

# The Modular Modeling of Biological Networks using Transcription Rates

Austin Che & Bogdan Fedeleş

December 12, 2002

## 1 Introduction

Biological systems are often described as networks involving interactions among DNA, RNA, and proteins. Including all these interactions in a comprehensive model is a monumentally challenging task, given the complexity of the systems. Nevertheless, it is unnecessary and often undesirable for a model to contain all of biology's complexity, as long as the model can predict outputs given some inputs, allowing for the cheap *in silico* prediction of cell fates and responses to changing environmental conditions.

### 1.1 Goals

We propose a modeling approach based on transcription rates that abstracts away most of the complexity of DNA, RNA, protein, and small molecule interactions. Cellular pathways are modeled by dividing pathways into modules that receive inputs and have outputs. All inputs and outputs for modules and interactions among modules are measured using a unifying dynamic currency – transcription initiations per second (TIPS), a measure of transcription rate.

### 1.2 Modularity

Figure 1 is a schematic of a simple module with a single input and single output. The input transcription rate causes transcription of a mRNA that is then translated to a protein. This protein interacts with other proteins and eventually activates a transcription factor that binds to DNA, causing transcription at a certain rate and giving a certain output TIPS. For this model, we do not need to know the details of the protein interactions, but only how the input is matched to the output. A transfer curve between the input and output completely describes the behavior of the module, so that the insides of the module can be treated as a black box.

Using a common unit of measurement for input and output signals allows modules to be interchangeable and studied separately. A modular approach has significant advantages, allowing for the easy engineering of synthetic networks

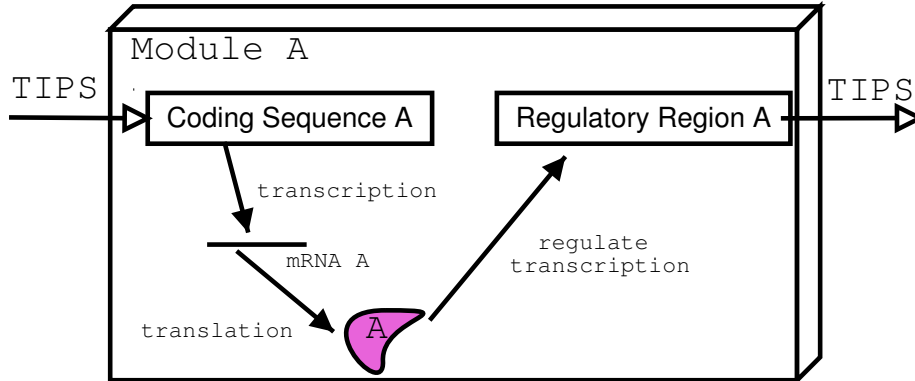


Figure 1: Representation of a prototypical module with input and output TIPS.

and functionality by moving modules around. In addition, by specifying a module in quantitative terms, we would have precise control over gene expression by varying input transcription rates.

Transfer curves for modules should be strictly a function of the inputs to the modules so that the transfer curve for a module is always valid under any biological conditions. This will avoid possible crosstalk among modules as all module interactions must involve the inputs and outputs. In practice, this requirement may lead to large modules with many inputs and outputs.

We want to prove that this kind of modeling allows accurate predictions of system dynamics when the transfer curves for each module can be estimated. The main caveat of this approach is that transcription rate data is almost non-existent in the literature, perhaps because there are no good methods for measuring transcription rates *in vivo*. We propose several methods for measuring transcription rates and transfer curves. In addition, we present several example networks built upon these principles and examine their functionality using simulated data.

## 2 Methods

### 2.1 Library of Regulatory Regions

Using TIPS as the universal currency introduces a new challenge of measuring TIPS. Because a direct method of measuring TIPS could be very laborious (see below), indirect but simpler methods are desirable.

RNA levels in a cell are not always indicative of transcription initiation rates, due to varying degradation and transcription elongation rates, but the ratio between RNA levels of two genes can be used as a better approximation of the

ratio between their transcription rates. This assumption allows us to measure TIPS of various promoters by comparing them with promoters of known TIPS.

One way to determine TIPS of an unknown regulatory region is to have a standard library of regulatory regions that have known increasing strengths, therefore known TIPS levels. Such a library is also useful in controlling gene expression in a system of choice. We can substitute the regulatory region of desired strength into the pathway under study and see the effects of changing transcription levels.

### **2.1.1 Creating the Library**

Creating such a library can be done in many ways. One way is to choose a constitutive promoter region from the study system and then generate a whole library of random regions using mutagenic PCR. Each of these regions can be inserted into a standard vector and put into the library. Many databases of promoter regions exist [4] [13] and can be used as a start to creating such a library.

### **2.1.2 Searching the Library**

We can transfect cells with the library of constitutive promoters created and then measure the relative levels of mRNA generated by the promoter in the library with respect to the level of mRNA of a standard. There is stochastic noise in gene expression in any system [3], but measuring the relative mRNA levels over a whole colony should average out random variations. To get more accurate comparisons, we can use two internal standards (two constitutive regulatory regions) and compare one to the other and each one to the test regulatory region. mRNA levels can be measured by standard techniques: isolating the RNA, reverse transcribing it to cDNA and using DNA chips to compare the relative amounts of cDNA.

An alternative to the more expensive DNA chips is to stretch the ratio assumption further to the protein levels. We can put the two promoters we are comparing in front of two fluorescent proteins, such as CFP and YFP and measure the ratio of the different fluorescences. However, due to random transcriptional and translational variations, the relative level of proteins may not be as accurate a predictor of transcription initiation rates as the relative levels of mRNA.

### **2.1.3 Matching TIPS Levels**

A different way to search the library is to have a selection mechanism that requires that two genes are expressed at the same rate, otherwise the cell dies. We can use two toxic peptides or proteins for a chosen system and express one of them with the standard promoter and the other with test promoters from the library. We engineer the RNA of the toxins such that each one is followed by the reverse complement of the other one (see Figure 2). In this way, whenever

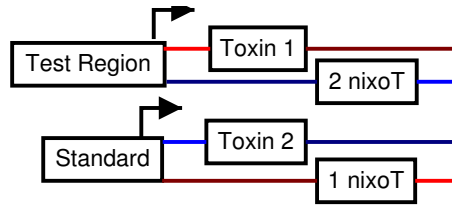


Figure 2: Selection mechanism for matching the output strength of a test region with a standard. The mRNA strands synthesized are complementary for the two promoters.

the levels of the two mRNAs are the same, they would anneal, forming a long dsRNA and be targeted for degradation. Since the dsRNA is very long, it is unlikely that it will trigger an RNAi mechanism. The control for such a system would be to use the same regulatory region for both sequences and checking that the cell live. We can adjust the sensitivity of the system by the choice of the toxins, determining how closely matched the two promoters need to be to avoid killing the cell.

This selection mechanism only allows selecting cells that have a test region with a TIPS levels equal to the standard. A modification to the experimental design should allow comparison between levels that differ  $n$ -fold, where  $n$  is a small integer. Instead of having one copy of the standard regulatory region, we have  $n$ -copies of it, all controlling the same toxin sequence. However, each sequence has a short (30-40 nucleotides) unique tag in front of the coding region. This unique region can be targeted by injecting a specific RNAi. We have to make sure that the RNAi does not interfere with anything else. In this way, we can turn on from one to all of the copies of the same standard module and then match them with the test promoters in the library.

## 2.2 Measuring TIPS

Although the above methods can match TIPS level with an existing standard, a method is necessary for directly measuring the transcription initiation rate of some standards. A couple methods are proposed. One prototype method that has been attempted experimentally in the lab with some success uses *in vitro* transcription with a fluorescent DNA base. Pyrrolo-dC (see [9]) base pairs well with guanine and has different fluorescence depending on whether it is in a double or single stranded DNA. During transcription, the DNA is made single stranded by the RNA polymerase, and, therefore, increasing the fluorescence level. By synthesizing pyrrolo-dC at some location and then performing *in vitro* transcription, the amount of fluorescence can be used as a measure of how much transcription is occurring at that point.

Another method is to use FRET. Fluorophores have been engineered on to DNA polymerase and used as one half of a FRET signal. Similarly, it is possible

to attach a fluorophore to the polymerase and a fluorophore to the DNA or free nucleotides. FRET can then be used to directly measure the transcription initiation events.

### 2.3 Transfer Curves

The goal is to obtain transfer curves for every module in the system. Using a standard library of constitutive promoters of varying input TIPS, it becomes possible to sample points along the transfer curve. For any module, the inputs can be set by replacing the inputs with fixed and known inputs. The output TIPS can then be measured using the proposed methods. Each data point collected in this way represents one point on the transfer curve. Interpolating the entire curve becomes straightforward given enough data points.

### 2.4 Biological Experiments

Creating a library of promoters of various strengths is useful not only for the modeling described here. Many biological experiments involve testing the effect of changing levels of expression of a gene. Most experiments involve either a complete knockout of a gene or high expression of a gene, but points in the middle are traditionally difficult to obtain. Having a library and a quantitative measure of how strong the promoters are in the library makes it useful as a tool for manipulating biological systems. Genes can be expressed with higher or lower transcription initiation levels and the effect upon the system can be studied.

### 2.5 Simulations

A computer simulation framework was written in Java. A network representing the model is read from a file. The initial conditions are either specified directly or set randomly. At each time step, the outputs of every module are set based on its inputs and the module's transfer curve.

The inputs and outputs for every module are measured in TIPS and was arbitrarily set to be limited to the range 0 to 100. Transfer curves for modules were in the following general classes: minimum, random, linear, and sigmoidal. Minimum transfer curves return the minimum value of all the inputs.

Random transfer curves were generated by generating random coefficients for terms of a Fourier series.

$$f(x) = \sum_{i=0}^n c_i \cos(i\omega x + \phi_i)$$

The base frequency  $\omega$  was set to be equal to  $2\pi/100$  so that the period was equal to the maximum TIPS value of 100.  $n$  could be set depending on how many random harmonics were desired, and  $c_i$  and  $\phi_i$  were randomly chosen. With multiple inputs, each input was treated separately in a random Fourier series and all terms were added together.

Linear curves are created by specifying  $n + 1$  coefficients for  $n$  inputs to generate a function of the form:

$$f(x_1, x_2, \dots, x_n) = c_0 + \sum_{i=1}^n c_i x_i$$

Sigmoidal curves were generated using functions of the form:

$$f(x) = a \cdot (\tanh((x - m)/w) + 1)$$

where  $m$  is the midpoint of the sigmoidal curve,  $w$  is a measure of the width or steepness of the curve, and  $a$  determines the amplitude of the curve. These can also be inverted to go from a high initial value to a low final value.

### 2.5.1 Predictions

In real biological experiments, there is no way to obtain the exact and complete transfer curve for modules. Part of the proposed methodology is a way to sample data points along the transfer curve. The goal is that with a set of data points, it would be possible to simulate the network with some degree of accuracy. To simulate this simulation of a network using a limited set of data points, random data points were generated. For each data point, a random module was selected. It was assumed that it would be possible to set the inputs to a discrete and finite set of TIPS values, corresponding to the elements present in the library. The inputs were randomly set to a value from this allowed set of TIPS values. The correct output value of the module given those inputs was calculated and used as a data point.

At each time step, a predicted output for each module was calculated in addition to the true output based on the complete transfer curve. The predicted output for a module was obtained by looking for the two closest data points for that module and interpolating using those two points. The initial values are assumed to be known. The difference between the predicted and the actual values can be calculated as time progresses for each module.

## 3 Results

### 3.1 Phage Lambda

Bacteriophage  $\lambda$  infects *E. coli* cells and has two possible paths. It can either enter the lytic cycle, lysing and killing the cell, or it can enter the lysogenic cycle, integrating its own DNA on to the host chromosome and allowing the host to replicate the phage DNA. This regulatory decision of lysis or lysogeny is well understood [11].

A map of the  $\lambda$  genome is shown in Figure 3. The relevant genes for the lytic decision were organized into a transcriptional model as shown in Figure 4. For clarity and simplicity, separate modules were created for various promoters

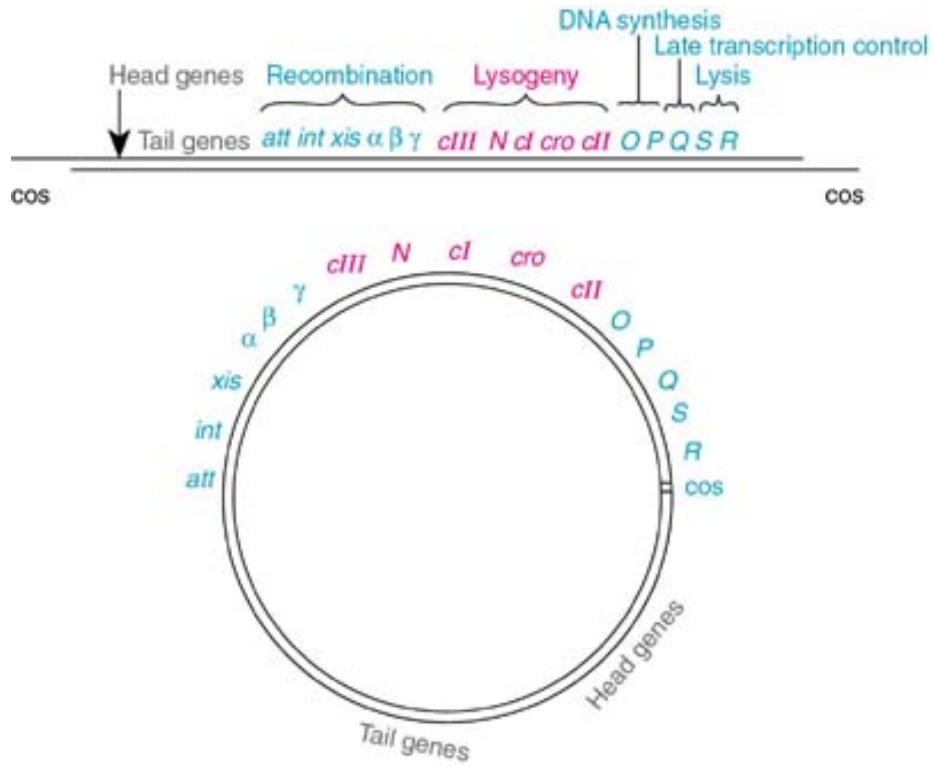


Figure 3: A genome map of  $\lambda$ . Source: [8].

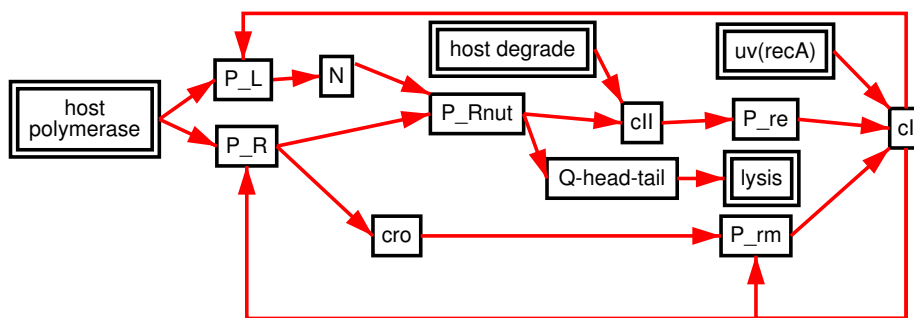


Figure 4: Model of promoters/genes involved in the lytic/lysogeny decision.

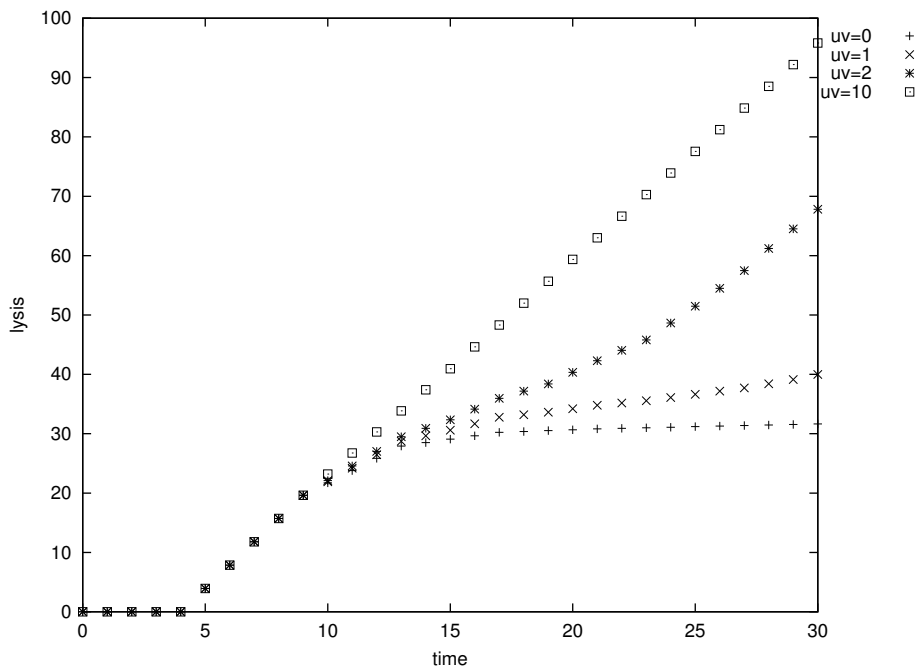


Figure 5: Response of  $\lambda$  lysis decision to UV conditions.

and genes. In addition, there are pseudo-modules for various input parameters to the model. These include the host polymerase transcription rate, which is responsible for the initial transcription of two of the phage's genes, the degradation rate of protein cII by the host, and the amount of ultraviolet stimulation. The output is measured as the cumulative amount of the Q-head-tail produced. The raw data file for the network is attached at the end.

Initially, the transfer curves for all modules were set to be linear relationships. Even with such simple curves, the expected behavior is seen. Normally, the phage enters the lysogeny cycle, where the repressor cI is expressed at high levels and all other genes are turned off. However, if there exists UV stimulation that activates the host's recA protein to destroy repressor cI, we see the phage enter the lytic cycle, with high levels of cro and low levels of cI.

For a slightly more realistic model, as it is known that repressor cI binds cooperatively in activating itself and repressing other genes, the transfer curves can be made into sigmoidal functions to show sharper threshold behavior. There is little qualitative difference on the model's behavior by using sigmoidal functions.

It is known that UV can cause a lysogenic cell to lyse and release phage. In Figure 5, we see the response of the  $\lambda$  circuit to varying levels of UV. The plot suggests that only small amounts of UV are needed to push the phage and the cell to lysis. A small change in the amount of repressor cI is amplified into large

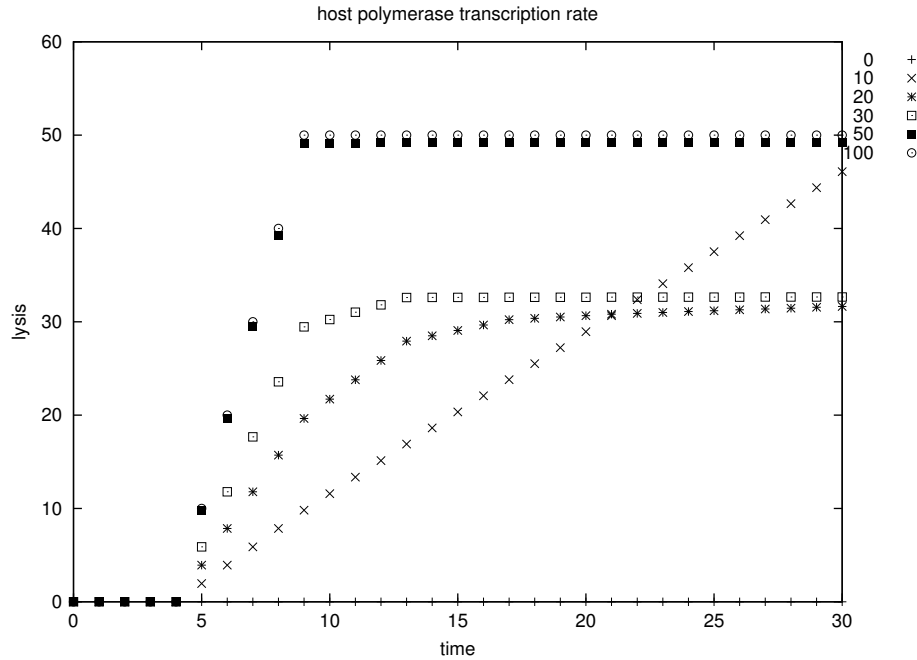


Figure 6: Response of  $\lambda$  lysis decision to initial host polymerase transcription rate.

changes through the various feedback loops.

The exact same response is seen if we increase the rate of the host degradation of  $cII$ . This is expected as  $cII$ 's role in regulating the system is in increasing the transcription of  $cI$ . Increasing the degradation of  $cII$  is equivalent to increasing the amount of UV or decreasing the amount of  $cI$ . Less  $cI$  transcription leads to more  $cro$  and more of the lysis genes.

In a result that was not initially expected, Figure 6 shows how the phage is influenced by the host cell's transcription rate. At a transcription rate of zero, the phage obviously has no impact on the host at all as nothing will be transcribed. At medium to high transcription rates, the curves look similar but can reach different steady state values. However, at low transcription rates, the lysis level increases steadily and does not reach a maximum. What is happening is that the levels of repressor  $cI$  are too low to cooperatively bind and efficiently repress the promoters  $P_L$  and  $P_R$ . This indicates that the host cell must be able to transcribe the phage's repressor at a high enough rate for lysogeny to occur. Weak transcription rates could be one indication of an unhealthy cell and it is logical for  $\lambda$  to lyse a host that would be unsuitable for lysogeny.

Although all the transfer curves were invented and not based on quantitative data, the model appears to be qualitatively correct and can give interesting insights. With real data about the transfer curves that can be collected as

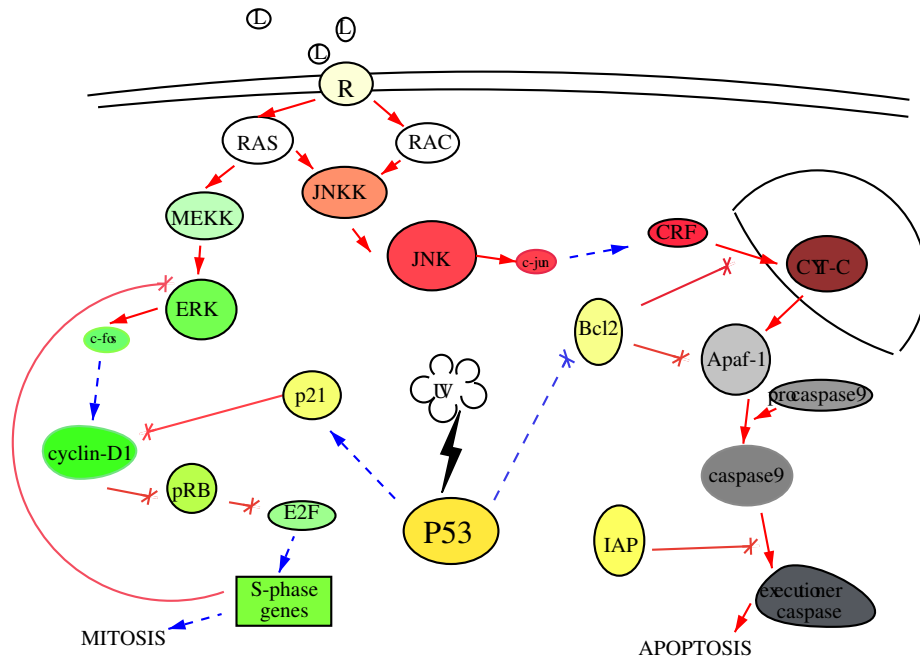


Figure 7: Part of the MAPK signaling pathway involved in mitosis and apoptosis. Solid lines indicate protein-protein interactions and dashed lines indicate transcriptional regulation.

described in the Methods section, this type of model may be useful in further probing the mechanism of  $\lambda$  regulation.

### 3.2 MAPK Signaling Pathway

Taking this modeling approach into a different biological area, we considered modeling a more complicated network that involves not only transcription but also protein-protein interactions. As mentioned before, we do not model directly these protein-protein interactions, but we account for them in the transfer curves.

At the center of our network is the MAPK signaling cascade. We chose to model a network that shows how MAPK cascade transduces an extracellular signal and then makes a decision based on the cell state and several other factors whether to go towards proliferation (mitosis) or towards apoptosis. Figure 7 presents a cartoon of the pathway analyzed.

We consider a growth factor or a stress factor initiating the signaling in our network. This signal is transduced by the MAPK pathway. We chose to separate the two end kinases ERK and JNK given that they control different transcription factors and different functions within the cell [6] [2] [10]. The ERK activates c-fos and this activates transcription of genes involved in cell prolifer-

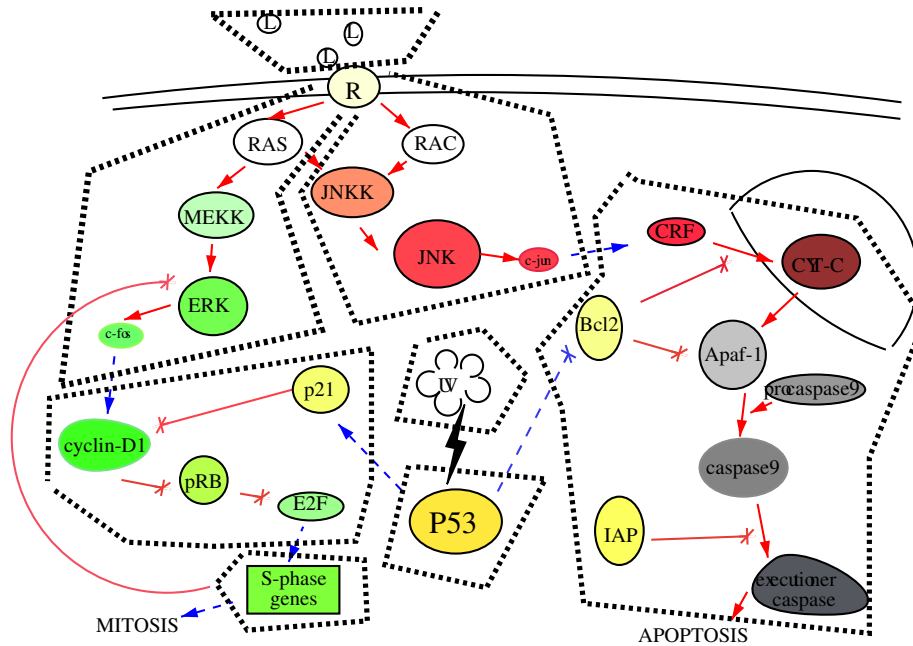


Figure 8: Transcriptional modules in the MAPK signaling pathway.

eration, including cyclin-D1. Cyclin-D1 through its cyclin dependent kinase (CDK) phosphorylates pRB causing it to release E2F, another transcription factor that stimulates transcription of the S-phase genes, eventually promoting mitosis [12]. There is a negative feedback loop included in the S-phase genes, preventing over-signaling of ERK [7].

JNK activates the c-jun transcription factor that transcribes a set of genes involved in stress response. These genes promote cytochrome C release from mitochondria, inducing the apoptosis factor Apaf1 to activate caspase 9. Caspase 9 will in turn activate executioner caspase that completes the apoptosis self destruction [5]. The balance between the pathways is maintained by the p53 protein that blocks cyclin-D1 by transcribing p21 [14]. Also, p53 triggers apoptosis by blocking Bcl2, an inhibitor of Apaf1 [5]. p53 is also a sensor of cellular stress and UV damage, being upregulated in such events [12]. Also, we took into account the inhibitory apoptosis proteins (IAP) that block activation of the executioner caspase as another negative regulator in the apoptosis pathway [5].

We separated the various interacting proteins at the level of transcriptional regulation, creating modules that receive TIPS as inputs and output TIPS (Figure 8). Each module has a transfer curve that matches inputs to outputs. We assume that most proteins that are not transcribed are at a constant level and this level is accounted for in the transfer functions. Figure 9 presents the network in terms of the modules. As in the previous case, we have to use pseudo-TIPS for the input boxes such as growth factor (GF) and UV radiation, and maintain

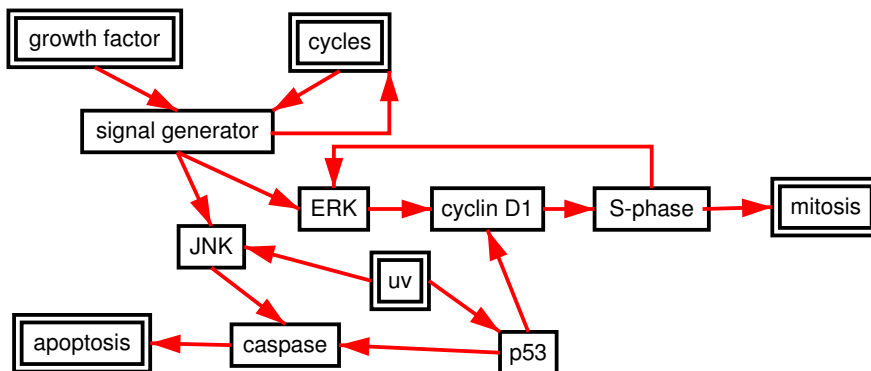


Figure 9: MAPK pathway for mitosis and apoptosis.

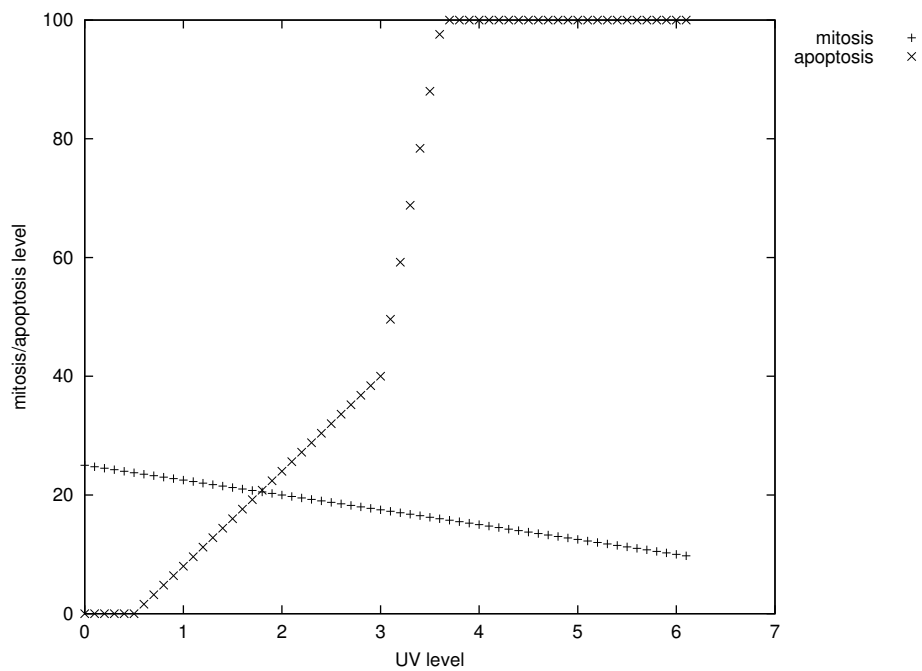


Figure 10: Effect of UV stimulation levels on the response of cell due to the MAPK signaling pathway.

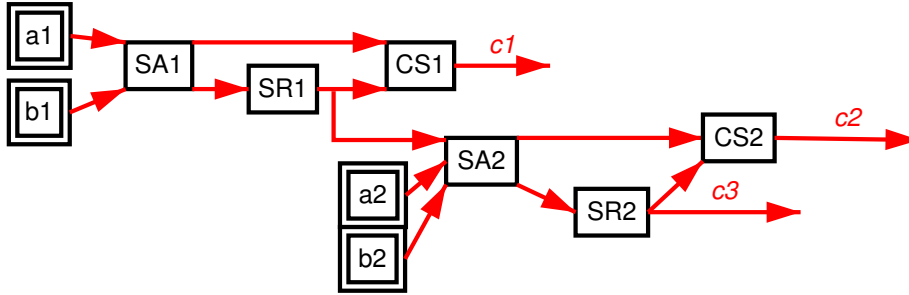


Figure 11: Design of a 2-bit adder ( $a_2a_1 + b_2b_1 = c_3c_2c_1$ )

a cumulative pseudo-TIPS for the outputs, in this case mitosis and apoptosis levels.

We tested several behaviors of this network, using simple linear functions for transfer curves. When signaling is high, both pathways are activated to some extent, mitosis dominating at lower levels of stimulation. At higher levels of stimulus, the cell starts experiencing stress and apoptosis dominates. Of course, this is a very crude modeling of this behavior, but the functions used are very simple. The network can be tuned using real data to model this behavior more closely. Also, if the UV/stress level is increased, the pathway that dominates is apoptosis (Figure 10). Again, this is a crude estimate, because this network is simple and the transfer curves are all linear. However, these data show that this modeling approach can be successful even with complicated pathways that contain many protein-protein interactions.

### 3.3 2-bit Adder

As an example of engineering a synthetic network, a 2-bit adder network was created (Figure 11). The network adds  $a_2a_1 + b_2b_1 = c_3c_2c_1$ . The inputs are the bits  $a_2, a_1, b_2, b_1$  and the output are the  $c$  bits. A digital “1” was defined to be at a level of 20 TIPS and a digital “0” at 0 TIPS.

The first stage of the network is to add the first two least significant bits  $a_1$  and  $b_1$ . The module SA1 outputs the analog sum of  $a_1 + b_1$  and the value  $0.5 \cdot a_1 + 0.5 \cdot b_1 - 5$  to approximate the digital AND of the inputs. To restore this back to a good digital value of either 20 or 0 TIPS, the SR1 (signal restoration) module is a sigmoidal curve that goes from 0 to 20 TIPS. The digital AND is the carry out from the addition of the first two bits. The CS1 module corrects the analog sum by subtracting 2 times the carry to output the correct digital bit  $c_1$ .

The carry from the first two bits and the second two bits,  $a_2$  and  $b_2$ , are added in a similar manner to the first two bits. The final output contains the three bits that represent the digital sum of the inputs.

As expected, this rationally designed network performs the digital addition function when run through the simulator. The input bits when set to either 0

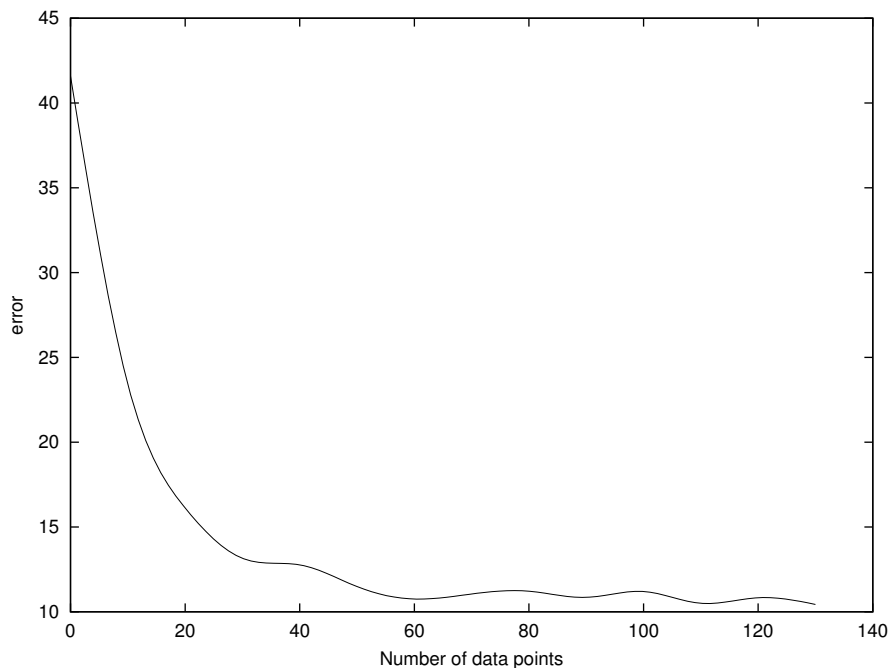


Figure 12: Average error in predictions as a function of the number of data points.

or 20, to represent either a digital 0 or 1, will give outputs close to 0 and 20 representing the correct bits. If the inputs are not exactly 0 or 20, depending on the exact inputs, the output may or may not also contain that noise. This is an example of the differences between digital and analog. The input signals could have been put through signal restoration modules to ensure a higher fidelity, at the cost of additional complexity and time to traverse the network.

### 3.4 Predicting Transfer Curves

The network structure seen in Figure 4 used to model  $\lambda$  was used to test the accuracy of predictions. Random transfer curves involving terms of a Fourier series were used as described previously. As the number of harmonics included is increased, a Fourier series can theoretically approach any reasonable function and certainly any function that can be found in real biological systems.

The accuracy of the prediction for the output lysis value as a function of the number of data points is shown in Figure 12. Each point represents the average of 100 separate simulations with different random transfer curves and errors averaged over 50 iterations for each simulation. The curve levels off around 40 data points and this number does not depend much on the number of harmonics used in the Fourier series.

## 4 Discussion

### 4.1 Comparisons with Other Models

Many other quantitative and useful modeling methods exist. Differential equations require the knowledge of many rate constants that cannot be measured in a standard way. Differential equations often contain more details about low level interactions and models often have more equations than one really cares about. General curve fitting methods and machine learning techniques such as neural networks often produce unintuitive and non-biologically significant functions.

The proposed model is in one sense a more general case of boolean models where the inputs are not binary and the transfer curves are more complex than logic functions. In another sense, boolean models can form more general models as a logical “1” can represent anything and not only a transcriptional event.

Perhaps the key difference is that models based on networks such as Bayesian networks, boolean networks, or neural networks do not have the modularity or flexibility in allowing for easy forward engineering of new networks. The interactions between modules are not in a standard unit like TIPS and so rearranging parts is not possible.

### 4.2 Conclusions

We have proposed a way of thinking of networks as modules that have inputs and outputs measured in terms of transcription rates. By using transcription rate, i.e. transcription initiations per second (TIPS), it becomes unnecessary to model protein levels. This is a good simplification when we do not care about such things, but also means that these models cannot directly model non-transcriptional interactions. Some networks such as metabolic pathways may not be well-suited for this type of approach. Other networks, such as the MAPK signaling pathway, involving many kinases and protein interactions, can be modeled depending on how the modules are defined. These models can be easily understood at an intuitive level and can allow for the engineering of synthetic biological networks.

One of the major benefits of using the transcriptional model is the modularity and ease of design of new systems [1]. The 2-bit adder shows the design of an artificial network, using modules that had the same type of transfer curves found in the other real biological models. If we could construct a library of such modules, creating a biological network to add or perform some other function would become much easier.

A proposed methodology for obtaining a library of standard modules with known output TIPS is given. From this library, it becomes possible to measure the transcriptional rate data needed for a given module, and from this data, to construct the transfer curves. The simulations showed that a small and reasonable number of data points could be used to estimate transfer curves and predict network behavior. Actual data will be needed to validate the feasibility of this approach toward modeling and engineering.

## References

- [1] BioBricks. <http://syntheticbiology.org/biobricks/>.
- [2] BioCarta. MAPKinase signaling pathway. [http://www.biocarta.com/pathfiles/h\\_MAPKPathway.asp](http://www.biocarta.com/pathfiles/h_MAPKPathway.asp).
- [3] Michael B. Elowitz, Arnold J. Levine, Eric D. Siggia, and Peter S. Swain. Stochastic gene expression in a single cell. *Science*, 297:1183–1186, August 2002.
- [4] Eukaryotic promoter database. <http://www.epd.isb-sib.ch/>.
- [5] Martin Fussenegger, James E. Bailey, and Jeffrey Varner. A mathematical model of caspase function in apoptosis. *Nature Biotechnology*, 18(7):768–774, July 2000.
- [6] Catherine A. Hazzalin and Louis C. Mahadevan. MAPK-regulated transcription: A continuously variable gene switch? *Nature Reviews Molecular Cell Biology*, 3(1):30–40, January 2002.
- [7] Sui Huang and Donald E. Ingber. Shape-dependent control of cell growth, differentiation, and apoptosis: Switching between attractors in cell regulatory networks. *Experimental Cell Research*, 261:91–103, 2000.
- [8] Bacteriophage lambda: A complex viral operon. <http://www.msu.edu/course/mmg/431/lambda2.html>.
- [9] Cuihua Liu and Craig T. Martin. Promoter clearance by T7 RNA polymerase: Initial bubble collapse and transcript dissociation monitored by base analog fluorescence. *Journal of Biological Chemistry*, 277(4):2725–2731, January 2002.
- [10] MAPK pathway. <http://www.cmb.uab.edu/courses/Lectures/yoder1.pdf>.
- [11] Mark Ptashne. *A Genetic Switch: Phage  $\lambda$  and Higher Organisms*. Blackwell Science, 2nd edition, 1992.
- [12] R. L. Sutherland, O. W. J. Prall, K. M. Alle, N. R. C. Wilcken, R. Hui, J. R. Ball, B. Sarcevic, S. M. Henshall, E. A. Musgrove, and C. K. W. Watts. Cyclin dependent kinases as downstream targets of oestrogen action: potential prognostic indicators and therapeutic targets. *Endocrine-Related Cancer*, 4(3):357–370, Autumn 1997.
- [13] Transcription regulatory region database. <http://wwwmgs.bionet.nsc.ru/mgs/dbases/trrd4/>.
- [14] Transcriptional regulation. <http://www.mblab.gla.ac.uk/~john/nlt.html>. Lecture Notes.

## 5 Appendix: Data Files

### 5.1 General Format

The general format of the data files used for the simulations is as follows. Lines beginning with a '#' and blank lines are ignored. Modules are specified by three lines. The first line is the module name, the second are the names of the input TIPS, and the third are the output TIPS. The first three lines do not specify a real module, but rather specify the name of the network, the inputs of the network, and the outputs of the network.

The input and output TIPS can have various parameters assigned to them. 'init' specifies the initial value and if an initial value is not explicitly specified, a random value is chosen. 'line', 'sigmoid', 'min', and 'rand' specify the form of the transfer function. Each is first followed by the number of inputs expected and then parameters specific to the type of function. Random and minimum functions have no parameters. Linear functions specify the  $c_i$  in order as given in Section 2.5. Sigmoidal functions specify first the index of the variable that will be used (counting from zero), followed by the initial value of the function, the final value, the location of the midpoint, and the width of the function. Values can be either numbers or a '%n' to represent the value of input  $n$ .

### 5.2 lambda

```
# Model phage lambda
# lysis or lysogeny decision
# Reference: A Genetic Switch 2nd edition by Mark Ptashne

Lambda
Ecoli_RNA_poly:init:20 coli_degrade:init:0 uv_recA:init:0
lysis:init:0

# Promoters
P_L
Ecoli_RNA_poly cI_out:init:0
P_L_out:sigmoid:2:1,%0,0,20,10

P_R
Ecoli_RNA_poly cI_out
P_R_out:sigmoid:2:1,%0,0,20,10

P_Rnut
P_R_out:init:0 N_out:init:0
P_Rnut_out:min:2

P_re
cII_out:init:0
```

```

P_re_out:line:1:0,1

P_rm
cI_out cro_out:init:0
P_rm_out:line:2:0,1,-1

# Genes
N
P_L_out:init:0
N_out:line:1:0,1

cro
P_R_out:init:0
cro_out:line:1:0,1

cI
P_re_out:init:0 P_rm_out:init:0 uv_recA
cI_out:line:3:0,1,1,-1

cII
P_Rnut_out:init:0 coli_degrade
cII_out:line:2:0,1,-1

Q_head_tail
P_Rnut_out:init:0
lyse_in:line:1:0,2

lyse
lyse_in:init:0 lysis
lysis:line:2:0,0.1,1

```

### 5.3 MAPK

```

# MAPK pathway leading to mitosis/apoptosis
# inputs are a growth factor signal, UV damage, and
# inhibitor of apoptosis proteins
# output is mitosis and/or apoptosis signal

```

```

MAPK
total:init:30 decrement:init:5 uv:init:1 iap:init:6
mitosis_level:init:0 apoptosis_level:init:0

```

```

# signal generator is a module put in so that we can have a time-limited growth
# factor signal. It'll be initially on with a value of decrement and the total
# signal will be total

```

```

signal_generator
total decrement
total:line:2:0,1,-1 gf:min:2

ERK
gf:init:0 sphase_out:init:0
erk_out:line:2:0,2,-1

cyclinD1
erk_out:init:0 p53_out1:init:0
cyclinD1out:line:2:0,1,-0.5

sphase
cyclinD1out:init:0
sphase_out:line:1:0,1 mitosis_in:line:1:0,1

mitosis
mitosis_in:init:0 mitosis_level
mitosis_level:line:2:0,1,1

p53
uv
p53_out1:line:1:10,1 p53_out2:line:1:0,1

JNK
gf uv
jnk_out:line:2:0,1,1

caspase
jnk_out:init:0 p53_out2:init:0 iap
apoptosis_in:line:3:0,1,1,-1

apoptosis
apoptosis_in:init:0 apoptosis_level
apoptosis_level:line:2:0,1,1

```

## 5.4 2-bit adder

```

# implements a two bit digital adder
# adds numbers a+b = c
# a2a1 + b2b1 = c3c2c1
# digital 1 is 20 and digital 0 is 0

```

```
Adder
```

```

a2:init:20 a1:init:0 b2:init:20 b1:init:20
c1:init:0 c2:init:0 c3:init:0

# sum 1st bits and do digital AND function
SA1
a1 b1
sum1:line:2:0,1,1 and1:line:2:5,0.5,0.5

# digital signal restoration result of AND back to 0,1
# sharp sigmoidal curve centered around 10
SR1
and1
carry1:sigmoid:1:0,0,20,10,0.5

# correct sum 1st bit by subtracting carry1
CS1
sum1 carry1
c1:line:2:0,1,-2

## now 2nd bit same but with extra carry
SA2
carry1 a2 b2
sum2:line:3:0,1,1,1 and2:line:3:5,0.5,0.5,0.5

SR2
and2
carry2:sigmoid:1:0,0,20,10,0.5

CS2
sum2 carry2
c2:line:2:0,1,-2

# Last carry becomes last bit
LastBit
carry2
c3:line:1:0,1

```