Published in final edited form as:

Methods Enzymol. 2011 ; 498: 137–152. doi:10.1016/B978-0-12-385120-8.00006-1.

SynBioSS-Aided Design of Synthetic Biological Constructs

Yiannis N. Kaznessis

Department of Chemical Engineering and Materials Science, and Digital Technology Center,
University of Minnesota, Minneapolis, Minnesota, USA

Abstract

We present walkthrough examples of using SynBioSS to design, model, and simulate synthetic gene regulatory networks. SynBioSS stands for Synthetic Biology Software Suite, a platform that is publicly available with Open Licenses at www.synbioass.org. An important aim of computational synthetic biology is the development of a mathematical modeling formalism that is applicable to a wide variety of simple synthetic biological constructs. SynBioSS-based modeling of biomolecular ensembles that interact away from the thermodynamic limit and not necessarily at steady state affords for a theoretical framework that is generally applicable to known synthetic biological systems, such as bistable switches, AND gates, and oscillators. Here, we discuss how SynBioSS creates links between DNA sequences and targeted dynamic phenotypes of these simple systems.

1. Introduction

Synthetic biology is a quest to engineer new functions in living organisms. The scale of these new functions varies, from autonomous logical and informational architectures, such as bistable switches, oscillators, and logic gates (Alon, 2003; Anderson *et al.*, 2007; Andrianantoandro *et al.*, 2006; Drubin *et al.*, 2007; Elowitz and Leibler, 2000; Gardner *et al.*, 2000; Kærn *et al.*, 2003; Lutz and Bujard, 1997; Ramalingam *et al.*, 2009; Tiggles *et al.*, 2009), to modular cascades of metabolic reactions (Fung *et al.*, 2005; Ro *et al.*, 2006; Zhang *et al.*, 2010), to engineered ecosystems of multicellular systems (Basu *et al.*, 2004; Bulter *et al.*, 2004; You *et al.*, 2004), all the way up to minimal genomes and whole synthetic cells (Gibson *et al.*, 2008, 2010; Glass *et al.*, 2006; Pennisi, 2010).

Synthetic biology may be viewed as the flip side of systems biology: synthetic biology is a forward engineering approach, whereas systems biology is a reverse engineering one. The former attempts to assemble components into a new whole. The latter attempts to capture the behavior of existing biological systems in a holistic way. Their paths are complementing: systems biology generates information on components and interactions that can be used in synthetic biology applications. Synthetic biology can be employed to probe mechanisms and provide mechanistic insight on how phenotypic complexity emerges from interacting molecules. Importantly, their ultimate goals overlap: efforts in both synthetic and systems biology are aimed at understanding and engineering complexity of biomolecular systems. Indeed, the goal to provide mechanistic explanations of complex biological phenomena in terms of biomolecular interactions is commonly shared by both systems biology and synthetic biology.

As a forward engineering approach, synthetic biology may benefit from mathematical models. Modeling can assist synthetic biology the same way modeling helps in aircraft engineering or architecture design: models and computer simulations can relatively quickly provide a clear picture of how different components influence the behavior of the whole. They can provide mechanistic insight that may guide design choices and engineering implementations. In other words, they may help in probing the relationship between DNA sequences designed by the synthetic biologists and the observed behavior of the synthetic biological system. An important challenge in computational synthetic biology is thus to derive mathematical modeling formalisms that are fit for analysis and design of synthetic constructs. Numerous software packages have been developed to address this challenge, such as CellDesigner (Funahashi *et al.*, 2003), GenNetDes (Rodrigo *et al.*, 2007), COPASI (Hoops *et al.*, 2006), and TinkerCell (Chandran *et al.*, 2009) among others (de Jong, 2002; Kaznessis, 2007; Marchisio and Stelling, 2008).

One important and widely used viewpoint associates biological systems with computer programs: the DNA sequence encodes a set of instructions which is implemented in series and produces well-prescribed outputs. Although useful as a metaphor, by viewing synthetic constructs as sets of serial instructions, in a vain similar to computer programs, one resorts to assumptions regarding the system that may not be pertinent, or accurate.

Important assumptions may be the following:

1. the system is context-free and environment-independent. That is the system behaves in precisely the same manner, regardless of the biological context, or the cellular and extracellular environments;
2. the system is at equilibrium, or at best, at a steady state. In other words, time is not an independent variable and instead of time derivatives, simple algebraic equations may describe the system outputs as a function of system inputs;
3. the system behavior is determinate. Noise, whether intrinsic to the system, or extrinsic due to environmental factors, does not influence the system.

Epigenetic studies pose hard questions regarding the validity of the first question (Costa, 2008; Russo *et al.*, 1996), where observed phenotypic variation is ascribed to nongenetic factors. The second assumption is arguably of limited use in synthetic biology, where the goal is to engineer time-dependent responses of outputs, such as the flipping of a bistable switch, to well-defined time-profiles of inputs. The third assumption has been unequivocally proven wrong for biomolecular systems, with the study of cellular populations (Bagh *et al.*, 2008; Blake *et al.*, 2003; Rosenfeld *et al.*, 2005).

Instead of viewing synthetic biological constructs as computer programs, an alternative is to view them as soups of chemicals. These chemicals interact according to physicochemical principles dictated by statistical mechanics. When at equilibrium, they interact minimizing the free energy of the system. When away from the thermodynamic limit they interact in a probabilistic manner, with time a dominant independent variable. These chemicals and their interactions may be subject to their environment and to the biological context of the organism that carries them.

We have developed the Synthetic Biology Software Suite to implement this modeling approach (Hill *et al.*, 2008; Weeding *et al.*, 2010). An important aim for the continued development of SynBioSS is the link between synthetic DNA sequences and targeted biological functions. Herein, we present examples of using SynBioSS to build models of synthetic biological constructs, conduct computer simulations to study the dynamic behavior, and guide the experimental construction and testing of modular logical architectures in bacteria.

2. SynBioSS Components

There are three components in SynBioSS: *Designer*, *Wiki*, and *Simulator* (Fig. 6.1). With *SynBioSS Designer*, gene network models are created automatically after the user enters molecular components and their relationships. *SynBioSS Designer* is a web-based tool, available at <http://www.synbioss.org>, with a user-friendly interface which uses biological rules to build a network of biomolecular interactions. The software automatically generates a kinetic model from a construct composed entirely of biological “parts,” such as promoters and terminators.

While these parts can be hypothetical, chosen at will by the user, *Designer* is especially efficient at creating models for devices composed of BioBrick parts. BioBricks are synthetic DNA sequences catalogued in the Registry of Standard Biological Parts, a repository of synthetic biological constructs (Weeding *et al.*, 2010). A BioBrick standard biological part has “a nucleic acid-encoded biological function (e.g., turn on/off gene expression), along with associated information defining and describing the part” (Shetty *et al.*, 2008). The sequential ordering of these BioBricks (or “bricks”) therefore describes a sequence of DNA by its intended function within a cell.

SynBioSS Designer now has a database that is populated using information extracted from the official Parts Registry, but organized in a way that is machine-readable, allowing for structured queries. At present, this data is hosted locally at the Minnesota Supercomputing Institute.

Designer has a tabbed interface, making the complete sequence of BioBricks visually accessible and easily manipulated. Clicking on a tab pulls up properties of that individual brick and allows the user to add, edit, and delete said properties. Properties are also easy to edit; clicking directly on an editable field causes a text input field or drop-down menu will appear, allowing the user to make appropriate changes.

A user can enter biological components, including BioBricks, in *SynBioSS Designer*, and receive as an output a file with a reaction network that models the synthetic construct. Every reaction in the model has a corresponding kinetic rate that describes the rate of association of its reactant molecules and the formation or destruction of any covalent bonds or stable noncovalent interactions. *SynBioSS Wiki* has been specifically created to store and recall just this sort of kinetic data. *SynBioSS Wiki* has two components: (i) a web interface based on the MediaWiki package and (ii) a database for storing molecular components, their interactions, and pertinent biological information. *SynBioSS Wiki* goes beyond the MediaWiki software in storing kinetic information in a formatted (and therefore machine-

searchable) format. The database of kinetic constants is easily searchable for participating species, reaction type, etc. Users can search or browse the Web site and select reactions to interactively build a model that can be exported in a SBML format. Each kinetic constant entered in the database is correlated with a reference field in the database as well as type-specific reference information (pdb ID for proteins, CAS ID for small molecules, PubMed ID for everything, etc.).

Given the vast and varied nature of biochemical reaction data, no single person or research group is best suited for the task of curating such a database, thus necessitating this distributed approach—in spite of the accompanying challenges faced by any open Wiki approach, such as Wikipedia. To avoid abuse and vandalism, *SynBioSS* users are asked to register with a valid email address in order to make changes.

The third component, the *SynBioSS Desktop Simulator*, is a package that is currently available for Windows platforms. It includes multiscale algorithms appropriate for modeling reaction networks with multiple time scales away from the thermodynamic limit (Canton, 2008; Salis and Kaznessis, 2005a,b; Sotiropoulos and Kaznessis, 2008; Sotiropoulos *et al.*, 2009). *SynBioSS Desktop* has a Windows GUI interface, which can be used for constructing and editing gene network models, choosing simulation parameters and conducting numerical simulations. Version 1.0.2 of *SynBioSS* is currently available with Open Licenses on <http://www.synbio.org>. *SynBioSS* has evolved from HySSS (Salis *et al.*, 2006), a software package for modeling reaction networks, which has a Matlab GUI and FORTRAN codes available for UNIX and Linux platforms. HySSS is also available with Open Licenses at hysss.sourceforge.net.

In what follows, we will discuss the process of building a model of a synthetic gene network, conducting simulations, and guiding the design of synthetic biological systems.

3. Simulations of Biologic AND Gates

As a demonstration, we present the steps in *Designer* for building and simulating a logic AND gate, a synthetic biological construct we have also built and tested experimentally (Ramalingam *et al.*, 2009). The desired behavior of the AND gate gene network is to produce green fluorescent protein (GFP) if and only if two signal molecules are present. To achieve this, we have constructed a promoter sequence both *in vivo* and *in silico* that combines elements from the tetracycline operon and the lactose operon in prokaryotes (Ramalingam *et al.*, 2009).

The synthetic DNA promoter sequence consists of three operator sites, each approximately 20 base pairs in length, placed sequentially and adjacently upstream of the gene coding for GFP. In an ideal situation, if any of the operator sites are occupied, RNA polymerase cannot bind to the promoter region, and GFP is not produced. Operator sites can be selected so as to bind TetR protein (T), LacI protein (L), or neither (N). Additionally, TetR can be bound by the small molecule inducer anhydrous tetracycline (aTc), and LacI can be bound by isopropyl β -D-1-thiogalactopyranoside (IPTG). In this induced state, both TetR and LacI undergo conformational changes that cause them to unbind from their respective DNA operator sites. An AND gate can be constructed by selecting a promoter region containing

both T and L operator sites: only in the presence of both aTc and IPTG are both TetR and LacI induced, causing them to unbind from the promoter region, thereby allowing RNA polymerase to bind, resulting in the expression of the GFP reporter. If only one of the two inducers is present, the uninduced repressor protein will remain bound, and GFP expression will be repressed. This behavior is shown graphically in Fig. 6.2.

The following few steps will result in a model of the AND gate:

1. Go to www.synbio.org and click on *Designer*.
2. Enter parts in order (i.e., Promoter → RBS → DNA → Terminator). From a drop-down menu, characterize each of these parts, one at a time (see Fig. 6.3 for screenshots of Designer during these steps). These parts may be user-defined, or existing BioBricks. In this example, we will combine both, starting with promoter K091101 (a dually repressed promoter by TetR and LacI), adding a user-defined ribosome binding site, adding protein E0040 (the reporter gene of GFP), and finally adding BioBrick terminator sites.
3. Provide the name of the protein for each coding DNA region (e.g., Registry part E0040 is GFP), add and characterize other proteins (activator; repressor [TetR and LacI]; reporter [GFP]; enzyme; other). There is no need to use Registry names. Any arbitrary name will be sufficient.
4. Specify promoters as constitutively ON or OFF (all ON in this example).
5. If the operators were not prespecified in the promoter BioBrick, in this step add operators to promoters and specify their relative position. This is an important step for defining regulatory relationships. The synthetic AND-gate promoters have tetO and lacO, and are dually repressed by TetR and LacI.
6. Enter any proteins constitutively expressed. In the AND-gate example, these are TetR and LacI.
7. Specify protein oligomeric structure (monomer [GFP, RFP], dimer [TetR2], tetramer [LacI4]).
8. Specify where transcription factors bind (TetR2-tetO; LacI4-lacO).
9. Enter any relevant effector molecules (e.g., inducers) present in the system (aTc and IPTG).
10. Specify how many times each effector binds to a protein (two aTc can bind to TetR2; four IPTG to LacI4).

Finally, the user can click on a button to generate a reaction network with all the interactions and save the reaction network in SBML or NetCDF file format. *Designer* generates the reactions with default kinetic constants. These are taken from known biomolecular interactions, stored in *SynBioSS Wiki*, and applied to the various interaction types (e.g., RNAP binding on promoters, ribosome binding on RBS, protein dimerization, protein-operator, protein-effector).

We have stored this file along with other example reaction networks in <http://synbiooss.sourceforge.net/simulator/examples/>.

The files are ready to upload on *SynBioSS Desktop* to run numerical simulations. The user can also upload the file on *SynBioSS Wiki* and carefully check all the reactions and parameters, searching in the *Wiki* database for available information on any interaction. If there is no available information, the user can choose to retain the default value entered or conduct simulations over a range of parameter values for a sensitivity analysis.

Uploading the SBML or NetCDF file with the reaction network in *SynBioSS Desktop* also allows the user to visually and carefully check the reactions, the kinetic constants, and the initial conditions and run numerical simulations (Fig. 6.4 shows a screenshot of the reaction network editor of *SynBioSS Desktop*).

Simulating gene regulatory networks is now simple. With the third component of SynBioSS, the *Desktop Simulator*, a user can run sophisticated numerical simulations of complex reaction networks quickly and seamlessly on a PC. SynBioSS *Desktop Simulator* can be downloaded as an installation executable for Windows. The steps are:

1. Go to www.synbiooss.org.
2. Click on “Simulator” on the upper left corner. This will take you to <http://synbiooss.sourceforge.net/simulator/>.
3. Click on “Download” in the middle of the webpage. This will take you to the sourceforge file directory.
4. Click on SynBioSSDSInstaller-1.0.2.exe. This downloads the installation executable on your computer.
5. Run the executable. This will install the current version of SynBioSS on your computer.
6. Click on the Start Menu to find and click the SynBioSS icon. This will launch SynBioSS.

More than 60 reactions comprise the network of components used to simulate AND gates. All reactions are modeled as initially occurring in a well mixed volume of 10^{-15} L, which represents a cell. Cell growth is handled by allowing the reaction volume to double over a period of time (average of 60 min), followed by an instantaneous halving of volume to represent cytokinesis.

In our earlier work (Ramalingam *et al.*, 2009), we described in detail the results of simulations. Importantly, we experimentally constructed and tested six different AND-gate designs, shuffling the operator positions in the promoter: TTL, TLT, LTT, LLT, LTL, and TLL.

Each of these promoters was cloned in the backbone plasmid pGlow (Invitrogen) and transformed in a DH5aPRO *E. coli* strain, which constitutively expresses TetR and LacI from the chromosome. *In vivo* GFP, fluorescence was measured using a Becton Dickinson

FACS Calibur flow cytometer. Details on materials and methods can be found in reference (Ramalingam *et al.*, 2009).

To compare the simulation with the experimental results, we determined the average number of GFP molecules per cell at 6 h, averaging over 1000 stochastic simulation trajectories, and the average fluorescence strength at 6 h, averaging over 100,000 cytometry measurements. As an example, Fig. 6.5 presents the binary logic output of the TTL synthetic designs, both simulated and experimentally measured for the grid of 36 aTc/IPTG pair concentrations.

Let us first focus on the experimental results (Fig. 6.5, right panel). A high-fidelity logic AND gate will have high GFP expression levels only at high concentrations of both aTc and IPTG. It is clear that the TTL biological gate is not of perfect digital fidelity. We find that this is the case for all tested designs, because of leakiness of the promoters. This was actually expected, since biological dynamic response cannot be absolutely binary, because of thermal noise.

Although there are discernible differences between the modeling and the experimental results, the models generally capture the experimentally observed behavior well. The models then can explain the emergence of synthetic biological phenotypes in terms of biomolecular interactions that follow the molecular biology dogma and obey statistical thermodynamics.

It is important to note that a single model with 63 reactions captured the dynamic behavior of GFP distributions for six designs and 36 aTc-IPTG concentrations. The only one parameter that changed between designs was the kinetic constants of leakiness reactions. These were modeled with RNA polymerase binding on the promoter and initiating transcription, even if the promoter was occupied by repressor molecules bound on their cognate operators as discussed in (Ramalingam *et al.*, 2009).

What this study illustrated was the need for bidirectional passing of information from the models to the experiments. As we stress in (Ramalingam *et al.*, 2009) the first models we constructed did not include leakiness dependent on promoter-topology. After the first set of experiments, it became clear that reactions capturing and quantifying the leakiness were required, and that different values for the kinetic parameters would lead to correlation with the designed promoters. What was gained was useful insight into the importance of leakiness. And although significant computational resources are demanded, SynBioSS tools lower the barriers and streamline the process for setting up, modeling and analyzing the AND-gate systems.

4. Advantages and Disadvantages of SynBioSS

Certainly, the modeling methodology adopted in SynBioSS has numerous disadvantages:

1. There is dearth of quantitative information on biomolecular interactions. Such information is hard to come by, because time-consuming and expensive experiments are necessary, involving the isolation and purification of the interacting molecules in large enough quantities to measure accurately, and requiring sophisticated experimental techniques, for example, surface plasmon resonance.

Certainly, this is not the case for the tetracycline operon, thanks to enormous efforts expended by a large community of biochemists and molecular biologists. Other systems, such as the lactose, tryptophan, and arabinose operons have been studied thoroughly and a lot of information on them is available in the literature. But the absence of quantitative information on biomolecular interactions in other systems will hamper the efforts to use a detailed mechanistic representation of very many synthetic biological constructs.

2. Quantitative information is more often available for biomolecular interactions in the form of equilibrium constants. In such cases, we may assume that the forward rate of binding of a large protein to its DNA or RNA binding site is diffusion-limited, use the size of the protein to calculate its forward binding kinetic constant, and then use the equilibrium data to calculate the unbinding kinetic constant. An estimate may all that is needed to obtain useful insight on the behavior of a biological system, but ultimately only experimental information can reliably provide accurate information.
3. The mechanism of expression may not precisely follow the molecular biology dogma, or the precise steps and biomolecular interactions may not be known. For example, the role of antisense regulation has emerged as an important one even in bacteria. Or, nonspecific DNA interactions due to proteins binding nonspecifically to DNA may be important.
4. The context and the environment of some biomolecular interactions may be different in a control experiment from the actual one in a bacterial cell. This may change the mechanism itself, or at the very least, alter the values of the kinetic constants. For example, in some contexts the folding and maturation time of GFP can become exceptionally long. The mechanism itself, of a first order reaction may not be appropriate, and even if it were, the kinetic constant could be different that the one found in the literature. Other variations may be present and important, such as cellular size variation, or variable metabolic load of a synthetic system on the cells. Appreciation is then important of the context and environment experimental measurements are made, and of the transferability of mechanisms and kinetic constants.
5. We are currently limited to bacterial species. Synthetic biology efforts are being expended in more complex organisms, like yeast or mammalian cells. But the knowledge of biomolecular mechanisms, although far from perfect, is more complete for bacterial species. Consequently, it would be far more challenging to try to link phenotypic complexity to biomolecular interactions in more complex organisms, where the molecular biology mechanisms are under vigorous investigation.

On the other hand, the SynBioSS approach has important advantages:

1. It is a general method for constructing models of synthetic biological systems and thus applicable to any synthetic gene regulatory network. It can then be written in algorithmic form and serve as the heart of software tools to assist synthetic

biologists' designs. The method is general because molecular interactions between a transcription or translation factor and its DNA or RNA binding site are universal and context-free, that is the kinetics of the molecular interaction remains unchanged when the binding site is moved to a different location in the DNA sequence of the same organism.

2. It provides a detailed mechanistic picture of the dynamic behavior of biological systems. To our knowledge, this is the first attempt at the systematic modeling of all the known biomolecular interactions involved in bacterial transcription, translation, regulation, and induction. This approach certainly challenges established molecular biology and in the absence of agreement between models and experiments it poses new questions and requires new avenues of investigation.
3. It has a strong predictive character, enabling rational engineering of regulatable gene transcription systems. Rational design principles come in terms of molecular components, the kinetics and the thermodynamics of their interaction. With simply built models, alternative designs can be tested and a detailed picture can emerge of how each piece of the construct influences the synthetic network behavior. Sensitivity analysis and optimization can be conducted to determine key components and decide on network topologies. Computer simulations make possible exhaustive searches of different network connectivities and molecular thermodynamic/kinetic parameters, greatly advancing the development of design principles through the mapping of interaction strengths on specific DNA mutant sequences.
4. This approach of constructing dynamic models of all the biomolecular interactions involved in gene expression and regulation pushes the limits of computational mathematics. Because of the large number of participating species and the complexity of their interactions, only sophisticated algorithms can accurately capture dynamic gene expression in a way fit for analysis and design.
5. This approach is well-suited for synthetic biology. Although numerically challenging, it always remains tractable, not hampered by the significant size and complexity of naturally occurring biological systems. Synthetic biology modeling efforts concentrate on systems that are not overwhelmingly large, under the assumption that they are independent of the bacterial expression and metabolism machinery. Importantly, since these systems are engineered, the synthetic biologist can choose to include molecular components for which there is ample quantitative information and refine the models in a careful, context-dependent manner (Salis and Kaznessis, 2005c, 2006; Tomshine and Kaznessis, 2006; Tuttle *et al.*, 2005). Returning to the AND-gate example, it quickly becomes clear that there are many possible designs that could achieve the desired behavior. The order and the number of tetO and lacO operator sites can be varied—for instance, one possible promoter consists of two tetO operator sites followed by a lacO operator site (designated as TTL), and another consists of two lacO operators followed by a tetO operator (designated as LLT). Another possible design parameter is the particular DNA sequence of the operator sites (TetO and LacO), which can be mutated to vary the

strength of binding between the repressor proteins TetR and LacI and their cognate operator sites. With mechanistic models these changes can be incorporated and tested in a simulation, before the experiments begin.

6. This approach also pushes the limits of quantitative biology, motivating the collection and employment of quantitative information regarding molecular mechanisms, biomolecular interactions, and their kinetic and equilibrium constants that is currently scattered throughout the literature. To collect information for components of well-studied systems, such as the tetracycline, lactose, and arabinose operons, that are widely used as parts in synthetic biological systems, we are also building software tools to facilitate the collection of this information by the synthetic biology community into a publicly available repository.

5. Concluding Remarks

Synthetic biology has all the characteristic features of an engineering discipline: applying technical and scientific knowledge to design and implement devices, systems, and processes that safely realize a desired objective.

Mathematical modeling has always been an important component of engineering disciplines. It can play an important role in synthetic biology the same way modeling helps in aircraft or architecture design: models and computer simulations can quickly provide a clear picture of how different components influence the behavior of the whole, reaching objectives quickly.

Here, we discussed a modeling methodology that may help scientists and engineers to construct complex synthetic biological systems. We are developing sophisticated mathematical models of synthetic biological systems that connect the targeted biological phenotype (what we want the synthetic biological system to do) to the DNA sequence (that we need to physically construct to realize the synthetic biological system). Using these mathematical models, we can conduct simulations of many alternate designs to decide on the optimum set of components before synthesizing and testing the designs in the wet lab. Of course, identifying and constructing a few actual designs early in the synthesis process will better guide the model construction itself. Again, the importance of combining theory and experiment can hardly be overstated. The numerical methods and software tools presented herein are standardized, so that the process for generating models of synthetic gene regulatory networks is applicable to any synthetic construct and is suitable for automation. Consequently, SynBioSS represents a first step toward the direction of an automated design process, while assisting in the much broader objective: to develop theoretical and computational models that describe how the physical interactions of molecules lead to complex biological phenotypes.

Acknowledgments

This work was supported by a grant from the National Science Foundation (CBET-0425882 and CBET-0644792), the National Institutes of Health (American Recovery and Reinvestment Act grant R01GM086865), and the University of Minnesota Biotechnology Institute. Computational support from the Minnesota Supercomputing Institute (MSI) is gratefully acknowledged. This work was also supported by the National Computational Science Alliance under TG-MCA04N033. In addition, the author thanks Jon Tomshine, Ben Swiniarski, and Kostas Biliouris for help with the illustrations.

References

- Alon U. Biological networks: The tinkerer as an engineer. *Science*. 2003; 301:1866–1867. [PubMed: 14512615]
- Anderson JC, Voigt CA, Arkin AP. Environmental signal integration by a modular AND gate. *Mol Syst Biol*. 2007; 3:133. [PubMed: 17700541]
- Andrianantoandro E, Basu S, Karig DK, Weiss R. Synthetic biology: New engineering rules for an emerging discipline. *Mol Syst Biol*. 2006; 2:0028. [PubMed: 16738572]
- Bagh S, et al. Plasmid-borne prokaryotic gene expression: Sources of variability and quantitative system characterization. *Phys Rev E Stat Nonlin Soft Matter Phys*. 2008; 77:021919. [PubMed: 18352063]
- Basu S, Mehreja R, Thiberge S, Chen MT, Weiss R. Spatiotemporal control of gene expression with pulse-generating networks. *Proc Natl Acad Sci USA*. 2004; 101:6355–6360. [PubMed: 15096621]
- Blake WJ, Kærn M, Cantor CR, Collins JJ. Noise in eukaryotic gene expression. *Nature*. 2003; 422:633–637. [PubMed: 12687005]
- Bulter T, et al. Design of artificial cell–cell communication using gene and metabolic networks. *Proc Natl Acad Sci USA*. 2004; 101:2299–2304. [PubMed: 14983004]
- Canton B, et al. Refinement and standardization of synthetic biological parts and devices. *Nat Biotech*. 2008; 26:787–793.
- Chandran D, Bergmann FT, Sauro HM. TinkerCell: Modular CAD tool for synthetic biology. *J Biol Eng*. 2009; 3:19. [PubMed: 19874625]
- Costa FF. Non-coding, RNAs, epigenetics and complexity. *Gene*. 2008; 410:9–17. [PubMed: 18226475]
- de Jong H. Modeling and simulation of genetic regulatory systems: A literature review. *J Comput Biol*. 2002; 9:67–103. [PubMed: 11911796]
- Drubin DA, Way JC, Silver PA. Designing biological systems. *Genes Dev*. 2007; 21:242–254. [PubMed: 17289915]
- Elowitz MB, Leibler S. A synthetic oscillatory network of transcriptional regulators. *Nature*. 2000; 403:335–338. [PubMed: 10659856]
- Funahashi A, Morohashi M, Kitano H, Tanimura N. Cell Designer: A process diagram editor for gene-regulatory and biochemical networks. *Biosilico*. 2003; 1:159–162.
- Fung E, Wong WW, Suen JK, Bulter T, Lee SG, Liao JC. A synthetic gene-metabolic oscillator. *Nature*. 2005; 435:118–122. [PubMed: 15875027]
- Gardner TS, Cantor CR, Collins JJ. Construction of a genetic toggle switch in *Escherichia coli*. *Nature*. 2000; 403:339–342. [PubMed: 10659857]
- Gibson DG, et al. Complete chemical synthesis, assembly, and cloning of a *Mycoplasma genitalium* genome. *Science*. 2008; 319:1215–1220. [PubMed: 18218864]
- Gibson DG, et al. Creation of a bacterial cell controlled by a chemically synthesized genome. *Science*. 2010; 329:52–56. [PubMed: 20488990]
- Glass JI, et al. Essential genes of a minimal bacterium. *Proc Natl Acad Sci USA*. 2006; 103:425. [PubMed: 16407165]
- Hill AD, Tomshine JR, Weeding EM, Sotiropoulos V, Kaznessis YN. SynBioSS: The synthetic biology modeling suite. *Bioinformatics*. 2008; 24:2551–2553. [PubMed: 18757873]
- Hoops S, Sahle S, Gauges R, Lee C, Pahle J, Simus N, Singhal M, Xu L, Mendes P, Kummer U. COPASI—A COMplex PATHway SIMulator. *Bioinformatics*. 2006; 22:3067–3074. [PubMed: 17032683]
- Kærn M, Blake WJ, Collins JJ. The engineering of gene regulatory networks. *Annu Rev Biomed Eng*. 2003; 5:179–206. [PubMed: 14527313]
- Kaznessis Y. Models for synthetic biology. *BMC Syst Biol*. 2007; 1:47. [PubMed: 17986347]
- Lutz R, Bujard H. Independent and tight regulation of transcriptional units in *Escherichia coli* via the LacR/O, the TetR/O and AraC/I1-I2 regulatory elements. *Nucleic Acids Res*. 1997; 25:1203. [PubMed: 9092630]

- Marchisio M, Stelling J. Computational design of synthetic gene circuits with composable parts. *Bioinformatics*. 2008; 24:1903–1910. [PubMed: 18579565]
- Pennisi E. Synthetic genome brings new life to bacterium. *Science*. 2010; 328:958. [PubMed: 20488994]
- Ramalingam KI, Tomshine J, Maynard JA, Kaznessis YN. Forward engineering of synthetic biological AND gates. *Biochem Eng J*. 2009; 47:38.
- Ro DK, et al. Production of the antimalarial drug precursor artemisinic acid in engineered yeast. *Nature*. 2006; 440:940–943. [PubMed: 16612385]
- Rodrigo G, Carrera J, Jaramillo A. Genetdes: Automatic design of transcriptional networks. *Bioinformatics*. 2007; 23:1857–1858. [PubMed: 17485427]
- Rosenfeld N, Young JW, Alon U, Swain PS, Elowitz MB. Gene regulation at the single-cell level. *Science*. 2005; 307:1962–1965. [PubMed: 15790856]
- Russo, VEA.; Martienssen, RA.; Riggs, AD. *Epigenetic Mechanisms of Gene Regulation*. Cold Spring Harbor Laboratory Press; Plainview, NY: 1996.
- Salis H, Kaznessis YN. Accurate hybrid stochastic simulation of a system of coupled chemical or biochemical reactions. *J Chem Phys*. 2005a; 122:054103, 1–13.
- Salis H, Kaznessis YN. An equation-free probabilistic steady state approximation: Dynamic application to the stochastic simulation of biochemical reaction networks. *J Chem Phys*. 2005b; 123:214106. [PubMed: 16356038]
- Salis H, Kaznessis YN. Stochastic simulations of gene regulatory modules. *Comput Chem Eng*. 2005c; 29:577–588.
- Salis H, Kaznessis YN. Computer-aided design of modular protein devices: Boolean AND gene activation. *Phys Biol*. 2006; 3:295–310. [PubMed: 17200605]
- Salis H, Sotiropoulos V, Kaznessis YN. Multiscale Hy3S: Hybrid stochastic simulations for supercomputers. *BMC Bioinform*. 2006; 7:93.
- Shetty RP, Endy D, Knight TF Jr. Engineering BioBrick vectors from BioBrick parts. *J Biol Eng*. 2008; 2:5. [PubMed: 18410688]
- Sotiropoulos V, Kaznessis YN. An adaptive time step scheme for a system of SDEs with multiple multiplicative noise. Chemical Langevin equation, a proof of concept. *J Chem Phys*. 2008; 128:014103. [PubMed: 18190181]
- Sotiropoulos V, Contou-Carrere MN, Daoutidis P, Kaznessis YN. Model reduction of multiscale chemical Langevin equations: A numerical case study. *IEEE/ACM Trans Comp Biol Bioinf*. 2009; 6:470.
- Tigges M, Marquez-Lago T, Stelling J, Fussenegger M. A tunable synthetic mammalian oscillator. *Nature*. 2009; 457:309–312. [PubMed: 19148099]
- Tomshine J, Kaznessis YN. Optimization of a stochastically simulated gene network model via simulated annealing. *Biophys J*. 2006; 91:3196–3205. [PubMed: 16920827]
- Tuttle L, Salis H, Tomshine J, Kaznessis YN. Model-driven design principles of gene networks: The oscillator. *Biophys J*. 2005; 89:3873–3883. [PubMed: 16183880]
- Weeding E, Houle J, Kaznessis YN. SynBioSS Designer: A web-based tool for the automated generation of kinetic models for synthetic biological constructs. *Brief Bioinform*. 2010; 11:394–402. [PubMed: 20639523]
- You L, Cox RS 3rd, Weiss R, Arnold FH. Programmed population control by cell–cell communication and regulated killing. *Nature*. 2004; 428:868–871. [PubMed: 15064770]
- Zhang K, Li H, Cho K, Liao JC. Expanding metabolism for total biosynthesis of the nonnatural amino acid L-homoalanine. *Proc Natl Acad Sci USA*. 2010; 107:6234–6239. [PubMed: 20332210]

Figure 6.1.
Schematic representation of SynBioSS components.

Figure 6.2.
Schematic representation of the synthetic logic-AND gate promoter.

Figure 6.3. Screenshots of consecutive SynBioSS Designer web-pages depicting user inputs of a synthetic logic-AND gate. Top: the first step is to add the components of the synthetic sequence, either by searching for BioBricks or by adding user-defined components. Bottom left: in the second step, regulatory proteins are added as needed and protein–DNA binding events defined. Bottom right: Effector molecules are added and protein-effector interactions defined.

Figure 6.4.

Screenshot of SynBioSS Desktop. With a graphics user interface, users can manipulate and simulate reaction network models generated by SynBioSS Designer.

Figure 6.5.

Comparison of model and experimental results for the TLT AND gate. The x and y axes form a grid of inducer concentrations: aTc (0–200 ng/ml) and IPTG (0–1 mM). The color scheme reflects the average strength of fluorescence from the experiments or the average number of GFP molecules in the simulations, scaled by the maximum strength/number of GFP molecules. In all cases, behavior is depicted 6 h after induction. The plotted model values are the means of 1000 independent stochastic kinetic simulations, whereas experimental values are the means of 100,000 FACS observations.