Modular Composition of Synthetic Biology Designs using Rule-Based Models

Citation for published version:
Misirli, G, Waites, W, Cavaliere, M, Zuliani, P, Danos, V & Wipat, A 2016, Modular Composition of Synthetic Biology Designs using Rule-Based Models. in *Proceedings of 8th International Workshop on Bio-Design Automation IWBDA 2016.* pp. 83-84, 8th International Workshop on Bio-Design Automation, Newcastle upon Tyne, United Kingdom, 16/08/16. <http://www.iwbdaconf.org/2016/docs/IWBDA_2016_Proceedings.pdf>

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
Proceedings of 8th International Workshop on Bio-Design Automation IWBDA 2016

Modular composition of synthetic biology designs using rule-based models

Göksel Mısırlı*
ICOS, School of Computing Science, Newcastle University
goksel.misirli@ncl.ac.uk

William Waites*
School of Informatics, University of Edinburgh
wwaites@tardis.ed.ac.uk

Matteo Cavaliere
School of Informatics, University of Edinburgh
mcavali2@staffmail.ed.ac.uk

Paolo Zuliani
ICOS, School of Computing Science, Newcastle University
paolo.zuliani@ncl.ac.uk

Vincent Danos†
School of Informatics, University of Edinburgh
vdanos@inf.ed.ac.uk

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

1. INTRODUCTION

Synthetic biology aims to extend classical genetic engineering by applying principles such as modularity, standardization and abstraction to design complex biological systems and even entire genomes. This process is challenging: biological systems have very large design spaces. As the complexity of engineered systems increases, computational approaches become ever more important to identify biologically feasible solutions. Computational modelling and simulation is crucial to design and verify genetic circuits in a efficient manner. The availability of modular and reusable models is desirable to automate this process. Such models can be composed to create larger models to specify the behaviour of genetic circuits *in vivo* or *in vitro*.

We previously developed an approach termed Standard Virtual Parts (SVPs) to represent models of DNA-based biological parts such as promoters, coding sequences (CDSs) and ribosome binding sites (RBSs) [3]. These models are reusable and annotated with machine-readable information to assist in automated composition. Models are currently implemented using a reaction-based formalism where much of the information required for composition is tightly coupled. Rule-based formalisms on the other hand offer inherent modularity and allow decoupling part models. This approach was previously demonstrated in the Kappa Bio-Brick Framework (KBBF) [4], which helps in the creation of rule-based models of genetic circuits. This framework provides rules that describe the transcription and translation of DNA parts and a modeller provides rules for the interactions of gene products such as transcription factors (TFs). A series of parameters is also provided by the modeller.

Here, we present our work based on SVPs and KBBF aimed at automating the design of genetic circuits using rule-based models (RBMs). Extending KBBF, we have defined abstract templates that can be used to instantiate rules associated with *cis* and *trans* interactions. We then sketch how to annotate rule-based models, using an annotation framework [2] previously developed, in order to automate composition for RBMs and produce executable composite rule-based models. We also outline protocols to incorporate model repositories into the design process. The overall described workflow bridges together well studied topics and suggests, in this way, a feasible protocol (currently, under development by the authors) for the modular design and composition of synthetic circuits.

2. RULE-BASED MODEL TEMPLATES

Templates were created by extending those from KBBF based on a general (and abstract) notion of sliding, docking (binding) and fall off [4]. Instantiating such a template yields specific rules modelling transcriptional and translational processes. For instance, templates can be then used to instantiate rules for:

- Binding: RNA polymerases (RNAPs) to promoter, ribosomes to RBSs and TFs to DNA.
- Sliding: RNAPs moving on DNA (transcription) and ribosomes moving on RNA (translation).
- Fall off: RNAPs or TFs from DNA, or ribosomes from RNA.

Although a single template can be used to instantiate multiple rules, their rates need not be uniform and can depend on the DNA parts present in the instance. For example, lower rates of a fall off instance can be used to model leakiness of stop codons or terminators.

Templates include the possibility to instantiate rules for modelling gene fusion. In fact, when stop codons are not included, ribosomes can keep sliding (Fig. 1) and produce protein chains (for simplicity, we assume that each protein in a chain retains its interactions). We also explicitly include the possibility to model the degradation of protein and of mRNA chains (extending KBBF) with the addition of templates for the binding of protease to proteins and the degradation of same. In this way, one can explicitly model induced degradation by increasing the binding rate of protease to chaperon proteins in a chain.

Docking templates are defined for arbitrary promoter architectures where activator or repressor sites (binding sequences) can be placed anywhere in a genetic circuit. These

*These authors contributed equally to this work

†To whom correspondance should be addressed

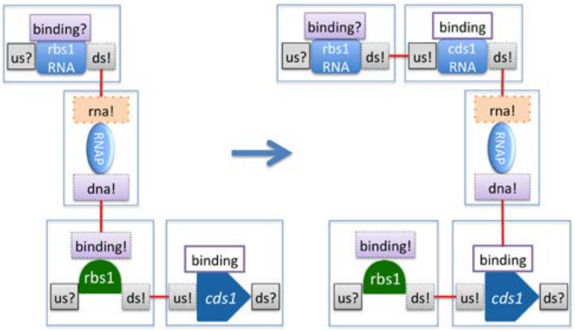

Figure 1: A transcriptional sliding rule. The RNAP starts sliding through DNA (left hand side). As a result, the mRNA chain is extended by transcribing the CDS (right hand side). Outer boxes include agents (rbs1, cds1, rbs1RNA, cds1RNA, RNAP) and their corresponding sites (us, ds, binding, rna, dna), and lines represent agent connections.

templates yield rules that take into account complex regulatory effects of TFs (e.g., Fig. 2).

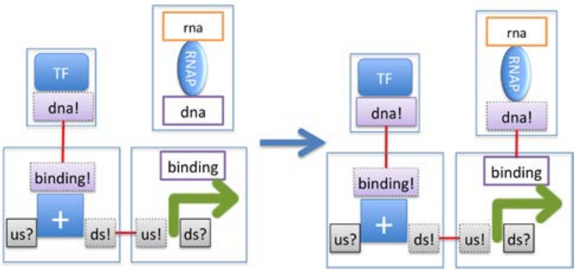

Figure 2: A rule showing the binding of a RNAP to a promoter, when a TF (which may be part of a larger chain) is bound to an upstream binding sequence.

3. MODEL COMPOSITION

Using these templates, we defined rule-based models of basic biological parts[1]. These models can be associated with quantitative parameters to create particular parts models, which can then be merged into executable models. This process can be automated using the information in the models, either available in the specific rule-based syntax or in machine-accessible annotations.

The standard templates use the `DNA` and `P` agent definitions for all DNA and protein entities respectively. A single molecule is distinguished from others using a corresponding identifier for the internal state of a `part` site. These agent definitions increase the reusability and modularity of rules. Models composed of these rules can then be joined together for given genetic circuits.

The specification for a genetic circuit can be given using the Synthetic Biology Open Language (SBOL) [1] terms. A `ComponentDefinition` entity with `subcomponents` can be used to specify a particular genetic circuit, or an abstract definition to represent large solution spaces. Models of parts ideally come from model repositories such as the Virtual Parts Repository. To specify custom rules that may not be in databases — such as for a host — explicit annotated rule-based models can be used. We devised the following protocol in order to access models from different model repositories:

- Mapping agents. The `bqbiol:is` term from the BioModels.net qualifiers is used to specify agents representing the same biological entity.
  ```
  #^ :PlacI bqbiol:is db:BBaR0010.
  ```
- Identifying the model of a part. The user specifies an abstraction, or set of templates, a list of parts and a list of non-part agent identifiers. For each part the client sends this information to the database which responds with and annotated Kappa file containing the corresponding rules.
- Communicating with repositories. The simplest option is just to use the same annotation mechanism and specify a source within the circuit for each part:
  ```
  #^ :MyCircuit rbc:sources (
  #^     rbc:part ex:prom; rbc:source <...>
  #^ ), ...
  ```
- Merging mapped agents. The Kappa files are then rewritten by a *union-find* operation driven by the `bqbiol:is` annotations followed by renaming and then simply concatenated and (naively) deduplicated.

4. CONCLUSIONS

We have briefly presented our work in progress based on SVPs and KBBF to automate the design of biological systems using RBMs. The proposed workflow is based on the definition of modular templates that can be used to instantiate rules for basic biological parts. Such rules can be annotated leading to a feasible protocol to automate their composition for the scalable modelling of synthetic systems.

5. ACKNOWLEDGMENTS

The Engineering and Physical Sciences Research Council grant EP/J02175X/1 (to A.W., V.D., G.M. and M.C.) and the European Union's Seventh Framework Programme for research, technological development and demonstration grant 320823 (to W.W.).

Additional Authors Ricardo Honorato-Zimmer (School of Informatics, University of Edinburgh)

References

[1] B. Bartley, J. Beal, K. Clancy, G. Misirli, N. Roehner, E. Oberortner, M. Pocock, M. Bissell, C. Madsen, T. Nguyen, Z. Zhang, J. H. Gennari, C. Myers, A. Wipat, and H. Sauro. Synthetic Biology Open Language (SBOL) Version 2.0.0. *J Integr Bioinform*, 12(2):272, 2015.

[2] G. Misirli, M. Cavaliere, W. Waites, M. Pocock, C. Madsen, O. Gilfellon, R. Honorato-Zimmer, P. Zuliani, V. Danos, and A. Wipat. Annotation of rule-based models with formal semantics to enable creation, analysis, reuse and visualisation. *Bioinformatics*, 2015.

[3] G. Misirli, J. Hallinan, and A. Wipat. Composable modular models for synthetic biology. *ACM J Emerg Technol Comput Syst*, 11(3):22:1–22:19, Dec. 2014.

[4] J. Wilson-Kanamori, V. Danos, T. Thomson, and R. Honorato-Zimmer. *Computational Methods in Synthetic Biology*, chapter Kappa Rule-Based Modeling in Synthetic Biology. 2015.

[1] Available at http://github.com/rbm/composition