Bioinformatics, 32(21), 2016, 3366–3368
doi: 10.1093/bioinformatics/btw469
Advance Access Publication Date: 8 July 2016
Applications Note

OXFORD

Systems biology

BioNetGen 2.2: advances in rule-based modeling

Leonard A. Harris[†], Justin S. Hogg, José-Juan Tapia, John A. P. Sekar, Sanjana Gupta, Ilya Korsunsky[‡], Arshi Arora[§], Dipak Barua[¶], Robert P. Sheehan, and James R. Faeder*

Department of Computational and Systems Biology, University of Pittsburgh School of Medicine, Pittsburgh, PA, USA

*To whom correspondence should be addressed.
[†]Present address: Department of Cancer Biology, Vanderbilt University School of Medicine, Nashville, TN, USA
[‡]Present address: Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, New York, NY and The Feinstein Institute for Medical Research, Manhasset, NY, USA
[§]Present address: Department of Epidemiology and Biostatistics, Memorial Sloan-Kettering Cancer Center, New York, NY, USA
[¶]Present address: Department of Chemical and Biochemical Engineering, Missouri University of Science and Technology, Rolla, MO, USA
Associate Editor: Jonathan Wren

Received on July 1, 2015; revised on June 5, 2016; accepted on June 27, 2016

Abstract

Summary: BioNetGen is an open-source software package for rule-based modeling of complex biochemical systems. Version 2.2 of the software introduces numerous new features for both model specification and simulation. Here, we report on these additions, discussing how they facilitate the construction, simulation and analysis of larger and more complex models than previously possible.
Availability and Implementation: Stable BioNetGen releases (Linux, Mac OS/X and Windows), with documentation, are available at http://bionetgen.org. Source code is available at http://github.com/RuleWorld/bionetgen.
Contact: bionetgen.help@gmail.com
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Rule-based modeling is an approach for addressing combinatorial complexity in models of biochemical systems (Chylek *et al.*, 2014, 2015). Instead of manually enumerating all possible species and reactions that can exist within a system, a rule-based model defines only the reactive motifs within macromolecular complexes and the interactions and modifications that involve those motifs. BioNetGen is an open-source software package for constructing and simulating rule-based models (Blinov *et al.*, 2004; Faeder *et al.*, 2009). It has been used to model a variety of biological processes, including cell signaling, gene regulation and metabolism (Chylek *et al.*, 2014, 2015, and references therein; see also bionetgen.org/index.php/Model_Examples). Models are written in a human-readable, text-based modeling language known as BNGL (BioNetGen language). Numerous user-specified actions can be added to a BNGL model file, including generating a reaction network and performing deterministic or stochastic simulations. Models can also be exported to different formats, such as SBML (Hucka *et al.*, 2003) and MATLAB language (The MathWorks Inc., Natick, MA). Furthermore, BioNetGen interfaces with NFsim (Sneddon *et al.*, 2011), a 'network-free' simulator that avoids enumeration of species and reactions, which may be intractable for large models. RuleBender is a

visual interface for BioNetGen, with features that include syntax checking/highlighting and visualizations for model debugging and comparison (Wenskovitch *et al.*, 2014; Xu *et al.*, 2011). BioNetGen is also used as a network generator and simulator in a number of third-party tools including the Virtual Cell (Moraru *et al.*, 2008), BioUML (Kolpakov *et al.*, 2006), SRSim (Grünert and Dittrich, 2011), Parts & Pools (Marchisio, 2014), pySB (Lopez *et al.*, 2013) and BioNetFit (Thomas *et al.*, 2016).

Taken together, the BioNetGen/NFsim/RuleBender suite of tools (Fig. 1) provides powerful capabilities for rule-based modeling and simulation. Until recently, however, there were several significant shortcomings: (i) BNGL models were limited to mass-action kinetics; (ii) network-based stochastic simulations were limited to computationally expensive 'exact' methods; (iii) network-free simulations were limited to fewer than a few million particles due to high memory load; (iv) there was no facility for importing models from SBML format. Each of these shortcomings has been addressed in the recent BioNetGen 2.2.x series of releases, as described below. Additional information can be found in extensive online documentation at http://bionetgen.org/index.php/Documentation (see Supplementary Information for specific links).

2 Additions

2.1 Functional rate laws

BioNetGen 2.2 introduces the ability to define arbitrary mathematical functions that can be used as rate laws. Functional rate laws are integrated into the BNGL grammar and are supported by both NFsim and the network-based simulators (deterministic and stochastic) included in the BioNetGen package. Functions are defined using 'observables,' which compute the concentrations of species with specified properties (Faeder *et al.*, 2009), and can either be evaluated *globally*, i.e. over the entire system, or *locally*, over a specified molecule or molecular complex. Local functions greatly expand the expressiveness of BNGL because they enable a single rule to specify many reactions with rates that depend on the specific properties of the reacting species (Sneddon *et al.*, 2011).

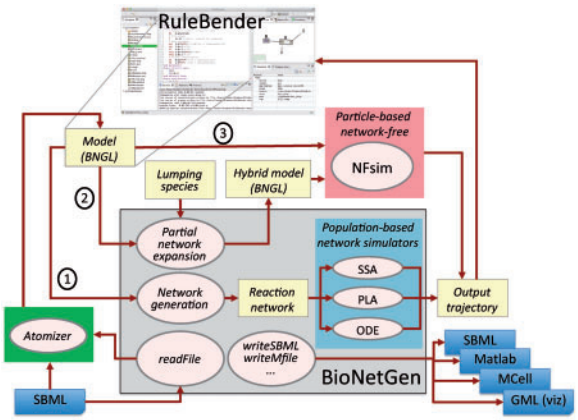

Fig. 1. Basic overview of the RuleBender/BioNetGen/NFsim software stack. BNGL models may be constructed and edited in RuleBender as well as translated from an SBML model using Atomizer (either with the BioNetGen `readFile` action or the web application at ratomizer.appspot.com). Models can then be simulated in three ways: (1) with a built-in population-based network simulator (ODE, SSA, PLA) after explicit network generation; (2) as a hybrid particle/population (HPP) model in NFsim after partial network expansion ('lumping' species must be provided (Hogg *et al.*, 2014)); (3) with NFsim directly. In all cases, output trajectories are passed back to RuleBender for display. BioNetGen also has a number of built-in methods for exporting to third-party formats, such as SBML and Matlab. (ODE: ordinary differential equations; SSA: stochastic simulation algorithm; PLA: partitioned-leaping algorithm; GML: Graph Modeling Language.)

2.2 Accelerated-stochastic simulation

τ leaping is an approach for accelerating stochastic simulations of biochemical networks (Gillespie, 2007). Numerous variants of τ leaping have been proposed in the literature, including a multiscale variant known as the partitioned-leaping algorithm (PLA) (Harris and Clancy, 2006). In their simplest realizations, τ -leaping methods are analagous to the explicit forward Euler method for solving ordinary differential equations (ODEs). Therefore, as with ODE integrators, higher-order and implicit versions of τ leaping algorithms are possible. BioNetGen 2.2 includes an explicit Runge-Kutta implementation of the PLA (RK-PLA) that can be used on any model for which a reaction network can be generated. Additional information, including a performance analysis, is provided in the Supplementary Information.

2.3 Hybrid particle/population simulation

Network-free simulation methods that do not enumerate species and reactions are often required to simulate complex models (Chylek *et al.*, 2014, 2015). However, they are limited by the fact that memory usage increases linearly with the number of particles. Hogg *et al.* (2014) introduced a hybrid simulation method that treats a user-defined set of species as population variables rather than as particles. This hybrid particle/population (HPP) approach avoids the memory costs associated with having large pools of identical particles and was shown to significantly reduce computational memory expense with no effect on simulation accuracy and little effect on run time (Hogg *et al.*, 2014). HPP is implemented within BioNetGen 2.2 and can be run using NFsim version 1.11 or later.

2.4 SBML-to-BNGL translation

SBML is a widely-used model exchange format in systems biology (Hucka *et al.*, 2003). Models encoded in SBML are *flat*, i.e. their species do not have internal structure, which limits their utility for rule-based modeling. BioNetGen 2.2 includes an SBML-to-BNGL translator, called Atomizer (also available as a web tool at ratomizer.appspot.com), that can extract implicit molecular structure from flat species (Tapia and Faeder, 2013). A full report on Atomizer and its application to the BioModels database (Li *et al.*, 2010) is currently in preparation. However, Tapia and Faeder (2013) reported that an early version of the tool could recover implicit structure for about 60% of species in models within the database that contain ≥ 20 species. Thus, Atomizer makes available a large set of pre-existing models in a rule-based format, facilitating their visualization (Wenskovitch *et al.*, 2014) and extension (Chylek *et al.*, 2015).

2.5 Additional features

BioNetGen 2.2 also introduces a number of additional features to those described above, including a null symbol ('0') that can act as a source or a sink in rules and new actions for performing parameter scans, generating MATLAB Executable (MEX) files, exporting to formats readable by third-party simulators (e.g. MCell (Kerr *et al.*, 2008)), and outputting graphical visualizations at different scales (Sekar *et al.*, 2015). We have also developed parameter estimation tool called `ptempest` (available at http://github.com/RuleWorld/ptempest) that combines Bayesian Monte Carlo methods(Klinke, 2014; Eydgahi *et al.*, 2013) with parallel tempering (Earl and Deem,

2005) to accelerate sampling. Many new simulation options have also been added, e.g. to allow simulation arguments to be loaded from file and to terminate a simulation if a user-defined logical condition is met. A comprehensive listing of all new actions and arguments added in BioNetGen 2.2 is provided in the Supplementary Information (see also http://bionetgen.org/index.php/BNG_Actions_Args).

3 Conclusion

BioNetGen is an active, open-source project that highly encourages contributions from interested members of the modeling community. Contact information, as well as numerous links to online documentation, is provided in the Supplementary Information. Ongoing development efforts include interfacing with spatial simulators (Sullivan *et al.*, 2015), free-energy-based modeling (Hogg, 2013), and improved support for community standards (e.g. SBML-multi, SED-ML). Recent independent efforts that leverage the BioNetGen framework include a standard for annotation of rule-based models (Misirli *et al.*, 2016) specified in BNGL or Kappa (Danos *et al.*, 2007), another widely-used rule-based modeling language and BioNetFit (Thomas *et al.*, 2016), a parameter estimation tool for models simulated using either BioNetGen or NFsim.

Acknowledgements

We thank Robert Clark, Thierry Emonet, Robert Engelke, William Hlavacek, Cihan Kaya, G. Elisabeta Marai, Nikketh Nair, Daniel Packer, James Pino, Adam Smith, Michael Sneddon, Lori Stover, Joseph Vigil & John Wenskovitch for technical help and useful discussions.

Funding

NIH grants P41 GM103712, R01 AI107825, R01 GM115805 and P01 HL114453 and NSF Expeditions in Computing Grant (award 0926181). JSH and RPS received support through T32 EB009403.

Conflict of Interest: none declared.

References

Blinov,M.L. *et al.* (2004) BioNetGen: software for rule-based modeling of signal transduction based on the interactions of molecular domains. *Bioinformatics*, **17**, 3289–3291.

Chylek,L.A. *et al.* (2014) Rule-based modeling: a computational approach for studying biomolecular site dynamics in cell signaling systems. *Wiley Interdiscip. Rev. Syst. Biol. Med.*, **6**, 13–36.

Chylek,L.A. *et al.* (2015) Modeling for (physical) biologists: an introduction to the rule-based approach. *Phys. Biol.*, **12**, 045007.

Danos,V. *et al.* (2007) Rule-based modelling of cellular signalling. *Lect. Notes Comput. Sci.*, **4703**, 17–41.

Earl,D.J. and Deem,M.W. (2005) Parallel tempering: theory, applications, and new perspectives. *Phys. Chem. Chem. Phys.*, **7**, 3910–3916.

Eydgahi,H. *et al.* (2013) Properties of cell death models calibrated and compared using Bayesian approaches. *Mol. Syst. Biol.*, **9**, 644.

Faeder,J.R. *et al.* (2009) Rule-based modeling of biochemical systems with BioNetGen. In: *Methods in Molecular Biology, vol. 500*. Humana Press, Clifton, NJ, pp. 113–167.

Gillespie,D.T. (2007) Stochastic simulation of chemical kinetics. *Annu. Rev. Phys. Chem.*, **58**, 35–55.

Grünert,G. and Dittrich,P. (2011) Using the SRSim software for spatial and rule-based modeling of combinatorially complex biochemical reaction systems. *Lect. Notes Comput. Sci.*, **6501**, 240–256.

Harris,L.A. and Clancy,P. (2006) A "partitioned leaping" approach for multiscale modeling of chemical reaction dynamics. *J. Chem. Phys.*, **125**, 144107.

Hogg,J.S. (2013). Advances in Rule-based Modeling: Compartments, Energy, and Hybrid Simulation, with Application to Sepsis and Cell Signaling. PhD thesis, University of Pittsburgh.

Hogg,J.S. *et al.* (2014) Exact hybrid particle/population simulation of rule-based models of biochemical systems. *PLoS Comput. Biol.*, **10**, e1003544.

Hucka,M. *et al.* (2003) The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics*, **19**, 524–531.

Kerr,R. *et al.* (2008) Fast Monte Carlo simulation methods for biological reaction-diffusion systems in solution and on surfaces. *SIAM J. Sci. Comput.*, **30**, 3126–3149.

Klinke,D.J. (2014) In silico model-based inference: A contemporary approach for hypothesis testing in network biology. *Biotechnol. Prog.*, **30**, 1247–1261.

Kolpakov,F. *et al.* (2006) BioUML: visual modeling, automated code generation and simulation of biological systems. In: *Proceedings of the Fifth International Conference on Bioinformatics of Genome Regulation and Structure*, pp. 281–284.

Li,C. *et al.* (2010) BioModels database: an enhanced, curated and annotated resource for published quantitative kinetic models. *BMC Syst. Biol.*, **4**, 92.

Lopez,C.F. *et al.* (2013) Programming biological models in Python using PySB. *Mol. Syst. Biol.*, **9**, 646.

Marchisio,M.A. (2014) Parts & pools: a framework for modular design of synthetic gene circuits. *Front. Bioeng. Biotechnol.*, **2**, 42.

Misirli,G. *et al.* (2016) Annotation of rule-based models with formal semantics to enable creation, analysis, reuse and visualization. *Bioinformatics*, **32**, 908–917.

Moraru,I.I. *et al.* (2008) Virtual Cell modelling and simulation software environment. *IET Syst. Biol.*, **2**, 352–362.

Sekar,J.A.P. *et al.* (2015) Visualizing regulation in rule-based models. *ArXiv e-prints*. arXiv:1509.00896 [q-bio.QM].

Sneddon,M.W. *et al.* (2011) Efficient modeling, simulation and coarse-graining of biological complexity with NFsim. *Nat. Methods*, **8**, 177–183.

Sullivan,D.P. *et al.* (2015) Design automation for biological models: a pipeline that incorporates spatial and molecular complexity. In: *Proceedings of the 25th edition on Great Lakes Symposium on VLSI*, ACM, Pittsburgh, PA, pp. 321–323.

Tapia,J.J. and Faeder,J.R. (2013) The Atomizer: extracting implicit molecular structure from reaction network models. In: *Proceedings of the International Conference on Bioinformatics, Computational Biology and Biomedical Informatics (BCB'13)*, ACM, New York, NY, pp. 726–727.

Thomas,B.R. *et al.* (2016) BioNetFit: a fitting tool compatible with BioNetGen, NFsim and distributed computing environments. *Bioinformatics*, **32**, 798–800.

Wenskovitch,J.E. *et al.* (2014) MOSBIE: a tool for comparison and analysis of rule-based biochemical models. *BMC Bioinformatics*, **15**, 316.

Xu,W. *et al.* (2011) RuleBender: a visual interface for rule-based modeling. *Bioinformatics*, **27**, 1721–1722.