BIOINFORMATICS APPLICATIONS NOTE Vol. 18 no. 2 2002
Pages 337–338

Generating samples under a Wright–Fisher neutral model of genetic variation

Richard R. Hudson

Department of Ecology and Evolution, University of Chicago, 1101 E. 57th Street, Chicago, IL 60637, USA

Received on August 8, 2001; revised and accepted on September 13, 2001

ABSTRACT

Summary: A Monte Carlo computer program is available to generate samples drawn from a population evolving according to a Wright–Fisher neutral model. The program assumes an infinite-sites model of mutation, and allows recombination, gene conversion, symmetric migration among subpopulations, and a variety of demographic histories. The samples produced can be used to investigate the sampling properties of any sample statistic under these neutral models.

Availability: The source code for the program (in the language C) is available at http://home.uchicago.edu/~rhudson1/source/mksamples.html.

Contact: rr-hudson@uchicago.edu

INTRODUCTION

Wright–Fisher models are population genetic models with finite population size, discrete generations and multinomial sampling to produce successive generations (Ewens, 1979). These stochastic models constitute an important class of models for the interpretation of molecular variation within populations. Samples drawn from such populations have randomness due to an evolutionary (or population level) component and a sampling component. Statistical properties of such samples are frequently very difficult to obtain from analytical or numerical methods. Consequently, a program to rapidly generate independent samples from populations evolving accord to these models can be of great use for studying the statistical properties of such samples. A program has been written in C, which can efficiently generate such samples under a variety of assumptions using a coalescent approach (Hudson, 1983, 1990). The program allows recombination, gene conversion, migration under a symmetric island model, and simple population size changes.

THE MODEL AND ITS PARAMETERS

The program assumes the standard coalescent approximation to the Wright–Fisher model. This approximation is excellent as long as the sample sizes considered are small relative to the population size. Only models without selection are considered. For each sample, the program generates a random genealogical history of a segment of a chromosome. Conditional on the genealogy of a sample, mutations are randomly placed on the genealogy according to the usual assumption that the number of mutations on a branch is Poisson distributed with mean given by the product of the mutation rate and the branch length. The times between nodes in the genealogy are approximated by continuous distributions. In the case of constant population size, these times are sums of one or more exponentially distributed times. The infinite-sites mutation model is assumed, and thus no recurrent mutation occurs. In the output, the ancestral state of an allele is represented by zero, and the derived (or mutated state by 1). The segment of the chromosome being simulated is represented by the interval (0, 1). The positions of the polymorphic sites, where mutations have occurred somewhere in the genealogy of the sample, are specified in the output on this (0, 1) scale. The mutation parameter for this model is $4N_0u$ where N_0 is the effective diploid population size and u is the neutral mutation rate for the entire segment being modeled. To simulate samples for a constant population size, without subpopulations and without recombination or gene conversion one need only specify the sample size, the number of replicate samples to produce and the mutation parameter. Thus to generate samples in this simplest case one types 'ms nsam nreps $-t 4N_0u$', where nsam is the number of chromosomes in each sample, and nreps is the number of replicate samples to generate. For example, to generate 100 samples of 5 chromosomes, with $4N_0u = 3.0$, one types: 'ms 5 100 $-t$ 3.0 >outfile'.

To incorporate other complications, such as recombination, gene conversion, migration, or changing population size, additional command line arguments must be specified. Documentation is available at the web site indicated above.

OUTPUT

The output consists of one line showing the command line, then one line with the random number generator's

seed value. Following these two lines, the samples are printed out. Each sample begins with a line with '//' on it. Following that is a line with the number of polymorphic sites. The next line has a list of the positions of the polymorphic sites on a scale of (0, 1). The following lines contain the gametes, one line per gamete. The gametes are a string of 0s and 1s indicating the ancestral and derived alleles at each polymorphic site. For example, the command line, 'ms 4 3 $-t$ 5.0 >outfile', would produce the following output:

```
ms 4 3 -t 5.0
1779988551

//
segsites: 5
positions: 0.0227 0.5520 0.6190 0.9200 0.9459
10001
00010
00000
01100

//
segsites: 4
positions: 0.6760 0.7866 0.9056 0.9606
0101
1000
0101
0110

//
segsites: 4
positions: 0.1259 0.4479 0.4520 0.6670
0001
1110
0000
0000
```

OPERATING SYSTEMS

The program is written in C and has been compiled on a variety of operating systems, Linux, SunOs and MacOsX. Command line arguments are an integral part of the program and must be accommodated by the compiler. (For some Mac and Windows compilers this can be a problem.) Code is provided to generate random numbers with drand48() or rand(). Random number generation and the seeding operations are isolated in the file rand1.c, and can easily be modified to accommodate other generators such as those provided by Press *et al.* (1992).

REFERENCES

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

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. In Futuyma,D. and Antonovics,J. (eds), *Oxford Surveys in Evolutionary Biology*, Vol. 7, pp. 1–44.

Press,W.H., Teukolsky,S.A., Vetterling,W.T. and Flannery,B.P. (1992) *Numerical Recipes in C*. Cambridge University Press, Cambridge.