Systems biology

Möbius: an integrated discrete-event modeling environment

Jean Peccoud¹,*, Tod Courtney²,* and William H. Sanders²

¹Virginia Bioinformatics Institute, Washington Street, MC0477, Blacksburg, VA 24061 and ²Coordinated Science Laboratory and Information Trust Institute, University of Illinois at Urbana-Champaign, 1308 W. Main Street, Urbana, IL 61801, USA

Received and revised on August 30, 2007; accepted on October 7, 2007

ABSTRACT

Summary: Möbius has found numerous applications in computational biology to build and solve stochastic models of biological processes. It provides the user with a modeling workflow and several sophisticated features that are not available in the simulation tools commonly used by computational biologists.

Availability: Möbius is free for academic users. It can be downloaded from www.mobius.uiuc.edu.

Contact: peccoud@vt.edu or tod@crihc.uiuc.edu

1 INTRODUCTION

Discrete-event models represent the operation of a system as a chronological discrete sequence of events. Each event occurs at an instant in time and marks a change of state in the system (Banks, 2005). Discrete-event models have been used in computer science to analyze the performance, reliability and availability of computer and network architectures. The modeling of biological systems has traditionally used formalisms representing the time-evolution of the systems as a continuous process represented by ordinary differential equations or reaction-diffusion models. Ten years ago, we demonstrated that a special category of discrete-event models called stochastic Petri nets could be used to analyze the stochastic dynamics of small populations of interacting molecules (Goss and Peccoud, 1998). Since then, the fluctuations of the dynamics of the molecular mechanisms of gene expression mechanisms, also known as molecular noise, have been characterized experimentally (Raser and O'Shea, 2005). As a result, stochastic models of gene networks have become mainstream in systems biology. Boolean models of gene networks are another type of discrete-event models commonly used in biology (Thomas and D’Ari, 1990). As a result, recent versions of modeling packages include discrete-event simulation engines based on the Gillespie algorithms and some of their approximations (Gillespie, 2006). Tools that contain this capability include, Copasi (Hoops et al., 2006) and the MATLAB SimBiology toolbox. The purpose of this note is to summarize the most original features of Möbius (Daly et al., 2000; Deavours et al., 2002; Derisavi et al., 2002).

2 MAIN FEATURES OF MOBIUS

2.1 Multiple modeling languages

Möbius supports multiple modeling languages, based on either graphical or textual representations. Supported model types include, among others, extensions to stochastic Petri nets called Stochastic Activity Networks (SANs), Buckets and Balls (Markov chains and extensions) and stochastic process algebras. The multiple modeling formalisms make it possible to construct models with the right level of detail and customized to the specific behavior of the system of interest. Models used in computational biology can be represented by SANs, stochastic process algebras, or, in some cases, by the less expressive Bucket and Balls formalism. SANs and Buckets and Balls models are graphically represented by a directed graph in which the Place nodes represent the quantities of substances of the system and the activities represent the reactions between these substances. Activities can be instantaneous or timed, with delays that are deterministic or stochastic, the latter defined by one of the many standard probability distribution functions. Through the use of various distribution functions, activities can represent a wide variety of biological phenomena, such as the binomial distributions of molecules during cell division, for instance (Goss and Peccoud, 1998). In addition, SANs have input and output gates that can be used to define system-specific functions that determine the effect the activity has on the system state.

2.2 Hierarchical modeling paradigm

Complex models are built from the ground up. Models are created to specify the behavior of individual components, and then those models are combined using the Join and Replicate operators to create a model of the complete system. For example, a model of the bistable toggle switch (Gardner et al., 2000) could be broken down into two models, one for each of the mechanisms controlling the expression of each of the two genes coding for the proteins U and V. The Join operator would then be used to connect the two models by sharing the nodes corresponding to proteins. The Join operator can also be used to represent compartments commonly found in physiological models. The Replicate operator can be used to create many copies of the same model, e.g. to build a model that analyzes the properties of a population of identical cells. This bottom-up approach to modeling makes it easy to combine components in multiple ways to examine alternative model hypotheses.
2.3 Customized measures of system properties
Detailed analysis of complex systems with large state spaces, such as those found in systems and synthetic biology, requires more than the visualization of a few simulation trajectories. Custom measures can be defined in Möbius, in the form of detailed expressions based on the nodes and activities in the model, to measure the exact information desired about the system. In the case of the bistable toggle switch, measures could be defined to characterize the ON and OFF states of the system from the state variables. Measures can be defined on activities to collect statistics on noise-induced transitions using simulation. More sophisticated measures expressing the ability of a gene network to adapt to a changing environment have been described elsewhere (Peccoud et al., 2004).

Möbius allows the user to specify what statistics of the measures should be estimated by simulation: mean, variance and/or a full distribution function. The choice makes it possible for the user to produce analysis results that are statistically and/or a full distribution function. The choice makes it possible for the user to produce analysis results that are statistically sufficient for each measure. Users specify the desired confidence level of the estimates of these statistics. The simulator will compute trajectories through the system until the specified measures reach the desired level of confidence, or the predetermined number of trajectories has been computed.

2.4 Parameterized models
Models can be created in a parameterized form, with properties of the system, such as reaction rates or initial concentrations of substances, represented by model parameters. Multiple experiments can be created, where each experiment contains a unique set of values for the parameters. Analysis can then be performed on each of the experiments. Model parameters can be assigned fixed values or a range of values to scan regions of the parameter space. This feature is convenient for performing parameter sensitivity analysis or comparing the phenotypes of different mutants.

2.5 Model analysis
The user can choose among several solvers to analyze model properties. Möbius’s simulators (both transient and steady state) repeatedly generate trajectories of the model from which measure statistics can be estimated. Alternatively, users can analyze certain classes of models using one of the analytic/numerical state-space-based solvers. Analytic solvers provide exact solutions for the specified transient and steady-state measures, but the complexity of the algorithms limits their use to certain classes and sizes of models. We continue to develop new symbolic state-space data structures and algorithms that make analytic solutions possible for models with tens of millions of states (Derisavi et al., 2004). Further discussion of the relative merits of simulation versus analytical solution is beyond the scope of this note and will be addressed in a follow-up article.

2.6 Distributed model evaluation
Möbius enables users to run the simulation or analytic/numerical solution of a model on multiple CPUs and multiple machines. Editors allow users to define the machines on which Möbius is installed and to assign them to specific analysis jobs. Statistical results are automatically collected and processed by the instance of Möbius that initiated the evaluation. Instances of Möbius running on different operating systems can work together on the same analysis. Möbius’s ability to distribute evaluation across a number of machines is self-contained, does not require any middleware or queuing system, and can be set up by any user with no specific system administration skills.

3 DISCUSSION
Möbius has its roots in the computer science community, but now has a strong and growing user base in computational biology. While the existing formalisms are general-purpose and not specific to biological applications, the Möbius API makes it possible to add new modeling formalisms to the tool, and there is currently a chemical reaction formalism under development. We have plans to include in Möbius a hybrid simulation algorithm to speed up the simulation of stiff discrete-event models (Griffith et al., 2006), but the code has not yet been integrated and is currently available as a separate tool.

Möbius users are provided with a rigorous modeling workflow enforced by the design of the software application. When learning how to use the software, even people with years of modeling experience can gain new insights about modeling. Many of the concepts with which Möbius has been designed can be transferred to less structured modeling environments. Möbius is unquestionably a powerful research tool, and is also an extremely valuable teaching tool that will give students at all levels access to sophisticated concepts developed in the computer science and modeling communities. Understanding of the underlying concepts will greatly increase their ability to analyze the properties of the increasingly complex models developed in the life sciences, irrespective of the tools they will use.

ACKNOWLEDGEMENTS
The authors would like to acknowledge the contributions to the Möbius tool by all the present and past members of the Möbius group and others from groups around the world. The authors would also like to thank Jenny Applequist for her editorial assistance.

Conflict of Interest: none declared.

REFERENCES


