BIOINFORMATICS APPLICATIONS NOTE Vol. 27 no. 5 2011, pages 734–735
doi:10.1093/bioinformatics/btq727

Systems biology

Advance Access publication January 5, 2011

MIRACH: efficient model checker for quantitative biological pathway models

Chuan Hock Koh[1,2,3], Masao Nagasaki[3,*], Ayumu Saito[3], Chen Li[3], Limsoon Wong[2] and Satoru Miyano[3]

[1]NUS Graduate School for Integrative Sciences and Engineering, Singapore 117597, [2]School of Computing, National University of Singapore, Computing Drive, Singapore 117417 and [3]Human Genome Center, Institute of Medical Science, University of Tokyo, 4-6-1 Shirokanedai, Minato-ku, Tokyo 108-8639, Japan

Associate Editor: Trey Ideker

ABSTRACT

Summary: Model checking is playing an increasingly important role in systems biology as larger and more complex biological pathways are being modeled. In this article we report the release of an efficient model checker MIRACH 1.0, which supports any model written in popular formats such as CSML and SBML. MIRACH is integrated with a Petri-net-based simulation engine, enabling efficient online (on-the-fly) checking. In our experiment, by using Levchenko *et al*. model, we reveal that timesaving gains by using MIRACH easily surpass 400% compared with its offline-based counterpart.

Availability and implementation: MIRACH 1.0 was developed using Java and thus executable on any platform installed with JDK 6.0 (not JRE 6.0) or later. MIRACH 1.0, along with its source codes, documentation and examples are available at http://sourceforge.net/projects/mirach/ under the LGPLv3 license.

Contact: masao@ims.u-tokyo.ac.jp

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received and revised on December 2, 2010; accepted on December 27, 2010

1 INTRODUCTION

As larger and more complex biological pathways are being modeled, the manual validation of these models becomes tedious if not impossible. Therefore, there is a growing interest in the development and application of model checking algorithms for biological pathway models.

PRISM is a probabilistic model checker that is widely used in many different domains (Heath *et al*., 2008). As PRISM is meant for a wide range of domains, it has its own specific PRISM format for models to adhere to. Clarke *et al*. (2008) introduced BioLab, an algorithm to verify properties written in probabilistic bounded linear temporal logic, using the BioNetGen modeling (rule-based) framework. Genetic Network Analyzer (GNA) is software for the modeling and simulation of qualitative models in the form of piecewise-linear differential equations, which also includes the ability to do model checking (Batt *et al*., 2005). Donaldson and Gilbert (2008a) developed a Monte Carlo offline-based model checker (MC2). MC2 has the advantage of being independent from the modeling framework and is able to perform model checking as long simulation results can be obtained. However, this implies that the full simulation needs to be completed and all traversed states recorded before model checking can commence. This wastes CPU and storage resources if the decision of validity or rejection for the simulation can be determined early in its execution. Online or on-the-fly model checking does the exactly this. It carries out model checking during the simulation run and results need not be recorded as simulation runs are only executed for as long as a decision needs to be made.

In this article, we present an on-the-fly probabilistic model checker, MIRACH, for quantitative pathway models that supports popular formats such as SBML (Hucka *et al*., 2003) and CSML (http://www.csml.org/). This quantitative model checker, MIRACH, would certainly be a valuable addition to the available arsenal of qualitative (GNA) and rule-based (BioLab) model checkers.

2 METHODS

To build a model checker, one of the first steps is defining a formalism to express the rules to be checked. For MIRACH, we have decided to use PLTL (Probabilistic Linear Temporal Logic) because it is sufficient for probabilistic model checking in general and is easy to write and interpret. Due to space constraints, PLTL syntax and semantics are supplied as supplementary Material (Supp. Doc. 1).

Our on-the-fly sample verification is as follows: at each time step of a simulation run, each LTL statement that has yet to be accepted or rejected is checked. A LTL statement is removed once its truth-value can be determined and the simulation stops when all LTL statements have been determined or the predetermined termination simulation time has been reached. If the LTL statement cannot be decided at a particular time point, the species involved in the LTL statement will be stored in memory for this time point as some temporal logics might need to refer to the values of previous states in order to make a decision.

The above paragraph describes how MIRACH decides the truth-value of properties for a single simulation run. However, for stochastic models, each simulation run produces different results. To understand stochastic models, we would then need to consider issues such as, does the model satisfies the property with at least (or at most) probability θ or what is the probability that a property holds.

To address the former question, the sample efficient hypothesis testing (Younes, 2006; Younes *et al*., 2006) was implemented. Hypothesis testing implemented is based on Wald's sequential probability ratio test (Wald, 1945), which could determine after each sample run whether another sample run is required or a hypothesis could be accepted with the prescribed strength using available samples. This is more efficient as opposed to the estimation approach where the probability that the property holds is computed using a predetermined number of samples and compared with the θ.

*To whom correspondence should be addressed.

Table 1. MIRACH versus MC2 (PLTLc) using Levchenko model

	100 samples	1000 samples
MIRACH		
Initialization	6.85 (0.24)	6.86 (0.31)
Simulation and Checking	5.34 (0.20)	40.74 (0.90)
Total time	12.19	47.6
MC2(PLTLc)		
Run simulation and log results	12.14 (0.40)	107.95 (1.52)
Load results and check	10.13 (0.29)	88.58 (1.11)
Total time	22.27	196.53

One hundred samples indicate that 100 simulation runs were executed (similarly for 1000 samples). Results shown are in seconds and are the average of 20 repeated runs. The number in brackets is the SD. All runs are performed on a laptop with 1.6 GHz dual core processor and 2 G RAM running on Linux.

As for the latter, we implemented Wilson interval (Wilson, 1927) to estimate the confidence interval of the probability that the property holds. We have chosen to use Wilson interval instead of the simpler normal approximation interval because normal approximation is known to perform badly when the sample probability is close to 0 or 1 and fail completely when it is at 0 or 1. Due to this, one cannot devise a sequential sampling algorithm that stops sampling once the confidence interval falls below a certain value (user defined). Wilson interval does not have these limitations and allows us more flexibility and efficiency in our model checker.

3 PERFORMANCE

It is not difficult to appreciate that an online approach is almost certainly more efficient than offline in terms of time efficiency since it only runs as long as it needs to and does not read and write to the hard disk.

One offline model checker similar to MIRACH is MC2 (PLTLc) by Donaldson and Gilbert (2008a). Both model checkers are written in Java and supports PLTL. Therefore, we will use MC2 (PLTLc) to illustrate the differences between online and offline checkers.

To draw comparisons between the two model checkers, we need a sample model that can be run on both of the checkers. Our model of choice is a SBML model by Levchenko *et al.* (2000) as it was also used as an example in Donaldson and Gilbert's (2008a) paper.

From Table 1, we see that MIRACH outperforms MC2 (PLTLc) and the time saved increases with sample size. When comparing the runtime for just 1000 samples, the time saved by using MIRACH is already 400%. The sample size needed depends on the problem at hand but in most situations, thousands of samples are insufficient especially with the growing trend of using model checkers as part of parameter estimation routine (Batt *et al.*, 2010; Donaldson *et al.*, 2008b). Our group has also combined MIRACH with parameter estimation (Koh *et al.*, 2010) to investigate cell fate determination of gustatory neurons in *Caenorhabditis elegans* (Saito *et al.*, 2006). In that work, we had to run 20 million samples.

Another performance measure is the minimum memory requirement. Precise memory requirements depend on several factors such as the model used and the properties to be checked. The memory requirement of online checking is likely to be higher than offline checking because the offline method does not carry out checking and simulation concurrently. As described in Section 2, in the checking step, MIRACH needs to store the values of involved species in memory (RAM) when a LTL cannot be decided (neither TRUE nor FALSE) at that time point. However, even in an extreme case, where there are 100 species involved and that property cannot be decided for 100 000 time points, the additional memory (RAM) needed is still <80 MB ($100 \times 100000 \times 8$ bytes). Note that this memory space used will be freed once that particular simulation ends and will not increase with the number of simulation runs.

4 CONCLUSION

In this article, we have presented an efficient model checker, MIRACH 1.0, for validating the ever-growing biological pathway simulation models—both in complexity and quantity. Major contributions include the implementation of the more efficient on-the-fly approach that saves significant amounts of computation time with minimal memory increase, the ability to accept quantitative models directly in the popular SBML and CSML formats, and the first model checker to be integrated with the HFPNe (Nagasaki *et al.*, 2010) simulation engine, an expressive and powerful Petri net framework for defining biological pathway models.

ACKNOWLEDGEMENTS

We are thankful to Sharene Lin for her help rendered in the manuscript preparation and proofreading of this manuscript.

Funding: Singapore National Research Foundation grant NRF-G-CRP-2997-04-082(d) (to L.W. and C.H.K., in parts); National University of Singapore NGS scholarship (to C.H.K., in parts).

Conflict of Interest: none declared.

REFERENCES

Batt,G. *et al.* (2005) Validation of qualitative models of genetic regulatory networks by model checking : analysis of the nutritional stress response in Escherichia coli. *Bioinformatics*, **21** (Suppl. 1), i19–i28.

Batt,G. *et al.* (2010) Efficient parameter search for qualitative models of regulatory networks using symbolic model checking. *Bioinformatics*, **26**, i603–i610.

Clarke,EM. *et al.* (2008) Statistical model checking in BioLab: applications to the automated analysis of T-cell receptor signaling pathway. In Heiner,M. and Uhrmacher,A.M. (eds) *Proceedings of the 6th International Conference CMSB 2008, Rostock, Germany*, Vol. 5307 of *Lecture Notes in Computer Science*, pp. 231–250.

Donaldson,R. and Gilbert,D. (2008a) A Monte Carlo model checker for probabilistic LTL with numerical constraints. *Technical Report TR-2008-282*. University of Glasgow, Department of Computing Science.

Donaldson,R. and Gilbert,D. (2008b) A model checking approach to the parameter estimation of biochemical pathways. In Heiner,M. and Uhrmacher,A.M. (eds) *Proceedings of the 6th International Conference CMSB 2008, Rostock, Germany*, Vol. 5307 of *Lecture Notes in Computer Science*, pp. 269–287.

Heath,J. *et al.* (2008) Probabilistic model checking of complex biological pathways. *Theor. Comput. Sci.,* **391**, 239–257.

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.

Koh,CH. *et al.* (2010) DA 1.0: parameter estimation of biological pathway using data assimilation approach. *Bioinformatics*, **26**, 1794–1796.

Levchenko,A. *et al.* (2000) Scaffold proteins may biphasically affect the levels of mitogen-activated protein kinase signaling and reduce its threshold properties. *Proc. Natl Acad. Sci. USA*, **97**, 5818–5823.

Nagasaki,M. *et al.* (2010) Cell illustrator 4.0: a computational platform for systems biology. *In Silico Biol.,* **10**, 0002.

Saito,A. *et al.* (2006) Cell fate simulation model of gustatory neurons with microRNAs double-negative feedback loop by hybrid functional petri net with extension. *Genome Inform.*, **17**, 100–111.

Wald,A. (1945) Sequential tests of statistical hypotheses. *Ann. Math. Stat.*, **16**, 117–186.

Wilson,E. (1927) Probable inference, the law of succession, and statistical inference. *J. Am. Stat. Assoc.*, **22**, 209–212.

Younes,H.L.S. (2006) Error control for probabilistic model checking. *Lect. Notes Comput. Sci.*, **3855**, 142–156.

Younes,H.L.S. *et al.* (2006) Numerical vs statistical probabilistic model checking. *Int. J. Software Tools Technol Tran.*, **8**, 216–228.