iBioSim 3: A Tool for Model-Based Genetic Circuit Design

Leandro Watanabe,[*,†] Tramy Nguyen,[*,†] Michael Zhang,[*,†] Zach Zundel,[*,†] Zhen Zhang,[*,‡] Curtis Madsen,[*,¶] Nicholas Roehner,[*,§] and Chris Myers[*,†]

†*Department of Electrical and Computer Engineering, University of Utah, Salt Lake City, Utah 84112, United States*

‡*Department of Electrical and Computer Engineering, Utah State University, Logan, Utah 84322, United States*

¶*Department of Electrical and Computer Engineering, Boston University, Boston, Massachusetts 02215, United States*

§*Raytheon BBN Technologies, Cambridge, Massachusetts 02138, United States*

E-mail: l.watanabe@utah.edu; tramy.nguyen@utah.edu; michael13162@gmail.com; zach.zundel@utah.edu; zhen.zhang@usu.edu; ckmadsen@bu.edu; nicholas.roehner@raytheon.com; myers@ece.utah.edu

Abstract

The iBioSim tool has been developed to facilitate the design of genetic circuits via a model-based design strategy. This paper illustrates the new features incorporated into the tool for DNA circuit design, design analysis, and design synthesis, all of which can be used in a workflow for the systematic construction of new genetic circuits.

Keywords

Genetic design automation, standards, SBML, SBOL, SED-ML, COMBINE Archive

iBioSim is a *genetic design automation* (GDA) tool for the modeling, analysis, and design of genetic circuits that is being actively developed at the University of Utah[1,2] (see Figure 1). iBioSim is enabled by community developed standards that promote the model-based design of genetic circuits and allow the sharing of these designs via data repositories. While there exists other genetic design automation tools, such as Cello,[3] iBioSim is not restricted to genetic circuits of logic circuits. In addition, iBioSim allows the use of a wider range of parts and the analysis is fully enabled by standards. While there are standard-compliant simulation tools, such as libRoadRunner,[4] COPASI,[5] and SBMLsimulator,[6] among others, they cannot be used for genetic circuit design. A high-level illustration of the key features of iBioSim is shown in Figure 2.

iBioSim emerged in 2003 as a systems biology tool. The first version included reb2sac,[7] a simulation tool that converts reaction-based networks to stochastic asynchronous circuits for efficient analysis, GeneNet,[8] a learning tool for inferring the connectivity of genetic circuits from time-series data, and an user-interface (UI) to facilitate the usage of reb2sac and GeneNet. iBioSim started targeting synthetic biology applications after the tool was used to design a Genetic C-element *in silico*.[9]

In the first version, the tool used a custom modeling representation called *genetic circuit model* (GCM) as a high-level abstraction to represent genetic regulatory networks. However, in the second version, the tool adopted standards for reproducibility and sharing of models and designs that included the *Systems Biology Markup Language* (SBML)[10] and the *Synthetic Biology Open Language* (SBOL).[11,12] A schematic editor was implemented for constructing models using a graphical user-interface (GUI). New analysis methods were also implemented, including the incremental stochastic simulation algorithm (iSSA),[13] which works with small time increments and checks statistics at the end of each time step to constrain the initial values of the next time step; *stochastic model checking*,[14] which uses continuous-time Markovian analysis to reason about the design's correctness with respect to stochastic properties that capture its critical behaviors; and grid-based models of dynamic cellular populations.[15]

This paper presents the features of the most recent version of iBioSim, which enables a design workflow that leverages models and their analysis to guide the design choices made when constructing genetic circuits. The remainder of this paper describes this workflow in further detail and highlights the new features incorporated into the new version of the tool.

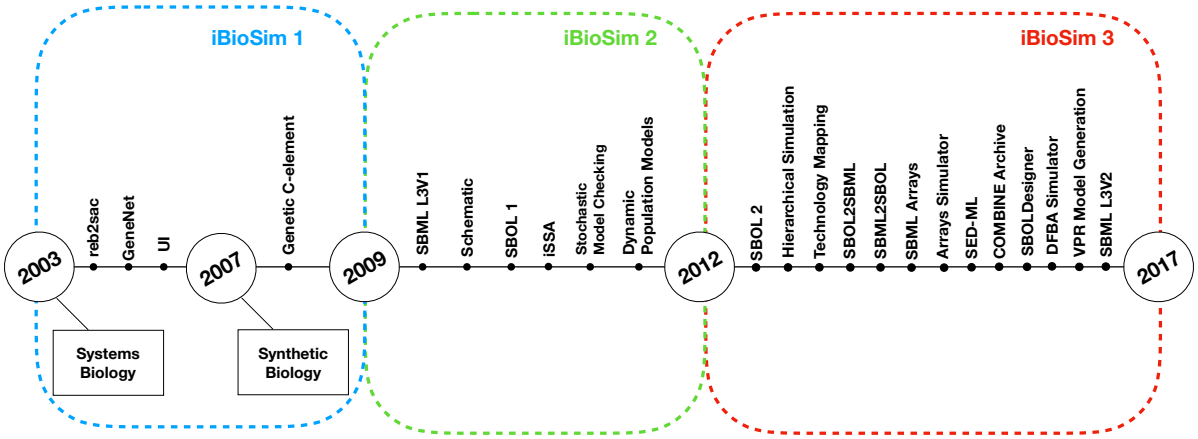

Figure 1: This timeline shows the evolution of iBioSim and highlights the key features implemented in each version. The latest version, iBioSim 3, includes support for new standards, the latest SBML and SBOL versions, and additional SBML packages. Furthermore, the tool supports new features for DNA circuit design, model generation, additional analysis methods, and synthesis methods.

DNA Circuit Design: A genetic circuit design in iBioSim begins by using the SBOLDesigner[16] tool to select genetic parts from the SynBioHub part repository (formally known as the SBOL Stack[17]). This DNA-level design is expressed using version 2 of the SBOL. SBOLDesigner is an intuitive sequence editor tool that is incorporated into iBioSim as a plugin. The structural layer of genetic designs can be viewed and created hierarchically in SBOLDesigner's canvas. SynBioHub is a repository for synthetic biology designs that allows storing and sharing genetic designs represented in SBOL. This feature facilitates model-based design of genetic circuits by providing the means to construct new designs from

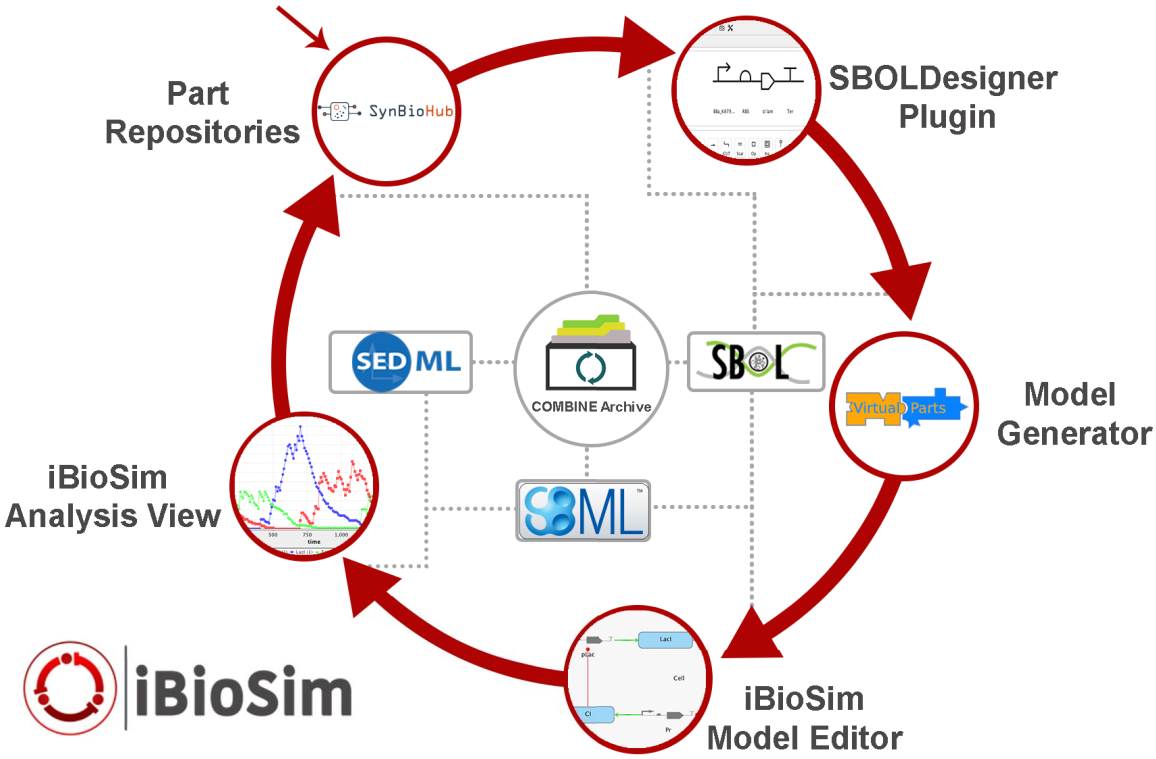

Figure 2: This is a high-level diagram of the genetic circuit design workflow supported by iBioSim. The red arrows indicate the flow between the different software components and dotted lines indicate the output of each step that is then used by the proceeding software component in the workflow. First, genetic parts encoded using SBOL are fetched from SynBioHub using the SBOLDesigner plugin to construct the DNA-level design encoded using SBOL. Next, the DNA design is augmented with interaction data using the Virtual Parts model generator, and the functional SBOL is converted into an SBML model. The resulting mathematical model can then be refined and parameters configured using iBioSim's model editor. The SBML model can be analyzed in iBioSim as described by an associated SED-ML document. The data created for the SBOL parts, the SBML model, and the analysis can be shared and documented by uploading these artifacts to SynBioHub as a COMBINE archive.

existing modeled parts.

Model Generation: The VIRTUAL PARTS REPOSITORY (VPR) model generator is utilized to obtain *interaction* data, as described in,[18,19] from the SYNBIOHUB to add functional information to the SBOL description. For example, it adds the proteins that act as *transcription factors* for the *promoters*, as well as their *coding sequences* in the DNA-level design. These *protein components* are coupled with the *DNA components* constructed by SBOLDESIGNER along with their interactions into functional *module definitions*. Next, an SBOL to SBML converter[20] can be applied to translate the structural and functional information of the corresponding SBOL into a quantitative model expressed in the SBML Level 3 Version 2. Since SBOL is used to represent qualitative models, the quantitative information required by SBML is inferred.[20] However, this SBML model can then be further refined and model parameters added using iBIOSIM's model editor. Any changes made can be mapped back to SBOL using the SBML to SBOL converter.[21]

Analysis: iBIOSIM supports simulation of SBML models using a variety of different simulation methods, such as *ordinary differential equations* (ODEs) and *stochastic simulation*. iBIOSIM is the first software tool that is capable of simulating SBML models that utilize the *hierarchical model composition* (*comp*)[22] and *arrays* packages without flattening out these structures,[23,24] a process that can potentially take longer than simulation. Another feature of iBIOSIM's simulation capabilities is the ability to perform *flux balance analysis* (FBA) on SBML models encoded using the *flux balance constraints* (*fbc*) package.[25] FBA is quite useful when kinetic information about the model is unknown. iBIOSIM also allows the coupling of kinetic models expressed using ODEs with constraint-based models expressed as FBA. Such hybrid models are called *dynamic FBA* (DFBA) and can be simulated within the tool. Since one of the goals of iBIOSIM is to use standards for the interoperability between tools, the *Simulation Experiment Description Markup Language* (SED-ML)[26] is integrated into iBIOSIM. Each iBIOSIM project is associated with a single SED-ML file, where each analysis corresponds to a single task that is used to specify how a model should be analyzed

(e.g., which simulator to use) and how the results are presented to the user (e.g., how the output plot should look like). The SBOL document, the SBML model, and the SED-ML file along with results of analysis can be collected within a COMBINE Archive[27] and uploaded to SynBioHub.

Synthesis: While the workflow shown in Figure 2 requires manual selections of parts for a genetic design, iBioSim also supports automated methods for part selection leveraging a process called technology mapping.[28] Rather than derive a model from manually composed parts, this process derives a genetic combinational circuit design from a given SBML model by automatically selecting parts to implement the model's specified function. The key challenge that has to be addressed is that the parts selected must not interfere with each other. Namely, there should be no unintended interactions between the proteins produced by each portion of the design.

Discussion: iBioSim is an active tool with many features still being developed. For design, we plan on fine-tuning the model generation procedure with enriched parts. Currently, the model generation infers default parameters when generating models because such information is not available in the parts repository. Such information would lead us to more accurate models. For synthesis, we also plan to take advantage of enriched parts for better part selection, which would help in selecting parts without unintended interactions. In addition, we can further expand on the part selection by integrating DFBA in the technology mapping procedure. Lastly, the technology mapping procedure in iBioSim is limited to combinational circuits. Currently, we are developing an extension to support asynchronous sequential circuits. For analysis, we plan on expanding on our SED-ML support to enable more complex analysis and to link experimental data to models.

Availability: With the exception of some older analysis methods[7] and model generation methods[8] that are written in C/C++, the majority of iBioSim is written in Java leveraging pure-Java libraries such as JSBML[29] and libSBOLj.[30] iBioSim is an open-source project available publicly at: `https://github.com/MyersResearchGroup/iBioSim`. Each

part described in the workflow can be used as a stand-alone application.

Author Contribution

L. Watanabe implemented hSSA, DFBA, arrays, and SBML L3V2 support. T. Nguyen implemented VPR model generation and SBML to SBOL conversion. M. Zhang added support for SBOLDesigner as a plugin in iBioSim. Z. Zundel moved the codebase from an internal repository to GitHub and added Maven support for the tool. N. Roehner implemented SBOL to SBML conversion. N. Roehner and T. Nguyen implemented technology mapping. C. Madsen refactored the codebase to use JSBML. C. Madsen and Z. Zhang implemented various stochastic model checking methods and the language for specifying stochastic properties. L. Watanabe and T. Nguyen refactored the codebase and separated the code into independent modules that can be used as stand-alone libraries. C. Myers added support for SED-ML and Combine Archive, and also supervised all of the work above.

Conflict of Interest

None declared.

Acknowledgement

The authors would like to thank Nathan Barker, Scott Glass, Kevin Jones, Hiroyuki Kuwahara, Nam Nguyen, Tyler Patterson, and Jason Stevens for their contribution to the development of earlier versions of iBioSim.

The authors of this work are supported by the National Science Foundation under Grant No. CCF-1218095, CCF-1748200, and DBI-1356041. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

(1) Myers, C. J.; Barker, N.; Jones, K.; Kuwahara, H.; Madsen, C.; Nguyen, N.-P. D. iBioSim: a tool for the analysis and design of genetic circuits. *Bioinformatics* **2009**, *25*, 2848–2849.

(2) Madsen, C.; Myers, C. J.; Patterson, T.; Roehner, N.; Stevens, J. T.; Winstead, C. Design and Test of Genetic Circuits Using iBioSim. *IEEE Design Test of Computers* **2012**, *29*, 32–39.

(3) Nielsen, A. A. K.; Der, B. S.; Shin, J.; Vaidyanathan, P.; Paralanov, V.; Strychalski, E. A.; Ross, D.; Densmore, D.; Voigt, C. A. Genetic circuit design automation. *Science* **2016**, *352*.

(4) Sauro, H. M.; Karlsson, T. T.; Swat, M.; Galdzicki, M.; Somogyi, A. libRoadRunner: a high performance SBML compliant simulator. *bioRxiv* **2013**,

(5) Hoops, S.; Sahle, S.; Gauges, R.; Lee, C.; Pahle, J.; Simus, N.; Singhal, M.; Xu, L.; Mendes, P.; Kummer, U. COPASI a complex pathway simulator. *Bioinformatics* **2006**, *22*, 3067–3074.

(6) Dörr, A.; Keller, R.; Zell, A.; Dräger, A. SBMLSimulator: A Java Tool for Model Simulation and Parameter Estimation in Systems Biology. *Computation* **2014**, *2*, 246–257.

(7) Kuwahara, H.; Myers, C. J.; Samoilov, M. S.; Barker, N. A.; Arkin, A. P. *Transactions on computational systems biology VI*; Springer, 2006; pp 150–175.

(8) Barker, N. A.; Myers, C. J.; Kuwahara, H. Learning genetic regulatory network connectivity from time series data. *IEEE/ACM transactions on computational biology and bioinformatics* **2011**, *8*, 152–165.

(9) Nguyen, N.-P. D.; Kuwahara, H.; Myers, C. J.; Keener, J. P. The Design of a Genetic Muller C-Element. 13th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC'07). 2007; pp 95–104.

(10) Hucka, M. et al. The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics* **2003**, *19*, 524–531.

(11) Roehner, N. et al. Sharing Structure and Function in Biological Design with SBOL 2.0. *ACS Synthetic Biology* **2016**, *5*, 498–506.

(12) Galdzicki, M. et al. The Synthetic Biology Open Language (SBOL) provides a community standard for communicating designs in synthetic biology. *Nat Biotech* **2014**, *32*, 545–550, Computational Biology.

(13) Winstead, C.; Madsen, C.; Myers, C. iSSA: An incremental stochastic simulation algorithm for genetic circuits. Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on. 2010; pp 553–556.

(14) Madsen, C.; Zhang, Z.; Roehner, N.; Winstead, C.; Myers, C. Stochastic Model Checking of Genetic Circuits. *J. Emerg. Technol. Comput. Syst.* **2014**, *11*, 23:1–23:21.

(15) Stevens, J. T.; Myers, C. J. Dynamic Modeling of Cellular Populations within iBioSim. *ACS Synthetic Biology* **2013**, *2*, 223–229.

(16) Zhang, M.; McLaughlin, J. A.; Wipat, A.; Myers, C. J. SBOLDesigner 2: An Intuitive Tool for Structural Genetic Design. *ACS Synthetic Biology* **2017**,

(17) Madsen, C.; McLaughlin, J. A.; Mısırlı, G.; Pocock, M.; Flanagan, K.; Hallinan, J.; Wipat, A. The SBOL Stack: A Platform for Storing, Publishing, and Sharing Synthetic Biology Designs. *ACS Synthetic Biology* **2016**, *5*, 487–497.

(18) Mısırlı, G.; Wipat, A.; Mullen, J.; James, K.; Pocock, M.; Smith, W.; Allenby, N.; Hallinan, F. S. BacillOndex: An integrated data resource for systems and synthetic biology. *Journal of Integrative Bioinformatics (JIB)* **2013**, *10*, 103–116.

(19) Mısırlı, G.; Hallinan, J.; Pocock, M.; Lord, P.; McLaughlin, J. A.; Sauro, H.; Wipat, A. Data Integration and Mining for Synthetic Biology Design. *ACS synthetic biology* **2016**, *5*, 1086–1097.

(20) Roehner, N.; Zhang, Z.; Nguyen, T.; J.Myers, C. Generating Systems Biology Markup Language Models from the Synthetic Biology Open Language. *ACS Synthetic Biology* **2015**, *4*, 873–879.

(21) Nguyen, T.; Roehner, N.; Zundel, Z.; Myers, C. J. A Converter from the Systems Biology Markup Language to the Synthetic Biology Open Language. *ACS Synthetic Biology* **2016**, *5*, 479–486.

(22) Smith, L. P.; Hucka, M.; Hoops, S.; Finney, A.; Ginkel, M.; Myers, C. J.; Moraru, I.; Liebermeister, W. SBML level 3 package: Hierarchical model composition, version 1 release 3. *Journal of Integrative Bioinformatics (JIB)* **2015**, *12*, 603–659.

(23) Watanabe, L. H.; Myers, C. J. Hierarchical stochastic simulation algorithm for SBML models of genetic circuits. *Frontiers in bioengineering and biotechnology* **2014**, *2*.

(24) Watanabe, L.; Myers, C. J. Efficient Analysis of Systems Biology Markup Language Models of Cellular Populations Using Arrays. *ACS synthetic biology* **2016**, *5*, 835–841.

(25) Olivier, B. G.; Bergmann, F. T. The systems biology markup language (SBML) level 3 package: Flux balance constraints. *Journal of Integrative Bioinformatics (JIB)* **2015**, *12*, 660–690.

(26) Waltemath, D.; Adams, R.; Bergmann, F. T.; Hucka, M.; Kolpakov, F.; Miller, A. K.; Moraru, I. I.; Nickerson, D.; Sahle, S.; Snoep, J. L.; Le Novère, N. Reproducible com-

putational biology experiments with SED-ML-the Simulation Experiment Description Markup Language. *BMC systems biology* **2011**, *5*, 198.

(27) Bergmann, F. T. et al. COMBINE archive and OMEX format: one file to share all information to reproduce a modeling project. *BMC Bioinformatics* **2014**, *15*, 369.

(28) Roehner, N.; Myers, C. J. Directed acyclic graph-based technology mapping of genetic circuit models. *ACS synthetic biology* **2014**, *3*, 543–555.

(29) Rodriguez, N. et al. JSBML 1.0: providing a smorgasbord of options to encode systems biology models. *Bioinformatics* **2015**, *31*, 3383–3386.

(30) Zhang, Z.; Nguyen, T.; Roehner, N.; Misirli, G.; Pocock, M.; Oberortner, E.; Samineni, M.; Zundel, Z.; Beal, J.; Clancy, K.; Wipat, A.; Myers, C. J. libSBOLj 2.0: A Java Library to Support SBOL 2.0. *IEEE Life Sciences Letters* **2015**, *1*, 34–37.

Graphical TOC Entry