BIOINFORMATICS APPLICATIONS NOTE Vol. 25 no. 11 2009, pages 1457–1460
doi:10.1093/bioinformatics/btp173

Systems biology

GetBonNie for building, analyzing and sharing rule-based models

Bin Hu[1,2], G. Matthew Fricke[3], James R. Faeder[4], Richard G. Posner[5] and William S. Hlavacek[1,2,6,*]

[1]Department of Biology, University of New Mexico, Albuquerque, NM 87131, [2]Theoretical Biology and Biophysics Group, Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM 87545, [3]Department of Computer Science, University of New Mexico, Albuquerque, NM 87131, [4]Department of Computational Biology, University of Pittsburgh School of Medicine, Pittsburgh, PA 15260, [5]Computational Biology Division, Translational Genomics Research Institute, Phoenix, AZ 85004 and [6]Center for Nonlinear Studies, Los Alamos National Laboratory, Los Alamos, NM 87545, USA

Received on November 21, 2008; revised on March 6, 2009; accepted on March 22, 2009

Advance Access publication March 25, 2009

Associate Editor: Olga Troyanskaya

ABSTRACT

Summary: GetBonNie is a web-based application for building, analyzing and sharing rule-based models encoded in the BioNetGen language (BNGL). Tools accessible within the GetBonNie environment include (i) an applet for drawing graphs that correspond to BNGL code; (ii) a network-generation engine for translating a set of rules into a chemical reaction network; (iii) simulation engines that implement generate-first, on-the-fly and network-free methods for simulating rule-based models; and (iv) a database for sharing models, parameter values, annotations, simulation tasks and results.

Availability: GetBonNie is free at http://getbonnie.org.

Contact: getbonnie@cs.unm.edu

1 INTRODUCTION

The systems-level dynamics of molecular interactions in cellular regulatory systems are difficult to model using approaches that rely on explicit specification of a chemical reaction network. The reason is combinatorial complexity (Hlavacek *et al*., 2006), the potential of molecular interactions to generate large numbers of chemical species and reactions. Rule-based modeling approaches have been developed to address this problem (Hlavacek *et al*., 2006). In these approaches, molecules and molecular complexes are typically represented using graphs, or the equivalent, and molecular interactions are represented using (graph-rewriting) rules. A rule implicitly defines the reactions that can be generated by the molecular interaction that it represents.

A variety of software tools have been developed to facilitate rule-based modeling (Hlavacek *et al*., 2006). These tools, which provide complementary capabilities, are based on similar formal foundations. However, interoperability of these tools is limited because a standard format, such as the Systems Biology Markup Language (SBML) (Hucka *et al*., 2003), has yet to emerge for the exchange of rule-based models. Moreover, a public repository, such as the BioModels Database (Le Novère *et al*., 2006) in which models are stored in SBML, is unavailable for rule-based models. SBML is notable because it allows chemical reaction networks to be specified in a form that can be processed by (numerous) SBML-compliant software tools (http://sbml.org). Although a set of rules can be translated into a chemical reaction network, a rule set often implies a vast reaction network, and as a result, the current version of SBML, Level 2, is not suited for efficient encoding of rule-based models. This deficiency has been recognized and a Level 3 extension of SBML appropriate for representation of rule-based models is under discussion (http://sbml.org/Community/Wiki).

In part, to provide a means to exchange rule-based models in terms of rules (rather than in terms of the networks implied by rules) and, in addition, to provide additional computational infrastructure for rule-based modeling of biochemical systems, we developed GetBonNie. GetBonNie provides a web-based toolkit for building, viewing, simulating and sharing rule-based models encoded in the BioNetGen language (BNGL) (Faeder *et al*., 2009), which is closely related to κ-calculus (Danos *et al*., 2007a), another language for encoding rule-based models. In GetBonNie, models are stored in an XML-based format, which is defined by an XML schema documented at the GetBonNie web site. The name GetBonNie is an anagram of BioNetGen.

2 FEATURES

GetBonNie combines the capabilities of several stand-alone programs in one freely accessible web-based application. Screenshots are shown in Figure 1. GetBonNie allows users to create rule-based models and share them within a private group or with the public. Public models are each assigned a unique accession number for reference purposes. Model specifications can be either uploaded to or built *de novo* within the GetBonNie environment, which provides interactive forms to guide model building and simulations (Fig. 1A). Model specifications are automatically checked to ensure that they are well formed. Online help is available.

2.1 Model building and sharing

GetBonNie represents and displays the elements of a model specification as a tree, which can be navigated by a user. A model is built by using web forms to add/delete and edit BNGL code (and associated annotations) at each leaf of a model tree. BNGL

*To whom correspondence should be addressed.

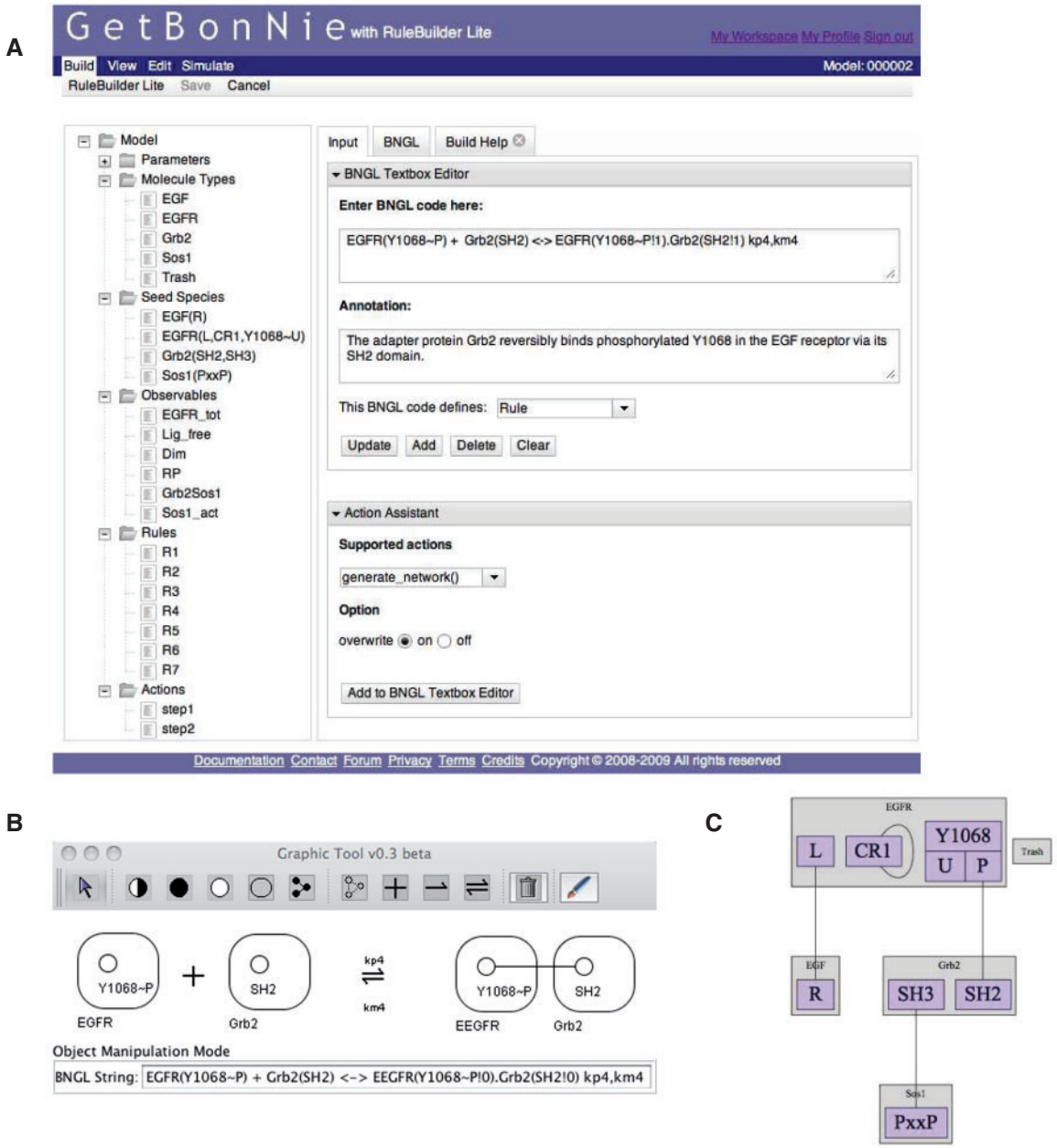

Fig. 1. Screenshots. (**A**) The Build tool illustrated here provides a form-based interface for specifying a rule-based model using BNGL. The tree view at left allows a model specification to be easily navigated. In this screenshot, the BNGL Textbox Editor displays one of the seven rules that comprise the model specification outlined in the tree view. This specification is available as a BioNetGen input file (simple_egfr.bngl) at the BioNetGen web site (http://bionetgen.org), and it is discussed at length in the tutorial of Faeder *et al*. (2009). This model is also available in the public library of models of GetBonNie (Model #000002). The Action Assistant allows a user to add actions to a model, such as simulation directives. A user can view a model specification in BNGL format by clicking on the BNGL tab at top center. By clicking on the Build Help tab, a user can view online help. Help tabs are available throughout GetBonNie. (**B**) The RuleBuilder Lite tool illustrated here can be used to visualize BNGL code entered in the box at bottom. It can also be used to generate BNGL code using the drawing tools at top. In the center of this screenshot, the BNGL code that appears in the BNGL Textbox Editor of the Build tool window shown in panel (**A**) is displayed according to the graphical conventions of Faeder (2005b). As can be seen, RuleBuilder Lite has a toolbar, a drawing area, a current status indicator (located above the box used to enter BNGL code) and an input box for BNGL code. (**C**) A contact map (Danos *et al*., 2007a) produced by the Viewer tool is shown here. This contact map illustrates the molecules, molecular components and molecular interactions defined in Model #000002. The contact map is generated by analyzing a model specification. A gray box represents a molecule, whereas a blue box inside a gray box represents a component of a molecule. The possible internal states of a component are also drawn in blue boxes that appear below component boxes. Lines connect binding partners. The contact map shown here indicates that a tyrosine of EGFR (Y1068) has two states: unphosphorylated (U) and phosphorylated (P). Note that, according to this contact map, the SH2 domain of Grb2 binds tyrosine 1068 of EGFR only if this tyrosine is phosphorylated.

code can be manipulated directly in text boxes of web forms, or RuleBuilder Lite, a Java applet, can be used to draw graph-based definitions of molecules and molecular interactions (Faeder *et al*., 2005b) and generate the corresponding BNGL code. This code can then be entered into a web form using copy–paste operations. RuleBuilder Lite also translates BNGL code into its graphical form, allowing model elements to be visualized (Fig. 1B). Models can be imported/exported as a BioNetGen input file, saved, checked for conformity with the conventions of BNGL and shared with selected users or the public. Public models can be copied to the private workspace of a user for reuse, revision and extension. GetBonNie is designed to encourage compliance with MIRIAM standards (Le Novère *et al*., 2005).

2.2 Model viewing

In addition to the visualization capabilities of RuleBuilder Lite, GetBonNie provides several ways for a user to inspect a model. For example, a contact map (Danos *et al*., 2007a) can be generated to summarize the site-specific molecular interactions included in a model (Fig. 1C). Contact maps can be downloaded in SVG or GraphViz .dot formats. Tables are generated to summarize the contents of a model specification, the component substructure of a molecule, the potential states and binding partners of a molecule or molecular component and the site-specific details of a molecular interaction. BNGL code defining a molecule type is automatically visualized when the code is selected for viewing or editing.

2.3 Simulating models

Simulation settings are specified using a form-based interface. GetBonNie provides access to simulation engines that implement stochastic and deterministic generate-first methods (Blinov *et al*., 2004), stochastic on-the-fly methods (Faeder *et al*., 2005a; Lok and Brent, 2005) and stochastic network-free methods (Colvin *et al*., 2009; Danos *et al*., 2007b; Yang *et al*., 2008) for well-mixed reaction systems. When a simulation is finished, a user can plot the results online and/or download the raw simulation data. Simulation settings can be saved and attached to a model for sharing. In addition, a server-side network generation engine can be used to translate a set of rules into a chemical reaction network, and the resulting network can be exported in BioNetGen .net, SBML Level 2 and MATLAB® M-file formats.

3 TECHNICAL DETAILS

Core GetBonNie functions (e.g. HTTP requests, model persistence and access control and parsing) are provided by the Linux-Apache-MySQL-PHP (LAMP) solution stack. Object-oriented PHP scripts were developed using the Zend Framework (http://framework.zend.com). Model metadata (e.g. ownership and accession number) are stored in a MySQL database. Models and related data (e.g. simulation settings) are stored in the local file system of the server. Models are stored in both plain BNGL and XML-encoded BNGL, which is defined by an XML schema. The XML schema and supporting documentation can be downloaded from the GetBonNie site. The software tool BioNetGen (Blinov *et al*., 2004; Faeder *et al*., 2009) is used to convert plain BNGL (uploaded or entered in a web form) into XML-encoded BNGL. A PHP script parses XML-encoded BNGL to generate drawing scripts for visualization of model elements. Drawing scripts are interpreted by GraphViz (Gansner and North, 2000) to produce SVG images for online display and downloading.

Interactive GetBonNie functions, such as web forms used for model specification, are provided using the asynchronous JavaScript and XML (Ajax) approach. JavaScript code was developed using the Dojo Toolkit (http://dojotoolkit.org). For client–server interactions, a PHP script interconverts models and related data between plain BNGL and JavaScript Object Notation (JSON). Use of JSON speeds up page display and facilitates storage of partial model specifications. Design of the RuleBuilder Lite applet is based on the stand-alone version of this tool (http://bionetgen.org), which provides a graphical user interface for BioNetGen (Faeder *et al*., 2009). Simulation results are plotted online using the Google™ visualization API. A web forum for discussion of GetBonNie is provided using the phpBB™ forum solution (http://www.phpbb.com).

GetBonNie uses the software tools BioNetGen and DYNSTOC (Colvin *et al*., 2009) as engines for simulation, network generation and network export. BioNetGen provides generate-first and on-the-fly simulation functions and network-generation and export functions. DYNSTOC provides a network-free simulation function.

4 FUTURE DIRECTIONS

The formalism of rule-based modeling enables one to make precise statements about the molecules and site-specific details of the molecular interactions in a system, and the formal elements of a model specification can be associated with annotation for the corresponding molecules and interactions. In the future, we plan to extend GetBonNie to allow biological knowledge, such as protein sequences, to be linked to formal elements of models. This type of annotation capability, which was one of the motivating factors for developing GetBonNie as a web-based application, should enable users to make models more understandable and easier to refine and extend.

ACKNOWLEDGEMENTS

We are grateful to the participants of the Rule-based Modeling of Biochemical Systems workshop (Santa Fe, NM, June 13–15, 2007) for helpful discussions of exchange formats for rule-based models. We thank Joshua Colvin, Kathryn Geyer and Ambarish Nag for helpful discussions.

Funding: National Institutes of Health (grants GM076570 and CA109552); Department of Energy (contract DE-AC52-06NA25396); and Arizona Biomedical Research Commission.

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*, **20**, 3289–3291.

Colvin,J. *et al*. (2009) Simulation of large-scale rule-based models. *Bioinformatics*, **25**, 910–917.

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

Danos,V. *et al*. (2007b) Scalable simulation of cellular signaling networks. *Lect. Notes Comput. Sci.*, **4807**, 139–157.

Faeder,J.R. *et al*. (2005a) Rule-based modeling of biochemical networks. *Complexity*, **10**, 22–41.

Faeder,J.R. *et al*. (2005b) Graphical rule-based representation of signal-transduction networks. In Liebrock,L.M. (ed.) *Proceedings of the 2005 ACM Symposium on Applied Computing*. ACM Press, New York, NY, pp. 133–140.

Faeder,J.R. *et al*. (2009) Rule-based modeling of biochemical systems with BioNetGen. *Methods Mol. Biol.*, **500**, 113–167.

Gansner,E.R. and North,S.C. (2000) Open graph visualization system and its applications to software engineering. *Softw. - Pract. Exp.*, **30**, 1203–1233.

Hlavacek,W.S. *et al*. (2006) Rules for modeling signal-transduction systems. *Sci. STKE*, **2006**, re6.

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.

Le Novère,N. *et al*. (2005) Minimum information requested in the annotation of biochemical models (MIRIAM). *Nat. Biotechnol.*, **23**, 1509–1515.

Le Novère,N. *et al*. (2006) BioModels database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems. *Nucleic Acids Res.*, **34**, D689–D691.

Lok,L. and Brent,R. (2005) Automatic generation of cellular reaction networks with Moleculizer 1.0. *Nat. Biotechnol.*, **23**, 131–136.

Yang,J. *et al*. (2008) Kinetic Monte Carlo method for rule-based modeling of biochemical networks. *Phys. Rev. E* **78**, 031910.