Implementing Arithmetic and Other Analytic Operations By Transcriptional Regulation

Sean M. Cory¹, Theodore J. Perkins²*

¹Department of Human Genetics, McGill University, Montreal, Quebec, Canada. ²School of Computer Science, McGill University, McGill Centre for Bioinformatics, Montreal, Quebec, Canada

Abstract

The transcriptional regulatory machinery of a gene can be viewed as a computational device, with transcription factor concentrations as inputs and expression level as the output. This view begs the question: what kinds of computations are possible? We show that different parameterizations of a simple chemical kinetic model of transcriptional regulation are able to approximate all four standard arithmetic operations: addition, subtraction, multiplication, and division, as well as various equality and inequality operations. This contrasts with other studies that emphasize logical or digital notions of computation in biological networks. We analyze the accuracy and precision of these approximations, showing that they depend on different sets of parameters, and are thus independently tunable. We demonstrate that networks of these “arithmetic” genes can be combined to accomplish yet more complicated computations by designing and simulating a network that detects statistically significant elevations in a time-varying signal. We also consider the much more general problem of approximating analytic functions, showing that this can be achieved by allowing multiple transcription factor binding sites on the promoter. These observations are important for the interpretation of naturally occurring networks and imply new possibilities for the design of synthetic networks.

Introduction

In grappling with the biochemical complexity of gene regulation, some have turned to computational metaphors for explaining gene behavior (e.g., [1–5]). The lac operon, for example, is often described as implementing a simple logical rule—the gene is on when lactose is present and glucose is absent [6]. Theoretically, a variety of nonlinear chemical systems, including gene regulatory systems, are capable of implementing arbitrary logical rules [7–11]. Networks of such systems can implement finite state machines [12], and families of such networks of increasing size can be said to implement arbitrary Turing-computable functions [13,14]. In practice, logical models have proven capable of accounting for the qualitative dynamics of a variety of genetic systems [15–23]. Models that combine logical rules with concentration thresholds for the action of regulatory molecules, as in the French-flag model of Wolpert [24] or Glass networks more generally [25], satisfactorily describe other systems either qualitatively [26–28] or quantitatively [29]. Synthetic biologistst have constructed gene networks that perform elementary logical operations, such as storing a bit of memory [30] or turning off and on with a fixed period [31].

However, detailed analysis of transcriptional regulatory networks reveals a behavior richer than logical responses. Yuh et al.’s model of the sea urchin developmental gene Endo-16 contains logical as well as additive and multiplicative operations [32]. Recent measurements of lac transcription rate as a function of the concentrations of cAMP and an analogue of allo lactose show four plateaus of different rates connected by smooth boundaries [4,5]. Even “logical” models of gene regulation often require more than two qualitatively distinct levels of gene expression (e.g., [19]), recognizing that some genes cannot be treated simply as on or off.

Besides logical or digital computation, several other notions of computation have been explored. Analog computations made by artificial neural networks can in principle be implemented by chemical systems [8,12,33]. Deckhard and Sauro experimented with evolving reaction networks to compute square and cubic roots of an input [34] and have since evolved networks to compute a variety of other arithmetic functions. This work shares the greatest commonality with ours, the differences being that we specifically study single-gene transcriptional regulatory networks and that our designs are arrived at prescriptively, by analysis of the steady state equations, rather than by a computational search.

A different line of reasoning has focussed on the robustness of biochemical networks to variability in inputs and parameters—intuitively important features for real systems [35–37]. Reduction in noise by development gene networks was experimentally observed [38,39] and confirmed as a property of mathematical models [20,40]. In this background, control theoretic concepts, especially noise filtering, were studied more carefully [41] and found in a wide variety of systems (see [37] for a summary). More recently, the behavior of several genes has been viewed in the context of information theory [42] and Bayesian decision theory [43].

The present work focuses on the general steady-state analog computational capacities of genes. Most of the paper considers a
Author Summary

The biochemistry of the cell is daunting in its complexity. In order to understand this complexity, we are often forced to use metaphors or construct analogies to systems that we understand better. One long-standing analogy is to digital computers, with their large networks of interacting components that manage to act in coherent and useful ways. Indeed, we know from both theoretical models and empirical observations that biological entities such as genes can sometimes be described accurately by digital, or logical, expressions—turning on or off in response to regulatory signals. However, far more sophisticated computations can also take place, as has been documented in the responses of genes such as the lac operon or Endo-16. We analyze chemical kinetic models of transcriptional gene regulation and show that even simple models are capable of nearly arbitrary analog computations, ranging from elementary arithmetic operations to general analytic functions. Understanding the computational capacities of genes, and of biochemical systems more generally, tells us what to look for when studying natural systems and tells us what we can hope to build by biological engineering.

Results

A chemical-kinetic model of transcriptional regulation

Figure 1 presents a diagram of the model of transcriptional regulation and gene expression that we analyze. It models a single gene regulated by two transcription factors, A and B. These factors may bind irreversibly to form an inert dimer, or they may bind individually or simultaneously to the promoter region of the gene, where they affect the transcription rate. Transcripts, T, are translated into proteins, Z. Both T and Z decay at fixed rates.

doi:10.1371/journal.pcbi.1000064.g001

Formalize the model. (Please note that Tables 1, 2, and 3 show the symbols employed in this paper and their meanings.)

\[ A + B ^{\leftrightarrow} C \]  \hspace{1cm} (1)

\[ P_0 ^{\leftrightarrow} P_A \]  \hspace{1cm} (2)

\[ P_0 ^{\leftrightarrow} P_B \]  \hspace{1cm} (3)

\[ P_B ^{\leftrightarrow} P_{AB} \]  \hspace{1cm} (4)

\[ P_A ^{\leftrightarrow} P_{AB} \]  \hspace{1cm} (5)

\[ P_0 ^{\rightleftharpoons} P_0 + T \]  \hspace{1cm} (6)

\[ P_A ^{\rightleftharpoons} P_A + T \]  \hspace{1cm} (7)

\[ P_B ^{\rightleftharpoons} P_B + T \]  \hspace{1cm} (8)

\[ P_{AB} ^{\rightleftharpoons} P_{AB} + T \]  \hspace{1cm} (9)

\[ T ^{\rightleftharpoons} T + Z \]  \hspace{1cm} (10)

\[ T \overset{d}{\rightarrow} \emptyset \]  \hspace{1cm} (11)

\[ Z \overset{d}{\rightarrow} \emptyset \]  \hspace{1cm} (12)

For the bidirectional reactions, equilibrium association constants are ratios of forward to backward rates (e.g., \( K_{ob} = f_{ob}/b_{ob} \)). We use \([X] \) to denote the steady state concentration of molecular species \( X \). None of the reactions create or destroy the factors A or B. We assume that the binding of single molecules of these factors to the DNA does not significantly affect the concentration of free A or B, so that \([A] + [C] = [A_{tot}]\) and \([B] + [C] = [B_{tot}]\), where \([A_{tot}]\) and \([B_{tot}]\) denote the total concentration of molecules A and B respectively, either bound or unbound.

Most of our analysis concerns the steady state behavior of this system. We use \( \tau_{c} \), \( \tau_{ub} \), \( \tau_{b} \), and \( \tau_{ob} \) to denote the fractions of time that the promoter spends unbound or bound by different combinations of transcription factors. We assume that the reactions for transcription factors binding to the promoter are at equilibrium.

\[ \tau_o = K_{ob}[A]\tau_o \]  \hspace{1cm} (13)
From this system of equations, one can deduce that $K_{oa}K_{ab} = K_{ob}K_{ba}$. However, the system is degenerate and does not lead to a solution for the occupancy times until we recognize that the occupancy times must sum to one.

$$\tau_0 + \tau_a + \tau_b + \tau_{ab} = 1 \quad (17)$$

With the addition of this equation, we obtain four linearly independent equations, which can be manipulated to solve for the occupancy times.

$$\tau_0 = 1/(1 + K_{oa}[A] + K_{ob}[B] + K_{oa}K_{ab}[A][B]) \quad (18)$$

$$\tau_a = K_{oa}[A]/(1 + K_{oa}[A] + K_{ob}[B] + K_{oa}K_{ab}[A][B]) \quad (19)$$

$$\tau_b = K_{ob}[B]/(1 + K_{oa}[A] + K_{ob}[B] + K_{oa}K_{ab}[A][B]) \quad (20)$$

$$\tau_{ab} = K_{oa}K_{ob}[A][B]/(1 + K_{oa}[A] + K_{ob}[B] + K_{oa}K_{ab}[A][B]) \quad (21)$$

This allows us to calculate the net rate of transcription.

$$r_t = r_0 \tau_0 + r_a \tau_a + r_b \tau_b + r_{ab} \tau_{ab} \quad (22)$$

The steady state concentration of protein $Z$ then follows.

$$[Z] = \frac{r_{2z} r_t}{dZ/dt} \quad (23)$$

$$= \frac{r_{2z} + r_2 K_{oa}[A] + r_b K_{ob}[B] + r_{ab} K_{oa}K_{ab}[A][B]}{1 + K_{oa}[A] + K_{ob}[B] + K_{oa}K_{ab}[A][B]} \quad (24)$$

Because $K_{oa}K_{ab} = K_{oa}K_{ba}$, the term $K_{oa}K_{ab}$ can be replaced by...
Analytic Operations by Transcriptional Regulation

Approximating arithmetic operations

By employing subsets of the allowed reactions and setting kinetic parameters appropriately, different arithmetic operations can be approximated, with \([A_{tot}]\) and \([B_{tot}]\) treated as the inputs. (See Figure 2 for a summary.) For example, consider Equation 24 under the conditions: (i) \(r_t = 0\), (ii) \(K_{ob} = K_{ba} = 0\), (iii) \(t_e = 0\), (iv) \(r_{tzr}K_{oa} = r_{tzr}K_{ob} = 0\), (v) \(K_{oa}[A] \ll 1\), and (vi) \(K_{ob}[B] \ll 1\). These conditions can be interpreted as: (i) \(A\) and \(B\) do not dimerize, (ii) \(A\) and \(B\) cannot both be bound to the DNA at the same time, (iii) there is a certain balance between the production and decay rates of mRNA and protein, and (v),(vi) either \(A\) and \(B\) bind to the DNA comparatively weakly, or else we are considering only comparatively small concentrations of \(A\) and \(B\). Then Equation 24 tells us that \([Z] = [A_{tot}] + [B_{tot}]\), so that the gene approximates the addition operation. (See Figure 2B for the exact steady state expression.)

The error in the approximation is analyzed in more detail in the next section.

Alternatively, suppose that \(A\) and \(B\) bind the promoter sequentially, perhaps because \(A\) is a cofactor without which \(B\) cannot bind, and that transcription is activated only when both are bound. If production and decay are balanced as \(r_{tzr}K_{oa} = d_{dz}\), and if \([K_{oa}[A_{tot}]][K_{oa}[A_{tot}][B_{tot}] < 1\), then \([Z] = [A_{tot}] + [B_{tot}]\) so that the gene approximates the multiplication operation. To achieve \([Z] = [A_{tot}] + [B_{tot}]\) one can assume that \(A\) and \(B\) can bind individually to promoter, and that \(A\) activates transcription whereas \(B\) represses. This approximation is most accurate when \(B\) binds strongly, so that \([K_{oa}[B_{tot}] > 1\), and \(A\) binds comparatively weakly, so that \([K_{oa}[A_{tot}] \ll [B_{tot}]\).

As \([Z]\) is always non-negative, one cannot approximate ordinary subtraction, \([A_{tot}] - [B_{tot}]\), when \([B_{tot}] > [A_{tot}]\). Instead, we consider

Figure 2. Regulatory architectures and parameters for approximating various arithmetic and comparison operations. (A) Diagrams depict the reactions employed to achieve each operation. The A–B dumbbell is bold if \(A\) and \(B\) dimerize and gray if they do not. The four circles connected as a diamond represent different binding states of the promoter. In bold are achievable binding states, with bold connecting bars depicting the reactions employed to achieve each operation. The "Production-decay balance" column and setting to zero those parameters implied to be zero by the diagrams in (A). The final column of the table describes under what conditions each operation is well approximated. The symbol \(\ominus\) denotes zero-truncated subtraction, defined as \(x \ominus y = \max(x-y, 0)\). doi:10.1371/journal.pcbi.1000064.g002
zero-truncated subtraction.

$$[A_{tot}] \odot [B_{tot}] = \max ([A_{tot}] - [B_{tot}], 0) =$$

$$\begin{cases} [A_{tot}] - [B_{tot}] & \text{if } [A_{tot}] \geq [B_{tot}] \\ 0 & \text{if } [A_{tot}] < [B_{tot}] \end{cases} \quad (25)$$

If $A$ and $B$ dimerize irreversibly with forward rate $r_f > 0$, then at steady state $C = \min([A_{tot}],[B_{tot}])$. Thus, $[A] = [A_{tot}] - C = [A_{tot}] - \min([A_{tot}],[B_{tot}]) = [A_{tot}] \odot [B_{tot}]$. Dimerization itself, then, solves the zero-truncated subtraction problem. If we desire $[Z] = [A_{tot}] \odot [B_{tot}]$, we need only assume that undimerized $A$ binds to the promoter and activates transcription. For an accurate approximation, binding should be weak, so that $K_{oa}[A] \ll 1$.

For the comparison operations, the goal is $[Z] = 0$ if the comparison is false and $[Z] = Z_{on}$ if the comparison is true, for some chosen $Z_{on} > 0$. The operation $[A_{tot}] \geq [B_{tot}]$ can be implemented with the same set of reactions as subtraction, but $A$ should bind the promoter strongly so that $[Z] = Z_{on}$ in the presence of any amount of undimerized $A$. For $[A_{tot}] > [B_{tot}]$ we assume that $A$ and $B$ dimerize, that $A$ and $B$ can individually bind the promoter, acting as repressors. If $[A_{tot}] = [B_{tot}]$, then there will be no undimerized $A$ or $B$, and $[Z] = Z_{on}$. Otherwise, there will be some undimerized $A$ or $B$, and, if it binds the promoter strongly, will turn off transcription so that $[Z] = 0$.

These arithmetic operations can be implemented in other ways, even restricting attention to the model in Figure 1. For example, the addition gene could allow $A$ and $B$ to bind simultaneously, as long as there is no transcription when both are bound. Similarly, the multiplication gene could allow $A$ and $B$ to bind to the promoter in either order, as long as there is only transcription when both are bound. However, these alternate versions are less accurate than the designs in Figure ref{fig:diagram} because they introduce unnecessary promoter binding states. Mathematically, the decrease in accuracy corresponds to extra terms appearing in the denominator of the second term in Equation 24. Variations on these models can produce other useful functions. Trivial modifications allow the other comparison operations: greater-than-or-equal, less-than, and less-than-or-equal. More interestingly, consider a model with a single transcription factor $A$, which can be obtained by dropping all equations from the model that involve $B$. Suppose that $A$ activates transcription. Further, suppose that $Z$ dimerizes at forward rate $r_z$ and that the dimer itself is inert and degrades at some rate $d_z$, but that $Z$ on its own does not decay. Then Equation 12 is replaced by

$$Z + Z \xrightarrow{d_z} Z_2 \quad (26)$$

$$Z_2 \xrightarrow{d_z} \emptyset \quad (27)$$

If $r_z K_{oa} = d_z$ and $K_{oa}[A_{tot}] \ll 1$, then at steady state $[Z] \approx \sqrt{[A_{tot}]}$. Genes that approximate other fractional powers can be constructed by assuming more complicated promoter-binding or degradation schemes.

**Accuracy depends on promoter occupancy**

The accuracy with which the models in Figure 2 approximate the desired operations depends on the kinetic parameters as well as the transcription factor concentrations $[A_{tot}]$ and $[B_{tot}]$. We analyze accuracy in terms of relative error. For the arithmetic operations of addition, multiplication, division, and zero-truncated subtraction, we define relative error as

$$\frac{|f([A_{tot}],[B_{tot}]) - [Z]|}{f([A_{tot}],[B_{tot}])},$$

where $f$ is the desired operation. This is not well defined when $f$ is zero. However, for the models in Figure 2, $[Z] = 0$ whenever $f$ is zero, so we can take the relative error to be zero. For the comparison operations we define relative error as

$$\frac{|f([A_{tot}],[B_{tot}]) - [Z]|}{Z_{on}}.$$  

The relative error can be calculated by substituting the formula for $[Z]$ in Figure 2B into the definition for relative error. Figure 3 shows the relative errors of the six gene models from Figure 2. An advantage to studying this notion of accuracy is that the relative error can be easily expressed in terms of the fraction of time the promoter spends in different binding states (Figure 3, third column).

For the arithmetic operations, relative error is generally increasing in $[A_{tot}]$ and $[B_{tot}]$ as well as the equilibrium association constants for the binding of $A$ and $B$ to the DNA (Figure 3, second column). Thus, the approximations are most accurate when the concentrations of the transcription factors are small or when they bind weakly to the DNA—as already emphasized in the third column of Figure 2B. Intuitively, this keeps the transcriptional response in the linear regime. Saturation of the transcription rate, due to high concentration of transcription factor(s) or strong transcription factor binding, reduces accuracy. Indeed, the relative error of the addition, multiplication and zero-truncated subtraction genes is simply the fraction of time that the DNA is bound by either transcription factor. The division gene is a partial exception to these rules. First, its relative error is decreasing in $K_{oa}$ and $[B_{tot}]$, not increasing. Second, the relative error is equal to the fraction of time the DNA is not bound by $B$.

The effects of these parameters cannot be considered in isolation, however, because the parameters as a group must satisfy the production-decay balance shown in Figure 2B. For the addition gene, for example, small transcription factor association constants, which result in an accurate approximation, must be balanced by large rates of transcription and/or translation or small rates of transcript and/or protein decay.

For the comparison operations, accuracy is highest when $A$ and $B$ bind strongly to the promoter. If $K_{oa}$ and $K_{ob}$ are small and if $[A_{tot}]$ and $[B_{tot}]$ are just slightly different, then there is only a small amount of undimerized transcription factor binding weakly to the promoter, and this does not provide sufficient transcriptional activation (in the case of $>$) or repression (in the case of $\equiv$). For either gene, the relative error is either zero or $\tau_n$, depending on whether the comparison is true or false.

**Noise in the output is independent of accuracy**

In a stochastic kinetics interpretation of the model in Figure 1 (Equations 1 to 12), the output protein concentration varies over time even if $[A_{tot}]$ and $[B_{tot}]$ are fixed. Let $X(t)$ denote the concentration of molecular species $X$ at time $t$. Noise in $Z(t)$, defined as the standard deviation of $Z(t)$ over time divided by the mean of $Z(t)$ over time, can be attributed to three sources: (i) inherent fluctuations due to the stochastic birth-death process for $Z$, (ii) variability in $T(t)$ due to its own inherently stochastic birth-death process, which in turn creates variability in the production rate of $Z$, and (iii) variability in the promoter state, which affects the production rate of $T$, and by extension, of $Z$. If one assumes that the transcription factor binding reactions are at steady state, then the third source of noise is absent, and the noise in $Z(t)$ is equivalent to that in the simpler system

$$\emptyset \xrightarrow{d_z} T \quad (28)$$

$$T \xrightarrow{d_z} T + Z \quad (29)$$
Figure 3. Accuracy with which the genetic designs in Figure 2 approximate the intended operations. We quantify accuracy in terms of relative error—\(\frac{|f(A_{\text{tot}})B_{\text{tot}}| - |Z|}{|f(A_{\text{tot}})B_{\text{tot}}|}\) for the arithmetic operations, and \(\frac{|f(A_{\text{tot}})B_{\text{tot}}| - |Z|}{|Z|}\) for the comparison operations, where \(f\) is the operation being approximated. The second column in the table gives relative error in terms of the kinetic parameters. The third column gives the relative error in terms of the fraction of time the promoter spends bound by different combinations of transcription factors at steady state.

doi:10.1371/journal.pcbi.1000064.g003

\[
T \xrightarrow{d_1} \emptyset
\]

\[
Z \xrightarrow{d_2} \emptyset
\]

where \(t_i\) is the net rate of transcript production.

\[
r_i = \frac{r_a + r_o K_{oa}[A] + r_b K_{ob}[B] + r_{ob} K_{oa} K_{ob}[A][B]}{1 + K_{oa}[A] + K_{ob}[B] + K_{oa} K_{ob}[A][B]}
\]

For this system, the moments of \(Z\) can be calculated exactly from the chemical master equation, and the noise is [44]

\[
\eta = \frac{1}{Z^2} \left( 1 + \frac{r_z}{d_1 + d_2} \right)
\]

Importantly, the noise in \(Z(t)\) bears no necessary relationship with the accuracy with which \(|Z|\) approximates a desired function, because accuracy and noise depend on different sets of parameters. For the addition gene, for example, accuracy is determined by \(K_{oa}\) and \(K_{ob}\). These could be large or small regardless of the noise level, which is determined by \(r_z\), \(d_1\), and \(d_2\). This is true even when considering the production-decay balance constraint for the gene, \(r_z r_o K_{oa} = r_z r_b K_{ob} = d_1 d_2\). The parameters \(r_z\) and \(r_o\) do not occur in the formulas for either accuracy or noise and can be used to ensure that the constraints are satisfied. This is essentially the same as the observation by Thattai and van Oudenaarden that the mean and variance of gene expression levels are controlled by independent sets of parameters [44].

If the transcription factor binding reactions are not at steady state, then the third source of noise in \(Z(t)\) returns. Intuitively, however, the faster the binding and unbinding reactions are compared to the rate of transcription, the more this noise is filtered out by the slower transcription process. For example, consider the addition gene with inputs \([A_{\text{tot}}] = 30\) molecules and \([B_{\text{tot}}] = 70\) molecules. We used the Gillespie algorithm [45] to simulate the stochastic chemical kinetics defined by Equations 1 to 12, but replacing the transcription factor-DNA binding and unbinding rates \((f_{oa}, f_{ob}, b_{oa}, b_{ob})\) by scaled versions \((\rho f_{oa}, \rho f_{ob}, \rho b_{oa}, \rho b_{ob})\). By varying \(\rho\), we could change the rates of binding and unbinding, while leaving \(K_{oa}\) and \(K_{ob}\) constant. (See Materials and Methods for the full set of kinetic parameters.) Figure 4A shows three sample traces of \(Z(t)\) for three different choices of \(\rho\), in which the noise in \(Z(t)\) can be seen to decrease for increasing \(\rho\). Figure 4B shows the noise in the simulated \(Z(t)\) for a wider range of \(\rho\). For sufficiently large \(\rho\), the steady state approximation for the promoter is good, and the noise is seen to converge to the value predicted by Equation 33, indicated by the dashed line. Figure 4C shows that the empirical mean concentration \(|Z|\) is at nearly the correctly value of 100, and is independent of \(\rho\). Independence of accuracy from promoter state fluctuations has been shown analytically via Master equation analysis for some binding scenarios [46], but should hold in general for our models, as noise depends on both the binding and unbinding rates of transcription factors, whereas accuracy depends only on the ratios of the rates.

**Example: A network for detecting statistically significant elevation in a time varying signal**

Arithmetic genes can be combined into networks to perform more sophisticated computations. As an example, consider a cell in which the concentration of a molecule \(I\) varies with time. Suppose it is important for the cell to detect times at which \(I(t)\) is significantly elevated. For example, the cell might be a bacterium and \(I(t)\) might be correlated to the concentration of an extracellular sugar. If the bacterium encounters a high-sugar environment, it may want to begin expressing the genes needed to transport and metabolize the sugar. Or, \(I\) may be a toxin, and high toxin levels might trigger a defensive or developmental decision, such as sporulation, to protect the bacterium. In a synthetic biology context, \(I\) might be a signal sent by the experimenter to trigger a response or a marker for a diseased cell that should be destroyed.
A single threshold-activated gene, for example the “greater than” gene of Figure 2, provides a simple solution to this problem. However, such a gene requires a predefined notion of how large a signal is considered elevated. Further, a chance fluctuation in the signal or in the transcriptional machinery itself might trigger a false response. The feedforward motif[47] is one way to reduce incorrect responses due to fluctuations. This motif describes a set of three genes in which gene 1 regulates gene 2, and both genes 1 and 2 regulate gene 3. The feedforward motif appears far more often than chance in natural genomes[48,49], and presents a variety of temporal information processing possibilities[50,51]. Among them is the ability to detect sustained, rather than merely transient or accidental, increases in an incoming signal. However, the level of signal that is considered elevated is still preordained.

We consider a more challenging version of the problem in which the statistics of the signal are not known ahead of time. This may be because the operating environment of the cell is not known a priori or because the signal itself is difficult to measure experimentally. Whether or not the signal is elevated at a given moment thus depends on the signal’s mean value and typical fluctuations—properties which may themselves change over time. More formally, if the statistics of $I(t)$ are relatively constant for some period, then a natural definition of elevated is $I(t) > I_{\text{mean}} + \sigma I$, where $I_{\text{mean}}$ is the mean of $I(t)$ over time, $\sigma I$ is the standard deviation, and $c$ is a constant specifying how large an elevation is of interest. To solve the problem, the cell must identify the signal statistics $I_{\text{mean}}$ and $\sigma I$, and then compute whether the signal is elevated at any particular time.

Figure 5A depicts a network of arithmetic genes that accomplishes this task. The circles represent genes and an arrow from one gene to another means that the first gene’s protein acts as an input to the second gene. The network comprises arithmetic functions for multiplication, addition, subtraction, taking the square root, and comparison, as well as two genes labeled by $m$. The arithmetic and comparison genes are assumed to reach steady state at a faster time scale than variations in the input signal $I(t)$. The two $m$ genes are simply activated proportionally to their input and are assumed to reach steady state at a slower time scale than variations in the input signal. Thus, the $m$ genes effectively compute a recency-weighted time-average of their inputs, the exact nature of which depends on the details of the kinetic parameters. Gene $G1$ multiplies $I(t)$ by itself, and because $G1$ operates at a faster time scale than $I$, we can approximate the expression of its protein as $Z^*(t) = I(t)^2$. Gene $G2$ averages $Z^*(t)$ over time, so we can approximate its expression as $E_2[Z^*(t)] = E_2[I(t)^2]$, where $E_2$ denotes averaging over time. By similar reasoning, the expression of genes $G3$ and $G4$ is $Z^t(t) = E_2(I(t))$ and $Z^s(t) = (Z^t(t))^2 = (E_2(I(t)))^2$. Genes $G5$ and $G6$ compute $\sqrt{Z^t(t) - Z^s(t)} = \sigma I$. (See Equations 26 and 27 and surrounding discussion for the square root function.) $G7$ is a modified addition gene which computes $Z^s(t) = \mu I + \sigma I$, for constant $c$, and $G8$ compares this result to $I(t)$, turning on when the latter is greater than the former.

To test this network, we simulated a differential equation model of transcript and protein dynamics subject to a time-varying input (see Figure 5B and Materials and Methods). The simulation covered 20 days during which the signal’s mean or standard deviation changed three times. There were also six brief spikes in the signal. We chose $c = 3$, so that $G8(t)$ should turn on, to an expression level of 100 nM, whenever $I(t)$ exceeded its mean plus three standard deviations. Details of the simulation, including the exact equations and kinetic parameters, can be found in the Materials and Methods section.

Figure 5C–E shows the results of our simulation. The simulation covers 20 days of simulated time, though we do not show the first 3 days during which a transient due to the initial conditions disappears. Figure 5B shows the input signal, and Figure 5C shows the overall response of the network to that signal. The network responds at the times that it should and at no other times. Its response to the input spikes and the increase in oscillation amplitude on day 14 are brief and do not reach the fully-on level of 100 nM, mainly because the elevations themselves are brief. When the signal jumps higher on day 8 and remains there, the network responds for longer and reaches a fully-on level. However, this response too vanishes as the network adjusts to the change in signal statistics. (The problem statement requires adjustment to changing signal statistics, so that elevation is relative to the recent mean and standard deviation of the signal. If a more sustained response is desired, $G8$ could be augmented with a positive feedback loop to keep it activated once triggered.) Figure 5D,E show the mean and standard deviation of the signal’s oscillations and the network’s recency-weighted estimates of signal mean and standard deviation. Interestingly, changes to the signal mean on days 8 and 11 are initially interpreted by the network as changes in the standard deviation of the signal. Only over the course of several days is the mistake rectified, with the mean estimate shifting to the correct value and the standard deviation estimate returning to its previous, correct level.

**Approximating analytic functions**

The elementary arithmetic operations provide a basis for analog computation, in that these operations can be combined to compute more complex functions. However, from the standpoint...
of the cell, it seems desirable to perform computations with as little biomolecular machinery as possible. From the form of Equation 24, it is not clear how other elementary operations, such as exponentiation, logarithms, or trigonometric functions, might be implemented. Whether these particular functions are used by cells can be debated, but certainly cells need to compute more complicated functions than elementary arithmetic. One way to approximate more complicated operations is by power series expansions.

The focus of this section is on analytic functions—functions with Taylor series expansions that converge pointwise to the correct values, at least for some range of inputs. For simplicity, we consider single-input functions. Generalization to multi-input functions is straightforward.

Figure 6A shows a schematic of a model describing a single gene regulated by one type of transcription factor, denoted by $A$. We assume $N$ binding sites for $A$ on the gene's promoter, each acting independently with equilibrium association binding constant $K$. We assume that transcription rate depends only on the number of binding sites occupied by $A$. This model is formalized by the reaction equations below.

$$
\begin{align*}
T & \xrightarrow{r_1} T + Z \\
T & \xrightarrow{d_1} \emptyset \\
Z & \xrightarrow{d_1} \emptyset
\end{align*}
$$

Here, $P_i$ denotes the promoter bound by $i$ copies of factor $A. T$ denotes transcript, and $Z$ denotes protein. The steady state expression of protein $Z$ is

$$
[Z] = \frac{r_n}{d_n} \times \frac{r_0 + r_1 K N [A] + \ldots + r_N K^N [A]^N}{(1 + K [A])^N}
$$

This can also be written as $[Z] = \frac{r_n}{d_n} \times \sum_{i=0}^{N} r_i \tau_i$, where $\tau_i = K^i \frac{[A]^i}{(1 + K [A])^N}$ is the fraction of time $i$ binding sites for $A$ are occupied. Now, consider an analytic function $f(x)$ with power series expansion

$$
f(x) = c_0 + c_1 x + c_2 x^2 / 2 + c_3 x^3 / 6 + \ldots
$$

Suppose first that all the $c_i$ are nonnegative. Then the gene can...
approximate the function up to order $N$ by assuming $K[A] \ll 1$, so that the term $(1+K[A])^N$ is approximately one, and matching the coefficients in Equations (39) and (40).

$$\frac{r_{2N}K^N}{dA} \left( \begin{array}{c} N \\ i \end{array} \right) = \frac{c_i}{n!}$$  \hspace{1cm} (41)

If some $c_i$ are negative, then the series can be divided into positive and negative components.

$$f(x) = c_0^+ + c_1^+ x + c_2^+ x^2/2 + c_3^+ x^3/6 + \ldots - c_0^- - c_1^- x - c_2^- x^2/2 - c_3^- x^3/6 - \ldots$$

where $c_i^+ = \max(c_i,0)$ and $c_i^- = -\min(c_i,0)$. Two genes of the type in Figure 6A can then combine to compute the function, one computing the positive part of the series and one computing the negative part. The difference in the protein concentrations of the two genes approximates $f$.

For example, suppose the cell wanted to approximate the cosine function using a pair of genes of the type in Figure 6A with five binding sites. The Taylor series expansion for cosine, centered at zero, has coefficients $c_0 = 0$, $c_1 = 4$, $c_2 = 0$, $c_3 = 0$, $c_4 = 4$, ... The blue and green curves in Figure 6B show, respectively, the cosine function and its 5th order Taylor series approximation. Using biologically plausible parameters for the transcription factor binding affinity, translation rate, and rates of transcript and protein decay (see Materials and Methods), Equation 41 can be solved for the necessary transcription rates, $r_i$. The red curve in Figure 6B shows $[Z] = [Z^+] - [Z^-]$, where $[Z^+]$ is the steady state protein concentration for the gene approximating the positive portion of the series, and $[Z^-]$ is the steady state protein concentration for the gene approximating the negative portion of the series. The Taylor series approximation and $[Z]$ match each other closely, though neither is a good approximation to $\cos([A])$ above a concentration of about $[A] = 2 \text{ nM}$. If the cell places importance on approximating the function at higher concentrations, then different parameters are need. The cyan curve in Figure 6B shows $[Z]$ for the same two-gene system when parameters are optimized to minimize the mean square error between $[Z]$ and $\cos([A])$ over the range $[A] \in [0,2\text{ nM}]$. (See Materials and Methods for details.) Sacrificing quality only slightly near zero allows the gene pair to capture a full period of the function, while retaining biologically plausible parameters.

**Discussion**

We have shown that different parameterizations of a simple model of transcriptional regulation can reproduce binary arithmetic operations (addition, zero-truncated subtraction, multiplication, division), various inequality comparisons, and fractional powers. Unary, ternary, quaternary, etc. operations can be similarly implemented at a biochemical level. These models are not the only way, and may not be the best way, that these operations can be implemented. For example, we have not considered models with cooperativity between binding sites, allosteric features for transcription factors, regulation of translation or degradation, or chromatin-related mechanisms. One way to explore the utility of these mechanisms would be to perform an explicit computational search through some appropriate space of allowed reaction systems (as in Deckard and Sauro [34], for example), searching for systems with superior performance in terms of various metrics (such as accuracy, noise, robustness, evolvability, or energy consumption). Nevertheless, our simple models demonstrate the possibility of arithmetic computations at the level of transcriptional regulation.

Other functions, such as exponentiation, logarithms, and trigonometric functions, could be computed by combinations of the arithmetic operations. However, this would be energetically wasteful and would require many genes. A more efficient approach is to approximate directly by using more complicated regulatory machinery. We showed that a pair of genes, each with $N$ binding sites for a transcription factor, can approximate any analytic function up to order $N$ by reproducing its power series approximation. We also showed that by adjusting the parameters of such a gene pair, the analytic function can be approximated well over a larger range of inputs. While this drastically reduces the number of genes used to compute such functions, other mechanisms may be yet more efficient, and this is an avenue for future research.

Our emphasis on arithmetic computations stands in contrast to work on logical or Boolean notions of computation in genetic networks. Interestingly, Hjelmfelt et al. [12] and Magnasco [13] both considered the problem of addition, but in the binary sense, with different chemical species effecting each bit of the operation.

---

**Figure 6. Approximation of analytic function.** (A) Schematic of interactions for a gene regulated by a single transcription factor, $A$, via $N$ independent binding sites. (B) For varying input levels, $[A]$, the four curves represent: $\cos([A])$, the 5th order Taylor series approximation of $\cos([A])$, and the steady state output when kinetic rates are optimized so that $[Z] = \cos([A])$, over the range $[A] \in [0,2\text{ nM}]$.

doi:10.1371/journal.pcbi.1000064.g006
In general, there is no relationship between an arithmetic interpretation of a gene’s behavior—which is accurate for a limited range of transcription factor concentrations—and a logical interpretation—which focuses on the extremes of low and high transcription factor concentrations. For example, in our model for an Addition gene, transcription occurs whenever factor A or factor B is bound, keeping in mind that factors A and B cannot bind simultaneously (Figure 2A). Such a gene could be interpreted as implementing the logical OR function, because it would be expressed if either factor A or factor B is in high concentration. Recall, however, that an alternate design for an Addition gene allows A and B to bind simultaneously, as long as there was no transcription when both were bound. This gene would be interpreted as implementing XOR, as it would be expressed if A or B, but not both, were in high concentrations. Conversely, our designs for the zero-truncated subtraction function and the greater than function have identical logical structure (Figure 2A), but behave quite differently as arithmetic operations. Thus, a gene’s arithmetic behavior, if any, is not determined by its logical behavior and vice versa. This is an important caveat for the interpretation of empirical measurements of gene response surfaces, which may be probed at non-physiological concentrations of regulatory molecules.

For our gene models to accurately approximate the desired operations, certain products of kinetic parameters must be equal. For example, the multiplication gene requires that $r_{z_{ab}K_{oa}K_{ob}} = d_{d_{z}}$. Such a constraint may be biologically implausible or difficult to maintain over different cellular conditions. If the constraint did not hold, the expression of the multiplication gene would still be proportional to the product of the transcription factor expression levels—the role of the constraint is only to scale the output expression correctly. Thus, violation of the constraint in a particular situation need not invalidate the interpretation of the gene as performing multiplication. The case is similar for all the genes in Figure 2 except the addition gene. This gene requires $r_{z_{ab}K_{oa}K_{ob}} = r_{d_{z}K_{oa}}$. If those equalities failed, but $r_{d_{z}K_{oa}} = r_{d_{z}K_{ob}}$, then the expression of the gene would still be proportional to the sum of the transcription factor expression levels. If the latter equality failed as well, then the gene could be interpreted as performing a weighted sum. Indeed, we took advantage of this possibility in our design for the network that detects significant elevations in a time varying signal (see gene G7 in Figure 5A).

In our models, precision (noise) in the output depends on the transcription factor binding and unbinding rates, while the accuracy of the output depends only on the ratio of the rates. This implies that there is no trade-off between precision and accuracy. Each can be selected independent of the other, by evolution or by synthetic biologists, subject to biological limitations on the parameters. We have not analyzed the speed with which these networks reach equilibrium, which could also be an important factor in some circumstances. Our Gillespie simulations for the analysis of noise for the Addition gene used a slightly simplified set of equations. Because factors A and B cannot bind the promoter simultaneously, we lumped the two bound states into one.

$$P_0 \overset{f(A + B)}{\rightarrow} P_1$$

(43)

$$P_1 \overset{b}{\rightarrow} P_0$$

(44)

where $P_i$ denotes the promoter bound by either $A$ or $B$, $f$ is the forward binding rate of $A$ and $B$ to the promoter, and $b$ is the unbinding rate. Transcription then follows the reaction

$$P_1 \overset{b}{\rightarrow} P_i + T$$

(45)

Translation, degradation of transcripts, and degradation of proteins remained the same, following Equations (10) to (12).

We expect that notions of computation more sophisticated than simple logical operations, already being documented experimentally [4,5,32,53], will play an increasing role in our understanding of transcriptional regulation and of biochemical systems more generally. Our analysis of the analog computational abilities of transcriptional regulatory networks is a step in this direction.

**Materials and Methods**

Our Gillespie simulations for the analysis of noise for the Addition gene used a slightly simplified set of equations. Because factors A and B cannot bind the promoter simultaneously, we lumped the two bound states into one.

$$P_0 \overset{f(A + B)}{\rightarrow} P_1$$

(43)

$$P_1 \overset{b}{\rightarrow} P_0$$

(44)

where $P_i$ denotes the promoter bound by either $A$ or $B$, $f$ is the forward binding rate of $A$ and $B$ to the promoter, and $b$ is the unbinding rate. Transcription then follows the reaction

$$P_1 \overset{b}{\rightarrow} P_i + T$$

(45)

Translation, degradation of transcripts, and degradation of proteins remained the same, following Equations (10) to (12). We used the parameters \([A_0] = [B] = 30 \text{nM}, [B] = 70 \text{nM}, f = \frac{\log_2(7)}{3600} \text{sec}^{-1}, b = \frac{\log_2(7)}{1800} \text{sec}^{-1}, \rho_1 = 1 \text{nM}^{-2} \text{sec}^{-1}, \rho_2 = 2 \text{nM}^{-2} \text{sec}^{-1}, d_1 = \frac{\log_2(7)}{3600} \text{sec}^{-1}, d_2 = \frac{\log_2(7)}{1800} \text{sec}^{-1}\), where $\rho$ is the dimensionless parameter we used to control the binding and unbinding rates of $A$ and $B$ to the promoter, while leaving the equilibrium association constant unchanged. For different values of $\rho$, the system was simulated from the initial condition $P_0 = 1, P_1 = T = Z = 0$, for one million reactions. Figure 4A plots initial portions of several of those trajectories, but Figure 4B,C are based on the full one million reactions.

Our simulations of the network for detecting elevated signals involved 17 variables: the exogenous input $I(t)$, and a transcript concentration $T(t)$ and protein concentration $Z(t)$ for each gene $i$. All variables have units nanomolar (nM). The input signal was primarily a sinusoidal oscillation with a period of four hours. Oscillations had mean 10 and amplitude 2 for the first eight of the 20 days simulated, mean 20 nM and amplitude 2 nM for the next three days, mean 10 nM and amplitude 2 nM for another 3 days, and mean 10 nM and amplitude 6 nM for the final six days. An independent Gaussian disturbance with mean zero and variance one was added to the signal during every 10 minute period. At the starts of days 5, 6, 7, 17, 18 and 19, there were 10 minute spikes in the signal at levels 16, 18, 20, 20, 22 and 24 nM respectively. We assumed the transcription factor binding equations were at steady state in order to speed the calculations. Transcription was driven at the rate expected based on the transcription factor concentrations. The exact differential equations simulated were:

$$T^1 = \frac{r_{z_{ab}K_{oa}K_{ob}I(t)}^2}{1 + K_{oa}I(t) + K_{ob}I(t)} - d_{z_{1}}T^1(t)$$

(46)

$$Z^1 = r_{z_{ab}T^1(t)} - d_{z_{2}}Z^1(t)$$

(47)
Table 4. Parameters for the simulation of the network for detected significant elevations in a time-varying signal.

| Gene | $d_{i}$ sec$^{-1}$ | $d_{j}$ sec$^{-1}$ | $r_{m}$ sec$^{-1}$ | $r_{o}$ nM sec$^{-1}$ | $r_{ab}$ nM sec$^{-1}$ | $K_{oa}$ nM$^{-1}$ | $K_{oa}$ nM$^{-1}$ | $K_{ab}$ nM$^{-1}$ |
|------|-------------------|-------------------|-------------------|-------------------|-------------------|------------------|------------------|------------------|
| 1    | log(2)            | log(2)            | log(2)            | 10               | log(2)            | 1                |                  |                  |
| 2    | log(2)            | log(2)            | log(2)            | 1                | log(2)            | 1                |                  |                  |
| 3    | log(2)            | log(2)            | log(2)            | 10               | log(2)            | 1                |                  |                  |
| 4    | log(2)            | log(2)            | log(2)            | 10               | log(2)            | 1                |                  |                  |
| 5    | log(2)            | log(2)            | log(2)            | 10               | log(2)            | 1                |                  |                  |
| 7    | log(2)            | log(2)            | log(2)            | 10               | log(2)            | log(2)           |                  |                  |
| 8    | log(2)            | log(2)            | log(2)            |                  |                  |                  |                  | 10               |

do:10.1371/journal.pcbi.1000064.t004

$T^{2} = \frac{r_{i}^{2}K_{oa}^{2}Z_{i}^{1}(t)}{1 + K_{oa}^{2}Z_{i}^{0}(t)} - d_{i}^{2}T^{2}(t)$  \hspace{1cm} (48)  

$Z^{2} = r_{i}^{2}T^{2}(t) - d_{i}^{2}Z^{2}(t)$  \hspace{1cm} (49)  

$T^{3} = \frac{r_{i}^{3}K_{oa}^{3}I(t)}{1 + K_{oa}^{3}I(t)} - d_{i}^{3}T^{3}(t)$  \hspace{1cm} (50)  

$Z^{3} = r_{i}^{3}T^{3}(t) - d_{i}^{3}Z^{3}(t)$  \hspace{1cm} (51)  

$T^{4} = \frac{r_{i}^{4}K_{oa}^{4}K_{ab}^{4}Z_{i}^{1}(t)^{2}}{1 + K_{oa}^{4}Z_{i}^{0}(t) + K_{oa}^{4}K_{ab}^{4}Z_{i}^{1}(t)^{2}} - d_{i}^{4}T^{4}(t)$  \hspace{1cm} (52)  

$Z^{4} = r_{i}^{4}T^{4}(t) - d_{i}^{4}Z^{4}(t)$  \hspace{1cm} (53)  

$T^{5} = \frac{r_{i}^{5}K_{oa}^{5}max(Z^{2}(t) - Z^{2}(t),0)}{1 + K_{oa}^{5}max(Z^{2}(t) - Z^{2}(t),0)} - d_{i}^{5}T^{5}(t)$  \hspace{1cm} (54)  

$Z^{5} = r_{i}^{5}T^{5}(t) - d_{i}^{5}Z^{5}(t)$  \hspace{1cm} (55)  

$T^{6} = \frac{r_{i}^{6}K_{oa}^{6}Z^{2}(t)}{1 + K_{oa}^{6}Z^{2}(t)} - d_{i}^{6}T^{6}(t)$  \hspace{1cm} (56)  

The system of differential equations was simulated using the `ode45` function of MATLAB. Equations 46 through 61 employ several simplifications compared to the equations one would obtain from a direct translation of the kinetic equations in Figure 1, mainly for the purpose of improving numerical stability. For example, we do not explicitly model the heterodimer of $Z^{2}$ and $Z^{3}$ called for by gene 5, which subtracts one concentration from the other. Rather, we drive gene 5 transcription at a rate corresponding to what would be observed if any available $Z^{2}$ and $Z^{3}$ dimerized instantly and irreversibly. Similarly, we do not model production and subsequent decay of the homodimer of $Z^{3}$ suggested by our design for the square root operation. Rather, we simply assume that $Z^{3}$ dimerizes at rate $d_{i}^{3}$ and instantly decays.

Table 5. Optimized parameters for a pair of genes approximating the cosine function.

| Gene | $r_{m}$ | $d_{i}$ | $d_{j}$ | $K$ | $r_{o}$ | $r_{ab}$ | $K_{oa}$ | $K_{oa}$ | $K_{ab}$ |
|------|---------|---------|---------|-----|---------|---------|---------|---------|---------|
| +    | 0.0049  | 0.0761  | 0.0013  | 0.0972 | 0.0020  | 0.0018  | 0.0142  | 0.0003  |
| −    | 0.0147  | 0.0031  | 0.0007  | 0.0419 | 0.00  | 0.0  | 0.0161  | 0.0214  |

Units for the parameters are the same as in Table 4.
do:10.1371/journal.pcbi.1000064.t005
The kinetic parameters are summarized in Table 4. The “fast” genes were given protein half-lives of 5 minutes and the “slow” genes were given protein half-lives of 10 hours. These values are somewhat extreme, but the short half-life is consistent with measurements for actively degraded proteins [54] and both half-lives are within the range of recent measurements for a genome-wide study of yeast [55]. All other parameters are in typical ranges.

For our study of the cosine approximation, we used parameters

\[ r_2 = \frac{\log_2 2}{6} \text{ sec}^{-1}, \quad d_1 = \frac{\log_2 2}{60} \text{ sec}^{-1}, \quad d_2 = \frac{\log_2 2}{600} \text{ sec}^{-1}, \quad \text{ and } K = 0.01 \text{ nM}^{-1}. \]

For the direct Taylor series approximation, Equation 41 implied transcription rates \( r_2 = 5.78 \times 10^{-5} \text{ nM sec}^{-1} \), \( r_3 = 2.28 \times 10^{-2} \text{ nM sec}^{-1} \), and \( r_4 = 48.1 \text{ nM sec}^{-1} \). For the optimized curve, we allowed all the parameters to vary, including the transcription rates of the positive and negative genes, and their rates of translation and transcript and protein decay. We use the matlab function “fminsearch” to optimize the parameters, minimizing the mean squared error between steady state \( [Z] \) and \( \cos([A]) \). Because all parameters should be non-negative, they were reformulated as \( p_j = \exp(b_j) \) where \( b_j \) is the \( j \)th parameter and \( b_i \) is the unconstrained parameter that we actually optimized. This resulted in the constrained parameters shown in Table 5.

Acknowledgments

The authors are indebted to Peter Swain and Mathieu Blanchette, who read several drafts of this work and made suggestions resulting in significant improvements to both content and presentation.

Author Contributions

Conceived and designed the experiments: SC TP. Performed the experiments: SC TP. Analyzed the data: SC TP. Wrote the paper: TP.
53. Andrews BW, Yi TM, Iglesias PA (2006) Optimal noise filtering in the chemotactic response of Escherichia coli. PLoS Comput Biol 2: e154.
54. Bachmair A, Finley D, Varshavsky A (1986) In vivo half-life of a protein is a function of its amino-terminal residue. Science 234: 179–186.
55. Belle A, Tanay A, Bitincka L, Shamir R, O’Shea EK (2006) Quantification of protein half-lives in the budding yeast proteome. Proc Natl Acad Sci U S A 103: 13004–13009.