Model-driven design of genetic regulatory networks using virtual parts

Göksel Mısırlı
g.misirli@keele.ac.uk
School of Computing and Mathematics, Keele University

Bill Yang
z.yang22@newcastle.ac.uk
School of Computing, Newcastle University

Anil Wipat
anil.wipat@newcastle.ac.uk
School of Computing, Newcastle University

1 INTRODUCTION

As technology advances, and the cost of DNA synthesis decreases, synthetic biology is moving towards data driven design applications. DNA fragments can be represented as electronic records ready to be composed virtually to create designs of complex genetic circuits. This approach opens the possibility of using representations of various sequence features such as promoters and coding sequences (CDSs) in a computer environment. Whether designs are created manually in a computer-aided environment, or created computationally using heuristic approaches, designs need to be verified and/or optimised. Model-driven design methodologies are already proven as a useful tool to map a virtual design to a physical system in a number of industries including software engineering, aerospace industry, and embedded system development. This approach is also key to create predictable biological applications.

We previously developed the Virtual Parts Repository [4], which provides reusable and modular models of biological parts and interactions. These models are called virtual parts and can be used to create complex models representing biological systems. Due to the increase and the availability of large number of biological parts, virtual parts are ideal to computationally explore large design spaces to create new designs or to find alternative biological solutions.

In parallel to developments in model-driven design approaches, the synthetic biology community developed the Synthetic Biology Open Language (SBOL) [1] to facilitate the exchange of genetic circuit designs. The language allows specifying the order of DNA-based biological parts, their types and interactions between these parts. SBOL is designed as a graph language and can be directly stored in graph repositories. One such database is SynBioHub [3], which allows uploading designs in the form of SBOL documents and querying the underlying data using the SBOL semantics directly. VPR1 uses SBOL only to export information and can only communicate with its built-in relational repository. Model composition is carried out by explicitly querying information about molecular constraints from this repository.

This paper presents the second version of the Virtual Parts Repository (VPR2) in relation to these latest developments. VPR2 has been developed using a modular architecture including components for a Web-based repository, a web service, a client library for computational tools, and a standalone data library to retrieve data from remote SBOL repositories (Figure 1). The web service can be used to retrieve virtual parts, and to create computational models of genetic circuits. Moreover, VPR2 can be used to work with a choice of a SynBioHub instance, which can be installed at a different geographical location. VPR2 has a graph-based repository and allows browsing of the underlying data and models. Moreover, SBOL is used as a domain specific language to control the composition of models.

2 THE VIRTUAL PARTS REPOSITORY 2.0 (VPR2)

VPR2 has been designed to decouple the modelling and genetic circuit design processes. Using this approach, design tools can start taking the benefit of computer simulations without delving into details of complex modelling abstractions. The integration between different tools and VPR2 is carried out using already existing and widely adopted data standards in synthetic biology, such as SBOL and the Systems Biology Markup Language (SBML) [2]. VPR2 uses SBOL to specify data that can be converted into modular and hierarchical SBML models, using two approaches:

- Connected mode. VPR2 is connected to a graph repository to create computational models.
- Disconnected mode. VPR2 uses SBOL documents provided by tools to create computational models. Nominal values are used to parameterise modelling entities.

In the connected mode, VPR2 works directly with a graph repository that can store SBOL data. The repository can be the VPR's default repository or can be selected from various SynBioHub instances. A genetic circuit can be represented using SBOL in terms of DNA-based biological parts (Figure 1), either using start and end positions, or using a relative order of parts. VPR2 queries the SBOL repository to retrieve detailed information about biological parts represented in the design.

Computational tools can use VPR2 to return either a detailed SBOL document or an SBML model as explained below:

- A detailed SBOL document. A simplified definition of a genetic circuit is extended with additional constraints from a

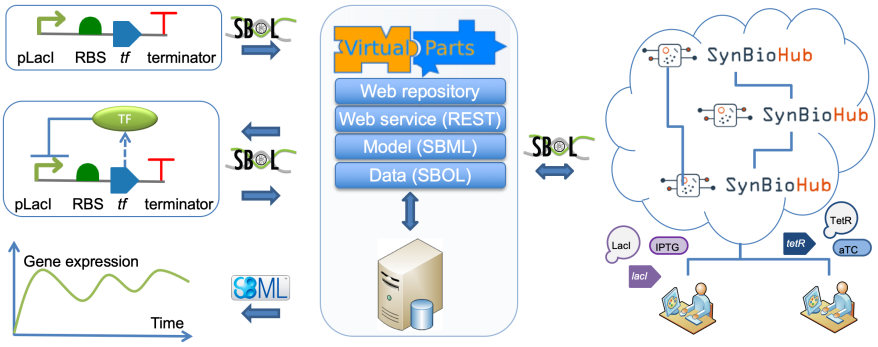

Figure 1: VPR2 has been designed to work with existing standards such as SBOL and SBML. The system can either use data in repositories to return a model for a DNA-based genetic circuit design, or use the information provided by the user to return a model that can be simulated. VPR2 includes a web repository, a web service, model and data layers and a client API.

user-specified SBOL repository. Information about biological constraints such as molecular interactions is used to populate the initial SBOL document. This approach can be used by tools which can create models using custom modelling abstractions and choices of modelling languages.

- An SBML model, capturing the complex relationships between the constituent biological parts.

In the disconnected mode, tools can specify genetic circuits using SBOL with detailed information about molecular interactions between biological parts. This detailed view of a genetic circuit can then be converted into an SBML model.

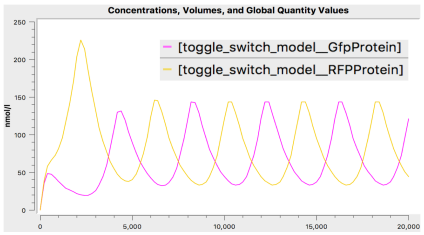

Figure 2: The toggle switch system was modelled using VPR2 and simulated using COPASI.

VPR2 has its own modelling abstraction, representing various biochemical reactions. Examples include the binding and unbinding of biological molecules, transcriptional activation and repression of transcriptional units and the degradation of biological molecules. Virtual parts can include entities for DNA-based parts, proteins and small molecules. Although VPR2 uses nominal values to create modelling entities, tools can override reaction parameters by providing additional details in SBOL documents that are used as input to derive models. Reaction parameters can currently be provided as inline annotations that can be embedded in SBOL documents.

Figure 2 demonstrates the use of VPR2 to model a toggle switch, which was previously modelled using the VPR2 data layer and iBioSim's modelling approach [5]. The aTC and IPTG signals are respectively used to set and unset the GFP and RFP outputs. VPR2 uses a different modelling abstraction utilising the binding and unbinding of molecules to facilitate modularity. In this paper, the system modelled using virtual parts corresponds to 49 biological entities such as promoters, CDSs, proteins and signalling molecules. The resulting model consists of 34 submodels, including 50 biological reactions and 74 different kinetic rate parameters. Submodels can represent virtual parts or templates that are used to instantiate virtual parts.

3 CONCLUSION

VPR2 has been developed to facilitate the search of large design spaces of biological systems using computer simulations. The use of a modular approach and the adoption of existing standards make VPR2 ideal for genetic design automation related workflows. Design tools can, hence, take the advantage of mathematical models to find optimum solutions.

Availability. The VPR2 development version is available at http://v2.virtualparts.org. Please see the *Documentation* menu to access the web service, the web service client library, examples and the data library.

REFERENCES

[1] Cox, R. S., et al. Synthetic biology open language (SBOL) version 2.2.0. *J Integr Bioinform 15*, 1 (2018).

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

[3] McLaughlin, J. A., et al. SynBioHub: A standards-enabled design repository for synthetic biology. *ACS Synth Biol 7*, 2 (2018), 682–688.

[4] Misirli, G., et al. Composable modular models for synthetic biology. *ACM J Emerg Technol Comput Syst 11*, 3 (2014), 22.

[5] Misirli, G., Nguyen, T., et al. A computational workflow for the automated generation of models of genetic designs. *ACS Synth Biol* (2018).