Supplementary Material

Mehmet Emin Bakir, Savas Konur, Marian Gheorghe, Natalio Krasnogor and Mike Stannett

1 Introduction

This material presents three distinct topics: a brief presentation of the most utilised properties, so called *property patterns*, expressing various aspects of behaviour, for a broad range of systems, including biological systems; an overview of the topological features, expressed as graph concepts, utilised in predicting the performance of stochastic simulation algorithm citessapredict and the most suitable model checker for given property patterns; and a short description of the statistical model checker predictor tool.

2 Property Patterns

Model checking algorithmically analyses whether a system model satisfies its requirements specification. Models are usually represented by a finite state transition system and the specification is expressed as a set of logical properties, usually represented by temporal logic formulas. Model checking has been used in system analysis in many areas. However, the very well-known state-space explosion problem associated with large non-deterministic systems, as a result of exhaustive analysis, has prevented it being applied to large systems. Statistical Model Checking (SMC) [17] is one of the methods introduced to alleviate the state-explosion problem issue.

The usage of model checking tools also requires a good understanding of the logical formalisms which is not easy for non-experts. Some of the most common drawbacks in using such tools by non-experts are listed below:

- different tools typically use different modelling and property specification languages, and support the analysis of different collections of properties – users therefore need to familiarise themselves with a range of different technologies;
- while some tools are self-contained, others depend on the use of external third party applications for pre-processing, which means that the users need to learn the techniques involved in using these other tools as well;
- the performance characteristics of any given tool may vary significantly according to the verifications being performed. Where one tool successfully verifies a model's properties efficiently, another may fail; and any given tool may succeed in validating certain properties of a model, but fail to verify others.

Model checking uses *temporal logics* as property specification language. In order to query probabilistic features, probabilistic temporal logics should be used. Several probabilistic property specification languages exist, such as Probabilistic Computational Tree Logic (PCTL) [8], PLTL with numerical constraints (PLTLc) [3] and Continuous Stochastic Logic (CSL) [1, 2, 9].

Table 1: Example biological properties for the quorum sensing system.

Pattern	**Example**
Eventually	The concentration of the signalling molecule AHL exceeds 0.1 μM with more than 90% chance.
Always	The concentration of the signalling molecule AHL is always below the threshold T with a probability greater than and equal to 0.8.
Follows	The emission of AHL is followed by the production of GFP.
Precedes	The production of GFP is preceded by a high concentration of AHL.
Never	The concentration of GFP never reaches 0.1 μM within the first 100 seconds.
Steady-state	In the steady state, the bacteria colony illuminates green fluorescent.
Until	For more than 90% cases, GFP will not be expressed until the concentration of AHL is greater than 0.2 μM.
Recurrence	GFP is repeatedly produced as long as AHL is emitted by *Pseudomonas*.
Next	The probability that the concentration of the AHL molecules exceeds 0.1 μM in the next time instant is less than 0.5.
Release	When the concentration of AHL is greater than 0.2 μM, the bacteria colony illuminates green fluorescent.
Weak Until	GFP will not be produced until the concentration of AHL is greater than 0.2 μM; if the concentration does not reach 0.2 μM, GFP will never be produced.

In order to ease the property specification process and facilitate the access of non-experts to these verification tools, a generic framework, called *property patterns*, to represent common property specifications expressing various aspects of behaviour has been identified. Patterns represent recurring properties, and are generally represented by natural language-like narratives. A number of studies have been conducted in this direction to identify appropriate pattern systems for biological models [12, 5, 4, 6].

We illustrate the pattern concept using a well-known example in systems biology, *quorum sensing*. Quorum sensing (QS) is a mechanism through which bacteria communicate with each other using some chemical molecules, called *signalling molecules*. In this way, they can all synchronise and act together as a group. Each bacterium can sense the signalling molecule, and responds to it in the same way, which creates a group behaviour. Once the QS process is activated, the concentration of the signalling molecule is an indicator of the number of cells in the colony. Here we consider a particular QS system [7], where each bacterium in the colony can sense other pathogenic *Pseudomonas aeruginosa (PA)* bacteria by detecting the AHL molecules (emitted by PA) as signalling molecule; and upon detecting AHL it illuminates green fluorescent protein (GFP). Table 1 provides some example properties for the QS system using the patterns selected from the current literature.

Table 1 provides an informal representation of patterns. Table 2 defines the property patterns formally, provides the corresponding translations into the formal temporal logic specification, and reports five SMC tools currently supporting the relevant pattern expressions.

Table 2: Property patterns.

Patterns	**Formal Description**	**Temporal Logic**	**Supported by**
Eventually	With probability > 0.9, ϕ ("the concentration of the signalling molecule AHL exceeds 0.1 μM") will eventually hold.	$P_{>0.9}\ [F\ \phi]$	PRISM, PLASMA-Lab, Ymer, MRMC and MC2
Always	With probability ≥ 0.8, ϕ ("the concentration of the signalling molecule AHL is below T") continuously holds.	$P_{\geq 0.8}\ [G\ \phi]$	PRISM, PLASMA-Lab, Ymer, MRMC and MC2
Follows	With probability ≥ 1, if ϕ_1 ("the emission of AHL") holds, then ϕ_2 ("the production of GFP") must hold.	$P_{\geq 1}\ [G\ (\phi_1 \rightarrow P_{\geq 1}\ [F\ \phi_2])]$	PLASMA-Lab, MRMC and MC2
Precedes	With probability ≥ 1, ϕ_1 ("high concentration of AHL") precedes or activates ϕ_2 ("the production of GFP").	$P_{\geq 1}\ [\neg\phi_2\ W\ \phi_1]$	PRISM, PLASMA-Lab, Ymer, MRMC and MC2
Never	With probability ≥ 1, ϕ ("the concentration of GFP reaches 0.1 μM within the first 100 seconds") will never hold.	$P_{\geq 1}\ [G\ \neg\phi]$	PRISM, PLASMA-Lab, Ymer, MRMC and MC2
Steady-State	With probability ≥ 1, in the long-run ϕ ("the bacteria colony illuminates green fluorescent") must hold.	$S_{\geq 1}\ [\phi]$ or $P_{\geq 1}\ [FG\ \phi]$	PLASMA-Lab, MRMC and MC2
Until	With probability ≥ 0.9, ϕ_1 ("GFP will not be expressed") holds continuously until ϕ_2 ("the concentration of AHL is greater than 0.2 μM") eventually hold.	$P_{\geq 0.9}\ [\phi_1\ U\ \phi_2]$	PRISM, PLASMA-Lab, Ymer, MRMC and MC2
Recurrence	With probability ≥ 1, ϕ ("GFP is produced as long as AHL is emitted by *Pseudomonas*") repeatedly holds.	$P_{\geq 1}\ [GF\ \phi]$	PLASMA-Lab and MC2
Next	With probability < 0.5, ϕ ("the concentration of the AHL molecules exceeds 0.1 μM") will hold in the next state.	$P_{<0.5}\ [X\ \phi]$	PRISM, PLASMA-Lab, MRMC and MC2
Release	With probability ≥ 1, ϕ_2 ("the concentration of AHL is greater than 0.2 μM") holds continuously until ϕ_1 (" the bacteria colony illuminates green fluorescent.") holds, namely ϕ_1 releases ϕ_2.	$P_{\geq 1}\ [\phi_1\ R\ \phi_2]$	PRISM,PLASMA-Lab, Ymer, MRMC and MC2
Weak Until	With probability ≥ 1, ϕ_1 ("the concentration of AHL is greater than 0.2 μM") holds continuously until ϕ_2 ("GFP is produced") holds, if ϕ_2 does not hold, then ϕ_1 holds forever.	$P_{\geq 1}\ [\phi_1\ W\ \phi_2]$	PRISM, PLASMA-Lab, Ymer, MRMC and MC2

In this table, ϕ_1, and ϕ_2 are state formulas; $\bowtie$ is a relation in the set $\{<, >, \leq, \geq\}$; p is a probability $\in [0, 1]$; and $\overline{\bowtie}$ is the negation of the corresponding equality, $\bowtie$.

Various standard patterns (Existence, Always, Precedes, Never, Until, Release and Weak Until) are supported by all five SMC tools, whereas the Next pattern is supported by all tools except Ymer. The Steady State pattern is supported only by PLASMA-Lab, MRMC and MC2. The Infinitely Often pattern is only supported by PLASMA-Lab and MC2.

3 Graph Concepts

Network analysis of biological systems is a fundamental component of systems biology, it helps to get better understanding of molecular interactions. Topological features of species and reaction graphs have been considered in [15] for predicting the performance of Stochastic Simulation Algorithm (SSA). Such graph related features and graph theoretical analysis have been used to identify complex interaction features of the biological systems [13]. Subsequenly are presented some graph related concepts.

Table 3: Topological features of models.

Graph Features used by [15]	**New Features**
Number of vertices	**Number of non-constant species**: regular species whose populations can change, e.g. catalysts cannot be in this category.
Number of edges	**Species×Reactions**: number of species multiplied by number of reactions
Density of graph	**Update values**: min, mean, max and total number of variable changes when reactions trigger
Degrees: min, mean and max number of incoming, outgoing and all edges	**Sum of the Degrees**: total number of incoming, outgoing and all edges, for each graph
Weakly connected components	
Articulation points	
Biconnected components	
Reciprocity of graph	

The "update values" introduced in this table refer to the number of species whose populations change when a reaction triggers. For example, if the reaction "$2A + 3B \to C$" triggers, the number of variables updated is 3, namely A, B and C.

A *graph* consists of vertices and edges. Typically, a *vertex* represents an entity of a biological system and an *edge* represents relationships between two entities (vertices). In *undirected graphs*, edges do not have direction, therefore, the relationship is symmetric. In *directed graphs*, each edge has a direction originating from one vertex and pointing to itself or to another vertex. One of the basic properties of a vertex is its *degree* which is the number of edges connected to it. In directed graphs, the number of edges pointing to a vertex is called *in-degree* and the number of outgoing edges is called *out-degree*. *Density* of a graph measures how sparse or dense the vertices are interconnected, which is the ratio between the number of existing edges and the total number of possible edges [13, 14]. A *connected component* in a graph is the maximal subset of vertices in which any two vertices are connected through one or more paths. *Strongly connected components* take the direction of the edges into account, whereas the *weakly connected components* ignore the directions [13, 10]. A *vertex cut* or *articulation point* is a vertex in a connected component of a graph whose removal causes the subgraph becoming disconnected [16]. A *biconnected component* is a subgraph which does not have articulation points [16]. *Reciprocity* in directed graphs measures the proportion of mutual connections, i.e. reciprocal edges [13].

The graph concepts used in [15] are presented in the left column of Table 3. The right column has a set of new features presented in our current paper.

4 Experiments

4.1 System and Software

All experiments were conducted on the same desktop computer (Intel i7-2600 CPU @ 3.40GHz 8 cores, 16GB RAM, running under Ubuntu 14.04).

For experimental purposes, we used version 2.0 beta2 of MC2 with the Gillespie2 simulator (which is bundled with the tool) to generate the required simulation traces. We also used: PRISM, version 4.2.1; PLASMA-Lab, version 1.3.2, with default settings and simulator; Ymer 4.2.1 and MRMC version 1.5 together with PRISM 4.2.1 for MRMC performance benchmarking.

4.2 Evaluating Different Feature Groups

In order to show how different types of features affect the prediction accuracy and highlight the quality of the final set of topological features chosen to characterise models, we have carried out some experiments, for which we have identified three different topological feature groups. The first group consists of the 32 features proposed by ([15]), the second group added in the 12 new features we proposed (44 features in total), and the third group is the same as the second group but excludes the computationally expensive graph-theoretic features (e.g. *reciprocity, weakly connected components, biconnected components* and *articulation points*) – this third group consists of 36 features.

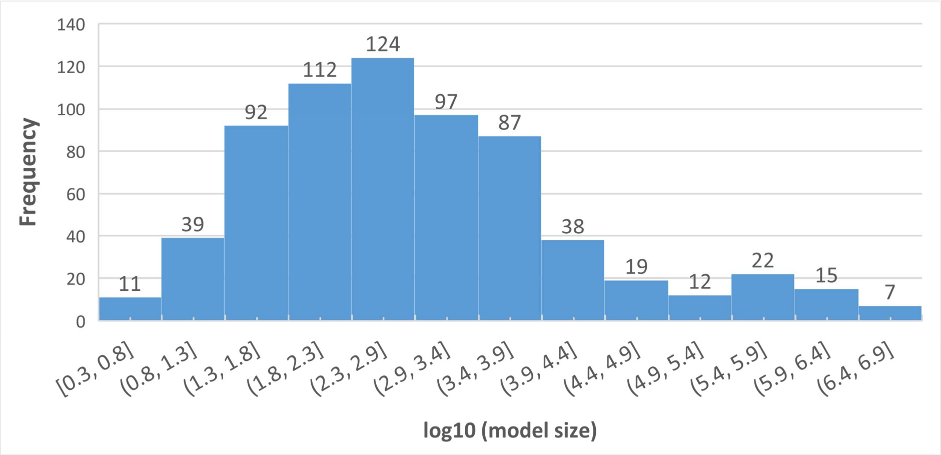


Figure 1: **Model size distribution.** The X-axis plots the (logarithm of) model size and the Y-axis shows the frequency of models within the corresponding X-axis interval. We take "size" as the product of species count and reaction count.

In this experiment, we have considered the biomodels obtained from the EBI database. The distribution of model sizes is presented in Figure 1. Each of the feature groups of these models was submitted to a linear SVC classifier for predicting the fastest *Stochastic Simulation Algorithm*. The prediction accuracies of 10-fold cross-validation for each group are shown in Table 4. The highest prediction accuracy (69.7%) is achieved by using the Group 2 features, but, as already stated, this group includes some relatively computationally expensive properties. By removing these (Group 3), the prediction accuracy dropped just by 1% relative to Group 2, but with considerable reduction in computation time. Therefore, to make automated prediction for large system tractable, we have used only the Group 3 features for the rest of experiments reported in this study.

Table 4: SSA simulation accuracies of the three different topological feature groups.

	Group 1	**Group 2**	**Group 3**
Accuracy	63%	69.7%	68.6%
Variance	0.009	0.005	0.004

4.3 Validation of Classifiers

The 10-fold cross validation method is a well known and widely used validation method of classifiers. However, it is also well known that results tend to be higher than those obtained when the trained classifiers are put into operation. We have therefore performed two additional tests to compare the results against cross validation method. In both of the experiments, we have kept a representative set of models aside for a blind test.

In the first experiment, we have randomly selected 80% of our original data set (i.e. 675 biomodels) for training and kept the remaining 20% for testing. In our second experiment, we have used the entire data set (i.e. 675 biomodels which were uploaded in 2017 and before) for training, and used the new bio-models which were uploaded to the EBI repository in 2018 for validation.

As illustrated in Table 5, the performance of the 10-fold cross validation method and that used in the first experiment is very similar. In 6 cases 10-fold cross validation is slightly better; whereas in 5 of them is the other way around. As for the second experiment, the accuracy is better than the other two for most of the patterns. We believe this is due to a relatively small number of models uploaded in 2018. We expect the accuracy of the blind validation gets less than the 10-cross validation method, as the size of test data grows.

Table 5: 10-fold cross validation vs blind validation.

Pattern	**10-fold cross val.**	**Experiment 1**	**Experiment 2**
Eventually	92.4%	92.6%	100%
Always	90.5%	90.4%	100%
Follows	95.0%	94.8%	92%
Precedes	97.2%	97.0%	100%
Never	91.0%	91.9%	100%
Steady State	94.2%	93.3%	85%
Until	92.8%	91.1%	100%
Infinitely Often	95.0%	94.8%	92%
Next	94.3%	95.6%	100%
Release	94.2%	94.1%	100%
Weak Until	92.3%	92.6%	100%

4.4 Prediction Accuracy vs Model Parameters

In this paper, our main focus was predicting the time performance of a set of available model checkers. As shown in a previous work [15, 11], we can reliably focus on the model *structure* only to make such performance prediction when trying to decide on the fastest simulation engine, and hence we can ignore the model parameters while still being able to robustly predict the best simulator to use.

In order to demonstrate that the model parameters do not affect the prediction accuracy, we have performed an experiment, where we took a subset of 15 models, and evaluated the accuracy of our prediction tool by verifying them without changing the model parameters.

In this experiment, we have considered two different accuracy scores. The first one reports the percentage of correct estimation of the fastest SMC tool. The second one considers a threshold bound for assessing a correct prediction, namely, whenever the relative time difference between the actual fastest SMC tool and the predicted fastest SMC tool is not more than 10% of the actual fastest SMC tool time, then the prediction is considered correct.

The results are presented in Table 6 (see Appendix for the full results). The prediction accuracy for these models is very high, at least 80% in the first case and no less than 93.3% in the second one.

Table 6: Accuracy scores for models with original parameters.

Pattern	**prediction accuracy**	**prediction accuracy with threshold**
Eventually	80%	100%
Always	80%	93.3%
Follows	93.3%	100%
Precedes	86.7%	93.3%
Never	80%	100%
Steady State	86.7%	93.3%
Until	92.8%	93.3%
Infinitely Often	100%	94.8%
Next	94.3%	93.3%
Release	86.7%	93.3%
Weak Until	93.3%	93.3%
Average	**87.2** %	**95.7**%

4.5 Performance Comparison

Figure 2 illustrates the verification time for each tool with respect to model size. Generally speaking, MC2 and MRMC require more time for verification, hence they are less efficient compared to the other tools. Especially, MRMC can verify very few models and its verification time increases exponentially for the larger models. The verification time for Ymer increases almost linearly, i.e. it is fast for small models, but the verification time constantly increases when the model size increases. PLASMA-Lab displays an exponential growth for small size models but it gets more efficient for large size models. Like PLASMA-Lab, PRISM generally is not the fastest option for small sized models whereas it can perform better for the larger models.

5 SMC Predictor Tool

The tool architecture and work-flow are shown in Fig 3. The tool modifies the received input model using model feature analysis. This is done by fixing the stochastic rate constant and the number of species to 1.0 and 100, respectively, and removing multiple compartments.

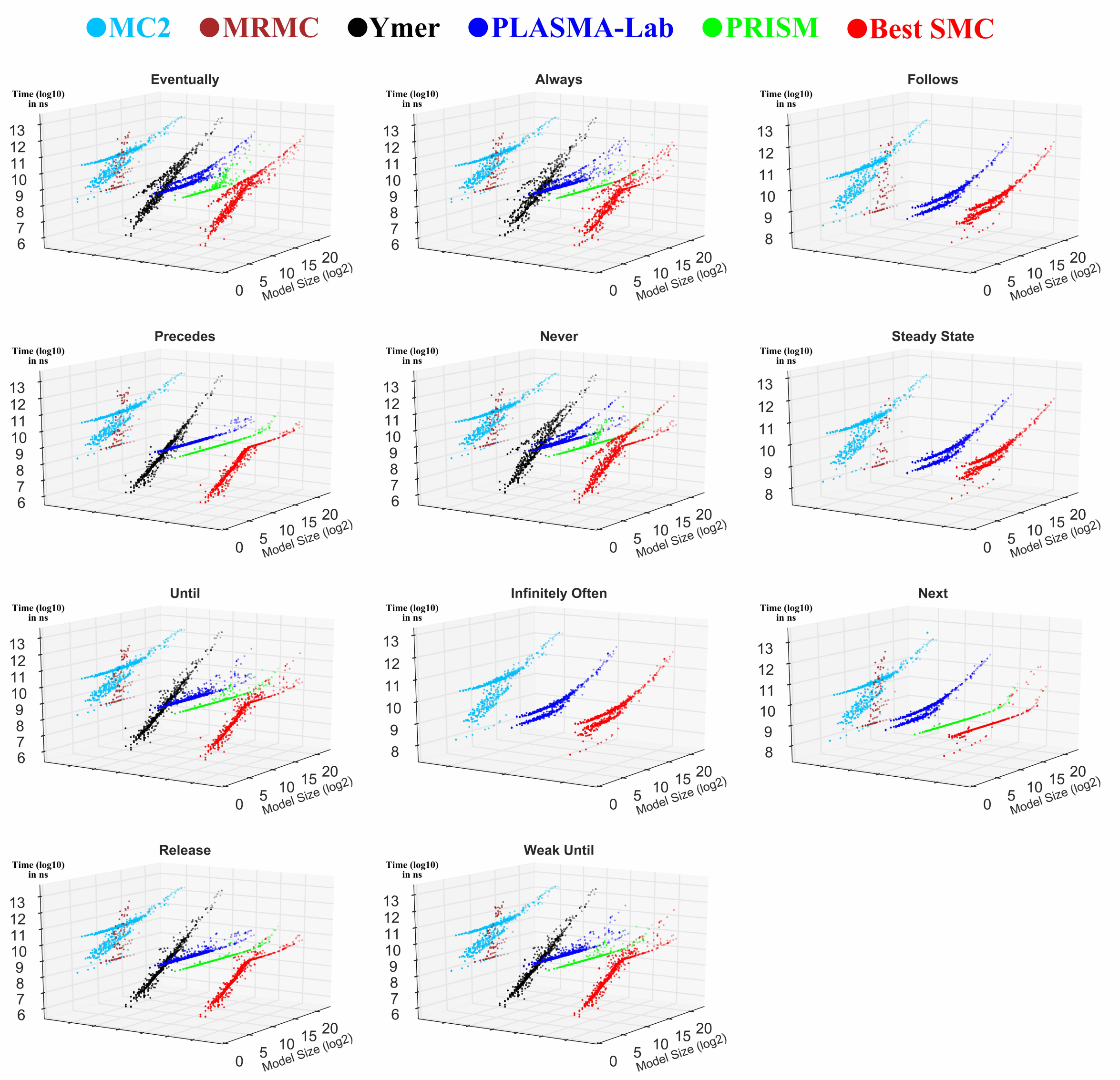


Figure 2: **Performance comparison.** For each property pattern, each tool performance is compared against the best performance, where, X-axes represent the model size (species×reactions) in logarithmic scale ($\log_2$), Y-axes show the relative performance of each SMC tool in comparison with the fastest one, and Z-axes show ($\log_{10}$ scale) the consumed time in nanoseconds.

The modified model is passed to the Model Topology Analysis component, which extracts relevant topological features of the species and reactions graphs (edges, degrees, etc.) as well as various non-graph features, such as the number of updates. These model features, together with the required property pattern, are delivered to the Predictor component. The Predictor component initializes the best classifier identified for use with the given property pattern, conveys the model features to the classifier to predict the fastest SMC tool, and then returns the prediction result.

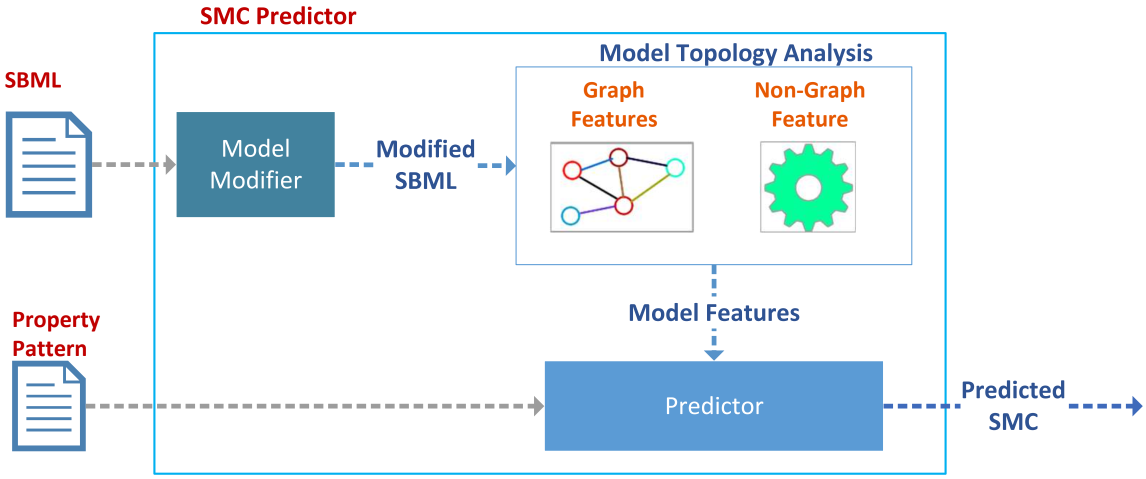


Figure 3: SMC predictor architecture and work-flow.

The SMC Predictor is available at `http://www.smcpredictor.com` together with the details on how to run it and format of the output produced. A tutorial showing through some examples how to run the SMC Predictor and then how to verify the models with the predicted SMC tool is also available.

References

[1] A. Aziz, K. Sanwal, V. Singhal, and R. Brayton. Model-checking continuous-time Markov chains. *ACM Trans. Comput. Logic*, 1(1):162–170, July 2000.

[2] C. Baier, B. Haverkort, H. Hermanns, and J. P. Katoen. Model checking algorithms for continuous-time Markov chains. *IEEE Transactions on Software Engineering*, 29(6):524–541, 2003.

[3] R. Donaldson and D. Gilbert. A Monte Carlo model checker for Probabilistic LTL with numerical constraints. Technical report, University of Glasgow, Department of Computing Science, 2008.

[4] M. Gheorghe, S. Konur, F. Ipate, L. Mierla, M. E. Bakir, and M. Stannett. An integrated model checking toolset for kernel P systems. In *Membrane Computing*, Lecture Notes in Computer Science, pages 153–170. Springer, 2015.

[5] S. Konur. Towards light-weight probabilistic model checking. *J. Applied Mathematics*, 2014:814159:1–814159:15, 2014.

[6] S. Konur and M. Gheorghe. A property-driven methodology for formal analysis of synthetic biology systems. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 12(2):360–371, 2015.

[7] S. Konur, M. Gheorghe, C. Dragomir, L. Mierla, F. Ipate, and N. Krasnogor. Qualitative and quantitative analysis of systems and synthetic biology constructs using P systems. *ACS Synthetic Biology*, 4(1):83–92, 2015.

[8] M. Kwiatkowska, G. Norman, and D. Parker. PRISM: Probabilistic symbolic model checker. In *Computer performance evaluation: modelling techniques and tools*, pages 200–204. Springer Berlin Heidelberg, 2002.

[9] M. Kwiatkowska, G. Norman, and D. Parker. Stochastic model checking. In *Proceedings of the 7th International Conference on Formal Methods for Performance Evaluation*, SFM'07, pages 220–270, Berlin, Heidelberg, 2007. Springer.

[10] H. W. Ma and A. P. Zeng. Chapter 7 - reconstruction of metabolic network from genome information and its structural and functional analysis. In A. Kriete and R. Eils, editors, *Computational Systems Biology (Second Edition)*, pages 113 – 131. Academic Press, Oxford, 2014.

[11] O. Markovitch and N. Krasnogor. Predicting species emergence in simulated complex pre-biotic networks. *PLOS ONE*, 13(2):1–19, 02 2018.

[12] P. T. Monteiro, D. Ropers, R. Mateescu, A. T. Freitas, and H. de Jong. Temporal logic patterns for querying dynamic models of cellular interaction networks. *Bioinformatics*, 24(16):i227–i233, Aug. 2008.

[13] M. Newman. *Networks: An Introduction.* Oxford University Press, Inc., New York, NY, USA, 2010.

[14] G. A. Pavlopoulos, M. Secrier, C. N. Moschopoulos, T. G. Soldatos, S. Kossida, J. Aerts, R. Schneider, and P. G. Bagos. Using graph theory to analyze biological networks. *BioData Mining*, 4(1):10, Apr 2011.

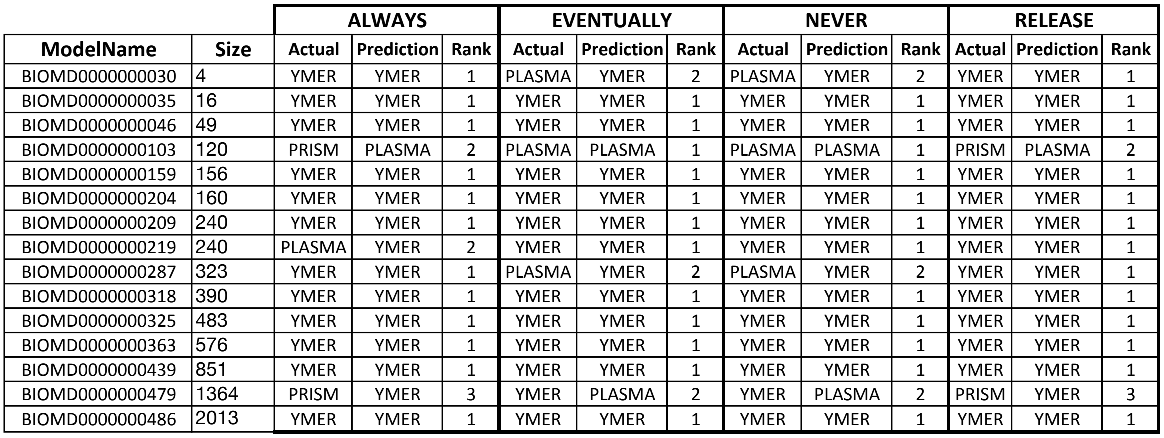
[15] D. Sanassy, P. Widera, and N. Krasnogor. Meta-stochastic simulation of biochemical models for systems and synthetic biology. *ACS Synthetic Biology*, 4(1):39–47, 2015.

[16] S. Wasserman, K. Faust, and D. Iacobucci. *Social Network Analysis : Methods and Applications (Structural Analysis in the Social Sciences).* Cambridge University Press, 1994.

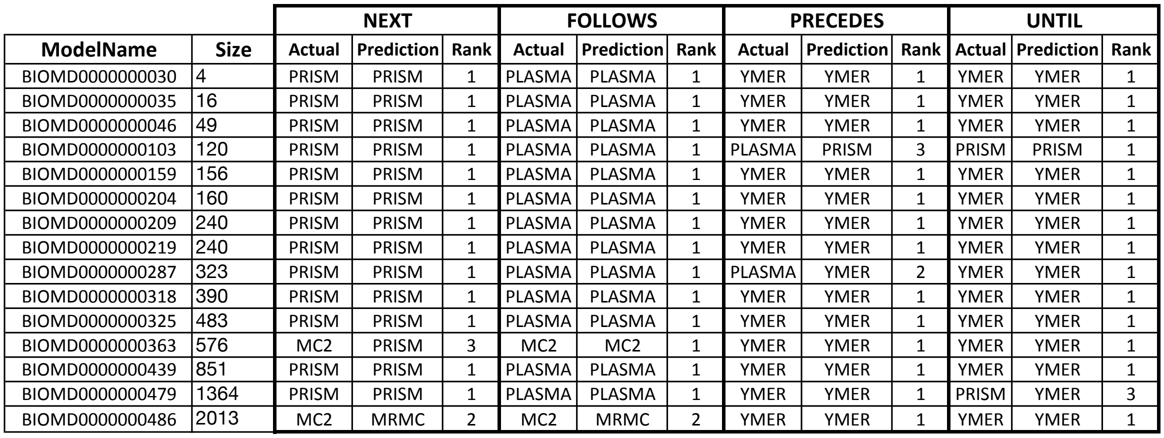
[17] H. L. S. Younes and R. G. Simmons. Probabilistic verification of discrete event systems using acceptance sampling. In *Proceedings of 14th International Conference on Computer Aided Verification*, pages 223–235. Springer, 2002.

Appendix

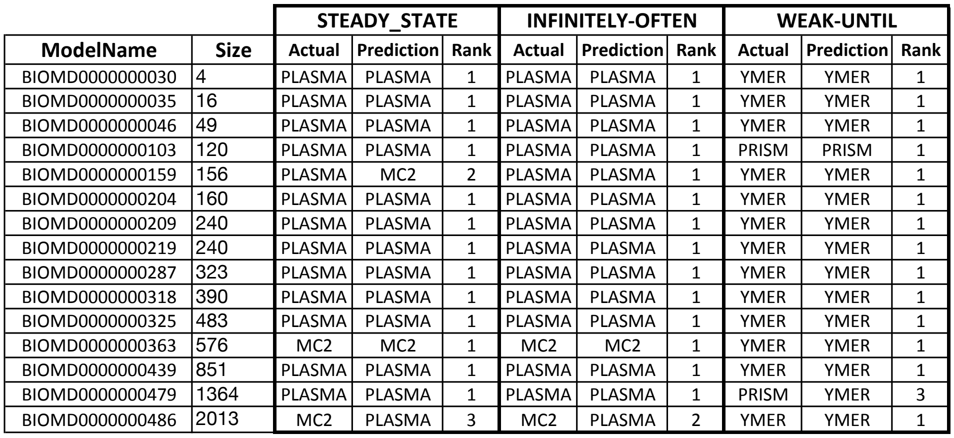
The table below shows the actual and predicted model checker for individual models. The 'Rank' column represents the ranking of the model checker predicted by our tool. For example, for BIOMD0000000103 model, the actual fastest model checker is PRISM, but the predicted one is PLASMA. Although PLASMA is not the fastest tool, it is the 2nd fastest model checker.



		ALWAYS			EVENTUALLY			NEVER			RELEASE		
ModelName	Size	Actual	Prediction	Rank	Actual	Prediction	Rank	Actual	Prediction	Rank	Actual	Prediction	Rank
BIOMD0000000030	4	YMER	YMER	1	PLASMA	YMER	2	PLASMA	YMER	2	YMER	YMER	1
BIOMD0000000035	16	YMER	YMER	1	YMER	YMER	1	YMER	YMER	1	YMER	YMER	1
BIOMD0000000046	49	YMER	YMER	1	YMER	YMER	1	YMER	YMER	1	YMER	YMER	1
BIOMD0000000103	120	PRISM	PLASMA	2	PLASMA	PLASMA	1	PLASMA	PLASMA	1	PRISM	PLASMA	2
BIOMD0000000159	156	YMER	YMER	1	YMER	YMER	1	YMER	YMER	1	YMER	YMER	1
BIOMD0000000204	160	YMER	YMER	1	YMER	YMER	1	YMER	YMER	1	YMER	YMER	1
BIOMD0000000209	240	YMER	YMER	1	YMER	YMER	1	YMER	YMER	1	YMER	YMER	1
BIOMD0000000219	240	PLASMA	YMER	2	YMER	YMER	1	YMER	YMER	1	YMER	YMER	1
BIOMD0000000287	323	YMER	YMER	1	PLASMA	YMER	2	PLASMA	YMER	2	YMER	YMER	1
BIOMD0000000318	390	YMER	YMER	1	YMER	YMER	1	YMER	YMER	1	YMER	YMER	1
BIOMD0000000325	483	YMER	YMER	1	YMER	YMER	1	YMER	YMER	1	YMER	YMER	1
BIOMD0000000363	576	YMER	YMER	1	YMER	YMER	1	YMER	YMER	1	YMER	YMER	1
BIOMD0000000439	851	YMER	YMER	1	YMER	YMER	1	YMER	YMER	1	YMER	YMER	1
BIOMD0000000479	1364	PRISM	YMER	3	YMER	PLASMA	2	YMER	PLASMA	2	PRISM	YMER	3
BIOMD0000000486	2013	YMER	YMER	1	YMER	YMER	1	YMER	YMER	1	YMER	YMER	1



		NEXT			FOLLOWS			PRECEDES			UNTIL		
ModelName	Size	Actual	Prediction	Rank	Actual	Prediction	Rank	Actual	Prediction	Rank	Actual	Prediction	Rank
BIOMD0000000030	4	PRISM	PRISM	1	PLASMA	PLASMA	1	YMER	YMER	1	YMER	YMER	1
BIOMD0000000035	16	PRISM	PRISM	1	PLASMA	PLASMA	1	YMER	YMER	1	YMER	YMER	1
BIOMD0000000046	49	PRISM	PRISM	1	PLASMA	PLASMA	1	YMER	YMER	1	YMER	YMER	1
BIOMD0000000103	120	PRISM	PRISM	1	PLASMA	PLASMA	1	PLASMA	PRISM	3	PRISM	PRISM	1
BIOMD0000000159	156	PRISM	PRISM	1	PLASMA	PLASMA	1	YMER	YMER	1	YMER	YMER	1
BIOMD0000000204	160	PRISM	PRISM	1	PLASMA	PLASMA	1	YMER	YMER	1	YMER	YMER	1
BIOMD0000000209	240	PRISM	PRISM	1	PLASMA	PLASMA	1	YMER	YMER	1	YMER	YMER	1
BIOMD0000000219	240	PRISM	PRISM	1	PLASMA	PLASMA	1	YMER	YMER	1	YMER	YMER	1
BIOMD0000000287	323	PRISM	PRISM	1	PLASMA	PLASMA	1	PLASMA	YMER	2	YMER	YMER	1
BIOMD0000000318	390	PRISM	PRISM	1	PLASMA	PLASMA	1	YMER	YMER	1	YMER	YMER	1
BIOMD0000000325	483	PRISM	PRISM	1	PLASMA	PLASMA	1	YMER	YMER	1	YMER	YMER	1
BIOMD0000000363	576	MC2	PRISM	3	MC2	MC2	1	YMER	YMER	1	YMER	YMER	1
BIOMD0000000439	851	PRISM	PRISM	1	PLASMA	PLASMA	1	YMER	YMER	1	YMER	YMER	1
BIOMD0000000479	1364	PRISM	PRISM	1	PLASMA	PLASMA	1	YMER	YMER	1	PRISM	YMER	3
BIOMD0000000486	2013	MC2	MRMC	2	MC2	MRMC	2	YMER	YMER	1	YMER	YMER	1



		STEADY_STATE			INFINITELY-OFTEN			WEAK-UNTIL		
ModelName	Size	Actual	Prediction	Rank	Actual	Prediction	Rank	Actual	Prediction	Rank
BIOMD0000000030	4	PLASMA	PLASMA	1	PLASMA	PLASMA	1	YMER	YMER	1
BIOMD0000000035	16	PLASMA	PLASMA	1	PLASMA	PLASMA	1	YMER	YMER	1
BIOMD0000000046	49	PLASMA	PLASMA	1	PLASMA	PLASMA	1	YMER	YMER	1
BIOMD0000000103	120	PLASMA	PLASMA	1	PLASMA	PLASMA	1	PRISM	PRISM	1
BIOMD0000000159	156	PLASMA	MC2	2	PLASMA	PLASMA	1	YMER	YMER	1
BIOMD0000000204	160	PLASMA	PLASMA	1	PLASMA	PLASMA	1	YMER	YMER	1
BIOMD0000000209	240	PLASMA	PLASMA	1	PLASMA	PLASMA	1	YMER	YMER	1
BIOMD0000000219	240	PLASMA	PLASMA	1	PLASMA	PLASMA	1	YMER	YMER	1
BIOMD0000000287	323	PLASMA	PLASMA	1	PLASMA	PLASMA	1	YMER	YMER	1
BIOMD0000000318	390	PLASMA	PLASMA	1	PLASMA	PLASMA	1	YMER	YMER	1
BIOMD0000000325	483	PLASMA	PLASMA	1	PLASMA	PLASMA	1	YMER	YMER	1
BIOMD0000000363	576	MC2	MC2	1	MC2	MC2	1	YMER	YMER	1
BIOMD0000000439	851	PLASMA	PLASMA	1	PLASMA	PLASMA	1	YMER	YMER	1
BIOMD0000000479	1364	PLASMA	PLASMA	1	PLASMA	PLASMA	1	PRISM	YMER	3
BIOMD0000000486	2013	MC2	PLASMA	3	MC2	PLASMA	2	YMER	YMER	1