

SBOLDesigner 2.0

Michael Zhang
Dept. of Electrical and Computer Engineering
michael13162@gmail.com

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

1. INTRODUCTION

Synthetic biology is an engineering discipline where biological components are assembled to form devices or systems with more complex functions. A workflow, such as the one shown in Fig. 1, is necessary to advance the field of synthetic biology by giving biologists the ability to abstract their designs and use automated software to ease the development process [9]. In this workflow, DNA sequences for known components are obtained from databases, such as the SBOL Stack, the iGem Registry, and the JBEI-ICE repository. These DNA sequences can then be edited and manipulated inside a sequence *computer-aided design* (CAD) tool, such as **SBOLDesigner** [3, 7], to create a complete structural design of genetic circuit components. Next, circuit *computer-aided engineering* (CAE) tools, such as **Cello** [6] and **iBioSim** [5], can be utilized to compose genetic circuit components into complete genetic circuit designs, including the addition of functional design information for simulation and analysis. Finally, the complete genetic circuit can be archived in the part repositories, completing the cycle.

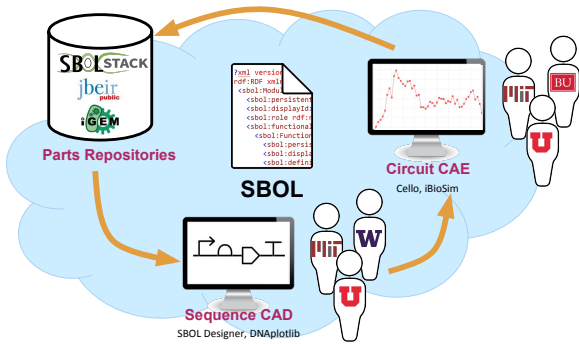


Figure 1: The SBOL enabled workflow for designing genetic circuits.

The *Synthetic Biology Open Language* (SBOL) facilitates communication between these tools and services [4]. SBOL is a standardized digital format that allows biologists to share genetic designs stored in a principled medium. The SBOL standard facilitates communication between experimental biologists, computational biologists, genetic engineers, and their computer tools. The latest version, SBOL 2.0, introduces the specification of generalized genetic components, enhances means to annotate and constrain sequence features, and enables the description of behavioral aspects of a biological design [1].

SBOLDesigner is a simple, biologist-friendly CAD software tool for creating and manipulating the sequences of genetic constructs using SBOL natively. This abstract describes our update of **SBOLDesigner** to support SBOL 2.0, as well as the enhancements that this conversion enables.

2. SBOLDESIGNER

SBOLDesigner has an simple user interface that allows biologists to visualize and edit the details of their creation. It supports hierarchical or nested assembly and offers generic, user-defined parts to ease fabrication from partial sequences to complete genetic constructs. Additionally, the user can flip the orientation of parts and view or edit their names and descriptions. Throughout the design process, SBOL Visual [8] symbols, a system of schematic glyphs, provide standardized visualizations of individual parts.

The original version of **SBOLDesigner** has been updated to SBOL 2.0. While the user interface remains largely the same, the transition from SBOL 1.1 to SBOL 2.0 as the backend data model required re-implementing features using the libSBOLj 2.0 Java library [10]. Fig. 2 shows the main parts of the SBOL 2.0 data model that are used. SBOL 2.0 separates **SequenceAnnotations** from SBOL 1.1 into **Components**, **SequenceAnnotations**, and **SequenceConstraints**, which requires a fundamental change in the representation of parts in **SBOLDesigner**. Overall, adapting **SBOLDesigner** to SBOL 2.0 enables the workflow described above and maintains the relevance of this CAD tool.

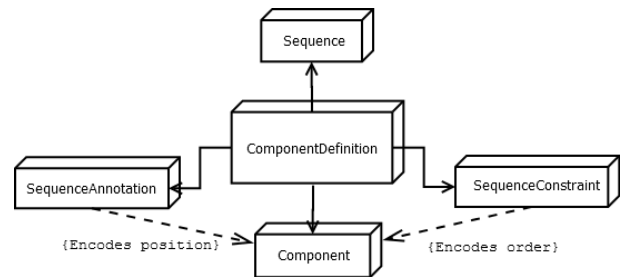


Figure 2: A simplified view of the structural portion of the SBOL 2.0 data model.

Fig. 3 shows the user interface of **SBOLDesigner** with part of the genetic toggle switch circuit on the canvas. This canvas represents a **ComponentDefinition** that brings together information on the design's **Sequence**, its **Components**, and their organization. Below the canvas is a row

of genetic elements that can be added to the design. When placed on the canvas, each element represents a **Component**. These **Components** are organized by **SequenceAnnotations** and **SequenceConstraints**. **SequenceAnnotations** specify the precise **Location** and *orientation* (position and direction) of a **Component**'s **Sequence**. **SequenceConstraints** encode the information on how **Components** are ordered. Clicking on the "focus in" button expands a **Component** to expose its **ComponentDefinition**, generating a nested canvas. This feature allows the user to create hierarchically defined designs.

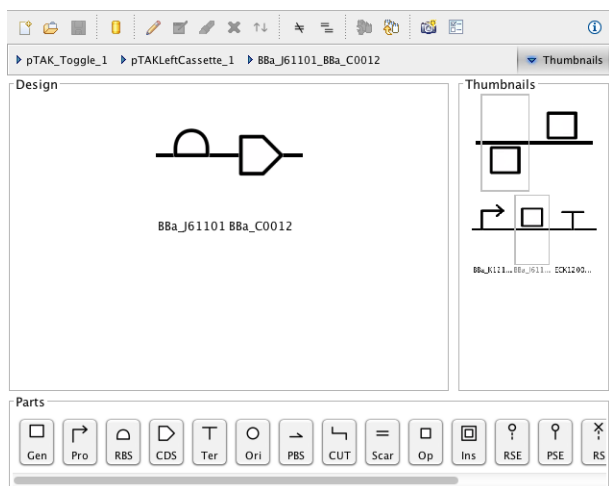


Figure 3: SBOLDesigner's user interface showing a hierarchical view of a genetic toggle switch circuit. This design is composed of two genetic inverters, and each inverter is composed of a promoter, ribosome binding site, coding sequence, and terminator.

Fig. 4 shows the menu for specifying and editing a **ComponentDefinition**'s *role*, *display ID*, *name*, *description*, and *sequence*. The role of the **Component** can be promoter, ribosome binding site, coding sequence, terminator, etc. A new feature is the ability to specify a more specific refinement role from the Sequence Ontology Project [2]. Additionally, **SBOLDesigner 2.0** supports importing **ComponentDefinitions** and **Sequences** from external SBOL, GenBank, and FASTA files.

Figure 4: SBOLDesigner's window for editing a ComponentDefinition for a promoter.

When the design is complete and ready to export, **SBOLDesigner** stitches together all of the specified **Sequences** to form a composite root **Sequence** that is attached to the root **ComponentDefinition**. If desired, this root sequence can then be sent to a DNA synthesis service for fabrication.

3. DISCUSSION

SBOLDesigner completes a workflow for users of *genetic design automation* tools. It combines a simple user interface with the power of the SBOL standard and serves as a launchpad for more detailed designs involving simulations and experiments. Some new features include SBOL Stack integration, the ability to import and write GenBank and FASTA files, extended ontology support, the ability to partially open designs with multiple root **ComponentDefinitions**, and backwards compatibility with SBOL 1.1. Support for RNA and protein parts, more general SequenceConstraints, user annotations, saving into existing designs, and automated **Sequence** optimization are being added. With sequence CAD tools like **SBOLDesigner**, genetic design software like **iBioSim** and **Cello**, and parts repositories like the SBOL Stack, biologists have a wide array of tools for prototyping and automating design of genetic circuits. These tools stimulate advancement of synthetic biology and allow biologists to easily approach genetic design.

4. ACKNOWLEDGMENTS

We thank Evren Sirin (Complexible), Michal Galdzicki (Arzeda), Bryan Bartley (U. of Washington), and John Gennari (U. of Washington) for their work on the original version of **SBOLDesigner**. This work is funded by the National Science Foundation under Grants CCF-1218095 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.

5. REFERENCES

- [1] B. Bartley et al. Synthetic biology open language (SBOL) version 2.0.0. *Journal of Int. Bioinfo.*, 2015.
- [2] K. Eilbeck et al. The Sequence Ontology: a tool for the unification of genome annotations. *GenBio.*, 2005.
- [3] M. Galdzicki, B. A. Bartley, S. C. Sleight, E. Sirin, and J. H. Gennari. Version control for synthetic biology. *IWBDA2013*, pages 19–20, 2013.
- [4] M. Galdzicki et al. The synthetic biology open language (SBOL) provides a community standard for communicating designs in synthetic biology. *Nature Biotechnology*, 32:545–550, 2014.
- [5] C. Madsen et al. Design and test of genetic circuits using iBioSim. *IEEE D. & T.*, pages 32–39, 2012.
- [6] A. Nielsen et al. Genetic circuit design automation. *Science*, page 53, 2016.
- [7] C. Olsen, K. Qaadri, H. Shearman, and H. Miller. Synthetic biology open language designer. *IWBDA2014*, pages 60–61, 2014.
- [8] J. Quinn et al. SBOL visual: A graphical language for genetic designs. *PLOS Biology*, 2015.
- [9] N. Roehner et al. Sharing structure and function in biological design with SBOL 2.0. *ACS SynBio*, 2016.
- [10] Z. Zhang et al. libSBOLj 2.0: A java library to support SBOL 2.0. *IEEE Life Sciences Letters*, 2016.