Why you may want to model gene replication in stochastic models of synthetic gene circuits (and how)

Samuel Clamons, Richard Murray

Caltech, Pasadena, CA, United States

Abstract

NO ABSTRACT YET.

1. Outline

In Section 2, we briefly describe several strategies for modeling biocircuits with explicitly-tracked, replicating genes, along with a few words on the advantages and disadvantages of each.

In Section 3, we introduce the standard CRN-based dynamical modeling used to describe synthetic gene circuits, and highlight several ubiquitous modeling assumptions that will be important to consider when modeling gene replication. If you are already familiar with mathematical modeling of biocircuits, you can probably skip this section. However, considering some of the standard assumptions of biocircuit modeling may clarify some of our later modeling decisions.

In Section 4, we consider two genetic circuits for which we may want to include an explicit representation of gene replication, and we explain why.

In Section 5, we consider a few obvious, straightforward solutions to the problems highlighted in Section 4, and explain when they fail.

In Section 6, we describe several extremely simple ways to model gene replication that may not resemble natural mechanisms, but may be good enough for modeling synthetic circuits.

In Section 7, we describe two mechanistic models of gene replication based on natural mechanisms, which are more complex than those of Section 6 but should more faithfully reproduce natural replication dynamics and copy number distributions.

Finally, we give concluding remarks in Section 8.

2. Quick Summary

If you need to model the dynamics of a replicating DNA sequence D as part of a biocircuit model, you may wish to use one of the following strategies:

1. **Naïve production and dilution**
2. **Unbinding rules at cell division**
3. **Dummy-triggered replication**
4. **Delayed replication reaction**
5. **Brendel & Perelson ColE1 replication model**
6. **Simplified Brendel & Perelson ColE1 replication model**

3. Introduction to Synthetic Circuit Modeling

Synthetic biologists who design and build genetic circuits face a common engineering problem – the systems they are engineering are complex and often difficult to predict using intuition alone. Building and testing a genetic circuit is sufficiently time-consuming and expensive that synthetic biologists like to have some guarantee that it might work before they build it. At the very least, they would like to know that *if they correctly understand the parts and interactions between parts in the circuit*, that the circuit will behave as expected.

A common way to gain some confidence about the correct behavior of a circuit prior to actually building and testing it is to model its behavior mathematically. Often this modeling is done with ordinary differential equations (ODEs) describing the activity of a chemical reaction network (CRN) representing the circuit, with some simplifications for ease of use and analysis.

A CRN consists of a set of *reactions*, each of which describes how a set of reactants transforms into a set of products, along with a rate constant setting the speed of the reaction. Here's a (very much simplified) example of a CRN modeling the production of glucose via photosynthesis, followed by production of maltose from enzymatic combination of two glucose molecules:

$$6CO_2 + 6H_2O \xrightarrow{k_1} \text{Glucose} + 6O_2$$

$$2\ \text{Glucose} + \text{Amylase} \xrightarrow{k_2} \text{Maltose} + \text{Amylase}$$

From a CRN, we can use well-known rules to derive a series of ODEs describing the change in concentration over time of the species in the CRN. For this example:

$$\begin{aligned}
\frac{d[CO_2]}{dt} &= \frac{d[H_2O]}{dt} = -k_1[CO_2]^6[H_2O]^6 \\
\frac{d[O_2]}{dt} &= k_1[CO_2]^6[H_2O]^6 \\
\frac{d[\text{Glucose}]}{dt} &= k_1[CO_2]^6[H_2O]^6 - k_2[\text{Glucose}]^2[\text{Amylase}] \\
\frac{d[\text{Maltose}]}{dt} &= k_2[\text{Glucose}]^2[\text{Amylase}]
\end{aligned}$$

Assuming the CRNs these ODEs represent take place in a well-mixed system where all species are free to instantly diffuse throughout the reaction, the dynamics of the CRN will approach the ODE dynamics exactly as the concentrations of species becomes large. They can be used to mathematically prove behavior of a circuit (*e.g.*, by identifying any steady states of the system as a function of the rate constants of the CRN), or can be numerically integrated to simulate the expected behavior of the circuit.

These ODE approximations are often good enough to generate useful intuition about biological circuits, but in reality genetic circuits operate in small volumes with small numbers of molecules, where stochastic ordering and frequency of reactions can have large impacts. The impact of stochastic noise on a biological circuit can be described with a chemical master equation CITE, but in practice these are difficult to use; more frequently, modelers will employ Gillespie's stochastic simulation algorithm (SSA) CITE and related algorithms to probabilistically sample example trajectories of the CRN over time.

Synthetic biologists typically use a number of common simplifying assumptions in gene circuit models, for a number of reasons. To understand some of these assumptions, let's consider a simple, classic example of a synthetic genetic circuit – Gardner, Cantor, & Collins's genetic toggle switch CITE. This circuit consists of two repressors engineered to repress each others' expression. Intuitively, either repressor may be active at a time, but once it is, it will repress the expression of the other, locking the toggle into whichever position it began in. In their real-world formulation, Gardner *et al.*

use *LacI* and *TetR* as their repressors of choice. Without loss of generality, we will instead consider two arbitrary repressor genes G_1 and G_2.

3.1. Protein Production

What is an appropriate CRN describing the production of a protein from a gene?

The first simplifying assumption we will make is that most of the details of transcription and translation don't matter for our purposes. Instead of modeling the action of transcription factors, polymerases, ribosomes, and all their accompanying factors, cofactors, and other metabolic inputs and outputs, we will simplify transcription and translation into a single reaction each, for each gene. For our toggle switch

$$G_1 \xrightarrow{k_1^{tx}} G_1 + M_1 \tag{1}$$

$$G_2 \xrightarrow{k_2^{tx}} G_2 + M_2 \tag{2}$$

$$M_1 \xrightarrow{k_1^{tl}} M_1 + P_1 \tag{3}$$

$$M_2 \xrightarrow{k_2^{tl}} M_2 + P_2 \tag{4}$$

where M_1 and P_1 are the mRNA and protein produced by G_1, and similarly for M_2 and P_2. Here, and continuing, we will write X instead of $[X]$ for concentration of a species X in our dynamical equations, for simplicity. All of the complex dynamics of transcription have been bundled up into the two rate constants k_1^{tx} and k_2^{tx}, and all of the complex dynamics of translation have been bundled up into two rate constants k_1^{tl} and k_2^{tl}.

Yet even these four equations may be needlessly complex. A common further simplification (used by Gardner *et al.* among others) is to say that transcription and translation are essentially one process of producing protein from gene:

$$G_1 \xrightarrow{k_1} G_1 + P_1 \tag{5}$$

$$G_2 \xrightarrow{k_2} G_2 + P_2 \tag{6}$$

These two equations cannot *exactly* replicate the dynamics of P_1 and P_2 in equations 1-4, but they can get quite close, and are simpler to analyze.

Because these production reactions do not change the concentration of G_1 or G_2, we can alternatively lump the concentration of gene into the rate constant k_1, removing any explicit dependence on gene concentration:

$$\emptyset \xrightarrow{k_1^*} P_1 \tag{7}$$

$$\emptyset \xrightarrow{k_2^*} P_2 \tag{8}$$

where $k_1^* = k_1 * [G_1]$ and $k_2^* = k_2 * [G_2]$.

3.2. Repression

Assuming G_1 and G_2 act like natural bacterial repressors like *LacI* and *TetR*, they function by binding to their target genes and blocking transcription. Importantly, natural repressors typically act as dimers or tetramers. For simplicity, let's assume that P_1 and P_2 form active dimers. Then we can model our repressors binding to their targets with

$$2\,P_1 \underset{k_1^{df}}{\overset{k_1^{dr}}{\leftrightharpoons}} P_1^D \tag{9}$$

$$2\,P_2 \underset{k_2^{df}}{\overset{k_2^{dr}}{\leftrightharpoons}} P_2^D \tag{10}$$

$$P_1^D + G_2 \underset{k_1^{f}}{\overset{k_1^{r}}{\leftrightharpoons}} G_2^* \tag{11}$$

$$P_2^D + G_1 \underset{k_2^{f}}{\overset{k_2^{r}}{\leftrightharpoons}} G_1^* \tag{12}$$

Note that each binding reaction is paired with an associated unbinding reaction. These four reactions (or eight, depending on how you count) are sufficient to model repression – whenever G_1 is in its G_1^* state, there is no reaction allowing it to express P_1, and similarly for G_2.

Repression is almost never modeled this way. By far the more common approach is to abstract away the mechanisms of repression by assuming that the output of the repressed promoter with respect to its repressor follows a Hill function, i.e.

$$F(G_1) \propto \frac{K_1^{N_1}}{P_2^{N_1} + K_1^{N_1}} \tag{13}$$

$$F(G_2) \propto \frac{K_2^{N_2}}{P_1^{N_2} + K_2^{N_2}} \tag{14}$$

where $F(G)$ is the rate of expression from G, K_1 and K_2 are constants determining the concentration of repressor which halfway represses the target promoter, and N_1 and N_2 are a measure of "cooperativity," which functionally measures the "sharpness" of the response of the promoter.

The Hill function assumption hides complexity and simplifies mathematical assumption, but it is not always accurate. Hill approximation is sometimes justified mathematically by showing that for a target T and a binding molecule B, the mass action law $T + nB \underset{k_r}{\overset{k_f}{\rightleftharpoons}} TB_n$ implies that at steady state, the fraction of T bound will be $\frac{[B]^n}{[B]^n + \frac{k_r}{k_f}^n}$. This is true, as written, but sometimes overlooked is that this (exact) function is written in terms of *free* B at equilibrium, *not* the total amount of B in the system (which would include TB_n). As such, the Hill function is only an approximation of the action of repression, and only holds under the assumption that most of the repressor is not bound. This is sometimes a good approximation (*AraC* in *E. coli* (CITE), or *Pho2* in *S. cerevisciae* (CITE)) but not always (*LacI* in *E. coli*(CITE)).

FIGURE: Promoter output as a function of repressor; Hill approximation based on physiological parameters; Hill function fit. Each of these for a few different hypothetical repressors.

In practice, Hill functions are good enough at capturing the most essential feature of repression – the output of the target gene is an S-shaped function of repressor concentration, with more or less sharpness. Hill function representations of actual biological systems are determined phenomenologically, by fitting the parameters of the Hill function directly to observed data, rather than by carefully measuring dissociation constants and mechanistically-derived cooperativities. Simply fitting a Hill function to observed data is tractable and practical, but the Hill approximation still obscures the mechanistic details of repression and makes certain questions difficult to answer. For example:

- What happens to a repression curve as the concentration of target DNA changes?
- How quickly does the repressed promoter become repressed, or unrepressed? (Notice that the Hill approximation assumes instantaneous steady-state between promoter and repressor – a good assumption for many repressors, but, as we will see, not for all of them.)
- What is the rate of exchange between repressor and DNA target? (This question is intimately related to the question of response speed.)

3.3. Dilution

So far, we not included any reactions that can remove repressor proteins once their production stops. One straightforward mechanism we can use to remove proteins is to degrade them with a protein degradase enzyme, using a reaction like

$$\text{Degradase} + P_1 \xrightarrow{k_{deg}} \text{Degradase} \tag{15}$$

or, if we are comfortable assuming that the degradase is present at high, constant concentration,

$$P_1 \xrightarrow{\gamma_{P_1}} \emptyset. \tag{16}$$

where now, k_{deg} implicitly includes the concentration of degradase. In the ODE describing the dynamics of P_1, this degradation reaction creates a characteristic exponential decay term:

$$\frac{dP_1}{dt} = \ldots - \gamma_{P_1} P_1. \tag{17}$$

In a fast-growing cells, there is another mechanism that can "remove" protein – dilution. If the volume of the cell grows exponentially, then the *concentration* of a protein with constant molecular count will fall exponentially... which implies the same exponential decay term in the ODE for that species as would be created by degradation, only at a different rate γ. Typically, we will say that a CRN has "dilution reactions" of the form $X \xrightarrow{\gamma} \emptyset$ which emulate the action of continuous dilution.

Notice that the strategy of representing dilution with an elimination reaction with a chemical elimination reaction has a somewhat unnatural interpretation in the context of stochastic simulation. The implication is that molecules will randomly disappear over time. In a real cell, we would expect reaction propensities involving diluted species to drop continuously as the cell grows. In stochastic simulation, we will instead see a discontinuous drop in propensity each time a molecule is destroyed by dilution, which could lead to unnatural behavior for molecules at very concentration. A more realistic alternative is to eschew the elimination reaction and use a stochastic simulation that explicitly models volume changes over time (CITE BIOSCRAPE).

3.4. Putting It Together: The Toggle Switch

Now we can write a "complete" description of our toggle switch:

$$\emptyset \xrightarrow{F_1(P_2)} P_1 \tag{18}$$

$$\emptyset \xrightarrow{F_2(P_1)} P_2 \tag{19}$$

$$P_1 \xrightarrow{\gamma} \emptyset \tag{20}$$

$$P_2 \xrightarrow{\gamma} \emptyset \tag{21}$$

where $F_1(P_2) = \alpha_1 \frac{K_1^{N_1}}{P_2^{N_1}+K_1^{N_1}}$ and similarly for $F_2(P_1)$. This system can be sampled with SSA, or we can use mass action laws to derive the following ODEs under assumptions of large volume and fast mixing:

$$\frac{dP_1}{dt} = \alpha_1 \frac{K_1^{N_1}}{P_2^{N_1} + K_1^{N_1}} - \gamma P_1 \tag{22}$$

$$\frac{dP_2}{dt} = \alpha_2 \frac{K_2^{N_2}}{P_1^{N_2} + K_2^{N_2}} - \gamma P_2 \tag{23}$$

$$\tag{24}$$

In their original treatment, Gardner *et al.* further non-dimensionalize and simplify their model to

$$\frac{dP_1}{dt} = \frac{\alpha_1}{1 + P_2^{N_1}} - P_1 \tag{25}$$

$$\frac{dP_2}{dt} = \frac{\alpha_2}{1 + P_1^{N_2}} - P_2 \tag{26}$$

$$\tag{27}$$

and use these equations to derive necessary conditions for the toggle switch to work in a way matching intuition (namely, $N_1 > 1$, $N_2 > 1$, and $\alpha_1 \sim \alpha_2$).

FIGURE: a) Schematic of toggle switch; b) stochastic and deterministic traces of toggle switch under different parameters.

4. Why Gene Replication? Motivating Examples

Notice that in our model of the toggle switch, we have quietly disposed of our DNA species G_1 and G_2 as species, and lumped their concentrations into the values of parameters. This is a reasonable simplification when two conditions hold true:

1. **The concentration of DNA species is held constant by the cell.** Whether on genomic DNA or plasmids, most genes are copy number controlled by more or less complex cellular processes, and we can usually assume that these processes are functioning properly in the background.
2. **We do not need to track binding of species to the DNA.** This is a natural consequence of the Hill assumption made in section 3.2 – by directly representing expression as a function of repressor, we eliminate the need to explicitly track bound and unbound DNA states.

For each of these assumptions, we will consider a circuit for which that assumption does not hold.

4.1. Integrase-Based Event Recording

First, let's look at a circuit that violates our assumption of constant DNA copy number, in this case by making the DNA itself an active readout of the circuit. Consider the integrase-based temporal logic gate [1]. This circuit logs the presence of either of two signal molecules A and B, along with information about the relative timing of their appearance, using serine integrases to modify a shared DNA logging site. The logging site is designed so that it can be irreversibly modified by either of two different serine integrases (call them Int_A and Int_B), each of which is expressed in response to one of the two signal molecules. If the cell detects A before it detects B, then Int_A flips part of the logging DNA in a way that activates a green fluorescent signal and primes the logger for detection of B. If B is detected in this primed state, then Int_B flips a different part of the logger in a way that switches the green fluorescent output to a red fluorescent output. If, in contrast, the cell detects B before it detects A, Int_B instead excises a critical promoter from the logging DNA and renders it incapable of expressing any fluorescent signal.

FIGURE: Overview of the integrase-based temporal logic gate

In their original work, Hsiao *et al.* show that, although the temporal logic gate produces a digital red-or-green-or-none signal in any particular cell, stochastic differences in timing in different cells of a recorder population bearing the circuit will lead to the population expressing some mix of red, green, and nothing. Furthermore, the fraction of the population expressing each color carries information about how long the population was exposed to each signal, and in what order.

Hsiao *et al.* used a DNA logger integrated into the genome of engineered *E. coli* cells, ensuring that each cell would only carry a single copy of the logger (technically not true, but functionally true – after a few division cycles, each logging cell reliably bears loggers descended from a single copy of a parent chromosome). Would this circuit still work if the logging DNA were on a plasmid, instead of genome-integrated? How would plasmid partitioning CITE and fluctuating copy number CITE affect the circuit's output? Could a single cell (or a small number of cells) with temporal logic gates on a high-copy plasmid stand in for a large population of cells with the same circuit on their genomes?

We can answer these questions with a model of the temporal logic gate. Consider a logging plasmid L which can take the states L, L_A, L_B, and L_{AB} depending on the action and order of action of Int_A and Int_B. We will say that the integrases are activated as a function of their respective inducers A and B:

$$\emptyset \xrightarrow{H_A(A)} Int_A \tag{28}$$

$$\emptyset \xrightarrow{H_B(B)} Int_B \tag{29}$$

$$\tag{30}$$

where $H_X(X)$ is a Hill function of inducer $X \in \{A, B\}$ such that $H_X(X) = \frac{X^{N_X}}{X^{N_X}+K_X}$. Along with a rule for dilution at rate γ, this gives ODEs describing integrase behavior:

$$\frac{dInt_A}{dt} = \frac{A^{N_A}}{A^{N_A}+K_A} - \gamma Int_A \tag{31}$$

$$\frac{dInt_B}{dt} = \frac{B^{N_B}}{B^{N_B}+K_B} - \gamma Int_B. \tag{32}$$

We can use a similar hill approximation to describe the action of integrases on the logger. We could instead model the individual binding and unbinding reactions of protein to and from DNA, as we will in the next example, but for this example we will model integrase activity as a simple Hill function to more clearly highlight the relevant feature of the temporal logic circuit model, which is that most of the important, explicitly-tracked species in the model are plasmids.

$$L \xrightarrow{H_{IntA}(Int_A)} L_A \tag{33}$$

$$L \xrightarrow{H_{IntB}(Int_B)} L_B \tag{34}$$

$$L_A \xrightarrow{H_{IntAB}(Int_B)} L_{AB} \tag{35}$$

$$\tag{36}$$

If we are content with a deterministic, bulk-action treatment of the temporal logic gate, then we are done. Overall logger plasmid copy numbers are conserved, and integrase functions only to switch logger plasmids between states. However, this model is useless for answering questions about how cells with mixed plasmid populations might drift or fix over time. To answer those questions, we'll need some mechanisms for replicating and diluting out plasmids. We can start by adding our standard dilution reactions

$$L \xrightarrow{\gamma} \emptyset \tag{37}$$

$$L_A \xrightarrow{\gamma} \emptyset \tag{38}$$

$$L_B \xrightarrow{\gamma} \emptyset \tag{39}$$

$$L_AB \xrightarrow{\gamma} \emptyset \tag{40}$$

but this leaves us with a set of reactions that can destroy logging plasmids but not create them, which can only lead to cells where the logger has been lost. We will need some way to model the production of logging plasmids, which brings us to the central question of this paper – what is the appropriate way to model replication of DNA components in a synthetic circuit?

Note how this circuit violates the assumption of constant copy number. Although the cell might control the *overall* copy number of logger plasmid $L + L_A + L_B + L_{AB}$, the fraction of plasmids in each subpopulation will have to change over time for the circuit to be useful.

4.2. A CRISPRi-based Repressilator

Now let's consider a circuit for which DNA copy number might remain constant, but for which details of binding and unbinding matter, making it potentially important to explicitly track different DNA states. One such circuit is the 5-node CRISPRi-based repressilator, shown in Figure FIGNUM.

Schematic of a 5-node CRISPRi repressilator, with deterministic simulation.

CRISPRi circuits use deactivated Cas9 (dCas) loaded with different guide RNAs (gRNAs) as their active components CITE. Different gRNAs are expressed to target a shared pool of dCas proteins to different DNA locations, with each gRNA causing dCas to bind to and repress the expression of one or more target genes. Guide RNAs can even be used to repress the transcription of other guide RNAs, allowing the creation of CRISPRi circuits. In the 5-node repressilator shown in Figure FIGNUM, five different gRNAs are arranged in a cycle with each gRNA repressing the next gRNA in the cycle. With appropriate parameter tuning, this circuit will cycle through the expression of each gRNA in turn, acting as an autonomous oscillator.

Readers familiar with the history of synthetic biology will note that this model is quite similar to the classic three-node repressilator, which was one of the earliest examples of synthetic biology CITE. The repressilator's behavior is well-captured by traditional models without explicitly tracking DNA species. Why wouldn't a CRISPRi repressilator be modeled just as easily?

One possible reason is that the action of CRISPRi components is relatively slow, with a single molecule of dCas derived from *S. pyogenes* taking approximately six hours to find a genomic target in *E. coli* [2]. Functional repression with dCas can be sped up to a time scale of minutes by using high concentrations of dCas, but in a multi-node CRISPRi circuit even a large pool of dCas must be split up between several gRNAs. We may wish to ask how the speed of CRISPRi action affects the function of a circuit with time-varying components (*e.g.* a repressilator), and a model with an instantaneous Hill function approximation of CRISPRi action will not answer that question.

Instead, we will use a model that explicitly tracks DNA binding states for each of five promoters G_1, G_2, G_3, G_4, and G_5. Each gene G_i will produce a guide $gRNA_i$ that, when complexed with dCas, targets G_{i+1} (with $gRNA_5$ wrapping back to repress G_1). Any guide $gRNA_i$ can reversibly bind to dCas (C) to form a complex C_i, which itself can reversibly bind to its target to form a non-transcribing complex G_{i+1}^C. In CRN form, including dilution of dCas and gRNAs,

$$\emptyset \xrightarrow{\alpha_C} C \tag{41}$$

$$G_i \xrightarrow{\alpha_i} G_i + gRNA_i \tag{42}$$

$$gRNA_i + C \underset{k^f_{Ci}}{\overset{k^r_{Ci}}{\leftrightharpoons}} C_i \tag{43}$$

$$C_i + G_{i+1} \underset{k^f_{Gi}}{\overset{k^r_{Gi}}{\leftrightharpoons}} G^C_{i+1} \tag{44}$$

$$gRNA_i \xrightarrow{\gamma_g+\gamma} \emptyset \tag{45}$$

$$C \xrightarrow{\gamma} \emptyset \tag{46}$$

$$C_i \xrightarrow{\gamma_g+\gamma} \emptyset \tag{47}$$

for $i \in 1, \ldots 5$, again with the last gRNA wrapping back to target the first (so $5 + 1 = 1$), and where γ_g is the rate at which the cell actively degrades guide RNAs, on top of dilution. Once again, we may want to consider

One additional fact which we may wish to include in this model is that, at least in fast-growing prokaryotes, dCas complexes do not typically unbind from their targets except when actively displaced by DNA replication [2]. This can be captured most simply by setting $k^r_{G_i} = \gamma$. We will return to this coupled-unbinding assumption in later sections.

How can we handle gene replication dynamics in this model? One option is to simply include rules for duplication and dilution of genes, at the rate of cell growth γ:

$$G_i \xrightarrow{\gamma} 2G_i \tag{48}$$

$$G^C_i \xrightarrow{\gamma} 2G_i + C_{i-1} \tag{49}$$

$$G_i \xrightarrow{\gamma} \emptyset \tag{50}$$

$$G^C_i \xrightarrow{\gamma} \emptyset. \tag{51}$$

where reaction (49) replaces the unbinding reaction in (44) We've coupled plasmid replication to dCas unbinding. If we wished to keep unbinding and replication separate, we could instead write dilution of complexed gene as $G^C_i \to G_i + G^C_i$ (note that DNA replication should not duplicate dCas protein!). As a sanity check, we can verify that this does indeed give us a model in which genes remain constant concentration, as for any gene G_i,

$$\frac{dG_i}{dt} = -k^f_{G_i} * C_{i-1} * G_i + \gamma G_i + 2\gamma G^C_i - \gamma G_i \tag{52}$$

$$\frac{dG^C_i}{dt} = k^f_{G_i} * C_{i-1} * G_i - \gamma G^C_i - \gamma G^C_i. \tag{53}$$

Adding these two equations together, we find that the rate of change of total gene with time is 0. As we can see, it is straightforward to model gene replication dynamics in a deterministic biocircuit model. We shall soon see, however, that in the stochastic realm, this simple fix will no longer work.

5. Obvious Solutions That Don't Work

5.1. Naïve Gene Duplication

As we saw at the end of Section 4.2, it is generally straightforward to write down chemical reactions that will replicate and dilute genes in a balanced way, and this strategy is generally sufficient for modeling genes in continuous, deterministic regimes. What about stochastic regimes?

We can see the problem with a naïve *gene* → *gene* + *gene* strategy with a few stochastic simulations:

FIGURE: Stochastic simulation of D -¿ 2D, D -¿ Ø. Show that copy number is unstable.

As we can see in Figure NUMBER, blind self-replication is not stable. This is because the overall rate of DNA production in this model is *proportional to the current concentration of DNA*, which means that any chance increase in copy number tends to lead to higher copy number and any chance decrease in copy number tends to lead to a lower copy number. DNA concentration effectively becomes a random walk, with an absorbing state at zero DNA.

Why doesn't this instability show up in deterministic versions of these models? Actually it does – change the gene replication rate to be slightly different from dilution rate, and the deterministic model will also predict either total loss of gene or an unbounded explosion in copy number. The naı"ve model only works when we can set tune constants of production and dilution to cancel out exactly.

6. Blatantly Cheating: Simple Hacks to Get Gene Replication

7. Gently Cheating: Mechanistic Models of Gene Replication

8. Conclusions

9. References

References

[1] Victoria Hsiao, Yutaka Hori, Paul WK Rothemund, and Richard M Murray. A population-based temporal logic gate for timing and recording chemical events. Molecular Systems Biology, 869(12), 2016.

[2] Daniel Lawson Jones, Prune Leroy, Cecilia Unoson, David Fange, Vladimir Ćurić, Michael J. Lawson, and Johan Elf. Kinetics of dCas9 target search in *Escherichia coli*. 357:1420–1424, 2017.