Systems biology

BIOCHAM: an environment for modeling biological systems and formalizing experimental knowledge

Laurence Calzone, François Fages and Sylvain Soliman*

Projet Contraintes, INRIA Rocquencourt, BP105, 78153 Le Chesnay Cedex, France

ABSTRACT

Summary: BIOCHAM (the BIOCHemical Abstract Machine) is a software environment for modeling biochemical systems. It is based on two aspects: (1) the analysis and simulation of boolean, kinetic and stochastic models and (2) the formalization of biological properties in temporal logic. BIOCHAM provides tools and languages for describing protein networks with a simple and straightforward syntax, and for integrating biological properties into the model. It then becomes possible to analyze, query, verify and maintain the model with respect to those properties. For kinetic models, BIOCHAM can search for appropriate parameter values in order to reproduce a specific behavior observed in experiments and formalized in temporal logic. Coupled with other methods such as bifurcation diagrams, this search assists the modeler/biologist in the modeling process.

Availability: BIOCHAM (v. 2.5) is a free software available for download, with example models, at http://contraintes.inria.fr/BIOCHAM/
Contact: Sylvain.Soliman@inria.fr

1 INTRODUCTION

With the outbreak of new techniques in experimental biology, there has been an increasing amount of data that needs to be treated, classified and analyzed. A typical way to organize these data is to gather all this information into a consensus diagram, or a mathematical model, that integrates heterogeneous pieces of data (observations in wild-type or mutated organisms under various conditions). The study of the networks and the corresponding mathematical models help understand complex systems, make predictions and drive future experiments.

The BIOCHemical Abstract Machine BIOCHAM software (Fages *et al*., 2004; Calzone *et al*., 2006) is part of the recent effort in computational systems biology to design formal languages for describing qualitative or quantitative models of biochemical systems, such as: Moleculizer (Lok and Brent, 2005), BioNetGen (Blinov *et al*., 2004), Pathway Logic (Eker *et al*., 2002), Bioambients (Regev *et al*., 2004), Hybrid Petri Nets (Hofestädt and Thelen, 1998), Hybrid Concurrent Constraint languages (Bockmayr and Courtois, 2002), the (stochastic) π-calculus (Regev *et al*., 2001), etc. However, there has been no comparable effort on formalizing the biological properties known from the experiments and used to build the models. The promise of such formal specifications

would be to systematically validate and maintain models using automated reasoning tools.

BIOCHAM is an attempt to make progress on this issue of automatic validation, using model-checking techniques. It is based on two formal languages: one straightforward rule-based language that allows the user to write models of biochemical networks and to perform multi-level analyses with a minimum knowledge of mathematics or computer science; and one powerful yet simple temporal logic language (CTL or LTL) used for formalizing experimental knowledge. The first versions of the software were restricted to boolean model analysis using the NuSMV model-checker (Cimatti *et al*., 2002). BIOCHAM now permits continuous or stochastic simulations, and also model validation or revision with respect to a formal qualitative or quantitative specification. As a result, BIOCHAM features functions to automatically check that no mistake is made at different stages of the model-building process. For example it is possible to verify that whenever an interaction or a molecule is added to the diagram, the global properties of the system, expressed by temporal logic formulae, are conserved. Similarly, it is possible to automatically search for parameter values that reproduce the specified behavior of the system in different conditions.

2 WRITING MODELS

A model is defined by a set of reaction rules, possibly equipped with kinetic expressions, a list of parameter values and initial conditions. A specification that accounts for the relevant biological properties can also be added to the model as a list of temporal logic formulae. A single BIOCHAM file can be used for boolean, continuous or stochastic analyses. According to the type of study chosen by the user, the model receives different interpretations, e.g. the kinetic expressions are respectively ignored, seen as reaction rates or interaction probabilities.

A network of protein interactions is thus modeled by a list of biochemical reaction rules such as `CycB + CDK => CycB-CDK` where `CycB` and `CDK` are two proteins and `CycB-CDK` is their complex. The locations of the interactions can also be explicitly specified by compartment names such as the nucleus, the cytoplasm, etc.: `CycB::cyto + CDK::cyto => CycB-CDK::cyto`, or in a transport rule: `CycB-CDK::cyto => CycB-CDK::nucleus`. A kinetic expression can be attached to a reaction rule, as follows: `k*[CycB]*[CDK] for CycB + CDK => CycB-CDK`. As mentioned above, this expression is ignored in the boolean view of the model, while in the continuous interpretation, it is
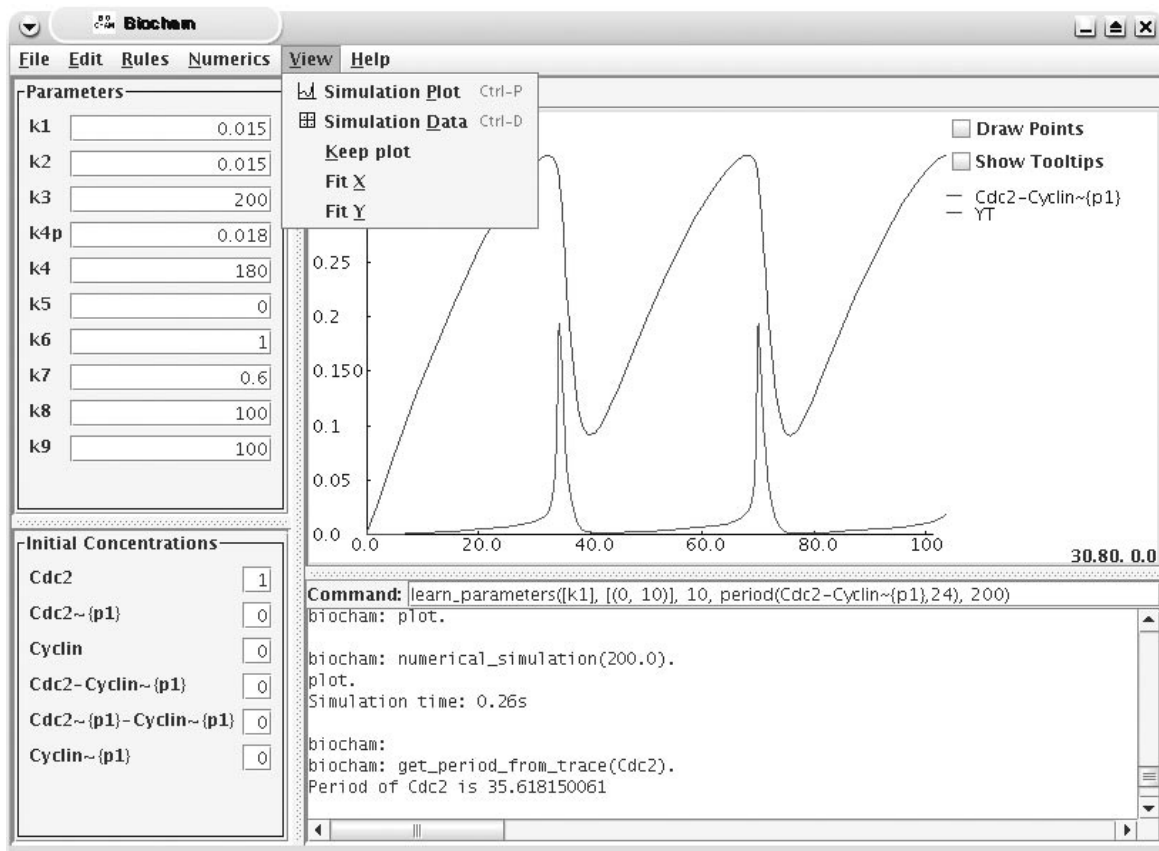
Fig. 1. Graphical user interface with a model of the cell cycle.

derived as a term in the differential equations of the reactants and products, such as $d(CycB-CDK)/dt = k^*[CycB]^*[CDK]$. The whole system of ODEs is thus automatically generated from the set of reaction rules. In the stochastic view, the kinetic expressions are interpreted as transition probabilities.

The input file is saved in the BIOCHAM format, with the extension .bc and can be loaded with the console or graphical user interface (depicted in Fig. 1). The BIOCHAM models (without their specification) can also be saved in, or imported from, SBML level 2 v. 1 format (http://www.sbml.org/).

3 CHECKING FORMAL BIOLOGICAL PROPERTIES

The main contribution of BIOCHAM is the formalization of the qualitative and quantitative experimental knowledge in temporal logic, namely CTL (Clarke *et al.*, 1999) and LTL with numerical constraints. For example, if the activity of molecule A is known to oscillate, this information can be inserted into the BIOCHAM specification as a CTL formula abbreviated by *oscil(A)*. If the period has been measured experimentally and is equal to 24 h, an LTL formula shortened as *period(A,24)* is added. Reachability, stability and checkpoint properties can be formalized as well (Calzone *et al.*, 2006).

The formalization of biological properties is an essential step to verify, and even automatically learn, biochemical reaction rules.

For example, if the structure of the protein network has already been established [e.g. Kohn's map (Kohn, 1999)], it is possible to formally check that no information was lost in the wiring of the network (Chabrier-Rivier *et al.*, 2004). In BIOCHAM, this verification has been implemented in Prolog for the numerical LTL properties (Calzone *et al.*, 2006), and through the NuSMV model-checker (Cimatti *et al.*, 2002) for the CTL properties. Furthermore, a model revision algorithm computes ways to complete or modify a model by adding or deleting biochemical rules, in order to satisfy a temporal logic specification.

For kinetic models, the choice of parameter values is often a tedious task. Usually, the modeler tweaks parameters until the desired behavior is obtained. This *trial-and-error* process can be partly automated by running an automatic search for one or more parameters (e.g. *k*) on a given interval ([0,10]) with a predefined number of steps (20) such that some properties of the system [*period(A,24)*] become true, over a given time horizon (200 h). The function is written as follows: `learn_parameters([k], [(0,10)], 20, period(A,24),200)`.

4 CONCLUSION

In BIOCHAM, two formal languages have been defined: one rule-based language for describing biochemical processes, interpreted at three abstraction levels (boolean, concentration, stochastic) and one temporal logic language for formalizing the relevant biological

properties as a specification. This allows BIOCHAM to assist the modeler by analyzing and verifying the structure of the model through simulation and property checking, and by proposing parameter values or reaction rules that satisfy the expected behavior of the system given as a specification.

The most important aspect of the environment is the possibility to provide the model with various types of information, qualitative or quantitative, that can be found in the literature. This documentation effort is a fundamental step to facilitate the modeling process and the re-use of models, as it offers a formal framework to compare and combine several models together.

ACKNOWLEDGEMENTS

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.

Bockmayer,A. and Courtois,A. (2002) Using hybrid concurrent constraint programming to model dynamic biological systems. In *Proceedings of the International Conference on Logic Programming (ICLP'02)*, 2002, Copenhagen, Springer-Verlag, Germany, 85–99.

Calzone,L. *et al.* (2006) Machine learning biochemical networks from temporal logic properties. *Trans. Comput. Syst. Biol.*, in press.

Chabrier-Rivier,N. *et al.* (2004) Modeling and querying biochemical interaction networks. *Theor. Comput. Sci.*, **325**, 25–44.

Cimatti,A., Clarke,E., Enrico Giunchiglia,F.G., Pistore,M., Roveri,M., Sebastiani,R. and Tacchella,A. (2002) Nusmv 2: an opensource tool for symbolic model checking. In *Proceedings of the International Conference on Computer-Aided Verification, CAV'02,* Copenhagen, Denmark, .

Clarke,E.M., Grumberg,O. and Peled,D.A. (1999) *Model Checking*. MIT Press, Cambridge, MA.

Eker,S., Knapp,M., Laderoute,K., Lincoln,P., Meseguer,J. and Sönmez,M.K. (2002) Pathway logic: symbolic analysis of biological signaling. In *Proceedings of the seventh Pacific Symposium on Biocomputing,* pp. 400–412.

Fages,F., Soliman,S. and Chabrier-Rivier,N. (2004) Modelling and querying interaction networks in the biochemical abstract machine BIOCHAM. *J. Biol. Phys. Chem.*, **4**, 64–73.

Hofestädt,R. and Thelen,S. (1998) Quantitative modeling of biochemical networks. *Silico Biol.*, **1**, 39–53.

Kohn,K.W. (1999) Molecular interaction map of the mammalian cell cycle control and DNA repair systems. *Mol. Biol. Cell*, **10**, 2703–2734.

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

Regev,A. *et al.* (2004) Bioambients: an abstraction for biological compartments. *Theor. Comput. Sci.*, **325**, 141–167.

Regev,A., Silverman,W. and Shapiro,E.Y. (2001) Representation and simulation of biochemical processes using the pi-calculus process algebra. In *Proceedings of the sixth Pacific Symposium of Biocomputing,* pp. 459–470.