BioNetCAD: design, simulation and experimental validation of synthetic biochemical networks

Stéphanie Rialle[1,*], Liza Felicori[1], Camila Dias-Lopes[1], Sabine Pérès[1], Sanaâ El Atia[1], Alain R. Thierry[1], Patrick Amar[2] and Franck Molina[1,*]

[1]SysDiag UMR 3145 CNRS/Bio-Rad, Modélisation et ingénierie de systèmes complexes biologiques pour le diagnostic, Cap Delta/Parc Euromédecine, 1682 rue de la Valsière, CS 61003, 34184 Montpellier Cedex 4 and [2]Laboratoire de Recherche en Informatique, CNRS, UMR 8623, Université Paris-Sud 11, 91405 Orsay Cedex, France

Associate Editor: Trey Ideker

ABSTRACT

Motivation: Synthetic biology studies how to design and construct biological systems with functions that do not exist in nature. Biochemical networks, although easier to control, have been used less frequently than genetic networks as a base to build a synthetic system. To date, no clear engineering principles exist to design such cell-free biochemical networks.

Results: We describe a methodology for the construction of synthetic biochemical networks based on three main steps: design, simulation and experimental validation. We developed BioNetCAD to help users to go through these steps. BioNetCAD allows designing abstract networks that can be implemented thanks to CompuBioTicDB, a database of parts for synthetic biology. BioNetCAD enables also simulations with the HSim software and the classical Ordinary Differential Equations (ODE). We demonstrate with a case study that BioNetCAD can rationalize and reduce further experimental validation during the construction of a biochemical network.

Availability and implementation: BioNetCAD is freely available at http://www.sysdiag.cnrs.fr/BioNetCAD. It is implemented in Java and supported on MS Windows. CompuBioTicDB is freely accessible at http://compubiotic.sysdiag.cnrs.fr/

Contact: stephanie.rialle@sysdiag.cnrs.fr; franck.molina@sysdiag .cnrs.fr

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

1 INTRODUCTION

Synthetic biology focuses on designing biological systems that are characterized by non-natural functions and wide-ranging domains of application: from bio-compatible devices (Elowitz and Leibler 2000; Gardner *et al*., 2000), biofuels (Dellomonaco *et al.* 2010; Marner, 2009), drugs production (Ro *et al*., 2006), creation of smart therapeutic systems (Lu and Collins 2009), to genetically modified organisms for environmental purposes (Cases and de Lorenzo 2005). To this aim engineering principles (i.e. a rigorous methodology that includes several key steps such as specifications, design, modeling,

fabrication and quality control) (Endy, 2005; Gulati *et al*., 2009) are commonly used. A pioneer example is the BioBricks project which focuses on the design and construction of genetic circuits and on their expression in host cells to produce a useful behavior or compound. To help in this long term goal, the registry of standard biological parts was initiated by the Massachusetts Institute of Technology (http://partsregistry.org/ Main_Page) in order to conceptualize, standardize, organize in a hierarchy and register compatible DNA parts. Another exciting approach is the use of biological parts, such as enzymes and metabolites, as signals and compounds to reproduce logic operations like those performed by electronic circuits (Arkin and Ross, 1994; Baron *et al*., 2006; Strack *et al*., 2008).

Given the complex nature of the biological material, most of the engineering principles have not been exploited as yet for the construction of artificial biological systems. Moreover, all the current achievements are 'hand-made' approaches that are difficult to generalize. The clear necessity to develop concepts, methods and technologies to support synthetic biology projects has been emphasized repeatedly (Andrianantoandro *et al*., 2006; Endy, 2005) and particularly the lack of computer-aided design (CAD) tools that could facilitate the conceptualization, design and simulation of such synthetic systems (Gulati *et al*., 2009). Ideally, CAD tools should be linked to registries of standardized, modular and re-usable biological parts, since well-defined independent parts will ease the efficient development of synthetic biological systems (Arkin, 2008; Peccoud *et al*., 2008). Several computational tools that specifically support the design of artificial gene circuits are currently available (Marchisio and Stelling, 2009). For instance, BioJADE (Goler *et al*., 2008) is a graphical design tool to engineer genetic systems by graphical representation of abstract components (like promoters, ribosome binding sites, coding sequences, etc.). GenoCAD (Cai *et al*., 2010) is a genetic design web tool based on context-free grammar and a library of genetic parts which allows the design of an easily downloadable DNA sequence. TinkerCell (Chandran *et al*., 2009; http://www.tinkercell.com) is a synthetic biology CAD tool for visually constructing and analyzing biological networks. It allows the design and the simulation of biological networks from a hierarchy of parts and modules that can be defined by the user.

Most of the currently available CAD applications for synthetic biology aim to develop genetic networks to be implemented in modified microorganisms. However, the use of host cells produces noise and not controllable behaviors, leading to poorly robust

---

*To whom correspondence should be addressed.

systems. Furthermore, the use of genetic networks implies ethical questions that concern their use for *in vivo* tests. Consequently, alternative, simplified protein/metabolic networks have been proposed to tackle such difficulties (Bromley *et al.*, 2008; Hold and Panke, 2009). This kind of network is mainly composed of sets of proteins performing related binding or catalytic activities and transforming metabolite compounds. They do not rely on any transcription or translation of genes, and therefore do not need any host cell. If needed, they can be encapsulated into artificial vesicles (Doktycz and Simpson 2007).

In this article, we describe a methodology to build protein-based synthetic networks in three steps: design, simulation and experimental validation. In order to assist the design and dynamic simulation of such synthetic networks, we developed BioNetCAD, an original bioinformatics tool coupled to CompuBioTicDB, a database of biological components. The aim was to facilitate the design of conceptual biochemical networks and to help the user in finding the most suitable molecules to achieve successful circuits that can be tested in a real biological environment. We illustrate BioNetCAD use with a case study leading to the construction of a network carrying a logical behavior.

## 2 METHODOLOGY FOR THE RATIONAL DESIGN AND IMPLEMENTATION OF SYNTHETIC SYSTEMS

Here we describe a three-step methodology for *de novo* design of biological systems that includes both conceptual and experimental aspects in order to build reliable biochemical networks. The first two steps (i.e. design and simulation) are carried out *in silico*, while the last one is the laboratory-based, experimental validation (Fig. 1).

2.1 Design

We propose to start this step by the construction of an abstract network, i.e. a biochemical network made of theoretical molecules, which will be implemented with well-characterized components. Consequently, the identification, formalization and storage of basic components are key requirements of this step in view of the necessity of predicting and controlling the behavior of the designed system. The precise functionality of the individual components must be described, i.e. how they act, in which conditions, their interactions, etc. As a consequence, this work benefitted of the previously developed BioΨ language (Maziere *et al.*, 2004; Peres *et al.*, 2010), which allows a multi-scale formal description of biological processes. Accordingly, the design of an abstract biological system is the result of the combination of basic components that carry elementary actions. We then pooled the identified components and their description in a database, CompuBioTicDB, in order to facilitate the construction of predictable and dependably controllable, robust networks. Its concept is similar to that of the BioBricks registry, but it aims to provide parts and devices useful to the construction of non-living protein-based synthetic systems rather than genetic-based systems.

2.2 Simulation

Simulation allows saving time and money by testing the consistency of the designed system, certifying its robustness and predicting and optimizing parameters, before the experimental validation assays.
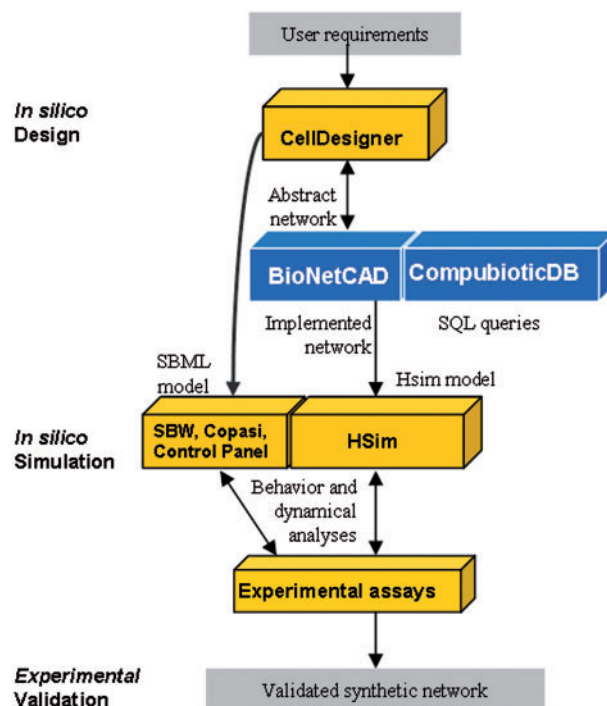
Fig. 1. BioNetCAD allows the integration of CellDesigner, CompuBioTicDB and simulation tools. An abstract network is designed with CellDesigner according to the requirements specified by the user. Through BioNetCAD, the CompuBioTicDB database is interrogated to identify candidate molecules to implement the network. BioNetCAD returns an implemented network that can be used for dynamical simulations by the HSim software or the classical ODE tools linked with CellDesigner thanks to the SBML format. Finally, the network can be experimentally validated.

The global behavior of the designed system is simulated with the HSim software (Amar *et al.*, 2008). This stochastic automaton allows multi-agent type of simulations that admit dynamical (temporal and spatial) analyses of the system and also tolerates very small concentrations of molecules.

2.3 Experimental validation

The final step is the experimental validation of the network: its biochemical stability and behavior are checked to verify that it accomplishes the expected task(s). Further optimization procedures can be conducted. Because of the heaviness of the experimental step (in regards to financial costs and time) we should beneficiate of the simulation to guide optimization process (initial quantities, dynamics, etc.). In turn, the experimental data can be used to refine the modeling and simulation parameters.

3 BIONETCAD: A SYNTHETIC BIOLOGICAL NETWORK COMPUTER-ASSISTED DESIGN TOOL

Our synthetic network methodology requires computational tools for assisting the user through its three steps. To accomplish this requirement, we developed BioNetCAD, a plug-in for CellDesigner, that integrates (i) the graphical functionalities (Kitano *et al.*, 2005) of CellDesigner (Funahashi *et al.*, 2003, 2008); (ii) CompuBioTicDB,
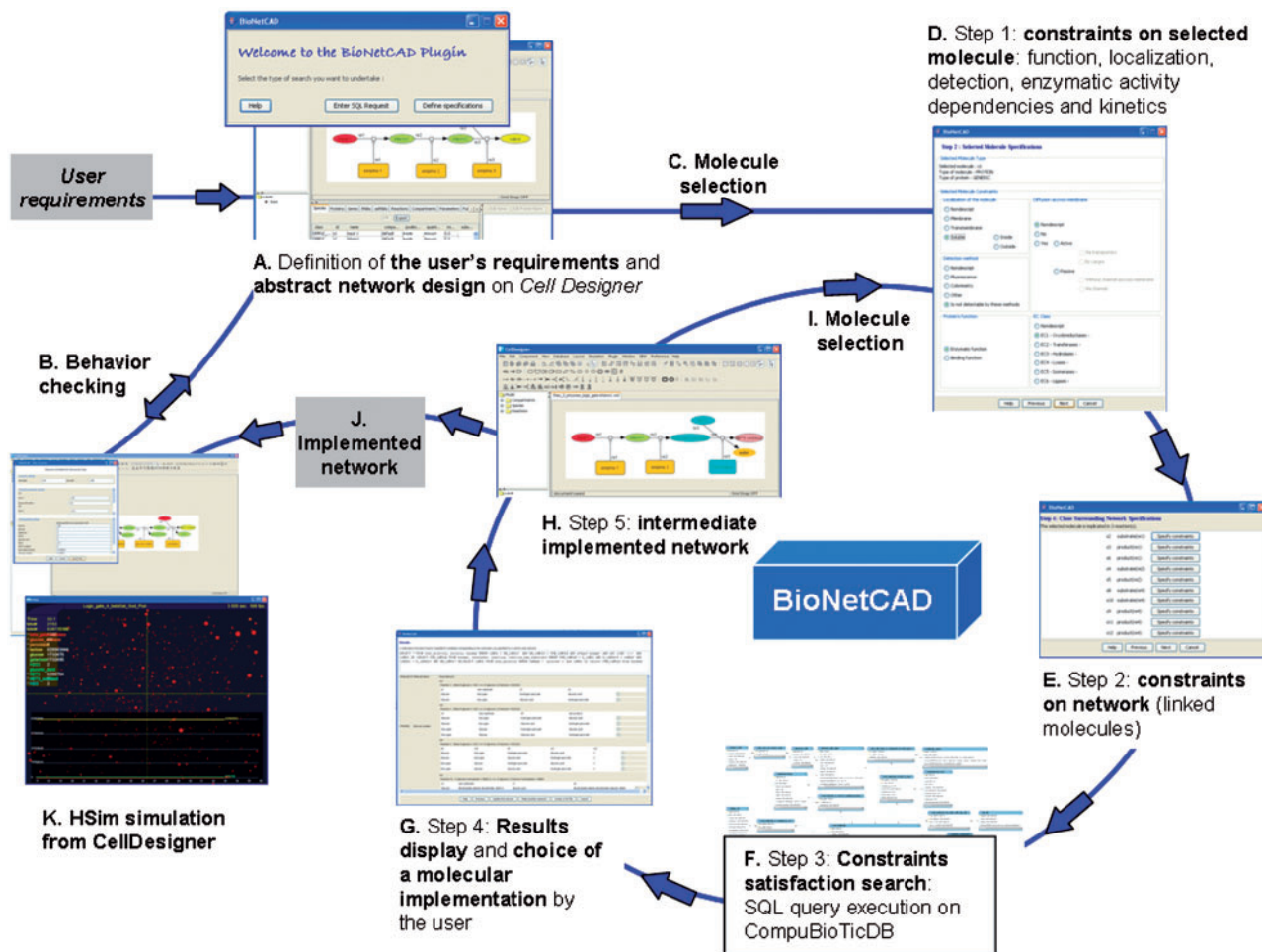
Fig. 2. Step-wise network design/implementation and simulation using BioNetCAD. The design strategy starts with an abstract network designed according to the user's requirements and its representation in CellDesigner. Once the plug-in is launched, the user is asked to select a molecule within the network. Then, several steps lead the user to specify different constraints about the selected molecule and the neighboring molecules in the network. For instance, if the user selects a protein that catalyzes a reaction, he will be able to specify the enzymatic class (EC number), the substrate(s) and the product(s) of such reaction. Next, BioNetCAD integrates the specifications and elaborates a SQL query on CompuBioTicDB to find molecules that suit these requirements. The results are displayed and the user can choose a molecule to implement the abstract molecule. The network is then updated with the chosen molecule and possibly with linked molecules, such as substrate(s) or product(s). The user can then start again from another theoretical molecule in the network, and so on until a fully implemented network is obtained. The HSim launcher functionality of BioNetCAD allows the automatic generation of HSim input files from the CellDesigner model as defined by the user and also launches the HSim software.

a database of biological elements; and (iii) simulation functions like the HSim software and classical Ordinary Differential Equations (ODEs). BioNetCAD assists the network design by iteratively matching the basic components of the abstract network with CompuBioTicDB components. Moreover, it can launch a simulation of the network, using HSim or the simulation platforms which are already integrated in CellDesigner (Fig. 1).

3.1 BioNetCAD for stepwise network design

When designing an abstract network, the user has to identify existing molecules that can perform the desired functionalities. For instance, which molecules can he choose to build a small biochemical (e.g. enzymatic) network that performs a given logic task? To help answering this question, BioNetCAD facilitates the selection of

molecules to implement a protein network, in a stepwise manner (Fig. 2 and Section 2 of Supplementary Material for the algorithm). First, the user designs an abstract network according to specific requirements and draws it using CellDesigner (Fig. 2A). At this level, the user can already check the qualitative behavior of this abstract network by launching qualitative simulations (Fig. 2B). Next, the user selects a starting molecule (Fig. 2C) and specifies the constraints concerning that molecule (Fig. 2D) and its surrounding molecular network (Fig. 2E). Based on these specifications, BioNetCAD elaborates a query and executes it on CompuBioTicDB in order to find suitable molecular implementations for the selected molecule and its surrounding network (Fig. 2F). BioNetCAD presents the different results provided by the database and the user can choose one of them (Fig. 2G) to build an intermediate implemented network (Fig. 2H). This cycle is then repeated (Fig. 2I)

until a fully implemented network is obtained (Fig. 2J), ready to be simulated (Fig. 2K) and tested experimentally. Section 4 illustrates the use of BioNetCAD with an example.

The adoption of CellDesigner as plug-in relies on the decision to represent graphically biological systems using standardized notations. Indeed, engineering methodology needs to be based on standard notations and representations; moreover, it seems important to start using tools that permit the exchange of information between different models. Accordingly, CellDesigner supports the SBML format (Hucka *et al.*, 2003), which is widely used in the systems biology community, to share networks models.

3.2 CompuBioTicDB for the choice and implementation of molecules

3.2.1 Parts and devices The CompuBioTicDB database stores two categories of components: biological molecules, such as proteins and metabolites, and devices such as sensors, switches or timekeepers, which use the biological molecules as basic parts. Following the analogy with the electronic components, we propose devices that can be assembled in order to form a synthetic biochemical system. These devices are molecular elements which play pre-determined well-defined roles, and are named by the role they play. An example of device is the switch: it can for instance switch on a synthetic biological system by initiating a reaction that produces a compound indispensable for the next reaction. Basically, those components are abstract, but their molecular implementations are real proteins and small molecules that are categorized in CompuBioTicDB. Devices constitute a higher level in the abstraction hierarchy of components for synthetic biology in comparison to basic biological molecules. We classified these functional devices in six categories: compartmentalization, compartmental environment control, time management, shape management, energy provider and biological signal management devices.

3.2.2 BioNetCAD molecule implementation using CompuBioTicDB CompuBioTicDB stores essentially proteins, in particular enzymes and small molecules, such as substrates or cofactors. CompuBioTicDB does not intend to create a new database with information already contained in other databases such as UniProtKB (The UniProt Consortium, 2010) (http://www.uniprot.org/) or Brenda enzymes (Chang *et al.*, 2009) (http://www.brenda-enzymes.org/), but only to integrate structural, functional and kinetics data that are needed when designing a synthetic network. Additionally, we have defined a synthetic biology score for proteins according to their easy of use in a synthetic biology context. This score reflects the robustness of a protein, i.e. its suitability for non-natural environments, also the potential interactions with other proteins and environmental factors, such as pH, temperature, ionic conditions, etc. This score is arbitrarily given and is defined as a scale of integers from 0 to 4 : 1 means that a protein is not easily usable in a synthetic biology system, while 4 indicates that it is a very good component for synthetic systems. For instance, widely used enzymes in the biotechnology field are considered good candidates for synthetic biology, since their robustness has been demonstrated by their use in various conditions. Consequently, they received by default a score of 3. Finally, 0 means that we do not know yet if the protein can be easily handled or not, and that experiments need to be performed to ascertain it.

3.3 BioNetCAD for the simulation of the dynamic network behavior using the HSim launcher

The second main functionality of BioNetCAD is to make a link between the network drawn with CellDesigner and the spatial and temporal simulation that can be performed using the stochastic automaton HSim (Amar *et al.*, 2008). Based on the CellDesigner network and parameters defined by the user, the HSim input configuration file is constructed automatically by BioNetCAD and the simulation can be launched. As shown in the next section, HSim allowed us to verify that the designed network could perform the expected tasks and helped in the choice/optimization of the experimental parameters.

4 A CASE STUDY: *IN SILICO* NETWORK DESIGN, SIMULATION AND *IN VITRO* EXPERIMENTAL ASSAY

The general objective was to design a network that performs the task of a molecular logic AND gate. The design of molecular bio-logic gates is of great interest since it may be useful to feed CompuBioTicDB with sophisticated components that could be reused in synthetic networks and under different conditions.
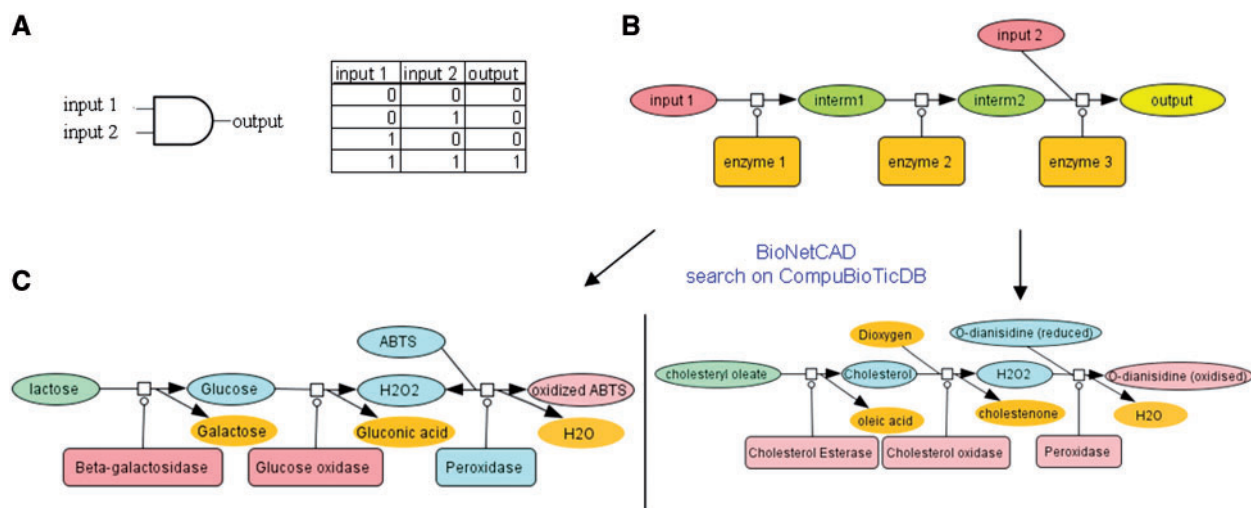
4.1 Design

4.1.1 Requirements and generation of an abstract network First, the BioNetCAD user had to define precisely the requirements. In this case, the objective was to build a multi-enzymatic biochemical 'AND' gate, made of three enzymes, in which two input substrates are transformed into one detectable product (one substrate being processed by one enzyme, the other by another enzyme). Such a multi-enzymatic AND gate opens various possibilities for different reaction branching thanks to its multiple input metabolites. Specifically, the requirements were defined as follows:

- A serial network of three enzymes: the product of the first one was the substrate of the second one and so on.
- Two inputs. By 'input' we meant a substrate the initial concentration of which was tunable.
- One input was the substrate of the first enzyme in the network, while the other was the substrate of the third enzyme.
- One output, which was a measurable product.
- The generation of the output was conditioned by the initial presence of the two substrates (Fig. 3B) following in our case an AND truth table as classically expressed in electronics (Fig. 3A, right panel).

4.1.2 Generation of implemented networks using BioNetCAD Starting from an abstract network drawn with CellDesigner, the user employed BioNetCAD to obtain an implemented network with real molecules. A 'retro-implementation' approach was followed, i.e. the implementation started from the end (output) of the network. First, the output molecule was selected with the constraint of being a detectable product, in our case by a colorimetric assay. Based on the results of the CompuBioTicDB search, BioNetCAD

Fig. 3. Case study: design of a logic AND gate. (**A**) Logic AND gate, scheme and expected behavior. Inputs and outputs are implemented with small molecules and the process is carried out by enzymatic reactions. (**B**) Abstract network of a multi-enzymatic AND gate with three enzymes. The abstract network is drawn in accordance to the following requirements: serial network of three enzymes, the product of the first one is the substrate of the second and so on, two inputs which are tunable small molecules, one input is the substrate of the first enzyme in the network, while the other is the substrate of the third enzyme, one output which is a measurable product, production of the output conditioned by the initial presence of the two substrates. (**C**) Two examples of implementation of a tri-enzyme AND gate. According to the choices made at each step of the BioNetCAD search on CompuBioTicDB, the same abstract network can lead to different molecular implementations that fulfill the initial requirements. In the case study, the left network was chosen; thus, after implementation, input 1 of the logic gate was represented by lactose, input 2 by ABTS and the output by oxidized ABTS.

proposed several molecules, such as oxidized ABTS or oxidized *o*-Dianisidine, which are products of reactions catalyzed by peroxidase, or nitrophenol, the product of the reaction catalyzed by alkaline phosphatase. By making a choice (in this case oxidized ABTS) the user created constraints on the upstream components of the network. This choice implied peroxidase as implementation of 'enzyme 3', ABTS as implementation of 'input 2' and H_2O_2 as implementation of 'interm2'. The H_2O molecule was automatically added into the network since it is a second product of the peroxidase reaction. In the next step, 'enzyme 2' was selected, and the user specified that it needed to give H_2O_2 as product. Again, several molecules were found by CompuBioTicDB and proposed to the user. The user chose glucose oxidase, which uses glucose as substrate (implementation of 'interm1'). Using the same method, the first enzyme (β-galactosidase) was then selected among the proposed molecules with lactose as substrate. Finally, a fully implemented network was obtained (Fig. 3C, left scheme). Several alternative implementations exist for a given abstract network depending on the choices made by the user at each step of the search with BioNetCAD in CompuBioTicDB. Thus, several networks can be simulated and possibly assayed experimentally to be validated. Figure 3C illustrates how a unique abstract network with specified characteristics can lead to two different implemented networks that share the same functionalities.
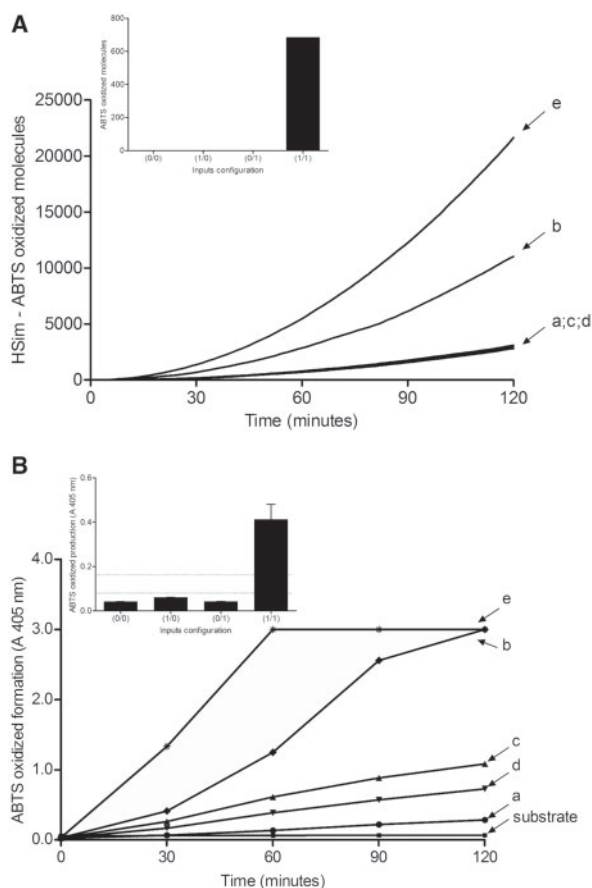
4.2 Modeling, simulation and experimental validation

Two modeling methods were chosen to simulate the behavior of the designed multi-enzymatic logic AND gate network: ODE continuous approach and HSim discrete approach. Both HSim (Fig. 4A, upper panel) and ODE (Supplementary Fig. S1)

simulations confirmed that the designed network behaved as a logic AND gate according to the expected AND truth table with lactose and ABTS as inputs. As expected, output production (oxidized ABTS = 1) was only observed when both inputs (lactose = 1 and ABTS = 1) were initially present.

To more precisely characterize the behavior and to study the global velocity of the network, different simulations were made using various concentrations of the three enzymes. HSim and ODE simulations gave comparable curves (Fig. 4A, lower panel and Supplementary Fig. S1). In order to optimize the concentration of the three enzymes and to reduce the time needed and the amount of material used during the experimental validation, a series of enzyme ratios were simulated. Test e, with a $3:3:3$ ratio (U/mL; β-galactosidase, glucose oxidase and peroxidase, respectively), showed the highest rate of output formation, as expected, but required high enzyme concentrations. Conversely, test b (3:1.6:0.04), even with two times less glucose oxidase and 75 times less peroxidase than test e, still presented a satisfactory output formation. Curves a (3:0.4:0.04), c (0.75:1.6:0.04) and d (1:1:1) were almost superimposed in both simulations. We verified in all simulations that substrates/inputs (lactose and ABTS) were not in limiting concentrations (Supplementary Material). The simulations show that the test b was the most satisfactory for output formation and low-enzymes concentration requirements compared to other tests.

The experimental validation, with the different input configurations (Fig. 4B. top-left) measured using the conditions of test b after 30 min, gave a AND gate output signal (see Supplementary Material for information about materials and methods). The cut off value of the AND gate was established for output 0 as absorbance <0.08 and for output 1 as absorbance >0.16.

Fig. 4. Case study: simulation and experimental validation of a logic AND gate. Production of oxidized ABTS (output) in the course of time (minutes) was assessed by HSim simulation (**A**) and biochemical assays (**B**), with the same initial concentrations of inputs (250 mM of lactose and 2.5 mM of ABTS) and different enzyme concentration ratios. For both graphics, concentrations of enzymes are: a = 3, 0.4 and 0.04; b = 3, 1.6 and 0.04; c = 0.75, 1.6 and 0.04; d = 1, 1 and 1; e = 3, 3 and 3 U/mL (β-galactosidase, glucose oxidase and peroxidase, respectively). Bar representation of the AND gate (condition b) output after 30 min of simulation/experimentation is showed in the top left panel of each graphic.

The experimental curves (Fig. 4B, lower panel) were in line with the results obtained during the simulations: assays e and b presented elevated rates of output formation, while assays c, d and a showed lower rates. On the other hand, the experimental validation gave a better resolution than the simulations for curves a, c, d. Unfortunately, after 60 min, curve e exceeds the maximum limit of the spectrophotometer (OD = 3).

Altogether, this demonstrates that the characterization of small networks using simulation is relevant for rationalizing the choice of experimental conditions used to validate a synthetic biochemical network.

5 CONCLUSION AND DISCUSSION

Although many synthetic biology projects aim to build genetic networks within living cells, only few recent works focused on the design of cell-free biochemical (signaling or metabolic) networks.

In this case, the synthetic system is not self-replicable but its composition and behavior might be better controlled. This may be of great advantage for eventual future use for health applications. Recently, Grunberg *et al.* (2010) proposed a framework for the design of synthetic protein networks using interactions between modular proteins. However they still based their work on DNA parts. Other studies aimed at rewiring signaling circuits (Pryciak 2009; Zeke *et al.*, 2009) by taking advantage of the amazing modularity of protein domains to produce fusion proteins in order to create non-natural interactions and thus modify existing pathways.

Here we propose to use engineering principles to design-controlled biochemical synthetic networks. To support our approach we developed a methodology based on three main steps (i.e. design, simulation and experimental validation) which can be easily carried out using the computing tool BioNetCAD.

In the design step, it is necessary to use BioNetCAD, integrated in the CellDesigner environment, to draw an abstract network using SBGN annotation standards. To efficiently implement the abstract network of interest we propose a retro-implementation method using BioNetCAD in combination with CompuBioTicDB. CompuBioTicDB is central during our design process because it assembles information on abstract functions, associated constraints and kinetic parameters in order to build an implemented network that satisfies the constraints of the network structure and of the contextual dependencies of the well-studied molecules found in CompuBioTicDB. To improve our design/implementation methodology it would be of great interest to have a multi-scale vision of the *de novo* design of protein networks (from protein domains to system levels) and thus to include non-natural well-characterized proteins as CompuBioTicDB parts. To this aim, we currently work on the integration in CompuBioTicDB of the recent concept of elementary bricks of action at the level of protein domains (Maziere *et al.*, 2004; Peres *et al.*, 2010).

Qualitative simulations can then be used during the design step to anticipate the global behavior of the abstract network. This can be done before any specific molecular implementation. Finally, before going into costly experimental validation, different kinds of quantitative *in silico* simulations can be performed to check the stability of the global network behavior and to optimize various parameters (quantities, kinetics, etc.). Currently, the kinetic parameters appear to be a limiting factor since they are not often available, or not relevant or contradictory. A simple example of design of a multi-enzymatic AND logic gate allowed demonstrating the robustness of our methodology which was also confirmed by experimental tests under various conditions.

Compared to TinkerCell (the only currently available CAD tool able of designing non genetic networks), BioNetCAD focuses on protein networks using current graphical and model exchange standards (see Supplementary Material). In addition, it integrates original tools as CompuBioTicDB and the spatio-temporal simulator HSim.

In conclusion, BioNetCAD constitutes a computer assisted design tool that greatly enhances the rational design and optimization of synthetic biological system. It can be used for various purposes ranging from bioassay design, system design or eventually modification of existing networks.

REFERENCES

Amar,P. *et al*. (2008) A stochastic automaton shows how enzyme assemblies may contribute to metabolic efficiency. *BMC Syst. Biol.*, **2**, 27.

Andrianantoandro,E. *et al*. (2006) Synthetic biology: new engineering rules for an emerging discipline. *Mol. Syst. Biol.*, **2**, 0028.

Arkin,A. (2008) Setting the standard in synthetic biology. *Nat. Biotechnol.*, **26**, 771–774.

Arkin,A. and Ross,J. (1994) Computational functions in biochemical reaction networks. *Biophys. J.*, **67**, 560–578.

Baron,R. *et al*. (2006) Elementary arithmetic operations by enzymes: a model for metabolic pathway based computing. *Angew. Chem. Int. Ed. Engl.*, **45**, 1572–1576.

Bromley,E.H. *et al*. (2008) Peptide and protein building blocks for synthetic biology: from programming biomolecules to self-organized biomolecular systems. *ACS Chem. Biol.*, **3**, 38–50.

Cai,Y. *et al*. (2010) GenoCAD for iGEM: a grammatical approach to the design of standard-compliant constructs. *Nucleic Acids Res*., **38**, 2637–2644.

Cases,I. and de Lorenzo,V. (2005) Genetically modified organisms for the environment: stories of success and failure and what we have learned from them. *Int. Microbiol.*, **8**, 213–222.

Chandran,D. *et al*. (2009) TinkerCell: modular CAD tool for synthetic biology. *J. Biol. Eng.* **3**, 19.

Chang,A. *et al*. (2009) BRENDA, AMENDA and FRENDA the enzyme information system: new content and tools in 2009. *Nucleic Acids Res*., **37**, D588–D592.

Dellomonaco,C. *et al*. (2010) The path to next generation biofuels: successes and challenges in the era of synthetic biology. *Microb. Cell Fact.*, **9**, 3.

Doktycz,M.J. and Simpson,M.L. (2007) Nano-enabled synthetic biology. *Mol. Syst. Biol.*, **3**, 125.

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

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

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

Funahashi,A. *et al*. (2008) CellDesigner 3.5: A Versatile Modeling Tool for Biochemical Networks. *Proc. IEEE*, **96**, 1254–1265.

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

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

Grunberg,R. *et al*. (2010) Building blocks for protein interaction devices. *Nucleic Acids Res.*, **38**, 2645–2662.

Gulati,S. *et al*. (2009) Opportunities for microfluidic technologies in synthetic biology. *J. R. Soc. Interface*, **6** (Suppl. 4), S493–S506.

Hold,C. and Panke,S. (2009) Towards the engineering of in vitro systems. *J. R. Soc. Interface*, **6** (Suppl. 4), S507–S521.

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.

Kitano,H. *et al*. (2005) Using process diagrams for the graphical representation of biological networks. *Nat. Biotechnol.*, **23**, 961–966.

Lu,T.K. and Collins,J.J. (2009) Engineered bacteriophage targeting gene networks as adjuvants for antibiotic therapy. *Proc. Natl Acad. Sci. USA*, **106**, 4629–4634.

Marchisio,M.A. and Stelling,J. (2009) Computational design tools for synthetic biology. *Curr. Opin. Biotechnol.*, **20**, 479–485.

Marner,W.D. (2009) Practical application of synthetic biology principles. *Biotechnol. J.*, **4**, 1406–1409.

Maziere,P. *et al*. (2004) A description scheme of biological processes based on elementary bricks of action. *J. Mol. Biol.*, **339**, 77–88.

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

Peres,S. *et al*. (2010) Computing biological functions using BioPsi, a formal description of biological processes based on elementary bricks of actions. *Bioinformatics*, **26**, 1542–1547.

Pryciak,P.M. (2009) Designing new cellular signaling pathways. *Chem. Biol.*, **16**, 249–254.

Ro,D.K. *et al*. (2006) Production of the antimalarial drug precursor artemisinic acid in engineered yeast. *Nature*, **440**, 940–943.

Strack,G. *et al*. (2008) Biocomputing security system: concatenated enzyme-based logic gates operating as a biomolecular keypad lock. *J. Am. Chem. Soc.*, **130**, 4234–4235.

TheUniProt Consortium (2010) The Universal Protein Resource (UniProt) in 2010. *Nucleic Acids Res*., **38**, D142–D148.

Zeke,A. *et al*. (2009) Scaffolds: interaction platforms for cellular signalling circuits. *Trends Cell Biol.*, **19**, 364–374.