As larger and more complex biological pathways are being modeled, long simulation results can be obtained. However, this implies that the modeling framework and is able to perform model checking (MC2). MC2 has the advantage of being independent from Gilbert (2008a) developed a Monte Carlo offline-based 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 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.
MIRACH versus MC2 (PLTLc) using Levchenko model

<table>
<thead>
<tr>
<th></th>
<th>100 samples</th>
<th>1000 samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialization</td>
<td>6.85 (0.24)</td>
<td>6.86 (0.31)</td>
</tr>
<tr>
<td>Simulation and Checking</td>
<td>5.34 (0.20)</td>
<td>40.74 (0.90)</td>
</tr>
<tr>
<td>Total time</td>
<td>12.19</td>
<td>47.6</td>
</tr>
<tr>
<td>MC2 (PLTLc) Run simulation and log results</td>
<td>12.14 (0.40)</td>
<td>107.95 (1.52)</td>
</tr>
<tr>
<td>Load results and check</td>
<td>10.13 (0.29)</td>
<td>88.58 (1.11)</td>
</tr>
<tr>
<td>Total time</td>
<td>22.27</td>
<td>196.53</td>
</tr>
</tbody>
</table>

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. 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 model 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) to investigate cell fate determination 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 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 × 10000 × 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


