A Converter from the Systems Biology Markup Language to the Synthetic Biology Open Language

Tramy Nguyen
Dept. of Electrical and Computer Engineering
tramy.nguyen@utah.edu

Chris J. Myers
Dept. of Electrical and Computer Engineering
myers@ece.utah.edu

1. INTRODUCTION

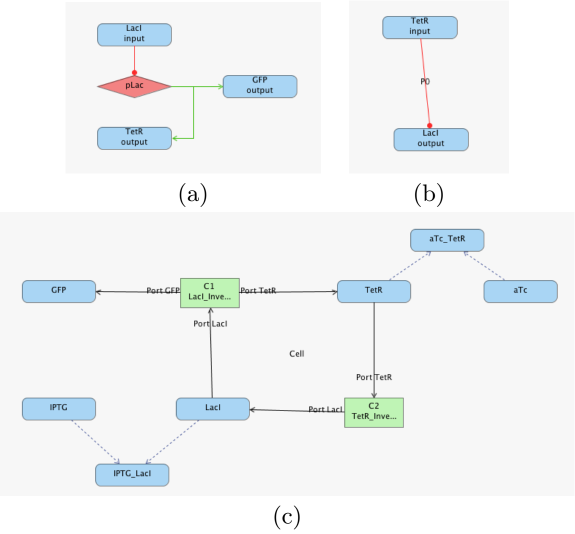
Recently, Version 2.0 of the *Synthetic Biology Open Language* (SBOL) has been released to describe genetic designs [4]. In this new version of SBOL, component types are generalized (Protein, RNA, small molecules, etc.), and new features are added to incorporate behavioral and hierarchical aspects. The *Systems Biology Markup Language* (SBML) [2] is a widely used standard for describing biological behavior. SBOL and SBML serve different purposes. SBOL is intended to describe the structural design of genetic circuits and only basic qualitative behavioral aspects, while SBML's goal is to create models that can be simulated.

Despite their differences, conversion between their common elements is useful. In earlier work, a converter between SBOL to SBML has been reported [5]. This abstract describes a converter from SBML to SBOL. In particular, this converter begins with an SBML model with annotations using the *Systems Biology Ontology* (SBO) [1], and it infers the structure and qualitative functional aspects of the model to produce an SBOL data file. Both of these converters are now integrated within the `iBioSim` *genetic design automation* (GDA) tool being developed at the University of Utah.

2. CONVERSION FROM SBML TO SBOL

SBML includes the following core constructs: *Compartments*, *Species*, *Reactions*, *Parameters*, etc.. In `iBioSim`, genetic designs are described using species and reactions annotated using the SBO. These annotations enable species to be identified as promoters, mRNA, or proteins, reactions as degradation, complex formation, or genetic productions, and modifiers to reactions as activators or inhibitors.

Fig. 1(a) shows an example of a model constructed in `iBioSim` for a LacI inverter. This model is composed of three proteins, LacI, TetR, and GFP represented as blue rectangles. These proteins are represented as species in SBML. In addition, the model contains a promoter pLac. Promoters are also represented as species in SBML. `iBioSim` also includes high-level constructs for genetic circuits. A red arc represents repression and green arcs represent genetic production. These are represented by a reaction that is annotated with SBO terms to describe these relationships. Using the *hierarchical model composition* (comp) package, SBML can instantiate models to construct more complex models, such as the full genetic toggle switch design shown in Fig. 1(c). The dashed arcs in the top-level model of the genetic toggle switch model represents complex formation.



(a) (b)

(c)

Figure 1: (a) A LacI inverter, (b) a TetR inverter, and (c) a genetic toggle switch in iBioSim.

SBOL 2.0 includes *Component Definitions* to describe DNA, RNA, protein, and other types of components. These components can have *Sequences* associated with them, and they can be related to each other through *Interactions*. The Component Definitions and Interactions can be grouped into *Module Definitions*. Module Definitions can be composed hierarchically to form more complex modules. Finally, Module Definitions can use *Model* objects to reference external models written in, for example, SBML.

The conversion begins with an empty *SBOL Document*. Beginning with the top level SBML model, the conversion process recursively converts each sub-model and adds the corresponding data to the SBOL document. This process builds a SBOL Module Definition for each sub-model with a SBOL Model element referencing its SBML model.

Next, each species is converted into a SBOL Component Definition and given a type of *DNA*, *protein*, *small molecule*, etc. If the species has been annotated with sequence information [3], then this can be referred to, as well. For example, species LacI, TetR, and GFP in Fig. 1(a) are converted to Component Definitions of type protein with role transcription factor, and pLac is converted to a Component Definition of type DNA with role promoter.

A *Functional Component* is created within the Module Definition for each species used in the sub-model, and its definition references the corresponding Component Definition for the species. The Functional Component is also marked as being an input, output, or none, and if it is an input or output, it is given a public access type (private, otherwise). For example, within the LacI inverter Module Definition, a public input Functional Component is created for the LacI species that is an instance of the LacI Component Definition.

Next, the converter checks the type of each SBML reaction. A reaction can be an ordinary chemical reaction, a degradation reaction, complex formation reaction, or a genetic production reaction. Each reaction is converted into one or more SBOL Interactions between the corresponding Functional Components. An Interaction in SBOL 2.0 is used to describe the functional relationship between the *reactants*, *products*, and *modifiers* of the reactions. For example, for an ordinary chemical reaction, an Interaction is created that includes all of them as Participants. A degradation reaction includes the degraded protein as a Participant. The complex formation reaction results in an Interaction that includes the separate proteins as reactants and the complex as a product. Finally, the genetic production reaction is converted into several Interactions. In particular, it creates one Interaction for each activator or inhibitor and the promoter, and it creates one production Interaction for each promoter with its products. For example, in Fig. 1(a) the repression arc is converted into an Interaction where LacI is an inhibitor Participant and pLac is a promoter Participant.

In hierarchical SBML models, the same species may appear at various levels of the hierarchy, which is indicated using replacements and replacedBy elements. In particular, a species in the top-level model may have a replacement that states that all instances of the species in the sub-model should be replaced by this top-level species. On the other hand, a replacedBy object indicates a species in the top-level model should be replaced by a species in a sub-model. In SBOL, this operation is accomplished using MapsTo objects. Namely, a MapsTo is used to identify when Component Instances used at different levels of the hierarchy are actually the same Component Instance. A SBOL MapsTo object is created for each SBML replacement or replacedBy object. The MapsTo object maps a local Component Instance to a remote Component Instance. In this case, the local reference is to a Functional Component for the species in the top-level model and the remote is the Functional Component for the species in the sub-model. For a replacement, the MapsTo object has a refinement type of *use local* indicating that the properties of this object should be taken from the Functional Component in the top-level object. For a replacedBy, the MapsTo object has a refinement type of *use remote* indicating that the properties of this object should be taken from the referenced object. For example, in Fig. 1(c), LacI in the TetR_Inverter replaces the top-level LacI which in turn replaces the LacI in the LacI_Inverter. In this case, two SBOL MapsTo objects are created. A visual representation of the generated SBOL is shown in Fig. 2.

3. DISCUSSION

Standards are an important feature of synthetic biology to help overcome the challenges to reproduce designs and reuse published work. While SBML is used to create models for simulation, SBOL is used for the structural design of genetic circuits. This abstract describes a conversion method from SBML to SBOL that extracts the structural and qualitative behavioral information. As a future goal, we plan to attach quantitative information, such as reaction rate constants, species initial amounts, stoichiometry, etc. enabling the ability to round-trip the conversion from SBOL back to SBML. In SBOL, this quantitative data will be represented in SBML using Generic Top Level objects and Annotations.

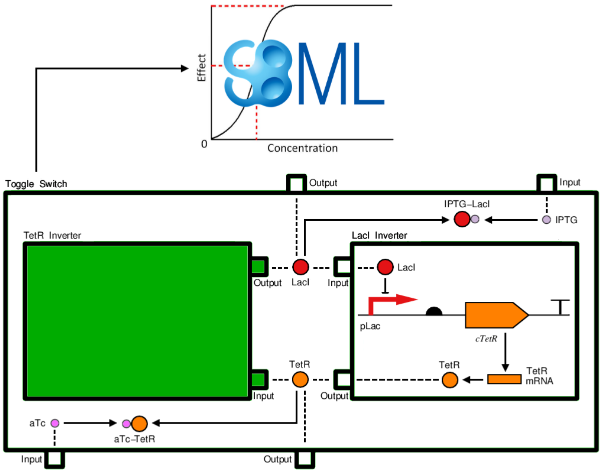


Figure 2: Genetic toggle switch in SBOL 2.0.

Acknowledgements

We thank Nicholas Roehner of Boston University for his help with this converter. This material is based upon work supported by the National Science Foundation under Grant Number 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.

4. REFERENCES

[1] M. Courtot et al. Controlled vocabularies and semantics in systems biology. *Molecular systems biology*, 7(1), 2011.

[2] M. Hucka et al. The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics*, 19(4):524–531, Mar. 2003.

[3] N. Roehner and C. Myers. A methodology to annotate systems biology markup language models with the synthetic biology open language. *ACS Synthetic Biology*, 5(2), 2013.

[4] N. Roehner, E. Oberortner, M. Pocock, J. Beal, K. Clancy, C. Madsen, G. Misirli, A. Wipat, H. Sauro, and C. J. Myers. Proposed data model for the next version of the synthetic biology open language. *ACS Synthetic Biology*, 4(1):57–71, 2015. PMID: 24896221.

[5] N. Roehner, Z. Zhang, T. Nguyen, and C. J. Myers. Generating systems biology markup language models from the synthetic biology open language. *ACS Synthetic Biology*, 2015.