BIOINFORMATICS APPLICATIONS NOTE
Vol. 22 no. 20 2006, pages 2556–2557
doi:10.1093/bioinformatics/btl415

Genetics and population analysis

Nemo: an evolutionary and population genetics programming framework

Frédéric Guillaume[1,2,*] and Jacques Rougemont[3]

[1]Department of Ecology and Evolution, University of Lausanne, Biofore, CH-1015 Lausanne, Switzerland, [2]Department of Zoology, University of British Columbia, 6270 University boullvard, Vancouver, British Columbia, Canada V6T 1Z4 and [3]Vital-IT, Swiss Institute of Bioinformatics, Quartier Sorge-Genopode, CH-1015 Lausanne, Switzerland

Received on May 5, 2006; revised on July 24, 2006; accepted on July 26, 2006
Advance Access publication July 31, 2006
Associate Editor: Martin Bishop

ABSTRACT

Summary: Nemo is an individual-based, genetically explicit and stochastic population computer program for the simulation of population genetics and life-history trait evolution in a metapopulation context. It comes as both a C++ programming framework and an executable program file. Its object-oriented programming design gives it the flexibility and extensibility needed to implement a large variety of forward-time evolutionary models. It provides developers with abstract models allowing them to implement their own life-history traits and life-cycle events. Nemo offers a large panel of population models, from the Island model to lattice models with demographic or environmental stochasticity and a variety of already implemented traits (deleterious mutations, neutral markers and more), life-cycle events (mating, dispersal, aging, selection, etc.) and output operators for saving data and statistics. It runs on all major computer platforms including parallel computing environments.

Availability: The source code, binaries and documentation are available under the GNU General Public License at http://nemo2.sourceforge.net.

Contact: guillaum@zoology.ubc.ca

Individual-based, genetically explicit and stochastic computer simulation approaches are increasingly applied to a large variety of questions in evolutionary biology, conservation biology and human genetics. However, the availability of a flexible and efficient software product was lacking. Indeed, most studies in these domains rely on homemade code rarely published, forcing newcomers to build their models from scratch and reinvent procedures often implemented several times elsewhere. Although it is clear that one unique model will not encompass all the aspects needed for modeling complex evolutionary scenarios or genetic architectures, a set of well-designed programing tools could greatly benefit the community of researchers in these domains. Nemo is an attempt to provide such a tool, focusing on flexibility and efficiency. Besides that, Nemo is also a high-level, end-user oriented simulation software that implements several evolutionary and genetic models as described in this note. Extensive documentation about the use of the software as well as low-level coding guidelines can be found on the dedicated website.

*To whom correspondence should be addressed.

Nemo is what is sometimes called a forward-time simulation model as opposed to coalescent or backward-time models (e.g. SIMCOLA2; Laval and Excoffier, 2004). A few forward-time individual-based simulation programs have been published recently; Easypop (Balloux, 2001), FPG (Hey, 2004, http://lifesci.rutgers.edu/~heylab/HeylabSoftware.htm#FPG) and simuPOP (Peng and Kimmel, 2005). Only one of these, simuPOP, can handle dynamical demographics and traits under selection while the others are designed as programs to generate genetic datasets in populations of constant size. In contrast, Nemo offers the possibility of both developing and executing simulation models of the evolution of life-history traits under a wide variety of population models. It is thus available as both an executable file and an extensible library. In Nemo, the addition of a new feature within the proposed framework does not require any modification of the released code. The linking against a dynamical library is sufficient for most cases.

Nemo implements four different types of simulation components that can be combined to build a model. These are the traits the individuals can bear, the life-cycle events the individuals will go through during their lifetime, the population models that specify how individuals are spatially organized and the output operators which specify how and what statistics will be saved. Below is a listing of implementations of those simulation components.

Traits: The currently available traits are dispersal (sex-specific), deleterious mutations, neutral markers (microsatellites) and *Wolbachia*, an endosymbiotic parasite causing cytoplasmic incompatibility. Traits parameters allow to select different mutation rates and models (e.g. SSM or KAM models for microsatellites), and recombination rates, when applicable.

Life cycle events: The main life-cycle events (LCE) are breeding (with random mating, polygyny, monogamy and selfing/hermaphroditism mating systems), dispersal (with migrant-pool and propagule-pool Island model, Stepping Stone model and 2D lattice models), population extinction, population density regulation and aging (without overlapping generations). An LCE may be linked with an existing trait to perform selection on that trait. Such linkages exist for breeding and deleterious mutations (i.e. viability selection) and for the dispersal LCE and traits, for instance. More LCEs are described in the user manual.

Population: Nemo offers a very flexible population model in which population size matrices can be given as input parameters thus allowing for a wide variety of population configurations. This is

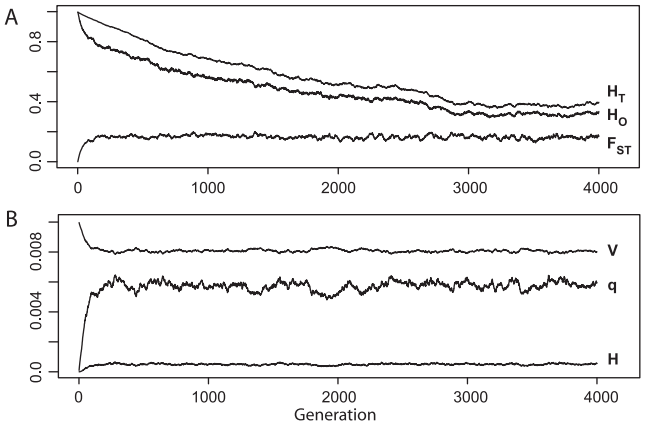

Fig. 1. Example of the evolution of neutral markers (100 loci) (A) and deleterious mutations (10 000 loci) (B) in one simulation over 4000 generations with 10 subpopulations of size 100. H_T, expected heterozygosity; H_O, observed heterozygosity; F_{ST}, index of population genetic structure; V, mean viability ($\times 10^{-2}$); q, mutations frequency; H, mutations homozygosity. This simulation took 14 min on a 3.06 Ghz Intel computer (without statistics recording).

further extended by the dispersal matrix parameter of the dispersal LCE that allows to specify each between-populations dispersal probabilities. Furthermore, owing to population size variability, the random fluctuation of population sizes may generate various degrees of demographic stochasticity in the system.

Outputs: The previous LCEs and traits define various statistics and output files that are computed and saved throughout the simulations preformed. Examples of these outputs are *F*-statistics, heterozygosities, coancestry coefficients and FSTAT files outputs (Goudet, 1995) for the neutral markers, measures of genetic load, heterosis, mutation frequencies, fitness per kinship classes and so on for the deleterious mutations. A complete listing of the statistics and output files is given in the user manual of Nemo.

MPI version: The MPI version of Nemo uses a master–slaves design where a master node distributes the replicates to be performed on the slave nodes and collects their results once carried out. The performance gain is linear in the number of slave nodes, since the amount of messaging involved is minimal.

Nemo's user interface is based on text input files specifying the simulation parameters, one per line in arbitrary order. Batch execution of several simulations with a single input file is straightforward (for details see the user manual). The number of populations/individuals/genes modeled is only limited by hardware capacity (e.g. see Figure 1).

Extending Nemo: The Nemo coding framework offers abstractions of the various simulation components needed to build an individual-based population model. Three main abstract classes may be derived. These are the `Trait`, `LifeCycleEvent` and `Handler` interfaces. The `Trait` class encapsulates the way an individual trait is handled. It declares the interface to the inheritance (including recombination), mutation and genotype-to-phenotype mapping methods. However, the framework does not yet provide any pre-implemented genetic architecture letting the implementer code its traits on the most suitable structure type, such as discrete (coded on one-byte variables, see our neutral markers) or continuous (coded on floating point variables as for quantitative traits like dispersal) allele models. In contrast, simuPOP provides only discrete allele models but allows for the precise chromosome mapping of the genes of interest. Nemo has been designed to be closer to the ecology of wild species, whereas simuPOP comes from the field of human epidemiology.

The `LifeCycleEvent` interface declares one main method, `execute()`, that is called at each time-step of the generation cycle. The life-cycle events can thus be viewed as operators that modify the population state through time. As the effect of a particular event is likely to depend on an individual's trait value, a linking interface between the `Trait` and `LifeCycleEvent` classes is also provided.

Finally, the `Handler` interface and its derived classes, `StatHandler` and `FileHandler`, are designed to implement the simulation data processors and collectors. Any kind of simulation component can use one or several of these handlers to compute statistics on a trait or record the population state and dump the results to various types of data files.

The features included in the present release of Nemo have been used in studies of the joint evolution of dispersal and inbreeding load (Guillaume and Perrin, 2006), of the spread of incompatibility-inducing parasites (M. Reuter, L. Lehmann and F. Guillaume, submitted for publication) and in studies of the effects of population extinction and recolonization on dispersal and the level of population genetic variation (Guillaume, 2005). Nemo's object-oriented design facilitated the implementation of such complex models. It has been developed so that future models could easily implement new life-history traits or life-cycle events without the burden of developing the simulation or population management aspects of such projects. Development guidelines can be found in the online documentation.

The flexibility in the number of traits and their genetic architecture (even non-genetic traits like infection status can be implemented) and in the ordering and number of life-cycle events building a life cycle makes Nemo a very powerful research tool. We hope extensions from contributed code will further improve the framework and make Nemo a platform of choice for a wide variety of researchers.

ACKNOWLEDGEMENTS

We thank Nicolas Perrin and Jérôme Goudet for their support all along this project. V. Reviol provided constant and enthusiastic support for the programing work. We also thank two anonymous reviewers for their constructive comments. Dr F. Balloux and his Easypop inspired a first version of Nemo. F.G. acknowledges a Swiss National Science Foundation (SNSF) grant number PBLAA-109652. This work has been supported by SNSF grants 31-059442 and 31-1000348 to Nicolas Perrin and 31-108194 to Jérôme Goudet.

Conflict of Interest: none declared.

REFERENCES

Balloux,F. (2001) Easypop, computer program for the simulation of population genetics. *J. Heredity*, **92**, 301–302.

Goudet,J. (1995) FSTAT (Version 1.2): a computer program to calculate *F*-statistics. *J. Heredity*, **86**, 485–486.

Guillaume,F. (2005) Multiple causes of the evolution of dispersal in metapopulations. PhD thesis, University of Lausanne, Switzerland.

Guillaume,F. and Perrin,N. (2006) Joint evolution of dispersal and inbreeding load. *Genetics*, **173**, 497–509.

Hey,J. (2004) A computer program for forward population genetic simulation.

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.

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