BBF RFC ##: SynBioSS Wiki: A Repository of BioBrick Models

Emma Weeding, Jason Houle, Ben Swiniarski, Patrick Smadbeck, Kristen Lindblad, Katherine Volzing, Poonam Srivastava, Vassilios Sotiropoulos, Kostas Biliouris, and Yiannis Kaznessis

Date

1. Purpose

This BioBricks Foundation Request for Comments (BBF RFC) provides instructions on how to search SynBioSS Wiki for models of BioBricks. Each BioBrick can be represented by a network of reactions and SynBioSS Wiki is a repository of BioBrick models. We are planning to parse all BioBricks through SynBioSS Designer and generate reaction network models for each, as described in BBF RFC 40. We will store these models in SynBioSS Wiki for users to access readily.

2. Relation to other BBF RFCs

BBSF RFC ## updates BBF RFC #40.

3. Copyright Notice

4. SynBioSS Wiki

Introduction to SynBioSS Wiki

SynBioSS Wiki is a component of the Synthetic Biology Software Suite (synbioss.sourceforge.net). SynBioSS Wiki is practically two things: i) a web interface based on the MediaWiki package and ii) a database for storing molecular components, their interactions and pertinent biological information. The purpose of SynBioSS Wiki is to enable the scientific community to store and retrieve information related to synthetic biology efforts, and to facilitate the creation of networks of biochemical reactions that can be modeled by the SynBioSS Desktop Simulator (DS). This document is meant to serve as a detailed description of the SynBioSS Wiki project in terms of the front-end

user experience and the underlying database, programming, and extensions to the MediaWiki software.

Database Structure

The SynBioSS Wiki goes beyond the MediaWiki software in storing kinetic information in a formatted (and therefore machine-searchable) format. In general, MediaWiki is designed to store "articles," as exemplified by the format of Wikipedia entries. While such articles may contain a wealth of information, this information is formatted for interpretation by a human reader rather than an automated algorithm. A database of kinetic constants, on the other hand, should be easily searchable by participating species, reaction type, etc.

The underlying SynBioSS Wiki database, called "biowikidb", contains two sections. The first section is the standard MediaWiki tables, necessary for the MediaWiki software to function properly. These tables have not been edited, and thus are fully documented on the MediaWiki website[1]. Note that for this project, each MediaWiki table has a prefix of "bw_", i.e. the standard table "archive" is "bw_archive" in our database.

The other section of the database contains the tables generated specifically for the SynBioSS Wiki project. A graphical representation is provided below, with all primary keys in bold text:

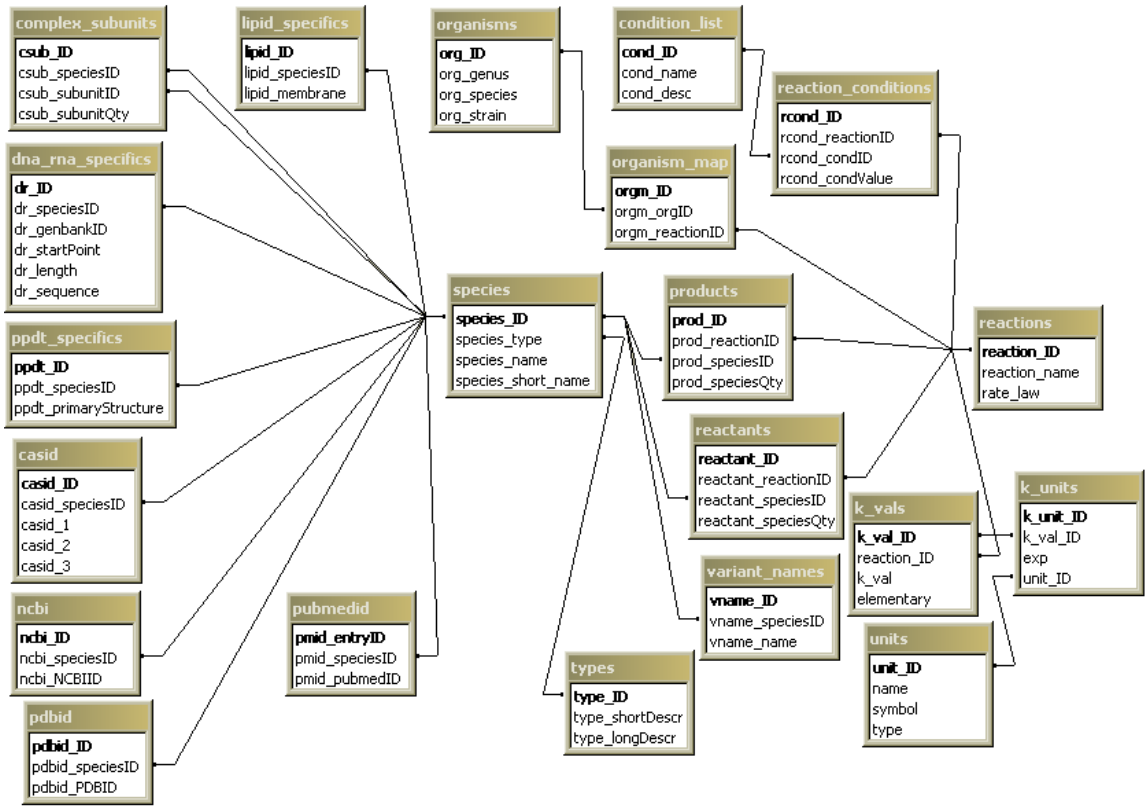

Figure 1 – Layout of the database supporting SynBioSS Wiki (excluding the tables normally associated with the MediaWiki software).

Many tables & fields are self explanatory, although certain fields merit clarification.

Table 1 – Descriptions of selected tables and fields depicted in Figure 1.

Table	Column	Description
casid	casid_1	CAS ID numbers are composed of three series of digits separated by dashes, e.g. 1234-56-7. The "casid_1" in this case is "1234".
casid	casid_2	casid_2 is "56" for previous example
casid	casid_3	casid_3 is "7" for previous example.
complex_subunits	csub_speciesID	The ID of the complex itself.
complex_subunits	csub_subunitID	The ID of a species within the complex.
k_vals	elementary	Equal to either 1 or 0. If "1", the reaction the k value refers to has an elementary rate law.
reactions	rate_law	If the reaction's rate is non-elementary, the complex rate law is stored here as MathML.
units	name	The full name for the unit, e.g. "second"
units	symbol	The symbol for the unit, e.g. "s"

units	type	Categories such as mass, volume, length.

MediaWiki Software and Web Interface

Licensed under the GNU General Public License, MediaWiki is a free software package. It is written in PHP and requires a database behind it; it may use either a MySQL or PostgreSQL database management system. The remainder of this document details how the MediaWiki package was used to develop a web interface to facilitate the creation of networks of biochemical reactions. These models may then be analyzed in any external simulation software of the user's choice.

The wiki's root directory is /var/www/wiki/. All files and directories referenced hereafter are relative to this root directory, i.e. the "includes" directory is /var/www/wiki/includes/.

A short overview of the languages used in this project follows.

- **PHP:** The main programming language used. PHP is a scripting language interpreted and executed on the web server.
- **SQL:** The query language used for communication between the PHP-scripted web pages and the database.
- **HTML:** Used to format how the wiki displays in a web browser. HTML and PHP often co-exist in the same files, though the PHP code is hidden from the end-user.
- **Javascript:** A scripting language embedded in HTML documents and interpreted/executed locally by the user's browser.
    - **AJAX:**Technique used to make the interface more user-friendly. More specifically, on the "Add a Species" and "Add a Reaction" pages, one may search for a species in the database without reloading the entire page.
    - **DHTML:**Utilized for similar reasons as AJAX. On the "Add a Species" page, once the user selects a type, the remainder of the form appears automatically. The page must reload, however.
- **XML**
    - **MathML:** Standardized format for storing mathematical expressions. Used to represent complex rate laws.

- **SBML:** Standardized format for representing networks of biochemical reactions. The desired output format for models created by the user via the wiki.

Custom Pages

A list of the currently used Special Pages is below. The name of the page is in bold, while its corresponding filename in both the /includes/ and /extensions/ directory is in parentheses (php files of identical names but different content exist in both directories simultaneously).

Add a Species *(SpecialSpecies.php)*

This page allows the user to add a new species to the database. Upon entering the name of the species, its PubMedID, its NCBI ID, and selecting its type, he/she may then enter further information contingent upon the type:

- **Complex:**A complex is any species composed of two or more simpler species bound together. Each constituent species must exist within the database separately. The name and quantity of species within the complex must be specified. This is done using the AJAX search function described earlier.
- **DNA/RNA:** Start point, length, sequence, GenBankID
- **Lipid:** Membrane specifics
- **Protein:** Primary structure, PDB ID
- **Small Molecule:** CAS registry number

Upon submitting a species, the user is redirected to the new page for that species. Specific information about these individual pages is outlined in the "Edited Pages" section.

Add/Edit a Reaction*(SpecialReaction.php)*

Depending on the way in which this page is accessed, a user may either add a new reaction or edit an existing one.

In the former case, the user must first use the AJAX search function (outlined earlier) to select a species, enter its stoichiometry, and indicate whether it should be a

reactant or a product. In general, catalysts are both reactants and products, and would be entered on both sides of the equation. Upon entering at least one reactant or product, the user's reaction will appear on the page. If desired they may select one or more species in the reaction and remove it, or clear the entire reaction.

Following the specification of reaction stoichiometry, the form provides fields for kinetic data, contingent upon the complexity of the reaction's rate law.

- **Elementary:** Add the kinetic constant and its units. The user need not enter the specific elementary rate used, as this is self-evident in his/her choice of units and reaction stoichiometry.
- **Arbitrary:** The user may upload a file containing content-MathML describing the equation for the rate law. The MathML is parsed and its parameters are displayed on the Add a Reaction page. The user may then specify information for each parameter. If the parameter is a constant, the user may enter its numerical value and units (if applicable). He/She may also choose from a list of common constants, such as Boltzmann's, if so desired. If the parameter is a variable, the user shall supply a brief description of what it represents (e.g. "Concentration of species A") and the units in which this value should be entered. Parameters whose values have been specified are displayed on the page for future reference.
-

If a user visits an individual species' page and selects a reaction to edit, they will be directed to this page again, where they may edit all aspects of the reaction.

Custom Model *(SpecialModel.php)*

On this page the user may view the reactions he/she has added to his/her model. The reactions and their corresponding kinetic data are displayed. The user may delete one or many reactions from his/her model or clear the whole model if desired. At the bottom of the page is a submit button which converts the user's model seen on the wiki into formatted SBML which the user may then save. This SBML file may be imported into any number of simulation programs and further analyzed.

Add an Organism *(SpecialOrganism.php)*

This page is used to add organisms to the database. A planned feature for the "Add a Reaction" page is to associate reactions with the organism(s) they were observed in. To do this, the user would search for organisms using an AJAX function similar to the one for species. Once again, this feature is not yet implemented, and there is currently no way for a user to view the organisms already in the database.

Edits to Existing MediaWiki Pages

The remaining SynBioSS Wiki features were implemented by editing and adding code to existing MediaWiki files; a list of these files follows. The design philosophy of the SynBioSS Wiki has centered on the creation of new pages; the editing of existing MediaWiki pages has been intentionally minimized to facilitate quick updates of the underlying MediaWiki code. A short description of the page is in bold, while the corresponding file path is in parentheses.

MediaWiki Search *(/includes/SpecialSearch.php)*

A link to the "Add a Species" page and the "Add a Reaction" page was added to the bottom of all search results. The standard MediaWiki search does not function properly with the added database tables, so the search engine has been modified. Specifically, lines 87-240 contain either edited or new code; all further expansions of the search function should be integrated there. The line `require_once("action_pages/sql.inc")`was also added to the beginning of the page, to allow for the use of the MediaWiki database functions.

MonoBook Skin *(/skins/MonoBook.php)*

One small segment was removed from immediately below line 166:

```
<input type='submit' name="go" class="searchButton" id="searchGoButton" value="<?php $this->msg('go') ?>" /> 
```

This removal causes the search bar on the left side of the wiki to display a "Search" button, but no "Go" button. Note that this change only applies to the wiki's default skin, MonoBook.

General Wiki Configuration *(/LocalSettings.php)*

A detailed explanation of wiki configuration can be found on the MediaWiki website[2]. The most notable settings adjusted for this project include the location of the database and the extensions to use. All browser caching has been deactivated, and titles of individual species pages are allowed to begin with a digit or lowercase letter.

Individual Species Pages *(/includes/Article.php)*

On each page, the species' name, PubMedID, NCBI ID, type, and additional specifics contingent on the type are listed. A list of reactions with corresponding kinetic data is displayed as well. The user may edit a reaction or add it to his/her model. Finally, by clicking the "edit" tab at the top of the page, one may add miscellaneous qualitative information about the species. Developers altering Article.php itself should add or edit code between lines 846-999, and add all `require_once()` statements to the beginning of the file.

Future Features

General

While the existing MediaWiki code and the SynBioSS Wiki extensions already compliment each other, we will continue to work to further integrate the two.For example, users should be able to view old versions of kinetic data, who edited them, and when. Clever use of this feature could alleviate concerns of data security, increase the freedom to which users can edit data, and reduce the need for manual moderation of submitted data. Currently, these features are available to users on the text-based portions of the Wiki. Reversions of kinetic data (in the case of incorrect input or intentional vandalism) are also possible, but must be effected by the database administrators.

SynBioSS Wiki / Designer Unification

Tighter integration between the SynBioSS Wiki and Designer is a high priority. As the amount of data seeding the Wiki grows, the Designer will pull kinetic data from the Wiki directly when generating models. While the small amount of data populating the Wiki currently makes such a feature "unhelpful," a populated back-end database (which the Wiki will gradually provide) will make the Designer's automatically-generated models vastly more useful by reducing or eliminating the need for manual input of kinetic parameters.

Add a Species

At present, a user must enter species information manually. Ideally, one should be able to query PubMed, GenBank, etc., from the wiki and insert the relevant information into the form automatically. This may or may not be possible with certain databases. Additionally, interaction between the wiki and BioBricks is to be implemented.

Although the table "variant_names" exists, the web interface can not yet be used to add alternative names for a species.

Add a Reaction

This page must be further adapted to the usage of MathML. While a user may already upload MathML representing a rate law, ideally one should be able to type an

equation which is then converted into MathML. In addition, after the MathML is parsed, the equation itself should display on the page in a convenient graphical format. This graphical display of the equation is very close to being implemented.

Users should also be able to store information on the source of data, the conditions under which this data was obtained, the method used, and the organism(s) in which the reaction was observed. The biowiki tables “organisms”, “organism_map”, “condition_list”, and “reaction_conditions” exist for this purpose, though these have not made their way into usage on the production version of the Wiki as yet.

Modeling tools can play an important role in synthetic biology the same way modeling helps in aircraft or architecture design: simulations can quickly provide a clear picture of how different components influence the behavior of the whole [1,2]. 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.

SynBioSS Designer is a web-based tool (available with an Open License at http://synbioss.sourceforge.net/), with a user-friendly interface which uses universal 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. Currently, Designer is in beta, which means that it works, but we are still making changes based on feedback from users.

SynBioSS Designer now has a connection to the official Registry of Standard Biological Parts, through the Standard Biological Parts Web Service. 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, 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.

5. How to Build a Model of a Biobrick

In this section we present an illustrative example of how to use Designer to generate models of 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. For brevity, all BioBrick IDs will hereafter be referred to without the "BBa_" prefix, e.g. "R0040" instead of "BBa_R0040".

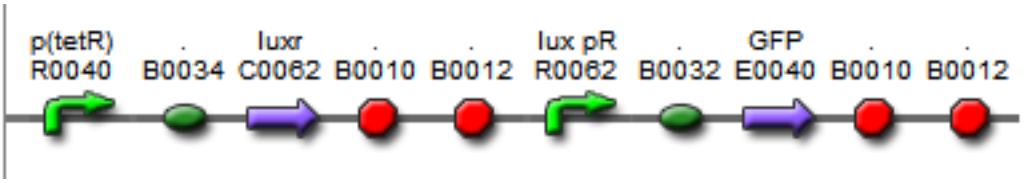

Figure 1. Schematic of BBa_T9002

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 1) the sequence of parts, 2) all proteins in the system, and 3) all effector molecules in the system. The brick sequence is shown in Figure 1. 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. 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. Now the user can click on the tab of each component, access the component's specific properties and change them if necessary.

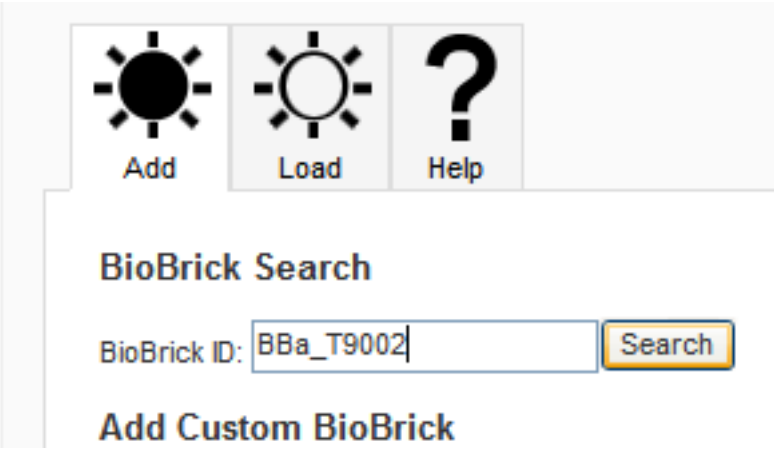

Figure 2. SynBioSS Designer Screenshot. Users can enter the name of any BioBrick.

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 button 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.

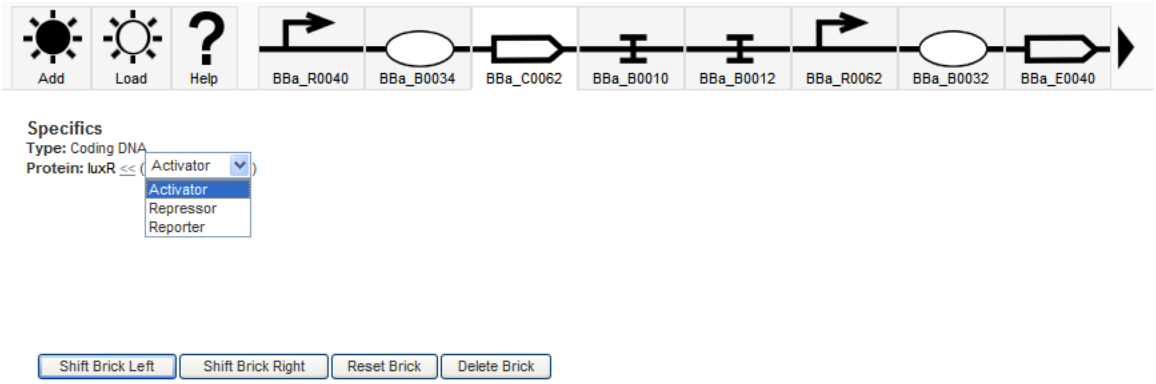

Figure 3. Designer Screenshot

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 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 ribsome 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 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).

Figure 4. Designer Screenshot

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 4 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.

6. Designer Output

The output is a network of reactions representing all the steps in gene expression and regulation, shown in Table I. 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 below:

Species Type	Initial Amount	Split on Cell Division
mRNA	0	Y
Promoters & Operators	1	N
Proteins	0	Y
Ribosome	600	N
RNAp	300	N
Other	0	N

Additionally, all species are designated to be saved to a file upon simulation, and the initial volume of the system is 1e-15L 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.

Table I. Reaction Network for BBa_T9002

Protein Multimerization	Kinetic Data
2 TetR → TetR2	1000000000
TetR2 → 2 TetR	0
2 LuxR → LuxR2	1000000000
LuxR2 → 2 LuxR	0
Transcription	
RNAp + BBA_R0040 + tetO2 + tetO1 → RNAp:BBA_R0040:tetO2:tetO1	0.0166
RNAp:BBA_R0040:tetO2:tetO1 → RNAp + BBA_R0040 + tetO2 + tetO1	0.75
RNAp:BBA_R0040:tetO2:tetO1 → RNAp:BBA_R0040:tetO2:tetO1*	0.3
RNAp:BBA_R0040:tetO2:tetO1* → RNAp:DNA_LuxR + BBA_R0040 + tetO2 + tetO1	30
RNAp:DNA_LuxR → RNAp + mRNA_LuxR	30 nt/s, 600 nt
Translation	
rib + mRNA_LuxR → rib:mRNA_LuxR	100000
rib:mRNA_LuxR → rib:mRNA_LuxR_1 + mRNA_LuxR	33
rib:mRNA_LuxR_1 → rib + LuxR	33
rib + mRNA_GFP → rib:mRNA_GFP	100000
rib:mRNA_GFP → rib:mRNA_GFP_1 + mRNA_GFP	33 aa/s
rib:mRNA_GFP_1 → rib + GFP	33 aa/s, 220 aa
Regulation	
TetR2 + tetO1 → TetR2:tetO1	1000000000
TetR2:tetO1 → TetR2 + tetO1	0.005
TetR2 + tetO2 → TetR2:tetO2	1000000000
TetR2:tetO2 → TetR2 tetO2	0.005
LuxR2 + HSL → LuxR2:HSL	50000000
LuxR2:HSL → LuxR2 + HSL	0.1
LuxR2:HSL + HSL → LuxR2:HSL2	50000000
LuxR2:HSL2 → LuxR2:HSL HSL	0.1
LuxR2:HSL2 + luxO1 → LuxR2:HSL2:luxO1	1000000
LuxR2:HSL2:luxO1 → LuxR2:HSL2 + luxO1	0.4
Induction	
TetR2 + aTc → TetR2:aTc	50000000
TetR2:aTc → TetR2 + aTc	0.1
TetR2:aTc + aTc → TetR2:aTc2	50000000
TetR2:aTc2 → TetR2:aTc + aTc	0.1
TetR2:aTc + tetO1 → TetR2:aTc:tetO1	1000000000
TetR2:aTc:tetO1 → TetR2:aTc + tetO1	0.7
TetR2:tetO1 + aTc → TetR2:aTc:tetO1	1000000
TetR2:aTc:tetO1 → TetR2:tetO1 + aTc	0.4
TetR2:aTc2 + tetO1 → TetR2:aTc2:tetO1	1000000
TetR2:aTc2:tetO1 → TetR2:aTc2 + tetO1	0.4
TetR2:aTc:tetO1 + aTc → TetR2:aTc2:tetO1	50000000
TetR2:aTc2:tetO1 → TetR2:aTc:tetO1 + aTc	0.1
TetR2:aTc + tetO2 → TetR2:aTc:tetO2	1000000000
TetR2:aTc:tetO2 → TetR2:aTc + tetO2	0.7
TetR2:tetO2 + aTc → TetR2:aTc:tetO2	1000000

Non-Specific DNA Interactions	Kinetic Data
TetR2 + nsDNA → TetR2:nsDNA	1000
TetR2:nsDNA → TetR2 + nsDNA	1.6225
TetR2:aTc + nsDNA → TetR2:aTc:nsDNA	1000
TetR2:aTc:nsDNA → TetR2:aTc + nsDNA	1.6225
TetR2:nsDNA + aTc → TetR2:aTc:nsDNA	1000
TetR2:aTc:nsDNA → TetR2:nsDNA + aTc	1.6225
TetR2:aTc2 + nsDNA → TetR2:aTc2:nsDNA	1000
TetR2:aTc2:nsDNA → TetR2:aTc2 + nsDNA	1.6225
TetR2:aTc:nsDNA + aTc → TetR2:aTc2:nsDNA	1000
TetR2:aTc2:nsDNA → TetR2:aTc:nsDNA + aTc	1.6225
LuxR2:HSL2 + nsDNA → LuxR2:HSL2:nsDNA	1000
LuxR2:HSL2:nsDNA → LuxR2:HSL2 + nsDNA	1.6225
Activation	
RNAp + BBA_R0062 + LuxR2:HSL2:luxO1 → RNAp:BBA_R0062:luxO1:LuxR2:HSL2	0.0166
RNAp:BBA_R0062:luxO1:LuxR2:HSL2 → RNAp + BBA_R0062 + LuxR2:HSL2:luxO1	0.75
RNAp:BBA_R0062:luxO1:LuxR2:HSL2 → RNAp:BBA_R0062:luxO1:LuxR2:HSL2*	3
RNAp:BBA_R0062:luxO1:LuxR2:HSL2* → RNAp:DNA_GFP + BBA_R0062 + LuxR2:HSL2:luxO1	30
Leakiness	
RNAp + BBA_R0040 + TetR2:tetO1 + tetO2 → RNAp:BBA_R0040:tetO2:tetO1:TetR2	10000000
RNAp:BBA_R0040:tetO2:tetO1:TetR2 → RNAp + BBA_R0040 + TetR2:tetO1 + tetO2	0.75
RNAp:BBA_R0040:tetO2:tetO1:TetR2 → RNAp:BBA_R0040:tetO2:tetO1:TetR2*	0.3
RNAp:BBA_R0040:tetO2:tetO1:TetR2* → RNAp:DNA_LuxR + BBA_R0040 + TetR2:tetO1 + tetO2	30
Transport	
Ø → TetR2	1.00E-10
Degradation	
GFP → Ø	0.000289
mRNA_GFP → Ø	0.0015
TetR2 → Ø	0.000289
TetR2:nsDNA → nsDNA	0.000193
LuxR2 → Ø	0.000289
mRNA_LuxR → Ø	0.0015
TetR2:aTc → aTc	0.000289
TetR2:aTc:nsDNA → aTc + nsDNA	0.000193
TetR2:aTc2 → 2 aTc	0.000289
TetR2:aTc2:nsDNA → 2 aTc + nsDNA	0.000193
LuxR2:HSL → HSL	0.000289
LuxR2:HSL2 → 2 HSL	0.000289
LuxR2:HSL2:nsDNA → 2HSL + nsDNA	0.000193

TetR2:aTc:tetO2 → TetR2:tetO2 + aTc	0.4
TetR2:aTc2 + tetO2 → TetR2:aTc2:tetO2	1000000
TetR2:aTc2:tetO2 → TetR2:aTc2 + tetO2	0.4
TetR2:aTc:tetO2 + aTc → TetR2:aTc2:tetO2	50000000
TetR2:aTc2:tetO2 → TetR2:aTc:tetO2 + aTc	0.1

7. Concluding Remarks

Modeling can assist synthetic biology the same way it assists other engineering disciplines, generating a) useful insight into how the components of synthetic constructs influence the whole, b) new ideas for solving complex biological problems, and c) 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.

Certainly, the user is REQUIRED to ascertain the relevance and quality of the model generated by Designer. The kinetic constants, for example, are ones drawn from well-known lactose and tetracycline operons, and are only added in the models as a starting point. Furthermore the steps of the molecular biology dogma that Designer employs to generate the model may not be applicable to any system. The user SHOULD carefully check the reactions and the parameters Designer produces.

What we plan on doing is to parse the entire BioBricks database through designer and generate models for all BioBricks. We can then create a compendium of models for BioBricks that all users can have access to.

8. Author's Contact Information

Emma Weeding: weed0019@umn.edu
Jason Houle: jasonmhoule@gmail.com
Ben Swiniarski: swini003@umn.edu
Patrick Smadbeck: patsmad@gmail.com
Kristen Lindblad: kristen-lindblad@bethel.edu
Katherine Volzing: katvolzing@gmail.com
Poonam Srivastava: sriva036@umn.edu
Vassilios Sotiropoulos: sotiropo@cems.umn.edu

Kostas Biliouris: biliouris@cems.umn.edu
Yiannis Kaznessis: yiannis@cems.umn.edu

References

1. Hill, J. Tomshine, E. Wedding, V. Sotiropoulos, Y. Kaznessis, **"SynBioSS: the Synthetic Biology Modeling Suite"**, Bioinformatics, 2008, 24(21):2551-255

2. Ramalingam, KI, Tomshine, JR, Maynard, JA, Kaznessis YN. " **Forward engineering of synthetic bio-logical AND gates**" Biochemical Engineering Journal, 2009, 47(1-3):38-47