Systems Biology

RULEBENDER: A Visual Interface for Rule-Based Modeling

Wen Xu[1], Adam M. Smith[1], James R. Faeder[2*], and G. Elisabeta Marai[1*]

[1]Department of Computer Science, University of Pittsburgh, Pittsburgh, PA 15260, USA and [2]Department of Computational and Systems Biology, University of Pittsburgh School of Medicine, Pittsburgh, PA 15260, USA

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Associate Editor: XXXXXXX

ABSTRACT

Summary: Rule-based modeling (RBM) is a powerful and increasingly popular approach to modeling intracellular biochemistry. Current interfaces for RBM are predominantly text-based and command-line driven. Better visual tools are needed to make RBM accessible to a broad range of users, to make specification of models less error prone, and to improve workflows. We present RULEBENDER, an open-source visual interface that facilitates interactive debugging, simulation, and analysis of RBMs.

Availability: RULEBENDER is freely available for Mac, Windows, and Linux at http://rulebender.org.

Contact: faeder@pitt.edu, marai@cs.pitt.edu

Supplementary information: http://rulebender.org

1 INTRODUCTION

Rule-based modeling (RBM) enables precise and concise specification of biochemical models by representing molecules as structured objects and molecular interactions as rules for transforming the attributes of these objects. The two most common languages for specifying RBMs, κ (Feret *et al.*, 2009) and the BIONETGEN language (Faeder *et al.*, 2009), have nearly identical syntax and semantics based on graphs and graph-rewriting. Each language and associated modeling framework has been used to construct a wide variety of models of biochemical processes. In this paper we focus on the BIONETGEN framework.

While the potential benefits of RBM to biology are outstanding, building an RBM and detecting and correcting modeling errors (i.e., debugging) can be tedious and frustrating. RBMs are typically defined by the user via a text file. The user defines a set of molecules and proceeds to write rules governing their interactions derived from specific knowledge about the system. Although *individual* rules are easy to write, it is often difficult to fully grasp the implications of a *set* of rules; the challenge in grasping the global perspective is particularly acute when trying to understand models written by different researchers. This problem complicates debugging and reduces the accessibility of RBM, especially for users with limited programming experience.

Simulating and analyzing RBMs pose additional challenges. Once a working model has been defined, an RBM can be simulated using a number of different approaches including ordinary differential equations, stochastic simulations, or particle-based stochastic simulations (Danos et al., 2007, Colvin et al., 2010, Sneddon et al., 2011). The typical workflow relies on an external plain text editor, command console, and external plotting tools for displaying simulation results, which is inconvenient because it requires modelers to switch between different tools over repeated cycles of model editing and simulation. The process gets further complicated when exploring alternative simulations and models.

Here we present RULEBENDER, an integrated development environment that aims to facilitate the construction, simulation, and analysis of RBMs through interactive editing, syntax checking, visualization, and integrated execution.

2 RULEBENDER

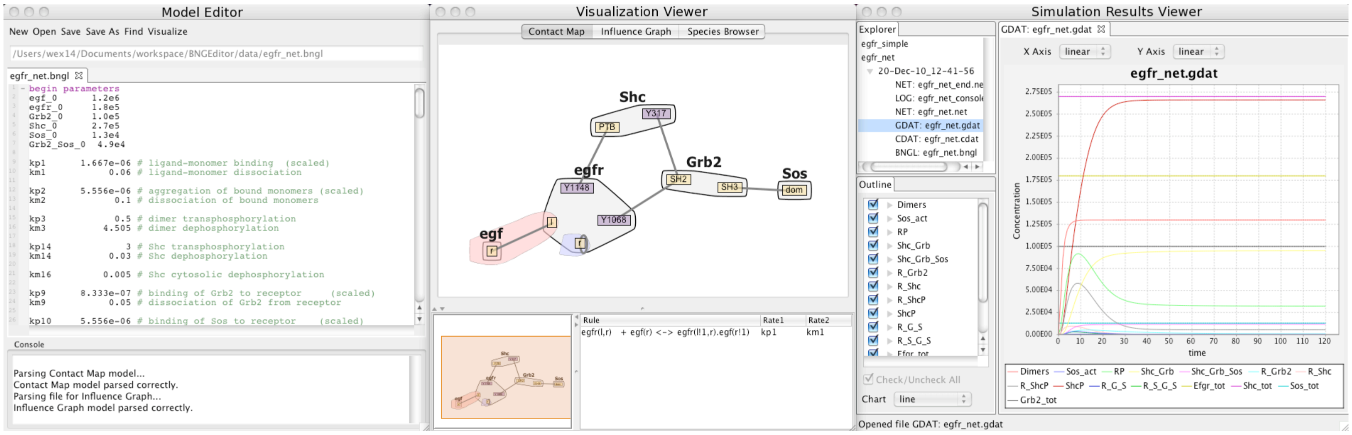
RULEBENDER provides a complete development environment for RBM. RULEBENDER is freely available and open-source (Java with SWT, PREFUSE (http://prefuse.org) and JFREECHART (http://www.jfree.org/jfreechart) libraries.) RULEBENDER's main features include syntax checking, syntax highlighting, visual global model exploration, integrated execution of simulations, support for multiple simulation modules, simulation journaling, interactive plotting including comparison of multiple datasets, and parameter scanning. The three main interface windows provide support for RBM construction and simulation through interactive editing, visual model exploration, and results analysis capabilities respectively as shown in Fig. 1.

Model Browsing and Editing. The Model Editor window (Fig. 1, left panel) in RULEBENDER consists of a toolbar, a tabbed editor pane, and a console that displays output of simulations and other tasks performed on the model. In addition to standard text editor capabilities, the editor provides a number of useful features for creating and editing RBMs in the BIONETGEN Language (BNGL) format. These features include syntax highlighting, code folding, error highlighting, and tabbing to allow multiple files to be open at the same time. The editor also provides a BNGL model template that expedites model construction. A detailed description of the BNGL language is given in Faeder *et al.*, 2009; an example rule is shown in Fig. S1. The editor window could easily be adapted to support other RBM languages.

Model Exploration and Debugging. To further support model understanding and debugging, the Visualization Viewer window (Fig. 1, middle panel) provides both global and local views of the model currently loaded in the Model Editor window. The views are automatically generated from models.

The first global view is the Contact Map, which is a graph representation of the molecules and rules in the model. Molecules (gray shapes in Fig. 1) are the basic units of a BIONETGEN model

[*]To whom correspondence should be addressed.



```
- begin parameters
  egf_0        1.2e6
  egfr_0       1.8e5
  Grb2_0       1.0e5
  Shc_0        2.7e5
  Sos_0        1.3e4
  Grb2_Sos_0   4.9e4

  kp1      1.667e-06 # ligand-monomer binding  (scaled)
  km1           0.06 # ligand-monomer dissociation

  kp2      5.556e-06 # aggregation of bound monomers (scaled)
  km2            0.1 # dissociation of bound monomers

  kp3            0.5 # dimer transphosphorylation
  km3          4.505 # dimer dephosphorylation

  kp14             3 # Shc transphosphorylation
  km14          0.03 # Shc dephosphorylation

  km16         0.005 # Shc cytosolic dephosphorylation

  kp9      8.333e-07 # binding of Grb2 to receptor     (scaled)
  km9           0.05 # dissociation of Grb2 from receptor

  kp10     5.556e-06 # binding of Sos to receptor    (scaled)
```

```
Parsing Contact Map model...
Contact Map model parsed correctly.
Parsing file for Influence Graph...
Influence Graph model parsed correctly.
```

Rule	Rate1	Rate2
egfr(l,r) + egf(r) <-> egfr(l!1,r).egf(r!1)	kp1	km1

Figure 1 The RULEBENDER interface with Model Editor window including Console for simulation (left), a Visualization Viewer window (middle), and a Simulation Results Viewer window (right). RULEBENDER's main features include syntax checking, syntax highlighting, visual global model exploration, integrated execution, support for multiple simulation modules, simulation journaling, interactive plotting including comparison of multiple datasets, and parameter scanning.

and may contain components (yellow and purple rectangles in Fig. 1), which represent internal and binding states of the molecule. Edges in the map show that two components may be bound through the action of a rule. Selecting an edge brings up a list of all rules that create or destroy bonds between the linked components. Selecting one of these rules brings up bubble sets (Collins *et al.*, 2009) that highlight separately the components modified by the rule (called the reaction center) and the components whose states are required to have specified values for the rule to be applied (called the reaction context), as shown in Figs. 1 and S2. The Contact Map provides global organization for a rule set and enables the user to browse the rules in a systematic fashion.

An alternative global view is the Influence Graph, which shows interactions between the rules. A rule can activate (inhibit) another rule by producing (consuming) a reactant required by the other rule. Selecting a rule node displays the rule text in the lower right information pane of the visualization window (Fig. S3). Both visualizations enable quick identification of "orphan" molecules or rules that do not interact with other molecules/rules, thus further supporting understanding and debugging of the models.

Simulation and Results Analysis. BIONETGEN simulator commands can be performed on the model currently loaded in the Editor window. If the model is valid, simulation *results* are displayed in the Simulation Results Viewer. Otherwise, syntax errors and other output are reported via the console in the lower section of the editor window. Parameter scanning (Fig. S4), which allows the researcher to interactively see the effects of varying the value of a single model parameter, is also supported.

The simulation results, along with the model files, parameters scan results, *etc.* are stored and managed by RULEBENDER through a journaling system that allows users to distinguish simulation results. The Simulation Results Viewer contains an explorer in the upper left quadrant of the window (Fig. 1, right panel), which makes it easy to retrieve the exact version of the model that produced a specific set of results.

Network files generated from a model may be displayed with syntax highlighting (Fig. S5). Charts in linear or log scale show the time series for concentrations of chemical species and observables, as seen in the far right window of Fig. 1. Any number of species and observables can be compared in the same chart (Fig. S6). Furthermore, multiple simulation runs can be compared in order to analyze the effects of changing the model (Fig. S7).

A species and observables browser is also provided (lower left of the results viewer in Fig. 1). Graph-based models of species and observables, similar to the Contact Map, are displayed by selecting an observable in the Simulation Results Viewer (Fig. S8).

3 CONCLUSION

RULEBENDER is a powerful tool for the development of RBMs that makes RBM accessible to users with a wide range of computational experience, while providing a uniform interface across computing platforms. The support of RBM creation, debugging, and interactive visualization expedites the RBM learning process and reduces model construction time; while built-in model simulation and analysis with multiple linked views streamline the execution and analysis of newly created models and generated networks. A development cycle that includes close interaction with expert users allows RULEBENDER to better serve the needs of the systems biology community.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation [grant numbers CCF-0829788, IIS-0952720] and by the National Institutes of Health [grant numbers GM-076570, UL1-RR024153]. Thanks are also due to Yao Sun, John Sekar and Bill Hlavacek.

REFERENCES

Collins,C. *et al.* (2009) Bubble Sets: Revealing Set Relations with Isocontours over Existing Visualizations. *IEEE Trans. Vis. and CG*, **15**, 1009-16

Colvin,J. *et al.* (2010) RuleMonkey: software for stochastic simulation of rule-based models. *BMC Bioinformatics* **11**, 404.

Danos,V. *et al.* (2007) Scalable Simulation of Cellular Signaling Networks. *Lect. Notes. Comput. Sci.* **4807**, 139-157

Faeder,J.R. *et al.* (2009) Rule-Based Modeling of Biochemical Systems with BioNetGen, *Methods Mol. Biol.*, **500**, 113-168

Feret,J. *et al.* (2009) Internal coarse-graining of molecular systems. *Proc. Natl. Acad. Sci. USA*, **106**, 6453-6458

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