

BBF RFC 40: How to Build Kinetic Models of BioBricks

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

October 10, 2009

1. Purpose

This BioBricks Foundation Request for Comments (BBF RFC) provides instructions on how to automatically generate a model of any BioBrick sequence. These models can be used in computer simulations of the dynamic behavior of all molecular components of the BioBrick.

2. Relation to other BBF RFCs

BBSF RFC 40 does not update or replace any earlier BBF RFC.

3. Copyright Notice

Copyright (C) The BioBricks Foundation (2009). All Rights Reserved.

4. SynBioSS Designer

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://synbioass.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”.

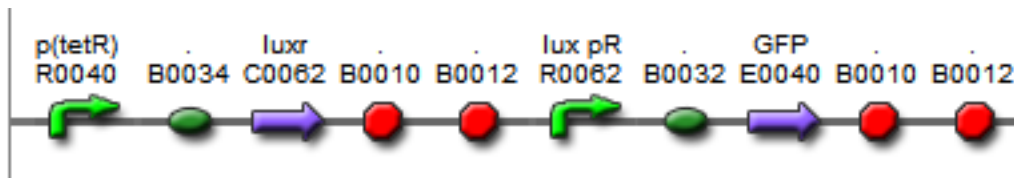


Figure 1. Schematic of BBa_T9002

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.

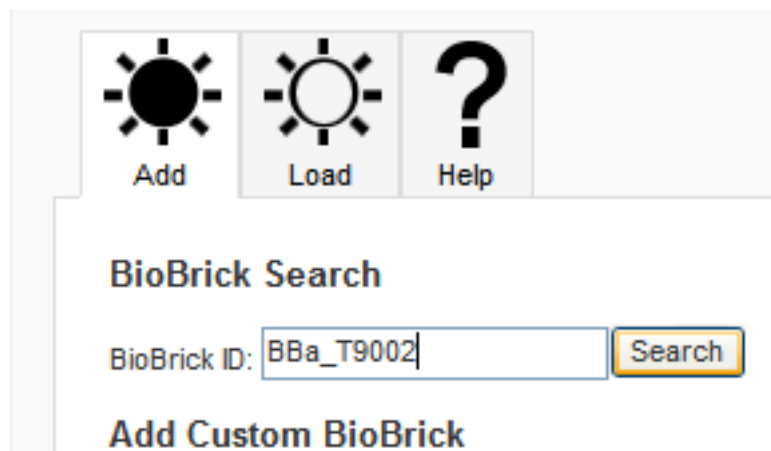


Figure 2. SynBioSS Designer Screenshot. Users can enter the name of any BioBrick.

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 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

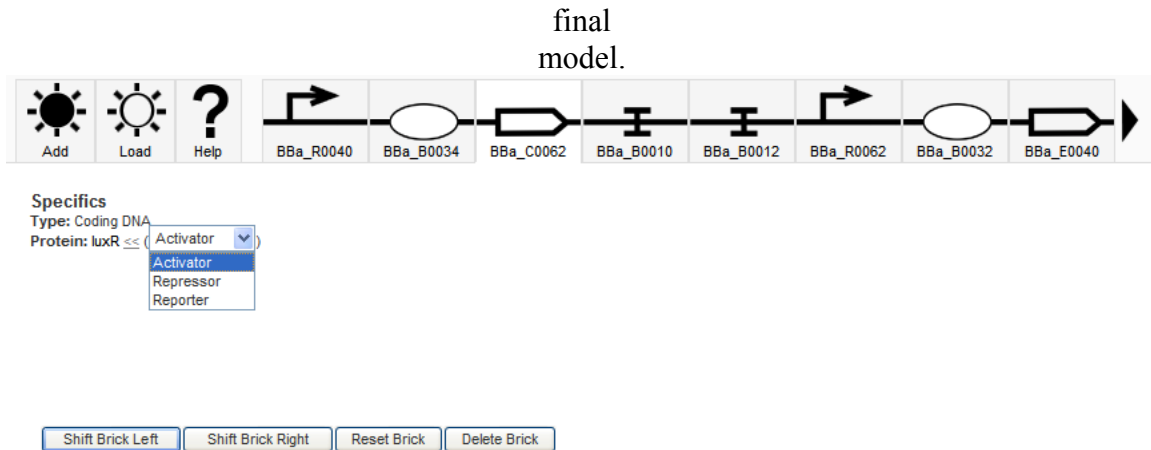


Figure 3. Designer Screenshot

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 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 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).

Binding Specifics

Please specify which operator sites the protein(s) can bind to (if any).

Protein: operator:

tetO1
 tetO2
luxO1

Current Proteins

Protein	Type	Complex	Binds
LuxR	Activator	LuxR2	<i>optional</i>
GFP	Reporter	GFP	<i>optional</i>
TetR	Repressor	TetR2	tetO1 tetO2

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	Non-Specific DNA Interactions	Kinetic Data
2 TetR → TetR2	1000000000	TetR2 + nsDNA → TetR2:nsDNA	1000
TetR2 → 2 TetR	0	TetR2:nsDNA → TetR2 + nsDNA	1.6225
2 LuxR → LuxR2	1000000000	TetR2:aTc + nsDNA → TetR2:aTc:nsDNA	1000

LuxR2 → 2 LuxR	0	TetR2:aTc:nsDNA → TetR2:aTc + nsDNA	1.6225
Transcription		TetR2:nsDNA + aTc → TetR2:aTc:nsDNA	1000
RNAP + BBA_R0040 + tetO2 + tetO1 →		TetR2:aTc:nsDNA → TetR2:nsDNA + aTc	1.6225
RNAP:BBA_R0040:tetO2:tetO1	0.0166	TetR2:aTc2 + nsDNA → TetR2:aTc2:nsDNA	1000
RNAP:BBA_R0040:tetO2:tetO1 → RNAP + BBA_R0040 + tetO2 + tetO1	0.75	TetR2:aTc2:nsDNA → TetR2:aTc2 + nsDNA	1.6225
RNAP:BBA_R0040:tetO2:tetO1		TetR2:aTc:nsDNA + aTc → TetR2:aTc2:nsDNA	1000
RNAP:BBA_R0040:tetO2:tetO1*	0.3	TetR2:aTc2:nsDNA → TetR2:aTc:nsDNA + aTc	1.6225
RNAP:BBA_R0040:tetO2:tetO1* → RNAP:DNA_LuxR + BBA_R0040 + tetO2 + tetO1	30	LuxR2:HSL2 + nsDNA → LuxR2:HSL2:nsDNA	1000
RNAP:DNA_LuxR → RNAP + mRNA_LuxR	30 nt/s, 600 nt	LuxR2:HSL2:nsDNA → LuxR2:HSL2 + nsDNA	1.6225
Translation		Activation	
rib + mRNA_LuxR → rib:mRNA_LuxR	100000	RNAP + BBA_R0062 + LuxR2:HSL2:luxO1 →	
rib:mRNA_LuxR → rib:mRNA_LuxR_1 + mRNA_LuxR	33	RNAP:BBA_R0062:luxO1:LuxR2:HSL2	0.0166
rib:mRNA_LuxR_1 → rib + LuxR	33	RNAP:BBA_R0062:luxO1:LuxR2:HSL2 → RNAP + BBA_R0062 + LuxR2:HSL2:luxO1	0.75
rib + mRNA_GFP → rib:mRNA_GFP	100000	RNAP:BBA_R0062:luxO1:LuxR2:HSL2	
rib:mRNA_GFP → rib:mRNA_GFP_1 + mRNA_GFP	33 aa/s	RNAP:BBA_R0062:luxO1:LuxR2:HSL2*	3
rib:mRNA_GFP_1 → rib + GFP	33 aa/s, 220 aa	RNAP:BBA_R0062:luxO1:LuxR2:HSL2* → RNAP:DNA_GFP + BBA_R0062 + LuxR2:HSL2:luxO1	30
Regulation		Leakiness	
TetR2 + tetO1 → TetR2:tetO1	1000000000	RNAP + BBA_R0040 + TetR2:tetO1 + tetO2 →	
TetR2:tetO1 → TetR2 + tetO1	0.005	RNAP:BBA_R0040:tetO2:tetO1:TetR2	10000000
TetR2 + tetO2 → TetR2:tetO2	1000000000	RNAP:BBA_R0040:tetO2:tetO1:TetR2 → RNAP + BBA_R0040 + TetR2:tetO1 + tetO2	0.75
TetR2:tetO2 → TetR2 + tetO2	0.005	RNAP:BBA_R0040:tetO2:tetO1:TetR2	
LuxR2 + HSL → LuxR2:HSL	50000000	RNAP:BBA_R0040:tetO2:tetO1:TetR2*	0.3
LuxR2:HSL → LuxR2 + HSL	0.1	RNAP:BBA_R0040:tetO2:tetO1:TetR2* → RNAP:DNA_LuxR + BBA_R0040 + TetR2:tetO1 + tetO2	30
LuxR2:HSL + HSL → LuxR2:HSL2	50000000	Transport	
LuxR2:HSL2 → LuxR2:HSL HSL	0.1	∅ → TetR2	1.00E-10
LuxR2:HSL2 + luxO1 → LuxR2:HSL2:luxO1	1000000	Degradation	
LuxR2:HSL2:luxO1 → LuxR2:HSL2 + luxO1	0.4	GFP → ∅	0.000289
Induction		mRNA_GFP → ∅	0.0015
TetR2 + aTc → TetR2:aTc	50000000	TetR2 → ∅	0.000289
TetR2:aTc → TetR2 + aTc	0.1	TetR2:nsDNA → nsDNA	0.000193
TetR2:aTc + aTc → TetR2:aTc2	50000000	LuxR2 → ∅	0.000289
TetR2:aTc2 → TetR2:aTc + aTc	0.1	mRNA_LuxR → ∅	0.0015
TetR2:aTc + tetO1 → TetR2:aTc:tetO1	1000000000	TetR2:aTc → aTc	0.000289
TetR2:aTc:tetO1 → TetR2:aTc + tetO1	0.7	TetR2:aTc:nsDNA → aTc + nsDNA	0.000193
TetR2:tetO1 + aTc → TetR2:aTc:tetO1	1000000	TetR2:aTc2 → 2 aTc	0.000289
TetR2:aTc:tetO1 → TetR2:tetO1 + aTc	0.4	TetR2:aTc2:nsDNA → 2 aTc + nsDNA	0.000193
TetR2:aTc2 + tetO1 → TetR2:aTc2:tetO1	1000000	LuxR2:HSL → HSL	0.000289
TetR2:aTc2:tetO1 → TetR2:aTc2 + tetO1	0.4	LuxR2:HSL2 → 2 HSL	0.000289
TetR2:aTc:tetO1 + aTc → TetR2:aTc2:tetO1	50000000	LuxR2:HSL2:nsDNA → 2HSL + nsDNA	0.000193
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		
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):2551-2555
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