“Semantics and Efficient Simulation Algorithms on an Expressive Multi-Level Modelling Language”, by Helms et al. presents new work on the domain-specific modelling and simulation language ML-Rules [3]. For the first time the language is given a formal semantics which establishes the relationship between the language and its underlying mathematical model, continuous time Markov chains. Furthermore subclasses of the language are identified for which it is possible to specify and implement more efficient approaches to simulation. These new algorithms are demonstrated on substantial case studies. This replicated computation report focusses on the ML-Rules modelling tool, and specifically the new algorithms as demonstrated in the case studies in the paper [1]. The software was straightforward to install and use and the experimental results from the paper could all be reproduced.

CCS Concepts: •Computer systems organization → Embedded systems; Redundancy; Robotics; •Networks → Network reliability;

Additional Key Words and Phrases: RCR report, formal semantics, efficient simulation algorithms, systems biology

ACM Reference format:
DOI: 0000001.0000001

1 INTRODUCTION
The ML-Rules modelling and simulation language is well-established and supported by a fairly mature software toolkit [4] Nevertheless the replication of the results of this paper took place in two stages. Firstly the ML-Rules tool was downloaded and installed and its functionality was investigated in a general sense. In the second stage the focus was explicitly on the results presented in the paper [1], “Semantics and Efficient Simulation Algorithms of an Expressive Multi-Level Modeling Language”. This work was supported by the authors as, in addition to running the models through the ML-Rules software, it was necessary to also to run the results through R scripts in order to create that plots that could be compared to the results in the paper. The authors provided appropriate scripts, allowing the evaluator to conduct the experiments in ML-Rules and the R processing, and these scripts were made available in the ML-Rules repository.
This approach to result replication is similar to the approach taken in the previous RCR report by Luck [2].

2 REPLICATION OF COMPUTATION RESULTS

2.1 Installation of ML-Rules tool

The ML-Rules software suite is already well-developed and is supported by a comprehensive and clear website. This repository contains not only the source code for the tool, and instructions for its installation, but also a sizeable collection of example models, grouped around the publications in which they appear. Downloading the software and installing it was straightforward and proceeded without any problems. Several simple example models were then explored to check that the functionality of the tool is as represented in the paper. This proved to be the case and the tool was intuitive to use both for editing and running models.

2.2 Repeating the Results from the paper

For the purposes of this replication report the focus was on the version of the ML-Rules tool corresponding to the paper [1]. The corresponding entry in the repository [4] provides the models which are used in the three presented case studies, the R scripts needed to create the plots and results as shown in the paper, as well as all the necessary software. Indeed each experiment has been packaged as a .jar file.

In the paper three case studies are presented and all three were scrutinised for this replication report. The first study considers the Wnt/β-catenin pathway, which has been shown to play an important role in cell differentiation and the development of cancer. The ML-Rules model developed in this case is shown to satisfy the characterisation of the $M_{static}$ subclass identified in the paper (its spatial compartmental structure does not change over the course of an execution of the model) allowing it to take advantage of the corresponding Static Simulator. The replication followed the same experimental settings as described in the paper — 100 replications of each parameter setting until simulation time 300. The results reported in Section 7.1 of the paper were reproduced without any problems.

The second case study in the paper considers a less restrictive class of models, where only some rules can be considered as static with respect to the compartmentalisation. This scheme is termed hybrid and the example considered in the paper relates to the unicellular organism Dictyostelium discoideum amoeba, commonly referred to as “slime mold”. This model captures both the intracellular chemical reactions and the physical movement (in a grid) of the amoebae, investigating the formation of a multicellular “slug” which occurs in some circumstances. This uses the Hybrid Simulator. Again the replication followed the same experimental settings as described in the paper — 20 simulation replications, each until simulation time 5000. The number of amoeba cells present was varied across the experiments and the objective was to investigate the formation of slugs. Considerable computational effort was needed but nevertheless the results reported in Section 7.3 of the paper were replicated.

The final case study of the paper presents a model in which the dynamic formation and dissolution of networks of mitochondria is investigated. In particular the model probes the influence of fission protein on mitochondrial structure and health. This model falls into the subclass $M_{bind}$, and the execution of the model takes advantage of the corresponding algorithm. The replication followed the same experimental settings as described in the paper — 20 replications of each experiment setting until simulation time 100. The results obtained were indistinguishable to those presented in the paper.
ACKNOWLEDGMENTS

The author would like to thank Dr. Tobias Helms of the University of Rostock for providing access to the appropriate scripts in the ML-Rules repository.

REFERENCES


Received January 2017