Systems biology

Computational design of synthetic gene circuits with composable parts

M.A. Marchisio* and J. Stelling

Department of Biosystems Science and Engineering and Swiss Institute of Bioinformatics, ETH Zurich, 8092 Zurich, Switzerland

ABSTRACT

Motivation: In principle, novel genetic circuits can be engineered using standard parts with well-understood functionalities. However, no model based on the simple composition of these parts has become a standard, mainly because it is difficult to define signal exchanges between biological units as unambiguously as in electrical engineering. Corresponding concepts and computational tools for easy circuit design in biology are missing.

Results: Taking inspiration from (and slightly modifying) ideas in the 'MIT Registry of Standard Biological Parts', we developed a method for the design of genetic circuits with composable parts. Gene expression requires four kinds of signal carriers: RNA polymerases, ribosomes, transcription factors and environmental 'messages' (inducers or corepressors). The flux of each of these types of molecules is a quantifiable biological signal exchanged between parts. Here, each part is modeled independently by the ordinary differential equations (ODE) formalism and integrated into the software ProMoT (Process Modeling Tool). In this way, we realized a 'drag and drop' tool, where genetic circuits are built just by placing biological parts on a canvas and by connecting them through 'wires' that enable flow of signal carriers, as it happens in electrical engineering. Our simulations of well-known synthetic circuits agree well with published computational and experimental results.

Availability: The code is available on request from the authors.

Contact: mario.marchisio@bsse.ethz.ch

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

1 INTRODUCTION

Synthetic biology deals with the purpose-driven design and implementation of novel biological functions such as engineered genetic circuits. The field has spurred the interest of many research groups that made efforts to build biological devices by means of well-known genetic structures. Application areas can be found in fields from environmental sciences to medicine and diagnostics (Sayut *et al.*, 2007) and several remarkable engineered biological circuits have been realized (for reviews see, for instance, Andrianantoandro *et al.*, 2006; Benner and Sismour, 2005; Drubin *et al.*, 2007; Hasty *et al.*, 2002).

In general, biological circuits can be constructed from a handful of basic parts. To completely implement a (basic) transcription unit, for instance, one needs promoters, ribosome binding sites (RBS), protein coding regions and terminators. Other parts encoding for spacers or for particular stem-loop RNAs can fine-tune gene regulation, or allow more degrees of freedom in controlling gene expression. An exhaustive repository of synthetic parts is the 'MIT Registry of Standard Biological Parts' (http://partsregistry.org/), a reference point for current research in synthetic biology. It contains not only basic parts but also more complex devices accompanied by some relevant information about their structures and functions. However, to build devices from basic parts efficiently, the complexity of the reactions as well as the variety of the molecules involved make it very difficult to accurately predict the behavior even of simpler biological devices.

For transcription networks, nevertheless, a qualitative depiction of their response to stimuli and an estimation of produced proteins can be obtained by employing mathematical modeling frameworks such as the ordinary differential equation (ODE) formalism (Alon, 2006). In a rough approximation, mRNA transcription and translation are treated as a single step. Control of gene expression—which may involve cooperativity, competition between transcription factors and processing of environmental signals—can be described by an appropriate choice of Michaelis–Menten type reaction kinetics and coefficients. Protein production then depends on the activity of the corresponding transcription units, the translation rates and the proteins' (constant) decay rates.

A more detailed view, which allows an estimation of the time delay between transcription and translation, demands to separate these two events by explicitly modeling the mRNA dynamics (Klipp *et al.*, 2005). This more accurate description of the system dynamics increases the number of model parameters. As many of the associated kinetic parameters have not been unequivocally determined yet, this adds uncertainty to the prediction of the system behavior (Tomshine and Kaznessis, 2006). A more realistic insight into a biological network can be obtained by treating it as a stochastic system. However, under precise conditions (as stated in Samoilov and Arkin, 2006) the ODE formalism is the continuous–deterministic limit of a discrete–stochastic system description. Hence, trade-offs between model accuracy and efforts needed for establishing the model are important considerations for synthetic biology, and generalizable frameworks are needed.

*To whom correspondence should be addressed.

Moreover, independent of the representation, a mathematical model of a biological circuit can hardly be based directly on the Registry's basic parts. Currently, the parts are not *composable*, that is, they do not share the same types of input and output. In circuit design for electrical engineering, parts such as batteries, resistors and solenoids can be assembled in many different ways because they all exchange information via the common 'currency' of a flux of charged particles that can be measured easily. This suggests that the implementation of biological circuits requires an exchange of information by fluxes of common signal carriers as well. Such a framework would enable us to represent biological networks more intuitively by separated modules (the parts) connected by wires. However, there exists no commonly accepted biological counterpart of the electric current yet. Mathematical models of genetic circuits based on the Registry parts are, in general, treated as unique structures that show no modularity. Hence, we urgently need concepts and tools for systematic computational design from re-usable parts as in other engineering disciplines as well as a database of the Registry part models, as pointed out in Rouilly *et al.* (2007).

A corresponding concept can start from the realization that the expression of every gene needs RNA polymerases (for transcription) and ribosomes (for translation). The 'Abstraction Hierarchy' pages of the Registry propose the flux of these signal carriers as units for characterizing the information exchange between parts. Polymerases per second (*PoPS*) and ribosomes per second (*RiPS*) could allow parts to communicate to each other just by means of a 'current' of polymerases and ribosomes (Endy *et al.*, 2005). This picture, however, does not seem sufficient to describe all information exchanges even in simple engineered gene circuits. We argue that other signal carriers like transcription factors and environmental 'messages' should be explicitly introduced and not indirectly estimated by means of *PoPS* and *RiPS*. This permits modeling the reactions involved in protein synthesis more precisely without loss of parts composability. Furthermore, we introduce *pools* of proteins and small molecules. They are connected to every transcription unit and distribute 'free' signal carriers correctly among the parts according to their affinities. These pools allow *scalability*; the system response to different signal carrier concentrations is particularly important in complex network simulations as shown in Supplementary Material.

Several software tools have functionalities analogous to those for electrical circuit design, but none of them combines ease-of-use, parts composability, and detailed modular modeling approaches. BioJADE (Goler, 2004) as one of the first tools provides a graphical user interface (GUI) to place, connect and even modify Registry parts, but it considers only one kind of signal carrier (RNA polymerases) and, hence, very simplified models of gene expression dynamics. CellDesigner (Funahashi *et al.*, 2003) has similar capabilities for graphical circuit composition. However, parts modularity and, consequently, circuit representation do not appear detailed enough because the Hill functions (Kærn and Weiss, 2006) employed assume *quasi-equilibrium* conditions. Total RNA polymerase and ribosome concentrations are *de facto* ignored and signal carriers are absent—this prevents precise simulations of large engineered networks. A very recent tool, Asmparts (Rodrigo, G. *et al.*, submitted for publication in *Systems and Synthetic Biology*), applies the same Hill formalism for the Registry parts, providing SBML code for each of them. Parts can be assembled from the command line (but not a GUI) into a unique circuit file. *PoPS* and *RiPS* based on the Hill functions are formally derived, but they

are not explicitly computed. Transcription factors (but not their fluxes, or environmental signals) are included as promoter input; however, we think that it is necessary to model each part in more detail to better depict the signal carrier dynamics (see Section 3). An opposite approach is realized in TABASCO (Kosuri *et al.*, 2007), which emphasizes the action of RNA polymerases and ribosomes at single base-pair resolution. The tool permits to estimate gene expression with high precision and it is a powerful instrument for circuit simulations. Nevertheless, it lacks part modularity, which limits the use for circuit design.

Here, we present a new framework for the design of synthetic circuits where each part is modeled independently following the ODE formalism. This results in a set of composable parts that communicate by fluxes of signal carriers, whose overall amount is constantly updated inside their corresponding pools. Basic parts, moreover, can be put together to build *composite devices* such as protein generators, reporters and inverters. Again, these are composable and able to communicate both with parts and pools. We have implemented the corresponding models into ProMoT (Process Modeling Tool), a software for the object-oriented and modular composition of models for dynamic processes (Ginkel *et al.*, 2003). This tool allows one to design a synthetic biological circuit easily, just by displaying its parts on the screen and by connecting them by 'wires' for the signal carrier exchange. We test the concept by representing some of the most well-known synthetic circuits: both their qualitative and quantitative behaviors can be fairly reproduced.

2 APPROACH

For modeling general genetic circuits, we can start by considering a simple one-step cascade circuit (Fig. 1). This small network needs at least four different kinds of signal carriers, namely RNA polymerases, ribosomes, transcription factors and environmental signals. To each of these (classes of) molecules, we can associate a different unit to quantify its flux along the parts: the already mentioned *PoPS* and *RiPS* as well as the factor per second (*FaPS*) and the Signal Per Second (*SiPS*). Following the Registry, *PoPS* can be defined as the quantity of RNA polymerases that passes a defined point on the DNA per time with unit molars per second (M/s). An analogous definition is valid for *RiPS*. *FaPS* are the quantity of transcription factors (activators or repressors) produced per second inside their corresponding coding regions. *SiPS* represent the amount of environmental signals (inducers or corepressors) that enters the cell per time unit. Thus, every flux is just a derivative of a concentration with respect to time so that it is straightforward to integrate it into an ODE-based model.

Every part is, hence, able to calculate one or more of these basic fluxes and to communicate them to the connected parts whose functioning is affected by this information. Note that composable parts do not need to exchange all four types of molecules, but just the ones they are interested in. In other words, parts composability does not mean that the parts themselves can be put together randomly inside a circuit, but they have to obey some biological constraints. For instance, a functional protein coding region cannot be connected directly to a promoter because it has to be preceded by a ribosome binding site to be translated. The composition of a synthetic circuit can be validated with parsing algorithms (Cai *et al.*, 2007). Furthermore, the total quantities of free signal carriers have to be updated continuously and must be visible to every

interested part. Hence, every promoter inside a circuit has to be connected to a polymerase pool. Additional connections to one or more signal and transcription factor pools depend on the nature of the promoter. Analogously, ribosome binding sites as well as protein coding regions must be connected to the ribosome pool. The coding regions, furthermore, access transcription factor pools whenever transcriptional repressors or activators are their products. Terminators, on the contrary, interact just with the polymerase pool, sending a flux of molecules that have finished the transcription of a gene. This picture implies that, for instance, the promoter is not a simple *PoPS* 'battery' that creates a signal *de novo*. The signal produced inside a promoter is regulated by the total pool of free polymerases and by the action of transcription factors, inducers and corepressors. All promoters constantly exchange information with the polymerase pool, leading to an interconnected network of genes.

The above units that characterize the exchange of signal carriers between parts are difficult to measure experimentally, for instance, because the molecules move discontinuously along a nucleotide chain or inside the cell. In our view, the strength of the concept is not to try to estimate the behavior of a given device just in terms of *PoPS* as inputs and outputs. Common signal carrier fluxes are most useful in providing abstractions that make parts composable and, consequently, facilitate design and simulation of biological circuits. The circuits' behavior will still be described in terms of protein produced per time or as a function of inducer/corepressor concentrations, for instance. Note that different networks might require other basic parts, which can imply the construction of new pools and the exchange of other signal carriers. This applies, for instance, to non-coding RNA parts (see Supplementary Material).

3 METHODS

Even though, like in the most traditional approach, we use the ODE formalism, the novelty of our method lies in the *composability* of parts. The parts are modeled independently and can be interconnected through the exchange of common signal carriers whose fluxes are expressed in the units explained in the previous section. In the following, we describe in detail the parts necessary to build a one-step cascade (Fig. 1). All variables represent concentrations (in M) except for the fluxes. Quantities in square brackets refer to biochemical complexes.

3.1 The *basic* promoter

The first transcription unit of a one-step-cascade network encodes for a transcription factor, namely a repressor. Its expression is supposed to be independent of any other transcription factors in the cell, so that it can be estimated by using an (unrealistic) *basic* promoter without operators. The promoter interacts just with RNA polymerases. We assume an initial condition where all the RNA polymerases are free (Pol^{free}) and stored inside their pool. They are seen by free promoters (P) and can bind to them following a Michaelis–Menten schema

$$Pol^{free} + P \overset{(k_1, k_{-1})}{\rightleftharpoons} [PPol] \xrightarrow{k_2} P + Pol^{cl} \qquad (1)$$
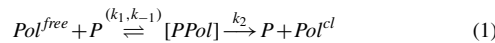
where $[PPol]$ represents the *initiation* complex formed by a polymerase and a promoter; Pol^{cl} refers to the RNA polymerase in the *clearance* phase during which transcription initiation is completed. The kinetic constants k_1 and k_{-1} are related to the formation and the dissociation of the $[PPol]$ complex, whereas k_2 is the transcription initiation frequency.

As the total promoter concentration (P_T) is fixed and given by the sum of free and occupied promoters: $P_T = P + [PPol]$, the state of the promoter
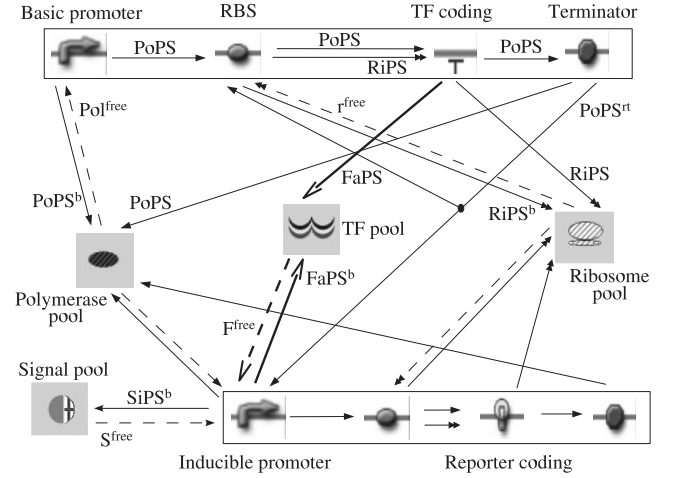

Fig. 1. One-step cascade network. The first transcription unit (box on the top) encodes for a repressor for the promoter leading the second transcription unit (box on the bottom) that produces a reporter protein. Environmental signals entering the inducible promoter can inactivate repressors and turn on protein synthesis. Solid and dashed arrows represent the fluxes and the available concentrations of the four different signal carriers, respectively (simple arrows: *PoPS* and *Pol^free*; double arrows: *RiPS* and *r^free*; line arrows: *FaPS* and *F^free*; concave arrows: *SiPS* and *S^free*). The transcription units are associated with two different composite devices: a *protein generator* and a *reporter*.

is captured by the $[PPol]$ amount, which follows the differential equation

$$\frac{d[PPol]}{dt} = k_1 Pol^{free} P_T - (k_1 Pol^{free} + k_{-1} + k_2)[PPol]. \qquad (2)$$

Two different polymerase fluxes leave the promoter part: one is a negative 'balance' flux ($PoPS^b$) sent to the polymerase pool, which corresponds to the variation of free polymerase concentrations due to the interaction with the promoter

$$PoPS^b = -k_1 Pol^{free} P_T + (k_1 Pol^{free} + k_{-1})[PPol] \qquad (3)$$

and the other is the outgoing flux ($PoPS^{out}$) directed to the next part in the transcription unit (in this case an RBS)

$$PoPS^{out} = k_2 [PPol]. \qquad (4)$$

From Equation (1), it is apparent that $PoPS^{out}$ is nothing else than the time derivative of the polymerase concentration in the clearance phase, Pol^{cl}.

3.2 The RBS

The polymerases per second leaving the promoter [see Equation (4)] represent the input signal for the RBS ($PoPS^{in}$). All the incoming RNA polymerases are supposed to bind, at the beginning of this region, to a site that we will call *B*. This gives rise to a new complex ($[PolB]$) before starting *mRNA* transcription with a constant elongation velocity:

$$PoPS^{in} \Longrightarrow [PolB] \xrightarrow{k_{el}^{RBS}} Pol^{el} + B. \qquad (5)$$
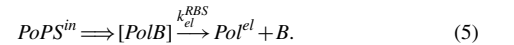
Note that, in principle, this model does not force RNA polymerase to have the same velocity inside different parts. The $[PolB]$ complex is an artifact to model passage of the RNA polymerases from the clearance to the elongation phase (Pol^{el}). The rate of Pol^{el} formation (k_{el}^{RBS}) is given by the ratio of the elongation velocity (v_{el}) to the RBS length (l_{RBS}): $k_{el}^{RBS} = v_{el}/l_{RBS}$.

Equation (5) allows us to estimate the amount of $PoPS^{out}$ leaving the RBS part. It is the time derivative of Pol^{el}, which corresponds to

$$PoPS^{out} = k_{el}^{RBS} [PolB], \qquad (6)$$

so that the time derivative of $[PolB]$ corresponds to the algebraic sum of the incoming and the outgoing polymerase fluxes

$$\frac{\mathrm{d}[PolB]}{\mathrm{d}t} = PoPS^{in} - PoPS^{out}. \tag{7}$$

While the mRNA leader region is transcribed, we assume that free ribosomes (r^{free}) leave their pool and bind to a binding site (b) on the mRNA forming the $[rb]$ complex with Michaelis–Menten type kinetics

$$r^{free} + b \overset{(k_{1r},k_{-1r})}{\rightleftharpoons} [rb] \overset{k_{2r}}{\to} r^{cl} + b, \tag{8}$$

where r^{cl} represents the ribosome concentration during the leader clearance phase. Just after clearing the RBS completely, ribosomes can bind to the start codon (AUG) located in the next part, the protein coding region, and start protein synthesis. Note that, similar to the $[PolB]$ complex, the $[rb]$ complex does not really exist. Ribosomes bind directly to the AUG codon, which belongs to the RBS. Nevertheless, the $[rb]$ complex is instrumental in estimating the initial value of *RiPS* directly from the translation initiation frequency (k_{2r}), which corresponds to the inverse of the RBS clearance time. Whereas the promoter generates a *PoPS* signal, the RBS is the *RiPS* signal generator.

From Equation (8), we can derive the time derivative of r^{cl}, which corresponds to the ribosome flux that leaves the RBS

$$RiPS^{out} = k_{2r}[rb]. \tag{9}$$

The *free* mRNA concentration (b) depends on the polymerase flux, the interaction with ribosomes [Equation (8)], and the mRNA degradation constant (k_d):

$$\frac{\mathrm{d}b}{\mathrm{d}t} = k_{el}^{RBS}[PolB] - k_{1r}r^{free}b \tag{10}$$

$$+ (k_{-1r} + k_{2r})[rb] - k_d b + PoPS^{rt}.$$

Here, furthermore, we included a term due to transcriptional *readthrough* ($PoPS^{rt}$), that is, the polymerase flux that passes the terminator and enters the next promoter. Assuming that $[rb]$ decays with the same degradation constant as b (producing free ribosomes), the time dependency of the $[rb]$ complex concentration obeys the differential equation

$$\frac{\mathrm{d}[rb]}{\mathrm{d}t} = k_{1r}r^{free}b - (k_{-1r} + k_{2r} + k_d)[rb] \tag{11}$$

and the ribosomes per second exchanged with the ribosome pool ($RiPS^b$) are given by

$$RiPS^b = -k_{1r}r^{free}b + (k_{-1r} + k_d)[rb]. \tag{12}$$

Note that $RiPS^b$ is a negative flux of ribosomes directed from the RBS to the ribosome pool.

Hence, the RBS part handles two different signal carriers: RNA polymerases and ribosomes. This permits to completely evaluate the total mRNA concentration in the system [Equation (10)], although the transcription process continues in the protein coding part, just by extending the mRNA chains here initiated.

3.3 The protein coding part

Both the polymerase and the ribosome flux produced inside the RBS go into a protein coding part representing a gene. Incoming RNA polymerases are supposed to form a new complex by binding to the *start point* position on the DNA (A) before going on transcribing the mRNA with the same average elongation velocity as inside the RBS

$$PoPS^{in} \Longrightarrow [PolA] \overset{k_{el}^{PC}}{\to} Pol^{el} + A. \tag{13}$$

Macroscopically, the transcription rate k_{el}^{PC} is much smaller than inside the RBS (k_{el}^{RBS}) because it is inversely proportional to the length of the gene. As

for the RBS, the outgoing polymerase flux, directed this time to a terminator, is given by the expression

$$PoPS^{out} = k_{el}^{PC}[PolA] \tag{14}$$

and the time derivative of the $[PolA]$ complex follows the equation

$$\frac{\mathrm{d}[PolA]}{\mathrm{d}t} = PoPS^{in} - PoPS^{out}. \tag{15}$$

A flux of ribosomes also enters this part. Ribosomes bind to the mRNA at the start codon (AUG), forming a complex indicated as $[ra]$. They translate mRNA until they encounter the stop codon (XXU), bind to it, and form another complex, $[ru]$. At this point, ribosomes are freed again and go back to their pool. Hence, whereas we have, as inside the other parts, just two polymerase fluxes ($PoPS^{in}$ and $PoPS^{out}$), one more flux is required to describe the ribosome dynamics. It is associated with the internal flux of ribosomes between the complexes $[ra]$ and $[ru]$ ($RiPS^{PC}$)

$$RiPS^{in} \Longrightarrow [ra] \overset{k_{el}^r}{\Longrightarrow} RiPS^{PC}, \tag{16}$$

$$RiPS^{PC} \Longrightarrow [ru] \overset{\zeta_r}{\to} r^{free} + XXU. \tag{17}$$

The ribosome elongation rate (k_{el}^r) is the ratio of the average translational elongation velocity (v_{el}^r) to the gene length. From Equation (16), we have

$$RiPS^{PC} = k_{el}^r[ra] \tag{18}$$

whereas the outgoing flux of free ribosomes toward their pool can be obtained from Equation (17)

$$RiPS^{out} = \zeta_r[ru]. \tag{19}$$

Here, ζ_r is the ribosome dissociation constant; it depends on the particular release factors involved in the translation termination process. The variation of $[ra]$ and $[ru]$ with respect to time is given by

$$\frac{\mathrm{d}[ra]}{\mathrm{d}t} = RiPS^{in} - RiPS^{PC} \text{ and} \tag{20}$$

$$\frac{\mathrm{d}[ru]}{\mathrm{d}t} = RiPS^{PC} - RiPS^{out} \tag{21}$$

whereas the total amount of synthesized protein (z) can be obtained by

$$\frac{\mathrm{d}z}{\mathrm{d}t} = RiPS^{PC} - k_D z \tag{22}$$

with the protein decay constant k_D. When z is a repressor or an activator, the coding part communicates with the appropriate transcription factor pool by a flux of proteins ($FaPS^{out}$):

$$FaPS^{out} = RiPS^{PC}. \tag{23}$$

Note that Equation (23) has no degradation term because it is calculated only once inside the pool.

3.4 The terminator

The RNA polymerases leaving the protein coding region enter the terminator (T) where they form a new complex ($[PolT]$) before becoming free and flowing again to their pool ($PoPS^{out}$):

$$PoPS^{in} \Longrightarrow [PolT], \tag{24}$$

$$[PolT] \overset{\zeta}{\to} \tag{25}$$

$$Pol^{free} + T, PoPS^{out} = \zeta[PolT]. \tag{26}$$

Depending on the terminator's efficiency (e), however, a fraction of the polymerases engaged in the $[PolT]$ complex may continue processing the next transcription unit. This generates a readthrough flux ($PoPS^{rt}$)

$$[PolT] \overset{\eta}{\to} Pol^{rt} + T, \tag{27}$$

$$PoPS^{rt} = \eta[PolT]. \tag{28}$$

The dissociation constant ζ and the readthrough constant η provide the terminator efficiency as: $e = \zeta/(\zeta + \eta)$. The time derivative of the $[PolT]$

complex corresponds to the sum of the incoming and outgoing polymerase fluxes

$$\frac{\mathrm{d}[PolT]}{\mathrm{d}t} = PoPS^{in} - PoPS^{out} - PoPS^{rt}. \tag{29}$$

This part usually terminates a transcription unit, although in many cases it can be followed by another terminator to reduce the readthrough effect.

3.5 The one-operator promoter

In the single-step cascade, the reporter's transcription unit is lead by an inducible promoter with one operator that can host repressors from the transcription factor pool. Inducers from the signal pool can enter the promoter part, bind to the repressors, and inactivate them. This increases the probability that RNA polymerases transcribe the reporter. Instead of the variable P used for the basic promoter, it is convenient to introduce a new variable O for the operator state. It can take two values: *free* (O^f), available to the RNA polymerase and *taken* (O^t), occupied by a repressor.

The Michaelis–Menten reaction of Equation (1) then becomes

$$Pol^{free} + O^f \underset{(k_1, k_{-1})}{\rightleftharpoons} [PolO^f] \xrightarrow{k_2} O^f + Pol^{cl}. \tag{30}$$
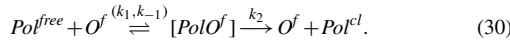
Repressors arrive at the promoter in their active form (R^a) and can interact directly with free operators and inducers (I)

$$R^a + O^f \underset{(\alpha, \beta)}{\rightleftharpoons} O^t \ ; \ nI + R^a \underset{(\lambda, \mu)}{\rightleftharpoons} R^i \tag{31}$$
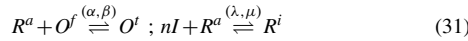
where n is the number of inducer molecules that cooperatively turn an active repressor into the inactive form (R^i). The value of n is lower than or equal to the number of repressor subunits. Inducers can also release repressors bound to the operators

$$nI + O^t \xrightarrow{\gamma} R^i + O^f. \tag{32}$$

We assume that repressors always decay with rate constant k_D, independent of their binding state. However, inside the promoter we calculate only R^i and O^t degradation explicitly, whereas R^a are supposed to decay inside their pool. Hence, we have

$$R^i \xrightarrow{k_D} nI \ ; \ O^t \xrightarrow{k_D} O^f. \tag{33}$$
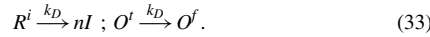
This promoter handles three different signal carriers: transcription factors (repressors), environmental signals (inducers) and RNA polymerases. It exchanges up to five different fluxes of these molecules. The fluxes of transcription factors and environmental signals are negative and directed toward the corresponding pools. $FaPS^b$ represents the time derivative of the free active repressor (R^a) due to the reactions in Equation (31)

$$FaPS^b = -\alpha R^a O^f + \beta O^t - \lambda R^a I^n + \mu R^i, \tag{34}$$

whereas $SiPS^b$ reflects the time variation of the total free inducer concentration caused by Equations (31–33)

$$SiPS^b = n(-\lambda R^a I^n + \mu R^i - \gamma O^t I^n + k_D R^i). \tag{35}$$

Note that the free promoter/operator concentration O^f in Equation (34) can be derived from: $O_T = O^f + [PolO^f] + O^t$, where O_T is the total promoter concentration. RNA polymerase is involved in four different fluxes, three of them exchanged with external parts. The promoter is connected to the terminator of the first transcription unit, from which it receives a readthrough signal ($PoPS^{rt}$). We can assume that these polymerases encounter only free promoters, which results in an increment of $[PolO^f]$

$$PoPS^{rt} \Longrightarrow [PolO^f]. \tag{36}$$

Furthermore, RNA polymerases can bind weakly to occupied promoters O^t, yielding a *leakage* flux ($PoPS^{lk}$) according to:

$$PoPS^{lk} = k_2^{lk} O^t \tag{37}$$

where the *basal* transcription initiation frequency k_2^{lk} is generally much lower than k_2. Leakage contributes to the outgoing polymerase flux and to the

negative flux back to the polymerase pool, respectively:

$$PoPS^{out} = k_2[PPol] + PoPS^{lk} \tag{38}$$

$$PoPS^b = -k_1 Pol^{free} O^f + k_{-1}[PPol] - PoPS^{lk}. \tag{39}$$

Conversely, polymerase readthrough enters the $[PolO^f]$ state equation

$$\frac{\mathrm{d}[PolO^f]}{\mathrm{d}t} = k_1 Pol^{free} O^f - (k_{-1} + k_2)[PolO^f] + PoPS^{rt} \tag{40}$$

[compare to the basic promoter, Equation (2)]. To complete the one-operator promoter description, we need two more ODEs for O^t and R^i, respectively:

$$\frac{\mathrm{d}O^t}{\mathrm{d}t} = \alpha R^a O^f - (\beta + \gamma I^n + k_D)O^t, \tag{41}$$

$$\frac{\mathrm{d}R^i}{\mathrm{d}t} = \lambda R^a I^n - \mu R^i + \gamma O^t I^n - k_D R^i. \tag{42}$$

3.6 The signal carrier pools

All pools in our model are new parts and not yet included in the Registry. The polymerase pool stores all free RNA polymerase molecules; it is connected to every promoter and terminator in a circuit. The total amount of free polymerases, constantly visible to the promoters, is calculated by the negative $PoPS^b$ flux from the promoter parts [Equations (3, 39)] and the $PoPS^{in}$ flux from the terminator parts [Equation (26)]

$$\frac{\mathrm{d}Pol^{free}}{\mathrm{d}t} = \sum_{i=1}^{N} (PoPS_i^b + PoPS_i^{in}), \tag{43}$$

where N is the number of transcription units in the network. The ribosome pool functions identically, but it is connected to the RBS and the protein coding parts. Hence, the free ribosome concentration is given by

$$\frac{\mathrm{d}r^{free}}{\mathrm{d}t} = \sum_{i=1}^{N} (RiPS_i^b + RiPS_i^{in}) \tag{44}$$

where $RiPS^b$ is a negative flux [calculated in Equation (12)] and $RiPS^{in}$ coincides with the quantity in Equation (19). The RBS part has constant access to the value of r^{free} updated through Equation (44).

In our example network, repressors are produced by the transcription factor coding part of the first gene. Repressor monomers (F^m) are sent to the transcription factor pool [Equation (23)]

$$FaPS^{in} \Longrightarrow F^m, \tag{45}$$

where they may dimerize (or: form higher order complexes) to enable interactions with operators and inducers

$$2F^m \underset{(\delta, \epsilon)}{\rightleftharpoons} F^{free}. \tag{46}$$

Here, δ and ϵ are the complex association and dissociation rate constants, respectively. Free dimers (F^{free}) coincide with the active repressors (R^a) that regulate the one-operator promoter. This results in a 'balance', negative $FaPS^b$ flux in Equation (34) from the promoter to the pool

$$FaPS^b \Longrightarrow F^{free}. \tag{47}$$

Free dimers and monomers are supposed to decay with identical rates (k_D)

$$F^m \xrightarrow{k_D} \ ; F^{free} \xrightarrow{k_D}. \tag{48}$$

Again, the role of the transcription factor pool is to update the total concentrations of free, active transcription factors by

$$\frac{\mathrm{d}F^{free}}{\mathrm{d}t} = \delta F^{m2} - \epsilon F^{free} - k_D F^{free} + FaPS^b \tag{49}$$

where F^m obeys the following differential equation

$$\frac{\mathrm{d}F^m}{\mathrm{d}t} = 2(-\delta F^{m2} + \epsilon F^{free}) - k_D F^m + FaPS^{in}. \tag{50}$$

Furthermore, as mentioned above, we use a model structure where free factors decay inside the pool, whereas factors bound to n signals are degraded inside the promoter part.

For free signals (inducers, S^{free}), we assume constant production

$$\xrightarrow{k} S^{free} \tag{51}$$

in their pool with production rate constant k. Free inducers can bind to the promoter and deactivate repressors as stated in Equations (31, 32). This creates a negative flux ($SiPS^b$) from the promoter to the pool [Equation (35)]

$$SiPS^b \Longrightarrow S^{free}. \tag{52}$$
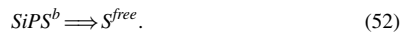
Free signal degradation takes place inside the pool, with a decay rate (k_{D_s}) that is small compared to the one of the associated transcription factor

$$S^{free} \xrightarrow{k_{D_s}}. \tag{53}$$

Clearly, the signal pool is needed to calculate the total concentration of free signal at each time step and to communicate it to the connected promoter(s)

$$\frac{\mathrm{d}S^{free}}{\mathrm{d}t} = k - k_{D_s} S^{free} + SiPS^b. \tag{54}$$

4 IMPLEMENTATION

As briefly mentioned in Section 1, ProMoT is a systems modeling and design tool that permits to reproduce the dynamics of a biochemical system through *modules*. Each module represents a system subunit, characterized by a certain degree of complexity and autonomy. It can estimate the temporal evolution of some general quantities and communicate it to other modules. Each of our biological parts (the basic ones as well as the composite devices) is associated with an appropriate module. Therefore, we encoded each part in MDL (Modeling Description Language), the object-oriented Lisp-based programming language of ProMoT. ProMoT, furthermore, provides the user with a Java GUI where one can just drag and drop the parts needed, without caring of their content, and then connect them through 'wires', as it is done in many electrical engineering tools [see, for instance, SPICE (Nagel and Pederson, 1973)].

More specifically, the MDL code of the parts needs the definition of *variables*, *terminals* and *equations*. Variables represent all time-varying quantities (state variables for ODE systems) as well as the constant parameters. Terminals are the interfaces between parts and contain all the variables necessary for information exchange. Equations can be simple algebraic relations or ODEs. The one-operator promoter for instance, has five terminals. One terminal connects to a terminal of the polymerase pool to get the amount of available free RNA polymerases (Pol^{free}) and to communicate the value of $PoPS^b$. A second terminal sends the produced $PoPS^{out}$ to another part (an RBS for instance). The last terminal associated with RNA polymerase will receive $PoPS^{rt}$ from an adjacent terminator. Two more terminals connect the promoter to the transcription factor and to the signal pool. These terminals receive the total amounts of free molecules (F^{free} and S^{free}) and send the values of $FaPS^b$ and $SiPS^b$, respectively. Note that whenever a flux is absent, the corresponding terminal can be blocked with a *plug* that simply gets or sends a null flux.

Basic parts can also be encapsulated into higher order modules to construct composite devices. They can then be put inside a circuit and connected to other simple or complex parts. For instance, an entire transcription unit may be embedded into a *protein generator* device or a *reporter* device, depending on the kind of protein synthesized (Fig. 1). The design and representation of an intricate network can, hence, be drastically simplified just by
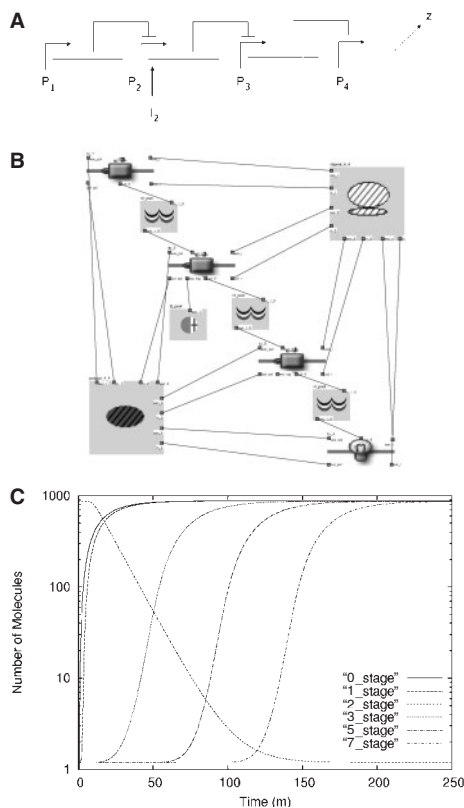

Fig. 2. Engineered cascades. (**A**) Scheme of the three-step cascade. Every stage is lead by a promoter (P_i); the first three genes produce a repressor, the last one a reporter protein (z). I_2 represents the inducer acting on the repressor of promoter P_2. (**B**) Implementation of the three-step cascade with ProMoT. A composite device (protein generator or reporter) is used for each transcription unit. (**C**) Multiple-step-cascade deterministic simulations. Beside the expression levels of some of the cascades between Steps 1 and 7, the single gene expression (0 stage) is shown.

putting basic parts, wherever possible, inside composite devices and by connecting these composite devices to the pools, to other basic parts and also between each other when necessary. Finally, the MDL-encoded model for a complete circuit can be directly exported into Matlab code for deterministic simulations. Alternatively, the model can be exported into the more general SBML format (Hucka *et al.*, 2003). After a parsing step through a stand-alone Perl script (due to the specific SBML format generated by ProMoT; see Supplementary Material), one can choose the most appropriate software to run deterministic or stochastic simulations.

5 RESULTS

For a proof-of-concept study, we tested our modeling framework on some of the best-established synthetic genetic circuits. The results presented in this section have been obtained by running deterministic simulations in COPASI (Hoops *et al.*, 2006) and stochastic simulations in Dizzy (Ramsey *et al.*, 2005), illustrating the compatibility of the concept with different software tools.

As a first benchmark, we chose the seven-step-cascade device (Hooshangi *et al.*, 2005). The simpler three-step cascade is shown in Figure 2A, B. In this circuit, every gene synthesizes a

repressor that acts only on the successive *cis*-regulatory part. In our implementation, we made use of a basic promoter in the first transcription unit. All the others units are controlled by inducible two-operator promoters (see Supplementary Material for details), although only the second-stage promoter is induced by a signal.

To compare simulation results with the stochastic simulations reported in Hooshangi *et al.* (2005), we used the given parameter values and only changed the translation initiation frequency and the leakage transcription rate. Moreover, we extrapolated the association and dissociation constants between RNA polymerases and promoters, and between ribosomes and RBSs from literature data (see Supplementary Material for all details). Following (Hooshangi *et al.*, 2005), every cascade step was reproduced in 20 copies. Simulations were run in two steps: first we let the system reach a steady state in the absence of external signals, then inducers were sent to the second-stage promoter with a fixed rate. Cooperativity between repressors has not been taken into account. Although our deterministic calculations give reporter molecule numbers (from the last gene expression unit) that are slightly lower for the basal production alone, the qualitative behavior of the system is correctly reproduced (Fig. 2C). In particular, the time delay between Steps 3 and 5 (as well as between Steps 5 and 7) is roughly 46 min, which matches well with the 44 min inferred by Hooshangi *et al.* (2005).

As another benchmark we considered the so-called *Repressilator*, a ring oscillator established in bacteria (Elowitz and Leibler, 2000). Following the original publication, we simulated it as a circuit made of three identical transcription units where the first gene represses the second gene, the second represses the third, and this in turn inactivates the first gene (Fig. 3A, B). Stochastic simulations (Fig. 3C) show that for the chosen parameter values, oscillations in the expression of the three repressor genes are sustained for a long time period. A detailed description of the circuit simulation together with a discussion of the RNA polymerase and ribosome dynamics inside this network is provided in Supplementary Material.

Besides these two benchmarks we also realized the *positive and negative feedback oscillator* (Atkinson *et al.*, 2003), the *pulse generating network* (Basu *et al.*, 2004) and the *bistable toggle switch* (Gardner *et al.*, 2000). In all cases, we were able to reproduce their behavior correctly (see Supplementary Material). In addition, we developed an 'artificial' large-scale circuit, which illustrates that even with moderate network complexity, the dependency of the behavior on global pools of, for instance, RNA polymerases is significant; correspondingly, one expects an impact of such circuits on the 'natural' cellular behavior, which needs to be accounted for (see Supplementary Material for details).

6 CONCLUSION

Conceptually, the design of synthetic gene circuits with composable parts has been proposed, but not yet fully realized in a corresponding model-based design tool. Here, we present a formal modeling framework based on the ODE formalism that permits modular model composition. A synthetic circuit can be simulated just by connecting the desired parts to each other. The interfaces between the parts are established by at least four different common signal carriers whose fluxes are exchanged between the parts themselves and the pools where these molecules are stored. To test the validity of the concept, we reproduced the behavior of several well-established
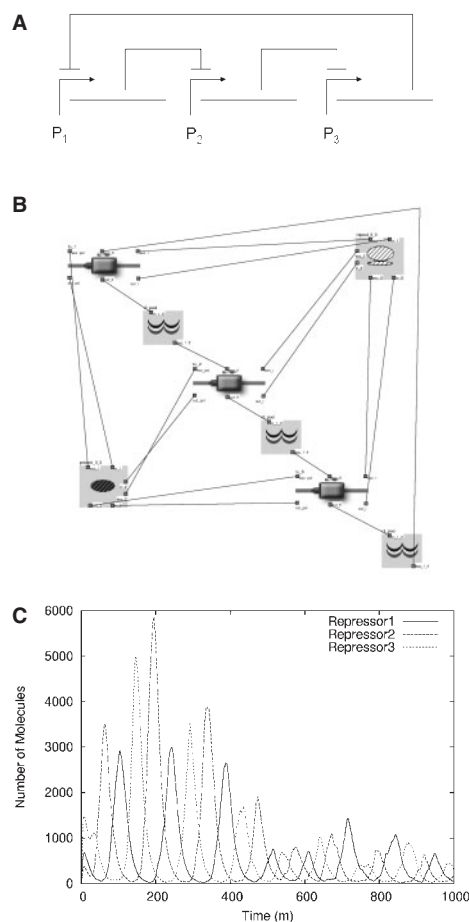

Fig. 3. The repressilator. (**A**) Circuit scheme. (**B**) Implementation with ProMoT. Three protein generators and five pools have been deployed on the canvas. (**C**) Result of stochastic simulations.

synthetic circuits; the simulation results were in good agreement with literature data.

Compared to other methods and tools for synthetic circuit design, our solution appears extremely easy and intuitive to use. It permits building circuits visually, just by displaying the desired parts on the screen and by connecting them through wires. This amounts to basically reproducing circuit schemes without caring about the underlying MDL part code. Starting from the basic parts, one can assemble composite devices of different degree of complexity so that even the design of a network made of dozens of genes is a relatively easy task. Simulations of complex networks, furthermore, can be run without particular constraints because of a detailed description of the reactions taking place inside each part. Compared to the traditional Hill formalism, this enables full scalability. As a consequence, one can directly estimate the value of parameters generally 'hidden' inside the Hill constants and coefficients, and understand their order of magnitude required to yield particular dynamic phenomena such as oscillations. Once the circuit model has been designed, its MDL code serves as a template that can be reloaded and modified in the GUI of ProMoT. Exported into SBML or Matlab format, the circuit model generality is retained. The associated files can be reused for all the necessary simulation studies.

To improve the method, we intend to generalize the promoter construction to enable combinatorial promoter modeling and to include cooperativity phenomena in more detail. More generally, combining the design tool with, for instance, the MIT Registry, literature databases, and other resources could eventually establish a new computational infrastructure for synthetic biology that enables researchers to select biological parts accurately and then to design and test the functioning of the genetic circuits under study in an intuitive, automated fashion.

ACKNOWLEDGEMENTS

REFERENCES

Alon,U. (2006) *An Introduction to Systems Biology*. Chapman & Hall/CRC Press, Boca Raton, FL.

Andrianantoandro,E. *et al*. (2006) Synthetic biology: new engineering rules for an emerging discipline. *Mol. Syst. Biol.*, **2**, 2006.0028.

Atkinson,M.R. *et al*. (2003) Development of genetic circuitry exhibiting toggle switch or oscillatory behavior in Escherichia coli. *Cell*, **113**, 597–607.

Basu,S. *et al*. (2004) Spatiotemporal control of gene expression with pulse-generating networks. *Proc. Natl Acad. Sci. USA*, **101**, 6355–6360.

Benner,S.A. and Sismour,A.M. (2005) Synthetic biology. *Nat. Rev. Genet.*, **6**, 533–543.

Cai,Y. *et al*. (2007) A syntactic model to design and verify synthetic genetic constructs derived from standard biological parts. *Bioinformatics*, **23**, 2760–2767.

Drubin,D.A. *et al*. (2007) Designing biological systems. *Genes Dev.*, **21**, 242–254.

Elowitz,M.B. and Leibler,S. (2000) A synthetic oscillatory network of transcriptional regulators. *Nature*, **403**, 335–338.

Endy,D. and Deese,I. with the MIT Synthetic Biology Working Group (2005) Adventures in synthetic biology. Appeared in Foundations for engineering biology p449, *Nature*, **438**, 449–453.

Funahashi,A. *et al*. (2003) CellDesigner: a process diagram editor for gene-regulatory and biochemical networks. *BIOSILICO*, **1**, 159–162.

Gardner,T.S. *et al*. (2000) Construction of a genetic toggle switch in Escherichia coli. *Nature*, **403**, 339–342.

Ginkel,M. *et al*. (2003) Modular modeling of cellular systems with ProMoT/Diva. *Bioinformatics*, **19**, 1169–1176.

Goler,J.A. (2004) BioJADE: A design and simulation tool for synthetic biological systems. *Technical report*, MIT, Cambridge, MA.

Hasty,J. *et al*. (2002) Engineered gene circuits. *Nature*, **420**, 224–230.

Hoops,S. *et al*. (2006) COPASI–a COmplex PAthway SImulator. *Bioinformatics*, **22**, 3067–3074.

Hooshangi,S. *et al*. (2005) Ultrasensitivity and noise propagation in a synthetic transcriptional cascade. *Proc. Natl Acad. Sci. USA*, **102**, 3581–3586.

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

Kærn,M. and Weiss,R. (2006) Synthetic gene regulatory systems. In Szallasi,Z. *et al*. (eds) *System Modeling in Cellular Biology*. The MIT Press, Cambridge, MA, pp. 269–295.

Klipp,E. *et al*. (2005) *Systems Biology in Practice*. WILEY-VCH Verlag, Weinheim.

Kosuri,S. *et al*. (2007) TABASCO: a single molecule, base-pair resolved gene expression simulator. *BMC Bioinformatics*, **8**, 480.

Nagel,L.W. and Pederson,D.O. (1973) SPICE (Simulation Program with Integrated Circuit Emphasis). *Memorandum No. ERL-M382*, University of California, Berkley.

Ramsey,S. *et al*. (2005) Dizzy: stochastic simulation of large-scale genetic regulatory networks. *J. Bioinform. Comput. Biol.*, **3**, 415–436.

Rouilly,V. *et al*. (2007) Registry of BioBrick models using CellML. *BMC Syst. Biol.*, **1**(Suppl. 1), 79–80.

Samoilov,M.S. and Arkin,A.P. (2006) Deviant effects in molecular reaction pathways. *Nat. Biotechnol.*, **24**, 1235–1240.

Sayut,D.J. *et al*. (2007) Engineering and applications of genetic circuits. *Mol. Biosyst.*, **3**, 835–840.

Tomshine,J. and Kaznessis,Y.N. (2006) Optimization of a stochastically simulated gene network model via simulated annealing. *Biophys. J.*, **91**, 3196–3205.