continuous-time approximation to effectively skip over the intermediate generations between important tree-generating events. 'Forwards' simulations start with the entire population of individuals — typically, many thousands — and then follow how all the genetic data in question are passed on from one generation to the next. One usually needs to simulate over many thousands of generations in order to arrive at an equilibrium in which the genetic characteristics of the population are independent of the original starting conditions. Finally, 'sideways' simulations start with a collection of real present-day genetic data, and use these as a template for generating new simulated data with similar properties. 'Sideways' algorithms can also be coalescent-based (and thus fit into both 'backwards' and 'sideways' categories) but some adopt simpler resampling strategies that do not explicitly consider changes over generational time in either direction.

Backwards simulators

Table 1 lists all programs that the authors were able to source via PubMed and other internet-based searches. A list maintained by Heng Li[35] was also helpful. Backwards (coalescent) approaches form the largest part of Table 1, reflecting the inherent attractiveness and computational efficiency of simulating just that part of the genealogy needed to produce the data in the simulated sample. Richard Hudson's *ms* program[8] remains one of the most popular for straightforward problems. msHOT,[9] *SNPsim*[17] and *COSI*[3] extend the algorithm to allow variable recombination rates along the DNA sequence, and *msHOT*, *COSI*, *CoaSim*[2] and *newgenecoal*[10] also allow (allelic) gene conversion in addition to crossovers as recombination events. *SIMCOAL*[15] introduces complex demographic models, *SIMCOAL2*[16] extends this to variable recombination, *Serial SIMCOAL*[14] allows sampling at multiple time points and *MODELER4SIMCOAL*[36,37] provides a handy graphical user interface. *SelSim*[13] implements a single-locus selection model. Flexible, but not necessarily easy to implement, coalescent simulators are provided by *CoaSim*,[2] *mlcoalsim*,[7] *SARG*[12] and *GeneArtisan*.[5]

For neutral loci, the tree or graph-generating step can be conveniently decoupled from the mutation-generating step, and the latter can be run via a separate program such as Andy Rambaut's *SeqGen* program[38] to produce a wide range of different types of genetic data from a range of different mutational models. It is also possible to decouple the sampling ascertainment process (eg to get case-control data) by applying this as an additional step to unascertained simulated data. Currently, however, there are no easy ways of doing this, as additional user coding would be needed to adapt the sampling algorithms available in, for example, CoaSim,[2] *SimuPOP*[28–30] or *FREGENE*.[23] Furthermore, there are as yet no completely flexible ascertainment options that would allow, for example, simulation of cases from models with more than one partially linked disease locus, or from more general causal models that have incorporated additional covariates.

Conventional coalescent algorithms break down for very large DNA regions such as whole chromosomes. This is because recombination gives rise to complex ancestral recombination graphs (ARGs) rather than simple binary genealogical trees, and more recombination leads to ever larger and more complex ARGs. The *FastCoal*[4] and *GENOME*[6] simulators employ approximations to the real coalescent-with-recombination that lead to simpler ARGs and thence to feasible genome-wide simulations. *MaCs*, a recent update to *FastCoal* which uses an improved approximation to the coalescent-with-recombination, is available on request from Jeff Wall (wallj@humgen.ucsf.edu). *FastCoal* is reported to be able to generate 2,000 50-megabase (Mb) diploid samples in two minutes on a standard workstation, and *GENOME* to generate 600 150 Mb diploid samples in 66 minutes.

Forwards simulators

Forwards-in-time simulators are more naturally capable of coping with complex modelling

Table 1. List of population genetic simulation software

Name, description and URL	Lang.	Loc.	R?	S?	~N?	G?	D?
Backwards simulators							
CoaSim[2] Requires additional user coding for more complex models. http://www.daimi.au.dk/~mailund/CoaSim/	Scheme or Python	s,c	•		•		•
COSI[3] Allows variable recombination. http://www.broad.mit.edu/~sfs/cosi	C	s,c	•				
FastCoal[4] Variable recombination and mutation models require additional post-processing. http://chp220mac.hsc.usc.edu/Marjoram/Software.html	C++	s,c	•		•	•	
GeneArtisan[5] Flexible disease models. Selection allowed. http://www.rannala.org/	C++	s,c	•	•	•		•
GENOME[6] Infinite sites discrete-generation model. http://www.sph.umich.edu/csg/liang/genome/	C++	s	•		•	•	
mlcoalsim[7] Flexible extensions to *ms* program. http://www.ub.es/softevol/mlcoalsim	C	s,c	•	•	•		
ms[8] Efficient, well-used program. Can pipe into *SeqGen* for additional mutation models. http://home.uchicago.edu/~rhudson1/source/mksamples.html	C	s,c	•				
msHOT[9] Allows variable recombination. Can pipe to *SeqGen*. http://home.uchicago.edu/~rhudson1/source/mksamples.html	C	s,c	•				
newgenecoal[10] Specialist software for gene duplications and large CNVs. http://molpopgen.org/software/coalescent.html	C	s,c	•				
Recodon[11] Designed for exons — allows codon-specific mutation model. http://darwin.uvigo.es/	C	s	•		•		

Continued

Table 1. Continued

Name, description and URL	Lang.	Loc.	R?	S?	~N?	G?	D?
SARG[12] Flexible. Mathematica front-end. http://walnut.usc.edu/Members/magnus/software/software/	C++	s	•	•	•		
SelSim[13] Selected site must be diallelic, but various mutational models allowed for other sites. http://mathgen.stats.ox.ac.uk/software.html	C++	s,c	•	•			
Serial SimCoal[14] Allows sampling from different time points. http://iod.ucsd.edu/simplex/ssc/	C++	s,c			•		
SIMCOAL[15] Discrete-generation model. Flexible demographic options. http://cmpg.unibe.ch/software/simcoal/	C++	s,c			•		
SIMCOAL2[16] Allows variable recombination and >1 coalescence per generation. http://cmpg.unibe.ch/software/simcoal2/	C++	s,c	•		•		
SNPsim[17] Infinite sites model. http://darwin.uvigo.es/software/snpsim.html	C	s	•		•		
SPLATCHE[18] Complex demographies via 2D demic isolation-by-distance model. http://cmpg.unibe.ch/software/splatche/	C++	s,c			•		
TREEVOLVE[19] Finite sites model. http://evolve.zoo.ox.ac.uk/software.html?name=Treevolve	C	s	•		•		
Forward simulators							
BottleSim[20] Simulates population bottlenecks. No mutational models. http://chkuo.name/software/BottleSim.html	C++	–			•		
EasyPOP[21] Allows complex demographies. http://www.unil.ch/dee/page36926_fr.html	C	s,c	•		•		

Continued

Table 1. Continued

Name, description and URL	Lang.	Loc.	R?	S?	~N?	G?	D?
FORWSIM[22] Efficient 'look-ahead' procedure, but limited model options. http://www.people.cornell.edu/pages/bp85/	C++	s				•	
FPG Infinite sites model. http://lifesci.rutgers.edu/~heylab/heylabsoftware.htm	C	s	•	•			
FREGENE[23] Diallelic loci only, finite sites allowed. Rescaling option adds computational efficiency. www.ebi.ac.uk/projects/BARGEN	C++	s	•	•	•	•	•
GenomePop[24] Flexible but no ascertained sampling. Rescaling option adds computational efficiency. http://webs.uvigo.es/acraaj/GenomePop.htm	C++	s	•	•	•	•	
GenomeSIM[25] Genome-wide data only possible if small N generations simulated. http://chgr.mc.vanderbilt.edu/genomeSIMLA/genomeSIMLA/Introduction.html	C++	s	•			•	•
Mendel's Accountant[26] Flexible mutation and selection options. http://mendelsaccountant.info/	C++	s	•	•	•		•
Nemo[27] Flexible demographic options, requires user coding. http://nemo2.sourceforge.net/	C++	s,c	•	•	•		•
SFS_CODE Finite sites model. http://cbsuapps.tc.cornell.edu/sfscode.aspx or http://sfscode.sourceforge.net/	C	s	•	•	•		
simuPOP[28–30] Requires additional user coding for more complex models. http://simupop.sourceforge.net/	Python	s,c	•	•	•		•
Sideways simulators							
gs[31] Stochastic haplotype extension method. Two-locus unlinked disease model. http://vorlon.case.edu/~jxl175/gs.html	C++	s		na	na	•	•

Continued

Table 1. Continued

Name, description and URL	Lang.	Loc.	R?	S?	∼N?	G?	D?
GWAsimulator[32] Stochastic haplotype extension method. Multi-locus unlinked disease model. http://biostat.mc.vanderbilt.edu/GWAsimulator	C++	s		na	na	•	•
hapgen[33] Coalescent-with-recombination approximation generates 'missing data'. http://www.stats.ox.ac.uk/~marchini/software/gwas/hapgen.html	C	s	•				•
HAP-SAMPLE[34] Bootstrap resampling from HapMap plus a single generation of recombination. http://www.hapsample.org/	Web-based	s	•	na	na	•	•

Key:
Lang. Programming language
Loc. Locus types allowed. s = SNPs, c = CNVs (including microsatellites)
R? Recombination allowed?
S? Selection allowed?
∼N? Variable population size and/or population structure allowed?
G? Genome-wide data possible?
D? Disease model allowed? (for ascertained samples such as case-control)
na Not applicable
CNV Copy number variation

scenarios, at the expense of decreased computational efficiency. Of these, the *FREGENE*[23] and *GenomePop*[24] programs make the biggest effort at maintaining speed, and, of these, only *FREGENE* allows for ascertained disease-gene sampling. A useful scaling option in both programs allows one to simulate a smaller population over a smaller number of generations and then use these results to approximate a larger population over more generations. Unfortunately, only diallelic SNP data can be simulated fast enough to cover large genomic regions. At smaller genomic scales, more complex nucleotide and codon models can be simulated by *GenomePop*, while copy number variation (CNV) and microsatellite data can be simulated by *simuPOP*[28–30] and *Nemo*.[27] *GenomeSIM*[25] claims to be able to generate genome-wide SNP data by forwards simulation, but only achieves this by simulating over a very limited ten or so generations, far fewer than that needed to achieve proper genetic equilibrium. Indeed, *FREGENE* and *GenomePop* could also generate genome-wide datasets in this way, and presumably could do so with greater computational efficiency.

Sideways simulators

Sideways simulators can, to some extent, side-step the whole issue of model complexity by relying on real data 'as is' to guide the simulation process. Simple bootstrap resampling breaks down for longer regions because the genetic diversity seen in the reference sample (usually the 270 individuals in HapMap) is not adequate to capture the full diversity among all humans. The situation will improve with the '1,000 genomes' project,[39] and also with the steady increase in publically available genome-wide SNP data, but it still seems sensible to apply an additional method to perturb the simulated data away from the narrow range seen in the real data. Dudbridge[40] proposed forming random diploid chromosomes from phased HapMap data followed by a single round of artificial meiosis, governed by empirical recombination rates also estimated from HapMap. This idea has been put to use in the *HAP-SAMPLE* software,[34] with an additional option to boost the baseline recombination rate (×100 recommended) to reduce long-range linkage disequilibrium. Durrant *et al.*[41] proposed an alternative idea based on sliding windows for introducing new variations into simulated data. This method has been implemented in the *GWA simulator* software,[32] and an improved extension to this idea, which allows a variable sliding window size, has been implemented in the *gs* software.[31] Jonathan Marchini's *hapgen* software,[33] based on the same underlying principles as his genotype imputation software *impute*, applies an approximation to the coalescent-with-recombination to generate new simulated data from existing phased HapMap data, but is slower than the other two sideways simulators. *HAP-SAMPLE* is reported to be able to generate 2,000 samples of a 100,000 genome-wide SNP chip in a few minutes on a standard workstation, and *gs* to generate 2,000 samples of chromosome 6 (36,000 SNPs) in 140 minutes.

Conclusions

In summary, no one program is capable of doing everything, but there exist some useful applications from all three main simulation approaches. For genome-wide SNP data, the main contenders are *FastCoal*,[4] *GENOME*,[6] *HAP-SAMPLE*[34] and *gs*.[31] For high model flexibility and sampling ascertainment at the 10 Mb scale or less (not whole-genome but still enough for many purposes), *FREGENE*[23] is recommended. Simulation of copy number variation and/or microsatellite data at larger genomic scales, and of more complex disease models allowing covariates and linked loci, remain areas for future program development.

References

1. Beaumont, M.A., Zhang, W. and Balding, D.J. (2002), 'Approximate Bayesian computation in population genetics', *Genetics* Vol. 162, pp. 2025–2035.
2. Mailund, T., Schierup, M.H., Pederson, C.N.S. *et al.* (2005), 'CoaSim: A flexible environment for simulating genetic data under coalescent models', *BMC Bioinformatics* Vol. 6, p. 252.
3. Schaffner, S.F., Foo, C., Gabriel, S. *et al.* (2005), 'Calibrating a coalescent simulation of human genome sequence variation', *Genome Res.* Vol. 15, pp. 1576–1583.

4. Marjoram, P. and Wall, J.D. (2006), 'Fast "coalescent" simulation a flexible environment for simulating genetic data under coalescent models', *BMC Genet*. Vol. 7, p. 16.
5. Wang, Y. and Rannala, B. (2005), 'In silico analysis of disease-association mapping strategies using the coalescent process and incorporating ascertainment and selection', *Am. J. Hum. Genet*. Vol. 76, pp. 1066–1073.
6. Liang, L., Zöllner, S. and Abecasis, G.R. (2007), 'GENOME: A rapid coalescent-based whole genome simulator', *Bioinformatics* Vol. 23, pp. 1565–1567.
7. Ramos-Onsins, S.E. and Mitchell-Olds, T. (2007), 'Mlcoalsim: Multilocus coalescent simulations', *Evol. Bioinformatics* Vol. 2, pp. 41–44.
8. Hudson, R.R. (2005), 'Generating samples under a Wright–Fisher neutral model of genetic variation', *Bioinformatics* Vol. 18, pp. 337–338.
9. Hellenthal, G. and Stephens, M. (2007), 'msHOT: modifying Hudson's ms simulator to incorporate crossover and gene conversion hotspots', *Bioinformatics* Vol. 23, pp. 520–521.
10. Thornton, K.R. (2007), 'The neutral coalescent process for recent gene duplications and copy-number variants', *Genetics* Vol. 177, pp. 987–1000.
11. Arenas, M. and Posada, D. (2007), 'Recodon: Coalescent simulation of coding DNA sequences with recombination, migration and demography', *BMC Bioinformatics* Vol. 8, p. 458.
12. Nordborg, M. and Innan, H. (2003), 'The genealogy of sequences containing multiple sites subject to strong selection in a subdivided population', *Genetics* Vol. 163, pp. 1201–1213.
13. Spencer, C.C. and Coop, G. (2004), 'SelSim: A program to simulate population genetic data with natural selection and recombination', *Bioinformatics* Vol. 20, pp. 3673–3675.
14. Anderson, C.N., Ramakrishnan, U., Chan, Y.L. and Hadly, E.A. (2005), 'Serial SimCoal: A population genetics model for data from multiple populations and points in time. *Bioinformatics* Vol. 21, pp. 1733–1734.
15. Excoffier, L., Novembre, J. and Schneider, S. (2000), 'SIMCOAL: A general coalescent program for the simulation of molecular data in interconnected populations with arbitrary demography', *J. Hered*. Vol. 91, pp. 506–509.
16. 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* Vol. 20, pp. 2485–2487.
17. Posada, D. and Wiuf, C. (2003), 'Simulating haplotype blocks in the human genome', *Bioinformatics* Vol. 19, pp. 289–290.
18. Currat, M., Ray, N. and Excoffier, L. (2004), 'SPLATCHE: A program to simulate genetic diversity taking into account environmental heterogeneity', *Mol. Ecol. Notes* Vol. 4, pp. 139–142.
19. Grassly, N.C., Harvey, P.H. and Holmes, E.C. (1999), 'Population dynamics of HIV-1 inferred from gene sequences', *Genetics* Vol. 151, pp. 427–438.
20. Kuo, C.-H. and Janzen, F.J. (2003), 'BOTTLESIM: A bottleneck simulation program for long-lived species with overlapping generations', *Mol. Ecol. Notes* Vol. 3, pp. 669–673.
21. Balloux, F. (2001), 'EASYPOP (Version 1.7): A computer program for population genetics simulations', *J. Hered*. Vol. 92, pp. 301–302.
22. Padhukasahasram, B., Marjoram, P., Wall, J.D. *et al.* (2008), 'Exploring population genetic models with recombination using efficient forward-time simulations', *Genetics* Vol. 178, pp. 2417–2427.
23. Hoggart, C.J., Chadeau-Hyam, M., Clark, T.G. *et al.* (2007), 'Sequence-level population simulations over large genomic regions', *Genetics* Vol. 177, pp. 1725–1731.
24. Carvajal-Rodríguez, A. (2008), 'GENOMEPOP: A program to simulate genomes in populations', *BMC Bioinformatics* Vol. 9, p. 223.
25. Dudek, S.M., Hotsinger, A.A., Velez, D.R. *et al.* (2006), 'Data simulation software for whole-genome association and other studies in human genetics', *Pac. Symp. Biocomput*. Vol. 11, pp. 499–510.
26. Sanford, J., Baumgardner, J., Brewer, W. *et al.* (2007), 'Mendel's Accountant: A biologically realistic forward-time population genetics program', *S.C.P.E.* Vol. 8, pp. 147–165. Available at: http://www.scpe.org/.
27. Guillaume, F. and Rougemont, J. (2006), 'Nemo: An evolutionary and population genetics programming framework', *Bioinformatics* Vol. 22, pp. 2556–2557.
28. Peng, B., Amos, C.I. and Kimmel, M. (2007), 'Forward-time simulations of human populations with complex diseases', *PLoS Genet*. Vol. 3, p. e47.
29. Peng, B. and Kimmel, M. (2005), 'simuPOP: A forward-time population genetics simulation environment', *Bioinformatics* Vol. 21, pp. 3686–3687.
30. Peng, B. and Amos, C.I. (2008), 'Forward-time simulations of nonrandom mating populations using simuPOP', *Bioinformatics* Vol. 24, pp. 1408–1409.
31. Li, J. and Chen, Y. (2008), 'Generating samples for association studies based on HapMap data', *BMC Bioinformatics* Vol. 9, p. 44.
32. Li, C. and Li, M. (2008), 'GWAsimulator: A rapid whole-genome simulation program', *Bioinformatics* Vol. 24, pp. 140–142.
33. Marchini, J., Howie, B., Myers, S. *et al.* (2007), 'A new multipoint method for genome-wide association studies by imputation of genotypes', *Nat. Genet*. Vol. 39, pp. 906–913.
34. Wright, F.A., Huang, H., Guan, X. *et al.* (2007), 'Simulating association studies: A data-based resampling method for candidate regions or whole genome scans', *Bioinformatics* Vol. 23, pp. 2581–2588.
35. http://www.sanger.ac.uk/Users/lh3/coal-simu.html.
36. Antao, T., Beja-Pereira, A. and Luikart, G. (2007), 'MODELER4SIMCOAL2: A user-friendly, extensible modeler of demography and linked loci for coalescent simulations. *Bioinformatics* Vol. 23, pp. 1848–1850.
37. http://popgen.eu/soft/m4s2/.
38. http://tree.bio.ed.ac.uk/software/seqgen/.
39. http://www.1000genomes.org/.
40. Dudbridge, F. (2006), 'A note on permutation tests in multistage association scans', *Am. J. Hum. Genet*. Vol. 78, pp. 1094–1095.
41. Durrant, C., Zondervan, K.T., Cardon, L.R. *et al.* (2004), 'Linkage disequilibrium mapping via cladistic analysis of single-nucleotide polymorphism haplotypes', *Am. J. Hum. Genet*. Vol. 75, pp. 35–43.