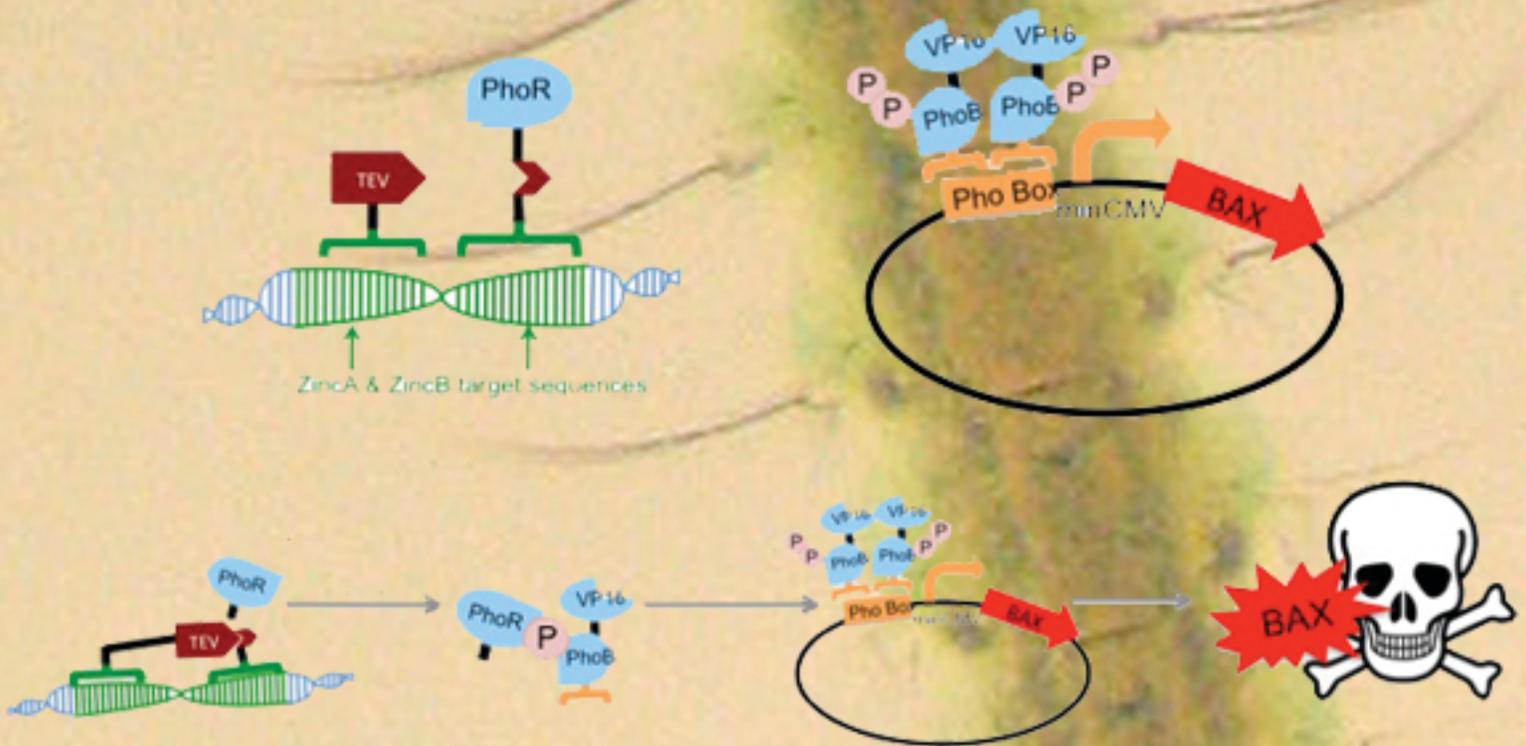


Principles of Synthetic Biology



Inside:

in vivo retroviral therapy (28)

Programmed mosquito pathogens (12)

New approaches to genetic memory (1,43)

Principles of Synthetic Biology

Volume 1, Issue 1, December 10th 2010

On the Cover:

Mosquito Larvae, from public domain image.

http://commons.wikimedia.org/wiki/File:Anopheles_larve.jpg

Synthetic circuit targeting HIV, Brophy and Rubens (p28)

This magazine was produced by the students of Bioengineering 190D/290D at the University of California Berkeley, Berkeley CA and Bioengineering 6.971/20.949 at the Massachusetts Institute of Technology, Cambridge, MA.

Layout design in LaTeX by Alistair Boettiger (UC Berkeley), adapted from pnastwo.cls by Amy Hendrickson, TeXnology Inc. (2004) amyh@texnology.com.

All material is distributed under Creative Commons License 3.0, BY-NC-SA. <http://creativecommons.org/licenses/by-nc-sa/3.0/>.

You may copy, distribute, adapt and transmit this work provided you attribute the authors of the material, you use the work for non-commercial purposes, and if you alter, transform or build on this work you redistribute it only under the same or similar license to this one.

Table of Contents

Genetic D-Latch for Clocked Biological Memory	1
<i>Patel N and Vora S</i>	
Engineering E. coli to undergo asymmetric division ...	7
<i>Freimar J and Weisenstein L</i>	
Novel E. coli mechanisms for mosquito control and programmed cell death ...	12
<i>Tulanont D and Mehdizadeh A</i>	
Modeling cell-to-cell communication: Lux quorum sensing and notch/delta ...	20
<i>Nofal MI and Saarni AR</i>	
In vivo genotyping to target therapeutics ...	28
<i>Brophy J and Rubens J</i>	
A biochemical comparator ...	34
<i>de Rond T</i>	
A model for self-activating signaling cascades ...	39
<i>Chang AY and Williams B</i>	
A cell division counter using DNA invertase chains as memory ...	43
<i>Feng J</i>	
Engineered bacteria to degrade stent-associated arterial plaques ...	47
<i>Spelke, D and Boyarskiy S</i>	
Modeling and investigating the role of positive feedback and ...	54
<i>Zervantonakis IK</i>	
A Programmable Decoder ...	59
<i>Shaw J</i>	
Analysis of noise propagation in biological cascades and feedback networks ...	66
<i>Petrangelo S and Goel R</i>	
Metabolic cost of synthetic circuits ...	73
<i>Chung H and Rubin R</i>	
A framework for programmable processing control in synthetic biological circuits ...	77
<i>McGough L and Thiagarajan A</i>	
Synthetic cell growth rate speedometer ...	97
<i>Malulur P.</i>	

Genetic D-Latch for Clocked Biological Memory

Nikit Patel* and Suhani Vora*,

*Department of Bioengineering, University of California, Berkeley

Submitted to Principles of Synthetic Biology

Synthetic biology is a growing field that aims to engineer biological systems for novel applications. To date, synthetic biologists have constructed many genetic circuits inspired by analogous electrical counterparts such as the genetic toggle switch and the repressilator. The design and creation of more complex genetic circuitry will allow biologists to gain a greater range of functionality. While various biologists have successfully implemented genetic combinatorial circuits, sequential logic circuits have yet to become prominent. A fundamental logic circuit in the field of electrical engineering, which possesses the ability to store memory in a timed fashion, is the clocked D-Latch. The D-Latch consists of a clock, a combinatorial clocking interface, and a memory storage unit. When multiple D-Latch elements are combined, sequential memory can be formed. However, a D-Latch circuit has yet to be constructed biologically. A unit for sequential memory is designed by modelling a simple genetic D-Latch circuit using parts previously characterized by synthetic biologists in tandem with a novel repressor cascade clocking interface. The repressor cascade interface allows for the implementation of a clock, eliminates the possibility of race conditions, and sharpens the transfer function between clock and memory unit.

Logic Circuit Design | Stochastic | Cascades | Repressor

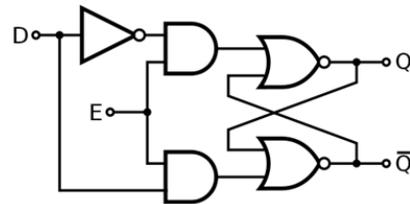
Introduction

While many biological circuits have been constructed to implement combinatorial logic, sequential logic remains a desirable circuit element with the potential to immensely advance the functionality of biologically engineered circuits. In order to generate a modular unit for sequential logic, a clocking mechanism can be interfaced with a memory unit. In the analogous case of electrical circuitry, D-Latches (also known as clocked SR Flip-Flops) are used as building blocks for sequential logic. The D-Latch serves as the inspiration for our circuit design.

The generic D-Latch is built from three primary components. First, an internal clock which ensures signals are read in a synchronized manner, eliminates the possibility of race conditions resulting from stochastic speeds of signals. Second, an interface of combinatorial gates reads the clock and input signal of interest, and makes a decision as to how the memory unit will be set. Often a pair of AND gates is employed for this interface. Finally, the memory unit stores information according to the inputs generated by the combinatorial gates. A circuit diagram with a corresponding logic table is shown in Figure 1.

Previous D-Latch circuits have been developed, incorporating transcriptional regulation and positive feedback or heterodimers to generate clocked memory storage [1,2]. We have designed a circuit analogous to the D-Latch by incorporating a cascade of repressors. The repressor cascades perform analogously to combinatorial AND gates, serving as an interface between the previously characterized repressilator clock constructed by Elowitz et al. and the genetic toggle switch memory unit constructed by Gardner et al. [3,4]. The genetic implementation of the circuit, Figure 2, is a useful addition to previous circuits as the clocking interface provides modularity. The clock signal can be exchanged for any other biological signal as long as it can be tied to a transcriptional repressor. This provides an advantage over the previously modelled positive feedback model which is specific to specialized constructs

utilizing highly tailored promoter units. Beyond the modularity of the clocking interface, our circuit utilizes repressor units within the interface which are more common than heterodimers that were employed by Fritz et al.



Clock (E)	Signal (D)	Out
0	0,1	Previous Out
1	0	0
0	1	1

Fig. 1: Circuit diagram (above) and logic truth table (below) of an electronic Clocked D-latch.

Additionally, the use of repressor cascades as an interface between the clocking and memory storage modules allows for sharpening of signal between the clock and toggle [5]. Finally, the integration of the clock, interface, and memory storage toggle provides the utility of a low pass filter, filtering high frequency inputs which are greater in frequency than the oscillating clock. Thus, the clocking frequency sets the threshold for the maximum frequency for a passing signal. Deterministic and stochastic analysis of this circuit confirm its functionality as a signal sharpening cascade and low pass filter.

Methods

Model Formulation. The genetic implementation of the clocked D-latch is shown in Figure 2. All repressors are assumed to bind cooperatively to their respective promoters. During the modelling process, parameters were chosen so as to ensure gate matching. A standard promoter part was used as a reference, with $ktr=10$ (no./s) and $km=50$ (no.) for transcription and translation under the control of a repressor. At each successive point in the cascade of repressors, the ktr of the promoter was increased to maintain gate-matching. A set of reactions for a single representative repression scheme (A cooperatively represses Pa1 which produces E) follows this form:

Creative Commons License:

CC-BY-NC-SA 3.0

<http://creativecommons.org/licenses/by-nc-sa/3.0/>

GENETIC IMPLEMENTATION

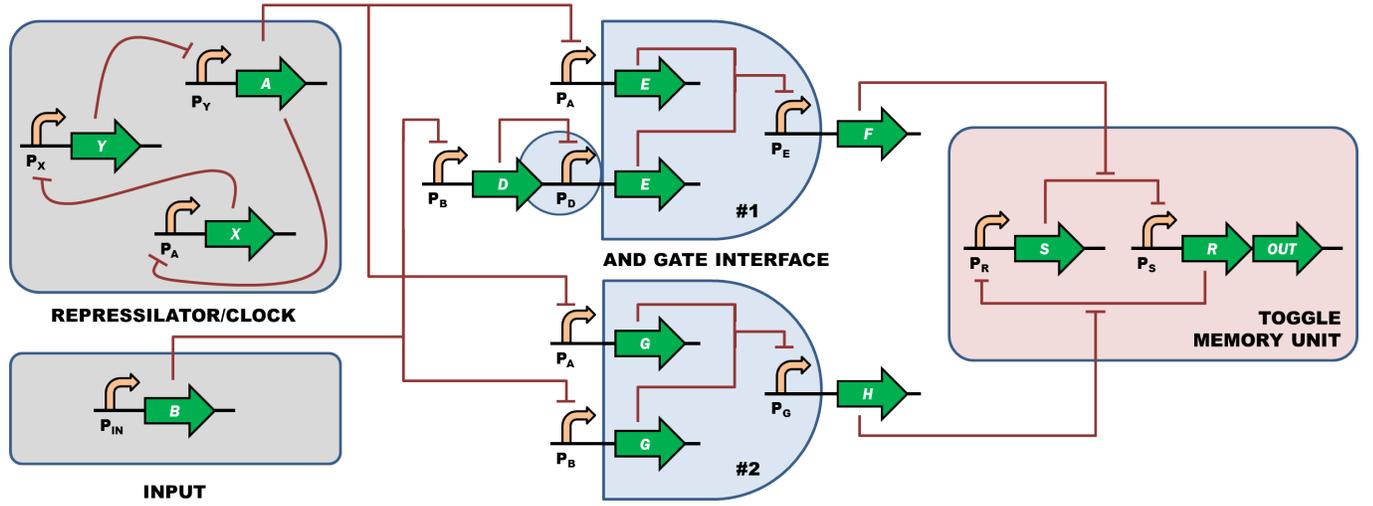
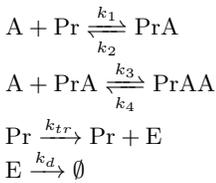


Fig. 2: Genetic Implementation of the Clocked D-Latch. The design involves a cascade of orthogonal repressors with similar biological properties (i.e. degradation rates). The repressilator provides a clock, the AND gate interface prevents a race condition from occurring, and the toggle switch stores memory.



An equivalent set of equations describes each repressor interaction, and together the repressors form repression cascades. Each repressor module deterministically reduces to a second order hill equation.

$$\frac{dE}{dt} = \frac{k_{tr} \frac{k_2 k_4}{k_1 k_3}}{\frac{k_2 k_4}{k_1 k_3} + \frac{k_4}{k_3} A + A^2} - k_d E \approx \frac{k_{tr} K_m^2}{K_m^2 + A^2} - k_d E$$

Reactions for the repressilator and the genetic toggle follow the standard form provided in the respective papers [3,4]. The repressilator consists of a cycle of three mutually repressive transcription factors leading to oscillations in expression. The toggle memory unit consists of two cooperative mutual repressors that can be inactivated by two inputs respectively. These interactions lead to a positive feedback loop that can be set and re-set.

Deterministic Analytical Methods. In order to gain an intuitive understanding of the system, the three modular units of the genetic D-Latch (clock, interface, and toggle) were independently tested deterministically. Ordinary differential equations modelling these units were derived under the assumptions of large numbers of molecules, a homogeneous well-mixed solution, and mass action kinetics. Once the repressilator, AND gates, and toggle were confirmed to work individually, the modules were adjusted for gate matching and subsequently combined for a complete deterministic simula-

tion. The deterministic equations for the system are broken down into the four modular units.

Repressilator Clock governing equations:

$$\frac{dA}{dt} = \frac{k_{tr3} K_{m0}^4}{K_{m0}^4 + Y^4} - k_{d0} A \quad [1]$$

$$\frac{dY}{dt} = \frac{k_{tr3} K_{m0}^4}{K_{m0}^4 + X^4} - k_{d0} Y \quad [2]$$

$$\frac{dX}{dt} = \frac{k_{tr3} K_{m0}^4}{K_{m0}^4 + A^4} - k_{d0} X \quad [3]$$

AND Gate 1 governing equations:

$$\frac{dD}{dt} = \frac{k_{tr1} K_{m1}^2}{K_{m1}^2 + B^2} - k_d D \quad [4]$$

$$\frac{dE}{dt} = \frac{k_{tr2} K_{m1}^2}{K_{m1}^2 + A^2} + \frac{k_{tr} K_{m1}^2}{K_{m1}^2 + D^2} - k_d E \quad [5]$$

$$\frac{dF}{dt} = \frac{k_{tr3} K_{m1}^2}{K_{m1}^2 + E^2} - k_d F \quad [6]$$

AND Gate 2 governing equations:

$$\frac{dG}{dt} = \frac{k_{tr2} K_{m1}^2}{K_{m1}^2 + A^2} + \frac{k_{tr} K_{m1}^2}{K_{m1}^2 + B^2} - k_d G \quad [7]$$

$$\frac{dH}{dt} = \frac{k_{tr3} K_{m1}^2}{K_{m1}^2 + G^2} - k_d H \quad [8]$$

Toggle Memory Unit governing equations:

$$\frac{dS}{dt} = \frac{k_{tr2} K_{m2}^2}{K_{m2}^2 + R^2} - k_d S - k_p S F \quad [9]$$

$$\frac{dR}{dt} = \frac{k_{tr2} K_{m2}^2}{K_{m2}^2 + S^2} - k_d R - k_p R H \quad [10]$$

For the equations above $k_{tr1}=20$, $k_{tr2}=10$, $k_{tr3}=5$, $k_{d0}=0.01$, $k_d=0.1$, $k_p=0.001$, $k_{m0}=1$, $k_{m1}=50$, and $k_{m2}=25$. These parameters were chosen with the assumption that, in the future, repressor parts can be made with similar biological

properties but can also work orthogonal to one another. In addition to the differential equations derived for the toggle, a derivation of parameters representative of a bistable toggle gate matched to the AND gates was performed in MATLAB. MATLAB analysis allowed for identification of values of $k_{tr3}=10$, $k_{m2}=25$, and $kd=0.1$ which provided a steady state high concentration of 100 and a switching concentration at 30. A plot of the final nullclines describing the toggle of our D-Latch is shown in Figure 3.

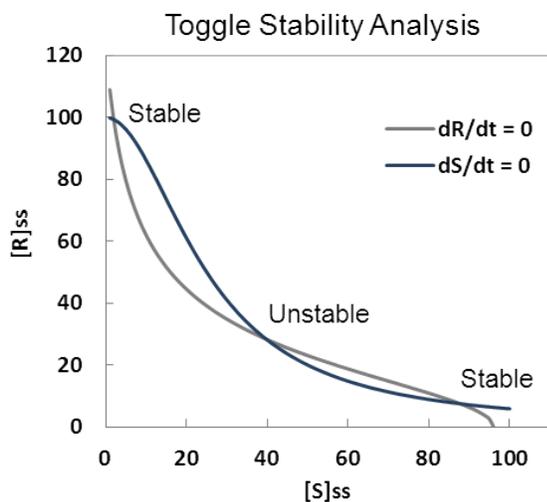


Fig. 3: Nullcline analysis of the toggle switch shows three real solutions indicating bistability. There are two stable and one unstable points.

Stochastic Analytical Methods. While both the repressilator and toggle have been modelled and tested experimentally in previous works, our repressor cascade clocking interface is a novel design for its purpose. Therefore, stochastic testing of this particular unit was necessary, in order to confirm functionality and signal sharpening behaviour. In COPASI, the set of reactions listed under model formulation were implemented as stochastic reactions in order to test the AND gate interface construct. The entire repressor cascade, including both gates, was tested within a single model.

Simulations. Using COPASI we initially simulated deterministic models for the three individual modules of the D-Latch circuit; these units were subsequently combined and tested for functionality and sensitivity. A stochastic model of the clocking interface, covering all potential initial states, was also tested for functionality and sensitivity. Deterministic models utilized the abstract rate laws derived above while the stochastic model was simulated by direct method using the mass action laws.

Results

Repressilator amplitude and frequency. A deterministic simulation of the repressilator clock shows oscillations which can be modified in terms of period and amplitude. In our genetic D-Latch, the species A serves as the clocking species. The period of a pulse of the clocking signal species A can be adjusted by altering the k_m of the species repressing the clock signal (species Y). This decreases the effective repression of

A in the high state. (Figure 4a) Amplitude of pulse can be attenuated by altering the k_{tr} for production of the clocking species. (Figure 4b)

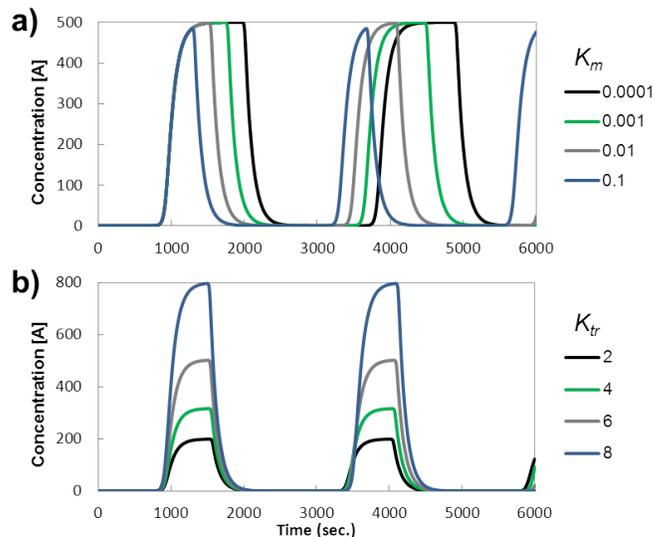


Fig. 4: Figure 4a shows how changing the k_m of the repressilator hill equations can change the periodicity of the oscillations. Figure 4b shows how changing the k_{tr} of each promoter affects the amplitude of the oscillations.

AND gates eliminate race condition. Within the D-Latch construct, signals F and H serve as inputs to the toggle memory unit. By virtue of employing AND logic between the clock and both the inverted and straight signal, it is predicted that outputs F and H of the system will never reach the high state at the same time. If the signals F and H never simultaneously reach the high state, a race condition cannot develop within the toggle. Therefore, the cell containing the genetic device will only ever be in the set or reset state. This provides an advantage over the genetic toggle by Gardner et al. which is still capable of entering a race condition if both inputs are high in a system. In order to test whether our D-Latch circuit removes the possibility of a race condition, a variety of initial concentrations were tested on the combined system to ensure the signals F and H never reach the high state at the same time. From both deterministic and stochastic figures it is apparent that this is the case; F and H cannot approach the high state in tandem, and therefore the toggle is always set to a defined, predictable state.

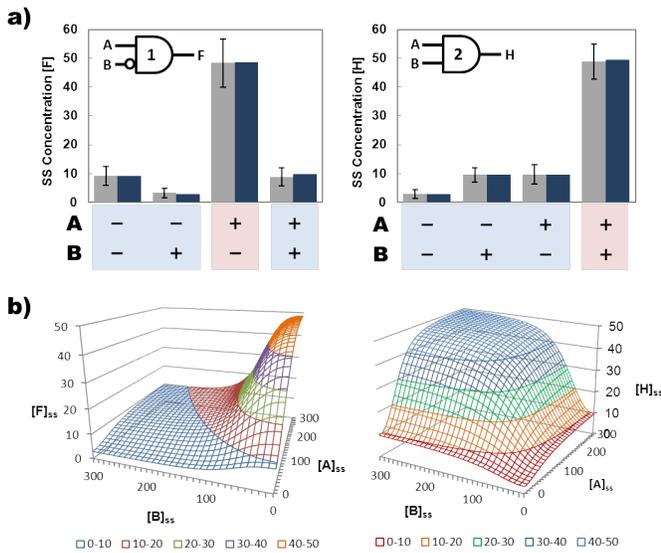


Fig. 5: Figure 5a shows the output from each AND gate deterministically (dark blue) and stochastically (grey) with steady state inputs of A and B. Stochastic simulation was taken over 25 runs - error bar shown. The input values for A/B were 300 and 0 for "+" and "-" respectively. Figure 5b shows surface plots (and transfer function) of the full range of steady state inputs A and B. This gives us intuition about what happens when A and B are switching between high and low states.

The AND gate interface was solved deterministically and stochastically for varying steady-state inputs of A and B. Steady-state concentrations of F and H are mapped out on surface plots to show how each scales in relation to A and B (Figure 5b). Histograms showing steady-state concentrations of the output proteases for both AND gates reflect how closely the stochastic simulations track with the deterministic results (Figure 5a). Looking at the fano factor from the results of the stochastic simulation, we found that the coefficient was close to 1. This signifies that noise is exhibiting outputs similar to a Poisson distribution. There is no significant clustering or dispersal of data. However, we cannot make too many conclusions regarding these observations because of the small number of stochastic runs (25 repetitions) that were conducted.

For AND gate 1, we observed protease F to be high only when A is high and B is low. For AND gate 2, we observed that protease H was only held at high concentrations when A and B were both high. Thus, our AND gates are functional

and have the feature of preventing race conditions because both gates have different responses to B when A is high.

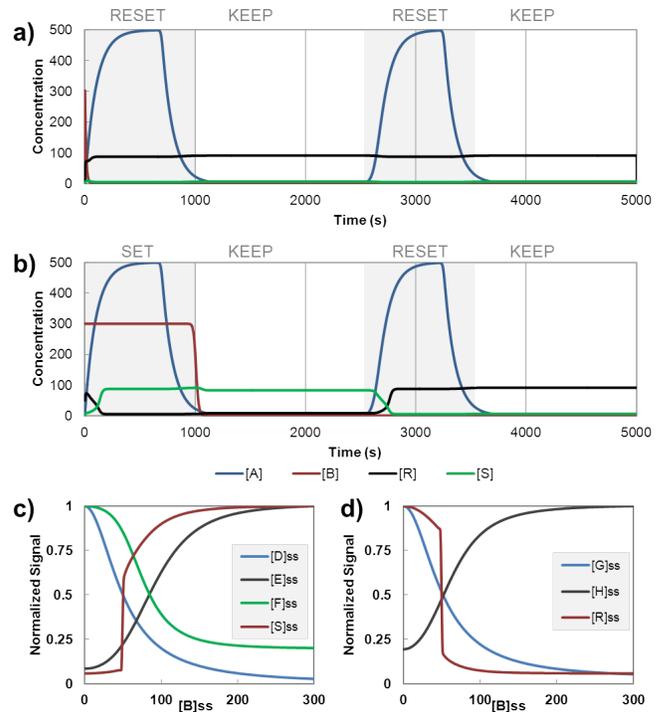


Fig. 6: Figure 6a shows our negative control in which the input is turned off before the clock is turned high. The clock maintains the "reset" state (when R is high.) Figure 6b shows how the latch is "set" high (S state) after a burst of input B. The state is maintained until the second period of the clock when no input is given. This resets the output and maintains this state as well. Figures 6c and 6d depict the normalized protein concentration for AND gates 1 and 2 over a range of steady state inputs of B. The curves show the signal sharpens as you go further into the cascade.

Combined system sets and maintains state as expected. The high state for the genetic D-Latch is represented by high concentration of transcriptional regulator R of the diagram, with low concentration of transcriptional regulator S. If the signal, B, is pulsed before the clock enters the high state, the toggle remains in the low state, as observed over time in Figure 6a. If the signal B is pulsed while the clock is in the high state, the toggle is set, and enters the high state as observed over time in Figure 6b.

Cascade of inverters sharpens signal. As shown in previous work, the use of cascades can lead to sharpening of the transfer function of an output signal assuming that inputs to each gate reach steady state [5]. Effective sharpening requires gate matching, and potentially multiple inverters. By gate matching the repressor cascades of 3 and 4 inverters, and observing the simultaneous transfer functions, we observe sharpened downstream signals as seen in Figures 6c and 6d. Sharpened signals are important in terms of making the system robust to fluctuations of input signal at low and high concentrations. Additionally, the sharpening effect of the D-Latch interface AND gates allows the interface to be modular. If another

memory unit which is not as robust as the genetic toggle switch is desired for a circuit, the D-Latch interface will allow for improved signalling to avoid transition concentrations.

Clocked D-Latch acts as low pass filter.

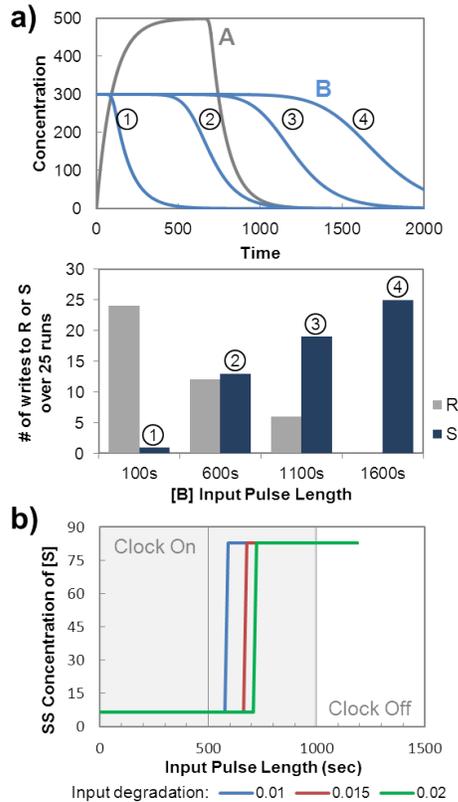


Fig. 7: Figure 7A shows the different input pulses of B added to our system. Stochastic simulations of 25 runs show how many cells were turned on (S state). B degrades at a rate of 0.01 no./s. The longer the input pulse is maintained during the clocking window, then the more likely the toggle can be set. Figure 7b shows that by changing the degradation rate of the input B, one can tune the signal time needed to set the clock to the S state (i.e. tuning the low pass filter.)

Due to the nature of the clocking interface and the toggle memory storage unit, input signals which fluctuate within the time span when the clock is high may not get read. This can be understood by looking at the way the circuit is structured as a whole. To begin with, whenever the clock is high, the circuit as a whole is in a “read” state, and the output will follow the input state. However, the instant the clock drops to the low state, the circuit as a whole enters a “save” state, where the last read output value becomes the permanent output value for the system until the clock goes high again. Therefore, if a signal pulses high and then low within a time span shorter than the pulse of the clock, the system will only temporarily register that the pulse has occurred but will not save the high state as the pulsed signal disappears before the clock goes low. Hence, as seen in Figure 7, only signals of a lower frequency than the frequency of the clock have a chance of being read accurately. Essentially, the ge-

netic D-Latch circuit acts as a low pass filter, with a threshold frequency set by the frequency of the clock.

Discussion

In terms of biological implementation, exact gate matching would be near impossible. Before implementing the D-Latch, a biologist must sync translation rates to be appropriately relative to one another (as described in the paper.) There is a wide array of tools synthetic biologists can use for gate matching. Translation rates can be tuned by creating RBS libraries as well as by tweaking amino acid codon usage. Changes can be made at the transcriptional level by manipulating the -10 and -35 box of a promoter site to affect how often a sigma factor would recruit an RNAP for transcription. The more closely the -10 and -35 box resembles the consensus sequence, then more RNA can be made per second, thus affecting translation as well. The half-lives of proteins can also be manipulated by fusing a short degradation tag to either the C or N termini. The Registry of Standard Biological Parts at MIT has been gathering such tags and, though they do not have exact numbers, they have labelled different tags as slow, medium, or fast degraders.

A problem one might encounter with genetic implementation of this circuit is finding necessary repressors that will bind cooperatively to their promoters. One goal of synthetic biology, especially protein engineering, is to devise methods to manipulate existing parts and create libraries of these proteins that can work orthogonal to one another but still maintain similar overall properties (like degradation rates). For example, since the clocked D-Latch needs 7 repressors that dimerize cooperatively, protein engineers would hopefully be able to create such orthogonal parts from an existing, well-established repressor like the lambda repressor.

Additionally, as expressed by Hooshangi et al., cascades can increase noise within the transition region of a signal. Therefore, a potential issue would arise in the case that the input signal has a high probability of existing in an intermediate concentration [5].

Finally, though the repressilator creates a nice and robust clock in the computational models, the repressilator will not remain in sync amidst a population of bacteria. Therefore, one might try utilize a quorum-sensing based clock by implementing AHL-LuxR systems. Recent endeavors to make these synchronized clocks have proved to be successful [6].

Despite these drawbacks, the clocked D-latch provides a more complex way of storing memory. One application would be for insulin delivery. A clock would turn off (turning down insulin production) when a person goes to sleep to make sure excess insulin isn't made on an empty stomach. When the person wakes up, the clock can turn on and now other inputs or circuit element can control insulin production. More examples like these provide interesting and useful applications for the clocked D-latch.

References

1. Fritz et al. “Designing sequential transcription logic: a simple genetic circuit for conditional memory.” *Systems and Synthetic Biology* (2007)
2. Lauria et al. “Building blocks of a biochemical CPU based on DNA transcription logic.” *Extended Abstract* (Not dated)
3. Elowitz, M B, and S Leibler. “A synthetic oscillatory network of transcriptional regulators.” *Nature* 403, no. 6767 (2000): 335-8.

4. Gardner, T S, C R Cantor, and J J Collins. "Construction of a genetic toggle switch in *Escherichia coli*." *Nature*. 403, no. 6767 (2000): 339-42.
5. Hooshangi et al. "Ultrasensitivity and noise propagation in a synthetic transcriptional cascade." *PNAS*. (2005)
6. Danino, Tal et al. "A synchronized quorum of genetic clocks." *Nature* 463(2010): 326-30

ACKNOWLEDGMENTS. We would like to thank Alistair Boettiger for late night brownies and unparalleled support, Professor Adam Arkin and Ron Weiss for introducing us to the world of synthetic biology, Rachel Bernstein for invaluable notes, and Noah Davidson. Also, we would like to thank the Berkeley and MIT pilot PoSB class of 2010 for stimulating and helpful discussions!

Engineering *E. coli* to undergo asymmetric division

Jake Freimer* and Lauren Weisenstein*,

*Department of Bioengineering, University of California, Berkeley

Submitted to Principles of Synthetic Biology

Survival for bacterial cell populations in fluctuating environments is often a coordinated and dynamic process. Many studies have focused on the observation of how bacterial cells, such as *E. coli* respond to drastic changes in environment, such as shifts in temperature and pH. These cell populations have developed clever hedging mechanisms by maintaining a heterogenous population that can maintain fitness in varying conditions. These hedging mechanisms are relatively well understood but can be difficult to manipulate and control due to the multitude of factors involved. Therefore, a manipulatable, engineered population of bacteria that can respond to fluctuating environments would be of great utility to the field of synthetic biology. We have designed and modeled a system of *E. coli* cells that can "choose" two different states based on asymmetric distribution of proteins during cell division. We hope that this system can be used to develop a cell population that can resist drastic changes in temperature, pH, or chemicals based on the chosen machinery. This will provide a versatile module for scientists to implement when engineering bacteria for real world uses.

Asymmetric Division | Genetically Engineered *E. coli* | Environmentally Resistant Bacteria

Introduction

In bacterial cell populations, phenotypic diversity is advantageous because it allows a population to maintain fitness in fluctuating environmental conditions. The ability for a population to respond dynamically to environmental changes (such as changes in pH or temperature) by changing state is often indispensable to bacterial population survival. As it is often difficult to precisely control the many environments where engineered bacteria will eventually be used, it would be advantageous to design a bacteria population that can survive drastic changes in their environment. Here we propose both a design and model of such a system: a bacterial cell population capable of asymmetric division into two populations that are fit for different environmental conditions.

In the field, several papers have demonstrated the ability of cell populations to change phenotype based on necessity. Suel et al. describe such activity in the ComK pathway of *Bacillus subtilis*, where cells can switch between a transiently differentiated competent state and a vegetative state. When in the competent state, cells can uptake DNA from the environment and then can switch out of the competent state back into the vegetative state [1, 2, 3]. However, switching in this system is very dependent on stochastic noise, whereas in our model switching is coupled to a repressor concentration which is tightly controlled. Kussell et al. model this type of responsive switching in comparison to random stochastic switching, noting that responsive switching is favored in frequently changing environments, as long as the energy cost of sensing the need to switch is low [4].

We have developed a system where a mother cell divides asymmetrically into two daughter cells. One daughter cell retains the same phenotype as the mother (type 1), whereas the other daughter cell assumes a different physiological state (type 2). After some time, all type 2 cells revert back to the initial type 1 phenotype. Type 2 cells are distinguished by high expression of what we called differentiation machinery (DM) proteins. These proteins could be any number of environmental resistance genes such as resistance to abnormal

pH, temperature, or surrounding chemicals. We have derived a series of differential equations that describe this process and modeled their behavior in COPASI [5].

Methods

Model Formulation. Figure 1 shows a schematic representation of how our engineered bacteria will undergo asymmetric division. Type 1 cells (blue circle) will divide into both a type 1 and a type 2 (red square) daughter cell. After some time, type 2 cells will revert back into the type 1 phenotype. The daughter's cell fate is determined by the number of molecules of protein R1-A1 that it inherits from the mother cell; this process will be explained in greater detail in later sections.

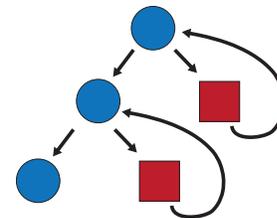


Fig. 1: Model of asymmetric division: A type 1 cell (blue circle) divides into another type 1 cell as well as a type 2 cell (red square). The type 2 cells will revert back to type 1 cells after a certain amount of time.

Figure 2 depicts our proposed genetic circuit. Repressor R1 is a strong, cooperative repressor that binds to P_a and turns off the expression of both the differentiation machinery (DM) and the Revert protein (RV). Activator A1 is a strong, cooperative activator that binds to P_a and turns on the expression of R1-A1. To ensure that a daughter cell gets a stoichiometric ratio of 1 R1 protein to 1 A1 protein after cell division, we will express them as a fusion protein (R1-A1) with a flexible linker. This set up ensures that when there is a high number of R1-A1 molecules, R1-A1 activates itself to keep itself in the high steady state and strongly represses P_a to keep DM in the low steady state. Conversely, when there is a low number of R1-A1 molecules, R1-A1 is not able to activate itself and remains in the low steady state. When R1-A1 is low, DM is highly expressed, since P_a is a constitutively active promoter and is not being repressed by R1-A1. Finally, when DM is on and the cell is in the type 2 state, the protein Revert (RV) is slowly made which weakly activates P_r and begins to slowly produce R1-A1 levels. Eventually R1-A1

Creative Commons License:

CC-BY-NC-SA 3.0

<http://creativecommons.org/licenses/by-nc-sa/3.0/>

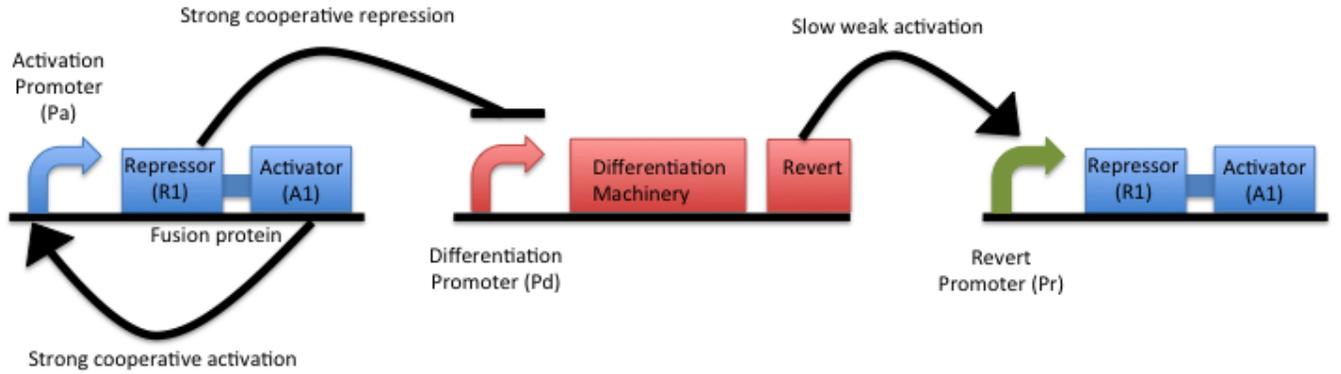
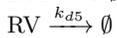
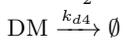
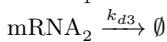
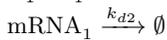
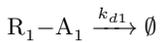
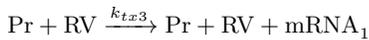
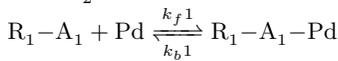
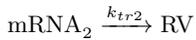
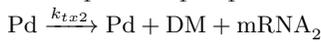
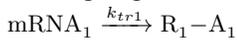
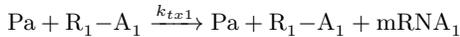


Fig. 2: Proposed Genetic Circuit: A fusion protein with a strong cooperative repressor R1 and a strong cooperative activator A1 is made by the activation promoter Pa. A1 activates Pa and turns on expression of R1-A1. R1 inhibits a separate differentiation promoter Pd. Pd, when active, produces the differentiation machinery DM and a revert protein RV. RV weakly activates a third promoter Pr, which also produces R1-A1.

will be able to activate itself and switch the cell from a type 2 phenotype back to a type 1 phenotype.

The following chemical equations were used to design the system:



Analytical Methods. To get the sharp switching that we required, we had to use both both cooperative activation and cooperative repression. We used the Hill equation for repression in [1] to model R1-A1 cooperatively repressing DM at Pa. We used the Hill equation for activation in [2] to model R1-A1 cooperatively activating its own production at Pa. To get a more switch like behavior for R1-A1 we included the mRNA intermediate with a fast translation rate and fast R1-A1 degradation to get bursty expression of R1-A1 and reduce noise at the switch point.

$$\frac{dDM}{dt} = \frac{\alpha(k_M)^n}{(k_M)^n + [R1 - A1]^n} - \beta[DM] \quad [1]$$

$$\frac{dmRNA1}{dt} = \frac{w[R1 - A1]^n}{(k_M)^n + [R1 - A1]^n} - z[mRNA1] \quad [2]$$

$$\frac{dR1 - A1}{dt} = P[mRNA1] + Y[mRNA1] - Q[R1 - A1] \quad [3]$$

$$\frac{dmRNA2}{dt} = H[RV] - J[mRNA2] \quad [4]$$

Using these equations, we performed a steady state analysis of the system to determine what parameters to use. In equation [1], alpha - beta[DM] sets the max value of DM, while k_M sets the switching point. We next analyzed equations [2] and

[3] at steady state. To simplify our calculations, we assumed that the system was infinitely cooperative and evaluated the equations as the limit as n goes to infinity. We found that to produce our desired behavior, the ratio of w:z must equal to 100 and both P and Q must be large values. We used these parameters to begin modeling our system in COPASI. Finally, we tried different values of H, J and Y to find values that would cause type 2 cells to revert back to type 1 cells after 2.5-3.5 hours.

Simulations. We began by first modeling our system in deterministic mode to ensure that calculations actually produced a system with the desired behavior. After that had been established, we used the Gibson + Bruck mode to model our systems stochastically. For figure 4d we plotted 10 repetitions on the same graph. For figure 4e and 4f, it was too computationally intensive to plot repetitions, but we looked at many individual graphs and our graph is representative of what was seen across simulations.

To generate the histogram showing R1-A1 distribution upon cell division we again used Copasi in stochastic mode to model 101 R1-A1 proteins being divided into two daughter cells. We used the parameter scan to perform 150 iterations and then plotted the results in Microsoft Excel.

Results

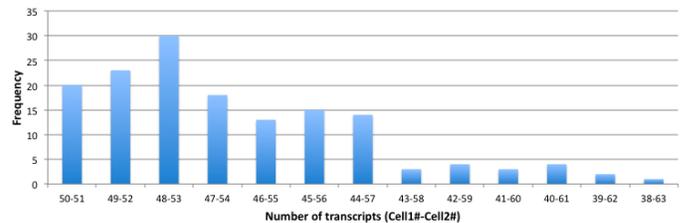
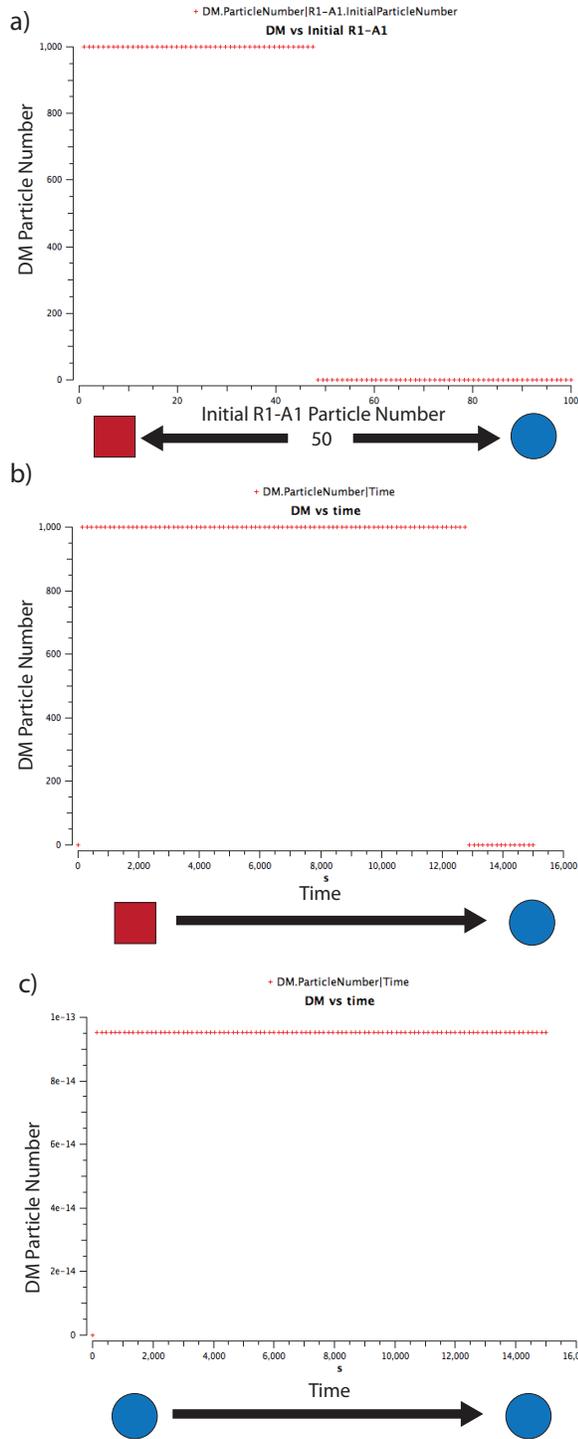


Fig. 3: Histogram showing the different distribution of R1-A1 proteins after cell division. Cells that receive more than 50 proteins will be in the high state those with less than 50 will be in the low state.

Distribution of Cell Division. The histogram in figure 3 shows a distribution of the amount of R1-A1 protein that will end

up in each of the daughter cells when the mother cell divides.

Deterministic Models



Stochastic Models

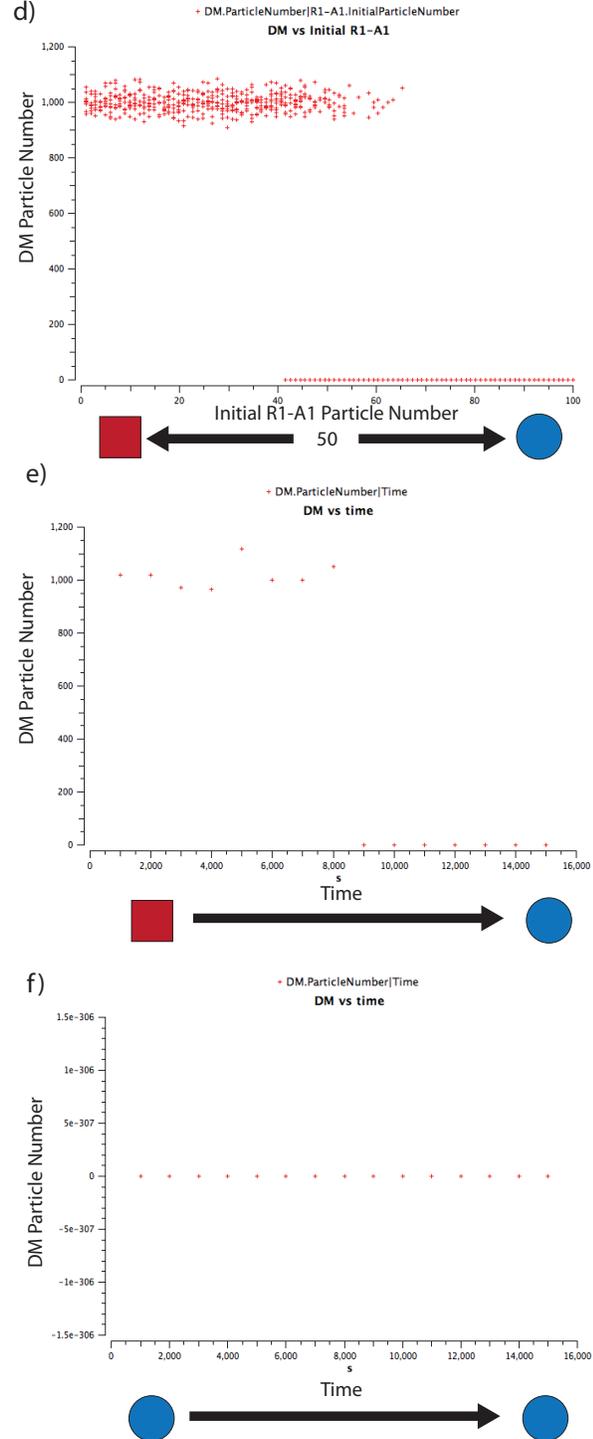


Fig. 5: Deterministic and Stochastic Simulations of differentiation machinery vs R1-A1. a) In the deterministic model when initial R1-A1 particle number is less than 50 the differentiation machinery is turned on and the cell is a type 2 cell, when R1-A1 is greater than 50 the differentiation machinery is turned off and the cell is a type 1 cell. b) Due to the slow production of R1-A1 in the type 2 state, the cell will eventually revert back to a type 1 cell. c) Cells that are in the type 1 state remain in the type 1 state. d) Stochastic simulation of a) showing a similar result, but with a wider switching range. e) Stochastic simulation of b) showing that type 2 cells revert back to type 1 cells, but quicker than in the deterministic case. f) Stochastic simulation of c) showing that cells in the type 1 state remain in the type 1 state.

The uneven distribution demonstrates that in most cases, one daughter cell will generally end up with more than 50 R1-A1 particles, produce more R1-A1 and completely turn off the differentiation machinery. The other daughter cell will generally end up with less than 50 R1-A1 particles and enter the low R1-A1 steady state where the differentiation machinery is turned on. This histogram shows how our system enables a mother cell to produce two daughter cells with different phenotypes depending on the number of R1-A1 molecules they inherit.

Deterministic Results. Figure 5a is a graph of differentiation machinery particle number vs initial R1-A1 particle number. The graph illustrates that when there are less than 50 R1-A1 particles, R1-A1 remains in the low state and the differentiation machinery is highly expressed. Conversely, when there are more than 50 R1-A1 particles, R1-A1 remains in the high state and the differentiation machinery is completely turned off. This graph demonstrates how our circuit enables asymmetric division. When a type 1 cell divides, its R1-A1 concentration will be randomly divided between the two daughter cells. One cell will receive less than 50 R1-A1 particles, whereas the other cell will receive more than 50 R1-A1 particles. The cell that receives less than 50 R1-A1 particles will not be able to repress the differentiation machinery and become a type 2 cell. The cell that receives more than 50 R1-A1 particles will keep the differentiation machinery repressed and remain a type 1 cell. Figure 5b is a graph of differentiation machinery particle number vs time when there are initially 40 molecules of R1-A1. Initially, the cell is a type 2 cell: differentiation machinery is turned on and R1-A1 particle number is low. After approximately 3.5 hours enough R1-A1 will be slowly produced by the Revert protein to activate the R1-A1 positive feedback loop. This will cause R1-A1 to enter the high steady state repressing the differentiation machinery and causing the type 2 cell to revert back to a type 1 cell. Figure 4c is a graph of the differentiation machinery particle number vs time when there are initially 80 molecules of R1-A1. This cell remains a type 1 cell for the duration of the simulation.

Stochastic Results. Figure 5d-f are stochastic simulations of the results presented in figure 5a-c. Overall, the stochastic results show the same trend as the deterministic results. The most notable difference is that in figure 5d, the differentiation machinery does not always switch from on to off at 50 R1-A1 molecules. In some cells the differentiation machinery becomes repressed with as few as 42 R1-A1 molecules and in other cases the differentiation machinery remains on even in the presence of 60 R1-A1 molecules. However, in a large population of bacteria we believe that these fluctuations will even out; even though some type 1 cells will not become type 2 cells even if they have less than 50 molecules of R1-A1, an equal number of type 1 cells will become type 2 cells even if they have more than 50 molecules of R1-A1. Additionally, another interesting observation can be observed by comparing figure 5b and 5e. In the deterministic case type 2 cells revert back to type 1 cells after approximately 3.5 hours. However, in the stochastic case type 2 cells revert back either earlier or later since there is a wider, more variable switching threshold in the stochastic simulation. In both the deterministic case and stochastic case, a type 1 cell will remain and type 1 cell for the duration of the simulation.

Discussion

Our results demonstrate a proof-of-principle genetic circuit that causes bacteria to undergo asymmetric division. We en-

vision that this circuit could be integrated in any genetically engineered bacteria to help make it resistant to fluctuating environmental conditions in real world usage. For instance, our differentiation machinery proteins could be replaced by proteins that make the bacteria more resistant to abnormal pH, temperature, or chemicals in the environment. This would allow a bacteria population to survive drastic changes in their environment. Figure 4 presents a hypothetical real world use for this system. Imagine that wild type or type 1 bacteria grow ideally in a temperature range of 34-43 degrees, but die outside of those ranges. Type 2 bacteria would grow more slowly in this temperature range, but contain genes that allow the bacteria to grow outside of this range. If the temperature suddenly jumps up to 48 degrees the type 1 bacteria will die, however the type 2 bacteria will survive. If the temperature returns to the normal range within several hours, the type 2 bacteria can revert back to the type 1 state and repopulate. Future work, could optimize this mechanism by integrating a way for the bacteria to sense whether they should remain type 2 cells or revert back to type 1 depending on the current conditions of the environment.

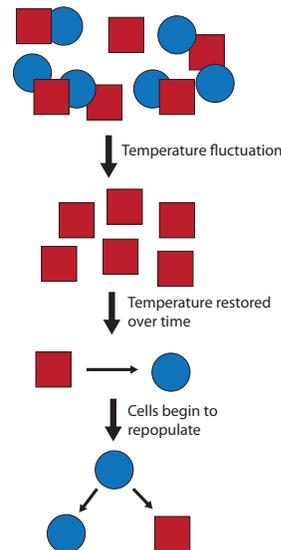


Fig. 4: Potential environmental resistance: Initially there is a heterogeneous population of type 1 and type 2 cells. However, at some point the temperature fluctuates outside of its normal range and type 1 cells die. Type 2 cells express a temperature resistance gene which allows them to survive in abnormal conditions. After the temperature returns to normal, type 2 cells can revert to type 1 cells and repopulate the environment.

Bibliography

1. Suel, G. M. et al. 2006. An excitable gene regulatory circuit induces transient cellular differentiation. *Nature* 440, 545-550.
2. Suel, G. M. et al. 2006. Tunability and noise dependence in differentiation dynamics. *Science* 315, 1716-1719.
3. Maamar, H. et al. 2007. Noise in gene expression determines cell fate in bacillus subtilis. *Science* 317, 526-529.
4. Kussell et al. 2005. Phenotypic diversity, population growth, and information in fluctuation environments. *Science* 309, 2075-2078.
5. COPASI: biochemical network simulator. <http://www.copasi.org/>

ACKNOWLEDGMENTS. We thank Adam Arkin, Ron Weiss, Noah Davidson and Alistair Boettiger for support and careful reading of this manuscript, and the Berkeley and MIT pilot PoSB class of 2010 for helpful discussions.

Novel *E. coli* mechanisms for mosquito control and programmed cell death

Dorothy Tulanont* and Amir Mehdizadeh*,

*Department of Bioengineering, University of California, Berkeley

Submitted to Principles of Synthetic Biology

Mosquitoes are known carriers of multiple disease-causing viruses and parasites and infect over 500 million people annually. By limiting or eliminating mosquito populations, instances of certain diseases such as malaria could be greatly reduced. We propose a form of biological control through the use of a genetically modified *E. coli*. These bacteria would produce toxins and attractants for mosquito larvae and be released into areas where mosquito breeding occurs. In order to prevent our modified bacteria populations from escaping, they would be programmed to undergo cell death after approximately three days. This is accomplished through the use of a timer, an internal clock for the system, and a set of toxin-antitoxin proteins. Models of our designed system showed that the engineered *E. coli* would produce toxins until it reaches a steady state and constantly expelling attractants to lure the larvae. The models also showed that the engineered *E. coli* were moderately effective at resisting mutations to the timed apoptosis mechanism. However, the chance of bacteria with our setup being able to escape cell death is too high and should not be used.

Mosquito | Programmed Cell Death | Apoptosis |

Abbreviations: IC, Intracellular; EC, Extracellular

Introduction

Mosquitoes play a significant role in human illnesses, as they are vectors of many fatal illnesses, such as malaria, filariasis, dengue, and yellow fever. Malaria alone results in over 500 million cases of illnesses and 2.7 million deaths annually with most occurring in sub-Saharan Africa [3]. Currently there are a variety of methods being employed for mosquito population control. These methods include source reduction by eliminating bodies of standing water where mosquitoes breed, either by completely removing the water or creating a flow [8]. A number of pesticides and other manmade chemicals are used to target and kill both larvae and adult mosquitoes, but they often have negative effects on other parts of the environment [4]. Another system used is biological control, where predators, pathogens and parasites that target mosquitoes and their larvae are introduced into areas with large populations [6].

One of the organisms used in biological control is *Bacillus thuringiensis* var. *israelensis* or BTI. BTI produces two δ -endotoxins, Cry4Aa and Cry4Ba that are highly toxic and have a high specificity for mosquito larvae [6]. Upon ingesting the bacteria, the proteins form crystals within the gut, killing the cells lining the stomach and as a result, the mosquito [9, 5]. We plan to design a synthetic *E. coli* that combines these toxins along with a chemical attractant for mosquito larvae. In particular, indole, which is found in several larvae food sources, will be used as an attractant [7]. We hope that releasing these modified bacteria into bodies of water where mosquitoes are known to breed would be an effective method of mosquito population control.

However, it would be both irresponsible and illegal to release genetically modified organisms or GMOs into nature, especially if they pose a hazard to other species and could potentially displace native organisms. Previous attempts of

controlling GMOs have typically involved creating infertile organisms. This allows engineers to have limited populations that cannot pass on their genetic information [1]. However, the concept of infertility does not transfer well to organisms such as bacteria that reproduce asexually via binary fission.

We propose a method of control that involves a regulated apoptosis. A timer molecule would activate two promoters which contain our apoptosis genes. One would produce an unstable antitoxin mazE and a stable toxin mazF; the other would also produce an unstable antitoxin relB and a stable toxin relE. Both mazF and relE cause cell death by cleaving mRNA strands that are necessary for cell survival. However, they have different modes of action in inhibiting translation. RelE specifically cleaves mRNA codons in the ribosomal A site. MazF cleaves mRNAs in a ribosome-independent manner, and it appears to be a sequence specific cleavage (ACA site) rather than a codon specific cleavage [2].

The antitoxin, mazE, consists of a structured N-terminal dimerization domain and an intrinsically unstructured C-terminal mazF binding domain. RelB, another antitoxin in the system, is a polypeptide that wraps around relE, preventing it from cleaving mRNA codons [2]. Thus, both the antitoxins bind directly to their respective toxins and inhibit their functions [1]. In addition to toxin-antitoxin interactions, the two toxins, will also bind to and inactivate each other via leucine zippers. Once the timer degrades, the promoter is inactivated, and the unstable antitoxins concentration will drop quickly, leaving only toxins that will cause the *E. coli* to undergo cell death. In order to determine if our proposed methods of mosquito control and programmed cell death are effective, we constructed and explored several models of our proposed systems.

Methods

Model Formulation.

Mosquito larvae killing mechanism

The *E. coli* was designed to continuously produce mosquito toxin, attractant, and efflux protein. To individually control the levels of each protein, three constitutively expressed promoters were used for Cry4Aa, the indole producing enzyme, tryptophanase, and an efflux protein. After the efflux protein is synthesized, it attaches to the *E. coli* membrane. By degrading tryptophan, tryptophanase produces indole which is expelled from the cell through the efflux proteins that are attached to the cell membrane. Cry4Aa will remain inside the *E. coli* until it is digested by a mosquito larva.

Creative Commons License:

CC-BY-NC-SA 3.0

<http://creativecommons.org/licenses/by-nc-sa/3.0/>

The chemical equations that describe this system are:

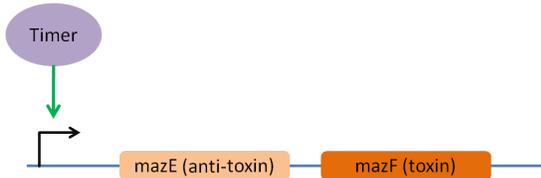
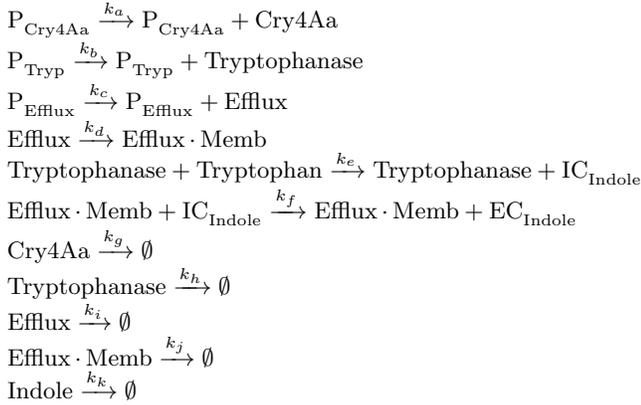


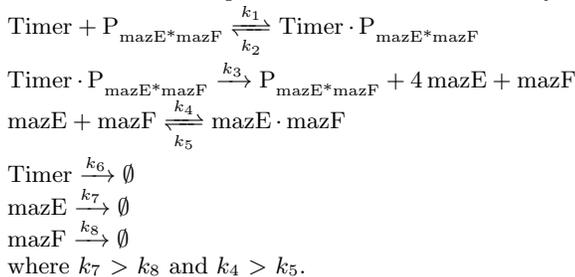
Fig. 1: MazEF gene circuit. The timer is an activator for the promoter of mazEF gene, where mazE is the antitoxin, and mazF is the toxin. The transcription and translation of this gene allows mazE protein to bind with mazF protein. When all the timer molecules degrade, transcription is stopped, and mazE begins to degrade at a higher rate than mazF. This leaves the cell to be populated with free mazF molecules, leading to apoptosis.

E. coli killing mechanism

In order to control the population size of *E. coli*, a timed apoptosis mechanism was implemented. The proposed mechanism is to introduce the mazEF toxin-antitoxin gene circuit (Figure 1) into the cell with a small molecule as an activator (Timer). A high amount of Timer is initially provided in the media where *E. coli* is stored until it is released into the wild.

MazE and mazF are the unstable antitoxin and stable toxin, respectively. When the Timer molecule induces transcription of the gene, mazE and mazF are produced. MazE and mazF have high affinity to each other and bind together.

The chemical equations that describe this system are:



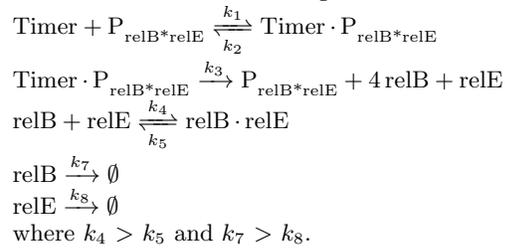
Since the Timer molecule induces the production of the toxin and antitoxin, the Timer molecule acts as a clock for the cells apoptosis. Once all the Timer within the cell degrades, the production of both toxin and antitoxin stops. Since the

antitoxin degrades faster than the toxin, the free toxin population level rises which causes cell death.

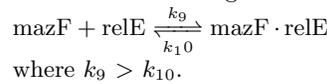
However, this population of *E. coli* is susceptible to negative selection. If mazF mutates and loses its toxicity, the cells will no longer undergo timed apoptosis and would continue to grow uncontrolled. To reduce this possibility, our original system was modified with the addition of a second set of toxin-antitoxin proteins, relBE (Figure 2).

The Timer activates both gene circuits, mazEF and relBE. mazE and relB are unstable antitoxins which binds to the toxins, mazF and relE, respectively. Moreover, a leucine zipper is inserted to the end of the two toxin genes. This allows mazF and relE to occasionally bind with each other and stop their mRNA cleavage mechanisms. Like the gene circuit in Figure 1, once all the Timer has degraded, more toxins will be present than antitoxins due to the toxins slow degradation rate. The toxin will cleave mRNA necessary for cells survival, causing the cell to apoptose.

The additional modified gene circuit elements are:



With the binding mechanism of the leucine zipper:



The addition of the leucine zipper allows the cells to undergo apoptosis at a slower rate. When the two toxins combine, the toxin loses its efficacy. If there is a mutation in either one of the toxin gene, the binding mechanism of the leucine zipper stops which allows for more build-up of the other toxin.

We also modelled the competition between populations of normal cells, populations that mutated and no longer express one of their toxic proteins, and populations that mutated and do not express any toxic proteins.

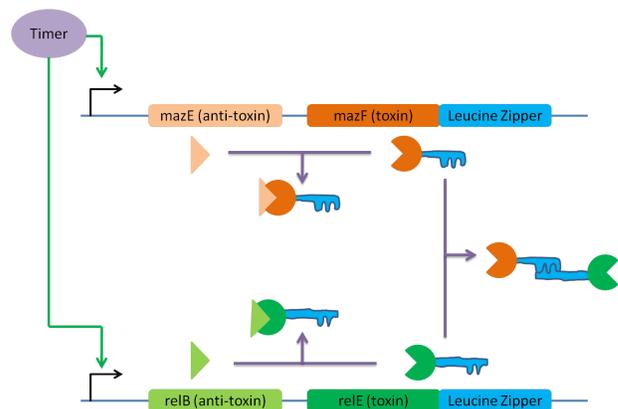


Fig. 2: MazEF and relBE modified genetic circuit. In addition to interacting with their respective antitoxins, the modified toxic proteins can bind to each other via a leucine zipper.

Analytical Methods.

Mosquito larvae killing mechanism

When modelling mosquito killing mechanism, it is assumed that there is an unlimited supply of tryptophan.

The equations covering our mosquito toxin and attractant are:

$$\frac{d[Cry4Aa]}{dt} = k_a[P_{Cry4Aa}] - k_g[Cry4Aa] \quad [1]$$

$$\frac{d[Indole]}{dt} = k_f[Efflux * Memb][IC_{Indole}] \quad [2]$$

E. coli killing mechanism

When modelling our *E. coli* timed apoptosis mechanism, we made the following assumptions. Mutations are irreversible and are equally likely to occur at any base pair on the gene. Therefore, longer gene sequences are more likely to experience a mutation. Once any mutation occurs at sequence encoding for a protein, the protein is no longer produced. If any mutation occurs at a promoter, all proteins associated with that promoter are no longer produced.

We modelled our mazEF and relBE proteins to have the same rates of production and degradation. The ribosome binding sites of the antitoxin mRNA are stronger than those of the toxins, leading to an increase in the antitoxin production rate. Degradation rates for the antitoxins are also higher than toxin degradation rates. Moreover, we assumed that *E. coli* will apoptose when there is an accumulation of 10 molecules of either toxin.

With these assumptions, the equations governing *E. coli* killing mechanism are:

$$\frac{d[Timer]}{dt} = -k_1[Timer][Promoter] + k_2[Timer * Promoter] - k_6[Timer] \quad [3]$$

$$\begin{aligned} \frac{d[Toxin]}{dt} = & k_3[Timer * Promoter] - k_4[Toxin][Antitoxin] \\ & + k_5[Toxin * Antitoxin] - k_8[Toxin] \\ & - k_9[Toxin_1][Toxin_2] \\ & + k_{10}[Toxin_1 * Toxin_2] \end{aligned} \quad [4]$$

$$\begin{aligned} \frac{d[Antitoxin]}{dt} = & k_3[Timer * Promoter] - k_4[Toxin][Antitoxin] \\ & + k_5[Toxin * Antitoxin] - k_7[Antitoxin] \\ & - k_9[Toxin_1][Toxin_2] \\ & + k_{10}[Toxin_1 * Toxin_2] \end{aligned} \quad [5]$$

When modelling population competition, we made the following assumptions. Cells that have no mutations have a moderate growth rate and moderate death rate. Cells with one mutation at any of the toxic genes will grow at a slower rate and die at a faster rate with respect to the non-mutated cells. Cells that have mutated twice and lost both toxic genes, will grow at a faster rate and die at a slower rate with respect to the non-mutated cells.

With these assumptions, the equations governing population competitions are:

$$\frac{d[norm]}{dt} = b_n \frac{norm(t)}{k + norm(t) + mut_1(t) + mut_2(t)} - d_n norm(t) \quad [6]$$

$$\frac{d[mut_1]}{dt} = b_{m_1} \frac{mut_1(t)}{k + norm(t) + mut_1(t) + mut_2(t)} - d_{m_1} mut_1(t) \quad [7]$$

$$\frac{d[mut_2]}{dt} = b_{m_2} \frac{mut_2(t)}{k + norm(t) + mut_1(t) + mut_2(t)} - d_{m_2} mut_2(t) \quad [8]$$

In logistic growth modelling, the mutation rates of both mutated-once and mutated-twice from previous result were inputted into COPASI. However, the Timer wasn't integrated into this model. Thus, the model is assumed to have infinite amount of Timer in the solution.

Simulations. The simulation for the proposed mechanisms were done using COPASI. Deterministic modelling and mass-action kinetic rate laws were used for *E. coli*'s production of mosquito larvae toxin and programmed cell death simulations. For the population competition model, an abstract rate law was inputted into COPASI, and the model was run in stochastic mode.

Results

Mosquito larvae killing mechanism. As expected, internal number of Cry4Aa and indole molecules quickly reached a steady state value. The number of indole molecules released into the wild increased linearly with time (Figure 3).

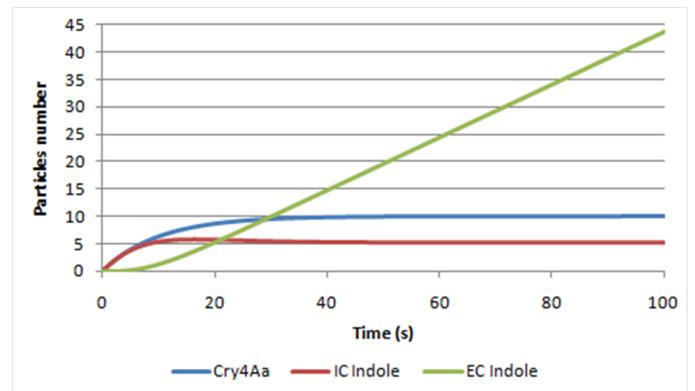


Fig. 3: *E. coli*'s production of intracellular and extracellular indole and Cry4Aa. The attractant, extracellular indole, for mosquito larvae rises linearly as time progresses. The amount of Cry4Aa toxin and intracellular level reaches steady state as a result of degradation or both the degradation and exportation, respectively.

Single set of toxin-antitoxin molecules. In this model, we saw that the number of toxic proteins slowly increased but remained well below lethal levels while timer molecules were still present. After 2 days, the number of mazF proteins began to rise rapidly and reached toxic levels at approximately

2.8 days. Eventually, a maximum level of 30 was reached, and mazF population began to decline since the production of the antitoxin had ceased. See Figure 4.

Double set of toxin-antitoxin molecules. The number of toxic proteins slowly increased while the Timer was present but at a rate slower than the previous single set model. MazF and relE levels were exactly the same since we modelled both proteins with the same rates. After 2 days, the number of mazF and relE began to increase more rapidly and reached toxic

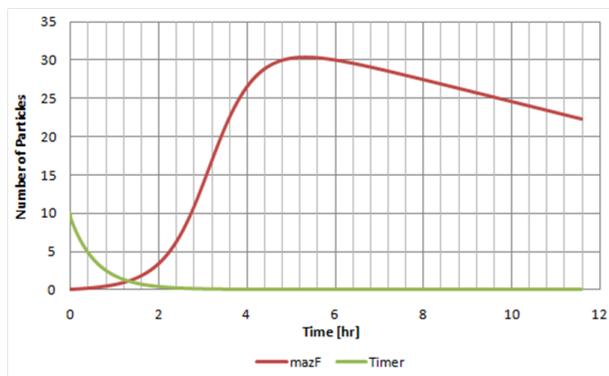


Fig. 4: Model of single toxin-antitoxin pair. Number of mazF protein reaches toxic levels within 2.8 days and rises rapidly to 30 mazF proteins.

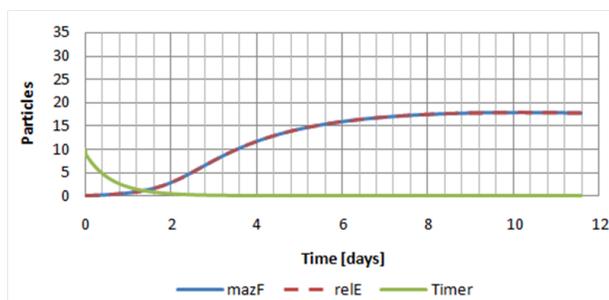


Fig. 5: Model of double toxin-antitoxin pair without mutation. The maximum amount of toxin for both toxin protein reaches 17 in 3.6 days, and the rate at which the maximum level is reached is much slower than the single set of toxin-antitoxin molecules.

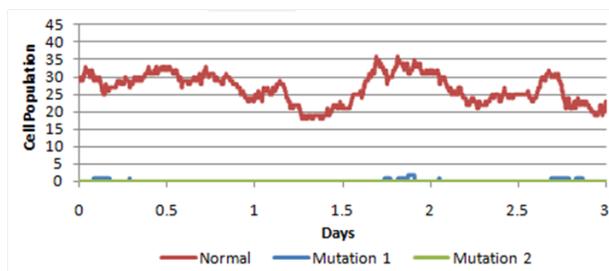


Fig. 6: Logistic growth of mutated and non-mutated populations. The growth of the non-mutated cells remained high while the mutated-once cells had occasional peaks of 1 or 2 cells. The mutated-twice cell growth was not observed in this specific simulation. In very rare cases, cells that lost both toxic proteins would begin to grow and quickly out-compete the other populations.

levels at about 3.6 days. A maximum level of 17 was reached for both proteins, and their numbers began to slowly decline afterwards. It is worth noting that both models accumulated fatal amounts of toxins, but the single set system reached higher levels of toxicity and did so faster (Figure 5).

Population Competition. The competition model was run stochastically 1000 times. The typical results is shown in Figure 6. The population of bacteria with no mutations remained near its initial levels. In nearly all the simulations, the single mutant cell populations stayed at 0 for the majority of time, with occasional growths to 1 or 2 and decreased back to 0. Rarely, this cell would mutate one more time and became a cell that no longer expressed any toxins. This event occurred approximately once for every 100 simulations.

Probabilities of losing genes coding for toxins. Based on the rate of mutations per base pair per second and the number of base pairs in the genes responsible for the promoter, toxin and antitoxin, the probability of a mutation occurring at each of these sequences was calculated. See Table 1.

Using these results, we also determined the probability of losing both toxic genes with respect to time (Figure 7). The longer the *E. coli* live, the more likely that both toxic genes will be mutated. However, the probability of mutation in both mazF and relE is still very low at 3.6 days.

Genes in the Cell	No Mutation	Only mazF Mutates	Only relE Mutates	Both mazF and relE Mutates
mazEF	0.9998	4.5726E-05	-	-
relBE	0.9998	-	3.8786E-05	-
mazEF and relBE	0.9996	5.9056E-05	5.0092E-05	1.4795E-09

Table 1: Mutation probability of different cell condition. The probability of both mazF and relE mutated in cells that have both toxin gene is much smaller than the probability of it mutating once.

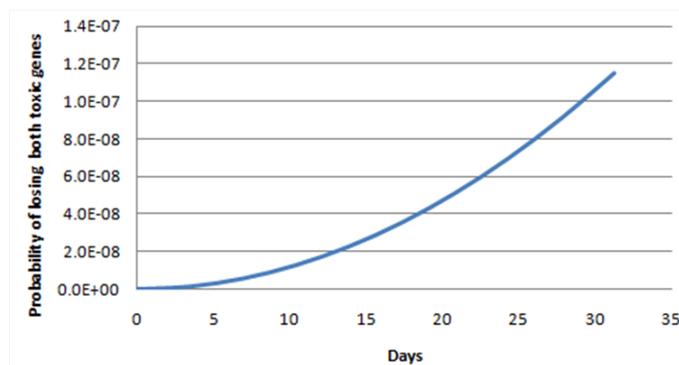


Fig. 7: The probability of mutating 2 genes. As time progresses, the chance of mutating both toxic genes increases. However, at 3.6 days, the time at which the engineered *E. coli* apoptose, the probability still remains low (1.5×10^{-9}).

Discussion

Mosquito larvae killing mechanism. Our simple mosquito larvae attractant and toxin system appeared to be an effective system. By continually producing the needed molecules, *E. coli* is always prepared to attract and poison larvae. However, constant production can put a strain on the cells and our engineered bacteria may run out of the limited resources to create the necessary proteins. If this occurs before the *E. coli* are detected and eaten by the larvae, the engineered bacteria would no longer be effective at reducing the mosquito larvae population. If larvae presence could be detected by our cells and used as a trigger to start production, the efficiency of our bacteria would greatly increase.

***E. coli* killing mechanism.** The use of a toxin-antitoxin gene as part of our programmed cell death mechanism provided multiple advantages over a simple toxin gene setup. What was particularly beneficial about the toxin-antitoxin system is the addictive properties it has. That is, once the gene starts to be expressed, the cell begins to depend on continued expression for survival because it becomes very unlikely to stop expression without initiating apoptosis. For example, when the Timer molecules in the system is depleted, the promoter is inactivated, and both toxin and antitoxin levels begin to drop. But since the antitoxin degrades quicker, it leaves more free toxins which lead to cell death. The same effect would be observed if a mutation occurs at the promoter, or if the entire plasmid with the toxin-antitoxin gene is expelled. If a mutation occurs at the location responsible for expressing the antitoxin gene, this would also create a fatal surplus of free toxin molecules. The only way the engineered *E. coli* could survive after gene expression begins is if a mutation occurs at the gene responsible for the toxin with no other mutation at the promoter or antitoxin regions.

E. coli has a higher chance of surviving when there is less free toxin molecules; therefore, the chance of survival increases as the rates of mechanisms that reduce free toxin increase. These include toxin degradation, toxin binding with antitoxin, and two different toxins binding together. Cells with a single mutated toxin have a lower chance of surviving compared to the non-mutated cells. This is because the binding mechanism of the leucine zipper ceases which allows the build-up of the non-mutated toxin. Cells with this mutation not only obtain higher level of the non-mutated toxin, but they also experience a faster build up. Since the chance of survival is lower with one toxin present, rather than two, this kind of mutation would not be evolutionarily selected.

If both toxic genes are mutated, the bacteria would divide faster and die at lower rates. This double mutated *E. coli* could potentially outcompete the non-mutated populations. The probability of mutations occurring are shown in Table 1. The chance that the mutations occur only in *mazF* and *relE* is 1.48×10^{-9} . While this number is low, we saw from our population competition model that with our current system, it is possible for double mutant cells to grow and form a large population.

To make a more robust system, one can increase the number of toxin-antitoxin pairs in the cell. There are several pairs

of such genes in *E. coli* including *chpB-chpIK*, *yefM-yoeB*, *dinJ-yafQ* and *ecnA-ecnB* [1]. However, increasing the number of toxin-antitoxin genes may not be sufficient. The system would have to be designed in a manner where the probability of surviving with $n - 1$ toxins being expressed, is smaller than the probability of surviving with n toxins present. This could be done through some sort of toxin-toxin interaction mechanisms.

This system for programmed cell death could be adapted for a variety of other cases. In order to increase the amount of time the *E. coli* lives in the wild, the degradation rate of Timer can be reduced. Another method would be to provide a cascade of promoters where the production of Timer is at the end of the circuit. Not only would a cascade create a time delay, but it could also be used to allow for other signals which could either stop or extend the timer.

ACKNOWLEDGMENTS. We thank Adam Arkin, Ron Weiss, Alistair Boettiger, Noah Davidson for support and careful reading of this manuscript, and the Berkeley and MIT pilot PoSB class of 2010 for helpful discussions.

References

1. H Engelberg-Kulka, S Amitai, I Kolodkin-Gal, and R Hazan. Bacterial programmed cell death and multicellular behavior in bacteria. *PLoS Genetics*, 2(10):e135, October 2006.
2. H Engelberg-Kulka, R Hazan, and S Amitai. *mazEF*: a chromosomal toxin-antitoxin module that triggers programmed cell death in bacteria. *Journal of Cell Science*, 118:4327–4332, September 2005.
3. R Holt et al. The genome sequence of the malaria mosquito *Anopheles gambiae*. *Science*, 298:471–473, October 2002.
4. J Hemingway and S H P P Karunaratne. Mosquito carboxylesterases: a review of the molecular biology and biochemistry of a major insecticide resistance mechanism. *Medical and Veterinary Entomology*, 12(1):1–12, January 1998.
5. Y Kanintronkul, I Sramala, G Katzenmeier, S Panyim, and C Angsuthanasombat. Specific mutations within the 45 loop of the *Bacillus thuringiensis* cry4b toxin reveal a crucial role for *asn-166* and *tyr-170*. *Molecular Biotechnology*, 24(1):11–19, May 2003.
6. G G Marten and J W Reid. Cyclopoid copepods. *American Mosquito Control Association*, 23(2):65–92, June 2007.
7. Y Xia, G Wang, D Buscariollo, R J Pitts, H Wenger, and L J Zwiebel. The molecular and cellular basis of olfactory-driven behavior in *Anopheles gambiae* larvae. *PNAS*, 105(17):6433–6438, April 2008.
8. M Yohannes, M Haile, T A Ghebreyesus, K. H. Witten, A Getachew, P Byass, and S W Lindsay. Can source reduction of mosquito larval habitat reduce malaria transmission in tigray, ethiopia? *A European Journal*, 10(12):1274–1285, December 2005.
9. H Yoshisue, T Fukada, K Yoshida, K Sen, S Kurosawa, H Sakai, and T Komano. Transcriptional regulation of *Bacillus thuringiensis* subsp. *Israelensis* mosquito larvicidal crystal protein gene *cryIva*. *Journal of Bacteriology*, 175(9):2750–2753, May 1993.

Modeling Cell-to-Cell Communication: Lux Quorum Sensing and Notch/Delta

Michel I. Nofal* and Andrew R. Saarni*,

*Department of Bioengineering, University of California, Berkeley

Submitted to Principles of Synthetic Biology

Cell-to-cell communication is a critical component of complex biological systems. Both prokaryotic and eukaryotic systems use signaling methods to allow cells to act in a coordinated fashion. In prokaryotes, quorum sensing through the lux operon allows single cells to sense trends in the cell population as a whole. In eukaryotes, signaling is achieved on a more local level, for example through Delta/Notch interactions. However, signaling is not well understood and cannot be easily engineered in synthetic gene circuits. In this paper, we describe the design and modeling of two distinct signaling systems: lux quorum sensing and Delta/Notch signaling. Further, we designed a band-pass filter responding to different input levels of Delta, and we used this engineered cellular response to examine higher-order multicellular systems with rules mimicking those of Conways Game of Life.

Signaling | Juxtacrine | Quorum | Notch | Lux | Model | Conway

Abbreviations: AHL, acyl homoserine lactone; GFP, green fluorescent protein

Introduction

Organisms are able to reproducibly create complex patterns by exploiting various signaling systems. In prokaryotes, quorum sensing is used to coordinate group behavior by using information about collective cell density to influence individual cell behavior. The prototypical quorum sensing system is the lux operon from *V. fischeri*. [1] In eukaryotes, juxtacrine, or contact-dependent, signaling is far more prevalent. A particularly well-known system of juxtacrine signaling is the NOTCH/Delta pathway.

The lux quorum sensing system consists primarily of LuxR and LuxI proteins, Acyl homoserine lactone (AHL), and the lux promoter, called the luxBox. AHL acts as a signaling molecule which binds to LuxR. [1] Once bound, the complex dimerizes, binds to the luxBox and promotes increased production of an output protein (in our case GFP), LuxR and LuxI. [2] LuxI produces more AHL which can then bind to LuxR and in this way, the system experiences positive feedback and will maintain the high output state once induced by a sufficient amount of AHL. The excess, unbound AHL is then free to diffuse into the environment and continue to propagate the signal cascade through the population.

Juxtacrine signaling through Notch/Delta is, unlike quorum sensing, restricted to only local signaling and spatial propagation of a cascade through this system depends on continuous cell-to-cell contact. Notch protein is found on the cell membrane and can interact with its ligand, Delta, which is also on the membrane and can be located either in trans or in cis to Notch. Trans binding initiates the Notch signal cascade, which consists of enzymatic cleavage of the intracellular Notch domain and subsequent transcriptional activation of target genes. Cis binding, on the other hand, inactivates Notch. [3] Sprinzak et al. have studied Delta/Notch cell-to-cell communication in an attempt to understand the role of signaling in the context of pattern formation given differences between cells in initial conditions. They examine how small initial differences can turn into a population of cells divided into sender cells which deliver the Delta ligand signal and

receiver cells which receive this signal through many Notch receptors. [4]

In this paper, we have developed stochastic single-cell models of both lux-based quorum sensing and NOTCH/delta signaling. We then further developed the NOTCH/Delta signaling model into a small, multicellular, spatial simulation governed by a set of rules mimicking Conways Game of Life. In Conways Game of Life, cells survive only if surrounded by a moderate number of neighbors. Otherwise, they die of starvation or overcrowding. [5] To mimic such behavior for use in our multi-cellular system, we developed a band-pass filtering behavior on the single cell level for human cells similar to those used in by Sprinzak et al.

Methods

Lux Quorum Sensing. In quorum sensing, an individual cell can either be in the ON state, meaning that the quorum circuit has been sufficiently activated, or the OFF state, meaning that the quorum circuit is effectively disabled. In the OFF state, the cell produces a low, basal level of both luxR and GFP. LuxI (and therefore AHL) is also produced at basal lev-

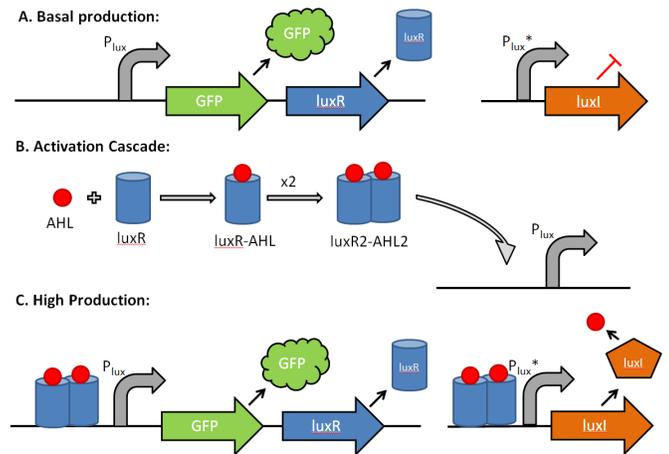


Fig. 1: A. OFF state, corresponding to basal expression of luxR and GFP and no expression of luxI. B. The activation cascade: luxR binds AHL, luxR-AHL dimerizes, luxR2-AHL2 dimer binds luxPr and enhances transcription. C. ON state, corresponding to activated expression of luxR, GFP and luxI.

Creative Commons License:

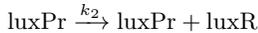
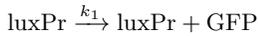
CC-BY-NC-SA 3.0

<http://creativecommons.org/licenses/by-nc-sa/3.0/>

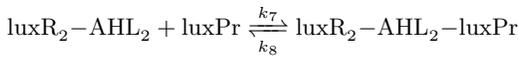
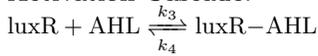
els in the standard lux operon, but in our model this basal production of luxI/AHL was removed to ensure that the only inducing effects resulted from external input rather than autoinduction. To achieve this elimination, the luxI gene was placed under the control of a mutant lux promoter which has no basal production, but which can still be activated by the binding of luxR-AHL homodimers. When an input of AHL is added, the AHL can bind to luxR and the resulting complex can dimerize and bind to the lux promoter to enhance transcription of the downstream genes, luxR, luxI and GFP. The binding of AHL to luxR is relatively slow and un-dimerization of luxR-AHL homodimers is favored so initiating the ON state requires a level of AHL that is high enough to sufficiently shift the equilibrium toward dimerization and subsequent promoter activation.

Model Reactions

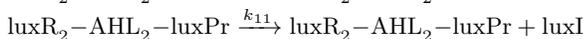
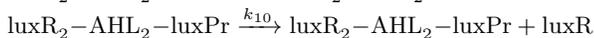
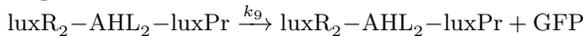
Basal Production:



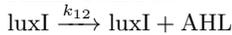
Activation Cascade:



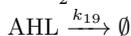
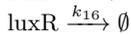
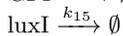
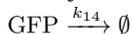
High Production:



AHL Control:



Decay:



Rate Constants:

$k_1 = 0.1$	$k_{11} = 4.8$
$k_2 = 0.1$	$k_{12} = 0.12$
$k_3 = 0.003$	$k_{13} = 0.01$
$k_4 = 0.003$	$k_{14} = 0.1$
$k_5 = 0.1$	$k_{15} = 1$
$k_6 = 10$	$k_{16} = 0.1$
$k_7 = 1$	$k_{17} = 0.3$
$k_8 = 1$	$k_{18} = 0.6$
$k_9 = 6$	$k_{19} = 0.1$
$k_{10} = 5$	

Initial Particle Numbers:

AHL = 0
AHLoutside = 0
GFP = 0
luxI = 0
luxPr = 25
luxR = 5
luxR-AHL = 0
luxR2-AHL2 = 0
luxR2-AHL2-luxPr = 0

Analytical Methods

It was assumed that, with the exception of the dimerization reaction, the activation cascade reactions are in equilibrium. The dimerization reaction favors the reverse direction and such a bias could be recreated by changing environmental conditions such as pH or temperature. The amount of AHL that diffuses out of the cell is tabulated during the simulation to allow facile integration into a multicellular simulation.

Simulations

Quorum sensing was simulated stochastically in COPASI using the Gibson-Bruck algorithm. Directly simulated mass-action kinetics were used.

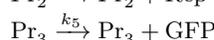
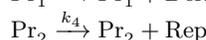
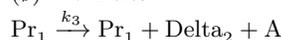
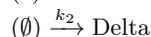
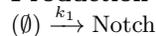
Notch/Delta. We aimed to design a system in which a population of cells laid out on a grid would have ON or OFF behavior shown by the expression or absence of GFP, respectively, according to neighbor-dependent conditions. In the design of our system, cells give and receive signals via juxtacrine signaling through Delta-Notch interactions. A cell should express GFP if one or two neighbors are ON, but should not if no neighbors are ON as if starving or if three or four neighbors are ON as if suffering from overcrowding. More specifically, a cell in the ON state would receive Delta from either one or two neighboring cells. Delta would bind to Notch receptors, causing cleave of the intracellular Notch domain. This intracellular domain travels to the nucleus where it acts as a nuclear transcription factor, activating GFP as well as Delta which travels to the cell membrane, giving the ON signal to neighboring cells.

We first designed a band-pass filter on the single cell level using Delta-Notch signaling. In order to simulate varying numbers of neighbors, we modulated Delta levels. Delta levels serve as the input in the single-cell simulation.

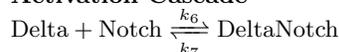
Consider the following equations:

Model Reactions

Production



Activation Cascade



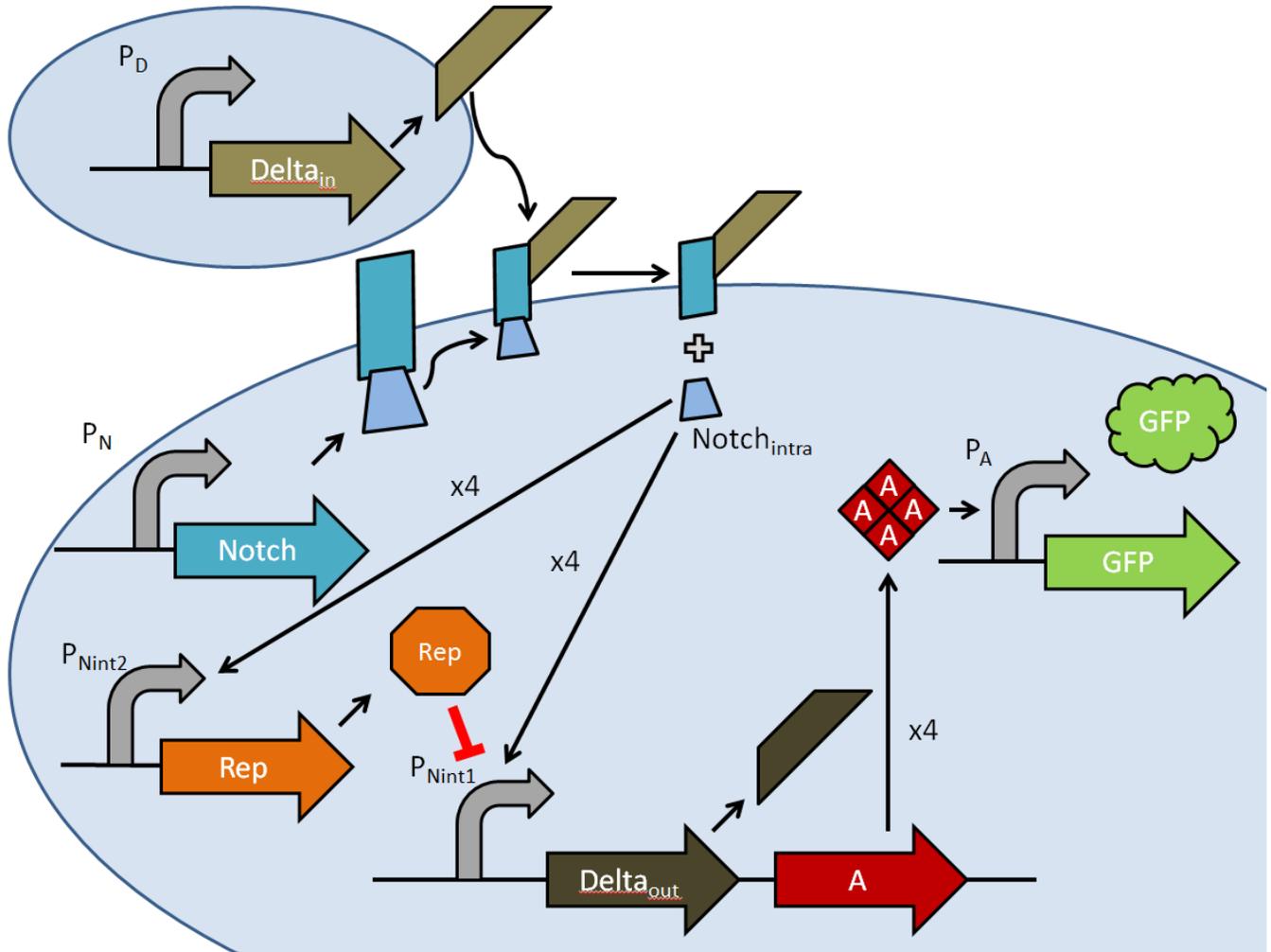
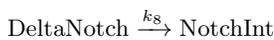
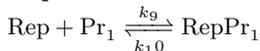


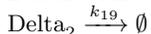
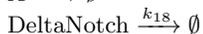
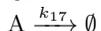
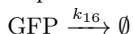
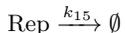
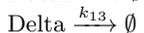
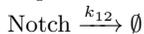
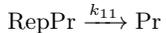
Fig. 2: Schematic for the single-cell gene circuit design based on Notch/Delta for a band-pass filter. The input is the Delta ligand and the output is GFP. When Notch binds Delta, the intracellular domain of Notch is cleaved and acts as a transcriptional activator. At very high NotchInt concentration, the NotchInt promoter 2 becomes active and produces a repressor that stops transcription from the NotchInt promoter 1.



Repression



Decay



As described above, Delta binds Notch receptors causing cleavage of the Notch intracellular domain, which, at low lev-

els, acts as a transcription factor to stimulate the expression of Delta and Protein A by inducing Promoter 1. Delta does not act in cis, so expression of Delta in the single-cell simulation is essentially meaningless. High levels of Protein A, an activator protein which tetramerizes and binds to Promoter A, yield GFP expression. However, at higher levels of the Notch transcription factor, Promoter 2 is induced, activating the Repressor protein, which represses Promoter 1 by competitively binding to it and arresting expression of Delta and Protein A. Consequently, GFP expression stops until expression of Protein A restarts.

In the multicellular system, we represented cells in a grid and assumed neighboring cells included the cells to the left and right and to the top and bottom in the grid. Diagonally neighboring cells were not considered neighbors in our model. The sizes of cell populations explored ranged from two to twelve, and the corresponding grid sizes ranged from two columns and one row to four columns and three rows. As our input in all of our models, we used the cells in the first column as sender cells. These cells behave exhibit the behavior of the cell in the single-cell system and do not respond to other cells. The other cells respond to these sender cells using the sum of

the delta levels of all neighbors as an input.

Analytical Methods

To achieve the difference in behavior between Promoter 1 and Promoter 2, we used the following rate laws:

$$k_3 = \alpha_1[Pr1] \frac{[NotchInt]^4}{\beta_1^4 + [NotchInt]^4} \quad [1]$$

$$k_4 = \alpha_2[Pr2] \frac{[NotchInt]^4}{\beta_2^4 + [NotchInt]^4} \quad [2]$$

In addition, we used a rate law of identical form for Promoter 3:

$$k_b = \alpha_3[Pr3] \frac{[A]^4}{\beta_3^4 + [A]^4} \quad [3]$$

α is proportional to maximum expression, while β determines at what point expression switches on. We assume that the Notch transcription factor somehow produces a switch-like response, perhaps through activation of a missing intermediate transcription factor with a high Hill coefficient (not included to avoid unnecessary complication of the model). This switch-like response is incorporated in the model by using a high number for the Hill coefficient. Importantly, the difference between Beta1 and Beta2 is significant. It is the difference between these two parameters which determines the levels of Notch transcription factor at which the system switches ON and OFF. A larger difference between the two variables results in a larger ON region. In addition, we assumed that all mRNA reactions occur at the same rate without any interference, and because all interactions between different molecular species occur at the transcriptional level, we were able to omit them from our models. Also, we assumed the cell uses some sort of negative feedback to maintain constant levels of Notch receptor on the cell surface. This assumption is reflected in the rate equation for the production of Notch.

$$k_1 = \frac{\alpha_4^4}{\beta_4^4 + [Notch]^4} \quad [4]$$

A similar equation is used to maintain constant levels of extracellular Delta, useful in controlling the input level over time.

$$k_2 = \frac{\alpha_5^4}{\beta_5^4 + [Delta]^4} \quad [5]$$

Equation 5 is of course only used in the single cell system and as the input to the sender cells in the multi-cellular system. All other cells use the Delta levels of neighboring cells as their dynamic input.

In the multi-cellular system, a few simplifying assumptions were made. First, any cell was able to bind with the whole pool of Delta ligands expressed by its neighbor. This means that any given ligand is visible by any neighbor rather than just one of the four. In addition, cells on the edges expressed the same level of Delta ligand, given an induced Promoter 1, as cells in the not touching any walls. Finally, if a previously bound Notch receptor and Delta ligand unbound, the Delta ligand would be returned randomly to any of its neighbors

regardless of where it came from.

Rate Constants:

$k_6 = 1$	$k_{16} = 0.1$
$k_7 = 1$	$k_{17} = 0.1$
$k_8 = 0.05$	$k_{18} = 0.1$
$k_9 = 1$	$k_{19} = 0.1$
$k_{10} = 0.05$	$\alpha_1 = 0.2$
$k_{11} = 0.2$	$\alpha_2 = 2.5$
$k_{12} = 0.1$	$\alpha_3 = 1$
$k_{13} = 0.1$	$\beta_1 = 40$
$k_{14} = 0.04$	$\beta_2 = 150$
$k_{15} = 0.2$	$\beta_3 = 100$
$\alpha_4 = 200$	$\beta_4 = 25$
$\alpha_5 = \text{input}$	$\beta_5 = 25$

Initial Particle Numbers:

Notch = 50
Delta = 50
DeltaNotch = 0
NotchInt = 0
$Pr_1 = 100$
$Pr_2 = 20$
$Pr_3 = 20$
Rep = 0
$RepPr_1 = 0$
A = 0
GFP = 0

Simulations

We used COPASI to simulate the single-cell system. We used stochastic simulation (Gibson and Bruck) to analyze the reproducibility of our results, and we used custom rate laws for specific reactions, derived from analytical approaches used to engineer specific promoter responses necessary for our system.

To run simulations in the multi-cellular system, we adapted a Java software package written by Dr. J. Christopher Anderson for a Genetic Devices Bioengineering course at U.C. Berkeley. The simulation runs very similar to simulations in COPASI. Differences exist in the graphing capabilities and possibly in the intricacies of the stochastic simulation. More importantly, programming the simulation in Java allows better control of the spatial component of the multi-cellular system and is more capable of handling simulations involving up to 162 chemical reactions.

Results

Lux Quorum Sensing.

Input/Output Relationships

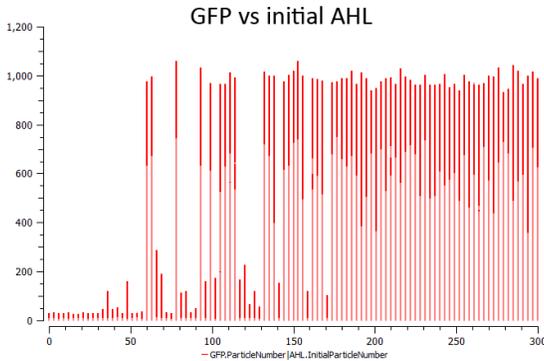


Fig. 3: GFP vs AHL initial. At low initial AHL concentrations, the cell remains in the OFF state. For intermediate values, the cell can either become ON or stay OFF. At high values, the cell always switches to the ON state.

As shown in Figure 3, at low levels of initial AHL, all cells remain off. At intermediate levels, the system becomes unstable and, with roughly equal probability, will either remain OFF or switch ON. Finally, at high levels of AHL, the equilibrium is shifted far enough to ensure that the cell will always turn ON.

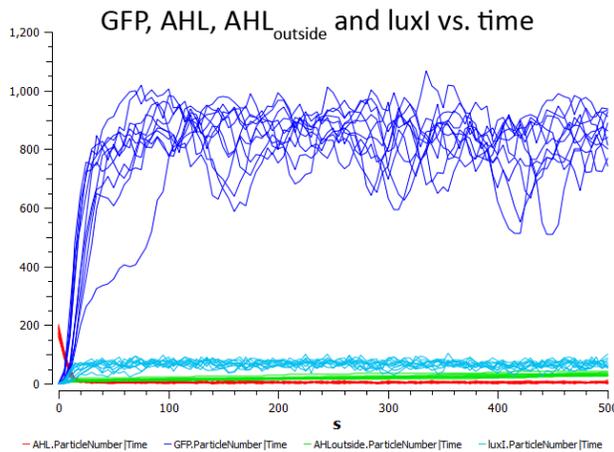


Fig. 4: Timecourse Graph. Various traces corresponding to initial values of AHL from 150-200. With these parameters, once a cell is switched ON, it remains ON.

Figure 4 shows that once switched ON, the cell will remain in that state. If two parameters are adjusted slightly, particularly changing k_{11} from 4.8 to 4.4 and k_{17} from 0.3 to 0.5, then the ON response can no longer be sustained by the same initial amounts of AHL. The result is a pulse of GFP that has magnitude and duration that are roughly proportional to the amount of initial AHL, as seen in Figure 5.

The qualitatively different types of behavior that emerge from this system as a result of minor parameter modifications

suggest that the system can be tweaked to serve a variety of purposes.

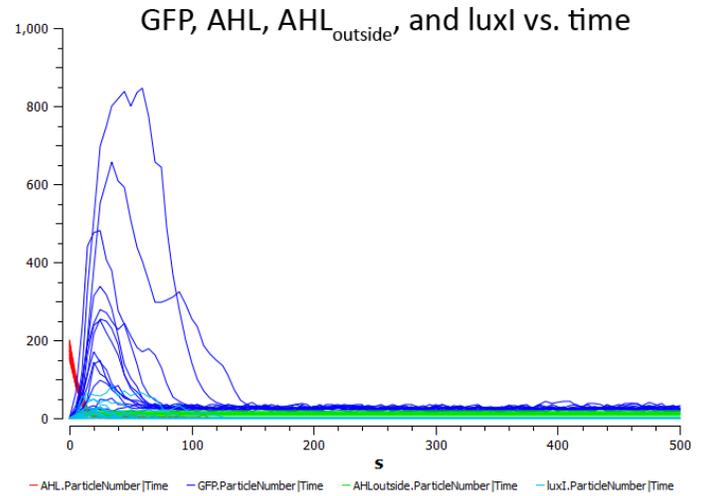


Fig. 5: Timecourse Graph. GFP pulses corresponding to initial values of AHL from 150-200.

Notch/Delta Signaling.

Single Cell Band-Pass Filter

In order for a wide range of input Delta levels to induce varying downstream effects, the range of the signal cannot be diminished too much during any step in the signaling cascade. Using Notch levels that are too low because constitutive expression of Notch is not high enough destroys the band-pass filter because at moderate levels of Delta, Notch receptors are saturated. In this case, the cell cannot produce high levels of Notch transcription factor, and Promoter 2, which requires high levels of Notch transcription factor to produce the Repressor protein, cannot be induced. As shown in Figure 6, the careful engineering of promoter responses and the simplifying assumptions made in the design of the system allow the cell to exhibit a band-pass response to varying input levels of Delta ligand. In Figure 6A, the production constant in the rate law describing production of Delta is 35, relatively low. At this level of Delta, the cell cannot make enough Notch transcription factor to induce production of GFP and turn the cell ON. In Figure 6B, production of Delta is moderate (production constant of 75). In this environment, the cell is ON, shown through high expression levels of GFP. In Figure 6C, the cell turns OFF due to very high input levels of Delta (production constant of 125). Under these conditions, Repressor protein is produced at high levels, and consequently, Promoter 1 is completely inactivated. The full spectrum of production constants is shown in Figure 6D.

Multi-Cellular Juxtacrine Signaling System

Individual cells with the band-pass response behavior described above were organized on grids of different sides and allowed to interact with one another. Figure 7A shows the results of a system of one row of five cells. The left-most cell (at position (0,0)) was constitutively on and did not respond to its neighbor. Because the cells were in a line, none had over two neighbors. Accordingly, each cell turned on permanently,

one after the other. As can be seen in the figure, the cell at (1,0) turned on first, followed by the cell at (2,0), etc. Figures 7B and 7C, show systems of cells in 2X3 and 3X3 grids (2X3: 2 rows, three columns). In each of these systems, only one cell turns off. In the 2X3 grid, the two cells in the rightmost column both are activated simultaneously. They repress each other along with their respective neighbors until they reach a certain point, the threshold where Repressor protein is no longer produced. Stochastically, one of the two cells jumps back to the ON state and the other switches completely OFF.

In the 3X3 system, the cell in the middle is surrounded by four neighbors in the ON state and turns OFF. Upon this switching event, all other cells in the system have exactly two neighbors and stay ON. Finally, in the 3X4 system, the cell in position (1,1) turns OFF due to its many neighbors in the ON position. However, in the second and third columns, the stability of the ON and OFF states breaks down, and neighbors fluctuate between ON and OFF states, taking turns expressing GFP as shown in Figure 7D.

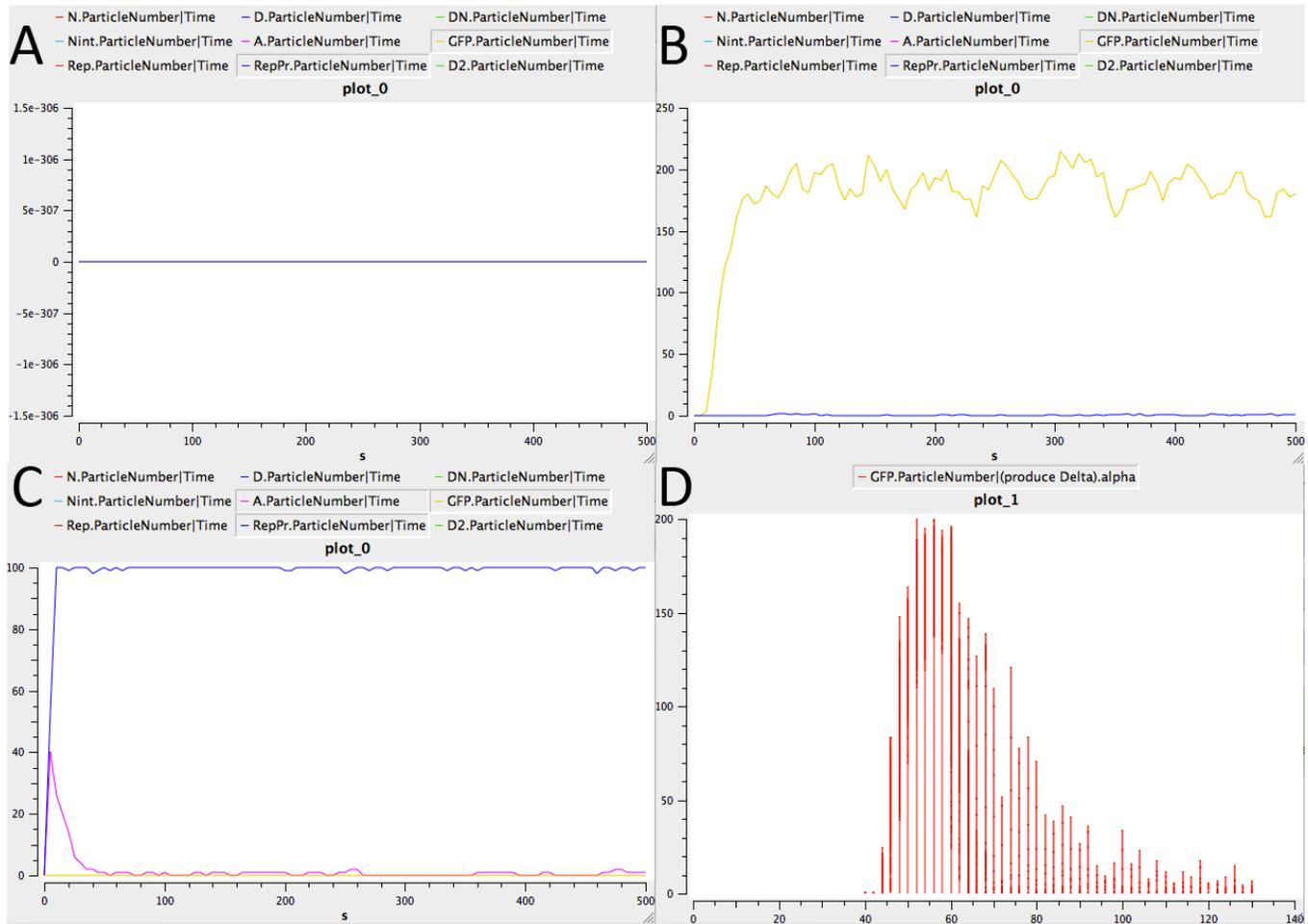


Fig. 6: COPASI simulations of the single-cell band-pass filter. (A-C) Input Delta level production constants: 35, 75, 125, respectively. (D) Input Delta level plotted against Output GFP level.

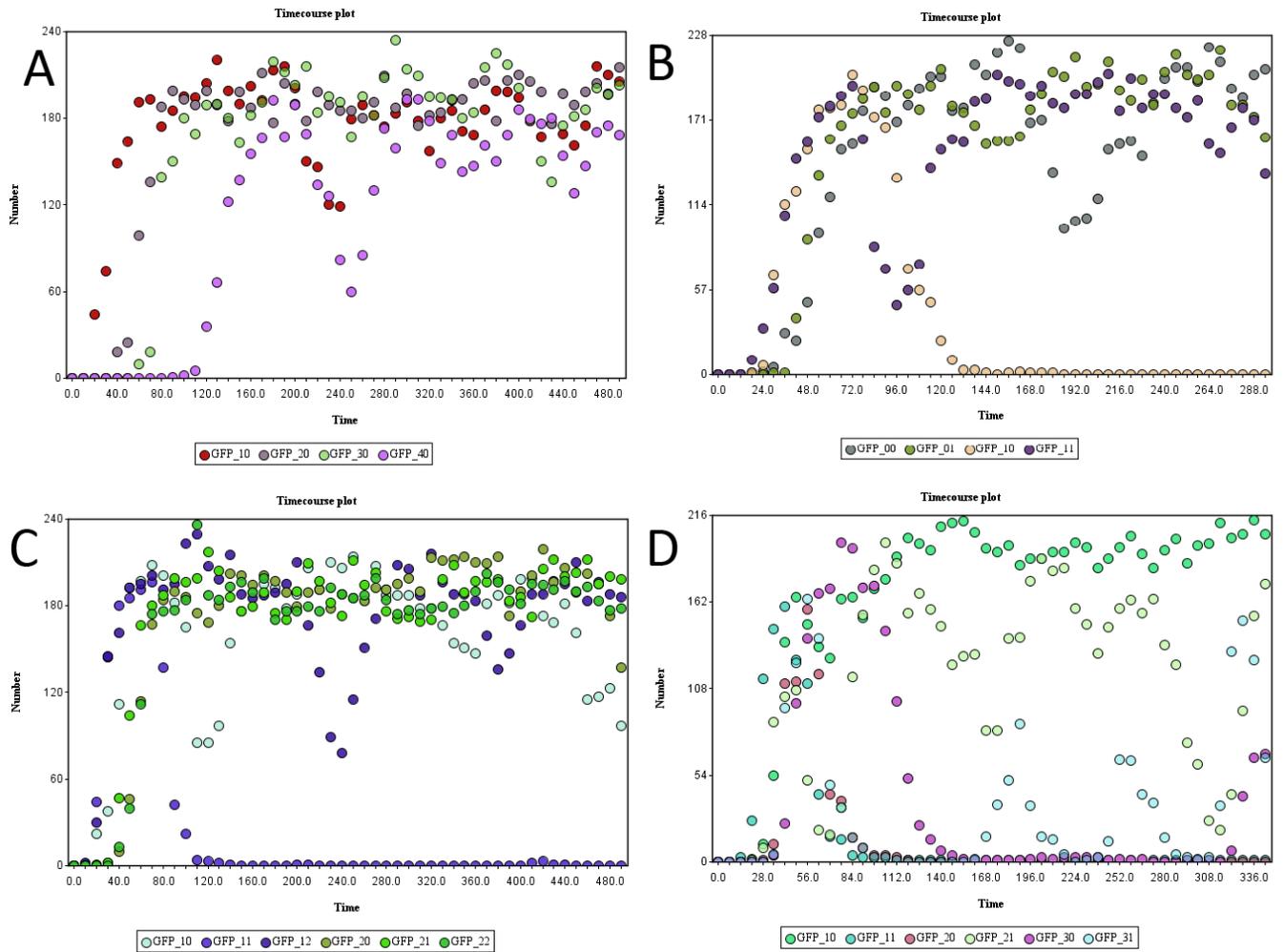


Fig. 7: Timecourse plots of GFP expression levels in different multi-cellular juxtacrine signaling systems. In each simulation, the cells in the left-most column are constitutively ON and do not respond to neighbors: (A) A system of five cells: one row, five columns. (B) A system of six cells: two rows, three columns. Data from the rightmost column was omitted. Both missing cells were constitutively ON. (C) A system of nine cells: three rows, three columns. Data from the leftmost column (sender cells) was omitted. (D) A system of twelve cells: three rows, four columns. Data from the leftmost column and bottom row was omitted. In the cases where data was left out, the missing data was either always ON or was symmetric to another data point. Rows/columns were omitted simply to make it easier to read the rest of the data points.

Discussion

The results shown in Figure 3 suggest a certain rule set for an implementation of Conway's Game of Life with quorum sensing. Each triggered cell ends up losing around 50 molecules of AHL to the environment. When the input AHL is less than 50, the cell is not triggered. At intermediate values, it can ei-

ther trigger or not, and at high values, it always triggers. For a simulation of Conway's Game of Life, having touching one adjacent activated cell (yielding roughly 50 AHL) would keep the cell off, touching 2-3 adjacent cells (100-150 AHL) could either trigger or not, and touching 4 or more cells (150-200+ AHL) would always trigger. The parameters and initial conditions would clearly have to be further optimized to create such a simulation, but it can most certainly be done.

Our design of the band-pass filter response of the single cell system yielded a very clean response which we were able

to use to design systems of multiple cells that interacted with each other according to rules analogous to those in Conway's Game of Life. Stochastic modeling of the multi-cellular systems showed interesting behaviors in different configurations, such as the two cells in the rightmost column competing for the ON state in Figure 7B.

The shortcomings of our design were shown in Figure 7D. Several modifications could be made to our model to enhance the ability of the cells to propagate the ON signal over many cells. First, a time delay could be added into the circuit so that cells are given a time to establish the ON state before recently induced neighbors stop it from reaching stability and cause unstable fluctuations to arrest signal propagation. Further, the addition of directionality of Delta ligands into the system design might contribute to more stable and deterministic behavior in the 3X4 system.

Additional work in this area might include analysis of the quorum sensing system in a spatial model such as the one used to investigate spatial Delta-Notch behavior. In addition, different starting configurations in the Delta-Notch signaling multi-cellular system could yield interesting behaviors not seen in the models described above. Furthermore, unification of global quorum sensing behavior and local band-pass

filtering behavior could be used to model more complex interactions between populations of cells and potentially generate novel patterning behavior.

ACKNOWLEDGMENTS. ANB was supported by the NSF GRFP. We thank Alistair Boettiger, Adam Arkin, Ron Weiss and Noah Davidson for support and careful reading of this manuscript, and the Berkeley and MIT pilot PoSB class of 2010 for helpful discussions.

1. N.A. Whitehead, A.M. Barnard, H. Slater, N.J. Simpson and G.P. Salmond, Quorum-sensing in Gram-negative bacteria. *FEMS Microbiol. Rev.* 25 (2001), pp. 365-404.
2. Eglund, K.A., Greenberg, E.P (1999) Quorum sensing in *Vibrio fischeri*: elements of the luxI promoter. *Mol. Microbiol.* 31, 1197-1204.
3. Artavanis-Tsakonas, S., Rand, M. D. and Lake, R. J. Notch signaling: cell fate control and signal integration in development. *Science* 284, 770-776 (1999).
4. Sprinzak, D. et al. Cis interactions between Notch and Delta generate mutually exclusive signaling states. *Nature* 25 April 2010 (doi: 10.1038/nature08959).
5. Wikipedia: Conway's Game of Life

In vivo genotyping to target therapeutics

Jennifer Brophy* and Jacob Rubens†

*Department of Biological Engineering, Massachusetts Institute of Technology, and †Department of Biology, Massachusetts Institute of Technology

Submitted to Principles of Synthetic Biology

Substantial progress has been made in the battle against HIV/AIDS. Much of this progress is due to the development of Highly Active Anti-Retroviral Therapy (HAART), which has decreased the HIV/AIDS mortality rate by inhibiting the lysogenic stage of the HIV lifecycle. However, HAART does not combat the latent HIV infection that may persist throughout a patient's lifetime. Novel strategies to address this facet of HIV infection are necessary. We have designed a Synthetic Biological therapy that is capable of diagnosing whether a cell is infected by latent HIV and subsequently either killing or sparing the cell. The therapy is delivered by modified HIV and is composed of three devices that are termed the Specificity Device, the Activation Device and the Kill Device. By stochastically modeling the synthetic system in infected and uninfected cells, we were able to optimize model parameters and achieve $100\% \pm 0.0\%$ infected cell death with $0.65\% \pm 0.50\%$ uninfected cell death. In concert with HAART, this treatment may someday lead to the eradication of HIV/AIDS.

Stochastic | Gene Expression | HIV | Genotyping

Introduction

Curing HIV/AIDS has been one of humanity's greatest challenges since the syndrome was first discovered in 1981. In 2004, there were 31.4 million people living with HIV and 2.2 million who died of AIDS [1]. By 2009, the number of people infected with HIV increased to 33.3, but the number of deaths decreased to 1.8 million [2]. Much of the progress in the battle against HIV/AIDS mortality may be attributed to the development of Highly Active Anti-Retroviral Therapy (HAART), which significantly delays the onset of AIDS, but it is not a cure for the disease.

HAART is a cocktail of anti-retroviral drugs that target various stages of the HIV-1 lifecycle. HIV-1 is the most prevalent and virulent strain of HIV and its lifecycle is characterized by two stages. First is the infection stage, in which HIV-1 infects host cells. Second is the replication stage, in which the virus replicates via either a lytic or lysogenic process. If the virus immediately enters a lytic cycle, it will quickly make copies of itself quickly make copies of itself, lysing the host cell and releasing new HIV-1 virions into the body to infect new host cells. If instead, the virus enters a lysogenic cycle, it will integrate into the host cell genome and remain dormant until stimulated to enter the lytic cycle. HAART is highly effective in combating HIV-1 lytic replication and often decreases the amount of HIV-1 in patients' blood plasma below detectable levels within several weeks of treatment initiation [3]. However, latent HIV-1 infection is not combated by HAART. Thus, a permanent reservoir of HIV-1 persists in the body even during HAART (Figure 1).

Latency Reactivation is a treatment that attempts to address this issue by coaxing proviral HIV-1 into the lysogenic cycle and subsequently inhibiting viral replication with HAART. Unfortunately, although there have been significant advances in Latency Reactivation, latent HIV-1 infection remains a significant obstacle to the eradication of HIV/AIDS [4, 5].

Synthetic Biology may offer a solution to latent infection through the engineering of a system designed to diagnose and kill latently infected host cells. The designed system utilizes three devices: (1) A Specificity Device that binds HIV gene

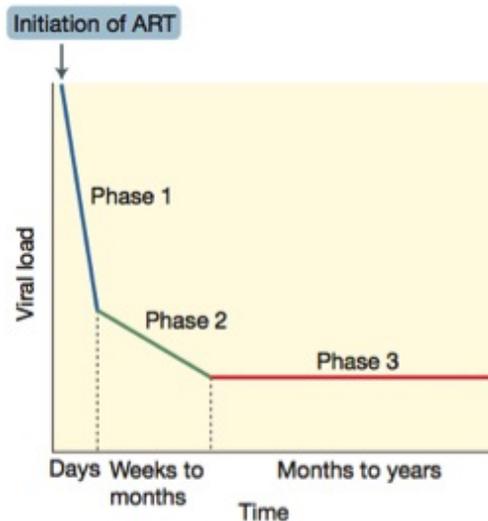


Fig. 1: Schematic representation of the HIV-1 decay curve after initiation of antiretroviral therapy [4]. Red line indicates sustained infection due to latently infected cells.

sequences integrated into the genome through zinc-finger targeting, (2) An Activation Device that activates the Kill Device with a phosphorylation cascade (3). A Kill Device that induces apoptosis in response to detection of HIV-1 infection. We expect that this synthetic system will work with current treatments, like HAART, to eradicate HIV/AIDS.

Methods

Synthetic System. The synthetic system we designed will consist of three devices. The first device, deemed the Specificity Device, will be made up of two zinc finger fusion proteins: zincA and zincB. ZincA will bind to a unique 18 nucleotide target DNA sequence that is indicative of HIV-1 integration into the host genome. ZincB will bind a different unique target DNA sequence, which is chosen to be indicative of HIV-1 genome integration and in close proximity to the zincA binding site. When both zinc finger fusion proteins bind their respective target sequences, a TEV protease tethered to zincA will cleave a peptide chain on zincB [7]. This will result in the release of PhoR from the zincB fusion protein (Figure 2A). PhoR is a phosphokinase that will translocate to the cytoplasm after cleavage [8]. Once in the cytoplasm, PhoR will phosphorylate PhoB, a component of the Activation Device. PhoB phosphorylation is the output of the Specificity Device.

Creative Commons License:

CC-BY-NC-SA 3.0

<http://creativecommons.org/licenses/by-nc-sa/3.0/>

The second device, deemed the Activation Device, will be made up of two components: a hybrid transcription factor and a hybrid promoter (Figure 2B). The hybrid transcription factor is composed of a bacterial transcription factor, PhoB, linked to a eukaryotic transcription factor, VP16. PhoB naturally regulates gene expression by inducing transcription at promoters that feature a Pho Box. VP16 is a eukaryotic transcription factor that is sufficient to cooperatively recruit transcription machinery. The Pho Box and a minimal mammalian promoter (minCMV) compose the hybrid promoter. The addition of more PhoBoxes to a hybrid promoter is known to increase the cooperativity of protein expression [9]. When the PhoB component of the hybrid transcription factor is phosphorylated, a nuclear localization tag becomes exposed that allows it to translocate into the nucleus [9]. Once in the nucleus, PhoB binds to Pho Boxes and VP16 induces transcription at the downstream MinCMV promoter. Once the MinCMV promoter is activated, the third device, deemed the Kill Device, is expressed.

The Kill Device is one protein, BAX. BAX is an apoptosis factor that creates pores in the mitochondrial membrane, releasing downstream apoptosis inducing inducers such as cytochrome C. The BAX gene is transcribed when the MinCMV promoter is active [10] (Figure 2C). After being transcribed, BAX mRNA is transported out of the nucleus and translated into a polypeptide [11]. When the MinCMV promoter is activated and BAX accumulates, apoptosis occurs and latently infected cells are killed.

All of these devices will be delivered with one HIV-derived retrovirus vector. This type of vector was chosen because it is capable of infecting the same host cells as a virulent HIV. Additionally, retroviruses can be engineered to deliver a specific

number of plasmids and peptides to target cells [6]. When the retrovirus invades a host cell, it deposits its contents into the cytoplasm. The hybrid transcription factor components of the Activation Device will remain in the cytoplasm, but the Specificity Device components display nuclear localization tags that allow them to translocate into the nucleus. The plasmid with the hybrid promoter and BAX gene will utilize transcription factor binding sites to localize to the nucleus.

Model Formulation. We modeled this system with a stochastic simulation because many of the underlying reactions involve a small number of molecules. For example, there will only be one of each zinc finger target DNA sequence per cell and the minCMV/BAX plasmid will be delivered in quantities fewer than fifty per cell. With a stochastic model, we can make small changes in the number of proteins and plasmids delivered by virus and visualize its impact on the efficacy of the proposed system. The stochastic model was formulated in MATLAB with a Gillespie algorithm based on discrete irreversible binding, dissociation, activation, production and degradation reactions.

The following reactions were used to model our system:

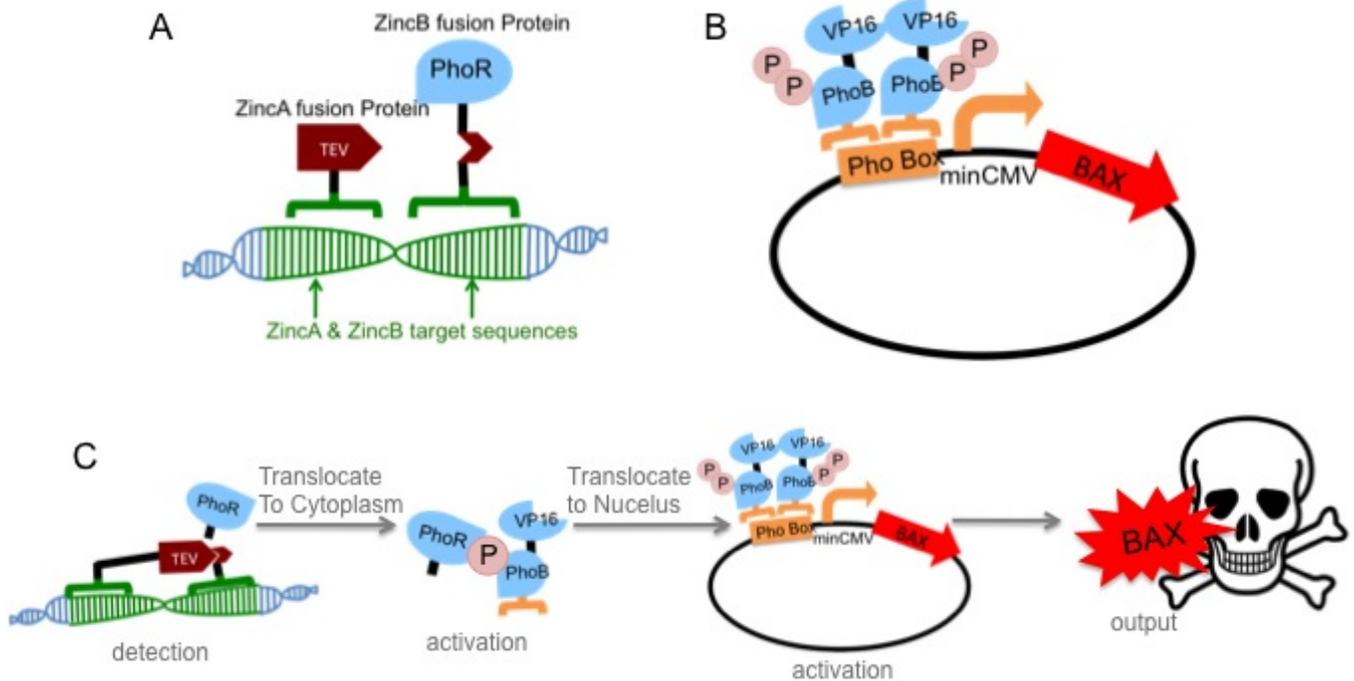
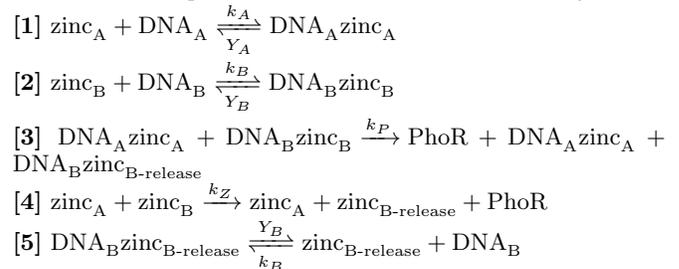
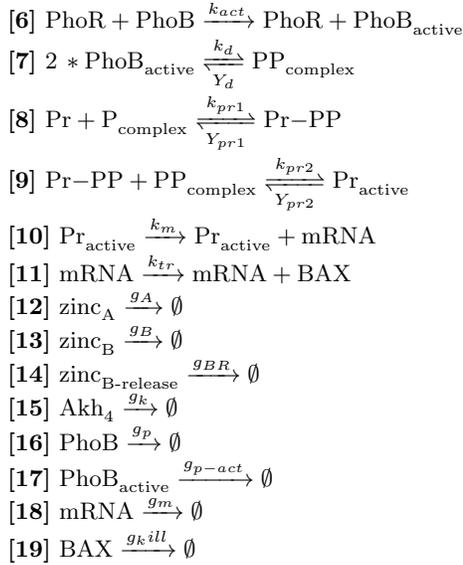


Fig. 2: Cartoon of the synthetic system **A** When both zinc finger fusion proteins bind their respective target sequences, a TEV protease tethered to zincA will cleave a peptide chain on zincB [7]. This will result in the release of PhoR from the zincB fusion protein. **B** Phosphorylated PhoB dimers will bind to the Pho Boxes of the minCMV promoter. VP16 will recruit mammalian transcription machinery and BAX will be transcribed. **C** Schematic showing the function of the entire synthetic system.



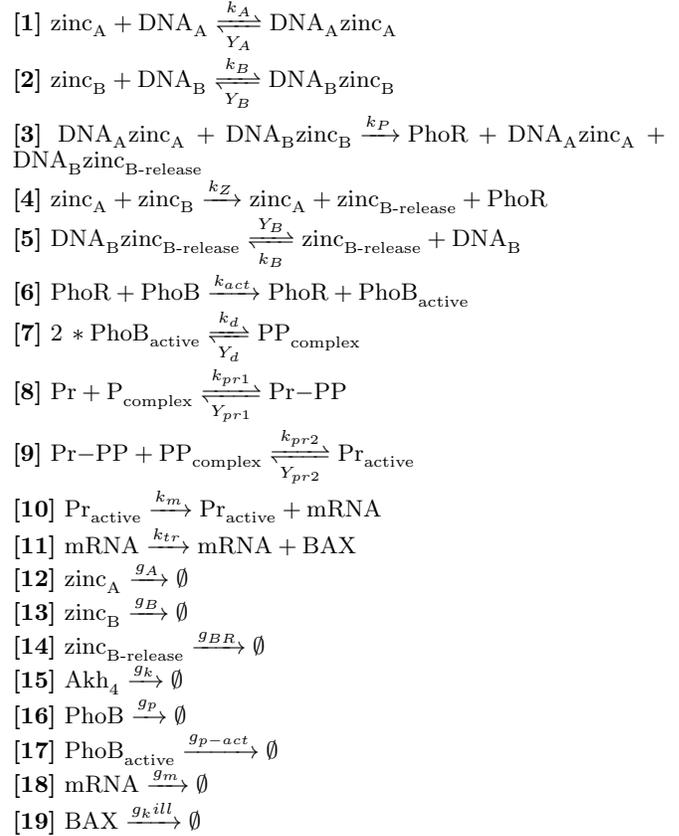
It will be assumed that a BAX protein concentration of ten molecules will result the irreversible stimulation of apoptosis because very few BAX proteins are required to induce apoptosis [11]. It is also assumed that each host cell is affected once and only once by a viral vector carrying the HIV genotyping synthetic devices. Unfortunately, the mechanism by which the addition of more Pho Boxes increases cooperativity is unknown. Thus, our model assumes that increased cooperativity of transcription is achieved through the cooperative binding of PhoB dimers to a Pho Box on the promoter. This cooperativity is modeled with equations [8] and [9] where $k_{pr1} \ll k_{pr2}$. Given that most parameter values for this system have non been determined experimentally, this model was built with semiarbitrary choice for rate constants. The relative values of rate constants were used to define the system and were chosen to be in a physiological range.

Simulations. The model tests the killing efficacy of our synthetic system. Each run of the model simulates the fate of one cell. At the onset of each stimulation a random number generator decides whether the next cell to be simulated is healthy or infected with HIV. To simulate an infected cell, the initial concentration of both zinc finger DNA target sequences (DNA_A and DNA_B) is set to one. To simulate a healthy cell, those concentrations are set to zero. Once the cell is established as healthy or infected, the stochastic simulation described above is run to see if our synthetic system is capable of recognizing the state of the host cell and delivering the proper response. To determine the best design of this synthetic system we varied a number of parameters such as: the number of synthetic device components delivered to a host cell, the degradation of active species, and the rate of "accidental" PhoR release.

For each permutation of the system, four hundred cells are simulated and the systems ability to correctly selectively kill infected cells is assessed. The percentage of infected cells killed is calculated and the percentage of uninfected cells killed is calculated. Then the model is run three more times in order to ascertain more data to ensure the percentages calculated are consistent over a large number of simulations. Ideally, the synthetic system would kill 100% of the HIV infected cells and 0% of the healthy cells. At the end of each simulation, another plot is generated to aid visualization of the efficacy of the system. In this plot, each of the four hundred cells

is represented as a circle. The outline of the circle is either red or green to indicate infected or healthy cells, respectively. The circle is filled in black to indicate cell death via apoptosis. When the synthetic system is functioning correctly, the cells will be either black with a red border or white with a green border. Any other permutation indicates that the synthetic system was not capable of correctly diagnosing and treating the host cell.

The following reactions were used to model our system:



Results

Number of molecules delivered determines the efficacy of the system. With a retroviral delivery system, we will be able to carefully control the number of synthetic device components delivered to each host cell. In order to optimize the design of our system, we varied the concentration of each synthetic device component delivered to the host cell and assessed the resultant system efficacies. Changes in the concentrations of the Specificity Device components had the greatest impact on the performance of the circuit. As seen in Figure 3, when zincA and zincB fusion proteins are delivered to the host cell in concentrations of 100 proteins/virus vector, the system fails completely. 100% \pm 0% of the infected cells are killed, but unfortunately 100% \pm 0.26% of the healthy cells are also killed (Figure 3B/E). We hypothesize that the non-specific killing is due to an excessively high concentration of zincA and zincB fusion proteins. Each fusion protein has one target DNA sequence per cell, therefore 100 copies of each fusion protein is excessive and increases the frequency of non-specific PhoR release. When large quantities of zincA and zincB fusion proteins are concentrated into the nucleus, the likelihood that the TEV protease domain on zincA will bind its cognate

cleavage site on zincB without DNA-localization increases, resulting in the death of uninfected cells.

To decrease the non-specific release of PhoR, we decreased the number of zincA and zincB fusion proteins delivered to the host cells by one order of magnitude. The resulting system functions fairly well. $100\% \pm 0\%$ of the infected cells are killed and $7.03\% \pm 2.12\%$ of the uninfected cells are killed (Figure 3A/E). Unfortunately this percentage of uninfected cell death is too high and would negatively effect patient health. Further decreasing the number of zincA and zincB fusion proteins resulted in the survival of a few infected cells, presumably because the Specificity Device proteins were degraded before finding their target DNA sequence (data not shown). Thus an alternative method for decreasing non-specific cell death was needed.

Addition of a Titration device. In order to decrease uninfected cell death, we decided to increase the degradation rates of PhoR, *PhoB_{active}*, and BAX mRNA, thereby titrating the concentrations of these species in the host cell. We hypothesized that increasing these degradation rates would decrease

unintended production of BAX by eliminating background activation due to non-specific release of PhoR. In the event that PhoR is released frequently in response to target DNA binding by the Specificity Device, the concentrations of PhoR, *PhoB_{active}*, and BAX mRNA will increase quickly and overcome the increased degradation, resulting in host cell death.

Unfortunately uninfected cell death was not significantly altered by increasing the degradation rates of each of these species 100-fold (healthy cells killed: $6.82\% \pm 2.59\%$ Figure 3C/E). The increased degradation rates also resulted in the survival of some infected cells (infected cells killed $\%99.2 \pm \%0.30$) because the activated species (PhoR, *PhoB_{active}*, and BAX mRNA) were degraded too quickly. Thus signal propagation from the Specificity Device to the Kill Device was disturbed. Consequently, we did not increase the degradation rates of these proteins/mRNAs further and concluded that this method of titration is ineffective.

Decreasing the Accidental Release of PhoR allows the system to function effectively. Another way to decrease unintended cell death is to decrease the frequency of unintended

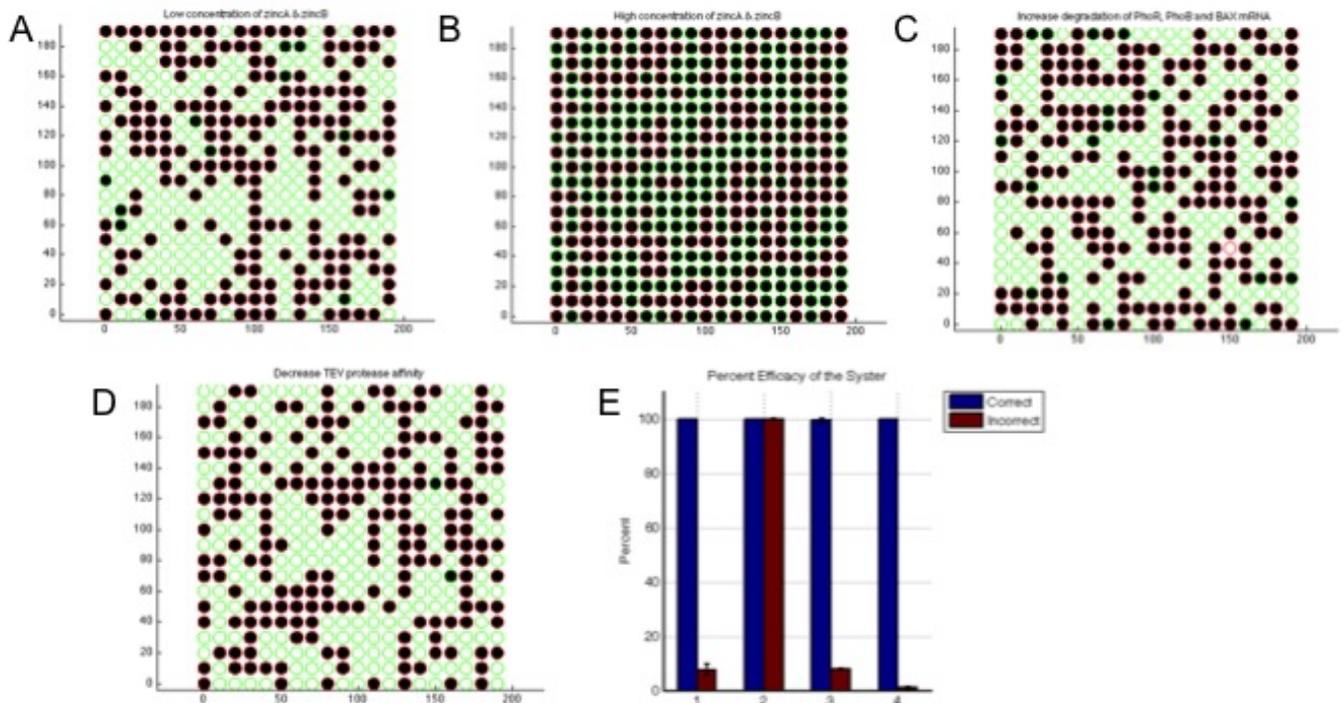


Fig. 3: Efficacies of the synthetic system with various parameter values. **A - D** Plots of four hundred simulated host cells for various permutations of the synthetic system. Each cell is represented as a circle, a green outline indicates a healthy cell, a red outline indicates an infected cell. Black fill indicates cell death via apoptosis and white fill indicates cell survival. When the synthetic system is functioning correctly, the cells will be either black with a red border or white with a green border. Any other permutation indicates that the synthetic system was not capable of correctly diagnosing and treating the host cell. **A** Virus capsid delivers ten zincA and zincB fusion proteins to each cell. $100\% \pm 0\%$ of the infected cells are killed and $7.03\% \pm 2.12\%$ of the uninfected cells are killed. This result is also shown in E1. **B** Virus capsid delivers one hundred zincA and zincB fusion proteins to each cell. $100\% \pm 0\%$ of the infected cells are killed, but unfortunately $100\% \pm 0.26\%$ of the healthy cells are also killed. This result is also shown in E2. **C** Degradation rates of PhoR, *PhoB_{active}*, and BAX mRNA are increased 1000X. $\%99.2 \pm \%0.30$ of the infected cells are killed and $6.82\% \pm 2.59\%$ of the healthy cells are killed. This result is also shown in E3. **D** The TEV protease binding affinity is decreased ten fold. $\%100 \pm \%0$ of the infected cells are killed and only $0.65\% \pm 0.50\%$ of the uninfected cells are killed. **E** Bar graph showing the percent of healthy and infected cells killed by our synthetic system. "Correct" indicates the percent of infected cells killed. "Incorrect" indicates the percent of uninfected cells killed. Each simulation as described above (A-D) is numbered 1-4.

zincA/zincB interactions that result in TEV protease cleavage and release of PhoR. Although it would be very difficult to engineer a system that kept zincA and zincB from encountering each other in the nucleus, it is possible to decrease the affinity of the TEV protease for the zincB cleavage site by mutating the amino acid sequence away from the consensus TEV-cleavage sequence. As a result, short non-specific interactions of zincA and zincB would be less likely to result in TEV cleavage/release of PhoR. When the rate constant governing non-specific PhoR release is decreased ten fold, the synthetic system functions significantly better than the original parameter set. Uninfected cell death decreases to $0.65\% \pm 0.50\%$ while the killing of infected cells remains at $100\% \pm 0\%$ (Figure 3D/E).

Discussion

The goal of our designed system is to reliably kill 100% of infected cells and 0% of uninfected cells. Killing all infected cells is essential because the survival of one HIV infected host cell will result in the persistence of infection, rendering our synthetic system unable to eradicate the virus. Additionally, killing uninfected cells is undesirable because the patient may become severely immunocompromised exacerbating the problem. To achieve our goal, we designed a system that is expected to have switch-like behavior in response to encountering HIV DNA and optimized the parameters of the system using stochastic modeling.

The system incorporates numerous Synthetic Biology design principles. First, the localization of the Specificity Devices and the subsequent release of PhoR is an AND gate that depends on both target DNA sequences being present. Second, the PhoR-PhoB-Pho Box activation sequence is a cascade that is designed to increase the switch-like behavior of the system. Buffering with PhoR strengthens this cascade: PhoR remains in the cytoplasm where it may continually autophosphorylate and activate other copies of the Activation Device. Third, there is compartmentalization between the Recognition Domain, the Activation Domain and the target plasmid, which is intended to reduce background activation. Fourth, we attempted to incorporate a Titration Device involving protein degradation tags and anti-sense mRNA to reduce background and increase the switch-like behavior of our system. And fifth, the entire synthetic system is modular and applicable to many different types of viral infection.

The initial concentrations of the various device components had the most significant effect on system efficacy. In practice these concentrations would be adjusted by engineering the retroviral delivery vector to package more or less of a given component. This is accomplished by incorporating the components DNA into different sites in the retroviral genome: components delivered in high amounts should be integrated near highly expressed viral genes and components delivered in low amounts should be integrated near lowly expressed genes. The impact of these parameters on the efficacy of our system is substantial. When one hundred copies of each Specificity Device component are delivered, 100% of infected and uninfected cells are killed. However, when 10 copies of each Specificity Device component are delivered, 100% of infected cells and $7.03\% \pm 2.12\%$ of uninfected cells are killed. Further decreasing the delivery amount of the Specificity Device components

decreased the kill rate for infected cells and adjusting other initial concentrations did not significantly impact the model (data not shown).

A shortcoming of our design is that the Titration Device did not improve system functionality. The Titration Device increases the degradation rate of PhoR, PhoBactive and BAX mRNA by adding a degradation tag to the two proteins and inhibiting BAX mRNA translation via antisense microRNA. We were first motivated to develop this device due to the observed high rate of uninfected cell death ($7.03\% \pm 2.12\%$). However, significantly increasing each of these degradation rates only marginally improved unintended cell death ($6.82\% \pm 2.59\%$). This led us to modulate the TEV protease binding affinity, which decreased unintended cell death to $0.65\% \pm 0.50\%$ with $100\% \pm 0.0\%$ of infected cells killed.

Modularity is a prominent feature in our system. The zinc finger targeting domains of the Specificity Device are modular and can be changed without disturbing overall function of the synthetic system. Thus other retroviral-based diseases (such as the Flu) can be treated with our synthetic devices. In the system presented, they were designed to target 18 bp of HIV DNA because it is unlikely that a target sequence of this length occurs naturally in the host cell genome (4^{18} combinations of bp). Adjusting the number of targeted bp to impact zinc finger DNA-binding kinetics did not impact the efficacy of the system (data not shown). Therefore the target sequences may be exchanged to satisfy recognize other sequences indicative of different latent viral infections should another disease target be chosen. The only design restrictions on the target DNA sequences are: they must be within a highly conserved viral gene so that all infected cells are likely to be recognized by the two zinc fingers and they must be in close enough proximity to facilitate TEV protease action between zincA and zincB but not so close as to inhibit it. The cooperativity of the minCMV promoter is another modular feature in our system. The number of Pho Boxes can be changed to yield greater cooperativity in transcription from the MinCMV promoter, which increases the hill coefficient of MinCMV activation. Increasing the number of VP16 proteins fused to each PhoB would be expected to have a similar effect. The degree of cooperativity of the minCMV was arbitrarily chosen for our system, but may be later adjusted for further optimization.

Substantial future work remains before this system can be implemented. All of the systems must be built and tested to ensure that they operate within the parameter space defined by this model. Each component would need to be characterized independently and in concert in the host cell context to truly model the system accurately *in silico*.

We designed and implemented *in silico* a Synthetic Biological system intended to kill cells latently infected with HIV. By altering the parameters of the model, we achieved a robust and highly specific treatment. We envision our treatment as part of a comprehensive therapy in concert with HAART to target all facets of the HIV-1 lifecycle. It is novel approaches to treating the disease, such as this, that may someday lead to the worldwide eradication of HIV/AIDS.

ACKNOWLEDGMENTS. ANB was supported by the NSF GRFP. We thank Adam Arkin, Ron Weiss, Noah Davidson for support and careful reading of this manuscript, and the Berkeley and MIT pilot PoSB class of 2010 for helpful discussions.

1. The Millenium Development Goals Report (2010) United Nations Print
2. Wordside AIDS and HIV Statistics. (2010) AIDS and HIV Information from the AIDS Charity AVERT AVERT web, Dec 2010
3. Finzi D, Blankson J, Siliciano JD, Margolick JB, Chadwick K (1999) Latent infection of CD4 T cells provides a mechanism for lifelong persistence of HIV-1, even in patients on effective combination therapy *Nat Med* 5: 512-517
4. Bowman MC, Archin NM, Margolis DM (2009) Pharmaceutical Approaches to Eradication of Persistent HIV Infection *Expert Reviews in Molecular Medicine* 11
5. Burnett JC, Lim K, Calafi A, Rossi JJ, Schaffer DV, Arkin AP (2010) Combinatorial Latency Reactivation for HIV-1 Subtypes and Variants *Journal of Virology* 84.12: 5958-5974
6. Voelkel C, Galla M, Maetzig T, Warlich E, Kuehle J, Zychlinksi D, Bode J, Cantz T, Schambach A, Baum C (2010) Protein Transduction from Retroviral Gag Precursors *PNAS* 107: 7805-7810
7. Tozser J, Tropea JE, Cherry S, Bagossi P, Copeland TD, Wlodawer A, Waugh DS (2005) Comparison of the Substrate Specificity of Two Potyvirus Proteases *FEBS Journal* 272: 514-523
8. Van Dien SJ, Keasling JD (1998) A Dynamic Model of the Escherichia Coli Phosphate-Starvation Response *Journal of Theoretical Biology* 190: 27-49
9. Antunes MS, Morey KJ, Tewari-Singh N, Bowen TA, Smith J, Webb CT, Hellinga HW, Medford JI (2009) Engineering Key Components in a Synthetic Eukaryotic Signal Transduction Pathway *Molecular Systems Biology* 5
10. Schlabach MR, Hu JK, Li M, Elledge SJ (2010) Synthetic Design of Strong Promoters *PNAS* 17: 2538-2543
11. Murphy KM, Ranganathan V, Farnsworth ML, Kavallaris M, Lock RB (2000) Bcl-2 Inhibits Bax Translocation from Cytosol to Mitochondria during Drug-induced Apoptosis of Human Tumor Cells *Nature* 7: 102-111

A biochemical comparator

Tristan de Rond *

*Chemical Biology Graduate Program and Department of Chemistry, University of California, Berkeley

Principles of Synthetic Biology

In the quest to be able to program biological cells to the extent that we can currently program computers, the synthetic biology community has already designed and implemented genetic and biochemical circuits that can perform basic computational functions such as boolean logic and storing a value in a stable state. A circuit that can compare two arbitrary inputs and return their qualitative or quantitative difference has as of yet not been reported, despite this function being an essential component in a complete set of basic computational operation. In this manuscript we present such a circuit and model its behavior deterministically and stochastically. In general, the simulations confirm our hypotheses about the circuit, and show that it is robust. We were surprised by a dependence of the output on the peak width of the input, a phenomenon currently ascribed to a flawed stochastic simulation algorithm. Now that a reliable biochemical circuit to perform signal comparisons *in silico* has been designed, we will need to implement the necessary genes into a model organism to test it *in vivo*. If this succeeds, the door is wide open for sophisticated algorithms that take advantage of subtraction or comparison (e.g. sorting, minimization, game theoretic analysis) to be implemented in a biological context.

Synthetic biology | Comparator | Enzyme catalysis | Difference | Deterministic simulation | Stochastic simulation |

Introduction

The field of synthetic biology has high hopes of one day being able to process any logic in the context of a biological cell, just like we can perform any thinkable logic calculation using electronic logic gates. Progress towards this goal is well underway. Proof-of-principle examples of cellular boolean logic [1, 5] as well as more complex logical functions with hysteresis [4] have been demonstrated.

An important logical function often employed in electrical engineering / computer engineering is the ability to compare two inputs. Computer programs might branch differently depending on the qualitative outcome of a comparison, or could use the result of a subtraction between two numbers to perform complex calculations.

In biology, the comparison of two inputs (in the quantitative case: the difference between two inputs) can be utilized for applications such as engineering a cell to chemotax based on which polar end of the cell receives a higher concentration of a chemoattractant, or making a decision depending on the differential availability of food sources.

In this paper we propose a biochemical network that can perform a comparison between two inputs in the form of PoPS or mRNA molecules, and output a signal depending on the result of the comparison.

Methods

Model Formulation. An example of the exact kind of biological behavior that we are aiming for is laid out in figure 5. The specification is *buffered*, in the sense that the Inputs A and B are 'tallied' until Input C is activated, after which the comparison calculation is performed and the results are Outputs A and B. If Input A was larger (that is, the area under the curve was larger) than Input B, Output A will be on and Output B will be off. The signal intensity of Output B will be a function of the difference in intensities between Inputs

A and B. If Input B was larger than Input A, then Output B will be on.

It should be feasible to implement both specifications using small molecule biochemistry. Because the signal is propagated in a transcription- and translation semi-independent manner it should be relatively low-latency, which is a desir-

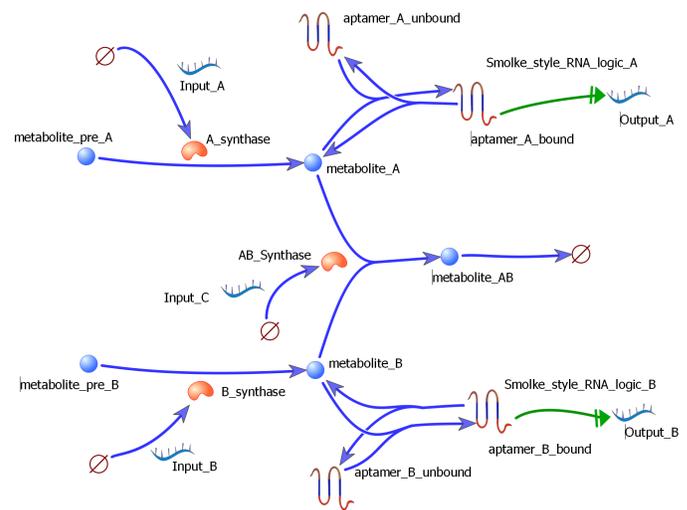


Fig. 1: A schematic of the biochemical circuit that implements that comparator. Drawn in TinkerCell [2]

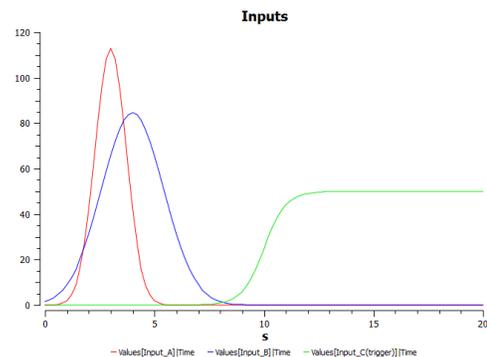


Fig. 2: An example of the circuit in action, simulated deterministically in COPASI. Shown here are the inputs.

Creative Commons License:

CC-BY-NC-SA 3.0

<http://creativecommons.org/licenses/by-nc-sa/3.0/>

¹ read: the input with the largest area under the curve, I will keep on using the shorthand 'largest input' to mean this throughout the manuscript

able feature in synthetic biology circuits. The conceptual implementations of these two specifications are laid out in figure . Both metabolites A and B are orthogonal to central carbon metabolism, and hence can function as a buffer for the input. Letting metabolites A and B react in a 1:1 stoichiometry will deplete the lowest-abundance metabolite, after which only the higher-abundance metabolite is around to activate the signal.

To simplify the analysis, I set the concentrations of pre-A and pre-B to be very high. This way, the enzymes A and B synthase are always saturated, and hence the rate of buildup of metabolites A and B is first-order with respect to its synthase, which should give the required behavior.

Simulations. COPASI [3] was used for all simulations. Some data analysis was performed using Microsoft Excel 2010. As mentioned in the Results section, both deterministic and stochastic simulations were performed. Stochastic simulations employed the tau-leaping algorithm, despite knowledge of inherent flaws in this algorithm, because it was the only simulation method that did not crash COPASI during the simulations. Everything was modeled using mass-action kinetics except for small-molecule turnover rates of enzymes, which were modeled using Henri-Michaelis-Menten kinetics and ping-pong-uni-bi kinetics. Input pulses were generated using a gaussian probability density function for inputs A and B (the area and standard deviation, or width, of which we will refer to again later in the manuscript), and a logistic function for input C.

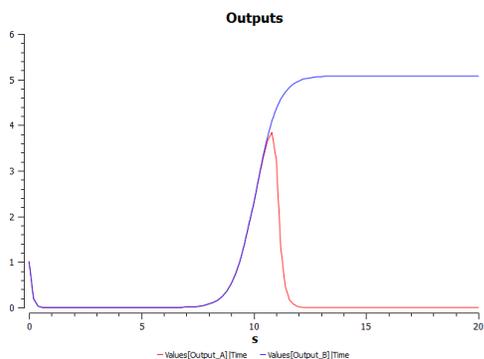


Fig. 3: An example of the circuit in action, simulated deterministically in COPASI. Shown here are the outputs.

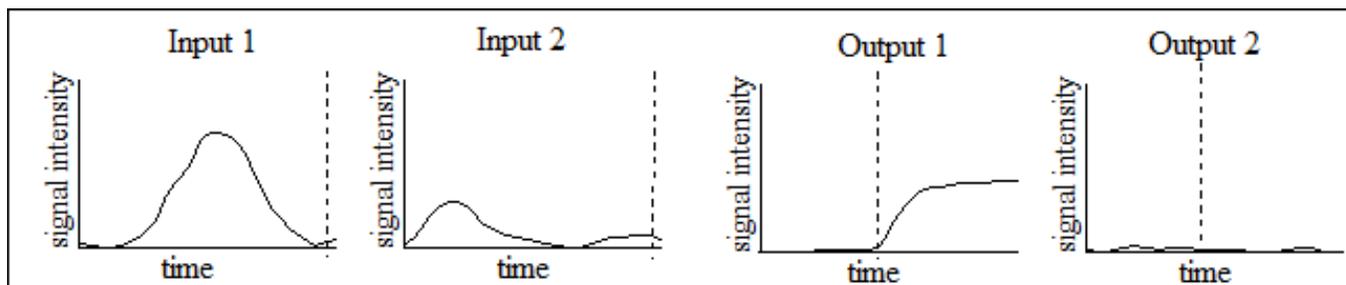


Fig. 5: An example of the kind of behavior expected from a buffered biological comparator. During the time before the dashed line, the system is 'recording' input levels 1 and 2. Then after input 3 gets activated (this happens at the dashed line, sometime in the future we could envision input 3 being driver by a 'clock' circuit for instance), whichever one of the inputs had the highest level throughout the data collection time (in these graphs, that corresponds to the areas under the input curves), its respective output will emit a constant signal (that is, until input 3 is shut off) proportional in intensity to the excess the larger input had compared to the smaller input.

Results

Deterministic simulations. Initial deterministic simulations of the biochemical circuit were successful. The largest input signal¹ had a large steady-state output signal, while the other output signal was near-zero (figures and).

To determine the relationship between the quantitative excess between the inputs A and B and their respective outputs, we performed a number of parameter scans. We kept the area of input A equal to 100 while challenging it with B input areas ranging from 0 to 200. As depicted in figure , the dissociation constant between aptamers A and B and their respective metabolites strongly influenced the response curve. Interestingly, we achieved a somewhat switch-like behavior at low dissociation constants, which is useful if one wants to implement a cellular analog of the computational 'greater than' operator. It is also interesting to note that this behavior appeared despite the lack of co-operativity in metabolite-aptamer binding (I used simple 1-1 mass action kinetics).

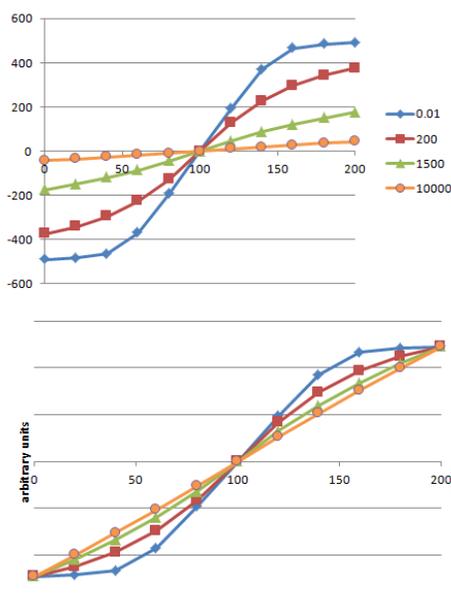


Fig. 4: An example of the circuit in action, simulated deterministically in COPASI. On top, the inputs, on the bottom, the outputs

Stochastic simulations. When the full system was to be simulated stochastically, all simulation methods crashed COPASI. The pathway was apparently too complex. Figuring that we already understood the effect of the aptamer given our previous experiment, we decided to complete stochastic simulations by eliminating the aptamer binding and unbinding steps, and directly monitoring the particle numbers of metabolites A and

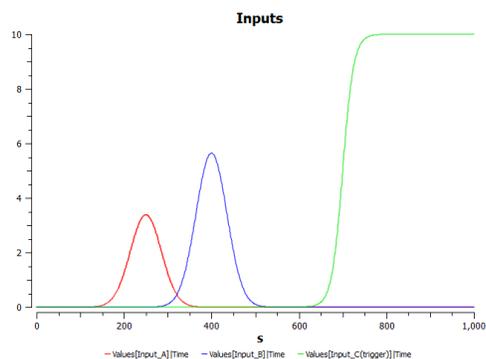


Fig. 6: An example of the circuit in action, simulated stochastically in COPASI. This figure is a counterpart to its output

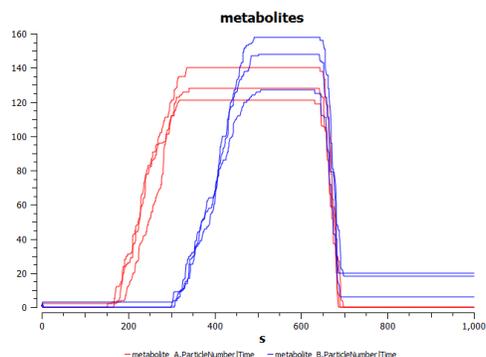


Fig. 7: An example of the circuit in action, simulated stochastically in COPASI. Shown are three independent runs. While not as predictable as the deterministic runs, qualitatively, they still give the correct result: input B was higher, and hence output B is nonzero

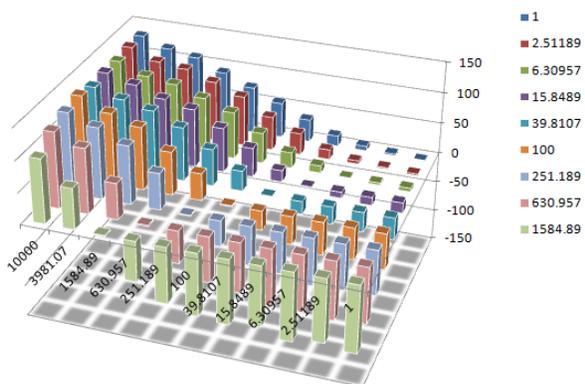


Fig. 8: The mean difference between the steady state levels of metabolites A and B depending on the peak areas of inputs A (across) and B (into the plane of the paper), over 25 trials. Input peak width was kept constant at 25.

B. Even after making this simplification, all simulation algorithms crashed, except for the tau-leaping algorithm. While realizing that this simulation algorithm has flaws, it was the only method that gave us any simulations at all. One successful example of a simulation is given in figures and .

To get a feel for how the circuit behaves stochastically over many trials with any different parameters, we did parameter scans over the input peak areas and widths. This set of simulations took about 36 hours to complete and is summarized in the following paragraphs and figures.

Figure shows the mean difference between outputs A and B depending on the inputs. The results here are as expected: If an input is larger than the other, the corresponding metabolite will end up in excess. Figures and show the standard deviation and the standard deviation divided by the mean, respectively, for these same trials. The standard deviation goes up as the inputs get larger, which is to be expected: larger quantities of metabolites are processed and hence there is more opportunity for error. Overall, the absolute standard deviations show nothing unexpected. Interestingly, the rela-

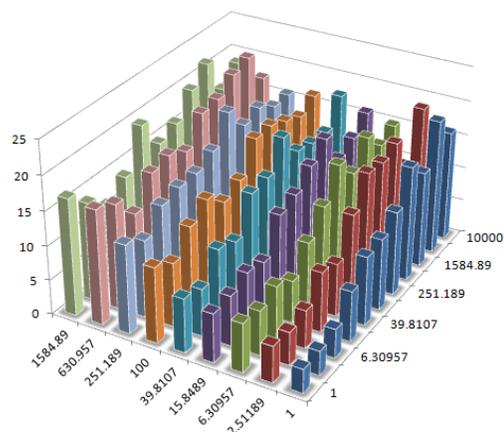


Fig. 9: The average deviation among the differences between the steady state levels of metabolites A and B depending on the peak areas of inputs A (right) and B (left), over 25 trials. Input peak width was kept constant at 25.

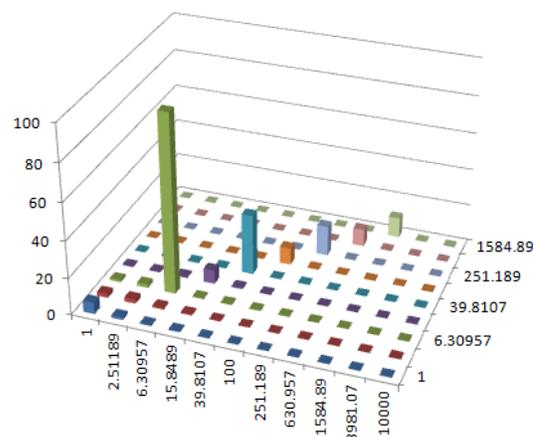


Fig. 10: The average deviation among the differences between the steady state levels of metabolites A and B depending on the peak areas of inputs A (across) and B (into the plane of the paper), over 25 trials. Input peak width was kept constant at 25.

tive standard deviations show that as the input signal strength increases, one actually gets a better signal to noise ratio. Relative errors are obviously larger when the two inputs are nearly the same and are expected to cancel each other out. This was also all expected.

A shock came to us when we compared the dependence of the mean metabolite levels on the peak widths. In theory this should have no effect because the areas of the inputs are still the same. Contrary to our expectations however, we found that the stochastic mean of the metabolite levels has a strong dependence on the input peak widths.

To find out what was causing this aberrant behavior during stochastic simulation, I ran a simulation where inputs A and B have the same area but different widths (figure), deterministically (figure), where it gave the expected behavior of canceling out the two metabolites, and stochastically (figure). It appears that the production of metabolite A continues on at the tail of the input, where the production should be very low. We ascribe this behavior to a flaw in the tau-leaping algorithm rather than a flaw in our design. To find out whether this a reasonable conclusion, it would be beneficial to test out the model using other stochastic simulation methods. Unfortunately, these flat-out crashed COPASI work no matter how much effort was invested into them.

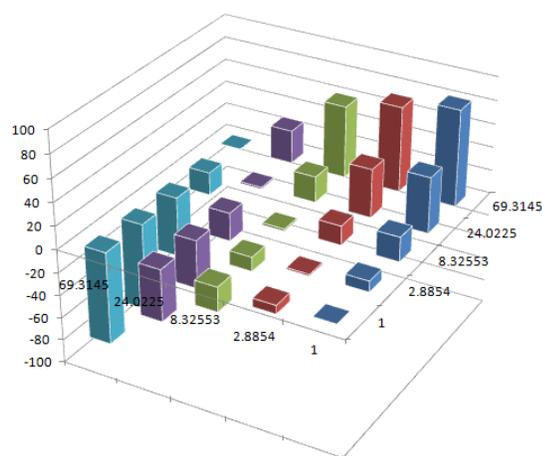


Fig. 11: The mean difference between the steady state levels of metabolites A and B depending on the peak widths of inputs A (right) and B (left), over 25 trials of each of 9 different peak areas

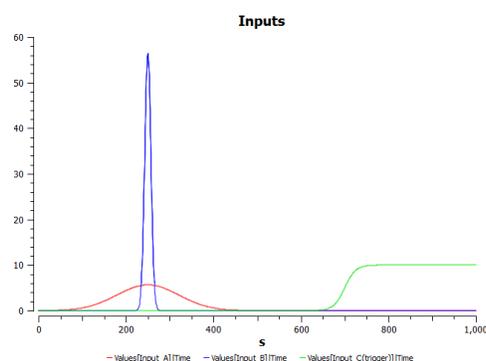


Fig. 12: The inputs of the experimental simulation to determine why peak width affects steady state metabolite levels in stochastic simulations

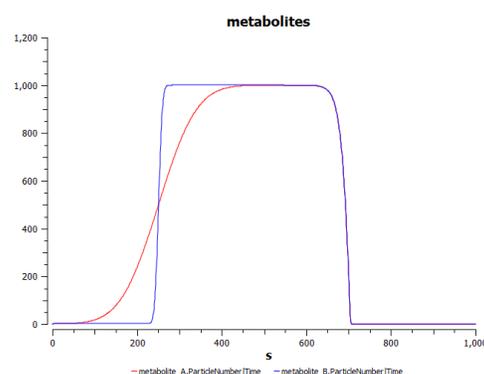


Fig. 13: The deterministic metabolite levels of the experimental simulation to determine why peak width affects steady state metabolite levels in stochastic simulations

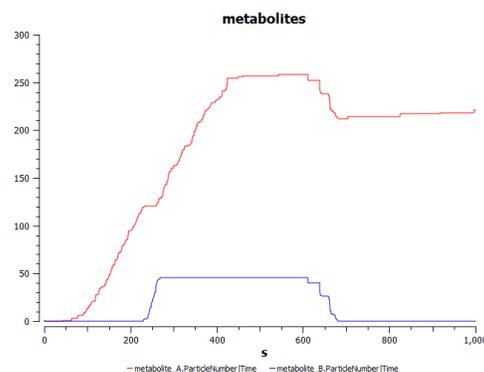


Fig. 14: The deterministic metabolite levels of the experimental simulation to determine why peak width affects steady state metabolite levels in stochastic simulations

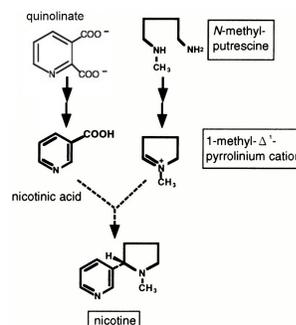


Fig. 15: The nicotine biosynthetic pathway. Adapted from [6]

Discussion

Up until now we have been referring to metabolites 'A' and 'B'. In a true biological implementation these metabolites would most likely be realized by the nicotine biosynthesis pathway (figure). Nicotinic acid would act as 'metabolite A', and the 1-methylpyrrolinium cation would act as 'metabolite B'. The benefit of using this pathway is that the last enzymatic steps of each of the two branches are essentially irreversible under cellular conditions. The output signals will be generated using SELEX-derived aptamers binding to Nicotinic acid and 1-methylpyrrolinium cation, fused to a gene regulation system such as that developed by Christina Smolke [7]. While it has traditionally been difficult to express enzymes of plant origins in microbes (which is what we would like to initially test our biochemical circuit in), We have heard from personal correspondences that new codon optimization and intron excision strategies are making such heterologous expression progressively more feasible.

Despite some uncertainties about the dependence of input peak width on the result of the circuit's computation, we have shown that a biochemical comparator circuit is, in theory, feasible. Next, we will clone the nicotine synthesis pathway genes from *Nicotiana tabacum* and transform them into a host (probably *E. coli*) that has the genes encoding enzymes that perform side reactions on the nicotine synthesis pathway, knocked out. Once we show that this comparator circuit is functional *in vivo*, there are limitless opportunities for the synthetic biology community to implement sophisticated algorithms in cells, taking advantage of the qualitative

or quantitative difference between two signals that this circuit can compute.

References

1. J. Christopher Anderson, Christopher A. Voigt, and Adam P. Arkin. Environmental signal integration by a modular and gate. *Molecular Systems Biology*, 3, 2007.
2. Deepak Chandran, Frank Bergmann, and Herbert Sauro. Tinkercell: modular cad tool for synthetic biology. *Journal of Biological Engineering*, 3:19, 10-29 2009.
3. S. Hoops, S. Sahle, R. Gauges, C. Lee, J. Pahle, N. Simus, M. Singhal, L. Xu, P. Mendes, and U. Kummer. Copasi—a complex pathway simulator. *Bioinformatics (Oxford, England)*, 22(24):3067–3074, Dec 15 2006.
4. Ahmad S. Khalil and James J. Collins. Synthetic biology: applications come of age. *Nature Reviews Genetics*, 11(5):367–379, 2010.
5. K. Rinaudo, L. Bleris, R. Maddamsetti, S. Subramanian, R. Weiss, and Y. Benenson. A universal rna-based logic evaluator that operates in mammalian cells. *Nature biotechnology*, 25(7):795–801, Jul 2007.
6. T. Shoji. Ethylene suppresses jasmonate-induced gene expression in nicotine biosynthesis. *Plant and Cell Physiology*, 41(9):1072–1076, 2000.
7. M. N. Win and C. D. Smolke. A modular and extensible rna-based gene-regulatory platform for engineering cellular function. *Proceedings of the National Academy of Sciences of the United States of America*, 104(36):14283–14288, Sep 4 2007.

A model for self-activating signaling cascades

Amy Y. Chang*, Bryan Williams*

*Department of Bioengineering, University of California, Berkeley

Submitted to Principles of Synthetic Biology, Fall 2010

Synthetic biology is a rapidly-developing field aimed at intelligent design of biological systems exhibiting complex behavioral patterns. The existing toolbox available to synthetic biologists contains a finite but well-defined assortment of genes, promoters, small molecules, and enzymes. Despite the relatively limited scope of available tools, scientists have successfully programmed cells exhibiting sophisticated responses to external inputs. This includes responses driven by logic gates [1] and cells synthesizing an unnatural 21st amino acid [2]. Interest has recently moved past simple cellular response to external input and has focused on understanding and implementing cell-cell communication mechanisms. These mechanisms hold great importance in many fields, as they are responsible for biological phenomena ranging from embryogenesis to neural signal propagation. Scientists have recently designed systems exhibiting robust cell-cell communication, but the vast majority of these systems depend on constant user input to renew cellular responses. Here we present a model for self-activating signaling cascades and show that with careful tuning of rate constants a population of cells can continuously re-activate the cascade even after the initial input has been depleted. This model serves as a theoretical foundation from which scientists can begin undertaking the difficult but promising task of implementing a self-renewing signaling cascade in homogeneous cell populations.

Cell-Cell Signaling | Fluorescent Proteins | COPASI

Abbreviations: AHL, acyl-homoserine lactone; BFP, blue fluorescent protein

Introduction

In recent years, scientists have successfully harnessed genetic and molecular tools to create synthetic biology systems capable of spontaneous pattern formation upon external initiation. This requires implementation of cell-cell communication systems, achieved through quorum sensing [3] or introduction of freely-diffusing signaling molecules [4] that promote or inhibit signal propagation. By manipulating molecular induction and repression kinetics, scientists have also succeeded in producing cells exhibiting a band-pass response to external signal [5]. New breakthroughs have produced novel and varied mechanisms of cell-cell signal propagation, but many of these systems produce static output and cannot re-initiate the signaling cascade.

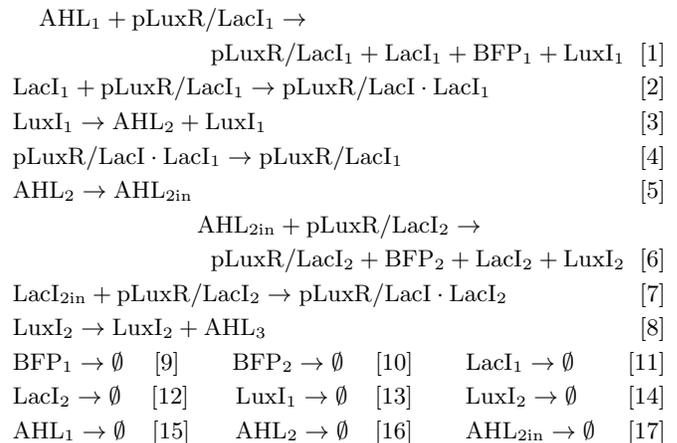
In this study we aim to show that, with properly-designed genetic circuitry, an initializing input suffices to induce a self-propagating cascade in a cyclic arrangement of cell colonies. When a cell receives an input, it turns "on," with a fluorescent reporter molecule signaling the "on" state. The cell then initiates production of the signaling molecule while simultaneously repressing production of both the fluorescent reporter and further signaling molecule production. By modulating the kinetics of activation and repression along with the degradation rates of each molecule, we have modeled a genetic circuit

that produces a transient "on" state and propagates the signal forward before turning itself "off." When cell colonies are arranged in a ring, unidirectional signal propagation is produced by placing a physical barrier adjacent to the initialized colony. As signal propagation continues along the ring, the barrier is removed, allowing the Nth colony to re-initialize the first colony. The colonies can then continue propagating the signal without any additional user input.

Methods

Model Formulation. Our proposed genetic circuit is initialized via direct injection of acyl-homoserine lactone (AHL) into the first colony. AHL is bound by LuxR, an AHL-sensitive transcriptional regulator that interacts with a LuxR promoter (pLuxR) and induces production of three molecules: (1) blue fluorescent protein (BFP) serving here as a reporter molecule, (2) a LuxI enzyme catalyzing synthesis of AHL and (3) a LacI molecule that represses a LacI promoter (pLacI) found downstream of pLuxR and upstream of the BFP, LuxI and LacI genes (Figure 1). For simplicity, this placement of pLacI downstream of pLuxR and upstream of the genes is referred to here as a single promoter entity, pLuxR/LacI. While BFP signals the "on" state, LuxI produces AHL, which is secreted from the cell before diffusing to neighboring colonies. Meanwhile, LacI continues to bind pLuxR/LacI, repressing further production of the gene products, which decay at a steady rate.

The initial cellular responses are modeled with the following reactions:



Creative Commons License:

CC-BY-NC-SA 3.0

<http://creativecommons.org/licenses/by-nc-sa/3.0/>

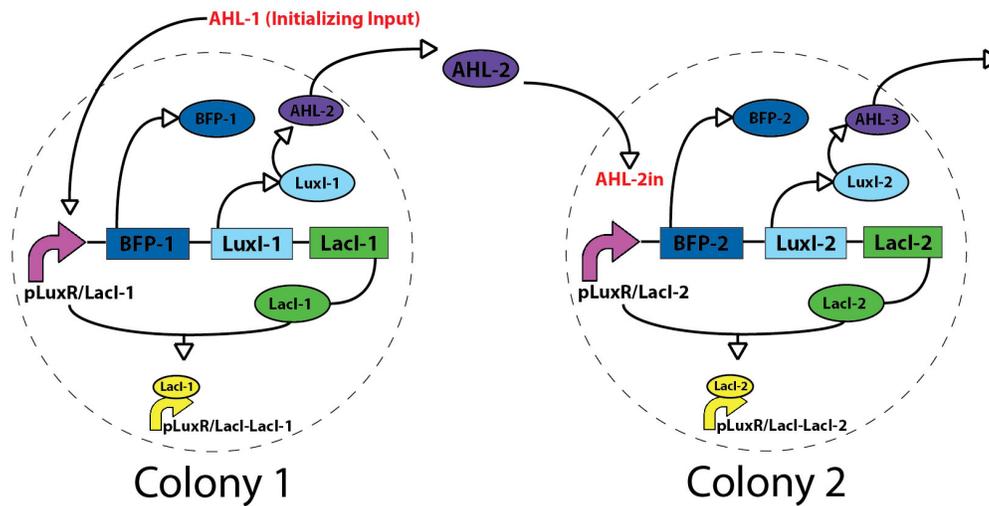


Fig. 1: Schematic of the genetic circuit. AHL is directly injected into Colony 1 as an initializing input, inducing AHL production and diffusion into Colony 2, beginning a signaling cascade. AHL production from Colony 1 is silenced by LacI repressor, characterized by slow unbinding and rapid decay.

Simulations. Simulations were performed in COPASI [6] using irreversible mass-action kinetics. The system was solved using the deterministic modeling approach and transient molecular concentrations output as a function of model time.

Results

Dynamic activation and repression. When an initializing input (AHL-1) is injected into the cell, the concentration of this input decreases rapidly as it complexes with LuxR and initiates transcriptions at pLuxR/LacI and production of BFP (BFP-1), LuxI (LuxI-1) and LacI (LacI-1) (Figure 2A). Fluorescent signal increases rapidly (BFP-1 curve) and gradually decreases as BFP is degraded. Full decay occurs more than 20 seconds after initialization, allowing sufficient time to observe the cell in its "on" state. A drastic increase in LacI levels is produced immediately following initialization, but these levels fall as LacI spontaneously degrades and, more importantly, binds to pLuxR/LacI and represses further transcription. As a result, the concentration of non-repressed promoter (pLuxR/LacI-1) decreases rapidly as the level of the promoter/LacI complex (pLuxR/LacI-LacI1) increases until it reaches steady-state. We initially modeled LacI binding to the promoter as an irreversible reaction, so once $[LacI] = [pLuxR/LacI]_0$, all pLuxR/LacI promoters in the cell become saturated and transcription ceases. In our model, we have carefully controlled the kinetics of repression and translation such that a significant amount of LuxI is produced before complete promoter saturation occurs. The level of LuxI is observed to increase drastically before gradually decreasing. LuxI catalyzes synthesis of AHL, but because it is an enzyme it is not consumed during the reaction. Instead, the gradual decrease in LuxI concentration results from steady LuxI degra-

ation. This slow degradation rate is designed to allow for significant production of AHL (AHL-2) before LuxI levels are depleted. AHL is then exported from the cell and eventually diffuses into neighboring cell colonies (AHL-2in) (Figure 2B). Shortly after uptake of AHL into cells of the second colony, increases in the concentrations of BFP, LuxI and LacI in these cells are observed (Figure 2C). This is a result of AHL complexing with LuxR, which activates the promoter and induces synthesis of these three gene products. The concentration of LacI produced by the second cell (LacI-2) increases after uptake, but then decreases as LacI once again represses the promoter. This is represented by a decrease in the concentration of non-repressed promoter (pLuxR/LacI-2) and a simultaneous increase in the concentration of repressed promoter (pLuxR/LacI-LacI-2) until it reaches a steady-state concentration. Before complete silencing occurs, both BFP (BFP-2) and LuxI (LuxI-2) are produced. The BFP degrades at a constant rate and signals the "on" state for more than 20 seconds, while LuxI will again produce AHL (AHL-3). This AHL will then be secreted from the second cell and transported across the cellular membrane of Colony 3 cells. This cycle will continue until the Nth colony of cells is activated, producing LuxI-N that catalyzes synthesis of AHL-(N+1), which is then secreted by Colony N and taken up by Colony 1, effectively re-initiating the signaling cascade.

Preparing the cell for re-activation. Binding of LacI to pLuxR/LacI was initially modeled as an irreversible reaction. However, if this were a truly irreversible process, diffusion of AHL produced by Colony N into Colony 1 would not re-initiate the cycle, as the LacI produced in the first cycle would irreversibly repress promoter activity. We thus propose a pseudo-irreversible repression model, where the reversion of pLuxR/LacI-LacI (repressed) back to pLuxR/LacI (non-repressed) is characterized by a very slow transition rate.

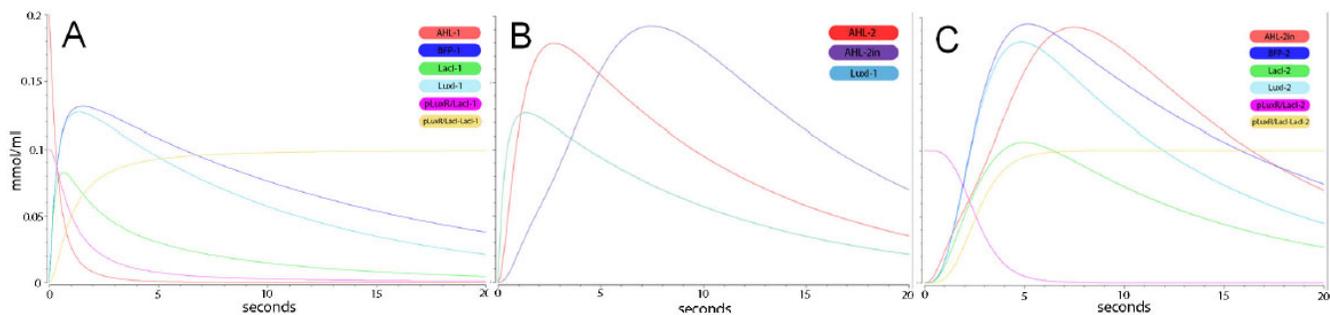


Fig. 2: Molecular concentrations as a function of model time. **A:** Concentrations in Colony 1 cells. Initializing AHL-1 binds to LuxR, activating transcription and increasing levels of BFP-1, LacI-1 and LuxI-1. LuxI-1 produces AHL-2 and degrades, while LacI-1 binds to the promoter and silences gene expression. Fluorescent signal is depleted after approximately 30 seconds. **B:** Extracellular molecular concentrations. LuxI-1 levels increase to produce AHL-2 and then decrease with slow degradation. AHL-2 diffuses into Colony 2 cells producing AHL-2in. **C:** Molecular concentrations in Colony 2 cells. An increase in AHL-2in levels initiates production of BFP-2, LacI-2, and LuxI-2, which produces AHL-3 that diffuses to Colony 3 cells, continuing the signaling cascade.

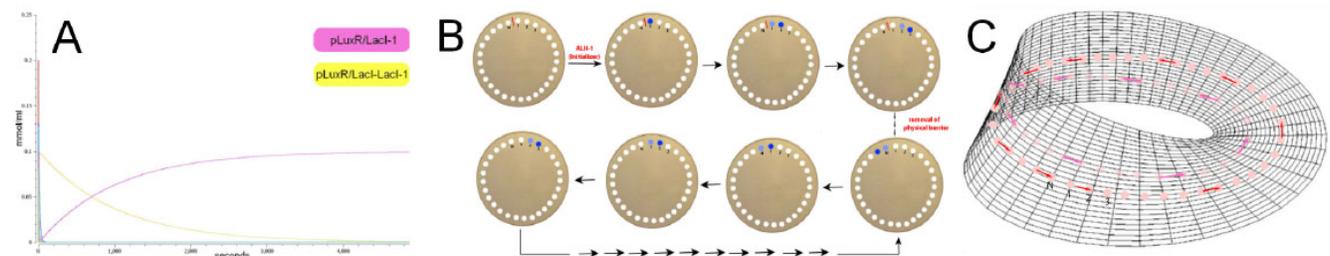


Fig. 3: **A:** Concentrations of pLuxR/LacI (non-repressed) and pLuxR/LacI-LacI (repressed) promoters in Colony 1 cells as a function of time. Population of pLuxR/LacI is depleted within the first 20 seconds after initiation as it is converted into pLuxR/LacI-LacI (see Figure 2A). Slow release of LacI from the promoter followed by quick degradation eventually reproduces the initial state, where all promoters have become "unrepressed" and are available for a second round of signal propagation. **B:** Cell colonies arranged in a ring. Initial physical barrier produces a unidirectional signaling cascade and is removed as the signal propagates around the ring and re-activates the initial colony, renewing the signaling cascade. **C:** Schematic of cell colonies arranged along a Mobius strip engaged in a self-activating signaling cascade. Use of an initial physical barrier allows initializing input at Colony 1 to produce unidirectional signal propagation in the direction of the red arrow. Signal is eventually generated in Colony N and re-activates the cascade in Colony 1. Red and pink arrows represent signal propagation "above" and "below" the strip, respectively.

This allows for the fluorescent reporter signal to decay completely, producing the "off" state and also allows the signal to propagate sufficiently far such that AHL diffusion back to the initial cell becomes negligible. Once in the "off" state, the cell must prepare itself to be re-activated by freeing up the promoter to bind with the AHL-LuxR complex. In a biological system, this is produced by tuning LacI to exhibit high binding affinity for the promoter but near-instantaneous decay upon unbinding. This could be accomplished by introducing a fourth gene product (not modeled here) capable of degrading free LacI. This product would be produced after LacI has fully saturated pLuxR/LacI to prevent premature degradation of LacI. Degradation of LacI as it releases from the promoter would allow the promoter to slowly transition back to the non-repressed state as the signal propagates around the circle. By the time the signal returns to the initial cells, they should have reverted to their original state where $[pLuxR/LacI] = [pLuxR/LacI]_0$ and $[pLuxR/LacI-LacI] = [BFP] = [LuxI] = [LacI] = 0$. Our models show that LuxI production in Colony

2 cells is depleted approximately 60 seconds after initiation at Colony 1. Using a low pLuxR/LacI-LacI to pLuxR/LacI transition rate, we have shown that levels of non-repressed promoter do not become significant until several hundreds of seconds into the simulation (Figure 3A), by which time the signal should have propagated several colonies away and diffusion of newly-produced LuxI into Colony 1 cells becomes negligible.

The principles of this signaling cascade can be applied to a variety of cell colony arrangements to produce self-propagating fluorescent patterns. To begin with, we modeled signal propagation around a single ring of identical cell colonies arranged along the periphery of a petri dish (Figure 3B). Propagation around this ring is designed to take more than 3000 seconds so that by the time the signal returns to the initialized colony, the pLuxR/LacI promoters are no longer repressed and are available for re-activation. An initial physical barrier is placed on one side of the initialized colony and removed once the signal has propagated a sufficient

distance away that the potential for premature re-activation becomes negligible. The physical barrier is removed and according to our model the system of cells should be capable of re-activating the initial colony and repeatedly propagating the signaling cascade around the ring without any further user input. Through careful genetic circuit design and modulation of reaction kinetics, we have successfully modeled a user-initialized, self-sufficient signaling cascade in a cyclical arrangement of identical cell colonies.

Discussion

The initial placement of cell colonies into a simple ring was chosen as a proof-of-principle experiment to show the robustness and feasibility of our genetic circuit. In the future, scientists may look to create more complex 3-dimensional self-activating signaling cascades. One possibility is to arrange agar gel onto a scaffold in the shape of a Mobius strip (Figure 3C). Though the propagation remains one-dimensional in that its location can be defined by a single variable, the strip exists in three spatial dimensions and affords an elegant and intriguing visual display of signal propagation across cell colonies.

1. J C Anderson, C A Voigt, and A P Arkin. Environmental signal integration by a modular AND gate. *Mol Syst Biol*, 3:133, August 2007.
2. R A Mehl, et al. Generation of a bacterium with a 21 amino acid genetic code. *J Am Chem Soc.*, 125-4:935-939, January 2003.
3. L You, R S Cox III, R Weiss, and F H Arnold. Programmed population control by cell-cell communication and regulated killing. *Nature*, 428:868-871, April 2004.

Here we have presented a theoretical model for a user-induced signaling cascade capable of self re-initialization. However, it remains to be seen if the proposed genetic circuits can be successfully implemented in biological systems and if the binding, degradation, and catalytic efficiencies can be tuned to produce the desired characteristics. This requires careful tuning not only of rate constants and binding affinities, but also a thorough understanding of the diffusion of small molecules and enzymes across agar plates and the rates at which these molecules diffuse across cellular membranes. A careful study of these parameters will inform not only the circuit design, but also the spacing of cell colonies and the concentration and duration of initializing input. It is our hope that through future work, we can successfully implement this system in bacterial colonies, providing a robust and tunable platform for scientists to study the fundamental mechanisms governing cell-cell communication in both lower- and higher-order organisms.

ACKNOWLEDGMENTS. We thank Alistair Boettiger for support in formulating our gene circuit and aid with COPASI simulations, Adam Arkin and Ron Weiss for teaching the pilot PoSB class of 2010, and the Berkeley PoSB class for helpful discussion.

4. S Basu, R Mehreja, S Thiberge, M T Chen, and R Weiss. Spatiotemporal control of gene expression with pulse-generation networks. *PNAS*, 101-17:6355-6360, April 2004.
5. S Basu, Y Gerchman, C H Collins, F H Arnold, and R Weiss. A synthetic multicellular system for programmed pattern formation. *Nature*, 434:1130-1134, April 2005.
6. S Hoops, et al. COPASI-a COMplex PATHway Simulator *Bioinformatics*, 22-24:3067-3074, December 2006.

A Cell Division Counter Using DNA Invertase Chains as Memory

Justin Feng *

*Department of Bioengineering, University of California, Berkeley

Submitted to Principles of Synthetic Biology, Fall 2010

The development of a single input counter could provide a useful device for many applications, most notably as a cell division counter. Using multiple chains of sequential DNA Invertase sites, a device is proposed and modeled that is activated using a single input, such as transcription factor E2F which is expressed during cell division. The devices maximum counting limit scales exponentially with the number of parts, allowing it to potentially greatly exceed the capabilities of current biological counters. Using COPASI modeling it is seen that the device does indeed function, but more work is needed to specify rate constants that would provide a fully robust and accurate counter. Despite this, the model provides useful insight into future directions in what is needed to create a single input counter.

Cell-Cell Signaling | Fluorescent Proteins | COPASI

Abbreviations: AHL, acyl-homoserine lactone; BFP, blue fluorescent protein

Introduction

A synthetic cell division counter could have many extremely useful applications within biology, potentially providing useful information for cell aging studies, growth rate measurements, or be used to toggle a kill switch or as memory device within genetic circuits modeled after digital circuits. Despite this, development of a robust, usable device has so far been extremely limited within synthetic biology. To date one of the most complex single input counters only enumerates up to three [1].

The difficulty in creating a scalable, modular cell division counter arises from two main issues. The first is the fact that it is assumed that all cell divisions are roughly equivalent to each other. This means that in order to respond specifically to cell divisions, the counter must respond to a single oscillatory input. The second issue that must be overcome is the limited number of usable parts for genetic circuits. Ideally, a device counting cell divisions would be able to count to large numbers while requiring a small number of parts. In order to accomplish this, the maximum counting capacity must scale exponentially with the number of networked parts. In this paper, a device is proposed and modeled that utilizes DNA Invertase sites as memory bits that are activated during cell division. DNA Invertases use FLP recombinase to invert a segment of DNA which allows for multiple states that are stable when FLP is not present. The inversion is especially useful as it allows for the creation of both writeable and non-writeable states depending on whether the cassette is oriented in the sense or anti-sense direction relative to the promoter. If the promoter were initially in the anti-sense direction, it would then only be able to be activated if there were also a burst of FLP which resulted in the DNA being inverted relative to the initial state. These two different states can be used to store information about previous inputs.

The proposed device uses a scalable design that involves separate chains of sequential invertases, with all invertases on a chain being driven by the same promoter. One potential promoter is the cyclin A promoter, which responds to E2F, a transcription factor expressed during G1 phase of the mammalian cell cycle [?].

For this model, the counter is composed of three chains of three invertases each, totaling nine unique invertases. All of the invertases start in the non-writeable state, and each burst of input (labeled cA in the model) from the cell division causes the production of a FLP for one of the invertases. Starting with all non-writeable states, each burst of input causes successive invertases to be flipped into the writeable state. After the first chain of invertases is completely flipped, a burst of B is made. B is the activator for each of the promoters on the second chain of invertases. Similar to the first chain, the second chain starts in the all non-writeable state, and each burst of B makes one burst of a FLP, causing each invertase on the second chain to flip successively. Along with each burst of FLP on the second chain a reset protein r1 is produced. This reset protein drives the creation of FLP on all invertases on the previous chain, causing it to reset into the initial non-writeable state. At the end of the second chain, protein C is made, which is the activator for all the promoters on the third chain. The invertases on the third chain start in the non-writeable state, and each burst of C produces FLP and r2. Protein r2 drives all of the flippases on the second chain, causing it to reset into the initial non-writeable state.

Once a single chain has been activated enough to reach the end, it activates the next chain and is reset, which allows for counting to a number much greater than the number of parts required to make the circuit. With n chains of m invertases each, this type of circuit is essentially a counter with mn possible states. Alternatively, it could be thought as counting in base m using n digits.

One potential problem is that if the burst of initial input is too long, there is the possibility that more than one invertase will flip during one cell division. In a traditional invertase memory unit, this may happen if the first invertase flips, causing the promoter for the second invertase to become writeable while the input has not fully decayed yet. To remedy this problem, repressor sites were placed upstream of the second and third invertases on each chain. Repressor production is induced along with the first flippase, causing repression of any production of the second or third flippase. This allows for the first invertase site to reach a steady state before any sequential invertase sites are induced. The same mechanism was used between the second and third invertase sites, as well as between the third invertase and protein B.

Methods

The cell division counter was modeled in COPASI using the stochastic (Gibson + Bruck) method [3]. The mechanism of the model was as described previously, except with some

Creative Commons License:

CC-BY-NC-SA 3.0

<http://creativecommons.org/licenses/by-nc-sa/3.0/>

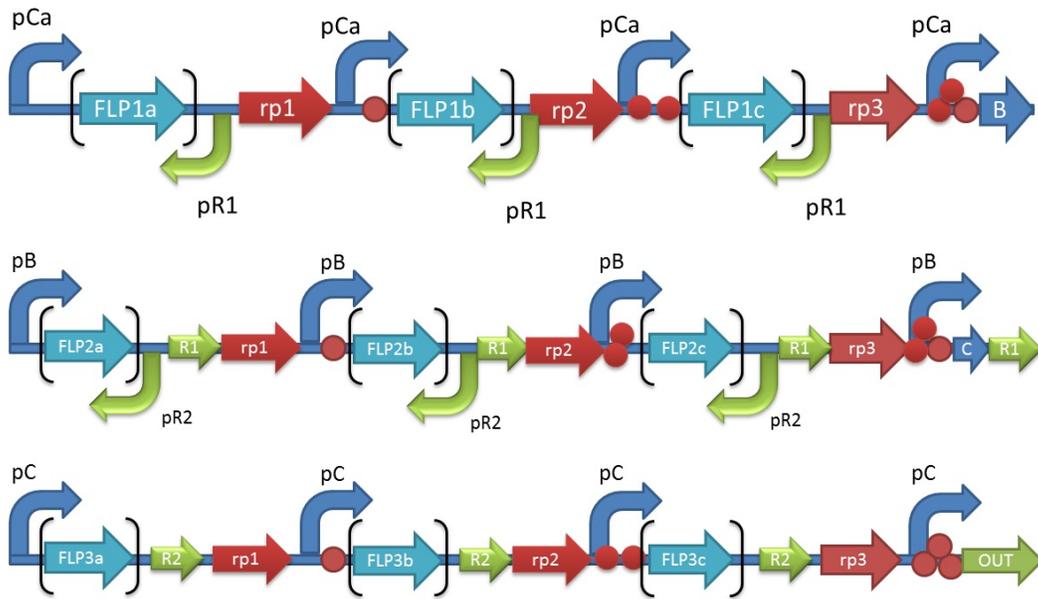


Fig. 1: Figure 1: Cell division counter: The first chain is activated by initial input cA and flippase along with repressor are produced if the cassette is oriented in the sense direction. Flippase production is repressed by any upstream repressors.

simplifying modifications. Transcription and translation were coupled together into one reaction. Also, for the reset function, each chain flipped between the fully inverted and fully non-inverted state without the possibility of intermediate states comprised of a mixture of flipped and non-flipped invertases. Each chain of invertases was modeled as one species, labeled 1a, 2a, and 3a in their completely un-inverted states. Through a reaction with flippase, each chain can be changed from the a state to b, c, or d states. State a represents the completely un-inverted state, state b occurs when the first invertase on a chain has inverted while the last two have not, state c occurs when the first two have inverted, and state d occurs when all three invertases have been flipped. All 9 flippases were modeled to be completely orthogonal. Due to combining transcription and translation, the repressors were modeled to degrade the products of the successive invertase with an extremely high rate constant rather than actually inhibiting the promoter. Production of the flippases were modeled to be switch-like with the promoter being activated only after the input (cA, B, or C) reaches a certain threshold K_m .

Rate Law for cA making flp:

$$\frac{AcA^n}{K_m^n + cA^n} \quad [1]$$

The initial input cA was modeled using an oscillatory repression (repressilator) mechanism, creating periodic bursts of cA. Rate Law for Repressilator:

$$\frac{K_{tr}K_m^n}{K_m^n + S^n} \quad [2]$$

Results

The cell division counter worked when modeled in COPASI with varying levels of success. In some instances the counter mechanism was able to successfully respond to each input burst by having flipping exactly one invertase per burst of cA (Figure 2-3).

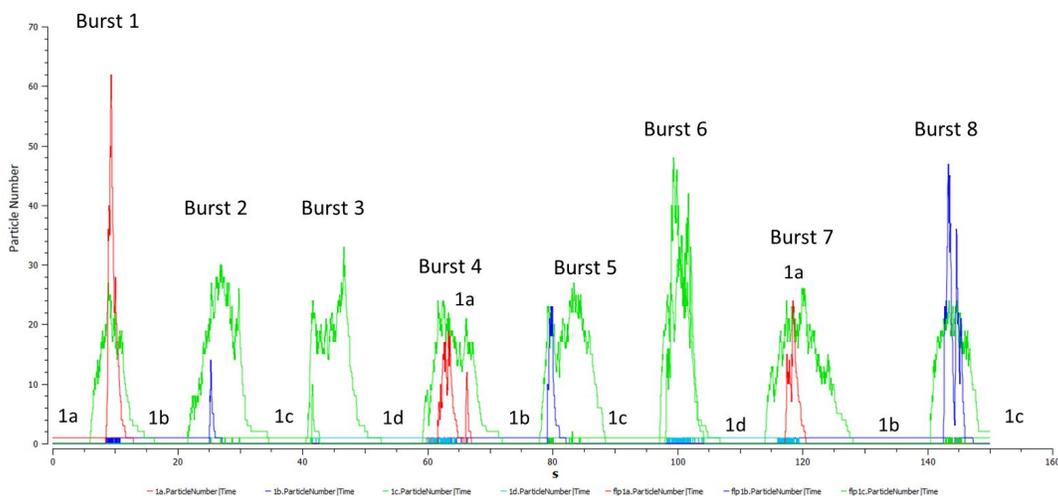


Fig. 2: Figure 2: Cell division counter accurately counting 8 successive bursts of cA. Large green bursts indicate cA red spikes indicate fp1a, blue indicates fp1b, and small green indicates fp1c.

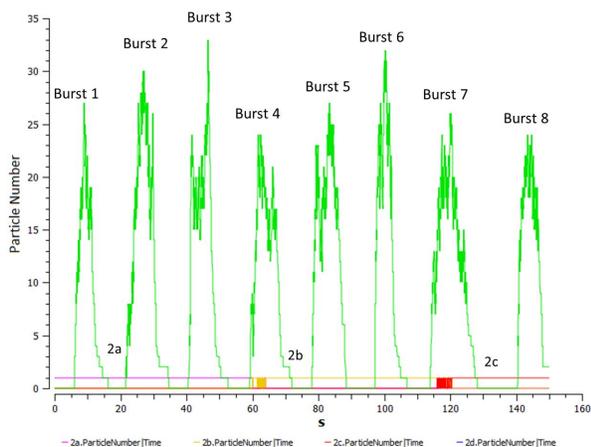


Fig. 3: Figure 3: Cell division counter accurately counting 8 successive bursts of cA. The second chain of invertases is shown, starting in the initial 2a state, flipping to 2b after 4 cycles of cA, then flipping to 2c after 3 more bursts.

In most cases the counter was able to respond to bursts of cA, but either flipped more than one invertase per burst or had the invertase flip an even amount of times before reaching steady state, resulting in no net change (Figure 4). The

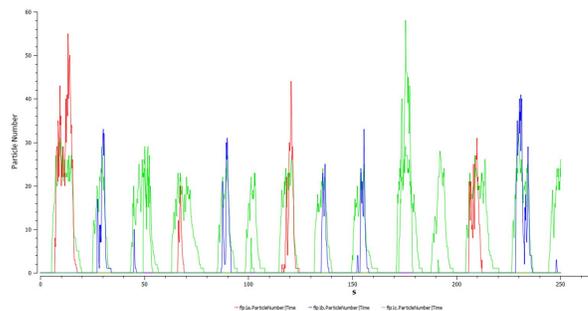


Fig. 4: Figure 4: Cell division counter inaccurately counting bursts of cA. Two successive large spikes of blue show inaccuracy between the second and third invertases on the first chain (states 1b and 1c)

circuit was able to accurately count to 6 approximately 10% of the time.

Discussion

The type of single input counter is one that has great potential. This design is especially scalable the maximum counting limit scales exponentially with the number of parts, while at the same time as the total number of parts is increased the number of unique parts required increases at a sub-linear rate. The number of orthogonal invertases required does increase

linearly with the size of the counter, but the same repressors can be used throughout the system. Along with this, the number of unique activators to communicate between chains only needs to be one less than the number of total chains. The inclusion of repressors in the design also adds robustness to the counting mechanism, reducing false positives and over-counting.

Despite these advantages, the results show that achieving robust, accurate counting is still a major hurdle. Even with the repression system to prevent early activation of successive invertases, the accuracy of the counter is fairly low. Part of this problem was a result of the model as the input bursts were spaced fairly close together due to the functionality of the repressor, and the accuracy of the counter decreases with lower time delays. However, it can be seen that optimizing the rate constants of the flippases and making sure gates are matched are very important to the creation of a robust device. To have the flipping such that each burst of input results in one invertase being flipped, the degradation rate of the flippase and the rate of flipping must both be high. This causes the invertase to rest in the stable state (the non-writeable state), as any time it is in the writeable state while there is input present, more flippase will be produced. The only time flippase will stop being produced is once the invertase cassette has been inverted.

In order to tune a physical counter, the magnitude and time scales input that is being used must be analyzed. The

rate constants of the flippase promoter and production can then be tuned to match the needs of the counter. An improved model could also be used to help predict rate constants that could be used. One main improvement would be to increase the time delay between input bursts. In this model, multiple invertases flipped during a single input cycle when the amount of input reacting with the promoter was small. This produced a small amount of flippase and repressor, and if the repressor decayed before the end of the burst of input, the next flippase was activated. If the delay between input bursts were very large, then the degradation of the repressor could be set to a lower value, which would decrease the occurrences of multiple invertases flipping at once.

With all these improvements left, it is clear that a physical creation of a DNA invertase counter is still far off. While the previously published invertase three-counter was a first step toward invertase based logic, more work is needed to be done to fully characterize the required parts and put them together in a circuit. Despite this, the presented model provides many insights into how a real cell division counter could be made and acts as an initial proof of concept that a physical device could one day be developed

ACKNOWLEDGMENTS. Many thanks to Alistair Boettiger, Professor Adam Arkin, and Professor Ron Weiss for their assistance and guidance in completing this project.

1. Friedland AE, Lu TK, Wang X, et al. Synthetic gene networks that count. *Science (New York, N.Y.)*. 2009;324(5931):1199-202. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/19478183>.
2. Schulze A, Zerfass K, Spitkovsky D, et al. Cell cycle regulation of the cyclin A gene promoter is mediated by a variant E2F site. *Proceedings of the National Academy of*

Sciences of the United States of America. 1995;92(24):11264-8.

3. S Hoops, et al. COPASI-a COMplex PATHway Simulator *Bioinformatics*, 22-24:3067-3074, December 2006.

Engineered Bacteria to Degrade Stent-Associated Arterial Plaques

Dawn Spelke *, Sergey Boyarskiy *

*Department of Bioengineering, University of California, Berkeley

Submitted to Principles of Synthetic Biology, Fall 2010

Atherosclerosis is a debilitating, widespread disease that is often difficult to treat due to the nature of the arterial plaques that cause it. Plaques, lipid masses composed of mostly lipids as well as lipoprotein and calcium, form in the walls of arterial vessels and can develop to occlude them. A common treatment for atherosclerosis is the surgical placement of a stent in the occluded region which alleviates the blockage, but this therapy does not remove the plaque nor have any effect on the nearby vessel region. We propose the design of a bacterial system controlled by engineered genetic circuits that will be introduced into the body following stent placement and will remove plaques through production of a lipase enzyme. The design of the bacteria entails engineering an immune-evading bacterium with genetic loci from pathogenic bacteria that will invade endothelial tissue near the stent region where plaques are formed. A time-delay circuit functions to control lipase production after the bacteria have recognized the stent and penetrated into the endothelial tissue. A threshold-gated circuit controls the amount of lipase that is released by the bacteria before the bacteria undergo apoptosis. Finally, an additional circuit maintains bacterial sensitivity to an antibiotic unless lipase is produced. This circuit is designed to allow for clearance of inactive bacteria following the injection and is stable against mutations. Through modeling, we have shown that these circuits are tunable in both the time-delay and threshold parameters.

Cell-Cell Signaling | Fluorescent Proteins | COPASI

Abbreviations: AHL, acyl-homoserine lactone; BFP, blue fluorescent protein

Introduction

Atherosclerosis is a widespread, deadly disease characterized by the development of plaques in arterial walls. A number of excellent reviews have been written describing the biology of this disease and current therapeutic options (Hansson, 2005; Insull, 2009; Mackman, 2008), so only a brief overview will be presented here. Arterial plaques develop due to the gradual infiltration through and retention of lipids, especially cholesterol and low density lipoprotein (LDL), in arterial walls. Accumulation of lipids under the endothelial layer leads to the recruitment of macrophages that take up the lipids and high levels of inflammation follow. Over time, extracellular lipid pools develop underneath a fibrous cap which is prone to rupture. Continued plaque build-up and platelet and clotting responses due to rupture can cause arterial wall hardening and thickening leading to arterial stenosis, a narrowing of blood vessels. Stenosis interferes with blood flow and can cause heart attack and stroke. One clinical intervention in patients with advanced atherosclerosis is the placement of a stent in the damaged artery. The stent expands, compressing the plaque, and opening up the artery. However, the plaque itself is not treated and the stent acts mechanically so it is only functional in its precise location. Drug-eluting stents can prevent restenosis due to scar tissue formation, but currently there is no stent procedure that prevents further plaque accumulation in the stent region or in the nearby vicinity. Therefore, a therapeutic tool that can degrade plaques around stents would greatly their effectiveness.

Bacteria are potentially an ideal vehicle for delivering therapeutics for a number of reasons. Bacteria can be genetically engineered to perform a variety of functions; sense

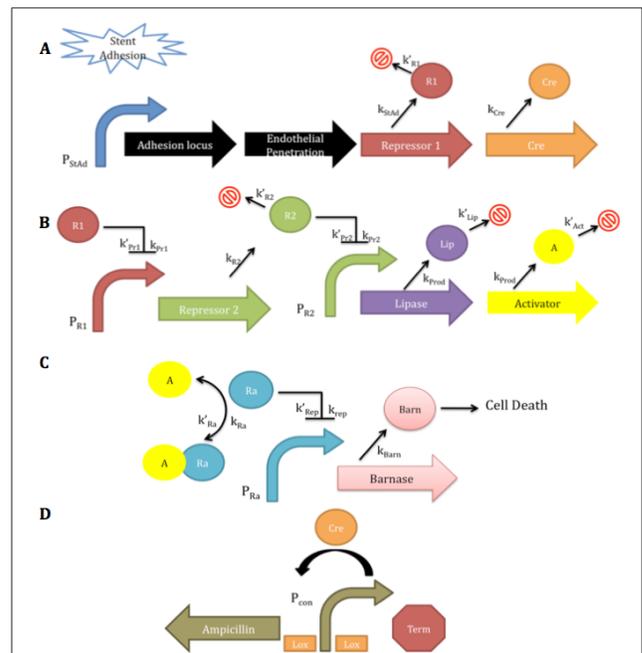


Fig. 1: Schematic of Genetic Circuits. (A) Stent adhesion (blue) results in the translation of multiple gene products from pathogenic bacteria for bacterial adhesion and penetration (black), Repressor1 for the time-delay (red), and Cre (orange). (B) Time-delay for lipase production (purple) is achieved through Repressor1 repression of Repressor2 (green), which removes lipase repression; Activator (yellow) is produced along with lipase (C) Threshold-gating of barnase is accomplished by Activator binding to the constitutively expressed repressor (teal) of barnase to allow barnase production. (D) Activated bacteria are not cleared by ampicillin administration because Cre production flips a constitutive promoter to allow translation of ampicillin resistance gene (tan). Block straight arrows: genes. Block curved arrows: promoters. Circles: gene products. Thin block arrows: activation reactions. Double-headed arrows: reversible reactions. Thin capped lines: repression reactions.

Creative Commons License:

CC-BY-NC-SA 3.0

<http://creativecommons.org/licenses/by-nc-sa/3.0/>

many biological signals; and naturally interact with the body. They also hold the potential to produce protein therapeutics directly in vivo. Previous studies have examined the therapeutic potential of engineered bacteria with some success (Hayashi, 2009), so a bacterial approach to attacking plaques is plausible. Therefore, we have designed a bacterium that will degrade arterial plaques in the vicinity of implanted stents.

Design

Bacterial Choice. Bacteria will be chosen that can survive in the bloodstream and penetrate the endothelial layer. Pathogenic *E. coli* that are able to survive in the bloodstream exist, but they are known to elicit an immune response (Lemichiez, 2010). As general toxic shock and, more specifically, inflammation around the stent region would be highly detrimental to this system, a strain of *E. coli* deficient in inducing an immune response will be used. Bacteria knocked out for the *msbB* gene are an ideal candidate. This strain is known to elicit 1000- 10000 times smaller immune response than wild-type *E. coli*. (Somerville, 1996), so it should be a safe choice.

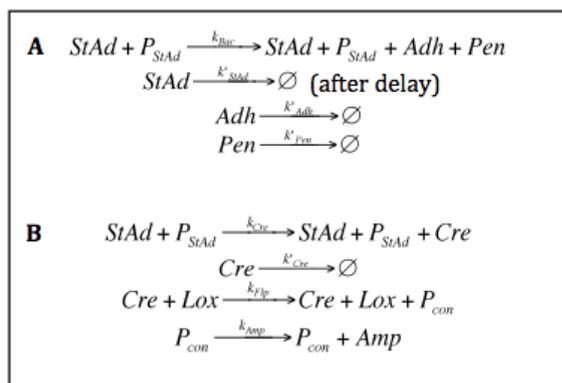
Bacterial Targeting. The bacteria will be targeted to plaques near the stents. Implanted stents will be functionalized with an adhesion molecule that the bacteria will be engineered to recognize. Recognition of the stent by a bacterium in the bloodstream will cause the bacterium to transiently adhere to the stent and be locally constrained to the stent region. Bacterial adhesion and penetration of endothelial layers has been documented, and genes responsible for the phenotype are known (Kim 2002). These genes will be placed downstream of a promoter activated by adhesion to the stent region, so that once bacteria transiently interact with the stent, they will be primed to cross nearby endothelial tissue and reach plaques (Fig. 1a).

Plaque Degradation by Lipase Production .Plaques are largely made up of lipids, so the bacteria will be engineered to express lipases. Ideally, lipoprotein lipase will be expressed as this enzyme hydrolyzes LDLs and has been studied in relation

to atherosclerosis. However, lipoprotein lipase must be glycosylated to function (Mead, 2002; Tsutsumi, 2003), which has not been easily achieved in *E. coli*. So, a glycosylation pathway will be engineered into the *E. coli*, a principle that has had some limited success (Bttcher, 2006). If this is unsuccessful, a bacterial lipase can alternatively be used, many of which have been studied (Schmidt-Dannert, 1998), and screens can be performed to find the most potent lipase for the destruction of arterial plaques. As the specific lipase functionality and activity will need to be determined experimentally, here a generic lipase will be discussed. Lipase production will be triggered by stent adhesion. To allow for bacterial penetration through the endothelial layer before lipase release begins, a time-delay circuit will be implemented between adhesion activation and lipase production (Fig. 1A,B). Expression of cell adhesion and penetration genes as a result of bacterial adhesion and interaction with the stent will be coupled with production of a repressor protein (Fig. 1A). This repressor will in turn repress the production of a secondary repressor. Upon decline in production of the second repressor, it will be removed over time due to its natural degradation, which will then alleviate the repression of downstream lipase production (Fig. 1B) Lipase release will be achieved by engineering the molecule to contain a twin-arginine translocation tag that will allow the folded protein to be secreted out of the bacteria and into the plaque environment (Tullman-Ercek, 2007).

Bacterial Suicide .To prevent excessive lipase activity, regulation of lipase levels will be a necessary control. Regulation will be achieved through threshold-gated expression of barnase, a suicide gene (Meiering, 1992) (Fig. 1C). Barnase production will be under the control of a promoter repressed by a constitutively expressed repressor. Lipase production will be coupled with production of an activator that will bind this repressor resulting in barnase expression. Importantly, binding of the repressor to the barnase promoter will be very tight, so a certain amount of activator will be required for the repression to be lifted. Thus, this circuit will act as a threshold gate in which bacteria die after a specific activator, and corresponding lipase, level is reached.

Clearance of Inactive Bacteria .Bacteria that do not interact with the stent and subsequently penetrate the endothelium and degrade plaques will need to be cleared from the body. Clearance will be achieved using ampicillin, an antibiotic. Bacteria will express an ampicillin resistance gene if they have been activated, so ampicillin administration will only target inactive bacteria. Selective expression of the resistance gene will be achieved using a Cre-Lox system (Fig. 1D). The gene will be placed upstream of a constitutive promoter flanked by loxP sites. Cre will be placed downstream of the stent adhesion-activated promoter described above (Fig. 1A), so upon activation of the bacteria, Cre will be expressed and flip the promoter to turn on the resistance gene. This selection criterion was chosen as it will be robust against mutations since mutations in the flanked promoter or the Cre molecule should not lead to an antibiotic resistance phenotype.



Box 1. Reactions modeling bacterial penetration and antibiotic resistance. (A) Stent adhesion mediated translation of adhesion and penetration genes. (B) Cre-controlled antibiotic resistance.

Methods

Model Creation. All modeling of genetic networks was completed utilizing the COPASI modeling suite (Version 4.6.33) available at www.copasi.org. All concentrations were modeled as particle numbers at constant cell volume. Reactions were modeled using irreversible mass-action kinetics reactions as shown in the results section.

Time-Delay Circuit. Models for the time-delay circuits were analyzed in deterministic mode. All protein production rates except lipase were 5/s. Degradation rates used were 0.1/s. Lipase production rate was 50/s to match up to downstream suicide gene production. To recreate fast binding, repressor binding promoter rate was 1000/s (200 times the protein production rate) with a $K_d=10^{-3}$. Although these values are somewhat arbitrary, their relative magnitudes are typical of natural reaction rates with relatively weak binders (Majka, 2007). All models began with an initial concentration of $R_2=50$ and $StAd=50$. Stent adhesion molecules were allowed to degrade 100s after simulation began to imitate diffusion away from the stent or penetration of the cell into the endothelium. Parameter scans on the degradation values used the values for degradation between 0.01 and 0.5 with 5 regular intervals. All time courses were modeled to 400s. This time was chosen to display the relevant characteristics of the circuit. Promoter concentrations were kept at 1 promoter per cell.

Threshold-Gated Circuit. Models for the threshold-gated circuit were analyzed in a stochastic expression mode. To match the time-delay circuit, lipase production rate was kept at 50/s with 0.1/s degradation for all proteins. Suicide gene production rate was chosen higher to be similar to lipase production rate at 50/s. Individual time courses were conducted over 10s. Parameter scans on activator-repressor binding were conducted by changing the on rate from 100 to 104/s in a logarithmic scale while maintain the off rate at 1/s. Promoter concentrations were kept at 1 promoter per cell.

Results

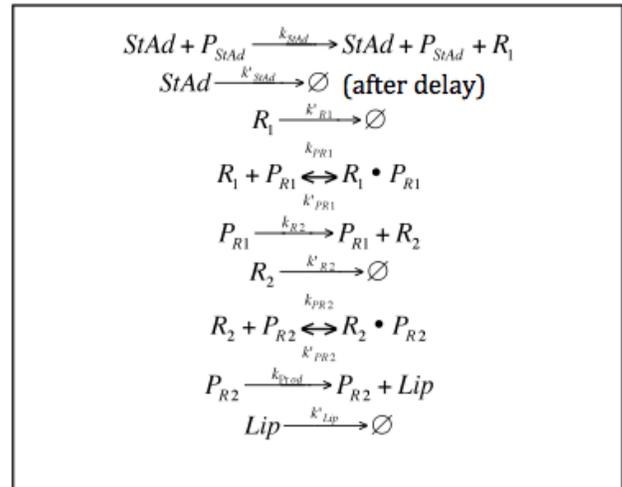
In modeling the system, all of the protein production reactions were simplified to couple transcription and translation of the gene product into a single kinetic parameter. Furthermore, activator binding to promoters was not modeled as it was taken to be instantaneous as these binding interactions occur at much higher rates (for example, see Box 1). In circuits where protein-promoter interactions are crucial to the system (such as the repressor time-delay circuit), these interactions were specifically modeled.

Penetration and Antibiotic Resistance. The function of the circuits described in Fig. 1A and 1D was modeled through the chemical equations show in Box 1. The amount of repressor R1 and Cre protein produced with respect to cell adhesion and penetration can be controlled through engineering ribosome binding sites. Originally the design for switching the antibiotic resistance state utilized the production of an activator protein instead of Cre that would induce production of the antibiotic resistance gene. This design, however, is not optimal because any leaky activator expression could, over time, lead to antibiotic resistance. The Cre system does not suffer from this drawback as a substantial amount of Cre is required to induce flipping of the lox sites (Ringrose, 1998). Furthermore, the Cre system is more robust to mutations as any mutation in the promoter region (increased strength) or in Cre itself will not turn the antibiotic resistance on.

Time-Delay. To maintain proper release of lipase into the plaque-containing endothelium, a bacterial cell must recognize that it has escaped the blood vessel and penetrated the endothelial layer. Unfortunately, feedback mechanisms for the transition from vessel to endothelial tissue are unknown. It is likely that bacteria recognize the concentration of certain

ions or proteins present in the endothelium, but these processes are not characterized in the literature in detail. To circumvent this feedback mechanism, a time-delay architecture will be implemented in the circuit. In the time-delay system, lipase production occurs at some point after the activation of the cell adhesion locus and the genes responsible for bacterial penetration into the endothelial layer. An initial model involved the production of an activator molecule following the stent interaction. This activator in turn activated the production of another activator, and so on for three rounds to activate the production of lipase. In principle this system would create a delay as each activator protein needs time to be produced before the next step in the system can begin. When modeled and solved deterministically, however, it became clear that the final production of lipase was an exponential curve with a rate constant equal to the product of the individual production rates (data not shown). While this would create some time-delay, the circuit did not function in the necessary switch-like manner.

A new network was therefore engineered to create a more switch-like time-delay (Fig. 1B). The topology of this network is similar to other time-delay systems reported in the literature (Weber, 2007). Here, interaction with the stent produces a repressor molecule R1 which in turn represses the production of another repressor R2. Prior to stent adhesion, R2 is constitutively expressed and degraded such that R2 is at steady-state and repressing the production of lipase and downstream products. When stent adhesion occurs, and R1 is expressed, the production of R2 stops and protein degra-



Box 2. Reactions modeling time delay.

$$\begin{aligned}
 \frac{d(R_1)}{dt} &= k_{StAd} \times [StAd] \times [P_{StAd}] - k'_{R1} \times [R_1] + k'_{PR1} [R_1 \cdot P_{R1}] - k_{PR1} \times [R_1] \times [P_{R1}] \\
 \frac{d(R_2)}{dt} &= k_{R2} \times [P_{R1}] - k'_{R2} \times [R_2] + k'_{PR2} [R_2 \cdot P_{R2}] - k_{PR2} \times [R_2] \times [P_{R2}] \\
 \frac{d(Lip)}{dt} &= k_{Lip} \times [P_{R2}] - k'_{Lip} \times [Lip] \\
 \frac{d(R_1 \cdot P_{R1})}{dt} &= k_{PR1} \times [R_1] \times [P_{R1}] - k'_{PR1} \times [R_1 \cdot P_{R1}] \\
 \frac{d(R_2 \cdot P_{R2})}{dt} &= k_{PR2} \times [R_2] \times [P_{R2}] - k'_{PR2} \times [R_2 \cdot P_{R2}] \\
 P_{R1} + R_1 \cdot P_{R1} &= 1; \quad P_{R2} + R_2 \cdot P_{R2} = 1
 \end{aligned}$$

Box 3. Mass action kinetic equations for time delay.

dation machinery slowly removes any R2 left over in the cell. The biochemical equations governing the process are shown below in Box 2 and relevant mass action kinetic equations are shown in Box 3.

To analyze the behavior of the system, parameter scans were performed on key reaction parameters. As expected, variations in the degradation rate of the stent adhesion molecules produce a varied amount of lipase. As stent molecule degradation rate is increased (due to degradation of downstream signaling molecules in the bacteria or diffusion of the bacteria away from the stent), the amount of lipase production is decreased. The time-delay, however, is still the same because the stent adhesion molecule concentration remains constant for the first 100 seconds of the simulation (Fig.

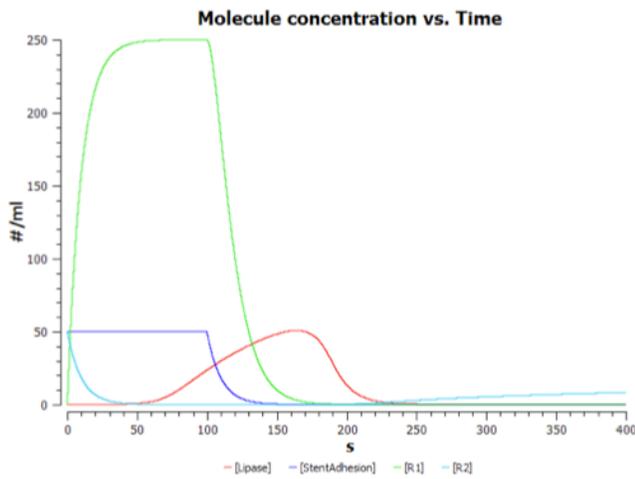


Fig. 2: Time response of molecules in time-delay circuit. Stent adhesion (blue) was modeled to be present for 100s as cells are bound to the stent, after which signal molecules are degraded. Lipase (red) production is turned on after a delay as the concentration of repressor R1 (green) increases and repressor R2 depletes. Kinetic parameters are described in the Methods section.

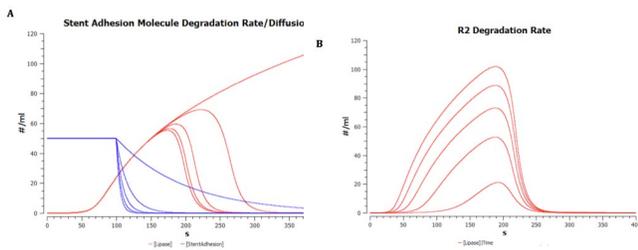


Fig. 3: Parameter scans of key reaction parameters. (A) Variation of the stent adhesion signaling decay. Lipase production rate and concentration increase with decreasing k'_{stAd} . (B) Time-delay of lipase production increases with decreasing degradation rate of the R_2 repressor, k'_{R2} . In both cases the degradation rates are being varied from 0.01 to 0.5 s^{-1} on a linear scale.

3A). Thus overall lipase production rates can be controlled by engineering systems that degrade the stent adhesion signal, or by choosing downstream signaling molecules with varying degradation rates. Lipase production was then modeled as a function of the degradation rate of the R2 repressor (Fig. 3B). As degradation rate increases, the time-delay associated

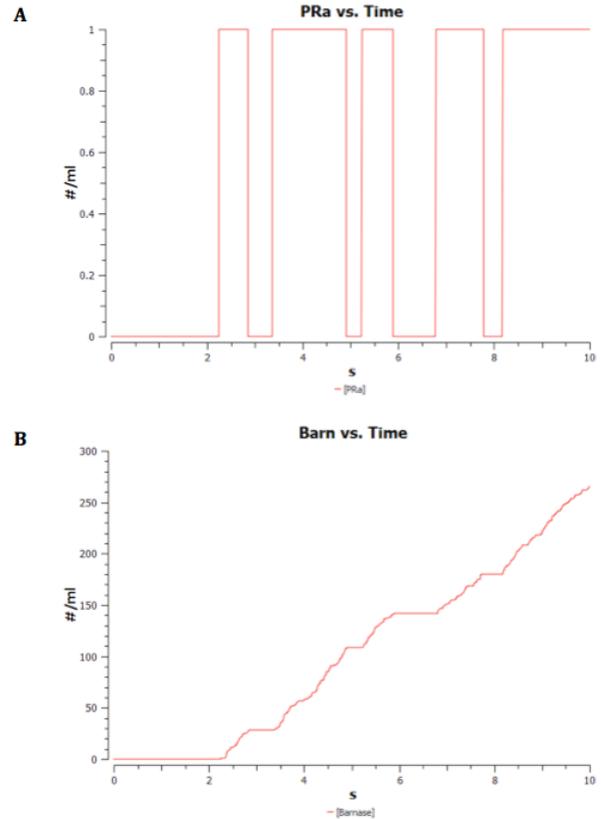


Fig. 4: Barnase production requires unrepressed promoter. (A) Free promoter over time. (B) Barnase production over time. Barnase is only produced when promoter is not being repressed.

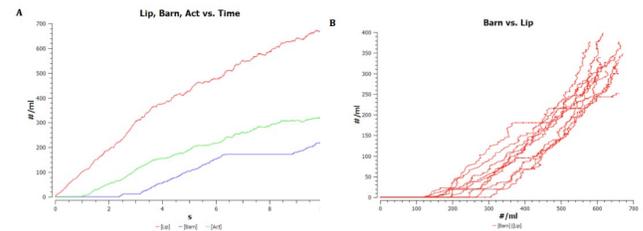


Fig. 5: Threshold-gating of barnase production. (A) Lipase, barnase, and activator over time. Barnase (blue) production is achieved after a threshold of activator molecule (green) is achieved. This threshold is directly proportional to lipase level (red). (B) Multiple simulations of barnase production versus lipase levels. Threshold behavior is visible as barnase production never occurs prior to the threshold value (100/ml). Barnase activation occurs at different lipase levels in each simulation due to stochastic effects.

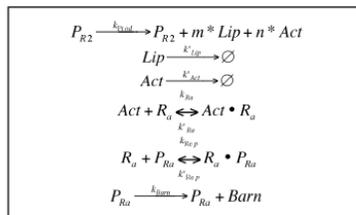
with the circuit decreases and the rate of lipase production is increased. Thus the delay time in the system is tunable by the variation of the degradation rate of the R2 repressor. The degradation rate of R2 could be modified by the introduction of protease recognition sequences or by engineering of the protein to be more resistant to proteases.

The above reactions can be analyzed using mass action kinetics and important equations are shown in Box 5.

$$\begin{aligned}
 \frac{d(Lip)}{dt} &= m \times k_{prod} \times [P_{R2}] - k'_{lip} \times [Lip] \\
 \frac{d(Act)}{dt} &= n \times k_{prod} \times [P_{R2}] - k'_{act} \times [Act] - k_R \times [Act] \times [R_a] + k'_R \times [Act \cdot R_a] \\
 \frac{d(R_a)}{dt} &= -k_{Rep} \times [Act] \times [R_a] + k'_{Rep} \times [Act \cdot R_a] - k_{Rep} \times [R_a] \times [P_{Ra}] + k'_{Rep} \times [R_a \cdot P_{Ra}] \\
 \frac{d(P_{Ra})}{dt} &= -k_{Rep} \times [R_a] \times [P_{Ra}] + k'_{Rep} \times [R_a \cdot P_{Ra}] \\
 \frac{d(Barn)}{dt} &= k_{Barn} \times [P_{Ra}] \\
 P_{Ra} + R_a \cdot P_{Ra} &= 1; (R_a)_{total} = R_a + Act \cdot R_a + R_a \cdot P_{Ra}
 \end{aligned}$$

Box 5. Mass action kinetic equations for threshold-gated bacterial suicide.

Threshold-Gated Bacterial Suicide. To regulate lipase production, bacterial suicide will be induced through threshold-gated barnase expression achieved through the circuit shown in Fig. 1C. The reactions representing this system are shown in Box 4. This component of the system is linked to the time-delay circuit explored above as PR2 is the same promoter upstream of lipase production. PR2 is also upstream of Act production, which plays a crucial role here. As the endpoint of the time-delay circuit is the removal of PR2 repression, this suicide component can be examined separately with PR2 functioning as a constitutively active promoter. This model is valid when PR2 is not being repressed, which is the time window of activation determined above. The following analysis assumes this simplification.



Box 4. Reactions modeling threshold-gated bacterial suicide.

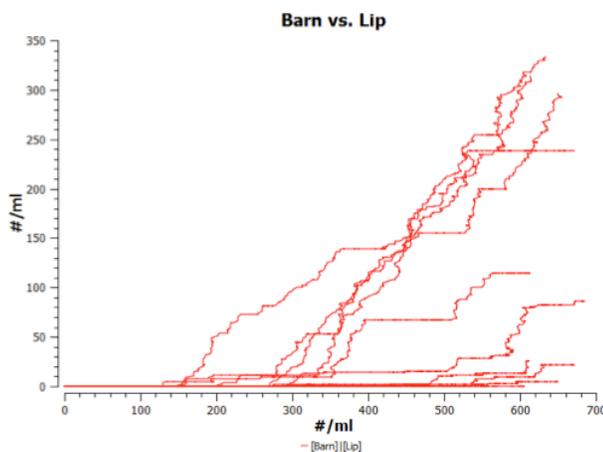


Fig. 6: Parameter scan of activator-repressor binding rate, k_R . As the binding becomes tighter, expression of barnase (y-axis) occurs closer to the threshold point. The variation in thresholding due to k_R is much greater than that of stochastic effects.

When no Lip and Act are being produced, which is the state of the inactive system, the steady state value of PRA is given by:

$$[P_{RA}] = \frac{k'_{Rep} R_a P_{Ra}}{k_{Rep} R_a} \quad [1]$$

If the constitutively expressed repressor tightly binds the promoter, modeled here by the value of k_{Rep} being 1000 times greater than that of k_{Rep} , the value of $[P_{RA}]$ is about 0. Thus at steady-state in inactive bacteria, there is virtually never free promoter and, therefore, no barnase production and no suicide. Furthermore, since there is only one promoter element in the system and since $[P_{RA}]$ in the expression above can take on continuous values, the calculated steady-state parameter is closely related to the probability of a free promoter in the system. Consequently, the probability that a molecule of barnase will be produced in a given time can be calculated and repressor-promoter binding can be engineered to minimize stochastic leaky expression of barnase.

When bacteria are active and Lip and Act production occurs, activator molecules bind repressor molecules. When free repressor disappears or is limited, the promoter loses its repression and barnase is produced. Modeling this system verifies this result: during activation, the promoter stochastically loses repression (Fig. 4A) and during these times barnase is produced (Fig. 4B).

Bacterial suicide is part of the design to ensure that lipase levels are controlled, so suicide gene production should begin after a certain threshold lipase concentration is reached. Lipase, itself, cannot bind the repressor, so the removal of repression of barnase production is directly related to activator levels rather than lipase levels. However, as lipase and activator production are both downstream of the same promoter (PR2), the lipase level is directly proportional to the activator level. Modeling lipase, activator, and barnase concentrations over time demonstrates that barnase production begins after lipase and activator levels have risen. And, if two molecules of lipase are made for every molecule of activator (for example, $m=2, n=1$ in above reactions), the proportionality of lipase and activator levels can be seen (Fig. 5A). Graphing barnase production versus lipase levels reveals the lipase threshold concentration. Over multiple runs, thresholding is consistently seen, but the actual threshold concentration is variable due to the stochastic nature of the system (Fig. 5B). The lower bound of the threshold level is determined by the number of repressor molecules as all of the repressor must be bound to have free PRA and barnase production. Thus, the threshold concentration of activator is at least the repressor concentration and the threshold concentration of lipase is proportional.

To ensure enough lipase is produced to effectively degrade plaques but not too much is made to be dangerous, the threshold level for barnase production is a critical parameter. Therefore, being able to control this value is key to the design. Two methods of tuning threshold levels have been examined. First, the ratio of lipase production to activator production can be altered (i.e. m and n are changed). This modification could be achieved by using different RBS sequences to control the translation. For example, if activator production is held constant ($n=1$), but lipase production is varied (m changes), the lipase threshold concentration varies by a factor of m (data not shown). Second, the rate of activator binding to the repressor can be modified using protein engineering techniques. This variability can be seen in a logarithmic parameter scan of kR with values 1 to 10,000 (Fig. 6). High values of kR represent tight binding so the threshold levels are low (close to the lower bound determined by repressor concentration). Low values of kR represent weaker binding so the threshold levels are higher as more activator (and lipase proportionally) is needed to ensure that there is no free repressor. The design of this suicide component coupled with the above tuning mechanisms will all for robust control of lipase production.

Discussion

Stochastic and deterministic modeling of the system showed the feasibility of the system. The time-delay architecture can be used to introduce more control into a genetic system by temporal separation of events, such as bacterial adhesion to arterial vessel cells and lipase production. We have identified several parameters in the system that can regulate the delay point, as well as the output amplification of the system. The activator-repressor threshold gating motif presented here shows very tight thresholding on the lower end of the activation curve, but due to stochastic effect, activation

can occur in a relatively large interval above the threshold parameter. Binding kinetics of the activator-repressor complex can modulate the response, and the ratio of activator to lipase production levels predictably affects the lower value of the threshold.

Future Directions. Although preliminary modeling completed in this work shows promise for the system, the system is highly complex and would require knowledge of many of the system parameters. Unfortunately, few works have described in mathematical detail the kinetics of promoters activated by protein rather than small molecules. Furthermore an appropriate molecule that can be attached to a stent and activate a downstream transcription factor needs to be analyzed. Alternatively small molecules can be eluted from the stent and used to induce the circuit; their concentration can be modeled by diffusion equations and time-delay parameters chosen so as to activate bacteria at only a certain distance away from the stent. In addition, a better atherosclerosis treatment system can be design if the bacteria were better able to achieve feedback from the lipid plaques themselves. Unfortunately we were unable to find any bacterial promoter whose activity was influenced by cholesterol levels in the environment, but the use of such a promoter would allow for direct feedback in the production of lipase and help implement a strategy for setting an upper threshold to lipase production. Clearly extensive circuit testing in bacterial models and then subsequent testing in mouse and human in vivo models is required. It is likely that many design characteristics would change as data is collected over the true dynamics of the system; however, the design and modeling presented in this work paves way to a novel therapeutic strategy for the treatment of atherosclerosis and overall use of engineered biological systems.

ACKNOWLEDGMENTS. Many thanks to Alistair Boettiger, Professor Adam Arkin, and Professor Ron Weiss for their assistance and guidance in completing this project.

1. Bottcher D., E. Brusehaber, K. Doderer, and U.T. Bornscheuer. 2007. Functional expression of the gamma-isoenzyme of pig liver carboxyl esterase in *Escherichia coli*. *Appl. Microbiol. Biotechnol.* 73:1282-9.
2. Hannsson, G.K. 2005. Inflammation, atherosclerosis, and coronary artery disease. *N. Engl. J. Med.* 352:1685-95.
3. Hayashi, K., M. Zhao, K. Yamauchi, N. Yamamoto, H. Tsuchiya, K. Tomita, and R.M. Hoffmann. 2009. Cancer metastasis directly eradicated by targeted therapy with a modified *Salmonella Typhimurium*. *J Cell Biochem.* 106:992-8. Insull, W. Jr. 2009. The Pathology of Atherosclerosis: Plaque development and plaque responses to medical treatment. *Am. J. Med.* 122(1 Suppl):S3-14.
4. Kim, K.S. 2002. Strategy of *Escherichia Coli* for crossing the blood-brain barrier. *J. Infectious Diseases.* 186:S220-4.
5. Lemichez, E., M. Lecuit, X. Nassif, and S. Bourdoulous. 2010. Breaking the wall: targeting of the endothelium by pathogenic bacteria. *Nat. Rev. Microbiol.* 8:93-104.
6. Mackman, N. 2008. Triggers, targets, and treatments for thrombosis. *Nature.* 451:914-8.
7. Majka, J. and C. Speck. 2007. Analysis of protein-DNA interaction using surface Plasmon resonance. *Adv. Biochem. Eng. Biotechnol.* 104:13-36.
8. Mead, J.R., S.A. Irvine, and D.P. Ramji. 2002. Lipoprotein lipase: structure, function, regulation, and role in disease. *J. Mol. Med.* 80:753-69.
9. Meiering, E.M., L. Serrano, and A.R. Fersht. 1992. Effect of active site residues in barnase on activity and stability. *J. Mol. Biol.* 225:585-9.
10. Ringrose, L., V. Lounnas, L. Ehrlich, F. Buchholz, R. Wade, and A.F. Stewart. 1998. Comparative kinetic analysis of FLP and cre recombinases: mathematical models for DNA binding and recombination. *J. Mol. Biol.* 284:363-84.
11. Schmidt-Dannert, C. 1999. Recombinant microbial lipases for biotechnological applications. *Bioorg. Med. Chem.* 7:2123-30.
12. Somerville, J. E. Jr., L. Cassiano, B. Bainbridge, M.D. Cunningham, and R.P. Darveau. 1996. A novel *Escherichia Coli* lipid A mutant that produces an anti-inflammatory lipopolysaccharide. *J. Clin Invest.* 97:359-365.
13. Tsutsumi, K. 2003. Lipoprotein lipase and atherosclerosis. *Curr. Vasc. Pharmacol.* 1:11-7.
14. Tullman-Ercek, D., M.P. DeLisa, Y. Kawarasaki, P. Iranpour, B. Ribnicky, T. Palmer, and G. Georgiou. 2007. Export pathway selectivity of *Escherichia coli* twin arginine translocation signal peptides. *J. Biol. Chem.* 277:511-25.
15. Weber, W., B.P. Kramer, and M. Fussenegger. 2007. A genetic time-delay circuitry in mammalian cells. *Biotechnol. Bioeng.* 98:894-902.
16. S Hoops, et al. COPASI-a CComplex PATHway Simulator *Bioinformatics*, 22-24:3067-3074, December 2006.

Modeling and investigating the role of positive feedback and network architecture in a single cell model of autocrine signal sensing and production: Adaptation and Bistability

Ioannis K. Zervantonakis *

*Department of Bioengineering, Massachusetts Institute of Technology

Submitted to Principles of Synthetic Biology, Fall 2010

In this work we present an autocrine single cell model, consisting of a cytokine sensing module in series with a cytokine production module including positive feedback. The aim of this work is to investigate the conditions and biochemical network architecture under which bistability and adaptation can be achieved. The system was modeled using deterministic mass action kinetics implemented in MATLAB to simulate the steady state behavior of the single cell model, as well as its response to a step impuls of external biochemical signal. A number of different model architectures were investigated including: no feedback, positive feedback with linear and non-linear autocrine signal production, and finally a model incorporating a time-delay between the cytokine sensing and production modules.

Feedback | Bistability | Adaptation | Modeling

Abbreviations: EGF, epidermal growth factor

Introduction

Autocrine signaling refers to a cells ability to produce biochemical signals, which can in turn induce signaling on the same cell through a feedback mechanism (Maly, Wiley et al. 2004). A well studied model of autocrine signaling is the epidermal growth factor (EGF) pathway, where a cell releases a given amount of growth factor recaptures a fraction of it, and the remaining can diffuse away signaling on distally located cells (Massague and Pandiella 1993). A cell does not only respond to these locally secreted signals, but also to other externally imposed signals and in turn may adapt its phenotype (e.g. decision to differentiate). In order to investigate this question, it is important to consider the concepts of adaptation (Ma, Trusina et al. 2009) and bistability (Ozbudak, Thattai et al. 2004), which are useful for investigating how a cell can exhibit memory to external stimuli and switch from a low to a high state, while also potentially adapting its response and potentially resetting its self in response to a stimulus.

To this end, we developed a single cell autocrine sensing model (figure 1) consisting of a cytokine production module coupled through a positive autocrine feedback to a cytokine sensing module. Our aim is to investigate how different architectures of the cytokine production module affect steady state and dynamical response to stimuli, bistability and adaptation in this system. To account for cytokine production having slower dynamics (e.g. influenced by diffusion) and exhibiting memory with respect to the cytokine sensing module we implemented a time delay from intracellular cytokine production to cytokine sensing. In the next section we present our simulation results, followed by a discussion of how this model compares to other existing models, suggestions for future applications/extensions and conclude with a methods section outlining some details of the model.

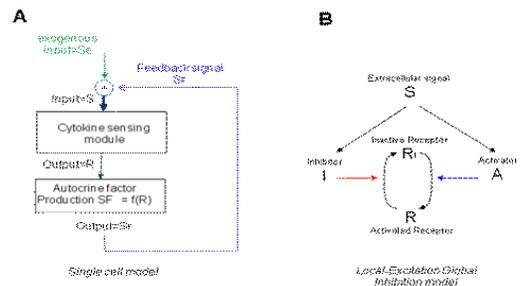


Fig. 1: (A) Architecture of single cell model developed in this project. (B) Local Excitation Global Inhibition Model comprising the cytokine sensing part of the single cell model

The modeled system is outlined in the following figure and described in detail in the methods section and appendix.

Methods

Derivation of system equations. The proposed model is outlined in figure 1 above and consists of a cytokine sensing module, the Local Excitation Global Inhibition (LEGI) (Figure 1B) model which transforms extracellular signals into activated receptors (e.g. via phosphorylation). In this work we have coupled this signal sensing module with a downstream signaling cascade which will result in cytokine secretion and diffusion into the surrounding environment which can bind on the same (autocrine) or a different cell (paracrine). The extracellular signal (S) is the sum of externally imposed signals (\cdot) and the autocrine feedback signal which can activate an activator (A) or an inhibitor (I), which in turn can switch the state of the response element from an inactive (R) to an active state (RI). In the following we provided the final equations implemented in MATLAB and provide their derivation in the appendix.

Creative Commons License:

CC-BY-NC-SA 3.0

<http://creativecommons.org/licenses/by-nc-sa/3.0/>

$$\frac{dA}{dt} = k_2 \frac{k_1}{k_2 + k_{1-}} \left(A_T - A \left(1 + \frac{k_1}{k_1 + k_{1-}} \left(\frac{R_T - R \left(1 + \frac{k_7}{k_1 + k_7} I + \frac{k_3}{k_{10} + k_3} B \right)}{1 + \frac{k_1}{k_1 + k_{1-}} A} \right) \right) \right) \quad (1)$$

$$- k_1 \frac{k_2}{k_2 + k_{2-}} A \left(\frac{R_T - R \left(1 + \frac{k_7}{k_1 + k_7} I + \frac{k_3}{k_{10} + k_3} B \right)}{1 + \frac{k_1}{k_1 + k_{1-}} A} \right) - (k_{1d} + k_{2d}) A \quad (2)$$

$$\frac{dI}{dt} = k_1 \frac{k_3}{k_1 + k_{3-}} \left(I_T - I \left(1 + \frac{k_7}{k_1 + k_7} R \right) \right) - S - k_1 \frac{k_2}{k_2 + k_{2-}} I \cdot R - (k_{1d} + k_{2d}) I \quad (2)$$

$$\frac{dR}{dt} = k_5 \frac{k_2}{k_5 + k_{5-}} A \left(\frac{R_T - R \left(1 + \frac{k_7}{k_1 + k_7} I + \frac{k_3}{k_{10} + k_3} B \right)}{1 + \frac{k_1}{k_1 + k_{1-}} A} \right) - k_5 \frac{k_7}{k_1 + k_7} I \cdot R - (k_{1d} + k_{2d}) R \quad (3)$$

$$\frac{dS_F}{dt} = \frac{k_{10} k_2}{k_{10} + k_{2-}} B \cdot R - k_{20} S_F \quad (4)$$

For the model with feedback: $S = S_F + S_E$

Models without feedback and with non-linear autocrine factor feedback.

- For the model equations without feedback, the input to the system is $S = S_E$
- For the model with non-linear production terms of S_F , we include receptor dimerization, so that we replace BR in the above equations with BR^2 .

Time delay modeling in autocrine positive feedback. We used Pades approximation to implement a delay (τ) in autocrine factor sensing. This is modeled by using Pades approximation to map the feedback signal S_F to a delayed signal $S_{F,D}$, and was implemented in Matlab using the following equation, which was solved simultaneously with the four above equation. To calculate the coefficients (a,b) we used the pade function in matlab for a given order (e.g. n=1 below) of the approximation,

$$a_0 S_F + a_1 \frac{dS_F}{dt} = b_0 S_{F,D} + b_1 \frac{dS_{F,D}}{dt}$$

Implementation. The above system of ODEs for all 4 different models (5 for the system with delay) was solved in matlab using ode45, with zero initial conditions, and the same parameter sets for comparing the model architecture. For the transient model response we use a step impulse function for the external signal was model as a step impulse, as given below:

$$S_E = S_{E,0} \text{ for } t < t_1$$

$$S_E = S_{E,0} + S_{E,A} \text{ for } t_1 < t < t_1 + \delta$$

$$S_E = S_{E,0} \text{ for } t > t_1 + \delta$$

Parameters used.

k_1, k_1	2	R_T	10
k_2	2	k_{C1}	1
k_1, k_1	1	k_{C2}	1
k_2	1	k_{C3}	1
k_3, k_3	1	k_{C4}	1
k_4, k_4	1	k_{C5}	1
k_5, k_5	1	k_{C6}	1
k_6, k_6	1	k_{C7}	1
k_7	1	k_{C8}	1
k_8	1	k_{C9}	1
k_9, k_9	1	k_{C10}	1
k_{10}	10	B	0.1
k_{10}, k_{10}	1	$S_{E,0}, S_{E,A}$	0.001 and 0.1
k_{10}	10	$t_1, \Delta t_p$	100 and 10
k_{10}	10	Total simulation time	1-30
k_{10}	10	and timestep	200 and 1

Results

A detailed derivation of the equations describing the system is given in the methods section, while in the following section we present the main results. Below we present the four variations of the single cell autocrine model we studied to understand how network architecture (linear vs. non-linear) and feedback affects the dynamics of adaptation in a single cell model:

After performing a parametric analysis on the model we found that we can have two qualitative different regimes: (a) Produced autocrine signal is of comparable magnitude to externally imposed signal, and (b) Produced autocrine signal is much smaller to the externally imposed signal. We found that the parameters controlling autocrine signal production/degradation critically influence this: B (substrate yielding production of the autocrine signal), rates for production (k10) and for receptor R - substrate B complex formation

Model architecture examined	Notes
No Positive Feedback	System only responds to external stimulus S_E
Positive Feedback	System responds to external stimulus S_E and autocrine factor S_F
Positive Feedback with time delay	System responds to external stimulus S_E and to autocrine factor S_F , whose sensing by the LEGI model is delay
Positive Non-linear Feedback	Autocrine signal produced by receptor dimer (n1=2) binding to substrate B

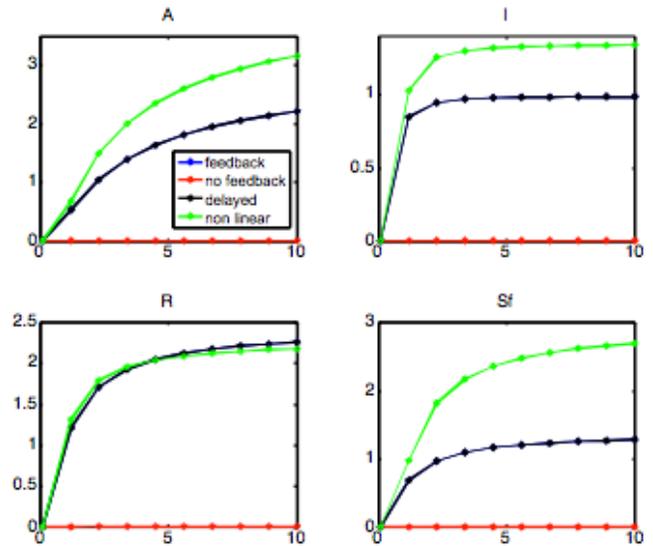


Fig. 2: Parameter study of the effect of the reaction rate k_2 (x-axis), on steady state solution values (y-axis) of activator (A), inhibitor (I), receptor (R) and autocrine signal (Sf) for the model architecture with no feedback (red), with feedback (blue line), with feedback including a time delay (black line) of 10 time-units, and a non-linear feedback (green line) involving receptor dimerization.

(k_9 , k_{9-}), and finally degradation rate for the autocrine signal ($kdSf$).

For the analysis performed below, we focus on the first scenario, where autocrine signal magnitude is comparable to external signal, and which makes the analysis interesting and the consideration of positive feedback essential. We provide the parameter values giving this behavior in the appendix.

System Stability. To investigate system stability we have computed the jacobian matrix evaluated around the steady state point, as shown in the appendix in part A2. Using the matlab function (`eig()`) we found for the 4x4 matrix corresponding to the 4 coupled equations (see methods) that all eigenvalues are real and positive, suggesting that the system is stable, as it can also be carefully seen by the dynamic response of the system to a step impulse.

Steady state solution: Parameter analysis. In the following we present two representative examples of steady state solution sensitivity to model parameters (as shown in methods). In figure 2, we plot the variation in activator (A top left panel), inhibitor (I top right), receptor (R bottom left panel) and autocrine signal (bottom right panel) with respect to the rate constant k_2 for all model architectures. The rate constant k_2 determines how strongly the input signal S induces production of activator and subsequently receptor activation. We see that the delayed feedback (black line) and the feedback (blue line) model show the same behavior at steady state with the two curves coinciding. This suggests that after sufficient time, where the transients have decayed, at steady state, the model exhibits adaptation independent of the delay. As expected, the non-linear model (green line), as shows maximal response, due to the amplifying positive feedback. Finally, since the regime examined here is for high values of autocrine signal, we see that the no feedback model has very low steady state values.

In figure 3, we present a counter-example to demonstrate how another parameter does not influence the system response at steady state. Here in all panels, the x-axis corresponds to the time-delay (τ), and we see that the steady state behavior

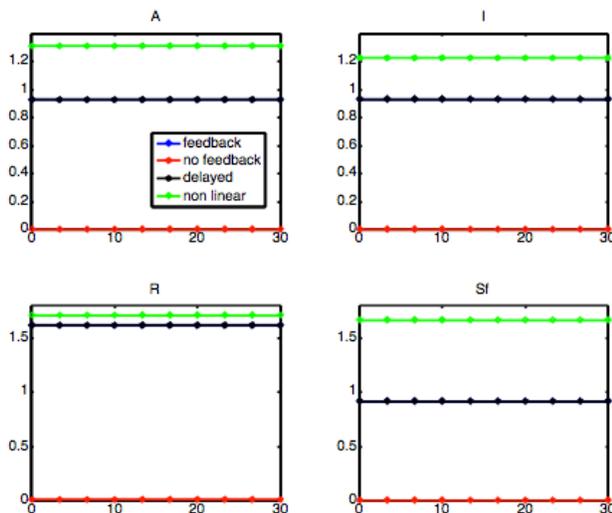


Fig. 4: Parameter study of the effect of the time delay τ (x-axis), on steady state solution. Legend is same as figure 2.

is independent of its value, as expected because the system has recovered from transients.

The above two examples demonstrate that the implemented model equations in MATLAB and the analysis give consistent results, and hence serve as evidence for the validity of the model/method. We should also note, that the steady state solutions were computed by solving the system dynamics and letting it run for long enough time (t 200 time units) until it reaches steady states. In the appendix we demonstrate how under simplifying assumption for the complexes ARI, BR, IR, etc formed we may compute analytic solutions of the steady state solutions.

Demonstration and analysis of bistability. Due to the autocrine signal feeding back into the cytokine sensing module, the equations describing the production term for this feedback signal become nonlinear and this allows for the nullclines to intersect at more than one point, allowing for multiple steady states.

In the left panel of figure 4, we see that only the system with non-linear (green) feedback exhibits two non-trivial (non-zero) steady state solutions, as shown by the low and high values for all the variables A, I, R, Sf. We further prove that the system is exhibiting bi-stability by changing the initial values from 0 to 10 and now we see that the system switches to the high solution.

By further examining the value of this second low steady state, we confirmed that it is not the trivial 0,0,0,0 solution, which is the initial condition that the system was started from. However it is still 3 orders of magnitude lower than the high stable state. The condition to achieve bistability as shown in the following figure (and can also be shown through the null-clines intersecting) is sensitive upon proper selection of the parameter values (here k_2 range 2 4 plotted in x-axis, and $B=1$, $kdSF=1.6$).

Although not shown here we also performed an analysis to see how the degree of non-linearity affects the bistability, as we found that a hill coefficient of $n=1.5$ was not enough to induce bistability. Finally, it is important to note here that in order to get bistability, it is not simply sufficient to change kinetics parameters arbitrarily, since the parameters influence simultaneously all 4 equations of the model. The capacity of the system to exhibit bistability is also shown through analytical work in the appendix A3.

System response to a pulse in external cytokine stimulus.

Apart from examining the system response at steady state it is also very important to examine system response to transient/dynamical inputs, where the different system architectures: positive feedback (linear/non-linear) and feedback with delay influence the dynamics of cytokine sensing and propagation to the downstream autocrine signal secretion unit. The external signal was model as a step impuls, as described in the methods, and also shown in the last panel of Figure 5.

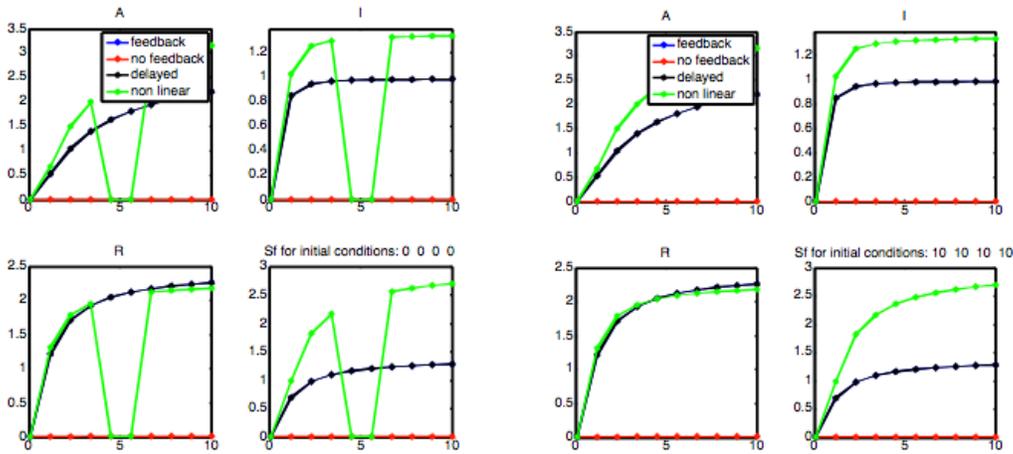


Fig. 5: Demonstration of bistability. Left panel: initial conditions $[0\ 0\ 0\ 0]$ and Right panel: initial conditions $[10\ 10\ 10\ 10]$. Parameter study of the effect of the reaction rate k_2 (x-axis), on steady state solution of activator (A), inhibitor (I), receptor (R) and autocrine signal for the model architecture with no feedback (red), with feedback (blue line-colocalizing with black line), with feedback and a time-delay of 10timeunits, and a non-linear feedback involving receptor.

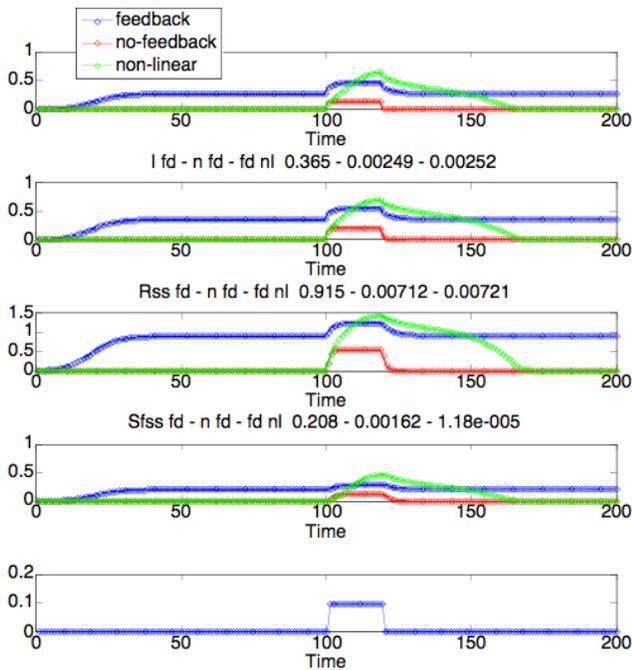


Fig. 3: Dynamical behavior of system: reaching steady state from zero initial conditions ($t=0$ 100), application of a step pulse in extracellular signal Se ($t=100$ 110) and removal, for all the four variables (A, I, R, Sf) and for three model architectures with feedback (blue line), no feedback (red) and non-linear (green) feedback.

In figure 4 we have plotted the evolution of activator (A top panel), inhibitor (I second panel), receptor (R third panel), autocrine signal (Sf fourth panel) and externally imposed signal (Se fifth panel) for the models with positive feedback (blue line). From the above figure we can make a number

of observations, we see that all three models adapt back to the steady state solution after the external signal has been removed, suggesting perfect adaptation. However, the dynamics of the response depend on the model architecture. We also see that the sensitivity to the applied external signal depends on the model architecture (non-linear model is more sensitive) compared to the linear model.

Discussion

From our analysis we can draw the following conclusions: The system has been shown to be stable for the above parameters, both by using linearized stability analysis (Jacobian computation) and by careful inspection of the dynamic analysis for a large number of parameter studies. We demonstrated the parameter set giving autocrine feedback signal values comparable to the external signal. The system responds to the step pulse of external signal, but the response depends on the level of feedback. For achieving bistability we have found that a non-linear response is required and that this is possible only in a range of parameter values. The system exhibits perfect adaptation at steady state, because of the architecture of the activator/inhibitor binding to the receptor. The dynamics of adaptation behavior depend critically on the rate constants of degradation/production of the autocrine signal, and the non-linearity (Hill coefficient)

In this work we developed a single cell autocrine model by combining a cytokine sensing module with a cytokine production model and implementing positive feedback. Through simulation we demonstrated how the reaction rate constants and model architecture affects steady state and dynamic behavior, while we also demonstrated that the system with non-linear feedback can exhibit bistability. The cytokine sensing module is known as the Local Excitation Global Inhibition Model (LEGI) which was discussed in (Levchenko and Iglesias 2002) and showed that can exhibit adaptation. In this work, we implemented a positive feedback loop to the LEGI and demonstrated that the system incorporating autocrine feedback can exhibit bistability under specific parameter choice.

Our results are consistent with other studies which have also demonstrated that in order to achieve bistability, non-linear generation of the involved protein and feedback is re-

quired. This feedback may be implemented in different locations in the model, such as downstream of receptor (Ozbudak, Thattai et al. 2004), and in this work we show that feedback through autocrine signaling can also lead to bistability, due to the equations becoming non-linear and the null-clines intersecting at multiple points. This is an important system characteristic and allows for memory, since as we demonstrated here the system can switch state depending on its previous state (initial conditions). In the future, one possible application of this model in the field of synthetic biology could be to construct this autocrine single cell module using synthetic receptors and protein secretory modules to program a cell (e.g. stem cell) to achieve bistability, while also exhibiting different response dynamics to externally applied signals through the action of positive feedback.

The developed model here may find application in understanding how stem cells differentiate into different cell types, based on extracellular biochemical signals. (Chickarmane, Troein et al. 2006) demonstrated that the underlying tran-

scriptional regulatory circuitry responsible for cell determination can be modeled as a bistable switch. Furthermore, the model could also be helpful in serving as a unit model system for modeling model cell-cell communication, where cells both sense each other through cytokine sensing and respond by secreting cytokines, which may act on the same cell or a distant cell. Although not studied in this work, we may also write the mass-reaction kinetics including diffusion of species within the cells, and this way we could model polarization of the signaling sensing machinery based on positive feedback loops, as also showed in a recent paper by (Xiong, Huang et al. 2010) Other possibilities, include modeling the system in a way that the external / internal signals feed into different receptors which in turn feed into the same downstream signaling node.

ACKNOWLEDGMENTS. We thank Alistair Boettiger for support in formulating our gene circuit and aid with COPASI simulations, Adam Arkin and Ron Weiss for teaching the pilot PoSB class of 2010, and the Berkeley PoSB class for helpful discussion.

1. Chickarmane, V., C. Troein, et al. (2006). "Transcriptional dynamics of the embryonic stem cell switch." *PLoS Comput Biol* 2(9): e123.
2. Levchenko, A. and P. A. Iglesias (2002). "Models of eukaryotic gradient sensing: application to chemotaxis of amoebae and neutrophils." *Biophys J* 82(1 Pt 1): 50-63.
3. Ma, W., A. Trusina, et al. (2009). "Defining network topologies that can achieve biochemical adaptation." *Cell* 138(4): 760-73.
4. Maly, I. V., H. S. Wiley, et al. (2004). "Self-organization of polarized cell signaling via autocrine circuits: computational model analysis." *Biophys J* 86(1 Pt 1): 10-22.
5. Massague, J. and A. Pandiella (1993). "Membrane-anchored growth factors." *Annu Rev Biochem* 62: 515-41.
6. Ozbudak, E. M., M. Thattai, et al. (2004). "Multistability in the lactose utilization network of *Escherichia coli*." *Nature* 427(6976): 737-40.
7. Xiong, Y., C. H. Huang, et al. (2010). "Cells navigate with a local-excitation, global-inhibition-biased excitable network." *Proc Natl Acad Sci U S A* 107(40): 17079-86.

A Programmable Decoder

Josephine Shaw *

*Department of Bioengineering, Massachusetts Institute of Technology

Submitted to Principles of Synthetic Biology, Fall 2010

Synthetic biologists have made sensors out of simple organisms by engineering genetic networks to achieve a desired response. Cells may be designed to respond to environmental stimuli, such as light, oxygen, nutrients, and toxins. It is difficult, however to sense multiple inputs without having multiple outputs, which may become complex and even unfeasible. Here I present a design for a programmable two-bit decoder that allows the user to choose which combination of two inputs will produce a simple bioluminescent output response. In this manner, only one type of output is necessary, alleviating the need for complicated or cumbersome techniques. The system is modeled and simulated here to demonstrate functionality.

Genetic Logic | Circuit | Modeling

Abbreviations: EGF, epidermal growth factor

Introduction

Simple living organisms can be used to create informative sensors that can respond to a variety of helpful or harmful environmental stimuli [1]. Using the digital logic of a decoder, genetically altered cells could be employed to sense not only specific inputs, but even combinations of those inputs. One could imagine that in different environments, disparate blends of environmental toxins or compounds would be of interest. With a programmable system, the user would be able to give control signals to the cells to dictate which combination of environmental cues will produce an output response. The ability to program the decoder would allow one system to be used as a sensor in multiple environments and applications. Furthermore, the programmable feature allows for the use of one simple output. In the case presented here, a common bioluminescent luciferase-based response is used as the final output. As more inputs are used in a sensor, the most direct way to view the output is to have a different-colored molecule corresponding to each input. However, this is limited by the number of colored proteins are available, the complexity of visualization, and the metabolic load that can be supported by the cells. For example, if multiple fluorescent proteins were used as outputs, the viewing of them would require complicated optics specific for each color. In addition, multiple colors of luciferases have been used, but the multi-colored data have had to be retrieved through complex deconvolution methods [2]. Thus it would be useful to create a system in which the same output molecule is used for different signals, as directed

by the user. Figure 1 diagrams the programmable two-bit decoder, and its corresponding truth table is in Table 1.

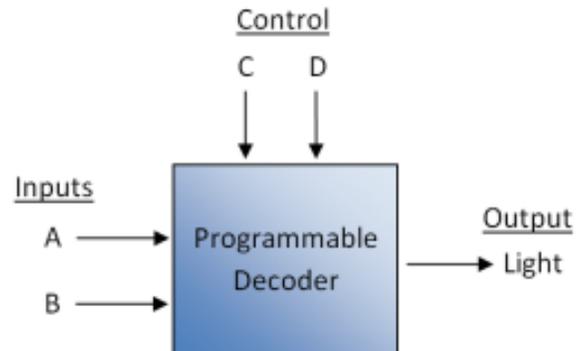


Fig. 1: Decoder setup

Table 1. Truth table for programmable decoder

A	B	C	D	Output
0	0	0	0	1
0	1	0	1	1
1	0	1	0	1
1	1	1	1	1
Every	-thing	else	is	0

Design In the design of the system, there is one section involving transcriptional regulation, and another section involving protein-protein interactions. Since the sensor aspect of the system should have a rapid response, it depends upon protein-protein interactions. However, because timing is not as important for programming the system, transcriptional regulation is used. The programmable functionality, using inputs C and D, will be referred to as the control segment, and the protein interaction part with inputs A and B will be referred to as the sensor segment. A schematic diagram of the biological circuit design is shown in Figure 2.

Creative Commons License:

CC-BY-NC-SA 3.0

<http://creativecommons.org/licenses/by-nc-sa/3.0/>

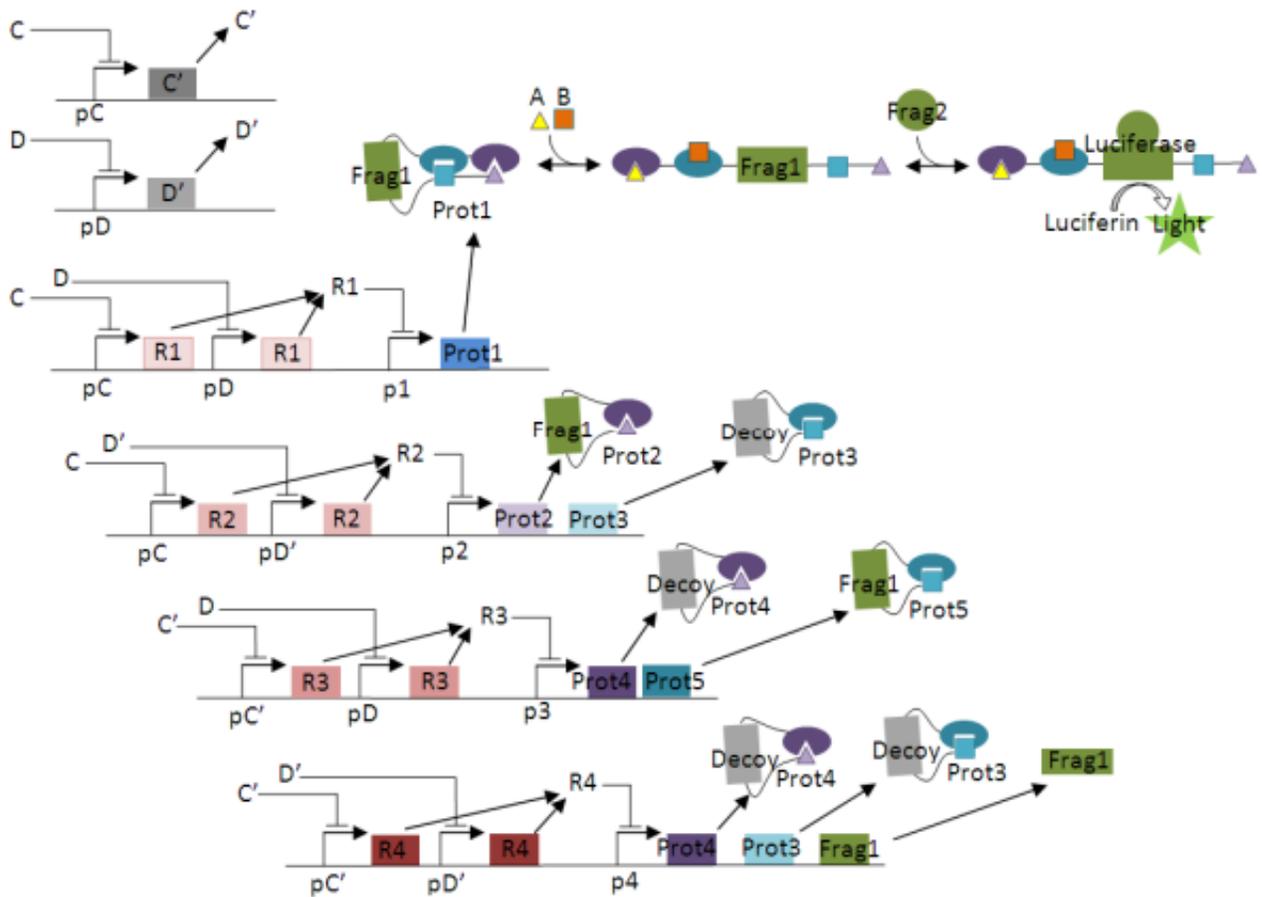


Fig. 2: Biological circuit diagram, involving both transcriptional regulation (left side), and protein-protein interactions (right side). Note that the decoy fragment has higher affinity to Frag2 than does Frag2 to diminish output when it is available to bind.

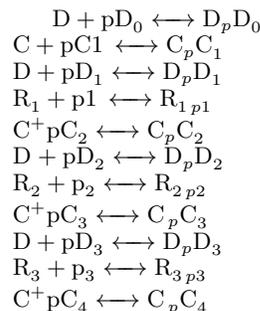
The control segment involves eight repressible promoters, and takes C and D as input as repressors of different promoters. Similar to a traditional digital two-bit decoder, the genetic circuit has two inverters and four AND gates to produce one of four sets of fusion proteins for sensing A and B. Thus, the control inputs dictate the protein expression of the cells. Next, the sensor segment uses the particular proteins produced by the control segment to generate the desired output. The fusion proteins are switches, similar to those described by Dueber et al [3]. For example, in Figure 2, Prot1 is an AND gate that has sections which bind to the small-molecule ligands A and B. The fusion protein has its own version of A and B tethered to itself, but with lower affinity than the actual intermolecular, non-tethered ligands. In the folded state, when Prot1 is bound to itself, it hinders Frag1 from binding to Frag2, both of which are fragments of a luciferase. When A and B are present, they bind to the fusion protein, causing it to change conformation to allow access to Frag2. Upon binding, Frag1 and Frag2 become an active luciferase, converting luciferin into the bioluminescent output signal. Prot2 and Prot5 are similar, but they only bind to one of the two input ligands. Then, Prot3 and Prot4 are different in that instead of having a Frag1 section, they have a decoy protein fragment. This decoy, when available to bind, has a much stronger affinity to the Frag2 fragment than does Frag1. Hence, if a decoy becomes open for binding, it com-

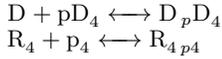
petes strongly against Frag1 to diminish the output signal. The idea of a decoy has been used in other protein interaction synthetic biology [4].

Kinetic Model

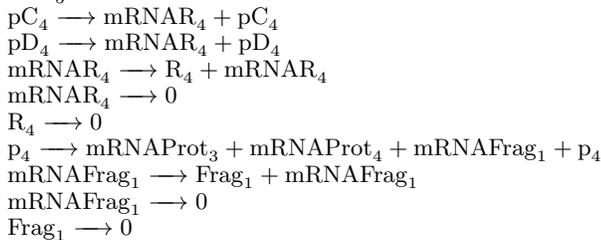
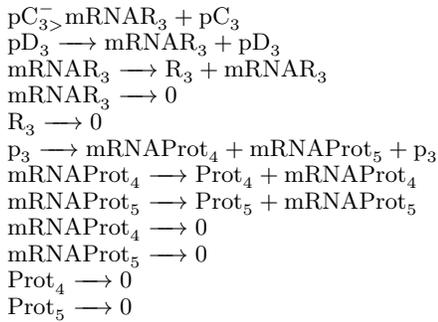
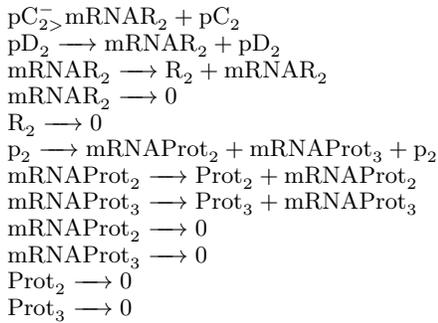
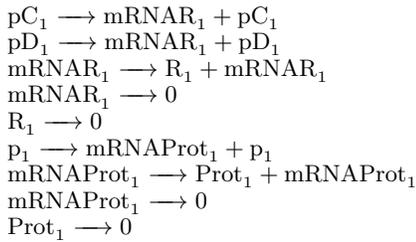
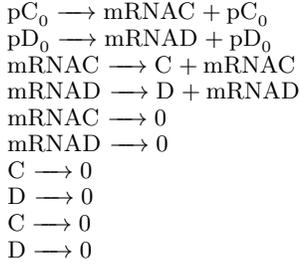
The biological circuit was modeled using mass-action kinetics. There are 183 equations in total (counting reversible reactions as two), which are shown below. The corresponding differential equations are included in the Supplementary Material, Part A. Michaelis-Menten kinetics is assumed for luciferase, as done by Ignowski et al [5].

Control Segment: Transcriptional Regulation Fast Reactions (Binding and Unbinding) $C + pC_0 \rightleftharpoons C_p C_0$



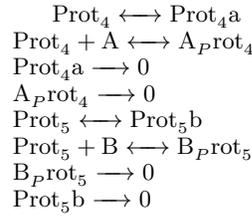
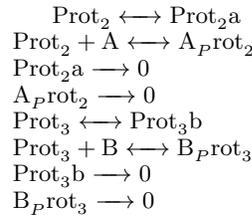
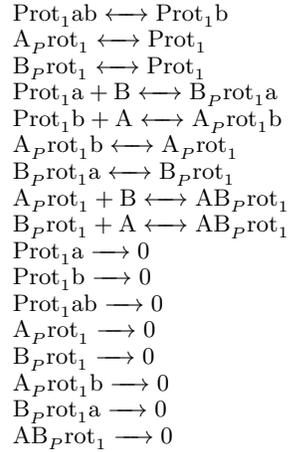


Slower Reactions

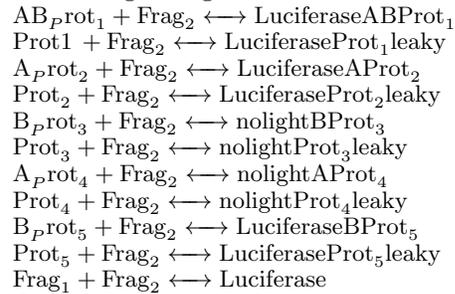


Sensor Segment: Protein-Protein Interactions Lower case a and b represent parts of the fusion protein, whereas capital A and B are the actual inputs of the system. Binding/Unbinding of protein to itself and inputs $\text{Prot}_1 a \longleftrightarrow \text{Prot}_1$

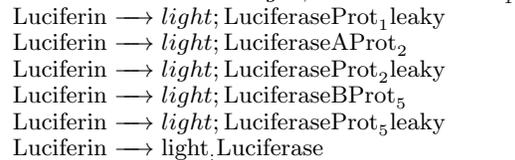
$$\text{Prot}_1 b \longleftrightarrow \text{Prot}_1$$

$$\text{Prot}_1 ab \longleftrightarrow \text{Prot}_1 a$$


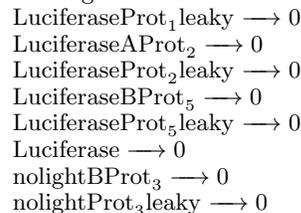
Binding of fragments to form active Luciferase

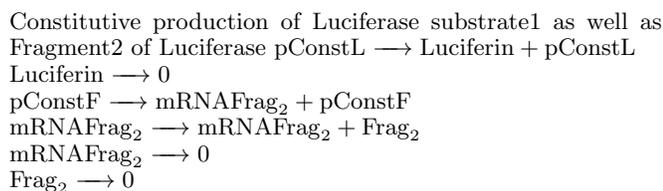
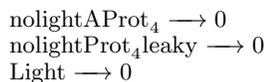


Active Luciferase producing light using Michaelis- Menten kinetics



Degradation LuciferaseABProt₁ → 0





Complexity in the model arises not only because of the many interacting species, but also the different conformations of those species and the disparate timescales of the reactions. For example, the fusion protein AND gate, Prot1, has nine different bound states, with different combinations of binding to intramolecular and intermolecular ligands [3]. Changing from each state to another requires a reversible reaction, totaling in twelve reactions for Prot1 alone. Furthermore, if the dynamics of the transcriptional regulation were the endpoint, one could easily make a simplifying assumption that the binding and unbinding reactions occur much faster than transcription and translation. Then, one could use equilibrium constants for the binding and unbinding reactions, rather than using the entire reaction with its own differential equations. However, in this system, it is of interest to observe the dynamics of the protein-protein interactions, since they compose the sensor. Thus, all the equations were kept in the model without simplification, and many kinetic constants were gleaned through a thorough search in existing literature.

Results

Deterministic model. The kinetic model was simulated in COPASI to observe the temporal dynamics of the system as well as verify that the desired behavior is indeed attainable with biologically relevant constants. The model was for a network within one cell, assuming that a cell has an approximate volume of 10-15L. Modeling of transcription, translation, luciferase activity, and general protein interactions has been done in literature, and although these constants did not always agree from one paper to another, I attempted to combine what I could while retaining a reasonable biological values [5-7].

In general, the rate of translation is slower than the rate of protein binding, and the degradation rate of proteins is slower than that of mRNA. A complete list of the kinetic constants used can be found in the Supplementary Material, Part B. Figure 3 depicts the results of the simulation. For each graph, the control state (each row) was set as having combinations of C high (C), C low (C), D high (D), and D low (D). Here, a high initial concentration was set to 106 molecules/fL, which corresponds to 1.7 mM concentration, and low was set to zero. Then, to see if the system could sense the combinations of small molecule inputs A and B, the high input was set to 1000 molecules/fL (1.7M), while the low input was set to 0 molecules/fL. In all cases, the initial values were set such that there was one of all the unbound promoters, Luciferin concentration of 106 molecules/fL, and other than the specified inputs, all other species were set to zero.

The slow rate of translation causes multiple proteins towards the end of the cascade to be produced, since it takes time before the repressors get expressed and perform their function. Even with enhanced ribosome binding sites for higher translation rates, taken account for in the model, there must still be a delay based upon the sequential, cascaded de-

sign of the transcriptional network. From data that is not shown, it takes approximately 600 seconds (10 minutes) for the programmed state to be set. As a more concrete example, referring to names given in Figure 2, if the user inputs high concentrations of C and D, it takes 600 seconds for the amount of Prot1 produced to exceed the amounts of other proteins, such as Prot2 and Prot3 which must wait for R2 to be transcribed and translated to repress their production. To acquire a a better signal, it is possible that waiting 600 seconds before introducing the system to a new environment to sense the presence of A and B may be beneficial. In the results shown, 1000 seconds was used as a final time point at which to compare the light output of each state. This timing corresponds to 6.7 minutes after the appropriate protein levels were reached. Thus, the protein interaction segment of the system was indeed faster to provide a response than was the transcriptionally-regulated segment. It is noteworthy that for a given programmed state, each corresponding to a row in Figure 3, the correct input provides high output.

These are shown by the double-boxed cells along the diagonal of the chart. For example, when the state is set by CD, the user wants high output signal when the system senses the presence of both A and B. The top left graph in Figure 3 demonstrates that for the given programmed state, the output when A and B are both present is at least twice as strong than signals from other combinations of A and B. All of the programmed states have similarly good output signals except for CD, where the desired high output is only 10% higher than the next highest output programmed by CD. This signal could easily be masked by noise and the stochasticity of the actual system. This result, however, was the best I could achieve given biological constraints with rate constants.

Stochastic simulations. As previously mentioned, the kinetic parameters of the system are suited to one cell, having one of each promoter in the design. Since the molecule numbers are low, stochastic behavior plays an important role in determining the dynamics of the system. Although in actual practice many cells would be used together to smooth out the overall response, it is still interesting to observe the stochastic cell-to-cell behavior. Here, I compare the noise seen in the protein response as a result of translational elements, versus the noise in light response generated via protein-protein interactions. Figure 4a shows the result of ten stochastic simulations of Prot1 production when the system is set with C and D both at high concentrations. To be consistent with the previous observation regarding timing, the level of Prot1 was taken for each simulation at 600 seconds. The fano factor, taken to be the variance over the mean, for Prot1 production is 0.885. The count for Prot1 also includes the other variations of Prot1 that may be bound to itself at any of its two ligand binding sites. In comparison, Figure 4b portrays 10 stochastic simulations where not only C and D are high, but A and B are at high concentrations also. The amount of light produced is highly variant. At 1000 seconds, the mean amount of light produced is 13.9 a.u. and the variance is 58.5. Thus, the fano factor for light production is 4.2, which indicates much more dispersion than Prot1 production seen earlier.

Although the values of these fano factors may not in themselves give biological insight because the kinetic models are simplistic compared to the actual mechanism of action, the comparison of the two values conveys information regarding noise in the system. It is thought that translation is primarily responsible for the noise in gene expression [8], which means that there should be noise in the expression of Prot1 from the transcriptional network. As the system is cascaded, having the protein-protein interaction sensor downstream, the

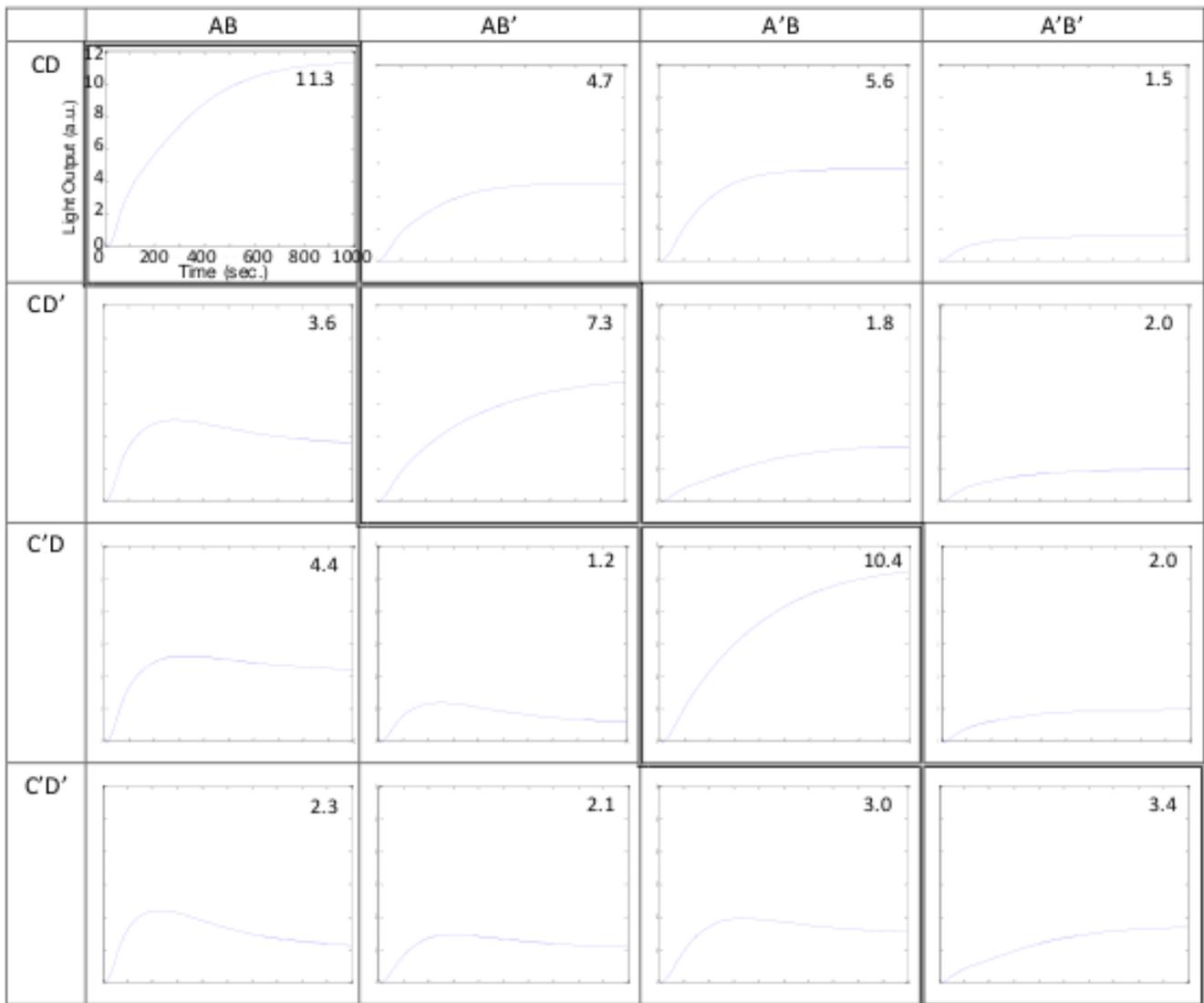


Fig. 3: The resultant graphs from COPASI simulations of each combination of inputs. Note that the double-bordered cells along the diagonal of this table are the conditions that should result in a high output. The axes are labeled only in the top left box to conserve space, but each graph is on the same scale. The resultant light output in arbitrary units (a.u.) after 1000 seconds (16.7 min) is reported in the upper right corner each graph. It is worth noting that in each row, that is, in each state that is set by the control inputs C and D, the highlighted box has the highest output value, corresponding to the proper function of the decoder.

noise will be amplified. Not only are the protein interactions variable intrinsically, they also depend upon protein concentrations that are determined by the noisy translation process. This dependency causes the noise to be amplified, as seen in the high variability of the final light output.

Discussion

After modeling the programmable decoder, the question arises as to whether or not such a system could be fabricated in real biology. Looking at the registry of parts, it seems that due to the necessity of eight distinct repressible promoters in the current design, it would only be feasibly carried out in *E. Coli*. Examples of common repressible promoters that could be used are pTet, pLac, and pBad. It is possible that finding eight promoters that work well may be challenging in *E. Coli* as well. If that is the case, the design may be altered to replace a few of the repressible promoters with some positively reg-

ulated promoters instead. The luciferase systems have been well studied, and there is a variety that can be used for the bioluminescent output, such as firefly luciferase, click beetle luciferase, and Renilla luciferase [2,9].

In fact, many luciferases are used in protein fragment complementation assays, where the two fragments are initially split and fused to other interacting proteins, and then they become an active enzyme upon binding. This is exactly what would be used in the fusion proteins described in the programmable decoder. Furthermore, in the case that a fluorescent readout is desired, split-GFP could be used. In addition, the system presented here may be complicated to produce because there would need to be some ribosome binding site alterations to increase translation rates. There may also be a need for some degradation tags, as some degradation rates needed to be slightly enhanced in order for the circuit to produce the desired behavior. Furthermore, the system behavior produced in the model was not quite adequate in producing the CD

state, as presented earlier (Fig. 3), which may indicate difficulty in doing so in actual biology. A possible improvement to the presented design is to use a different method of protein interactions to produce the output. The decoy proteins were required to compete with the active fragments and were not always very effective in adequately diminishing the output signal, even though the decoy has higher binding affinity to the second luciferase fragment. It may be of interest to use a protein such as TetR that could be differentially induced to bind or unbind. Although this could obviate the need for a decoy, it may not be as versatile since not many proteins have that degree of controllability. With the current design, the fusion proteins flanking the luciferase fragment or decoy fragment can be altered depending on which input ligands are to be sensed by the organism. The user can decide which inputs to sense and design the fusion proteins accordingly. For example, it could sense nutrients, metals, or toxins in the environment. Then, the user can program which combination of inputs will generate an output signal, requiring only a simple visual cue to know if the desired inputs are present. Furthermore, the sensing of the inputs through protein interactions

occurs rapidly within minutes. With further optimization, the programmable system may be attainable.

ACKNOWLEDGMENTS. I would like to thank Professor Weiss for help with the overall idea of the project, as well as Noah Davidson and Deepak Mishra for helpful discussions.

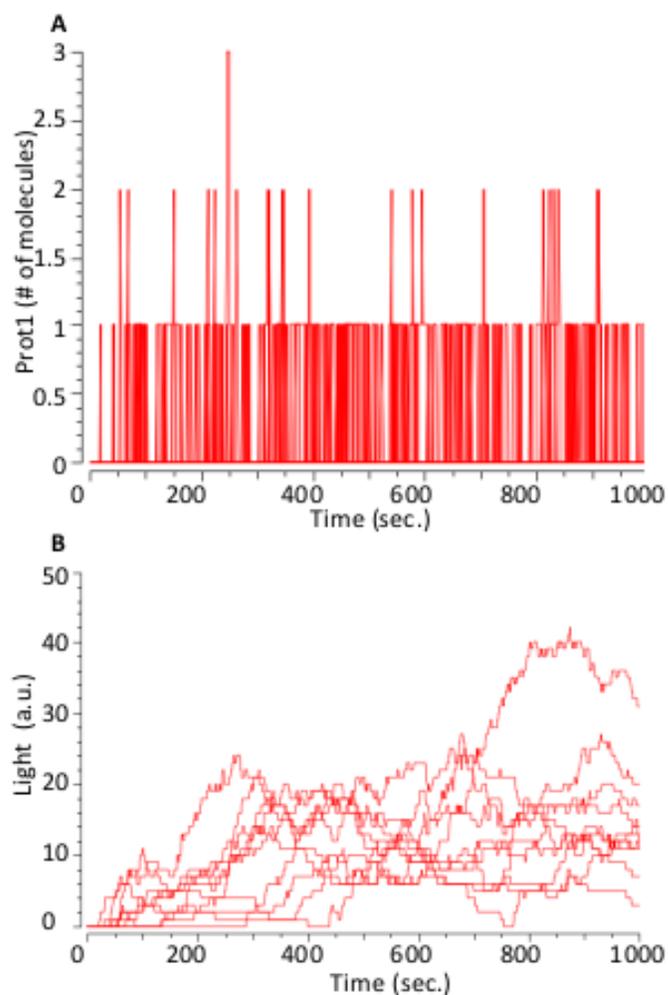


Fig. 4: Results from ten stochastic simulations. A) Prot1 levels when C and D are high, and A and B are zero. B) Light output when C and D are high, and A and B are high.

1. K. Yagi, Applications of whole-cell bacterial sensors in biotechnology and environmental science., *Applied microbiology and biotechnology*, vol. 73, Jan. 2007, pp. 1251- 8.
2. V. Villalobos, S. Naik, M. Bruinsma, R.S. Dothager, M.-H. Pan, M. Samrakandi, B. Moss, A. Elhammali, and D. Piwnica-Worms, Dual-color click beetle luciferase hetero-protein fragment complementation assays., *Chemistry & biology*, vol. 17, Sep. 2010, pp. 1018-29.
3. J.E. Dueber, B.J. Yeh, K. Chak, and W. a Lim, Reprogramming control of an allosteric signaling switch through modular recombination., *Science (New York, N.Y.)*, vol. 301, Sep. 2003, pp. 1904-8.
4. C.J. Bashor, N.C. Helman, S. Yan, and W. a Lim, Using engineered scaffold interactions to reshape MAP kinase pathway signaling dynamics., *Science (New York, N.Y.)*, vol. 319, Mar. 2008, pp. 1539-43.
5. J.M. Ignowski and D.V. Schaffer, Kinetic analysis and modeling of firefly luciferase as a quantitative reporter gene in live mammalian cells., *Biotechnology and bioengineering*, vol. 86, Jun. 2004, pp. 827- 34.
6. V. Villalobos, S. Naik, M. Bruinsma, R.S. Dothager, M.-H. Pan, M. Samrakandi, B. Moss, A. Elhammali, and D. Piwnica-Worms, Dual-color click beetle luciferase hetero-protein fragment complementation assays., *Chemistry & biology*, vol. 17, Sep. 2010, pp. 1018-29.
7. A. M. Kierzek, J. Zaim, and P. Zielenkiewicz, The effect of transcription and translation initiation frequencies on the stochastic fluctuations in prokaryotic gene expression., *The Journal of biological chemistry*, vol. 276, Mar. 2001, pp. 8165-72.

Analysis of Noise Propagation in Biological Cascades and Feedback Networks

Sabino Pietrangelo* and Rushil Goel*

*EECS MIT

Submitted to Principles of Synthetic Biology

Stochastic effects are an important, yet often poorly understood, aspect of biological circuit behavior. In most current work, stochastic effects are often quantified late in the design process and stochastic simulations lead to limited understanding of the underlying noise contributions. In this project, we discuss a markedly different design paradigm wherein we develop an analytical framework, based on ideas from control theory and electrical engineering, for the evaluation of noise propagation in biological cascades and feedback networks. By comparing noise characteristics of the analytical model to simulation results, we are able to show that our model captures the expected noise properties, while at the same time preserving insight. It is our belief that such a tractable model of noise propagation will allow synthetic biologists to make logical trade-offs during the design process in order to achieve desired noise characteristics.

Introduction

Noise is inherent in biological circuits and can often drastically change the expected output [4]. In more complicated circuit topologies, like cascades and feedback networks, noise characteristics depend on the topology itself as well as the different reaction rate parameters, often in a not so obvious manner. This limited understanding often restricts a designer's ability to build biological circuits with desired noise characteristics. In this work, we utilize techniques from control theory and electrical engineering to develop an approximate noise model for several biological circuit topologies. Such a model will allow synthetic biologists to better understand the effect of changing different design parameters and topologies on noise, and hence build complicated circuits with desired noise properties.

Current work on understanding noise propagation, including [3] is often driven by simulations over a wide range of parameter values to try and gain insight into the factors that affect noise. Such an approach is not particularly useful for the rational design of circuits and parameters since even running a large number of simulations doesn't always allow us to understand the role of different parameters in affecting noise. Nevertheless, there has been work on developing a mathematically rigorous model based on differential equations for noise in the expression of a single gene [5, 6]. While such a model is able to clearly show the effect of different parameters on noise, it is hard to extend it to more complex topologies since the set of differential equations becomes very complicated even for moderately sized circuits.

In this work, we build biological parallels of two canonical constructs from control theory - the two-stage cascade and negative feedback networks to demonstrate the utility of our approach. We first build a model for single gene expression, and then combine stages using block diagrams to build more complicated topologies. We then derive the output noise characteristics of these circuits as a function of various parameters of the circuit. These biological circuits are then simulated stochastically using the Gillespie algorithm. The simulated noise strength, measured by the Fano factor [1], under varying parameter values is then compared with results from our analytical model.

Methods

Single Gene Expression Model. We first consider the single gene expression model with induction, and develop a block diagram based model that captures the expected noise characteristics. Several such blocks can then be composed to build bigger circuits, like cascades and feedback networks, and their noise properties analyzed. A single gene expressing a protein under the influence of an inducer is represented as Figure 1 (for an activating inducer) and Figure 2 (for a repressing inducer).

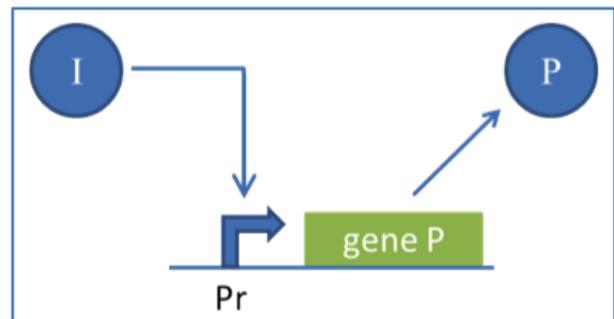


Fig. 1: Single Gene with Activating Inducer

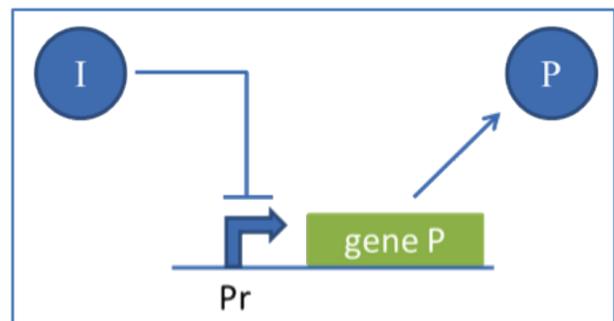


Fig. 2: Single Gene with Repressing Inducer

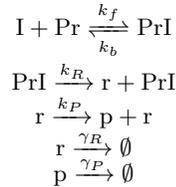
Each of these schematics can be then be transformed into a set of chemical reactions. Throughout this work, we model

Creative Commons License:

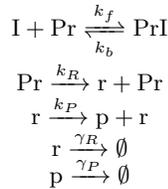
CC-BY-NC-SA 3.0

<http://creativecommons.org/licenses/by-nc-sa/3.0/>

single molecule (monomer) binding of the inducer to the promoter. Higher-order effects such as inducer dimerization and cooperative binding at the promoter are not considered. Also, in reality, an inducer typically binds to a Transcription Factor, which subsequently binds to the promoter to activate or repress expression. Here, we disregard this detail while developing our model. However, the analytical framework developed here can be extended to take such effects into account. For the case of single gene activation, we obtain the chemical reactions:



Similarly, for the case of single gene repression, we get:



where Pr is the free promoter, I is the unbound inducer, PrI is the bound promoter, r is the mRNA number, and p is the protein number. By approximating mRNA number r and protein number p as continuous quantities, a set of differential equations is obtained for each chemical system. This continuous quantity approximation is increasingly valid as mRNA number and protein number become large. We assume fast binding and unbinding of the inducer to the promoter (relative to other reactions in the system) to enable use of the quasi-steady state approximation. Lastly, as introduced in [5], we introduce noise disturbances in the mRNA and protein number due to the quantum nature intrinsic to their production and decay. These noise disturbances are modeled by Gaussian white noise sources, denoted η_R and η_P . Stochastic effects due to promoter binding and extrinsic noise are important sources of noise in genetic circuits, but are not considered in this work. Under the previous assumptions, we get the following set of stochastic differential equations for the single gene activation model:

$$\frac{dr}{dt} = Pr_T k_R \frac{(I/K_d)}{1 + (I/K_d)} - r\gamma_R + \eta_R \quad [1]$$

$$\frac{dp}{dt} = rk_P - p\gamma_P + \eta_P \quad [2]$$

Similarly, for the single gene repression model, we get:

$$\frac{dr}{dt} = Pr_T k_R \frac{1}{1 + (I/K_d)} - r\gamma_R + \eta_R \quad [3]$$

$$\frac{dp}{dt} = rk_P - p\gamma_P + \eta_P \quad [4]$$

where $K_d = k_b/k_f$ and Pr_T is the total amount of promoter, a conserved quantity. As we can see, the activator and repressor models only differ in the mRNA production rate and hence we can define effective mRNA production rates to allow the use of common expressions for activation and repression. Thus for activation,

$$k_{R,eff} = Pr_T k_R \frac{(I/K_d)}{1 + (I/K_d)}$$

and for repression,

$$k_{R,eff} = \frac{Pr_T k_R}{1 + (I/K_d)}$$

These differential equations can be Fourier transformed and written in a standard block diagram form as shown in Figure 3.

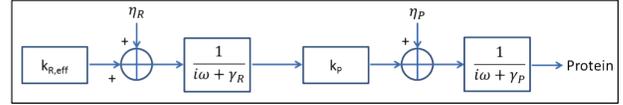


Fig. 3: Single Gene Expression Block Diagram

Looking at the set of differential equations, it is easy to see that the mean output protein is given by $\langle p \rangle = \frac{k_{R,eff} k_P}{\gamma_R \gamma_P}$. Also, it is typically assumed that mRNA is quite unstable in comparison to the output protein. The decay rate of mRNA is thus much higher than that of the output protein, $\gamma_R \gg \gamma_P$. The low-pass filtering effect of the $\frac{1}{i\omega + \gamma_R}$ block can therefore be neglected for noise analysis purposes. Noise disturbances η_R and η_P are set to be consistent with Poisson statistics at steady-state yielding:

$$\langle |\eta_R(\omega)|^2 \rangle = 2k_{R,eff}$$

$$\langle |\eta_P(\omega)|^2 \rangle = \frac{2k_{R,eff} k_P}{\gamma_R}$$

Since we are interested in protein variance at the output, transfer characteristics must be squared to yield dimensional consistency. Lastly, noise disturbances are assumed to be small-signal in nature. Therefore, the system is solved deterministically and a steady-state solution is obtained. Stochastic effects can then be evaluated using the linearized transfer function about the steady-state operating point:

$$a = \left. \frac{d\langle p \rangle}{dI} \right|_{I=\langle I \rangle}$$

This linearization technique proves inadequate for systems where steady-state is never reached (e.g., oscillators) or where noise disturbances are large enough such that the linear approximation no longer holds. The purpose of this work, however, is to develop an approximate first-order noise model for rapid parameter selection. Such techniques, although not formally correct in certain regimes, may still provide insight into systems that cannot be solved accurately by linearization of the steady-state solution. Using the above approximations, we obtain a simplified block diagram for stochastic analysis (Figure 4).

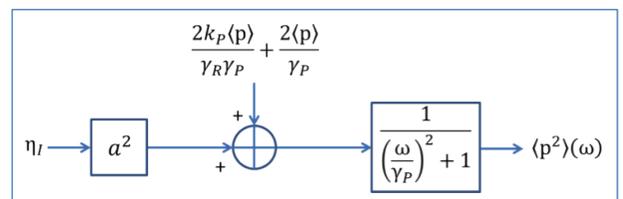


Fig. 4: Simplified Single Gene Expression Block Diagram

Output protein variance (σ_P^2) can then be evaluated by integrating noise at the output across all frequencies. The following integration identity proves useful for this purpose:

$$\int_{-\infty}^{\infty} \frac{1}{1 + \left(\frac{2\pi f}{\gamma_P}\right)^2} df = \frac{\gamma_P}{2}$$

For single gene expression, the inducer level is held constant and therefore does not contribute noise to the output protein level ($\eta_I = 0$). This is, however, not the case in more complex networks where the output protein level of one stage acts as an inducer to the subsequent stage, as in the case of cascades and feedback networks that we consider next. Output protein variance in the single gene expression case can therefore be computed:

$$\sigma_P^2 = \left(\frac{2k_P \langle p \rangle}{\gamma_P \gamma_R} + \frac{2 \langle p \rangle}{\gamma_P} \right) \int_{-\infty}^{\infty} \frac{1}{1 + \left(\frac{2\pi f}{\gamma_P}\right)^2} df = \langle p \rangle \left(1 + \frac{k_P}{\gamma_R} \right)$$

Then, we get the Fano factor as

$$FF = \frac{\sigma_P^2}{\langle p \rangle} = 1 + \frac{k_P}{\gamma_R}$$

This result is the same as obtained by [5]. Intuitively, we see for single gene expression that noise strength at the output protein is dependent only on the average number of proteins produced per mRNA transcript. Thus for a given mean protein level, noise strength at the output is reduced by using larger mRNA production rates (k_R) and smaller protein production rates (k_P), but at the cost of being more energy intensive. With a reduced model for single gene expression developed, we now consider more complex biological networks.

Two-stage cascade. A two-stage cascade is formed by linking together two single gene expression circuits. For this work, we consider the output protein of the first stage to act as an activating inducer to the second stage, as in Figure 5.

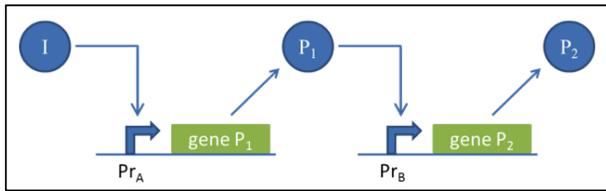
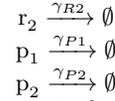
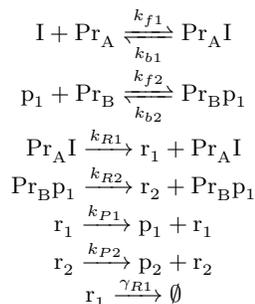


Fig. 5: Two stage cascade

The system is described by the following set of chemical reactions:



We define an effective decay rate for P_1 , denoted $\gamma_{P1,eff}$, since there is no decay reaction for inducer bound to the promoter. Thus, the effective P_1 decay rate is simply the ratio of bound P_1 to total P_1 multiplied by the original decay rate. Assuming fast binding and unbinding (relative to other reactions), the effective P_1 decay rate then is:

$$\gamma_{P1,eff} = \gamma_{P1} \frac{1}{1 + \left(\frac{Pr_{B,T}}{p_1 + K_{d2}} \right)}$$

If we assume $(p_1 + K_{d2}) \gg Pr_{B,T}$, a likely assumption for forward-biased protein reactions as $t \rightarrow \infty$, then loading effects due to Pr_B are negligible and $\gamma_{P1,eff} \approx \gamma_{P1}$. Protein levels at steady-state can then be evaluated:

$$\langle p_1 \rangle = \frac{Pr_{A,T} k_{R1} k_{P1}}{\gamma_{R1} \gamma_{P1}} \frac{(I/K_{d1})}{(I/K_{d1}) + 1}$$

$$\langle p_2 \rangle = \frac{Pr_{B,T} k_{R2} k_{P2}}{\gamma_{R2} \gamma_{P2}} \frac{(\langle p_1 \rangle / K_{d2})}{(\langle p_1 \rangle / K_{d2}) + 1}$$

Assuming the input inducer is held constant and therefore does not contribute noise at the input (i.e., $\eta_I = 0$), the stochastic block diagram model for the two-stage cascade is reduced to Figure 6.

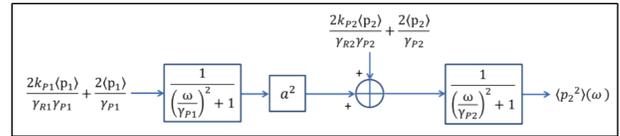


Fig. 6: Two stage cascade Block Diagram

Recall that a is obtained by linearization about the steady-state operating point:

$$a = \left. \frac{d \langle p_2 \rangle}{d p_1} \right|_{p_1 = \langle p_1 \rangle} = \frac{Pr_{B,T} k_{R2} k_{P2}}{\gamma_{R2} \gamma_{P2}} \frac{K_{d2}}{(\langle p_1 \rangle + K_{d2})^2}$$

Output protein variance is evaluated by integrating noise at the output across all frequencies:

$$\begin{aligned} \sigma_{P2}^2 &= \frac{2a^2 \langle p_1 \rangle}{\gamma_{P1}} \left(1 + \frac{k_{P1}}{\gamma_{R1}} \right) \int_{-\infty}^{\infty} \frac{1}{1 + \left(\frac{2\pi f}{\gamma_{P1}}\right)^2} \frac{1}{1 + \left(\frac{2\pi f}{\gamma_{P2}}\right)^2} df \\ &\quad + \frac{2 \langle p_2 \rangle}{\gamma_{P2}} \left(1 + \frac{k_{P2}}{\gamma_{R2}} \right) \int_{-\infty}^{\infty} \frac{1}{1 + \left(\frac{2\pi f}{\gamma_{P2}}\right)^2} df \end{aligned}$$

which can be solved exactly to give:

$$\sigma_{P2}^2 = \langle p_1 \rangle a^2 \gamma_{P2} \left(1 + \frac{k_{P1}}{\gamma_{R1}} \right) \frac{1}{\gamma_{P1} + \gamma_{P2}} + \langle p_2 \rangle \left(1 + \frac{k_{P2}}{\gamma_{R2}} \right)$$

The above expression, along with steady-state protein levels, allows the noise strength (i.e., Fano factor) of the output protein to be rapidly computed as a function of system parameters. It clearly shows the role of each parameter in the output noise and hence can guide the designer in choosing parameter values to attain the desired noise characteristics. In this way, we can rigorously arrive at an intuition about the factors affecting noise - something that would be impossible to glean

from a set of complicated differential equations. One of the observations we can make in the two-stage cascade example is that the noise disturbance at the first stage is amplified by the a^2 block before reaching the output while the noise disturbance at the second stage is not. For noise sources of equal magnitude, if a (which depends on $\langle p_1 \rangle / K_{d2}$) is large, as happens in the transition region, we expect the noise disturbance of the first stage to be the primary contributor of noise at the output. However, its contribution is reduced in the saturated region where a is small.

Negative Feedback Network. A negative feedback network is formed by using the output protein to regulate transcription at the input of the system. For this work, we consider the output protein of the second-stage to act as a repressive inducer to the first stage, as in Figure 7.

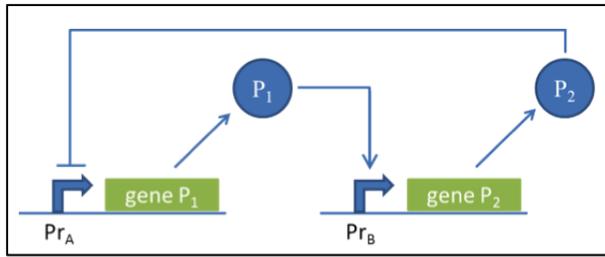
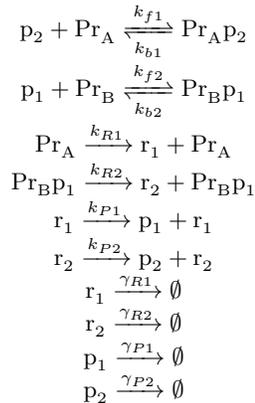


Fig. 7: Negative Feedback

This system is then described by the following set of chemical reactions:



We can build a model for this network in a way similar to the two-stage cascade. First, we assume that $(p_1 + K_{d2}) \gg Pr_{B,T}$ and $(p_2 + K_{d1}) \gg Pr_{A,T}$ such that loading effects are negligible and hence $\gamma_{P1,eff} \approx \gamma_{P1}$ and $\gamma_{P2,eff} \approx \gamma_{P2}$. Protein levels at steady-state can then be evaluated, but as interdependent expressions:

$$\begin{aligned}
 \langle p_1 \rangle &= \frac{Pr_{A,T} k_{R1} k_{P1}}{\gamma_{R1} \gamma_{P1}} \frac{1}{(\langle p_2 \rangle / K_{d1}) + 1} \\
 \langle p_2 \rangle &= \frac{Pr_{B,T} k_{R2} k_{P2}}{\gamma_{R2} \gamma_{P2}} \frac{\langle p_1 \rangle / K_{d2}}{(\langle p_1 \rangle / K_{d2}) + 1}
 \end{aligned}$$

Intermediate variables are defined as follows for notational simplicity:

$$\chi_1 = \frac{Pr_{A,T} k_{R1} k_{P1}}{\gamma_{R1} \gamma_{P1}}$$

$$\chi_2 = \frac{Pr_{B,T} k_{R2} k_{P2}}{\gamma_{R2} \gamma_{P2}}$$

We can get the mean protein output levels then by solving explicitly through substitution:

$$\begin{aligned}
 \langle p_2 \rangle &= \frac{\sqrt{\chi_1^2 K_{d1}^2 + 2\chi_1 K_{d2}^2 K_{d2} + K_{d1}^2 K_{d2}^2 + 4K_{d1} K_{d2} \chi_1 \chi_2}}{2K_{d2}} \\
 &\dots + \frac{-\chi_1 K_{d1} - K_{d1} K_{d2}}{2K_{d2}}
 \end{aligned}$$

We can then build a block diagram for the feedback network, as shown in Figure 8.

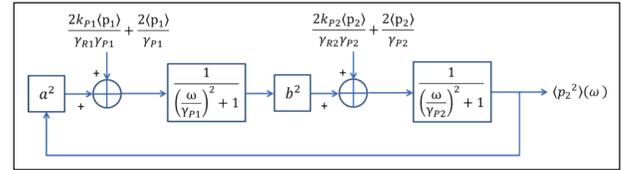


Fig. 8: Negative Feedback Block Diagram

Recall that a and b are linearized transfer functions about the steady-state operating point:

$$\begin{aligned}
 a &= \left. \frac{d\langle p_1 \rangle}{d p_2} \right|_{p_2 = \langle p_2 \rangle} = \frac{Pr_{A,T} k_{R1} k_{P1}}{\gamma_{R1} \gamma_{P1}} \frac{K_{d1}}{(\langle p_2 \rangle + K_{d1})^2} \\
 b &= \left. \frac{d\langle p_2 \rangle}{d p_1} \right|_{p_1 = \langle p_1 \rangle} = \frac{Pr_{B,T} k_{R2} k_{P2}}{\gamma_{R2} \gamma_{P2}} \frac{K_{d2}}{(\langle p_1 \rangle + K_{d2})^2}
 \end{aligned}$$

The frequency dependent output noise disturbance is found as:

$$\begin{aligned}
 \sigma_{P2}^2 &= \int_{-\infty}^{\infty} \langle p_2^2 \rangle(f) df \\
 &= \int_{-\infty}^{\infty} \left[\frac{\frac{2b^2 \langle p_1 \rangle}{\gamma_{P1}} \left(1 + \frac{k_{P1}}{\gamma_{R1}}\right) + \frac{2\langle p_2 \rangle}{\gamma_{P2}} \left(1 + \frac{k_{P2}}{\gamma_{R2}}\right)}{\left(1 + \frac{\omega^2}{\gamma_{P1}^2}\right) \left(1 + \frac{\omega^2}{\gamma_{P2}^2}\right) - \frac{a^2 b^2}{\left(1 + \frac{\omega^2}{\gamma_{P1}^2}\right) \left(1 + \frac{\omega^2}{\gamma_{P2}^2}\right)}} \right] \frac{d\omega}{2\pi}
 \end{aligned}$$

While the integral is hard to calculate manually, it can be readily solved numerically for given parameter values. Nevertheless, one can still gain an intuition about the system by inspecting the above expressions and block diagram. This is particularly useful for the feedback network, where the effect of different parameters of the system on the output noise is not understood that well.

Simulations. In order to validate our analytical model, we run simulations using Gillespie’s algorithm on our models for a wide range of parameter values to compare the noise strengths to our analytical results. This will allow us to verify that the noise strength depends on the model parameters in a way predicted by our analytical models and hence develop an intuition regarding the role of different components of the system in influencing the noise in the output. We use COPASI [2] to run our simulations. We build our networks as set of chemical reactions (as discussed above) with their kinetics determined by mass-action laws. We work with particle numbers and run stochastic simulations using the Direct Gillespie method. We typically run a very long simulation (compared to the time constants of the system) and compute noise strength only using the set of data for which the system has converged to its steady-state operating point.

Results

For each simulation, we run a sweep over certain parameter values to show the dependence of the noise strength on those parameters. The other parameters of the system, that do not change between runs, are listed in the appendix.

Single gene expression. We first look at the single gene expression model with an activating inducer. We run simulations to verify the effect of changing the protein production rate as well as the inducer level on noise strength (i.e., Fano factor). We first consider the role of the inducer by varying the inducer amount from 25 to 2000 #/ml. We observe an almost perfect match between the simulation and the analytical results as shown by the following figure (Figure 9).

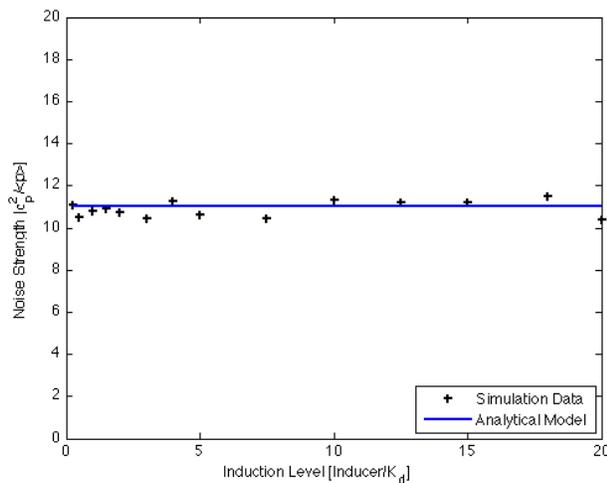


Fig. 9: Fano factor vs. Inducer Level for Single Gene Expression model with an activating Inducer

Next, for the same model, we investigate the effect of varying the protein production rate from 0.03/s to 500/s. Again,

we see a remarkable agreement between the model and the simulations, as we can see in Figure 10.

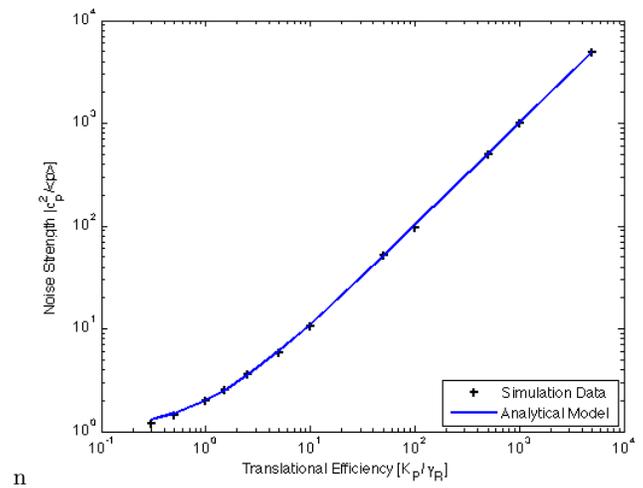


Fig. 10: Fano factor vs. Translational Efficiency for Single Gene Expression model with an activating Inducer

Also, in order to verify that the system behaves as expected with a repressing Inducer, we run parameter scans on the translation rate and the inducer level across the same ranges as above. With this, we get Figure 11 for varying the Inducer level, and Figure 12 for varying the translation rate. Again, both show an excellent match between the analytical model and simulations.

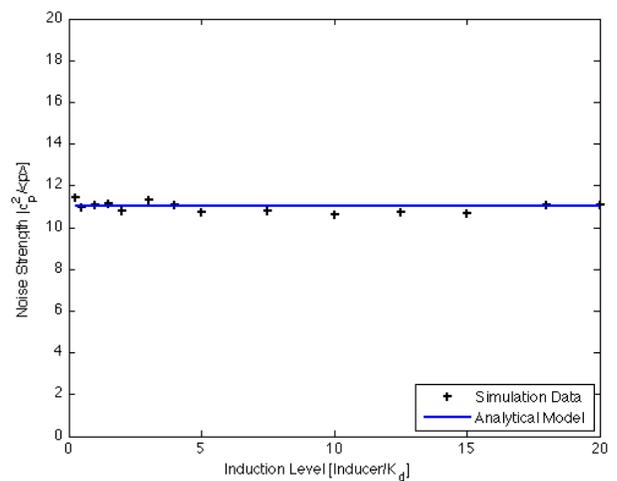


Fig. 11: Fano factor vs. Inducer Level for Single Gene Expression model with a repressing Inducer

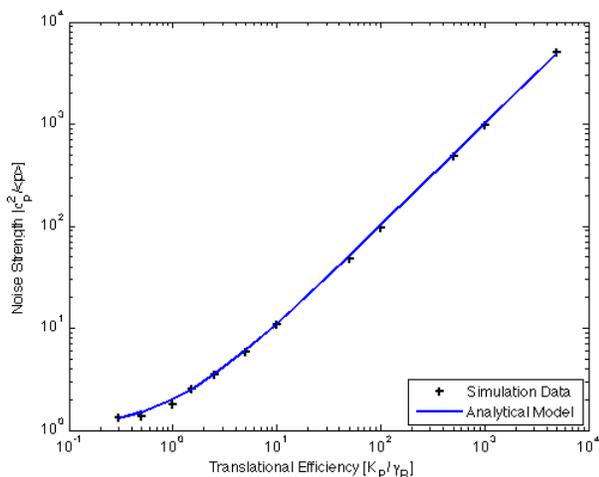


Fig. 12: Fano factor vs. Translational Efficiency for Single Gene Expression model with a repressing Inducer

Two-stage cascade. We now compare the analytical model for the two-stage cascade to simulation results to establish the correctness of our model. We vary K_{d2} and the inducer level over a wide range of parameters. Both these parameters, through their effect on a , determine whether the system is in the saturated or transition regions, and hence influence the output noise significantly. We first consider the effect of K_{d2} by varying it from 0.25 to 800 $\#/\text{ml}$. As we can see from Figure 13, the model shows a similar trend as seen in the simulations.

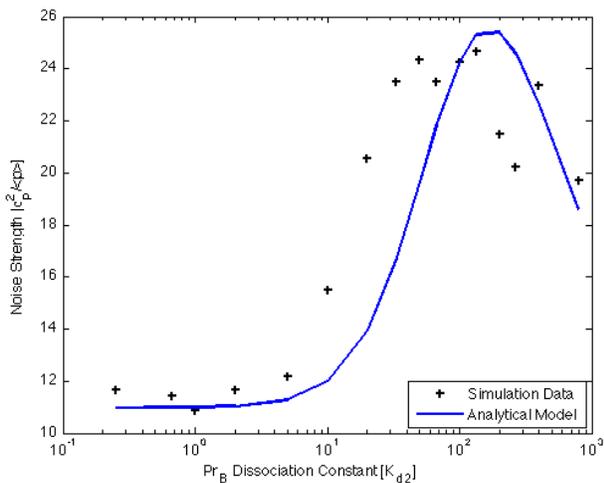


Fig. 13: Fano factor vs. K_{d2} for Two-stage Cascade

Next, we investigate the effect of varying the inducer level on the output noise, and compare that to the analytical model. We vary the inducer level from 15 to 10000 $\#/\text{ml}$ and we can see (as in Figure 14) an excellent agreement between the simulations and what we expect from our analytical model. In this way, as discussed in the Methods section, our model accurately captures the results from the simulations, but also

yields greater insight into the effect of where the system is operating (saturated or transition region) on the noise in the output.

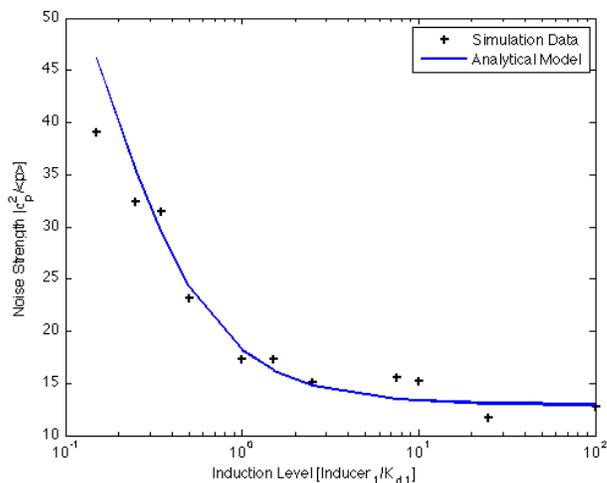


Fig. 14: Fano factor vs. Inducer Level for Two-stage Cascade

Negative Feedback network. For the negative feedback network, we vary k_{P2} (i.e., the protein production rate) of the second stage and compare our analytical model to the simulation results. We chose to vary k_{P2} from 0.01 to 2.5/s since it yields insight into how changing that parameter determines which regime the first and second stages of the feedback network are operating in, and hence how the output noise is affected. When k_{P2} is small, the first stage (the a^2 block) is in the transition region while the second (the b^2 block) is saturated, yielding low output noise as the first stage noise doesn't get propagated to the last stage. However, when k_{P2} is large, the first stage is saturated while the second stage is in the transition region. Therefore, the noise contribution from the first stage is dominant. This is exactly what we see in our simulations (Figure 15), thereby showing the our model can accurately capture how the systems works in reality.

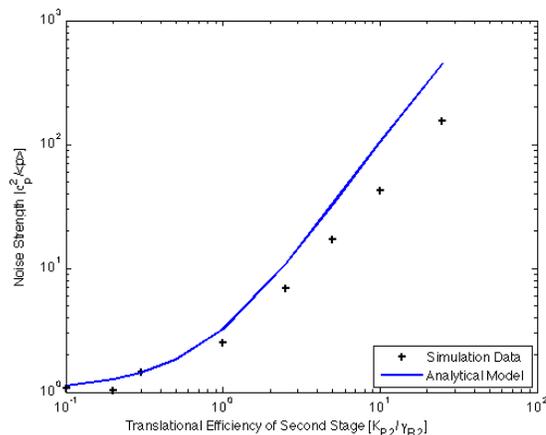


Fig. 15: Fano factor vs. Translational efficiency of the second stage for Negative Feedback network

Discussion

In this work, we have proposed an approximate model, based on concepts from control theory, that can accurately capture the noise characteristics of biological circuits. We have demonstrated the correctness of the model using two canonical constructs that are ubiquitous in biology as well as electrical engineering - the cascade and feedback network. Furthermore, as discussed earlier, our approach has the added benefit of yielding greater insight into the role of different parameters in affecting the output noise strength. We believe that this aspect enhances the usability of our approach over current techniques as it will allow synthetic biologists to logically design biological systems in order to achieve desired noise characteristics.

Nevertheless, we have shown our approach to work using a simplified model of real biology. One of the future goals is to extend the simple gene expression model to more accurately model higher order effects. At the same time, it would be interesting to apply this approach to more complicated topologies both to validate model accuracy and to develop more intuitive understanding of more complex networks.

Appendix: Parameters for Simulations

We list the parameters we use for each simulation. The parameters that we actually sweep over are discussed in the main text.

Single Gene Activation with Inducer sweep: $Pr_T = 1 \text{ \#}/\text{ml}$, $K_P = 1/s$, $\gamma_P = 0.002/s$, $\gamma_R = 0.1/s$, $k_R = 0.1/s$, $K_b = 10000/s$, $K_f = 100\text{ml}/(\text{\#s})$, Time interval = 0.2s, Simulation Time = 500000s

Single Gene Activation with Protein Production rate sweep: $Pr_T = 1 \text{ \#}/\text{ml}$, $\gamma_P = 0.002/s$, $\gamma_R = 0.1/s$, $k_R = 0.1/s$, $K_b = 10000/s$, $K_f = 100\text{ml}/(\text{\#s})$, $I = 100 \text{ \#}/\text{ml}$, Time interval = 0.2s, Simulation Time = 500000s

Single Gene Repression with Inducer sweep: $Pr_T = 1 \text{ \#}/\text{ml}$, $K_P = 1/s$, $\gamma_P = 0.002/s$, $\gamma_R = 0.1/s$, $k_R = 0.1/s$, $K_b = 10000/s$, $K_f = 100\text{ml}/(\text{\#s})$, Time interval = 0.2s, Simulation Time = 500000s

Single Gene Activation with Protein Production rate sweep: $Pr_T = 1 \text{ \#}/\text{ml}$, $\gamma_P = 0.002/s$, $\gamma_R = 0.1/s$, $k_R = 0.1/s$, $K_b = 10000/s$, $K_f = 100\text{ml}/(\text{\#s})$, $I = 100 \text{ \#}/\text{ml}$, Time interval = 0.2s, Simulation Time = 500000s

Two-stage cascade with K_{d2} sweep: $Pr_{A,T} = 1 \text{ \#}/\text{ml}$, $k_{R1} = 0.025/s$, $k_{P1} = 2/s$, $\gamma_{R1} = 0.05/s$, $\gamma_{P1} = 0.004/s$, $K_{b1} = 10000/s$, $K_{f1} = 100\text{ml}/(\text{\#s})$, $I = 50 \text{ \#}/\text{ml}$, $Pr_{B,T} = 1 \text{ \#}/\text{ml}$, $k_{R2} = 0.1/s$, $k_{P2} = 1/s$, $\gamma_{R2} = 0.1/s$, $\gamma_{P2} = 0.001/s$, Time interval = 0.2s, Simulation Time = 400000s

Two-stage cascade with Inducer sweep: $Pr_{A,T} = 1 \text{ \#}/\text{ml}$, $k_{R1} = 0.025/s$, $k_{P1} = 2/s$, $\gamma_{R1} = 0.05/s$, $\gamma_{P1} = 0.004/s$, $K_{b1} = 10000/s$, $K_{f1} = 100\text{ml}/(\text{\#s})$, $Pr_{B,T} = 1 \text{ \#}/\text{ml}$, $k_{R2} = 0.1/s$, $k_{P2} = 1/s$, $\gamma_{R2} = 0.1/s$, $\gamma_{P2} = 0.001/s$, $K_{b2} = 10000/s$, $K_{f2} = 100\text{ml}/(\text{\#s})$, Time interval = 0.2s, Simulation Time = 400000s

Negative feedback network with k_{P2} sweep: $Pr_{A,T} = 1 \text{ \#}/\text{ml}$, $k_{R1} = 0.025/s$, $k_{P1} = 5/s$, $\gamma_{R1} = 0.1/s$, $\gamma_{P1} = 0.003/s$, $K_{b1} = 10000/s$, $K_{f1} = 100\text{ml}/(\text{\#s})$, $Pr_{B,T} = 1 \text{ \#}/\text{ml}$, $k_{R2} = 0.2/s$, $\gamma_{R2} = 0.1/s$, $\gamma_{P2} = 0.001/s$, $K_{b2} = 10000/s$, $K_{f2} = 100\text{ml}/(\text{\#s})$, Time interval = 0.2s, Simulation Time = 400000s

References

1. U. Fano. Ionization yield of radiations. II. The fluctuations of the number of ions. *Physical Review*, 72(1):26–29, 1947.
2. S. Hoops, S. Sahle, R. Gauges, C. Lee, J. Pahle, N. Simus, M. Singhal, L. Xu, P. Mendes, and U. Kummer. COPASI—a COmplex Pathway Simulator. *Bioinformatics*, 22(24):3067, 2006.
3. S. Hooshangi and R. Weiss. The effect of negative feedback on noise propagation in transcriptional gene networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 16:026108, 2006.
4. H.H. McAdams and A. Arkin. It's a noisy business! Genetic regulation at the nanomolar scale. *Trends in Genetics*, 15(2):65–69, 1999.
5. E.M. Ozbudak, M. Thattai, I. Kurtser, A.D. Grossman, and A. van Oudenaarden. Regulation of noise in the expression of a single gene. *Nature Genetics*, 31(1):69–73, 2002.
6. M. Thattai and A. Van Oudenaarden. Intrinsic noise in gene regulatory networks. *Proceedings of the National Academy of Sciences of the United States of America*, 98(15):8614, 2001.

Metabolic Cost of Synthetic Circuits

Hattie Chung* and Adam Rubin*

*EECS MIT

Submitted to Principles of Synthetic Biology

As the size and complexity of in vivo synthetic biological systems continues to increase, the effect of these systems on natural host cell behavior must be taken into account. The concept of metabolic burden is used widely at a high level in metabolic engineering to assess reductions in host cell growth rate due to expression of foreign proteins. While in traditional metabolic engineering the focus is often on overexpression of a single protein, synthetic biology involves precise dynamics of many elements. It is therefore critical for synthetic biology to have a more nuanced understanding of metabolic load than merely its bulk effect on growth rate. In this work, we propose a method to quantitatively translate the function of a synthetic circuit into a metabolic load factor that has a global influence on the host cells ability to carry out gene expression. We simulate the effect of this function on two versions of a model system, one traditional circuit contained in a single cell and one multicellular circuit designed with the intention of minimizing the detrimental effects of metabolic load.

Metabolic load | Synthetic circuit | Optimization

Table 1. Factors contributing to metabolic burden.

	mRNA	Protein	
Tools	Polymerases	Ribosomes, Chaperone proteins	Synthesis
Materials	Nucleotides	Amino Acids	Accumulation

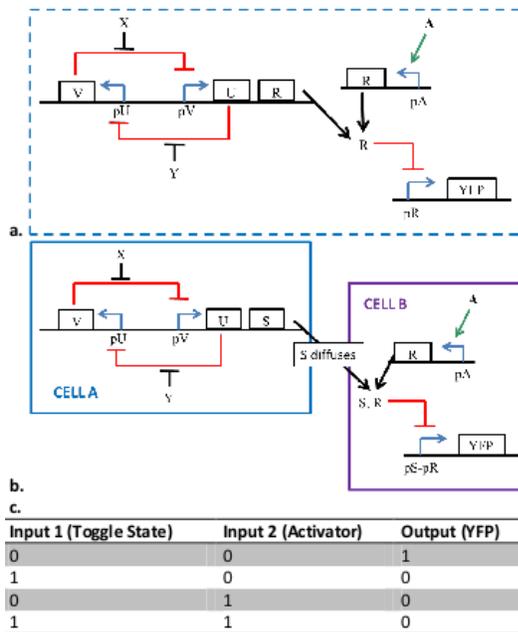


Fig. 1: Circuit models. a) Single cell NOR gate. b) Distributed NOR gate between two subpopulations. c) NOR gate truth table. Input 1 designates the toggle state, where 0 indicates that V is on, and U, R/S are off, and 1 indicates that U and R/S are on and V is off

Introduction

Introducing synthetic components into a cell creates metabolic burden as the cell adapts to synthesize something that is non-natural. Previously reported works have looked at how metabolic burden of synthetic circuits affect cell growth rate [1,2, 3]. However, there has not been a study on how metabolic burden affects the performance of the synthetic circuits themselves. Metabolic load affects the cells natural processes, such as provision of synthesis machinery, which will in turn alter the behavior of the synthetic circuit itself. As synthetic biology scales up with more promoter-gene parts, quantifying this burden will become important to understand when the predictable nature of synthetic circuits will break down.

In this study, we characterized two different types of burden introduced by synthetic genes: 1) saturation of synthesis machinery, and 2) depletion of the finite pool of synthesis materials. The use of synthesis machinery such (RNA polymerases and ribosomes) will impair the global synthesis rate for both endogenous and foreign transcripts, and the accumulation of foreign protein products represents a reduction in the finite pool of available building blocks such as nucleotides and amino acids. These factors are summarized in Table 1.

In order to test the effects of metabolic burden, we designed a novel circuit implementation of a NOR gate using the Collins toggle switch⁴ and the Voigt NOR gate (Figure 1a). This system was modeled in a deterministic fashion in MATLAB (the full ODEs are available in the appendix).

$$\text{Rate of transcription by an RNA polymerase}^8: \frac{28-89 \text{ nucleotides}}{\text{sec}} \rightarrow \text{Use } 50$$

$$\left(\frac{50 \text{ nucleotides}}{\text{sec}} \right) \cdot \frac{1}{(1083 \text{ bp})} = 0.04617 \frac{\text{mRNA}}{\text{sec}} + 50 \frac{\text{copies of mRNA}}{\text{mRNA}} = 2.3 \frac{\text{copies of mRNA}}{\text{sec}}$$

$$\text{Concentration of mRNA}^9$$

$$\frac{1380 \text{ mRNA}}{\text{cell}} \cdot \frac{1 \text{ cell}}{1 \mu\text{m}^3} \cdot \left(\frac{10^4 \mu\text{m}}{1 \text{ cm}} \right)^3 \cdot \frac{1 \text{ cm}^3}{1 \text{ mL}} \cdot \frac{1000 \text{ mL}}{1 \text{ L}} \cdot \frac{1 \text{ mol}}{6.023 \cdot 10^{23} \text{ RNA}} = 2 \mu\text{M}$$

$$\text{Typical transcription rate:}$$

$$\frac{2.3 \text{ mRNA}}{\text{sec}} \cdot \frac{60 \text{ sec}}{1 \text{ min}} \cdot \frac{2 \mu\text{M}}{1380 \text{ mRNA}} = \frac{0.2 \mu\text{M}}{\text{min}}$$

$$\text{Rate of translation by a ribosome}^{10}: \frac{12-21 \text{ AA}}{\text{sec}} \rightarrow \text{Use } 17$$

$$\left(\frac{17 \text{ AA}}{\text{sec}} \right) \cdot \frac{1}{(361 \text{ AA})} = 0.047 \frac{\text{prot}}{\text{sec}} + 60 \frac{\text{copies of prot}}{\text{mRNA}} = 2.83 \frac{\text{copies of prot}}{\text{sec}}$$

$$\text{Typical translation rate:}$$

$$\frac{2.83 \text{ prot}}{\text{sec}} \cdot \frac{60 \text{ sec}}{1 \text{ min}} \cdot \frac{1 \mu\text{M}}{500 \text{ prot}} = \frac{0.28 \mu\text{M}}{\text{min}}$$

Creative Commons License:

CC-BY-NC-SA 3.0

<http://creativecommons.org/licenses/by-nc-sa/3.0/>

Methods

We separated the effects of metabolic burden into synthesis and accumulation factors. The net burden from synthesis was calculated by summing the rate of synthesis for each species from the ODEs:
mRNA synthesis:

$$Ms_m = a_v/(1 + U^g amma) + a_u/(1 + V^b eta) + a_u/(1 + U^b eta) + kx1 * A_p A + kx2 * pR \quad [1]$$

Protein synthesis:

$$Ms_p = kl1 * Mv + kl2 * Mu + kl3 * Mr + kl4 * Myfp$$

The net burden from accumulation was calculated by summing the amount of each species:
mRNA accumulation:

$$Ma_m = Mv + Mu + Mr + Myfp$$

Protein accumulation:

$$Ma_p = V + U + R + YFP$$

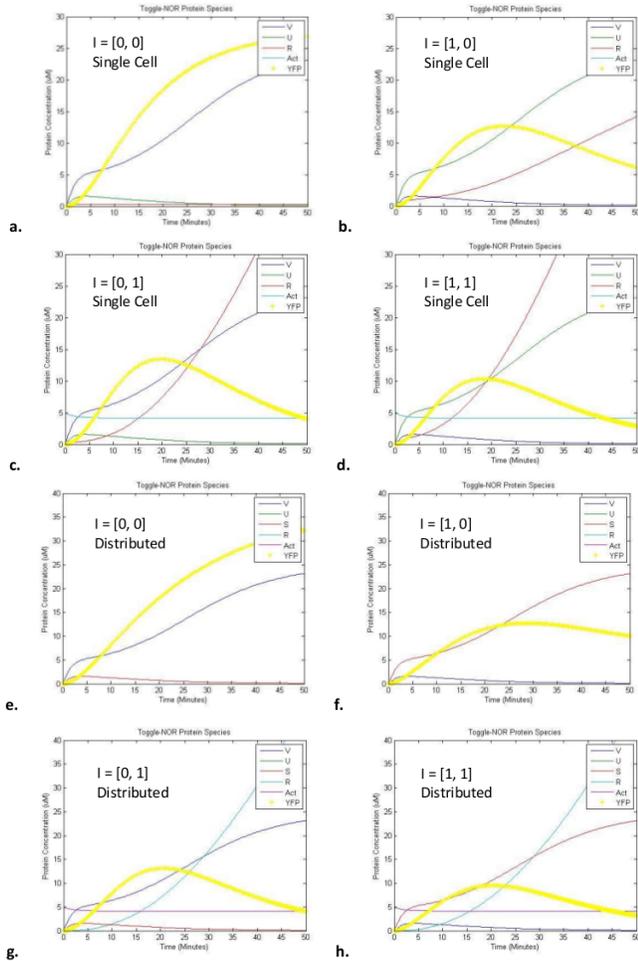


Fig. 2: Circuit function without metabolic burden. All four set of inputs were characterized for both the single cell (a-d) and distributed models (e-h). The inputs are shown by $I = [toggle\ state, activator]$, and the truth table of expected output is shown in Figure 1c.

Thus, the total burden on transcription resulting from both mRNA synthesis and mRNA accumulation is:

$$Mm = (Ms_m + Ma_m)^2$$

And the total burden on translation resulting from both protein synthesis and protein accumulation is:

$$Mp = (Ms_p + Ma_p)^2$$

The notion of increasing marginal cost (a concept typically used to describe economics) of synthesizing and sustaining foreign mRNA and protein is represented in the above equations

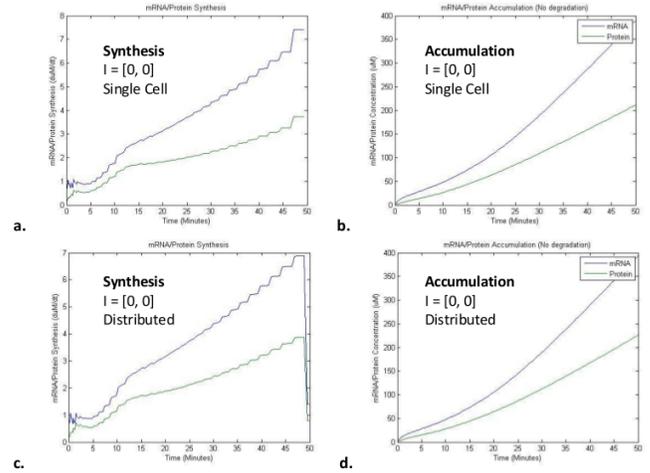


Fig. 3: Quantification of sources of metabolic burden: mRNA and protein accumulation over time. Each of the metabolic burden factors (Table 1) were plotted for the single cell (a, b) and distributed (c, d) models. Here, we only show the results for one set of inputs, $I = [0, 0]$, the only set of inputs which results in a high YFP state.

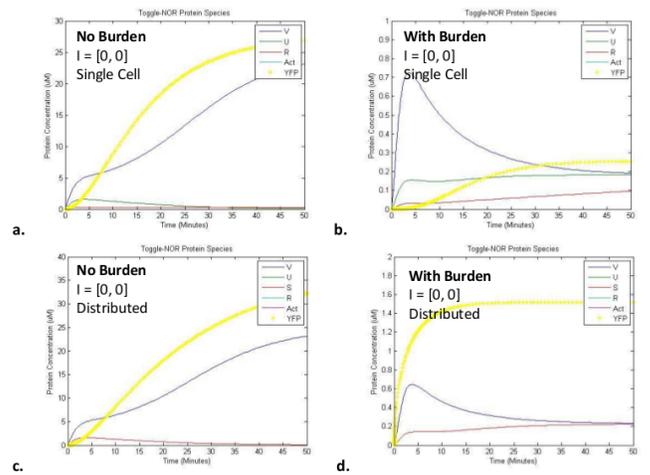


Fig. 4: Implementation of metabolic burden. Both the single cell (a, b) and distributed (c, d) model were characterized with and without metabolic burden for $I = [0, 0]$. Implementing the metabolic burden factors for synthesis and accumulation not only scaled down the concentration of each species, it also caused a qualitative breakdown of the toggle switch, as explained in the results.

by squaring the sum of synthesis and accumulation terms. That is, the cost of each additional foreign mRNA or protein (synthesized or accumulated) is greater than the cost of the last mRNA or protein because of further constraint on what we assume to be finite cellular resources. We applied these metabolic burden factors to the ODE model by inversely correlating transcription and translation rate constants with their global metabolic burden factors, M_m and M_p , respectively.

Metabolic burden from a large circuit in a single cell could be ameliorated by distributing modules of the circuit among multiple subpopulations. In order to test this prediction, we designed a second model that distributes the two modules of the NOR gate among two subpopulations, as illustrated in Figure 1b.

The MATLAB code is attached in the appendix.

A model of a cell containing the Collins toggle switch and a NOR gate was implemented in MATLAB. We used the equation from the simplified Collins toggle model⁴ for transcription, and used a mass-action rate law for translation. Toggle rate constants were obtained from a stochastic model of the Collins toggle⁶, and all other rate constants were converted to $\mu\text{M}/\text{min}$. In the distributed cell model, the output from the toggle switch, designated S , represents a diffusible transcription factor. We used a diffusible TF, which exists in plants⁷, instead of using quorum sensing molecules in order to eliminate additional variables from introducing the production of the lux synthase.

We picked LacI to use as the standard for calculating rate constants. $\text{LacI} = 1083 \text{ bp}, 361 \text{ AA}$.

Results

We implemented our models as sets of ordinary differential equations that we solved using a standard MATLAB ODE solver. Rate constants were either taken from literature or estimated based on other known parameters. We tested performance under each of four possible input sets that determine the NOR logic (Table 1) in the single cell (Figure 2 a-d) and the distributed (Figure 2 e-h) model. These results, obtained without our calculation for metabolic burden, demonstrate that our system behaves in a predictable manner in both the single cell and the distributed implementations (Figure 2).

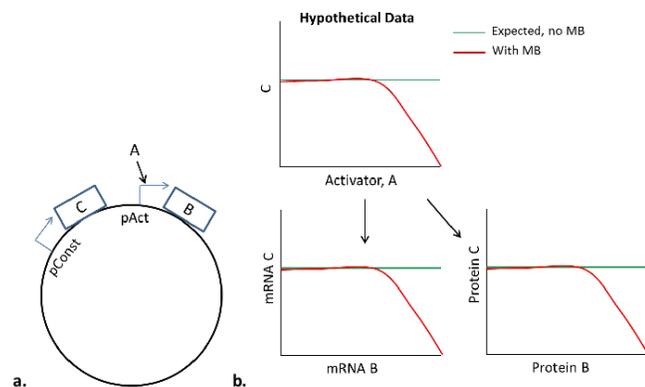


Fig. 5: Proposed experimental studies with hypothetical data to quantify metabolic burden. a. A proposed construct, where a single reporter gene C is under constitutive expression, and gene B is induced by an activator A. b. Hypothetical data to show that metabolic might affect C at both an mRNA and protein level.

In order to quantify the factors contributing to burden due to the presence of our circuit, we adjusted the model to track synthesis and accumulation of mRNA and protein. Synthesis activity was accounted for by pooling the production of all mRNA or protein species with no effect from degradation. This term reflects the total use of cellular synthesis machinery by the system. Accumulation was determined as the pool of all mRNA or protein species at any given time including depletion due to degradation. This term reflects the reduction in the pool of nucleic acid and amino acid and related elements required for mRNA and protein expression. Degradation is factored in because any nucleic acid or amino acid freed from a foreign mRNA or protein returns to the pool of materials available to the cell for its own activities. We found that total foreign synthesis increases linearly with time and that total foreign accumulation increases slightly exponentially (Figure 3).

Implementing the metabolic burden functions resulted in a significant change in system dynamics and output. As expected, the metabolic burden functions caused a general decrease in the values of all species. However the effect was not merely to scale down each species linearly. Interestingly, the metabolic burden functions caused a breakdown in toggle stability in both the single cell and distributed models. Instead of experiencing bistability in which one toggle repressor is expressed at a much higher level than its complement (Figure 4, a and c), the toggle repressors converged to a single value when metabolic burden functions were included (Figure 4, b and d). This breakdown in toggle behavior corresponded to elimination of the NOR function, as differentiating the effects of initial toggle states became impossible.

Discussion

Metabolic Burden. In this model we attempted to quantify the burden of synthetic circuits on the cells general expression activity. We took a first step towards quantifying the activity of a synthetic circuit into global changes in transcription and translation due to added constraints on cellular resources and analyzing their effect on circuit behavior. As shown in our preliminary model results, it may also be possible for metabolic burden to interfere with the function of well-characterized modules, such as the toggle switch, which works well only when its parameters are within a certain range. In a similar fashion the reduction of transcriptional and translational activity due to metabolic burden may also affect the impedance matching between an upstream part feeding into a downstream part if metabolic burden affects circuit species unequally.

Our proposed mechanism to alleviate these constraints was to distribute a circuit among multiple strains that communicate and form a consortium. In this way, the total synthetic burden on any single cell would be distributed, possibly leading to improved tolerance of the burden. However, the results from our distributed model showed the same qualitative breakdown of system dynamics, indicating that more work must be done to develop strategies to alleviate metabolic burden.

Future Model Improvements. We recognize that this work was a first step toward understanding the quantitative and qualitative effects of metabolic burden in order to allow for more accurate prediction of the behavior of a synthetic gene circuit in vivo. For future work, we have identified five critical areas of improvement to create a more realistic representation of metabolic burden. 1. Our metabolic burden functions did not account for total natural expression in order to measure

synthetic burden as a relative factor. The featured metabolic burden functions are based solely on absolute synthesis and accumulation values. A more meaningful statistic that takes into account the cells ability to tolerate foreign gene expression is the relative synthesis or accumulation in the context of total natural activity. Generally, a synthetic circuit that produces mRNA and protein at levels very small relative to natural expression levels will incur less of a burden than a circuit that represents a large portion of the cells total expression at any time.

2. Distributing a circuit across subpopulations would also require a mechanism for population control. Differential growth rates due to even small differences in metabolic load would amount to strong selective pressures against strains carrying the highest metabolic load. Recent work has featured systems to control co-culture population levels using predator-prey interactions⁵. These systems show promise but will incur additional metabolic load themselves. Therefore a tradeoff exists when implementing a distributed circuit. The metabolic cost of population control must be weighed against the benefits of reduced metabolic load on any single cell. Future work should aim to quantify the effects of each factor to determine when a single cell vs. distributed circuit may be beneficial.

3. It is critical to understand the cellular programmed response to constraints in expression inputs or machinery. The cell has natural mechanisms to sense the availability of these critical components. In some cases, the cell may respond to drastic fluctuations in these components as a signal of virus invasion, and thus initiate a global anti-virus response. On the other hand, the cell may merely compensate by increasing the production of elements with low abundance. Genetic programs can up-regulate expression of the synthesis machinery as well as the pathways responsible for supplying the nucleotide and amino acid pools.

4. Our initial model assumes that all mRNA and protein species are treated equally. This assumption ignores the fact that the activity of certain proteins such as transcription factors may also drastically influence global expression capability. More directly, it ignores the well-known issue of codon bias between foreign and host-specific genes. Organisms develop optimized nucleotide and amino acid pools to account for the average abundance of each type of building block in their own genomes expression profile. Expression of a foreign protein, for example, may quickly deplete one amino acid that is not common to the host, thus quickly initiating a response.

5. The activity of degradation machinery almost certainly affects metabolic burden. In the short-term a cells degradation machinery is more-or-less fixed. Considering that protein stability (often limited by quick degradation) plays a critical role in regulation, slowed or otherwise inhibited degradation from natural levels could easily influence expression patterns. Accounting for this natural behavior will require analysis of

degradation machinery dynamics in the presence of synthetic circuits.

Proposed Experimental Work. While many of the factors discussed above require very delicate experimental techniques, we propose a first step experiment to quantify metabolic burden with a general function that would naturally include a wide breadth of influences. By examining a single reporter gene C under constitutive expression as a second gene B is induced by activator A across a wide spectrum of activity, we would be able to take fluctuations in gene B as indication of metabolic burden (Figure 5).

While there are elements to improve in future work, the methods featured in this paper represent a first step toward quantifying the effect of metabolic load on foreign expression systems. Metabolic load factors, which are almost never accounted for in modeling, may begin to explain common differences between models and experimental results.

ACKNOWLEDGMENTS. We thank Ron Weiss, Adam Arkin, Noah Davidson and Alistair Boettiger for instruction and assistance throughout the course and project.

References

1. W Bentley et al. Plasmid-Encoded Protein: The Principal Factor in the Metabolic Burden Associated with Recombinant Bacteria. *Biotechnology and Bioengineering*, Vol. 35, pp. 668-681 (1990).
2. D Ow et al. Global transcriptional analysis of metabolic burden due to plasmid maintenance in *Escherichia coli* DH5a during batch fermentation. *Enzyme and Microbial Technology* 39 (2006). 391-398.
3. M Scott et al. Interdependence of Cell Growth and Gene Expressions: Origins and Consequences. *Science* 2010. 330:1099-1102.
4. TS Gardner et al. Construction of a genetic toggle witch in *Escherichia coli*. *Nature* 2000. 403:339-342.
4. FK Balagadde et al. A synthetic *Escherichia coli* predator-prey ecosystem. *Molecular Systems Biology* 2008. 4:187.
5. T Tian et al. Stochastic models for regulatory networks of the genetic toggle switch. *Proceedings of the National Academy of Sciences* 2006. 103(22):8372-8377.
6. X Wu et al. Modes of intercellular transcription factor movement in the *Arabidopsis* apex. *Development* 2003. 130:3735-3745.
7. Vogel U, Jensen KF. The RNA chain elongation rate in *Escherichia coli* depends on the growth rate. *J Bacteriol.* May 1994. 176(10):2807-13
8. Neidhardt F.C. *Escherichia coli* and *Salmonella*: Cellular and Molecular Biology. Vol 1. ASM Press 1996.
9. 10. Bremer, H., Dennis, P. P. (1996) Modulation of chemical composition and other parameters of the cell by growth rate.

A framework for programmable processing control in synthetic biological circuits

Arvind Thiagarajan *, and Lauren McGough *

*Massachusetts Institute of Technology, Cambridge, MA

Submitted to Principles of Synthetic Biology

Synthetic biology aims to integrate rational design principles with current knowledge about biological systems in order to engineer life forms to carry out desired functions. One major focus of synthetic biology is the design and implementation of complicated circuits from smaller, modular parts.

Synthetic biological circuits can often be described by a series of inputs upon which some computation is carried out by the cell (actuation), which then makes some internal or external change (an “output”) based on the inputs. However, one can imagine a more complicated design where the actuation carried out by the cell itself depends on other external signals or inputs to the cell; that is, the cell performs different computations depending on, for example, its environment. This is analogous to computer science, whereby an input program specifies some desired computation to be performed on a set of data.

This paper explores several possible implementations for such a system. We give two complete circuit designs for novel multicellular systems that take two inputs along with a several-input biological “program,” perform computations on the two inputs depending on the program, and output a desired protein, in this case green fluorescent protein (GFP), in response. We compare the advantages and disadvantages of the two designs, and give considerations for the biological implementation of such a system, obtained from numerical simulation. This work exemplifies the biological design of several electrical-engineering-inspired circuits, and provides an expansion of the concept of a biological circuit, indicating that the inputs to a biological system need not merely be data upon which simple computations are computed, but that they can control the actuation process itself, giving us a higher-level interpretation of cellular inputs.

Biological circuit | Decoder | Lookup table | Synthetic biology | Biological computation

Abbreviations: GFP, green fluorescent protein; T7, T7 RNA polymerase

Introduction

Systems in synthetic biology are often described using the electrical-engineering-inspired language of circuits composed of logic gates. Describing biological circuits in electrical engineering terminology is often a useful abstraction, and immediately implies the use of rational design principles, such as modularity and feedback, in synthetic biological designs.

The digital circuit paradigm often leads one to think of biological circuits as systems that take in inputs, often approximated as being in either a “HIGH” or a “LOW” state, perform some logical computations on those circuits in agreement with some truth table, and output a “value” (a chemical, for example) or a series of values based on the results of that computation. Previous researchers have had success in implementing logic circuit elements in cells; for example [2], [5]. A circuit’s output can also be more complicated, taking advantage of time delays and memory. Examples of such systems have included the “repressilator,” a biological oscillator [3] and bistable toggles that provide memory to the cell, such as the widely used Collins toggle [4]. Still, even circuits that implement circuits with higher levels of sophistication are based off

the same basic idea of inputs going through a fixed processing step that leads to an appropriate output.

However, in computer science, for example, one often works with computer programs, which take as input not only data to be processed, but also a “program” that dictates what processing will be done to the inputs. This idea of being able to easily control and change not just the inputs to a computation, but also the computation itself, is fundamental to software design and to the very nature of computers. This idea has a natural generalization to biological circuits: why not design biological circuits that have “programs,” such that some of the inputs to the circuits specify the *processing* to be carried out on other inputs to the circuit?

Such a system, a biological analogue to a computer program, could have several advantages over a more traditional fixed biological circuit. It could be more flexible and adaptable than a traditional circuit; a cell might want to respond differently to high levels of different chemicals depending on other external factors. This system could also circumvent some of the practical difficulties of implementing complicated biological circuits. Often, complicated biological circuits are difficult to implement because two seemingly-independent parts do not function as they did previously when they are arranged in series. Moreover, the number of fully characterized, orthogonal parts is often a limiting factor in physically implementing a large biological circuit. A multicellular system with programmable processing abilities might be able to overcome some of these difficulties, as different inputs could be localized, allowing cells in different regions to perform different computations, and cell-cell communication could then replace intracellular connecting of large circuits. Such a system requires fewer orthogonal parts, since it is implemented across different cell types interacting in one system, and might require less specific coordination of parts, often difficult to achieve in the variable stochastic environment of the individual cell.

As a first step toward creating such biological programs, we have designed and simulated two systems designed to take in two chemical inputs that are either HIGH or LOW. The systems then output either a HIGH level of GFP or a LOW level. The truth table that relates the inputs to the outputs is designated by a biological program. In the first system, this biological program consists of two additional chemical inputs that can be applied to the system for different amounts of time. In the second system, the biological program is implemented by means of an analogue to a lookup table in electrical engineering ([6]), where cells of different types are grown in the environments that we wish to correspond to high outputs, and then pulsed with a chemical signal that is remembered in the cell as a “write” bit. The first system can act as any

Creative Commons License:

CC-BY-NC-SA 3.0

<http://creativecommons.org/licenses/by-nc-sa/3.0/>

one of the common combinational logic gates – AND, NAND, OR, NOR, XOR, and XNOR – while the second system can implement any truth table on four inputs.

In the next section, Methods, we give detailed designs for the two systems, including high-level descriptions, circuit diagrams, and allowed inputs. We also describe methods used to numerically simulate these designs to see if they in fact act as planned, and under what conditions. In the following section, Results, we give the results of the simulations, including graphs indicating that the systems act as hoped under the correct conditions, and general design observations implied by the simulations. In the final section, Discussions, we discuss the differences between the two systems: their potential applications as well as potential shortcomings. We indicate additional concerns that would be of importance for a biological implementation of such a system, and finally, we suggest next steps along this line of developing biological systems with programmable processing.

Methods

Model Formulation. While both of our systems were designed to give processing control in a biological system, their program types and logical implementations are separate and quite different from one another. Before going into the details of each model, however, there are a few constants that should be understood. Both systems are multicellular systems; that is, they are comprised of many cells of different types that perform different computations. Both systems perform processing on two hypothesized chemical inputs, which we label *A* and *B*, each of which can be either high or low. In both cases, the biological program consists of some combination of chemical inputs being input at certain times or for certain lengths of time, though the specifics of the biological encoding depends on the individual case. Both systems are comprised of transcription networks; that is, they rely on genetic regulation for processing. In particular, this means that almost all chemicals involved are proteins that act either as activators or as repressors of some promoters. Finally, both systems use GFP as an output protein; this is essentially arbitrary, and was chosen so that researchers could easily identify and measure the output in a true biological implementation of the system.

Logic gate system

In the multicellular logic gate system, there are six cell types, each of which corresponds to a logic gate. We label these cell types as follows:

1. AND
2. NAND
3. OR
4. NOR
5. XOR
6. XNOR.

While we used the common logic gates as proof of concept, it will become clear that the system could be extended to include any truth table on two elements without too much difficulty.

A biological program then specifies which one of the six cell types will perform the computation; that is, which logic gate the system will act as. For this system, the biological programs consist of six different possibilities involving two hypothesized chemical inputs *C* and *D*, separate from *A* and *B*:

1. No program - the system acts as its default value.
2. Both *C* and *D* are in the HIGH state.

3. *C* is in the HIGH state for a long period of time while *D* remains off.
4. *D* is in the HIGH state for a long period of time while *C* remains off.
5. *C* is in the HIGH state for a short period of time while *D* remains off.
6. *D* is in the HIGH state for a short period of time while *D* remains off.

We note that these input programs are “remembered” by the cell via a toggle that switches states if the cell is exposed to the correct program. In order for such an implementation to be useful, we have also included a RESET option, a separate chemical signal that puts the toggle back to its default value (makes the cell “forget” whether or not it has seen *C* or *D* before and in what combination).

We arbitrarily assigned input programs to logic gates as follows:

1. By default, the system acts as an AND gate.
2. If the system is exposed to both *C* and *D* simultaneously, the system acts as a NAND gate until the next RESET.
3. If the system is exposed to a long “pulse” of *C* alone, it acts as an OR gate until the next RESET.
4. If the system is exposed to a long “pulse” of *D* alone, it acts as a NOR gate until the next RESET.
5. If the system is exposed to a short “pulse” of *C* alone, it acts as an XOR gate until the next RESET.
6. If the system is exposed to a short “pulse” of *D* alone, it acts as an XNOR gate until the next RESET.

We note that this does not cover all the possible combinations of *C* and *D*. For example, what if the system is exposed to a short pulse of *C* followed by a short pulse of *D*? We make the assumption that such programs are disallowed and thus never applied to the system. If they are, they will create unwanted behavior. This could be changed if such programs were to be allowed, but for simplicity we have assumed that disallowed programs are not applied.

In order to implement the systems above, we made use of several previously published biological logic gate designs (namely, [2]), with the remaining gates being designed as combinations of generic hypothesized promoters and repressors. The biological programs were implemented using these logic gates as well as toggles and, in several cases, coherent feed-forward loops. Toggles were used to give the systems memory, so that a system would change state if the RESET or the correct biological program were applied, thus causing the system to output the results of its computation or not as necessary. Coherent feed forward loops provided the timing aspects of the biological programs, as coherent feed forward loops allow cells to distinguish between short and long signals of an input ([1]). Cells that needed to respond only to long pulses of *C* or *D* were built with a feed forward loops that filtered out short signals, whereas cells that needed to respond only to short pulses of *C* or *D* were built with feed forward loops that produced a repressor. The repressor would turn on and deactivate the remaining computations as needed only if the pulse of *C* or *D* were long enough. The feed forward loop gave the system its temporal sensitivity, an important feature of this system’s biological program encoding.

The general idea behind the system is as follows. There are six cell types, each of which performs a type of computation. Initially, the AND cells are on. At first, the appropriate biological program involving *C* and *D* is applied to the cells so that the system switches to the correct state. Then, the system responds to the levels of *A* and *B* according to which gate is on at the given time, with output changing if levels of *A* and *B* change. If the system needs to switch to a new

state, the RESET signal is applied, putting the cells back to their initial states. After the RESET, the system is ready for a new biological program.

Figures 1 through 6 give detailed circuit diagrams for this system, including all promoters, activators and repressors. In some cases, specific proteins are necessary for the system's

proper functioning; we have indicated these cases. In general, though, parts are hypothesized to fit certain general requirements satisfied by most biological parts; a biological implementation of the system would then need to choose and match the appropriate parts.

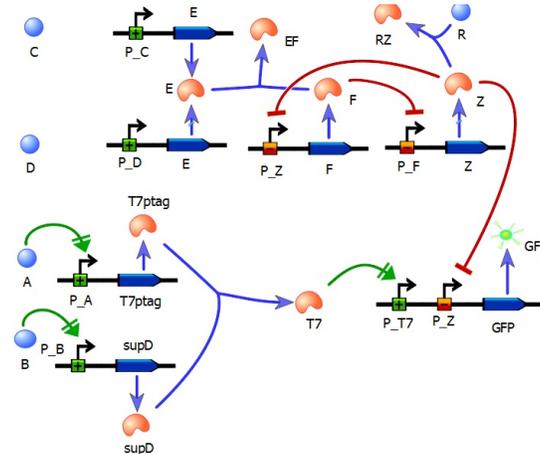


Fig. 1: The AND cell type in system 1.

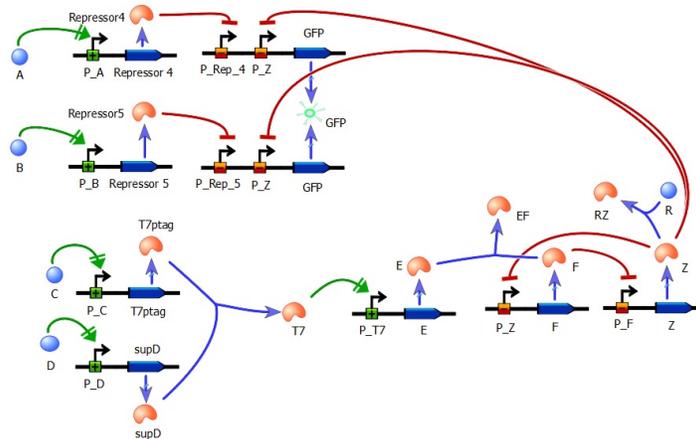


Fig. 2: The NAND cell type in system 1.

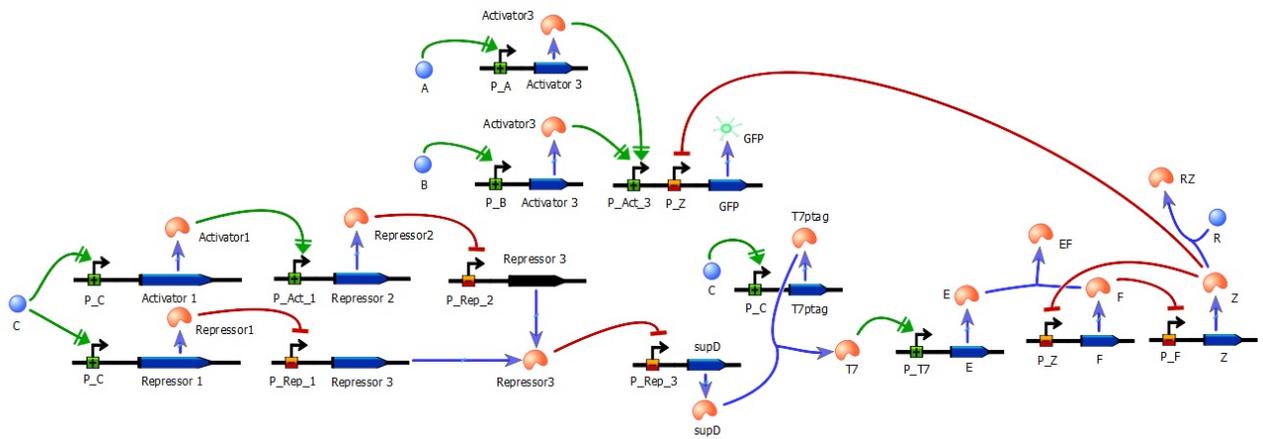


Fig. 3: The OR cell type in system 1.

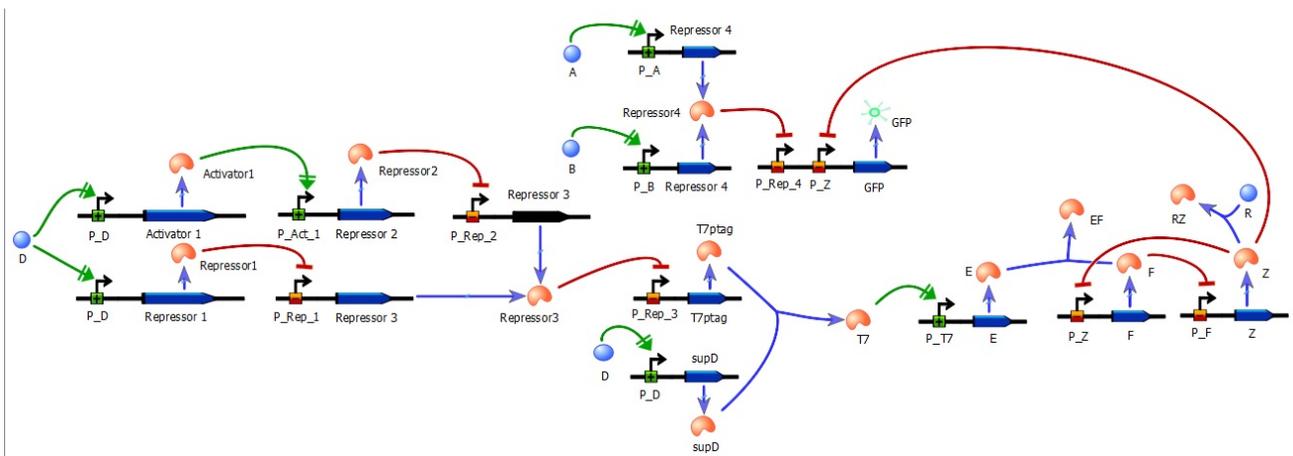


Fig. 4: The NOR cell type in system 1.

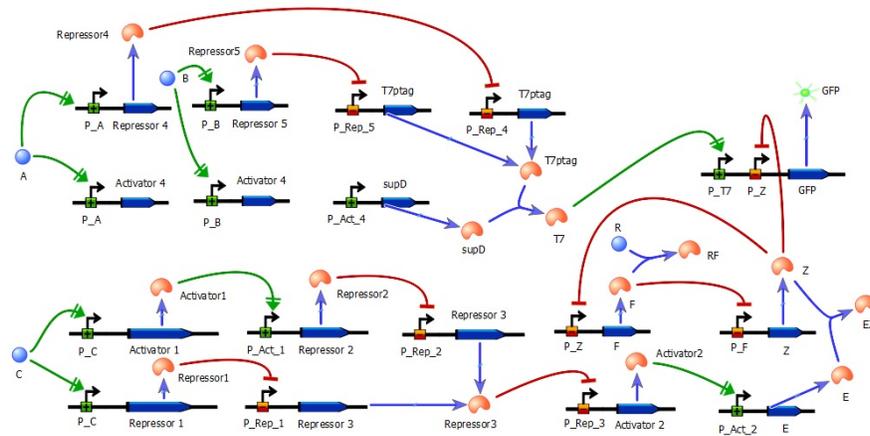


Fig. 5: The XOR cell type in system 1.

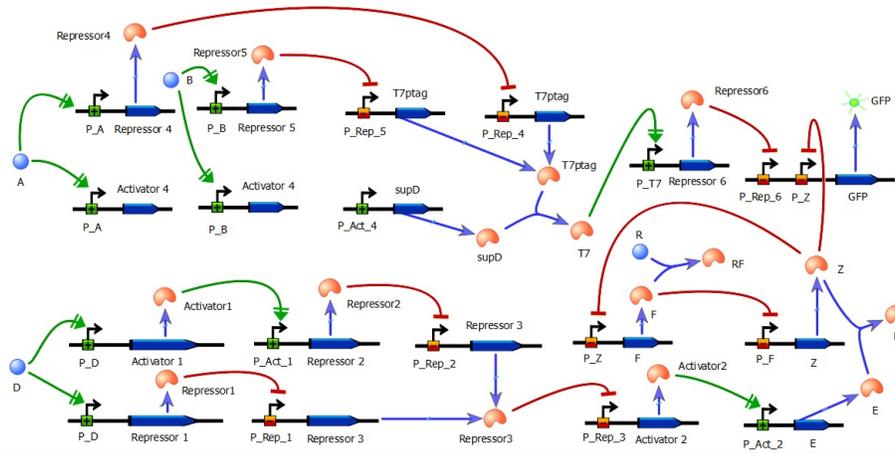


Fig. 6: The XNOR cell type in system 1.

Lookup table and decoder

The second system contains four types of cells, enumerated by which combinations of A and B to which they respond. In particular, we have the following types of cells:

1. Responds to $A = 0, B = 0$,
2. Responds to $A = 0, B = 1$,

3. Responds to $A = 1, B = 0$,
4. Responds to $A = 1, B = 1$.

Each cell “responds” in the following way. Each cell also contains a write “bit” – a protein that is either produced (write = 1) or not (write = 0). If a cell’s write bit is on and that cell is exposed to its corresponding combination of A and B levels, that cell outputs a high level of GFP. Otherwise, the cell outputs a low level of GFP.

How does one set the write bit for a cell? The cell's genetic circuitry is designed such that there is a toggle, initially in the off state. If the cell is grown in a medium containing its corresponding levels of A and B and then pulsed with a chemical, say S , then the toggle of the cell switches states: the write bit is turned on. In particular, if we want to create an OR gate, for example, this way, we can first grow the cells in high levels of A only, followed by high levels of B only and then by high levels of both A and B . In each case, we also expose the cells to the write chemical for a certain period of time. After going through this process, all cells will have write bits set to on except for cells that respond to $A = 0$, $B = 0$. We can then transfer the cells to any media with pos-

sibly changing amounts of A and B , and obtain an OR gate, since GFP output will be high unless the cells are not exposed to high levels of A or B at any given time. The write bit then gives us a way to "set 1s in our truth table"; this is why this system allows us to implement any truth table.

As in system 1, we must also implement a reset signal for this system, so that once we have set a certain truth table, we can change this truth table by first resetting the system and then implementing a new truth table.

Each cell of this system is again implemented with logic gates and a toggle. Figures 7 to 10 give detailed circuit diagrams including all promoters, repressors and activators.

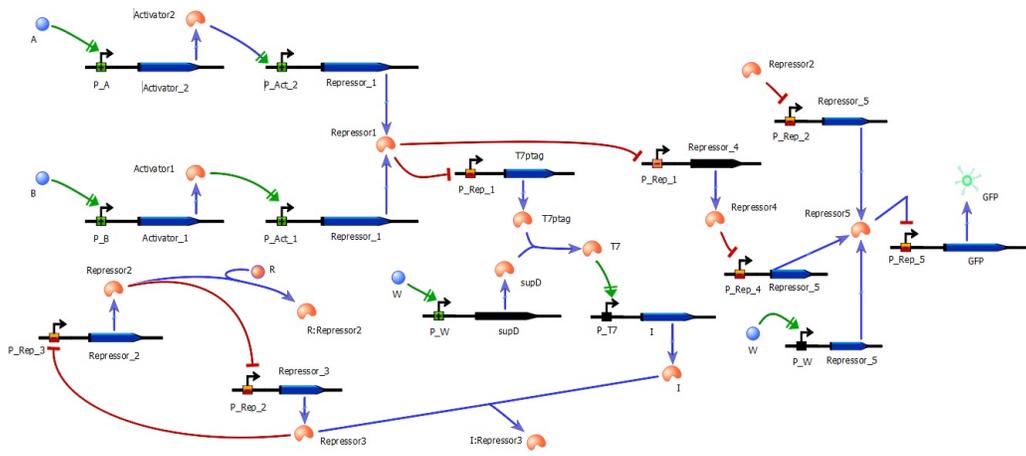


Fig. 7: Cell type 1 of system 2.

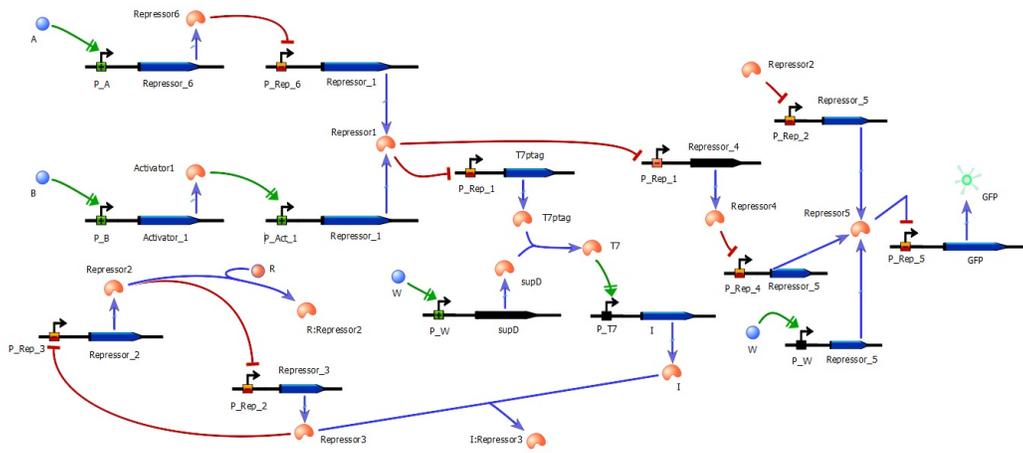


Fig. 8: Cell type 2 of system 2.

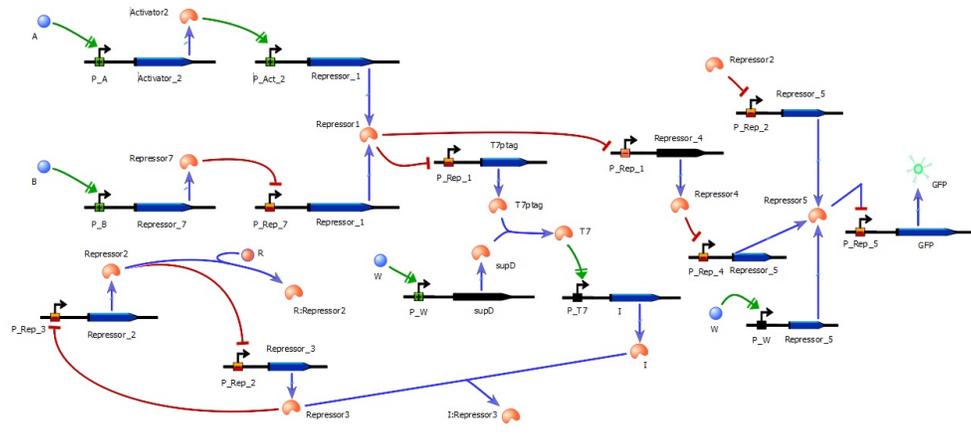


Fig. 9: Cell type 3 of system 2.

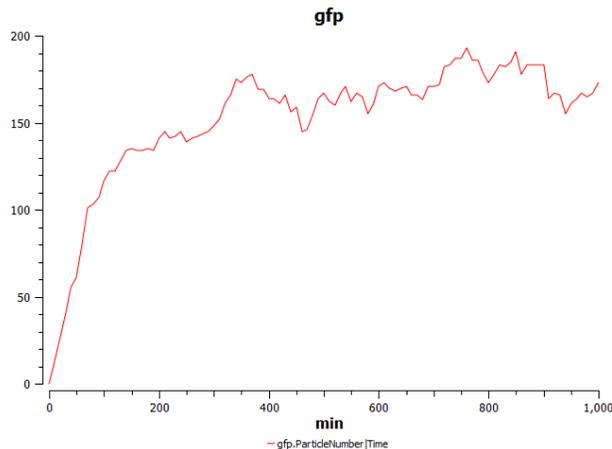


Fig. 11: GFP output of the cell becomes high after an initial delay when A and B are both high and the cell has not been exposed to C or D since the last refresh.

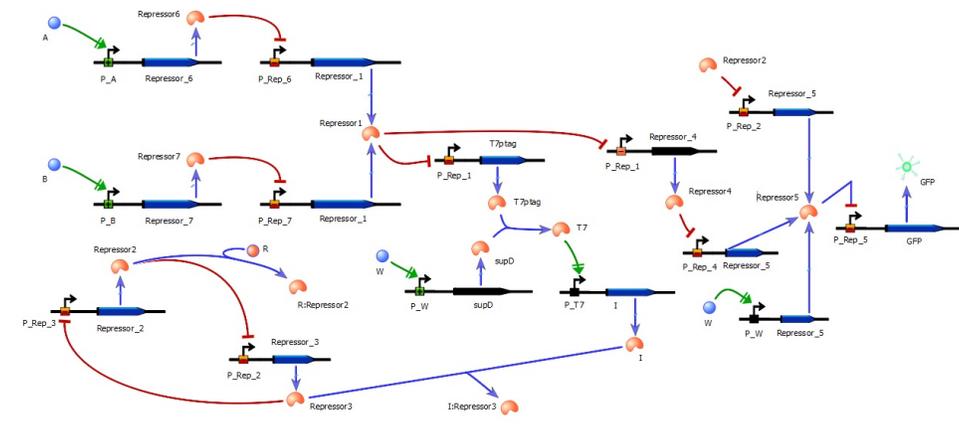


Fig. 10: Cell type 4 of system 2.

5. Cell volume: 10^{-15} L.

Simulations. We numerically simulated our systems using a Gibson-Bruck stochastic solver in COPASI to demonstrate that our systems perform as designed under biologically plausible initial conditions. All reactions were simulated using simple mass-action kinetics using mostly hypothesized parts, except where a specific part was indicated. Typical values we used were as follows:

1. Promoter binding and protein-protein complex formation: $\frac{60}{\text{nmol} \cdot \text{min}}$, $K_D = 1 \text{ nmol}$
2. Active protein production: $\frac{1}{\text{min}}$
3. Leaky protein production: $\frac{1}{10^2 \text{ min}}$
4. Protein decay: $\frac{1}{10^2 \text{ min}}$ for inactive decay, $\frac{1}{10 \text{ min}}$ for active decay

We simulated different cells separately, assuming different cells to be independent. In the case where cells compete for resources or where there is significant cell-to-cell communication this assumption may become invalid, though we did not consider these cases in our simulation.

Results

We present data that shows that even under our rather coarse assumptions, our systems demonstrated its essential property of programmable processing control. For proof of concept, we demonstrate simulations for AND and OR in the first system, and several truth tables for our second system. We tested the other cell-types with similar results not shown here.

Multi-logic gate system.

AND

In our simulations, we first tested the AND circuitry while keeping the levels of *C* and *D* at 0 to ensure that the AND circuitry worked correctly. We found that the AND circuitry worked as expected; in particular, when *A* and *B* levels were

both high (50 molecules in the cell), the cell had a high level of GFP output, as shown in Figure 11.

However, when either *C* or *D* was turned on, the cell had a low level of GFP output. This is the expected behavior since an input of *C* or *D* at any level indicates that the system should no longer be in the default, AND state. As is evident in Figures 12 and 13, after *C* or *D* has been turned on, the GFP output is low even when the input levels of *A* and *B* are both high.

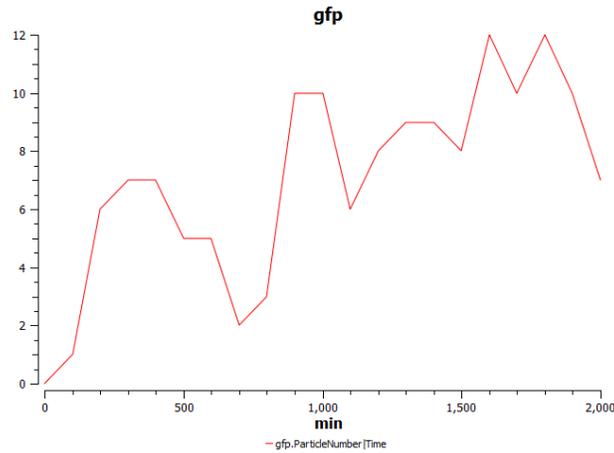


Fig. 12: GFP output of the cell remains low when *A* and *B* are both high if the cell has been exposed to *C* since the most recent reset.

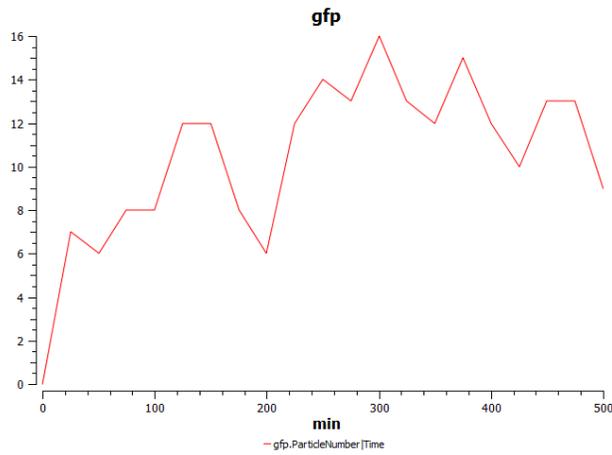


Fig. 13: GFP output of the cell remains low when A and B are both high if the cell has been exposed to D since the most recent reset.

OR

Directly following a reset, the OR cell output is low regardless of the values of A and B , as indicated in Figure 14.

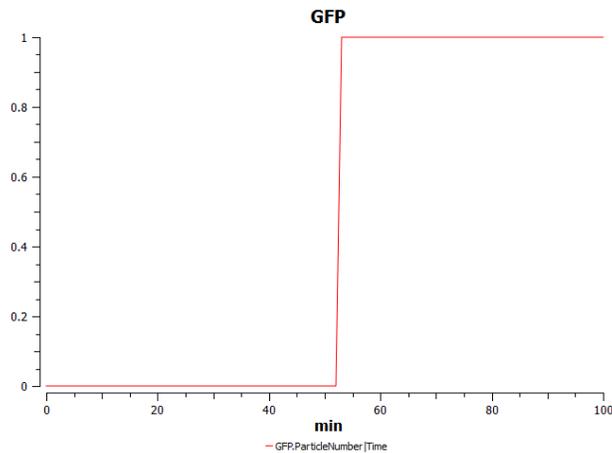


Fig. 14: Following a reset, the OR cell output is low even when A is on, for example.

However, as indicated in Figure 15, after leaving C ON for a long period of time, the OR circuitry turns on, and GFP output is high if either A or B is high.

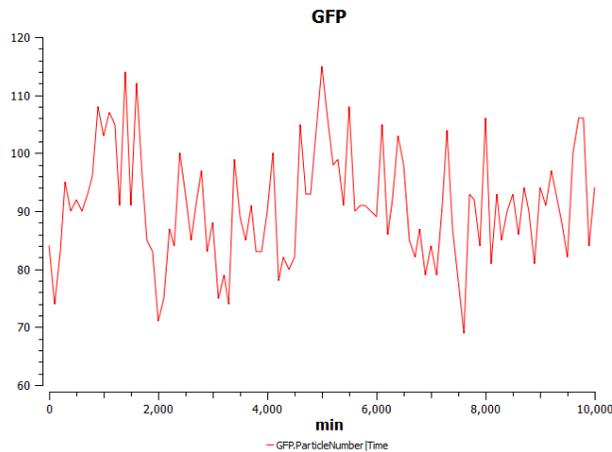


Fig. 15: GFP output of the cell becomes high according to OR logic if the cell has been exposed to C for a long period of time. Here, B is high, for example.

We also observed that the toggle does not switch states until after approximately 100 minutes. Thus for pulses of C that last for fewer than 100 minutes, the OR circuitry does not turn on. This is consistent with the design of the OR cell system, which is meant to only turn on after long pulses of C .

General themes. While ensuring that the individual subsystems worked, several general patterns necessary for obtaining desired behavior emerged.

Increase decay rate of activators

Several of these systems are extremely sensitive to certain activators; for example, the AND system is very sensitive to leaky expression of T7. In order to make up for this sensitivity, it is often necessary to increase the decay rate of activators in order to prevent leaky expression of whatever the activator controls. Biologically, this could correspond to actively degrading activators.

Use proteases for toggle control

The toggles we used generally consisted of two mutually repressing proteins, F and Z , which bind to E and R respec-

tively. Thus proteins E and R “repress the repression” of F and Z and help change the state of the toggle. One way to optimize this repression is to use proteases for E and R ; that is, make $E - F$ -binding disable the repressing action of F , and make $R - Z$ -binding disable the repressing action of Z . This allows the toggle to more efficiently switch state, and prevents the toggle from getting caught in a state where both F and Z are at low, non-zero levels of expression.

Improved circuit design prevents delays

A final general design principle that emerged while simulating was the necessity of handling delays and timing mismatches. In particular, occasionally it was necessary to add buffers in order to synchronize events and prevent incorrect levels of expression. By optimizing our circuit design, we were able to prevent hazards and ensure correct expression level (after potential unavoidable delays due to protein production).

Lookup table and decoder system.

Resetting the System.

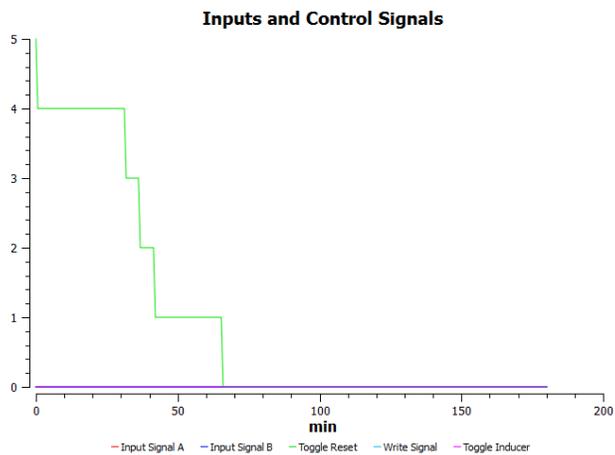


Fig. 16: The reset molecule is expressed, driving the system toggle back to the off state..

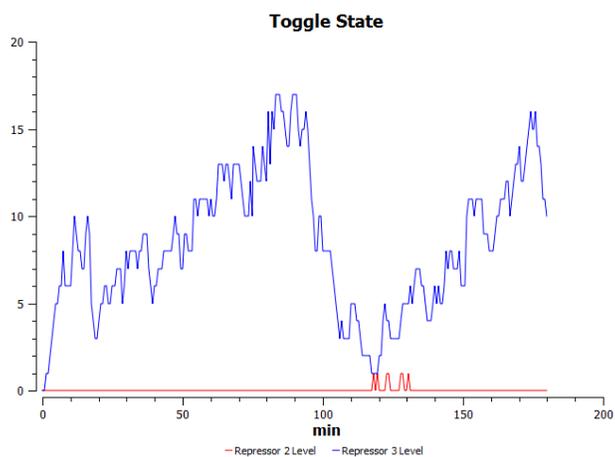


Fig. 17: The toggle has been switched to the off state, in which Repressor 3 is expressed.

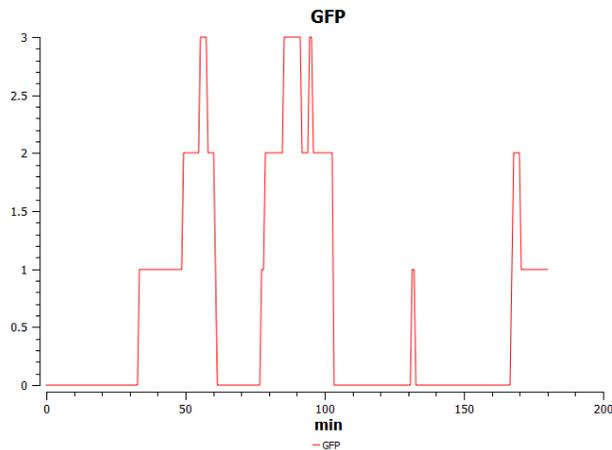


Fig. 18: Leaky GFP Output

The graphs shown here for the lookup table design correspond to the cell type that records the system's response to the input A and B. It was confirmed that the other cell types responded in a similar manner to the stimuli for which they were intended to respond, but in the interest of space the resultant graphs are not shown here. The initial images show the state of the system after the reset signal has been applied

briefly. Although only 5 molecules of the reset stimulus were applied, as shown in Fig 16, the system was successfully restored to its default state, as shown in Fig 17. The level of repressor 3 is high while the level of repressor 2 is low, indicating that the toggle switch is in the off state. Consequently, GFP expression is limited to the production caused by the stochastic variation inherent to the system, as shown in Fig 18.

Programming a Lookup Table.

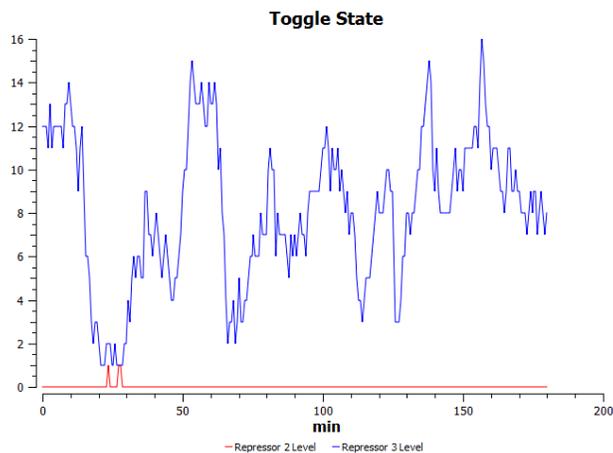


Fig. 19: Without B, the cell can neither respond properly nor be programmed.

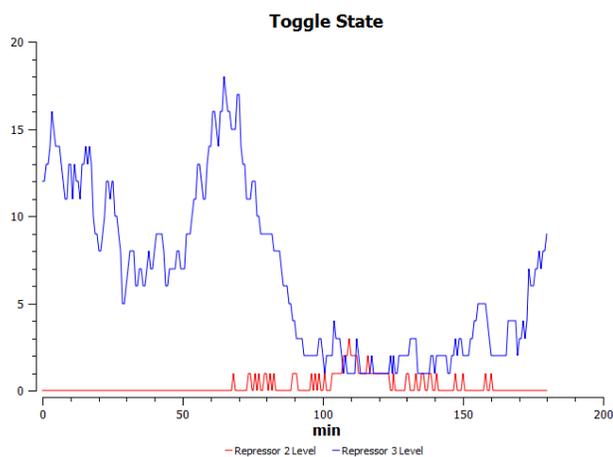


Fig. 20: Without the write signal, the cell cannot be programmed.

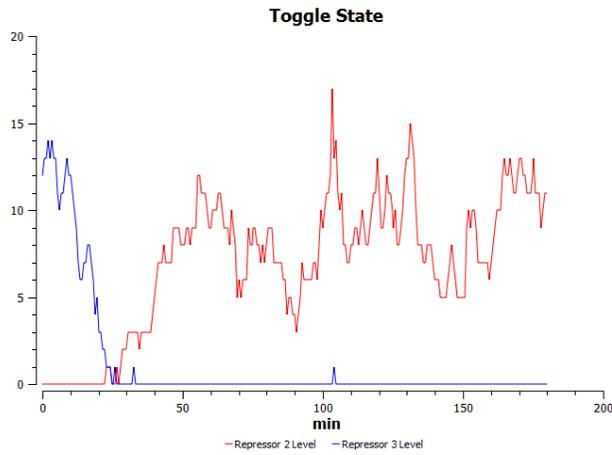


Fig. 21: In the presence of A, B, and the write signal W, the toggle switches from off to on.

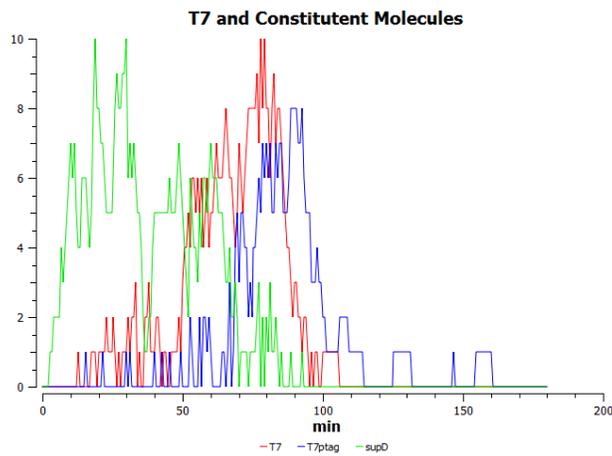


Fig. 22: In the presence of A, B, and W, T7 RNA Polymerase is produced.

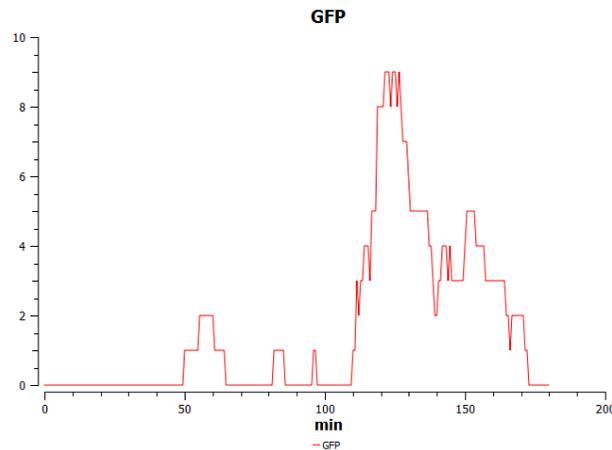


Fig. 23: GFP Production Upon Programming of Circuit

The next series of images indicates the response of the system, having been reset, to a variety of stimuli. As can be seen, the toggle is not permanently flipped unless all three input stimuli, namely A, B, and W are applied, as in Fig 21. When any one of these stimuli is missing, T7 is not formed in sufficient amounts, and so the toggle is not flipped. However, when A, B, and W are all present, T7 is formed, as in Fig 22, flipping the toggle. It should be noted that a pulse of GFP is observed in Fig 23 when this happens. This is due to the fact

that the systems response to A and B is propagated through a number of signaling layers, while the inhibitory effect of the write signal on GFP production spans only one level of the system. As such, by the time the toggle is switched on, the residual effects of A and B, namely the T7 levels, are still present, while the inhibitory write signal has long since been degraded out of the system. This is a flaw in the system design, albeit an easily corrected one: a few buffering promoters should suffice to eliminate the Boolean delay, or hazard.

Behavior of the Programmed System.

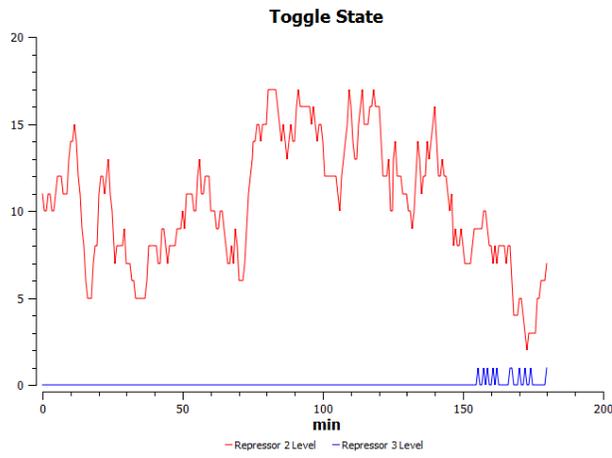


Fig. 24: Without B, the cell cannot respond properly

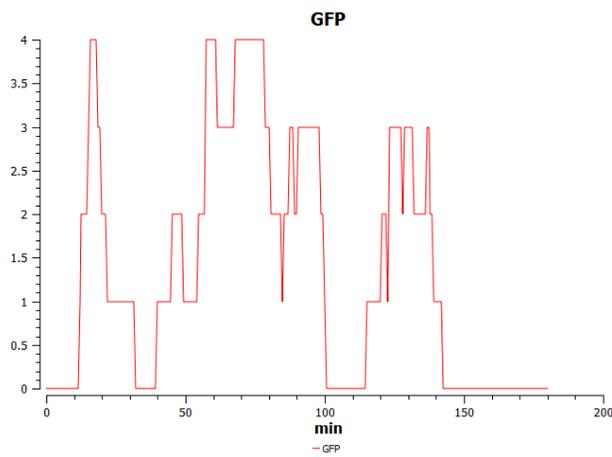


Fig. 25: Without B, the cell cannot produce GFP

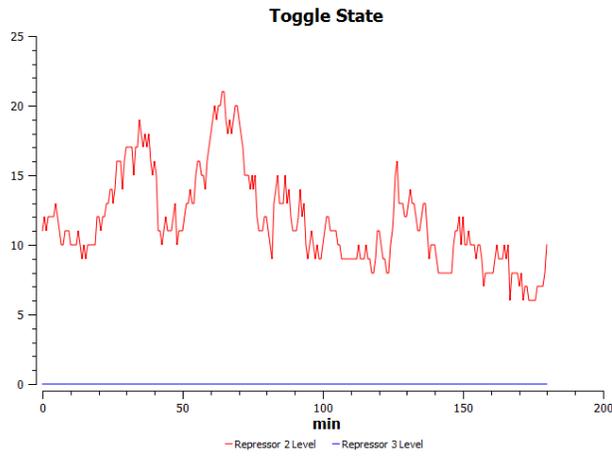


Fig. 26: In the presence of W, the cell cannot respond properly.

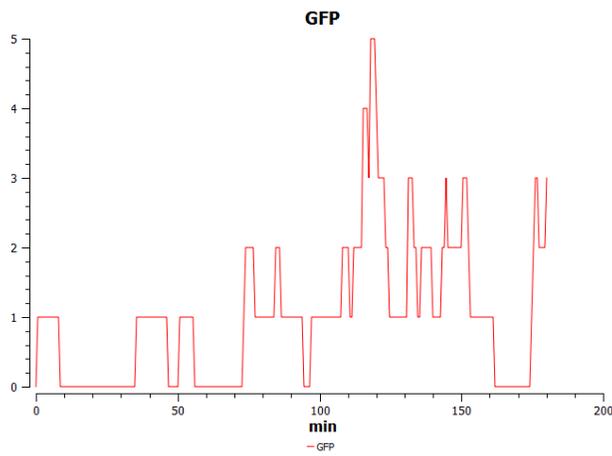


Fig. 27: In the presence of W, the cell cannot produce GFP.

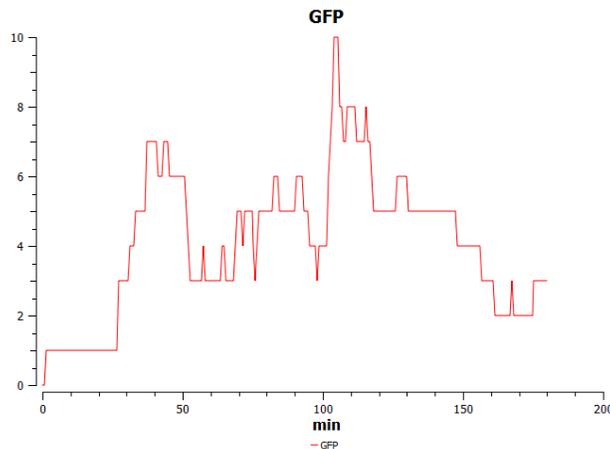


Fig. 28: The cell successfully processes the appropriate input.

Finally, once the system has been programmed to respond positively to simultaneous expression of A and B, we see that it behaves more or less as expected. The state of the toggle switch does not change, and the system responds only with leaky expression of GFP in the default case. However, in the presence of both A and B and the absence of W, GFP is expressed, as shown in Fig 28. It is worthwhile to note that GFP, when expressed in a non-leaky manner, is only twice as intense as when it is leakily expressed. This indicates that leaky expression of GFP is quite great. In modifying the system, this might be controlled in a number of ways. However, the most critical controller of the system's sensitivity is T7 production and T7 driven expression of inducer molecule. The dissociation constant for T7 binding to its promoter was increased in an effort to increase the difference in GFP intensity in the on and off states. While this measure did help to increase the difference in intensities, it is clear that they system can be improved still in this venue.

Discussion

Comparison of the two models. The two design paradigms used in this study differ in certain crucial manners. Both function as field programmable logic gates in biological contexts, of course. In the first design, multiple cell types, each encoding a pre-specified logic gate, are utilized. Each cell type further contains a memory element, a bistable toggle switch after the original toggle switch design by James Collins, with which to encode whether the system of cells is currently meant to encode the logic gate corresponding to the cell type in question. At any given time, the toggle is in the off state in the cells of all but one of the cell types. In these cells, then, an inhibitor molecule is produced in the low state of the toggle, preventing the cell from producing GFP, even in the presence of an appropriate stimulus.

In the second design, four cell types are utilized. However, instead of encoding the function of a particular logic

gate, each cell type encodes the system's response to a particular pair of binary inputs. Each cell type contains a memory element, again a bistable toggle switch, with which it encodes whether the system's response to the corresponding two bit input ought to be to express GFP or not. The system's state is specified by pulsing the inputs for which the system ought to be one while simultaneously to a write signal. GFP is then only expressed when the write signal is absent and the appropriate, previously programmed inputs are present.

In theoretical terms, both designs represent unique but valid ways of programming a logic gate. However, from a biological standpoint, each has its opposing strengths and weaknesses. The second design allows for an extremely modular design, in which the four cell types all possess identically structured synthetic genetic circuits, differing only in the identity of at most two promoters and two genes. This allows a streamlined construction of the genetic circuits involved, and consequently an extremely cost effective process for manufacturing the cell types involved. The first design also follows a modular architecture, but it is hardly surprising that the different cell types differ in more than a handful of promoters. As such, the construction of the genetic circuit for the second design is likely more difficult than the construction of the circuit for the first design. Both designs carry a significant metabolic cost, even when no GFP is observed. This is because both circuits process their inputs, even when the system response to a given set of inputs is to not produce GFP. However, since there are more cell types present in culture for the first design, it follows that the density of cells being used must be greater for the first design, and so the first design will incur a greater metabolic cost, on average, than will the second design. Finally, the second design achieves programmability with the use of four binary inputs, while the first design requires the use of two binary inputs, one binary control input, and two continuous control inputs.

It would seem, then, that the second design is preferable to the first design in terms of ease of manufacturing, expenditure of energy, and robustness to specific stimuli. Given

these advantages, the second design would clearly appear to be the design of choice. However, the first design has one very salient advantage over the second design: the first system can be programmed incredibly quickly, indeed up to four times faster than the second system. Though both circuits can be programmed on the time scale of hours, there may be situations where an immediate, timely response is required, and in this case system 1 should be utilized. Thus, while system 2 has many advantages over system 1, they both have worthy merits and each should be produced in the appropriate situation.

Notes on implementation. The systems we have modeled here do correspond, of course, to real systems that may be constructed. These systems must be prepared as follows. A well mixed solution containing all the cell types must be prepared. This solution must then be spread across the surface of top agar, or equivalently a substance that will preserve the spatial orientation of the cells that it supports. This will allow appropriate localization of the stimuli provided to the cells.

Due to the presence of multiple cell types, there is a possibility of uneven system response to different inputs. In particular, if one cell type is more susceptible to fatal environmental conditions or fluctuations, the system might not appear to perform as expected. Since each cell type was simulated separately in this study, and further since selection for and against the circuits under investigation was not considered, it is different to say how much of a concern this poses. Nonetheless, it ought to be considered while moving forward to experimental validation of these systems.

Applications and next steps. While this study serves primarily as a proof of concept, there are a number of ways in which one might envision applying programmable gates. In a very general sense, these gates can be applied in situations where it is not initially clear how to respond to a certain set of stimuli. For instance, suppose there has been a chemical spill, and while mechanisms to deal with individual chemical components of the spill are known, it is known how the responses to each chemical will affect or be affected by the other chemicals, or by the responses to the other chemicals. In this case, it will be necessary to tune the output of different response elements based on the chemical stimuli against which each response will be the most effective. Similar scenarios exist in medical settings as well.

Going forward, it is clear that expanding the framework of programmable logic functions in synthetic biology holds the key to truly constructing a biological computer. However, this expansion is far from trivial. Using the framework of a lookup table, the number of chemical input signals required to program an n input Boolean function is simply n , but the number of physical inputs that need to be provided may be as many as 2^n . Thus, although this framework scales in a very straightforward way, the complexity and time required to specify a logical function grows exponentially with the scale of input. On the other hand, as the number of inputs required for a logical function grows, the multiple gate design approach offers a simpler alternative to the lookup table design. Unfortunately, this approach does not scale easily. As such, we are limited by both time and ingenuity of design in scaling the result obtained in this study to a field programmable gate array in biology. Since the timescale on which biological events happen is extremely difficult to tamper with, however, it seems plausible that the multiple gate design will prove to be the foundational approach moving forward.

ACKNOWLEDGMENTS. We thank Ron Weiss, Adam Arkin, and Noah Davidson for support and careful reading of this manuscript, and the Berkeley and MIT pilot PoSB class of 2010 for helpful discussions. We also thank Alistair Boettiger for his LaTeX template. Finally, we would like to acknowledge COPASI, the tool that allowed us to perform our simulations.

References

1. Alon, U. *An Introduction to Systems Biology. Design Principles of Biological Circuits.* Chapman & Hall/CRC. Boca Raton, FL. 2007.
2. Anderson J-C, Voight C-A, Arkin A-P. (2007) Environmental signal integration by a modular AND gate. *Mol Syst Biol* 3:133
3. Elowitz M-B, Leibler S. (2000) A synthetic oscillatory network of transcriptional regulators. *Nature* 403(6767):335-8
4. Gardner T-S, Cantor C-R, Collins J-J (1999) Construction of a genetic toggle switch in *Escheria coli*. *Nature* [403] 339-342
5. Tamsir A, Tabor J-J, Voight C-A. (2010) Robust multicellular computing using genetically encoded NOR gates and chemical “wires.” *Nature*
6. Winiewski, R. (2009) Synthesis of compositional microprogram control units for programmable devices. Zielona Gra: University of Zielona Gra. pp. 153. ISBN 978-83-7481-293-1.

Synthetic Cell Growth Rate Speedometer

Paresh-Malulur *, On going research project of Weiss Lab

*Bioengineering Dept, Massachusetts Institute of Technology

Submitted to Principles of Synthetic Biology

Cancerous cells are typically characterized by their increased growth rates from their non-cancerous counterparts. One potential method to prevent the rapid replication of these cells is to quantify the growth rate of a cell and trigger cell death if the growth rate crosses a certain threshold.

In order to achieve this, we first need to be able to determine the growth rate of cells and distinguish between cells growing at slightly different rates. In this paper, I describe the genetic circuit that I constructed to solve this very problem.

First, I discuss the requirements of the various parts of the speedometer circuit. I then explain circuit restrictions that make it hard to directly implement the circuit and very simple and practical ways to circumvent these barriers to design a fully tunable genetic speedometer.

This paper also discusses an the impedance matching of circuit elements, a crucial design aspect of many biological circuits and a mechanism to do this through simple genetic modifications rather than through infeasible mechanisms like protein engineering.

With the speedometer circuit in place, we can start thinking about how it can be used to distinguish between cancerous and non-cancerous cells and how one can restrict cancerous cells from rapid proliferation.

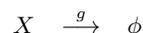
Cancer | Synthetic Biology | Difference Amplification | Growth Rate

Introduction

Cancerous cells exhibit faster growth rates than normal cells in the body. However, it is very difficult to find differences between cancerous cells and normal cells in the body. This makes it difficult to spot these cells and direct drug delivery to these cells. This lack of ability to distinguish between cancerous and non-cancerous cells results in ineffective treatments with many direct side effects.

It would be useful if we could use the fact that cancerous cells divide rapidly in order to distinguish them from normal cells in the body and then act upon this difference to trigger cell death in cancerous cells or express some genes that would help us to more effectively deliver drugs to these cells. This would help us lower the doses of the toxic treatments given to cancer patients and also reduce the number of non-cancerous cells that are killed which reduces the side effects of treatment.

Cell growth introduces a dilution effect on the concentration of all proteins in the cell. For a cell growing at rate γ , the dilution of protein X can be modeled as a degradation term represented as



Now, assuming that all other conditions are the same in a cell except for the growth rate, we can see that a higher growth rate will result in lower steady state concentration of X . Unfortunately, this difference is typically too small to directly harness into a synthetic circuit to distinguish between the growth rates.

In this paper, I construct a cell speedometer that takes as input a protein whose steady state value varies slightly based on the growth rate. Specifically, given two growth rates γ_1 and γ_2 , I try to construct a circuit that can distinguish between these growth rates by expressing YFP in the cell that divides faster (GFP is expressed in the cell that divides slower). I try to achieve this effect by designing a difference amplification module that expresses very high levels of some protein C when a cell is dividing at a fast rate γ_1 and very low levels of C when the cell is dividing at the lower growth rate γ_2 .

We will first explore how we can amplify the difference between two protein levels theoretically. We will then see some practical limitations of directly implementing the designed circuit biologically and find corresponding tricks to work around this problem to design a biologically implementable speedometer. Next, I will demonstrate a simple addition to the circuit that makes the speedometer tunable to any growth rate without having to modify the circuit for different growth rates.

With the theory explained and model set up, I will prove that the circuit works through ODE simu-

Creative Commons License:

CC-BY-NC-SA 3.0

<http://creativecommons.org/licenses/by-nc-sa/3.0/>

lations and stochastic simulations where we will see that the circuit is resilient to stochastic noise.

Finally, we will explore how we can use the cell speedometer to distinguish between cancerous cells and non-cancerous cells and discuss how one can use this to prevent cells from turning cancerous and potentially treat cancer with minimal side effects.

Amplifying the Difference. Let us assume that we are trying to build a speedometer that can distinguish between two growth rates γ_1 and γ_2 where $\gamma_1 > \gamma_2$. This would imply that the steady state level of protein $[X]_1 < [X]_2$.

Ideally, in order to amplify the difference between protein concentrations at level $[X]_1$ and $[X]_2$, we would like to find an input response transfer function that takes $[X]$ as an input and has a highly sigmoid response between $([X]_1, [X]_2)$.

Further, we would like to amplify $[X]_2$ and at the same time attenuate $[X]_1$. In other words, we would like to have $[Y]_1 < [X]_1$ while keeping $[Y]_2 > [X]_2$. In order to achieve this effect, we must choose a transfer function such that the point $[Y] = [X] = \chi$ lies in-between $[X]_1$ and $[X]_2$.

This can be demonstrated in Figure 1.

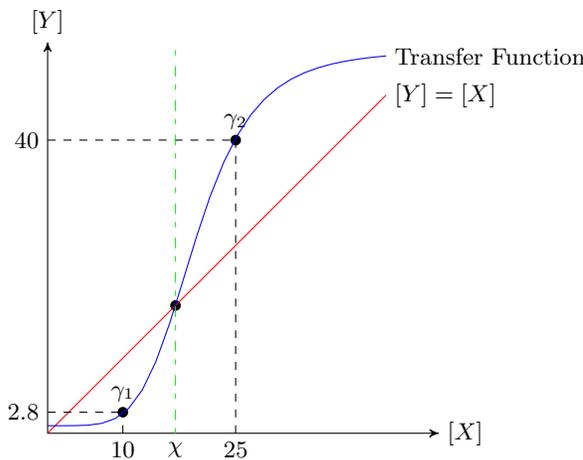


Fig. 1. A small difference in $[X]$ can be magnified to a large difference in $[Y]$ using a sigmoid transfer function

We can observe that the difference in output levels of protein Y is much larger than the difference in the concentrations of the input protein X . This can be achieved by designing a genetic circuit with X

as a transcription factor that cooperatively binds to the promoter P_Y of the DNA fragment that encodes for the gene of Y (Figure ??).

It is critical that the steady state levels of protein X , the input to the transfer function, lie near the sigmoid response region. If this is not true, we may get very ineffective or no amplification of the input difference (Figures 2, 3).

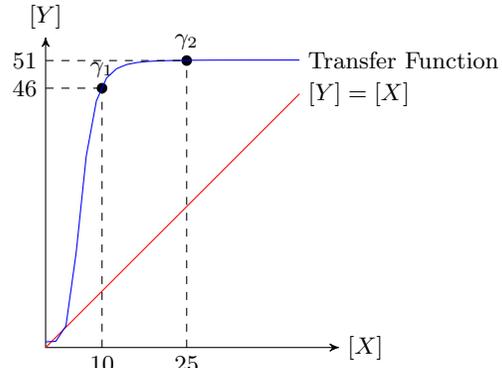


Fig. 2. If the response region is too low, we get no amplification of the difference in input concentrations

Based on the requirements and the transfer function, a single step may not be sufficient to amplify the difference even if we ensure that the protein levels are within the sigmoid response regime. This problem can be very simply overcome by cascading the difference amplification modules to create a chain of cooperatively binding transcription factors and promoters all set such that every component of

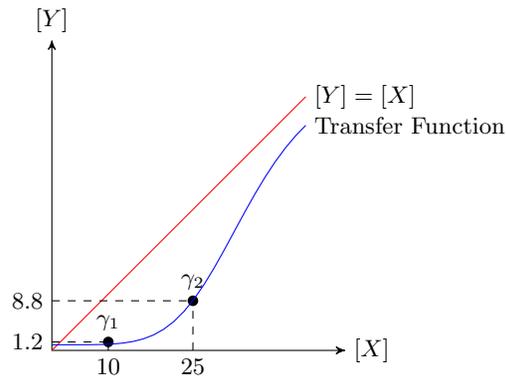


Fig. 3. If the response region is too high, we get no amplification of the difference in input concentrations

the cascade operates in the sigmoid regime. This would result in the final output of the cascade to have a very large difference between its steady state concentrations even if the input protein concentrations don't vary very much.

Genetic Circuit

Using the incite from above, we can construct a cascade $T \rightarrow L \rightarrow C$ where C is very high if the cell is diving slowly and very low if the cell is diving quickly. We can then add proteins that allow us to visualize the level of C . Here, C activates YFP and inhibits GFP thereby causing the cell to fluoresce yellow if it is growing rapidly and fluoresce green otherwise.

In the above scenario, if we were to ignore the R_0 sub-circuit and set P_T to be a constitutive promoter, then protein T would exhibit slightly different steady state concentrations at different growth rates of the cell.

If we have the magical ability to modify proteins and their interactions, then we can pick transcription factor T and promoter sequence P_L such that the cooperatively of the interaction between T and P_L results in a sigmoid response near the steady state levels of T . This would result in the amplification of the difference between the steady state levels of T at different growth rates.

By tuning the other proteins L and C and the promoters they interact with just as we did with T , then we can amplify a small percentage difference in the level of $[T]$ into orders of magnitude of difference between the levels of $[C]$ at different growth rates.

Sigmoid Response Region. A typical rate equation of a downstream protein Y regulated by a protein X with a sigmoid transfer function is given by

$$\frac{\partial[Y]}{\partial t} = \alpha_{0Y} + \frac{\alpha_Y}{1 + \left(\frac{[X]}{\beta_X}\right)^{\eta_X}} - (\gamma + \gamma_X) \cdot [X]$$

Here, α_{0Y} represents the leakage protein production rate (transcription and translation) of Y while α_Y represents the maximal protein production rate at promoter P_Y . The degradation rate of the protein is given by γ_Y while γ represents the dilution term that arises from cell growth. The parameters β_X and the hill coefficient η_X are the factors that reflect the cooperatively of the interaction of the transcription factor X with the promoter P_Y .

Let us explore how varying these parameters changes the transfer function and which parameters are practically feasible to change. Here, let us

consider two cell growth rates γ_1 and γ_2 . Let the corresponding input protein concentrations be $[X]_1$ and $[X]_2$ and let the resulting output protein concentrations be $[Y]_1$ and $[Y]_2$.

Varying α_{0Y}

As we decrease α_{0Y} , we can decrease the y -intercept of the response curve. This would give us the ability to lower $[Y]_1$.

Different promoter sequences have different basal affinities to RNA polymerase. We can hence modify the promoter sequence to obtain different values of α_{0Y} .

Further if we need to boost the leakage protein levels by a significant amount, then we can simply add another constitutive promoter with the required transcription rate. If this promoter has a transcription rate of α' , then we effectively have a transcription rate of $\alpha' + \alpha_{0Y}$.

Fig. 6. Decreasing α_{0Y} slides the transfer function down

Fig. 4. Increasing α_Y allows for a greater difference in the output for inputs in the sigmoid regime

As we increase α_Y , we increase the maximum transcription rate. This results in a sharper rise in the sigmoid response regime of the transfer function. Hence, we can obtain a greater difference between $[Y]_1$ and $[Y]_2$ for fixed $[X]_1$ and $[X]_2$.

Different promoter sequences have different maximum transcription rates. Hence, one option is to change the promoter. Optionally, one can also increase the maximal transcription rate by placing more activation sites on the operator for gene Y .

Fig. 5. Increasing β_X allows us to slide the ultra-sensitive region operate on higher concentrations of $[X]$

As we increase β_X , the sigmoid response regime of the transfer function shifts to the right. We can also observe that the response becomes less sharp. This results in less effective amplification of differences in input $[X]$.

Changing β_X typically means that we need to change the reaction constants of our proteins. This requires us to pick different proteins or engineer existing proteins, both of which are not practically feasible.

Varying η_X

Fig. 7. Increasing η_X allows us to sharpen the ultra-sensitive regime allowing for a more effective amplification of differences in concentrations of $[X]$

As we increase η_X , the sigmoid response regime of the transfer function becomes sharper without much horizontal movement of the ultra-sensitive input range.

Changing η_X can be done by changing the cooperatively of binding or by changing the feedback characteristics of the circuit. Changing the cooperatively of binding typically requires us to select different transcription factors that have different hill coefficients in their binding to DNA. This is not practically feasible due to our limited library of well categorized transcription factors.

TODO : feedback

Circumventing the Transfer Function Problem.

From section , we see that it is easy for us to modify α_{0_Y} and α_Y while it is difficult for us to modify the other parameters β_X and η_X . However, the parameter β_X allows us to shift the transfer function to slide the ultra-sensitive region to the required protein concentration range.

In this section, I will demonstrate how we can circumvent this problem and achieve the desired difference amplification by just modifying the parameters α_{0_Y} and α_Y . Let us assume that we are trying to differentiate between known, fixed growth rates γ_1 and γ_2 .

Instead of focusing on how we can more the transfer function around to match the input protein concentrations, let us instead fix the transfer function and try to control the input protein concentrations. It is important to note that if we are cascading the reactions, then we need to control both the input and output protein concentrations.

Let us recall from Section that ideally, we would like to place χ in-between the two input concentrations $[X]_1 < \chi < [X]_2$. Without loss of generality, let us assume that the values lower than χ such that $[X]_1 < [X]_2 < \chi$. Mathematically, the most direct way to move the curve is to reduce β_X until χ lies between $[X]_1$ and $[X]_2$. However, as discussed in Section , this biologically a non-trivial task.

Instead of shifting the transfer function, we are going to alter the steady state concentrations of X . This can be achieved by “boosting” the values of X such that $[X]_1^* < \chi < [X]_2^*$. This can be achieved (Section) by either altering the transcription rate of the promoter X is transcribed from or by adding an-

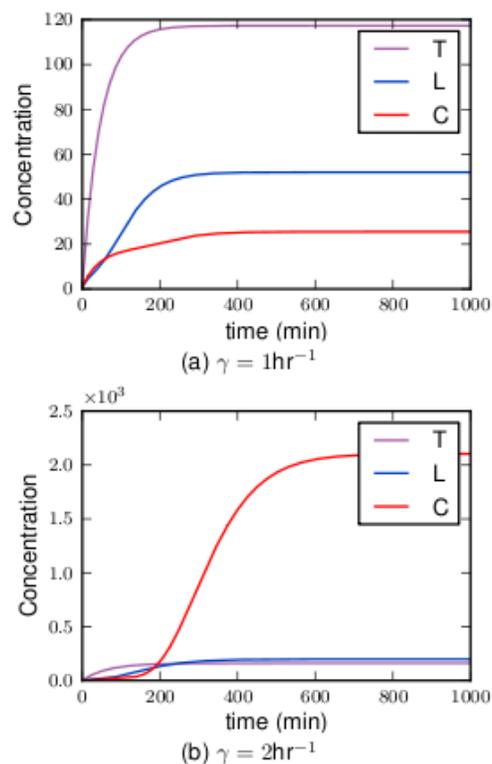


Fig. 8. ODE simulation results for difference amplification showing very low levels of C when the cell divides quickly and very high levels of C when the cell divides more slowly

other constitutive promoter with the required transcription rate.

Conversely, if we have that the steady state values of X is too high, we can decrease the transcription rates of the promoter.

With this simple trick, we can start cascading our circuit by altering the transcription rates of the i^{th} promoter to set the steady state values of transcription factor X_i to fall on either side of χ_i for the given growth rates γ_1 and γ_2

Speedometer Tunability. In the previous section, we made the assumption that the growth rates γ_1 and γ_2 were known and fixed. Hence, every time we want to distinguish between growth rates different from γ_1 and γ_2 , we need to rebuild our circuit to set the steady state values of our first protein in the cascade, T , to lie at the sensitive region of the rest of the amplification cascade. This requires us to constantly modify the transcription rate α_T .

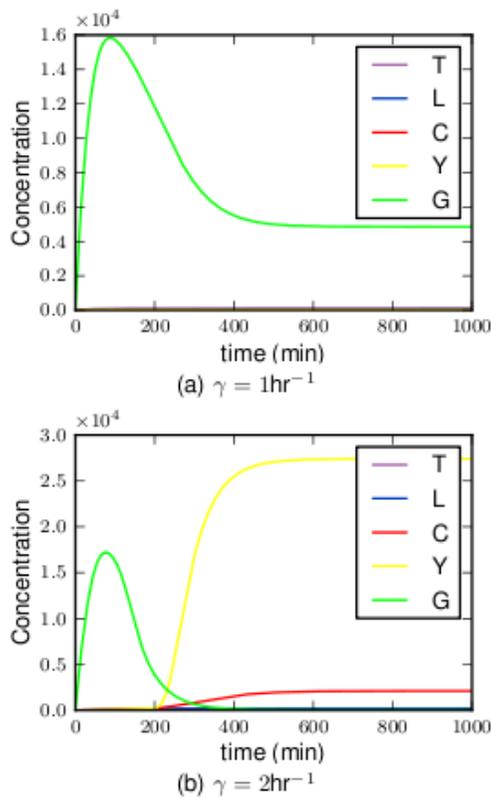


Fig. 9. ODE simulation results for difference amplification showing GFP when the cell divides quickly and YFP when the cell divides more slowly

However, this same effect can be achieved by adding a transcription factor for T , R_0 , upstream of T , whose binding affinity to the promoter region of T is affected by a small molecule S . We can pick S such that it is readily absorbed by the cell and whose concentration in the environment can be controlled in-vitro or by IV.

TODO: find aTc reference, glucose - lactose reference for LacI

Now, in order to tune the circuit at any arbitrary growth rate γ_a, γ_b , we can simply alter the concentration of S in our environment. Once S is absorbed into the cells, it alters the binding rate of R_0 to P_T and thereby changes the steady state level of T . We just need to keep changing $[S]$ until $[T]_a, [T]_b$ fall on either side of χ_{cascade} .

ODE Simulations

Let us focus on the difference amplification part of the circuit. IE, let us ignore the small molecule tuning part of the cascade and assume that we just want to distinguish between two fixed growth rates.

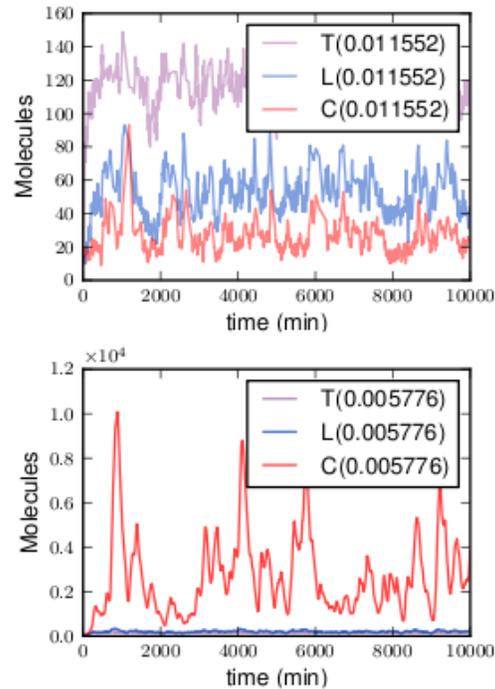


Fig. 10. In stochastic simulations of the speedometer circuit, we observe that steady state values of C is orders of magnitude higher when the cell is dividing slowly when compared to the case where the cell is dividing slower

From the circuit, we obtain the following set of differential equations representing the rate equations for the species T, L, C, G and Y .

$$\begin{aligned}\frac{\partial[T]}{\partial t} &= \alpha_T - (\gamma + \gamma_T) \cdot [T] \\ \frac{\partial[L]}{\partial t} &= \alpha_{0L} + \frac{\alpha_L}{1 + \left(\frac{[T]}{\beta_T}\right)^{\eta_T}} - (\gamma + \gamma_L) \cdot [L] \\ \frac{\partial[C]}{\partial t} &= \alpha_{0C} + \frac{\alpha_C}{1 + \left(\frac{[L]}{\beta_L}\right)^{\eta_L}} - (\gamma + \gamma_C) \cdot [C] \\ \frac{\partial[Y]}{\partial t} &= \alpha_{0Y} + \frac{\alpha_Y}{1 + \left(\frac{[C]}{\beta_C}\right)^{\eta_C}} - (\gamma + \gamma_Y) \cdot [Y] \\ \frac{\partial[G]}{\partial t} &= \alpha_{0G} + \frac{\alpha_G}{1 + \left(\frac{[C]}{\beta_C}\right)^{\eta_C}} - (\gamma + \gamma_G) \cdot [G]\end{aligned}$$

In the following simulations, I tried to use my circuit to distinguish between growth rates of 1/hr and 2/hr. In half-life units, we obtain that $\gamma_1 = 0.005776 \text{ min}^{-1}$ and $\gamma_2 = 0.011552 \text{ min}^{-1}$.

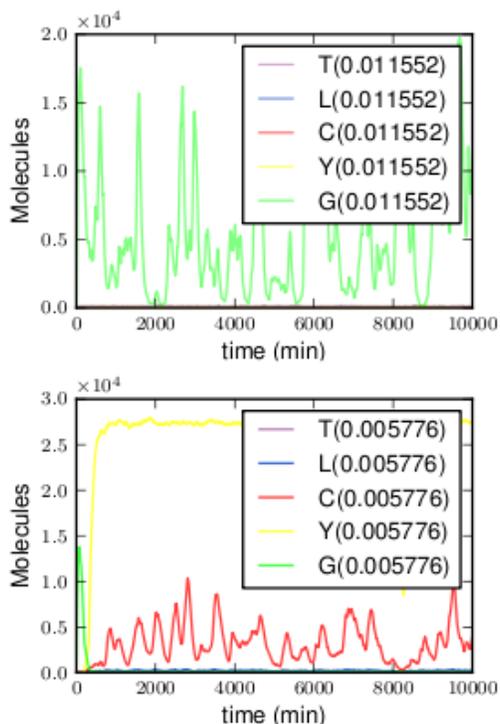


Fig. 11. We see that the fluorescent protein response is just as expected even in the stochastic model. The cell turns Yellow when it divides faster than the base growth rate and remains green otherwise.

Now, in order to demonstrate that we don't need different transfer functions that align with our protein concentrations, I have picked one transfer function and assumed that every step in the difference amplification cascade has to operate with fixed values of $\beta = 400.00$ and $\eta = -4.0615$.

For the fluorescent protein display, I chose $\beta_{CY} = 400$, $\eta_{CY} = -4.06$ and $\beta_{CG} = 20.0$, $\eta_{CG} = 4.92$. This part of the circuit is only reports the result of the difference amplification cascade. The sequence $T \rightarrow L \rightarrow C$ is the heart of the speedometer circuit. The α and α_0 values and the protein

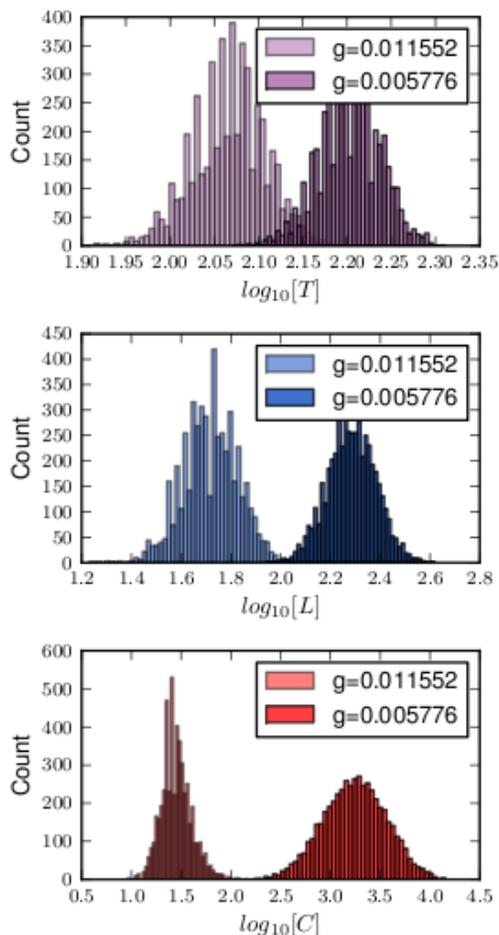


Fig. 12. Steady state distribution of protein concentrations after 1000 min over 5000 stochastic simulations. We can observe as we go from T to C , the distributions separate and thus show difference amplification

degradation rates γ were set as enumerated in table 1.

From the results of the simulation we can see that our output levels of C are much lower than the expression levels of T when the cell divides quickly and much higher than T when the cell divides slowly (Figure). Now, by adding the fluorescent protein reporting construct to the circuit, we can see how the cell turns from Green to Yellow as the cell's growth rate increases (Figure).

	α_0	α	γ (10^{-2}hr^{-1})
T	-	1.7	0.98
L	0.37	150	1.42
C	0.40	610	1.00
Y	0.76	500	1.23
G	0.76	500	1.23

Stochastic Simulations

We can simulate the speedometer circuit using Gillespie's algorithm by assuming that the binding and unbinding reactions are in steady state. This is a valid assumption since the time scale of binding and unbinding of proteins is orders of magnitude smaller than the transcription and translation of proteins.

The parameters used in the stochastic simulations were identical to those in the ODE section.

We can see from Figures that we get the expected behavior of the speedometer even in the stochastic setting. We can see that the differences in steady state levels of proteins increases as we go from T to C . The steady state difference in T of few tens of molecules (120 vs 160) to a difference of thousands of molecules in C (25 vs 3000). We also note that the reporter proteins G and Y produce the desired output of switching from green to yellow if the cell divides faster than a threshold level as in Figure .

From the results of the stochastic simulation (Figure), we see that the speedometer is very resilient to stochastic noise and continues to provide the correct reading through the noise. We obtain the expected log-normal plots for T , L and C for the different growth rates with the means growing apart, demonstrating the speedometer's ability to clearly distinguish between the two growth rates.

Discussion

If we were to introduce the above speedometer into the first few cells of a cell culture, then the cell population that grows from these parent cells will all have the speedometer machinery in them. Hence, we can potentially get direct visual feedback of which cells are actively dividing and which cells have stopped growing. Further, by tuning the circuit, we can scan different growth rates and distinguish between the cells that are most rapidly dividing and those that are not dividing quite so rapidly.

By introducing the speedometer into an embryo, we can very easily visualize which cells divide fast during embryo-genesis and which cells specialize early on in development. Further, in the developed organism, we will be easily able to spot the exact location of stem cells and actively dividing cells.

Now, if this organism were to develop cancer, we would be able to immediately spot new regions of increased growth and hence distinguish between cancerous and non-cancerous cells. This would allow for these cells to be surgically killed with a laser without having to harm any of the other cells in the body.

If we were to set the output of the circuit to be a cell surface marker, then we could get the cells to automatically express special proteins on their surface when they divide rapidly, thereby enabling us to design drugs that target these markers and hence more effectively target cancerous cells. Optionally, we could set the circuit to express a toxin or protein that triggers apoptosis of the cell. By turning the circuit on for a while and switching it off, we can get the rapidly dividing cells to be eliminated and then allowing the body to recover after the circuit is switched off.

One advantage of the tunable feature of the circuit is the ability to shut off the system by not providing the required small molecule required for the activation of the cascade. This can help make the metabolic load of the circuitry almost zero when the circuit is not required while allowing us easily turn the circuit on and off when required.

If we desire to create a circuit which responds to a fixed range of growth rates, then we can easily put in two speedometers that are tuned to different levels (or respond to different small molecules) and set the output to be a simple $A \cdot \bar{B}$ of the two speedometer circuit outputs A and B . We now can obtain all the above features for any selected range of growth rates.