SynBioSS designer: a web-based tool for the automated generation of kinetic models for synthetic biological constructs

Emma Weeding, Jason Houle and Yiannis N. Kaznessis

Submitted: 27th October 2009; Received (in revised form): 12th January 2010

Abstract

Modeling tools can play an important role in synthetic biology the same way modeling helps in other engineering disciplines: simulations can quickly probe mechanisms and provide a clear picture of how different components influence the behavior of the whole. We present a brief review of available tools and present SynBioSS Designer. The Synthetic Biology Software Suite (SynBioSS) is used for the generation, storing, retrieval and quantitative simulation of synthetic biological networks. SynBioSS consists of three distinct components: the Desktop Simulator, the Wiki, and the Designer. SynBioSS Designer takes as input molecular parts involved in gene expression and regulation (e.g. promoters, transcription factors, ribosome binding sites, etc.), and automatically generates complete networks of reactions that represent transcription, translation, regulation, induction and degradation of those parts. Effectively, Designer uses DNA sequences as input and generates networks of biomolecular reactions as output. In this paper we describe how Designer uses universal principles of molecular biology to generate models of any arbitrary synthetic biological system. These models are useful as they explain biological phenotypic complexity in mechanistic terms. In turn, such mechanistic explanations can assist in designing synthetic biological systems. We also discuss, giving practical guidance to users, how Designer interfaces with the Registry of Standard Biological Parts, the de facto compendium of parts used in synthetic biology applications.

Keywords: synthetic biology; computational biology; multiscale models; automated design

INTRODUCTION

Humans can now construct and piece together DNA sequences to design new biological systems and organisms. We can do this more quickly and less expensively than ever. Applications abound for our synthetic biological constructs, from inexpensive therapeutics to environmentally friendly biofuels. Synthetic biology focuses on the construction of novel biological systems, and mathematical modeling is becoming an integral component of synthetic biology efforts, capturing the behavior of such synthetic biological constructs and guiding the design of new ones. Numerous software packages exist for modeling biomolecular systems, such as CellDesigner [1],

Corresponding author. Yiannis N. Kaznessis, Department of Chemical Engineering and Materials Science, University of Minnesota, 421 Washington Ave. S.E., Minneapolis, MN 55455-0132, USA. Tel: +1-612-624-4197; Fax: +1-612-626-7246; E-mail: yiannis@cemis.umn.edu

Emma Weeding is a senior undergraduate student at the Department of Chemical Engineering and Materials Science at the University of Minnesota. Emma interned at the University of Minnesota Bioinformatics Summer Institute in the summers of 2007, 2008 and 2009. She has been developing and maintaining SynBioSS Designer.

Jason Houle is a senior undergraduate student in the Department of Chemical Engineering and Materials Science at the University of Minnesota. Jason interned at the University of Minnesota Bioinformatics Summer Institute in the summer of 2009.

Yiannis N. Kaznessis is Associate Professor in the Department of Chemical Engineering and Materials Science at the University of Minnesota. His research interests focus on computer modeling of biological matter.

The overarching goal of modelers is to connect the DNA sequence of a synthetic biological construct to its behavior. Presumably, the synthetic biologist starts with a targeted phenotype and proceeds with designing a DNA sequence that when expressed gives rise to this phenotype. For example, the synthetic biologist may be interested in constructing a synthetic gene network that exhibits oscillatory behavior. The challenge in this forward engineering task lies with the choice and assembly of DNA sequences. Which molecular components to choose and how to sequence them together? Pioneering work has already demonstrated how to do this [13–16]. The choices for different, alternative designs can now be overwhelming though. And this difficulty becomes even more apparent when entirely novel constructs are to be synthesized.

There is an opportunity then for the development of computer software tools that can tackle the challenges facing the synthetic biology community. Computer models of new gene networks can, in principle, shift through alternative designs and propose synthetic construction approaches before the synthetic biologist begins work in the wet lab.

Despite some progress in the development of computer models, there are gaps in the process of connecting DNA sequences to targeted phenotypes using software tools. For example, there are no universally accepted methods for representing synthetic biomolecular systems. Should all of the molecular components be included (e.g. promoters, operators, ribosome binding sites, RNA polymerase, ribosomes, among others), or is a more reduced representation more appropriate (e.g. DNA gives RNA gives protein, which regulates DNA)? Is the dynamic behavior important, or are steady-state approximations? And can there be a modeling formalism that is applicable to a wide variety of synthetic construct and amenable to automation?

In this manuscript we present a standardized algorithmic process for generating models of synthetic gene regulatory networks that is applicable to any synthetic construct and is suitable for automation. We recently published the Synthetic Biology Software Suite, a software package that puts these modeling tools at the fingertips of synthetic biology practitioners [17]. SynBioSS is publicly available on synbioss.sourceforge.net. With SynBioSS, any synthetic biologist can quickly model and simulate the dynamic behavior of any new synthetic biological system.

SynBioSS has three main components: Designer, Wiki and Desktop Simulator. Previously, we discussed the statistical mechanical methods underlying the algorithms in the Desktop Simulator, which can be used to simulate networks of reactive systems away from the thermodynamic limit [18–25]. We also presented the Wiki, which is a repository of biochemical reactions with relevant kinetic constants [17]. In this manuscript, we focus on Designer.

SynBioSS Designer automatically generates models of synthetic constructs. Given a set of molecular components in a gene network (e.g., promoters, transcription factors, reporter genes), Designer applies universal molecular biology principles to build a set of biochemical reactions, including all reactions in transcription, translation, regulation, and induction.

The construction of the model itself is an important step in mathematically representing synthetic biological systems. When done manually, this quickly becomes an arduous and error-prone task, even for the simplest of biological systems. With Designer the construction is automated and generalized for any synthetic construct.

ELEMENTS OF SYNBIOSS DESIGNER

SynBioSS Designer is a web-based tool, available at http://synbioss.sourceforge.net/, 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.

What is notable is that although 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 [26]. 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’ [27]. 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 connection to the official Registry of Standard Biological Parts, through the Standard Biological Parts Web Service [28]. This database 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 by Google Base [29], a free service provided by Google for publicly indexing and hosting databases.

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 BioBricks. With the database of BioBricks, Designer can be used to streamline model construction. All information is now automatically, quickly, and accurately retrieved, added to the current Designer construct, and displayed, ready to be edited if necessary.

In Figure 1, a schematic representation is presented of the information a user enter in SynBioSS Designer, of the steps in Designer’s algorithm, and of the output reaction network. In the following section we present an illustrative example of how to use Designer to generate models of synthetic biological constructs. More information on the algorithmic steps of Designer is presented in the Supplementary Material, and is available on synbios.sourceforge.net, along with tutorials with additional example cases.

**MODELING BIOBRICKS**

In this section we present an illustrative example of how to use Designer to generate networks of reactions that model synthetic biological constructs. The example emphasizes the connection with the Standard Biological Parts Web Service. Another example is presented in the Supplementary Material that illustrates how a user-defined synthetic device can be created.

Consider the composite BioBrick BBa_T9002, a GFP Producer controlled by 3OC6HSL Receiver Device [27]. For brevity, all BioBrick IDs will hereafter be referred to without the ‘BBa_’ prefix, e.g. ‘R0040’ instead of ‘BBa_R0040’.

This composite part is divided into two transcriptional units: the first begins with a TetR repressible promoter (R0040) which regulates the expression of LuxR (C0062), and the second beginning with a LuxR activated promoter (R0062) which controls production of GFP (E0040). Also included are several ribosome binding sites (B0034 and B0032) and terminators (B0010 and B0012). As a whole, this series of parts functions as an AND gate, where GFP is the output, and HSL and aTc are the inputs. The latter input causes a conformational change in TetR, preventing its repression of promoter R0040 and thus inducing the production of LuxR. Similarly, HSL produces a conformational change...
in LuxR, allowing it to activate promoter R0062 and produce GFP. Therefore, unless both aTc and HSL are present in the system, no GFP will be produced.

Designer requires the input and specification of (i) the sequence of parts, (ii) all proteins in the system and (iii) all effector molecules in the system. The brick sequence is shown in Figure 2. Proteins in the system include LuxR, GFP, and the constitutively expressed TetR. Effectors in the system are aTc and HSL. Next, we detail how a user communicates these specifics to Designer.

**Entering and manipulating BioBricks**

The Designer webpage opens to display the ‘Add Brick’ page, which offers three different methods of adding a BioBrick.

If the user knows which BioBrick he/she wishes to add, the BioBrick’s ID can be entered into the ‘BioBrick Search’ field. Adding a brick from the Parts Registry can be accomplished in two ways. Individual bricks, such as the ten that form T9002, may be added manually, one at a time. Alternatively, addition of individual composite bricks such as T9002 can be performed using only one query by searching for the composite brick ID itself (Figure 3). Designer runs through a list of constituent bricks in the composite and adds each one progressively.

Designer retrieves and displays the sequence and features of all components of BBa_T9002 (Figure 4). Now the user can click on the tab of each component, access the component’s specific properties and change them if necessary.

All information on the brick besides its ID and type may be edited. Any changes to the default values pulled from the Parts Registry are highlighted with red text. If errors are made during input, they may be undone by clicking the gray, underlined ‘rewind’ symbol directly to the right of the property. Doing so resets the property to its original value, assuming one exists. Clicking the ‘Reset Brick’ button at the bottom of the page resets all the properties of the brick.

In the present example, the default names of the R0040 operator sites are ‘TetR_1’ and ‘TetR_2’. Such notation may become confusing due to the similarly named protein dimer ‘TetR2’. We rename the operator sites to ‘tetO1’ and ‘tetO2’.

Using the buttons at the bottom of the tab, bricks may also be moved within the construct. Clicking ‘Shift Brick Right’ moves R0040 to the second position in the construct, while clicking the newly-available ‘Shift Brick Left’ button moves it back to its proper location. Finally, the user may delete a brick from the construct by clicking the ‘Delete Brick’ button.

Promoters may be designated as constitutively ‘ON’ or ‘OFF’. For this system, the promoter should be considered constitutively ‘OFF’, as it is only activated when bound by a complex of LuxR and HSL. Alternatively, the user may leave the brick constitutively ‘ON’ and instead reduce the kinetic constants for basal transcription in the final model.

While checking promoters is recommended, viewing and editing Coding DNA regions is necessary. This is because protein type (e.g. activator, repressor) must be decided by the user. Additionally, if a custom Coding DNA region is added, its protein does not have a name. Clicking on the C0062 tab, the user is able to view and edit its specifics, and is directed to adjust the protein’s type by the red text ‘Please select type’. Within the

![Figure 2: Schematic representation of BioBrick BBa.T9002.](image-url)

![Figure 3: Schematic of SynBioSS Designer web-interface. Users can enter any BioBrickID.](image-url)
example construct, LuxR serves as an activator, which is selected in the drop-down menu.

Clicking on E0040 allows the user to select the protein’s type as ‘Reporter,’ as well as to shorten its name to simply ‘GFP’ from ‘GFP_protein’.

Checking ribosome binding sites and terminators is possible but unnecessary, as Designer assigns these bricks no properties. Once all properties are set as desired and all the bricks are in the correct order, the user may click ‘Continue to 2/3’ at the very bottom (below the tabbed display) in order to more completely describe the proteins in the system.

**Protein input and specifics**

Once the construct and its specifics have been entered, the second step is to input any additional proteins, if any, as well as to describe the activity of all proteins. Since LuxR and GFP are produced by the construct, the only additional protein present in this example system is the constitutively expressed TetR. This is added in the ‘Input Proteins’ field by entering its name and selecting its type (Repressor).

Once added, TetR appears in the ‘Current Proteins’ table at the bottom of the page, and becomes available in the dropdown menus in the remaining fields.

The next field, ‘Complex Specifics’, is required. Generally, regulatory proteins form an active complex before binding to DNA. The number of subunits in these complexes must be specified for every protein. In the example system, both LuxR and TetR dimerize, forming complexes referred to as ‘LuxR2’ and ‘TetR2’. This is described within Designer by selecting the appropriate protein from the dropdown menu, typing the number of subunits into the appropriate field, and clicking ‘Add Complex Info’. Note that this must also be carried out for GFP, which does not form a complex, by entering a 1 into the field (meaning it exists only as a monomer and/or it forms no complex of interest).

Finally, protein interactions with operator sites must be described in the ‘Binding Specifics’ field. This is accomplished simply by selecting matching pairs of proteins and operators in the two dropdown boxes, then clicking ‘Add Binding Info’. For example, TetR binds with both sites ‘tetO1’ and ‘tetO2’ on promoter R0040. TetR is selected in the first dropdown menu, and tetO1 is selected in the second. Once submitted, the selected operator appears below ‘Binds’ in the selected protein’s row in the ‘Current Proteins’ table. The user may select as many protein/operator binding pairs as desired.

Figure 5 displays the appearance of the ‘Current Proteins’ table immediately before the final binding specification, which describes LuxR binding to the luxO1 operator site. Once that information is added, the system is ready for addition of effector information in step 3, which is navigated to by clicking ‘Continue to 3/3’.

**Effector input and specifics**

Effector molecules must be individually added by typing the effector’s name in the ‘Input Effectors’ field and clicking ‘Add Effector’. Recall that the effectors in this example system are aTc and HSL. Once at least one effector is added, it appears in a ‘Current Effectors’ table at the bottom of the page, and also brings up an ‘Effector Specifics’ field in the middle of the page.

The user must also specify how many times an effector may bind to a protein complex. In the example system, aTc may bind to the TetR dimer.
a maximum of two times. To communicate this information to Designer, the user selects aTc in the effector menu and TetR in the protein menu, and then enters ‘2’ as the ‘Max Effectors per Complex’. Likewise, a maximum of 2 molecules of HSL may bind to the LuxR dimer.

The LuxR/HSL complex requires additional input, as only this protein-effector complex binds DNA; a LuxR dimer by itself does not. Essentially, LuxR and HSL act in concert to bind DNA, thus this pairing must be selected in the second set of boxes and submitted by clicking ‘Act in Concert’. The effector specification is then complete.

Designer now has enough information to describe the entire system in terms of a reaction network and component molecules and entities. The user finally has the option of generating this as a NetCDF file or an SBML file. The contents of this file are outlined in Section 5.1.

**DESIGNER OUTPUT**

The output is a network of reactions representing all the steps in gene expression and regulation, shown in Table 1. For simulation purposes, besides the reactions, the initial and the environmental conditions are required. Designer assigns initial amounts and a ‘split on cell division’ property to all species. A summary of the default values is shown as follows:

<table>
<thead>
<tr>
<th>Species type</th>
<th>Initial amount</th>
<th>Split on cell division</th>
</tr>
</thead>
<tbody>
<tr>
<td>mRNA</td>
<td>0</td>
<td>Y</td>
</tr>
<tr>
<td>Promoters &amp; Operators</td>
<td>1</td>
<td>N</td>
</tr>
<tr>
<td>Proteins</td>
<td>0</td>
<td>Y</td>
</tr>
<tr>
<td>Ribosome</td>
<td>600</td>
<td>N</td>
</tr>
<tr>
<td>RNAP</td>
<td>300</td>
<td>N</td>
</tr>
<tr>
<td>Other</td>
<td>0</td>
<td>N</td>
</tr>
</tbody>
</table>

Additionally, all species are designated to be saved to a file upon simulation, and the initial volume of the system is $1e-15$ l by default. All reactions in the following table have elementary rates laws with kinetic constants in terms of moles, liters, and seconds.

Along with this reaction network, Designer also assigns reasonable default kinetic constants to each reaction. These constants are not customized to the specific system, however, and as such must be retrieved manually from SynBioSS Wiki or the relevant literature. Designer outputs either a NetCDF or SBML file, which can then be loaded in simulation software of the user’s choice, such as SynBioSS Desktop Simulator. The network of reactions can be simulated in time to decipher how the biological phenotype emerges as a result of biomolecular interactions.

**CURRENT LIMITATIONS**

SynBioSS Designer has numerous limitations. Herein, we discuss the most important ones.

First, the lack of available quantitative information on biomolecular interactions hampers the construction of realistic models. Designer currently assigns only representative values for kinetic constants, employing information from well-studied interactions, such as the ones found in the tetracycline operon. These kinetic constant values will most probably not be accurate for other systems. Searching for available kinetic value constants is a tedious, if not futile, effort, which is not currently facilitated by Designer. Our hope is that as quantitative information becomes increasingly available, storing this information in SynBioSS Wiki, the third component of SynBioSS, may generate a compendium of reactions and relevant kinetic constants. Although we are currently far from it, a goal of ours is to better integrate Designer with the Wiki, so that an automatic search of Wiki components, reactions and kinetic constants is conducted by Designer to match the parsed components of BioBricks.

Second, strictly following the molecular biology dogma restricts the use of alternative representations of expression mechanisms. Antisense regulation, transcriptional inhibition, RNA silencing are but a few of molecular mechanisms that may be encoded in a BioBrick. Currently, Designer cannot take into account any but the simplest of steps in protein...
### Table 1: Reaction network for BBaT9002

<table>
<thead>
<tr>
<th>Protein multimerization</th>
<th>Kinetic data</th>
<th>Non-specific DNA interactions</th>
<th>Kinetic data</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 TetR → TetR2</td>
<td>10000000000</td>
<td>TetR2 + nsDNA → TetR2;nsDNA</td>
<td>1000</td>
</tr>
<tr>
<td>TetR2 → 2 TetR</td>
<td>0</td>
<td>TetR2;nsDNA → TetR2 + nsDNA</td>
<td>1.6225</td>
</tr>
<tr>
<td>2 LuxR → LuxR2</td>
<td>10000000000</td>
<td>TetR2;αc + nsDNA → TetR2;αc;nsDNA</td>
<td>1000</td>
</tr>
<tr>
<td>LuxR2 → 2 LuxR</td>
<td>0</td>
<td>TetR2;αc;nsDNA → TetR2;αc + nsDNA</td>
<td>1.6225</td>
</tr>
<tr>
<td>Transcription</td>
<td></td>
<td>TetR2;αc + nsDNA + αTc</td>
<td>1.6225</td>
</tr>
<tr>
<td>RNAp + BBA.R0040 + tetO2 + tetOl → RNAp + BBA.R0040 + tetO2 + tetOl</td>
<td>0.066</td>
<td>TetR2;αc + nsDNA + αTc → TetR2;αc + nsDNA</td>
<td>1.6225</td>
</tr>
<tr>
<td>RNAp + BBA.R0040 + tetO2 + tetOl → RNAp + BBA.R0040 + tetO2 + tetOl</td>
<td>0.75</td>
<td>RNAp + BBA.R0040 + tetO2 + tetOl</td>
<td>0.066</td>
</tr>
<tr>
<td>RNAp + BBA.R0040 + tetO2 + tetOl → RNAp + BBA.R0040 + tetO2 + tetOl</td>
<td>3</td>
<td>RNAp + BBA.R0040 + tetO2 + tetOl</td>
<td>0.75</td>
</tr>
<tr>
<td>RNAp + BBA.R0040 + tetO2 + tetOl → RNAp + BBA.R0040 + tetO2 + tetOl</td>
<td>30</td>
<td>RNAp + BBA.R0040 + tetO2 + tetOl</td>
<td>3</td>
</tr>
<tr>
<td>RNAp + BBA.R0040 + tetO2 + tetOl → RNAp + BBA.R0040 + tetO2 + tetOl</td>
<td>30</td>
<td>RNAp + BBA.R0040 + tetO2 + tetOl</td>
<td>30</td>
</tr>
<tr>
<td>Translation</td>
<td></td>
<td>RNAp + BBA.R0040 + LuxR2.HSL2:luxO1</td>
<td>30</td>
</tr>
<tr>
<td>rib + mRNA.LuxR → rib + mRNA.LuxR</td>
<td>10000</td>
<td>RNAp + BBA.R0040 + LuxR2.HSL2:luxO1</td>
<td>30</td>
</tr>
<tr>
<td>ribmRNA.LuxR → rib + mRNA.LuxR</td>
<td>33</td>
<td>RNAp + BBA.R0040 + LuxR2.HSL2:luxO1</td>
<td>30</td>
</tr>
<tr>
<td>ribmRNA.LuxR + rib + LuxR</td>
<td>33</td>
<td>RNAp + BBA.R0040 + LuxR2.HSL2:luxO1</td>
<td>30</td>
</tr>
<tr>
<td>ribmRNA.GFP → rib + mRNA.GFP</td>
<td>33 aa/s</td>
<td>RNAp + BBA.R0040 + LuxR2.HSL2:luxO1</td>
<td>30</td>
</tr>
<tr>
<td>ribmRNA.GFP + rib + mRNA.GFP.I + mRNA.GFP</td>
<td>33 aa/s, 220 aa</td>
<td>RNAp + BBA.R0040 + LuxR2.HSL2:luxO1</td>
<td>30</td>
</tr>
<tr>
<td>Regulation</td>
<td></td>
<td>RNAp + BBA.R0040 + TetR2;tetOl</td>
<td>0.0005</td>
</tr>
<tr>
<td>TetR2 + tetOl → TetR2;tetOl</td>
<td>10000000000</td>
<td>RNAp + BBA.R0040 + TetR2;tetOl</td>
<td>0.0005</td>
</tr>
<tr>
<td>TetR2;tetOl → TetR2 + tetOl</td>
<td>0.005</td>
<td>RNAp + BBA.R0040 + TetR2;tetOl</td>
<td>0.0005</td>
</tr>
<tr>
<td>TetR2 + tetOl → TetR2;tetOl</td>
<td>10000000000</td>
<td>RNAp + BBA.R0040 + TetR2;tetOl</td>
<td>0.0005</td>
</tr>
<tr>
<td>TetR2;tetOl → TetR2 + tetOl</td>
<td>0.005</td>
<td>RNAp + BBA.R0040 + TetR2;tetOl</td>
<td>0.0005</td>
</tr>
<tr>
<td>LuxR2 + HSL → LuxR2:HSL</td>
<td>5000000000</td>
<td>RNAp + BBA.R0040 + TetR2;tetOl</td>
<td>0.0005</td>
</tr>
<tr>
<td>LuxR2:HSL → LuxR2 + HSL</td>
<td>0.1</td>
<td>RNAp + BBA.R0040 + TetR2;tetOl</td>
<td>0.0005</td>
</tr>
<tr>
<td>LuxR2:HSL + LuxR2 → LuxR2:HSL2</td>
<td>5000000000</td>
<td>RNAp + BBA.R0040 + TetR2;tetOl</td>
<td>0.0005</td>
</tr>
<tr>
<td>LuxR2:HSL2 → LuxR2 + HSL</td>
<td>0.1</td>
<td>RNAp + BBA.R0040 + TetR2;tetOl</td>
<td>0.0005</td>
</tr>
<tr>
<td>LuxR2:HSL2 + luxO1 → LuxR2:HSL2</td>
<td>10000000000</td>
<td>RNAp + BBA.R0040 + TetR2;tetOl</td>
<td>0.0005</td>
</tr>
<tr>
<td>LuxR2:HSL2 + luxO1 → LuxR2:HSL2 + luxO1</td>
<td>0.4</td>
<td>RNAp + BBA.R0040 + TetR2;tetOl</td>
<td>0.0005</td>
</tr>
<tr>
<td>Induction</td>
<td></td>
<td>RNAp + BBA.R0040 + TetR2;tetOl</td>
<td>0.0005</td>
</tr>
<tr>
<td>TetR2 + αc → TetR2;αc</td>
<td>5000000000</td>
<td>RNAp + BBA.R0040 + TetR2;tetOl</td>
<td>0.0005</td>
</tr>
<tr>
<td>TetR2 + αc + αTc → TetR2;αc + αTc</td>
<td>0.1</td>
<td>RNAp + BBA.R0040 + TetR2;tetOl</td>
<td>0.0005</td>
</tr>
<tr>
<td>TetR2 + αc → TetR2;αc + αTc</td>
<td>0.05</td>
<td>RNAp + BBA.R0040 + TetR2;tetOl</td>
<td>0.0005</td>
</tr>
<tr>
<td>TetR2 + αc + αTc → TetR2;αc + αTc</td>
<td>0.05</td>
<td>RNAp + BBA.R0040 + TetR2;tetOl</td>
<td>0.0005</td>
</tr>
<tr>
<td>TetR2 + αc + αTc + αTc → TetR2;αc + αTc + αTc</td>
<td>0.05</td>
<td>RNAp + BBA.R0040 + TetR2;tetOl</td>
<td>0.0005</td>
</tr>
<tr>
<td>TetR2 + αc + αTc + αTc + αTc → TetR2;αc + αTc + αTc + αTc</td>
<td>0.05</td>
<td>RNAp + BBA.R0040 + TetR2;tetOl</td>
<td>0.0005</td>
</tr>
<tr>
<td>TetR2;αc + αTc + αTc → TetR2;αc + αTc + αTc</td>
<td>0.05</td>
<td>RNAp + BBA.R0040 + TetR2;tetOl</td>
<td>0.0005</td>
</tr>
<tr>
<td>TetR2;αc + αTc + αTc + αTc → TetR2;αc + αTc + αTc + αTc</td>
<td>0.05</td>
<td>RNAp + BBA.R0040 + TetR2;tetOl</td>
<td>0.0005</td>
</tr>
<tr>
<td>TetR2;αc + αTc + αTc + αTc + αTc → TetR2;αc + αTc + αTc + αTc + αTc</td>
<td>0.05</td>
<td>RNAp + BBA.R0040 + TetR2;tetOl</td>
<td>0.0005</td>
</tr>
<tr>
<td>TetR2;αc + αTc + αTc + αTc + αTc + αTc → TetR2;αc + αTc + αTc + αTc + αTc + αTc</td>
<td>0.05</td>
<td>RNAp + BBA.R0040 + TetR2;tetOl</td>
<td>0.0005</td>
</tr>
</tbody>
</table>
expression and regulation. We are exploring the expansion Designer’s capabilities, so that Designer can recognize and model various other biomolecular mechanisms. This may come at a significant cost of interface complexity and system applicability, and it is likely that regardless of how many mechanisms we add, there may still be need for significant manual intervention by the user. It may indeed be best to keep Designer as a bare-bones reaction network generator, and require the user to expertly add to the network of reactions.

Third, the assumption often fails that all the necessary components are presently available or noted as such in all BioBricks. For instance, there are many BioBricks that do not correctly identify the promoter region, the operator sequences, the ribosome binding sites, etc. Designer then returns an error message, and the user is required to either change the BioBrick labels and description in the Registry of Biological Parts, or manually add components and designations in Designer. We are currently cataloging the BioBricks for which Designer generates a network of reactions without errors and we are attempting to identify sources of possible discrepancies. Certainly, the first output of Designer is dependent on the quality and completeness of BioBrick information in the Registry, yet there is always the option for the user to add, delete, or alter BioBrick components with the Designer interface.

CONCLUDING REMARKS
Modeling can assist synthetic biology the same way it assists other engineering disciplines, generating (i) useful insight into how the components of synthetic constructs influence the whole, (ii) new ideas for solving complex biological problems, and (iii) new ways for testing hypotheses. Standardization of modeling tools can facilitate further development of the tools themselves by the community of computational scientists and can render their use more agreeable by the community of wet lab scientists.

SynBioSS Designer facilitates modeling of synthetic biological systems by quickly generating reaction networks that represent any arbitrary construct. Users can input information in Designer from the Registry of Parts, the de facto repository of biological components used in synthetic biology, and with minimal manipulation get a network of reactions. When simulated such a network can reasonably capture the expected behavior of the synthetic construct. These simulation results can in turn prove useful in making design-related decisions.

SUPPLEMENTARY DATA
Supplementary data are available online at http://bib.oxfordjournals.org/.

Key Points
- Modeling tools can play an important role in synthetic biology: the same modeling helps in aircraft or architecture design; simulations can quickly provide a clear picture of how different components influence the behavior of the whole. We present a brief review of available tools and present SynBioSS Designer.
- With SynBioSS Designer synthetic biologists and engineers can quickly construct models of synthetic biological systems.
- Users can enter DNA sequences, including BioBricks, and Designer returns a network of reactions that model all the steps of the molecular biology dogma.
- With these models, users can study the dynamic behavior of the synthetic construct and make the necessary connection between DNA sequence and biological phenotype.

Acknowledgements
The authors are grateful to Cesar Rodriguez, Douglas Densmore, and Drew Endy for providing early versions of the Standard Biological Parts Web Service and the database of BioBricks.

FUNDING
Funding was provided by US National Science Foundation grants CBET-0644792 and EEC-0608910, as well as a 2009 American Recovery and Reinvestment Act grant from the National Institutes of Health (GM086865).

References
5. Kurata H, Matoba N, Shimizu N. CADLIVE for constructing a large-scale biochemical network based on a