Standard Enabled Model Generator for Genetic Circuit Design

Göksel Mısırlı
ICOS, School of Comp. Science
Newcastle University
gokselmisirli@gmail.com

Tramy Nguyen
Dept. of Elect. and Comp. Eng.
University of Utah
tramy.nguyen@utah.edu

James Alastair McLaughlin
ICOS, School of Comp. Science
Newcastle University
j.a.mclaughlin@ncl.ac.uk

Chris Myers
Dept. of Elect. and Comp. Eng.
University of Utah
myers@ece.utah.edu

Anil Wipat
ICOS, School of Comp. Science
Newcastle University
anil.wipat@ncl.ac.uk

1 INTRODUCTION

A substantial amount of information is being produced about biological parts that can be used to implement complex designs. However, this information is usually available for human interpretation and often at the DNA sequence level. Computational modeling *in silico* is often exercised manually in order to predict the behavior of designs that can be implemented *in vivo* or *in vitro*. In these models, functional relationships and design constraints between parts in a design are captured in a formal modeling language for simulations. Although this approach may be sufficient for small designs and part libraries, automation of the model generation process is necessary to evaluate larger combinatorial design spaces.

Data standards are particularly important to facilitate design automation, and to pass information between different computational tools when implementing complex workflows. The *Synthetic Biology Open Language* (SBOL) [1, 3, 11] has emerged as an international standard to exchange genetic circuit designs. This standard is useful to specify designs in terms of constituent components. The order and sequences of these components in a design can be captured, and these designs can then be hierarchically used in more complex designs. Critically, SBOL supports capturing molecular interactions between these components. This information is invaluable when creating computational models. Deriving simulatable models, in the form of the *Systems Biology Markup Language* (SBML) [4] documents, from such SBOL designs has already been demonstrated [10]. In this earlier approach, interactions are manually provided.

Building upon these promising efforts, this paper presents a data integration based approach, enabled by data standards, to facilitate the automated creation of computational models from simple definitions of genetic circuits. These definitions may include minimum information that is necessary for DNA synthesis. Computational models are then constructed by extracting knowledge about these DNA components and other interacting entities such as proteins, small molecules, complexes, etc.

2 MODEL GENERATION WORKFLOW

Our model generation workflow is depicted in Figure 1. This workflow consists of three components. SynBioHub (which now incorporates the SBOL Stack [6]), a data repository that stores information about genetic parts and their interactions, iBioSim [5], a software for constructing and modeling genetic circuit designs,

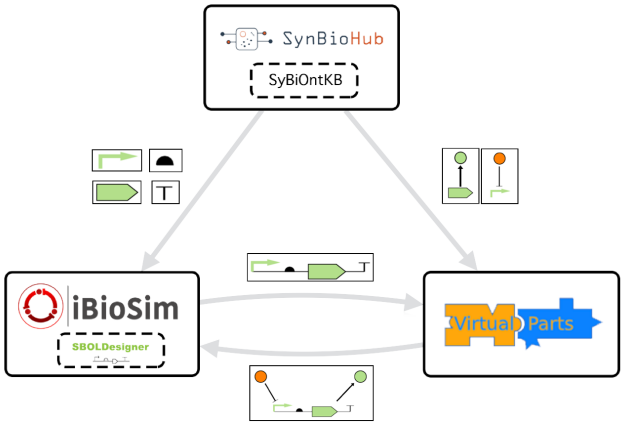

Figure 1: The workflow described in this abstract is summarized in this diagram. The process begins by designing a simple genetic design in iBioSim using the SBOLDesigner plugin. The parts for this design are retrieved from SynBioHub. The generated design is submitted to the VPR, which adds the interactions between the components to the original design. The resulting functional design retrieved in SBOL is translated into an SBML computational model, which can now be simulated using iBioSim.

and the Virtual Parts Repository (VPR) [9], a methodology for storing modular, composable models of parts, together with an interface that allows their composition into larger models. Each of these components is described in more detail in the following sections.

2.1 Data Integration

While a substantial amount of biological information has been produced, this data is often available in different formats and the meaning of data varies between different databases. To make the most of this data in synthetic biology, it is important that these heterogeneous datasets are integrated so that they can be used easily both by humans and software tools. SyBiOntKB [7] is an integrated dataset for synthetic biology applications that has initially been populated with an integrated *Bacillus subtilis* dataset [8]. SyBiOntKB is represented in RDF and the semantics of entities are defined

using the SyBiOnt ontology. SyBiOntKB is now hosted in a publicly shared SynBioHub instance (https://synbiohub.org). SyBiOntKB is then mined for genetic parts which are recorded back in SynBioHub in the form of SBOL objects. The SynBioHub database is backed-by an RDF triplestore and allows uploading, downloading SBOL documents and querying the underlying data using SPARQL.

2.2 Genetic Circuit Construction

In order to construct the DNA-level genetic circuit, our workflow uses the SBOLDesigner [12] plug-in within iBioSim[5]. SBOLDesigner can connect to SynBioHub in order to query for parts by their role, such as promoter, coding sequence, etc., and the user can then select a desired part, and download its sequence and other meta-data. This process can be repeated until a complete, DNA structural-level genetic design is produced.

2.3 Enriching SBOL Designs

The next step of the workflow uses the VPR to examine the DNA-level information provided by SBOLDesigner and enrich it with further details using data found in the SynBioHub repository. This work extends the VPR [9] API in providing functionality which enriches SBOL objects with information about interactions of DNA components with other biological molecules. The resulting enriched SBOL designs are returned from the VPR API now including additional design components for interacting proteins, small molecules, and protein complexes. Interactions such as the translation of proteins, the activation and inhibition of promoters, and complex formation are also incorporated. The following rules are applied for this process:

- A single SBOL `ModuleDefinition` entity is created for a given transcriptional unit design, which may be formed of promoters, ribosome binding sites, coding sequences and terminators. This entity is used to encapsulate molecular interactions between biological components.
- Biological molecules interacting with any of the DNA-based components are added to the `ModuleDefinition`.
- First-level interactions of proteins produced by the transcriptional unit are also included in the `ModuleDefinition`.
- If a biological molecule is not produced using all the entities in the enriched SBOL design, then the corresponding SBOL `FunctionalComponent` is marked as an input. Otherwise, the corresponding `FunctionalComponent` is marked as both input and output. These inputs and outputs are further used when creating hierarchical designs.

In addition to the syntax provided by the SBOL standard, defining the semantics has been an important aspect of the workflow presented here. For example, the VPR uses the *Systems Biology Ontology* (SBO) [2] terms when providing types of interactions, and roles of participants in each interaction, as described here [1]. These terms make the resulting SBOL documents further machine tractable and facilitate deriving models.

2.4 Deriving Dynamic Models

Expressing a model in SBML is necessary to verify the behavior of a design since SBOL, by design, does not include all of the information needed for dynamic simulation. To derive SBML from SBOL, a conversion tool is applied [10]. This tool uses ontology terms to help translate components from one standard to another. SBOL `ComponentDefinitions` are mapped to SBML `Species` and SBOL `Interactions` are mapped to SBML `Reactions`. The `ModuleDefinition` that the `Interaction`(s) are nested in are mapped to SBML `ModelDefinitions`. The mapped reactions use default kinetic laws and reaction rates, specified in [10]. Once the SBOL data has been converted to SBML, the default kinetic laws and reaction rates can be altered through the model editor provided in iBioSim. In the future, these rates parameters may be stored as annotations within the SBOL returned from the VPR. The SBML model constructed in this way can be simulated using a variety of methods, including ordinary differential equations (ODEs) or stochastic simulation methods.

3 CONCLUSIONS

This standard enabled design workflow is important to abstract the details of complexity when dealing with computational models. As demonstrated, simplifying the design process using a tool such as SBOLDesigner has significant benefits. Designs can relatively easily be created by users who can design circuits using DNA parts and still benefit from computational simulations. Moreover, this approach facilitates automation and exploring large designs spaces of biological systems. The standards SBOL and SBML are critical since they serve as domain specific languages that seamlessly connect all the tools within this workflow.

4 ACKNOWLEDGMENTS

The authors of this work are supported by The Engineering and Physical Sciences Research Council grant EP/J02175X/1 (G.M., J.M., A.W.) and the National Science Foundation under Grant No. CCF-1218095 and DBI-1356041 (T.N. and C.M.). 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 funding agencies.

REFERENCES

[1] J. Beal et al. Synthetic biology open language (SBOL) version 2.1.0. *Journal of Integrative Bioinformatics*, 13(3):30–132, 2016.
[2] M. Courtot et al. Controlled vocabularies and semantics in systems biology. *Mol. Syst. Biol.*, 7:543, 2011.
[3] M. Galdzicki et al. The synthetic biology open language (SBOL) provides a community standard for communicating designs in synthetic biology. *Nature biotechnology*, 32(6):545–550, 2014.
[4] 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.
[5] C. Madsen et al. Design and test of genetic circuits using iBioSim. *IEEE Design and Test*, pages 32–39, 2012.
[6] C. Madsen et al. The SBOL Stack: A platform for storing, publishing, and sharing synthetic biology designs. *ACS Synthetic Biology*, 5(6):487–497, 2016.
[7] G. Misirli et al. Data integration and mining for synthetic biology design. *ACS Synthetic Biolology*, 5(10):1086–1097.
[8] G. Misirli et al. Bacillondex: An integrated data resource for systems and synthetic biology. *Journal of Integrative Bioinformatics*, 10(2):224, 2013.
[9] G. Misirli et al. Composable modular models for synthetic biology. *ACM Journal on Emerging Technologies in Computing Systems*, 11(3):22:1–22:19, Dec. 2014.
[10] N. Roehner et al. Generating systems biology markup language models from the synthetic biology open language. *ACS Synthetic Biology*, 4(8):873–879, 2015.
[11] N. Roehner et al. Sharing structure and function in biological design with SBOL 2.0. *ACS Synthetic Biology*, 5(6):498–506, 2016.
[12] M. Zhang, J. A. McLaughlin, A. Wipat, and C. J. Myers. SBOLDesigner 2: An intuitive tool for structural genetic design. *ACS Synthetic Biology*, 2017.