BIOINFORMATICS ORIGINAL PAPER

Vol. 26 no. 7 2010, pages 925–931
doi:10.1093/bioinformatics/btq063

Systems biology

Advance Access publication February 16, 2010

Standard virtual biological parts: a repository of modular modeling components for synthetic biology

M. T. Cooling[1,*], V. Rouilly[2], G. Misirli[3], J. Lawson[1], T. Yu[1], J. Hallinan[3] and A. Wipat[3]

[1]Auckland Bioengineering Institute, University of Auckland, Auckland, New Zealand, [2]Department of Bioengineering, Imperial College London, London SW7 2AZ and [3]School of Computing Science, Newcastle University, Newcastle upon Tyne NE1 7RU, UK

Associate Editor: Alfonso Valencia

ABSTRACT

Motivation: Fabrication of synthetic biological systems is greatly enhanced by incorporating engineering design principles and techniques such as computer-aided design. To this end, the ongoing standardization of biological parts presents an opportunity to develop libraries of standard *virtual* parts in the form of mathematical models that can be combined to inform system design.

Results: We present an online Repository, populated with a collection of standardized models that can readily be recombined to model different biological systems using the inherent modularity support of the CellML 1.1 model exchange format. The applicability of this approach is demonstrated by modeling gold-medal winning iGEM machines.

Availability and Implementation: The Repository is available online as part of http://models.cellml.org. We hope to stimulate the worldwide community to reuse and extend the models therein, and contribute to the Repository of Standard Virtual Parts thus founded. Systems Model architecture information for the Systems Model described here, along with an additional example and a tutorial, is also available as Supplementary information. The example Systems Model from this manuscript can be found at http://models.cellml.org/workspace/bugbuster. The Template models used in the example can be found at http://models.cellml.org/workspace/SVP_Templates200906.

Contact: m.cooling@auckland.ac.nz

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on July 12, 2009; revised on February 3, 2010; accepted on February 11, 2010

1 INTRODUCTION

Since the discovery of recombinant DNA technology, scientists have manipulated living organisms in order to produce biofuels, drugs or other biomaterials. Over the years, a biotechnology industry has emerged exploiting this technology and delivered a number of successes (Carlson, 2007). However, in most cases, the development of biotechnology applications has been the product of a manually driven, trial-and-error-based approach.

In order to achieve efficient and reliable biological system fabrication, synthetic biology promotes the application of engineering principles such as abstraction, standardization and characterization to biology (Endy, 2005). These concepts have proven to be crucial in other engineering disciplines in order to mature from 'dedicated craftsmanship' to successful industrial solutions. Arguably, to date in synthetic biology, the best example of such an approach is the Registry of Standard Biological Parts (SBPs) (Peccoud *et al.*, 2008). The Registry (http://www.partsregistry.org) provides a collection of standard DNA parts (BioBricks) (Knight, 2005) that have been designed to facilitate DNA assembly. Through the International Genetically Engineered Machine (iGEM, http://www.igem.org) competition, the use of the Registry has clearly demonstrated the power of standardization in biology to stimulate innovation and creativity (Goodman, 2008).

A critical lesson learnt from other engineering disciplines is that mathematical modeling can dramatically increase the speed of the design process as well as reducing the cost of development. A 'Holy Grail' in biological modeling would be to design reliable and robust biological systems *in silico* prior to fabrication, just as aeronautic engineers design planes using their computer-aided design (CAD) tools.

CAD tools are already being developed in order to ease the process of designing synthetic biological systems (Goler *et al.*, 2008). However, they currently lack access to modular and reusable mathematical models. Accurate models of SBPs are required for the prediction of system function, but it is also crucial that mechanisms to easily compose part models into complete systems are available. Therefore, in parallel to increasing the number of parts available and characterizing them experimentally, a logical extension to the Registry would be to build a repository of modular models of SBPs to complement the physical part Registry (Rouilly *et al.*, 2007).

Here we describe the development of an online repository of Standard *Virtual* Biological Parts (SVPs)—mathematical model components describing the function of SBPs which can be downloaded, extended and recombined to aid the design, *in silico*, of synthetic biological systems.

Repositories of models are already available, such as the BioModels database (LeNovere *et al.*, 2006). However, the curated models in this database are monolithic and do not allow further composition without some modification. Previous work has already explored the importance of modularity in modeling biological systems. For example, Rodrigo et al. reports the use of a library of parts encoded in SBML (Rodrigo *et al.,* 2007). The composition of models has also been demonstrated using the modeling system ProMoT (Mirschel *et al.*, 2009) and the Modeling Description

*To whom correspondence should be addressed.

Language (Marchisio and Stelling, 2008). Both studies make valuable contributions; however, the model composition must be supported directly by the software, rather than being supported directly by the model description language.

CellML (Cuellar *et al*., 2003) is a widely used model exchange protocol supported by domain-non-specific tools, technologies and initiatives. Importantly, version 1.1 of the CellML specification includes explicit support for modularity, allowing the construction of complex models from components without modification (Cooling *et al.*, 2008). CellML models are ordinary differential equation (ODE) based, so may not be applicable when modeling very small numbers of molecules or where intrinsically stochastic processes (such as noise-induced phenomena) are considered present and important enough to model explicitly. However, ODE systems can be considered to represent the average behavior of a large class of even stochastic systems assuming that the biological reactions take place in 'well-stirred' compartments (Schilstra *et al*., 2008), and are useful for general synthetic biological system design. Computer science research has yielded promising alternative formalisms such as BlenX (Dematte *et al*., 2008), or more recently P-systems extended for modularity (Romero-Campero *et al.*, 2009) which also provide composition of modular models. However, CellML has a proven track record in representing intracellular processes in systems biology (Hunter and Borg, 2003), and already has an established framework for multi-scale modeling (Nickerson *et al*., 2006), and established tools (Garny *et al.*, 2008). These features make CellML an apt choice for the model representation format.

Since the requirements for standard models may not be known without varied experience, we advocate a 'bottom-up' approach to the development of a standard via iterative use by, and feedback from, the community. To begin the process, we present here an architecture for SVPs and an online repository to support them. We demonstrate the concepts by developing SVPs for some common SBP types, and illustrate further by combining these modular CellML models into models of synthetic biological systems—specifically, examples from gold-medal winning iGEM projects. We make all these models publically accessible online for future reuse and enhancement by the global synthetic biology community.

2 SYSTEM AND METHODS

We begin by describing the overall architecture for SVPs and how they can be combined into models of synthetic biological systems. We then describe the Repository developed to cater for the collaborative development of models that fit this architecture.

2.1 SVPs overview

Following insights on model modularization derived in previous work in Systems Biology (Cooling *et al.,* 2008), mathematical models of common SBP types from the Registry—promoters, ribosome binding sites (RBSes), RNA and protein coding sequences (CDSes)—were developed. The models were constructed with well-defined interfaces such that they are composable without modification (see Section 3 for more details).

Figure 1A shows a schematic of a simple genetic circuit designed to produce a protein 'A'. From a promoter, RNA containing a single RBS and a CDS is transcribed. These elements are SBPs as might be contained in the Registry of SBPs.

While SBPs cover genetic elements, there are potentially many other intracellular events occurring in a single cell or chassis. These include the reactions between gene products which are crucial for the genetic circuit to influence the biological system, and may also include proteins and processes that are abstracted to lumped-parameter sub-models, such as degradation of gene products or significant reactants. We represent these entities and processes under the umbrella term of 'bioenvironment' models. As shown in Figure 1A, protein 'A', produced by translation, is a reactant in the bidirectional reaction forming complex 'C' on combination with existing protein 'B'. This reaction, and the corresponding degradation reactions, do not relate to SBPs, but are nonetheless crucial to the functioning of the system. In our formulation, they are considered part of the bioenvironment, and like SBPs are modeled as SVPs.

Figure 1B shows how the genetic circuit and associated bioenvironment would be modeled with SVPs. We use three levels of modeling abstraction. The top level, denoted 'Systems Models', contains models of entire systems of interest. Systems Models link to models in the lower levels—the SVPs and Templates—aggregating in-memory copies of them to build up the

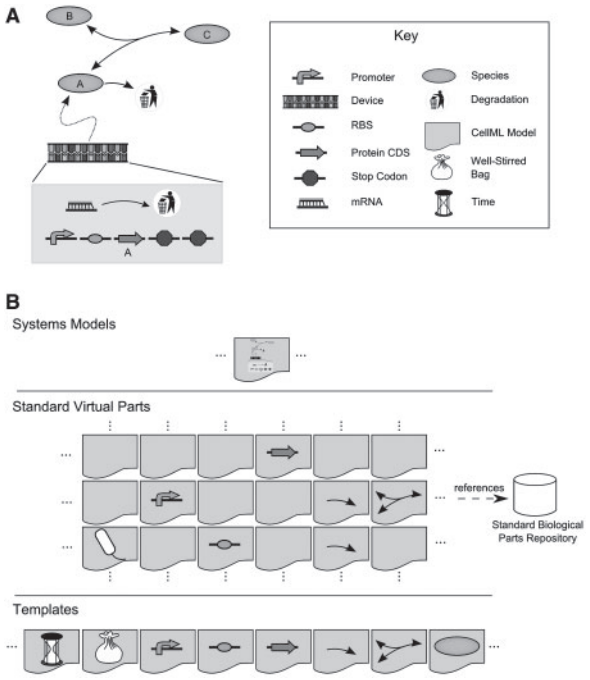

Fig. 1. (**A**) Schematic of a simple genetic circuit and associated bioenvironmental reactions. Protein A encoded by a 'composite device' forms complex C on combination with protein B. (**B**) CellML model architecture for the circuit and bioenvironment shown in (A). A Systems Model representing the complete system of interest aggregates specific (shown with symbols displayed) SVPs from a library of composable CellML models. From left to right, these include an *Escherichia coli* chassis (to give volume information), a promoter, an RBS, a protein CDS, two degradation reactions (for the RNA and for protein A) and the C complex formation reaction. Both the Systems Model and the SVPs may also aggregate components representing mathematical templates, including, from left to right across the bottom of (B): Time, Well-stirred Bag (used by *E.coli* chassis component), Constitutive Promoter (used by the promoter SVP), RBS (used by the RBS SVP), Protein CDS (used by the protein CDS SVP for species A), a unidirectional reaction (used by the degradation reaction SVPs) and a bidirectional reaction (used by the C complex formation reaction SVP). The species Template (housing an ODE for keeping track of the concentration of the species) is also used multiple times, once for each molecular species of interest in the Systems model, including the RNA. SVPs may represent SBPs or bioenvironmental elements, with the former potentially being linked to a record in the SBP Registry.

desired biological functionality. SVPs consist of mathematical formulations that model an SBP or bioenvironmental function, coupled with associated kinetic parameters. Their inputs and outputs are so designed that they can be easily reused and composed without modification into a 'Systems Model'. The lowest level is the 'Template' level. Here, a model is given for each specifically different mathematical formulation (Wimalaratne *et al*., 2009). For example, one Template is given for bidirectional mass-action kinetic reactions with two reactants and one product, another is given for constitutive promoters, a third for promoters with embedded inhibitor functions and so on.

In addition to the species and processes, time and space are added to the model through the import of a Time Template, and a set of Templates relating to cell volumes. In our formulation, models are one-dimensional but compartmentalized into volumes. We provide a 'Well-Stirred Bag' Template to reflect the concept of a three-dimensional volume, and have, as examples, derived specific volume components for particular prokaryotic cells.

These Template models make it easy to derive new SVPs, providing useful general mathematical formulations which only need parameterizing to become SBP- or bioenvironment specific. This modularity and reuse at both Template and SVP levels is made possible by the modular nature of the CellML language, as will be discussed further in Section 3.

2.2 Repository overview

All of the CellML models discussed in this article are freely accessible online at permanent, unique locations within the CellML Model Repository (http://models.cellml.org). At the core of the Repository lies the Physiome Model Repository 2 (PMR2) software. PMR2 is built upon a distributed version control system (DVCS) which stores and versions the models and associated files. A web interface layer is provided to access the data and to configure user and access controls for any particular model. This web interface can also be used to generate content pages which describe the model and also display metadata.

A synthetic biological system can be considered similar to a software program. Initially, this program is constructed *in silico* for prototyping, and then it is reconstructed *in vitro/in vivo* to create the real system. As such, it makes sense for us to make use of infrastructure designed for software development for the *in silico* stage of this process. PMR2's DVCS treats models and associated files, such as documentation, simulation data, etc. in a similar manner to software projects, providing researchers with an infrastructure for collaborative model development. Each file within PMR2 is tightly version controlled, and each version of the model is associated with a commit message intended to describe what has been changed since the previous version.

The web interface allows controlled accessibility to the models stored within. For example, a modeler may set the permissions to his model such that only his supervisors or external parties have access to it for review purposes, and then make it publically accessible once it has been reviewed. This kind of atomic access control allows researchers to collaborate on models without necessarily making them public. Because models are assigned permanent, unique URLs within the Repository, publications can be furnished with permanent links to associated model code. Once published in the CellML Model Repository, models will be made freely available for redistribution and reuse by anyone, as long as proper attribution is made.

Models are uploaded into *workspaces*, which contain the model and associated files listed in a *manifest.* Each file is given a unique URL which is version specific and by which the model components can be used by other models via CellML *imports* (see Section 3.2 for more details). Each SVP is contained in its own workspace, since it represents a defined piece of biological functionality, whose model may be revised by the community—perhaps more accurately parameterized over time, or given alternate mathematical formulations—independently from other component models. Alternatively, SVP Templates, since this set is envisaged to change less frequently, and only by the addition of new Templates, are considered to be a library of standard mathematical formulations and are thus grouped together in a single workspace. Synthetic systems are developed by modelers in their separate workspaces, using model components from SVP and the Template workspaces as needed. This architecture is shown in Figure 2.

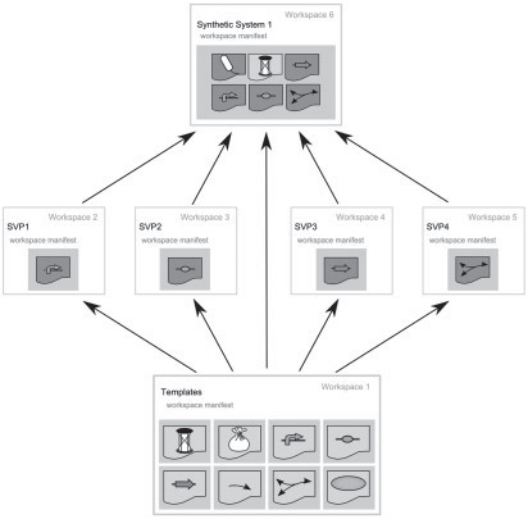

Fig. 2. The workspace architecture of SVP-based models in the Repository. Separate files are shown by the shaded boxes within workspaces. Information flows between workspaces via CellML imports (see Section 3.2) are represented as arrows. Template models are imported and re-parameterized to form SVPs with specific properties (shown by the darker shading of the CellML models), which are then combined via imports to form a model of a synthetic system. Time and Species are imported directly from the Template library. The system-level workspace contains a single CellML file, which collates its constituent modules by referencing information in other workspaces via unique URLs.

Expert curators are responsible for ensuring the coherency, reliability and accuracy of the Repository as a data resource by organizing, indexing and annotating workspaces and their constituent files. It should be noted that curation is not intended to take the place of established peer review by judging the scientific merit of a model; rather, curators ensure that a set of minimum metadata has been added to the model. This minimum set of metadata defines who created the model and when, and any relevant citation information should the model be related to a publication. Modelers can also annotate elements of the CellML model with semantic information about the biological functionality that they represent. Metadata can be added to a model at any point in its lifecycle, subject to the approval of either the model author or Repository curators.

The CellML Model Repository is under ongoing development to act as a research community hub. The web interface provides infrastructure for web-based collection and moderation of user-generated content. Researchers can work on models together, download, reuse, modify, annotate or combine models, and discuss their work, gradually building up the available SVP models for others to reuse. After submission, curators organize and may annotate models to ensure quality (Peccoud *et al.*, 2008) and to assist modelers in choosing the appropriate component for their work.

The combination of systems modularity within CellML, together with the ability to collaborate on documented, versioned libraries of modular model components in the publically accessible online Repository, provide a solid platform for rapid *in silico* prototyping of synthetic biological systems.

3 IMPLEMENTATION

We illustrate the concept of modular modeling by developing Template (and from them, SVP) models encompassing a range

of useful synthetic biological functions. As a first iteration, we have chosen to model several core SBP types from the Parts Registry, to give modelers basic genetic circuit construction functionality: namely, promoters, messenger RNA, RBSes and CDSes. Terminators did not require a specific Template or SVP in this formulation. We also develop Templates for some common bioenvironmental processes such as protein-to-protein and degradation reactions.

First we describe the mathematical formulation of these Templates, and then we discuss how these are encoded into our modeling architecture with CellML. Finally, we provide an example of how we have used our SVPs and Templates to produce a working Systems Model of a gold medal winning iGEM project.

3.1 Mathematical formulation

In order to make SVPs composable, it is important to define the mathematics simply, and with clear interfaces. The formulations for the Templates (and therefore the SVPs) are designed to make use of the popular polymerases per second (PoPs) and ribosomes per second (RiPs) units (Braff *et al.*, 2005), and to express volumes and concentrations in femtoliters and nanomolar, respectively, which are appropriate scales for unicellular systems. The formulations for the basic Templates developed in this first iteration will now be described in turn.

Our formulation focuses primarily on proteins and mRNA, and not on the background transcription/translation machinery of the cell. In contrast to other systems [such as in (Marchisio and Stelling, 2008)], the concentrations of ribosomes and polymerase are assumed not to be rate limiting, and pools of those potential species are not modeled explicitly, reducing complexity. However, components taking these concentrations into account could be formulated if desired.

3.1.1 Promoter formulation The first Template is the promoter which has the general form of:

$$J = j * c_1(V) \tag{1}$$

where j is a constant giving the rate of transcription from the promoter, measured in PoPs. $c_1(V)$ is a conversion factor scaling the PoPs to nanomolar of RNA per second (J) produced from the promoter, and is a function of the volume V(in femtoliters) of the cellular compartment where transcription takes place [for more details on the units and conversion factors used in Equations (1–5), please see the Supplementary Material]. For a constitutive promoter, j might simply equal some constant k, but j can also be expressed in more complex ways for different kinds of promoter. For example, an inducible promoter might have the formulation:

$$j = k * \frac{I^n}{(Km^n + I^n)} \tag{2}$$

where I is the concentration of an inducer species, with an associated coefficient Km, and the Hill coefficient n. Similarly, a repressible promoter might have a formulation for j thus:

$$j = k * \frac{Km^n}{(Km^n + R^n)} \tag{3}$$

where R is the concentration (in nanomolar) of some repressing transcription factor.

3.1.2 mRNA formulation mRNA is handled as an ODE tracking nanomolar of mRNA from a particular DNA molecule. It can degrade or participate in reactions like any other molecular species and so is considered part of the bioenvironment, in contrast to other schemes in which it is handled more implicitly [e.g. (Marchisio and Stelling, 2008) and (Rodrigo *et al.*, 2007)].

3.1.3 RBS formulation The RBS converts the concentration of mRNA for the device into a flux expressed in RiPs.

$$R = k * \mathrm{mRNA} * c_2(V) \tag{4}$$

where k is a rate constant for translation, in units of RiPs. This is multiplied by the concentration of appropriate mRNA (in nanomolar, available from the corresponding mRNA species concentration as in Section 3.1.2), and a conversion factor $c_2(V)$ which is a function of the volume V of the cellular compartment (in femtoliters) in which translation takes place. The rate of translation R is expressed in units of RiPs in attomoles, i.e. how many attomoles of RiPs are translating the DNA downstream of the RBS. Attomoles have been chosen to help keep value ranges such that numerical precision is likely to be maintained.

3.1.4 CDS formulation The Protein CDS formulation is designed to take the attomoles of RiPs from an upstream RBS (R) and produce a flux J of protein produced in nanomolar per second:

$$J = \frac{R}{c_2(V)} \tag{5}$$

using the conversion factor $c_2(V)$ from Equation (4). The specification of V terms in Promoter, RBS and CDS Templates allows modeling of chassis where transcription takes place in a different compartment from translation.

3.1.5 Reaction formulations Species-to-species interactions are generally modeled according to mass-action kinetics. For example, a bidirectional reaction with two reactants and one product would be:

$$J = kf * A * B - kr * C \tag{6}$$

where A, B and C are concentrations of the reactants and product, respectively, and kf and kr are forward and reverse rate constants, respectively. Concentrations are measured in nanomolar, and the flux J in nanomolar per second. A different template would exist for each combination of uni- or bidirectional reactions of different number of products and reactants (Wimalaratne *et al.*, 2009). Templates also allow reactions to be modeled in different formalisms if appropriate. For example, an enzymatic reaction might be modeled according to Michaelis–Menten kinetics:

$$J = k * E * \frac{R}{R + Km} \tag{7}$$

where k is a rate constant, E is the concentration of the enzyme, R is the concentration of the reactant, and is associated with the enzyme's Km.

Unlike other formulations where degradation is part of the model for a Standard Part, for flexibility in our formulation degradation is modeled as a bioenvironmental process acting on species such as proteins or mRNA. Degradation is implemented by a unidirectional reaction of a species using mass-action kinetics:

$$J = k * s \tag{8}$$

where k is a degradation rate constant, and s is the concentration of the species.

3.2 CellML model implementation

A Template model was created for each of the equations in the above formulation, contained in their own CellML *components* (Cuellar *et al.,* 2003), and housed in separate files.

These Template components are unparameterized, and are *encapsulated* (Cuellar *et al.*, 2003) into more specific SVP models, which house the parameter values required for the SVP to reflect a specific SBP or bioenvironmental process. This encapsulation means that an SVP can be considered as a separate model describing the behavior of the SBP or bioenvironmental process on which it is based, to be (re)used and extended independently from others.

To construct a model of a system, the modeler *imports* (Cuellar *et al.*, 2003) SVP and Template models relating to the genetic and bioenvironmental processes of interest into a Systems Model. A single Template or SVP may be imported many times, such as the species Template which is imported once for each species the modeler wishes to track in the model. SVP models do not need to be modified in order to make the necessary connections between them—instead, CellML *connection* (Cuellar *et al.*, 2003) elements are added so that a Systems Model can be thought of as a network of chained components.Following the precepts in Cooling *et al.* (2008), interface components written at model aggregation time handle the combination of flux terms contingent on molecular species, summing them as appropriate to yield an overall total flux term. The total flux term is connected back into the Species Template that was instantiated for a particular species. This architecture means that any number of flux terms can be made contingent on a species simply by adding them to the mathematics in the interface components, which thus act as malleable 'glue' for the aggregation of immutable SVP models.

SVPs can be easily reused between Systems Models, or even as multiple copies within a Systems Model (e.g. in the case of multiple copies of a CDS downstream from an RBS), simply by importing them. A tutorial describing the construction of an example Systems Model from existing SVPs, using open-source software, is provided in the Supplementary Material.

Alternative formulations of, or even new components—such as promoters with more detailed mechanisms or including ribosomal pools—can easily be implemented in CellML as new Templates, from which SVPs can then be derived. Care would need to be taken to ensure that new formulations have 'input' and 'output' variables that are compatible with each other, in order for derived SVPs to be connectable with one another. CellML's strict enforcement of consistent units reduced errors when connecting components together, helping to ensure that appropriate connections are made.

3.3 Example systems model

To highlight the composability of SVPs and their applicability to real biological problems, we demonstrate their use by modeling Newcastle University's iGEM 2008 gold-medal winning medical science project 'BugBuster', where SVPs were a foundational technology of the project. A second example, and a tutorial on constructing a simplified Systems Model from SVPs, is given in the Supplementary Material.

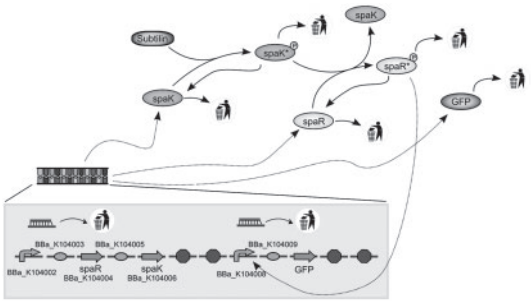

Fig. 3. Schematic of the Systems Model for BugBuster. The system consists of a single Device, containing a constitutive promoter contingent on spaK and spaR CDSes. SpaR is activated by SpaK on exposure to subtilin. Activated SpaR activates the pspaS promoter included at the 3′ end of the device. All protein and mRNA species can degrade, except for Subtilin which for the purposes of the model is assumed to have constant concentration. BioBrick numbers are shown for those SVPs with corresponding SBPs.

3.3.1 System background The 'BugBuster' peptide receiver is a proof-of-concept genetic circuit for the development of a bacterium-based pathogen-sensing system. Using *Bacillus subtilis* as a chassis, the system was designed to sense the peptide lantibiotic subtilin, an antibacterial compound normally produced by *B.subtilis* strain ATCC6633.

The receiver design was computationally constructed 'bottom-up' by assembling the SVPs necessary for encoding the two-component system that senses subtilin. A CDS for the sensor kinase (*spaK)* and its corresponding response regulator (*spaR*) were arranged in an operon downstream of a sigma-H regulated promoter. The spaR-responsive promoter (*pspaS*) was used to drive the expression of a GFP coding sequence.

3.3.2 Model structure A schematic of the Systems Model is shown in Figure 3. A sigma-H promoter is linked to both a *spaK* and a *spaR* CDS. Together these genes, each with their own RBS, comprise the phosphate-mediated signaling system required to form an activated SpaR protein. Activated SpaR is used as a transcriptional activator for the second inducible promoter spaS [which imports a template encapsulating the mathematical formulation given in Equation (2)], which is designed to be upstream of some reporter genes of interest.

The SVPs that have corresponding SBPs are linked to their Registry records (BBa_K104002-BBa_K104009) from the CellML Repository. A diagrammatic summary of the specific CellML components used for the BugBuster Systems Model is given in the Supplementary Material. The Systems Model and associated CellML files can be found online in the CellML Model Repository at http://models.cellml.org/workspace/bugbuster.

3.3.3 Model output The Systems Model was constructed in the open source CellML environment OpenCell (Garny *et al.*, 2008) and simulated. To demonstrate the functioning of the genetic circuit and associated bioenvironment, we track the virtual production of the reporter GFP produced on stimulation with various levels of subtilin, as shown in Figure 4.

The model for the virtual Systems Model was used to inform the synthesis of a DNA sequence that was integrated into the *B.subtilis*

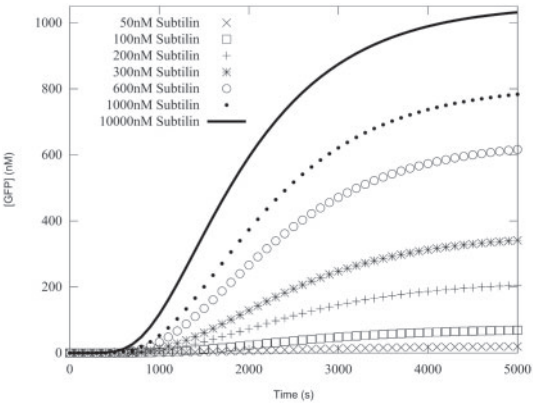

Fig. 4. Sample output of the BugBuster Systems Model. Here the timecourse for the concentration of reporter GFP is plotted for several levels of subtilin signal.

168 chromosome, allowing the strain to be used as a fluorescent-based biosensor for subtilin.

4 DISCUSSION

We have developed the first iteration of standardized modular synthetic biology models in CellML, incorporating both common genetic elements and biochemical reactions. We show how models using these formulations can be reused and extended to compose complex models during the design of synthetic biological systems, without modification of the aggregated models. We provide an online repository for collaborative development of further model components and Systems Models by the global synthetic biology modeling community. We demonstrate the applicability of this approach by modeling examples from award-winning iGEM machines.

We hope to foster the development of more Template, SVP and Systems Models by the modeling community and their storage online. As more models are contributed, and existing models have their parameters refined, the usefulness of the Repository will increase.

This work provides an initial set of designs for Templates and SVPs, and as such does have some limitations. For maximum benefit from modeling, the kinetic parameters for each SVP would have to be accurately measured. This issue is common to the entire field of synthetic biology. Some important measures such as PoPs are not directly measurable at present. However, as is already occurring in the related field of Systems Biology, we expect accurate measurement for the parameterization of system components to be conducted as a process of iterative refinement. The development of a library of promoters with quantified activity was recently demonstrated (Ellis *et al.*, 2009), and specification sheets for biological parts have also been proposed and developed (Canton *et al.*, 2008). Efforts such as these may help to standardize the SBPs that we are modeling with SVPs. Pragmatically, useful formulations may depend upon what is measurable. Until measurement issues are resolved it may be that the optimal mathematical formulation for synthetic device design purposes remains unknown with complete certainty.

Appropriate formulations may also be refined by the investigation of non-specific effects in genetic systems. Designed systems are sometimes partially limited in that we do not fully comprehend all the interactions in a living cell, particularly if the cell grows and divides. In the future, we hope to extend our models to realistically simulate the system output against the background functions of the cell on a time scale comparable with the cell's growth cycle. To facilitate this, it would be useful to develop experiment-based tests for SVPs and Systems Models built from them—does the biological system actually behave as the simulation? If not, why not? Pursuing these questions may lead to further insights both for modeling and biological science.

At the technical level, there are also opportunities for the development of more tools to enhance model construction. The encapsulation of Template models by SVPs is currently done interactively in environments such as OpenCell, but could in theory be performed automatically by software, prompting the user only for the appropriate parameterization. The connection of the SVPs in the Systems Model is done by forming interface components. These components tend to follow a standard format and could also be produced semi-automatically by software with the user specifying which SVPs to connect and the positive or negative effect of the associated fluxes on particular species. This work presents the foundation on which tools catering for these more streamlined processes could be built.

To aid computational composition and synthetic biology CAD, annotation of the models with metadata to give them semantic meaning would be of great benefit. A research project in this area is currently underway. While at present, models link, where appropriate, to the Registry of SBPs, it would also be advantageous to link from the Registry to the SVP Repository. It may also be useful to link SVP models with other repositories besides the Parts Registry. For example, molecular species such as enzymes may benefit from having specific information associated with them (such as EC number or similar), which could be implemented via each species having its own SVP with metadata linking to the appropriate repository. Such metadata would facilitate much-needed computer-assisted searching for appropriate SVPs for particular modeling projects (Beard *et al.*, 2009). In addition, currently SVPs can be combined in any order where the units of their connecting variables allow; however, aside from units checking it is up to the modeler to ensure that the connections 'make sense' with respect to the biological reality being modeled. Metadata and metadata-aware tools could potentially provide checking that SVP connections are biologically meaningful, reducing cognitive load on the modeler and structural modeling mistakes. Since, however, the supporting metadata specification is still being defined, these features have not been incorporated for this first design iteration.

The CellML language is itself under development, with proposals for version 1.2 being considered. Reusing SVP and Templates without modification has been demonstrated here through the use of flexible 'glue code', but reusing entire Systems Models, also without modification, may become possible through proposed enhancements borrowed from set theory. Another proposal concerns extending CellML simulators to enable stochastic simulation, which if implemented, would increase the applicability of SVPs still further.

Finally, the CellML Model Repository is under continuous development, hence interface enhancements and new functionality

desired by the community using SVPs, such as programmatic interfaces to the Repository, are expected to be prioritized and delivered as part of the ongoing developments in this area. As priorities become more apparent with wider use, we anticipate increasing utility from this technology.

ACKNOWLEDGEMENTS

We thank B. Canton, D. Gilbert, A. Miller, P. Nielsen, and the Glasgow 2007 and Newcastle 2008 iGEM teams for helpful suggestions and discussions.

Funding: Maurice Wilkins Centre for Molecular Biodiscovery; MTC acknowledges a University of Auckland Faculty Research Development Fund award to E. J. Crampin, and the authors acknowledge Research Councils UK funding for J.H.

Conflict of Interest: none declared.

REFERENCES

Beard,D.A. *et al.* (2009) CellML metadata standards, associated tools and repositories. *Trans. R. Soc. A.*, **367**, 1845–1867.

Braff,J.C. *et al.* (2005) Definitions and Measures of Performance for Standard Biological Parts. In International Conference Systems Biology (ICSB 2005) Poster. Available at http://hdl.handle.net/1721.1/31335 (last accessed date March 1, 2010).

Canton,B. *et al.* (2008) Refinement and standardization of synthetic biological parts and devices. *Nature Biotechnol.*, **26**, 787–793.

Carlson,R. (2007) Laying the foundations for a bio-economy. *Sys. Syn. Biol.*, **1**, 109–117.

Cooling,M.T *et al.* (2008) Modeling biological modularity with CellML. *IET Sys. Biol.*, **2**, 73–79.

Cuellar,A.A. *et al.* (2003) An Overview of CellML 1.1, a biological model description language. *Simulation*, **79**, 740–747.

Dematté,L. *et al.* (2008) The BlenX Language: a tutorial. In Bernado,M., Degano,P. and Zavattaro,G. (eds) *Formal Methods for Computational Systems Biology*, Vol. 5016. Springer Berlin, Heidelberg, pp. 313–365.

Ellis,T. *et al.* (2009) Diversity-based, model-guided construction of synthetic gene networks with predicted functions, *Nature Biol.*, **27**, 465–471.

Endy,D. (2005) Foundations for engineering biology. *Nature*, **438**, 449–453.

Garny,A. *et al.* (2008) CellML and associated tools and techniques. *Phil. Trans. R. Soc. Lon. A*, **366**, 3017–3043.

Goler,J.A. *et al.* (2008) Genetic design: rising above the sequence. *Trends Biotechnol.,* **26**, 538–544.

Goodman,C. (2008) Engineering ingenuity at iGEM. *Nature Chem. Biol.*, **4**, 13.

Hunter,P.J. and Borg,T.K. (2003) Integration from proteins to organs: the Physiome Project. *Nat. Rev. Mol. Cell Biol.*, **4**, 237–243.

Knight,T.F. (2005) Engineering novel life. *Mol. Syst. Biol.*, **1**, [Epub ahead of print, doi:10.1038/msb4100028 pmid:16729055, September 13, 2005].

Le Novere,N. *et al.* (2006) BioModels Database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems. *Nucleic Acids Res.*, **34**(Suppl. 1), D689–D691.

Marchisio,M.A. and Stelling,J. (2008) Computational design of synthetic gene circuits with composable parts. *Bioinformatics*, **24**, 1903–1910.

Mirschel,S. *et al.* (2009) PROMOT: modular modeling for systems biology. *Bioinformatics*, **25**, 687–689.

Nickerson,D.P. *et al.* (2006) Computational multiscale modeling in the IUPS Physiome Project: modeling cardiac electromechanics. *IBM J. Res. Dev.*, **50**, 617–630.

Peccoud,J. *et al.* (2008) Targeted development of registries of biological parts. *PLoS ONE*, **3**, e2671.

Rodrigo,G. *et al.* (2007) Asmparts: assembly of biological model parts. *Sys. Syn. Biol.*, **1**, 167–170.

Romero-Campero,F.J. *et al.* (2009) Modular assembly of cell systems biology models using P systems. *Int. J. Found. Comp. Sci.*, **20**, 427–442.

Rouilly,V. *et al.* (2007) Registry of BioBricks models using CellML. *BMC Sys. Biol.*, **1**(Suppl. 1), P79.

Schilstra,M.J. *et al.* (2008) Methods for simulating the dynamics of complex biological processes. In Correia,D.J.J. and Detrich,H.W. (eds) *Biophysical Tools for Biologists. Volume One: In Vitro Techniques.* Vol. 84. Academic Press, San Diego, pp. 807–842.

Wimalaratne,S.M. *et al.* (2009) Facilitating modularity and reuse: guidelines for structuring CellML 1.1 models by isolating common biophysical concepts. *Exp. Physiol.*, **94**, 472–485.