Chapter 4

Robustness in Feed-Forward Loops: Clustering of Responses

Even for models of very simple networks, like those in Chapters 2 and 3, describing the system requires many parameters. These parameters are often unknown or uncertain. Consequently, predicting the response of a gene circuit may require infering gene circuit function from data on circuit structure alone. By using the feed-forward loop as a model system, this chapter introduces a technique for classifying gene circuit function given a set structure. Temporal responses of a comprehensive set of feed-forward loop models are calculated for a range of parameter values. The responses are clustered, and the relation between clusters and circuit types is analyzed. Some designs are robust, producing one unique type of response regardless of parameter selection. Other designs may exhibit a variety of responses, depending upon parameter values.

4.1 Background

As discussed in the Introduction, certain patterns of genetic regulatory interactions occur more frequently than would be expected in randomized networks with similar connection statistics [11]. The feed-forward loop is one such design; an example is regulation of *araBAD* by both the local transcription factor AraC and the global transcription factor CRP in *E. coli* (see review [45] and references therein). Two other naturally occurring feed-forward loops are introduced in Chapter 5.

Given that there are recurring structural designs found in genetic regulatory networks, it is logical to ask: (1) What is the function of a design and (2) why might one design be preferred over others?

First, even once a particular circuit configuration is selected, the function of the circuit is not necessarily transparent. For the feed-forward loop, Mangan and Alon [12] explored several possible circuit functions by using a mathematical model of the feed-forward loop where a signal Sx interacts with X, and a different signal Sy interacts with Y. For a constant level of Sy, they noticed pulsing, ON/OFF, and OFF/ON behaviors in gene expression levels in response to a step input in Sx.

Their efforts produced a preliminary classification of responses for feed-forward loops, but in order to thoroughly characterize feed-forward loop function, it is desirable to explore a larger range of parameters, and to consider circuit types in which the same signal can interact with both X and Y.

Answering the second question—why some designs are preferred—requires an understanding of performance criteria relevant to natural selection in gene circuits [46]. Broad classification of possible circuit functions can eventually help clarify why certain circuit designs are better than others.

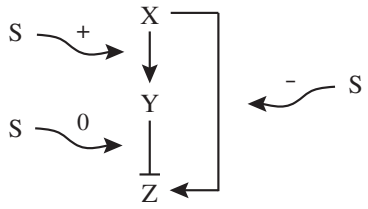

Figure 4.1: Feed-forward loop network motif. X and Y represent transcription factors. Z is the target gene. Activator connections are drawn as normal arrows and repressor connections are drawn as arrows with T-shaped ends. Signal effects are shown with characters {+, -, 0}.

The feed-forward loop network motif, shown in Fig. 4.1, has two transcription factors, X and Y, which control expression levels of a target gene Z. X additionally regulates transcription of Y. Here, the response of the feed-forward loop refers to the expression level of Z as a function of time.

Signaling molecules also play a significant role in gene expression. Signals may be small-molecule metabolites or other molecules that bind to the transcription factor, enabling or blocking its activity.

As in [12], we consider feed-forward loop models in which each of three genetic regulatory interactions can take on one of two possible values (activator, repressor). Unlike their study, which only considers changes in a signal that enables the global activity of X, we consider models in which a signal may have one of three effects (+, -, 0, described below) on each genetic regulatory interaction. Instead of considering just 8 ($= 2^3$), we consider 216 ($= 2^3 3^3$) different ways of wiring a feed-forward loop. Fig. 4.1 is just one example.

4.2 Mathematical Models

The general feed-forward loop is modeled using a pair of nonlinear differential equations:

$$\dot{Y} = \alpha_y \frac{1}{1 + (\frac{S_{yx}X}{K_{yx}})^{n_{yx}}} - \beta_y Y \tag{4.1}$$

$$\dot{Z} = \alpha_z \frac{1}{1 + (\frac{S_{zy}Y}{K_{zy}})^{n_{zy}}} \frac{1}{1 + (\frac{S_{zx}X}{K_{zx}})^{n_{zx}}} - \beta_z Z. \tag{4.2}$$

X is treated as a constitutively expressed protein, modeled here as a constant. α_i is the regulatable transcription rate and β_i is the decay rate through degradation and dilution. S_{ij}, discussed in further

detail below, is a binary value that describes the signal effect. K_{ij} is a threshold value, and the Hill coefficient, n_{ij}, is negative if the connection is an activator and positive if it is a repressor.

Signal interactions are modeled by inserting a binary term, $S_{ij} \in \{0, 1\}$, in the Hill function argument. S_{ij} takes on different values depending upon the level of signal in the environment and the type of signal interaction. Table 4.1 is used to determine S_{ij}.

	Signal < Threshold	Signal > Threshold
+	0	1
-	1	0
0	1	1

Table 4.1: S_{ij} Values

The nonlinear dynamics described in Eqns. (4.1)-(4.2) have a single equilibrium point

$$\begin{aligned} Y_{eq} &= \frac{\alpha_y}{\beta_y} \frac{1}{(1 + (\frac{S_{yx} X}{K_{yx}})^{n_{yx}})} \\ Z_{eq} &= \frac{\alpha_z}{\beta_z} \frac{1}{(1 + (\frac{S_{zy} Y_{eq}}{K_{zy}})^{n_{zy}})} \frac{1}{1 + (\frac{S_{zx} X}{K_{zx}})^{n_{zx}}}. \end{aligned}$$

Linearizing the system and moving the equilibrium point to the origin we find

$$\begin{pmatrix} \dot{y} \\ \dot{z} \end{pmatrix} = \begin{pmatrix} -\beta_y & 0 \\ \frac{d\dot{Z}}{dY}|_{eq} & -\beta_z \end{pmatrix} \begin{pmatrix} y \\ z \end{pmatrix},$$

where $y = Y - Y_{eq}$, $z = Z - Z_{eq}$. Since the Jacobian matrix is lower triangular, the eigenvalues are the diagonal matrix elements ($-\beta_y$ and $-\beta_z$). For realistic biological systems β_y, $\beta_z > 0$, thus the system has a stable equilibrium point with two real eigenvalues.

4.3 Simulations

The initial conditions for all simulations are the steady state values of Y and Z when the signal is below the threshold level. We are interested in the dynamical behavior that results from changing signal levels.

Fig. 4.2 shows the response of one representative feed-forward loop to changing signal levels. The level of transcription factor Y increases to its steady-state value following a decaying exponential curve. Nonlinear effects cause overshoot in Z before it reaches steady state.

There are seven parameters in Eqns. (4.1)–(4.2) that can be varied. The values of α_i and β_i are selected randomly from a specified range ([0.1, 10] for data shown in the following section). To

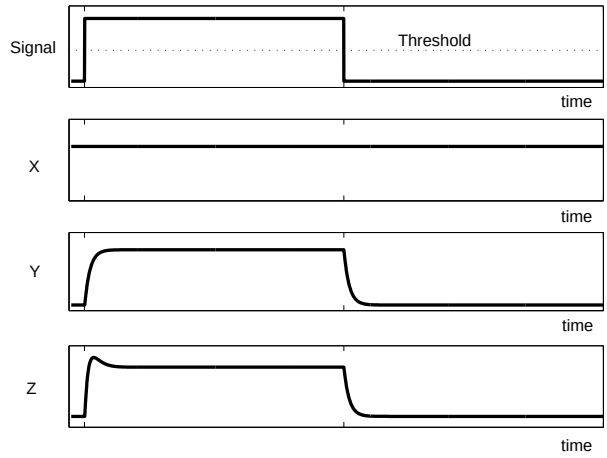

Figure 4.2: Sample response. Simulation results for levels of X, Y, and Z expression as a function of time. The input signal and threshold are shown in the top plot. The configuration of the simulated system is (X-Y, Y-Z, X-Z) = (activator, repressor, activator) and (Signal X-Y, Signal Y-Z, Signal X-Z) = (+, +, +). Parameter values shown here are: $\alpha_i = \beta_i = K_{ij} = 1$, $X = 10$, $|n_{ij}| = 2$.

ensure log-uniform parameter selection, for each parameter, a random number, r, is mapped using

$$10^{(2r-1)log_{10}M_{\alpha\beta}},$$

where $M_{\alpha\beta}$ is the maximum value α_i or β_i can take on (10 for this example). This mapping ensures that we are equally likely to assign values less and greater than 1 to the parameters.

Additionally, the threshold parameters, K_{ij}, are varied. The ratio of transcription factor concentration to the threshold value is the relevant quantity (e.g., $\frac{X}{K_{yx}}$). These three ratios are allowed to take on values less than 1, equal to 1, and greater than 1. All 27 possible combinations are considered.

Recognizing symmetry in signaling effects reduces the size of this problem and decreases computation time. A circuit with signaling interaction type + will respond to an ON pulse in the signal in the same way type − will respond to an OFF pulse.

4.4 Clustering Feed-Forward Loop Responses

Large numbers of feed-forward loops can be modeled with these techniques. For each of the 216 wiring patterns there are multiple threshold and rate parameters that are either unknown or uncertain in biological systems. Broad range limits on parameter values can be assumed to make the problem tractable, but the number of systems remains large.

Although a great number of feed-forward loops can be modeled, many of the feed-forward loop

responses appear to be similar. We use an automated clustering algorithm to classify responses into different categories based upon their similarity.

4.4.1 Clustering Algorithm

A greedy approximation algorithm was used to cluster responses [47]. The algorithm uses a metric $d(x, y)$ that characterizes the distance between x and y. Given an input of a set X of n points $x_1,\ ...\ x_n$ and a metric d on X, we want to find a set C of K points $c_1,\ ...\ c_K \in X$ that minimizes $\max_{1 \le i \le n} d(x_i, C)$. In other words, we want to cluster the points into K different groups where the size of the largest cluster is as small as possible.

This K-center clustering problem is NP-hard in general, but the approximation algorithm can quickly compute clusters where the maximum error is within a factor of two of the actual solution [47]. Thus, the maximum radius of all clusters is, at worst, two times larger than it needs to be.

The clustering algorithm is performed as follows: First, K points must be selected as cluster centers. The first center, c_1, is chosen at random. After that ($i = 2,\ ...\ K$) let c_i be the point x of X that maximizes $d(x, \{c_1,\ ...\ c_{i-1}\})$. This is equivalent to assigning all the remaining non-center points to clusters, determining which is furthest from its center point, assigning that point as a new center, and throwing the rest of the points back into the pool of non-centers. After all K centers have been assigned, the remaining points $x_{K+1},\ ...\ x_n$ are assigned to clusters.

This algorithm is used to cluster feed-forward loop responses. Defining a distance measure, d, is the primary complication in extending the clustering algorithm to the present task. Each response is a vector $z \in \mathbb{R}^N$ where the vector contains the values of Z running from $t = 0$ to $t = N - 1$.

A correlation coefficient is used to measure the distance between two response vectors, z_1 and z_2:

$$d(z_1, z_2) = \frac{1}{2} - \frac{\langle z_1 - \bar{z_1},\ z_2 - \bar{z_2} \rangle}{2\ \ \|z_1 - \bar{z_1}\|_2\ \ \|z_2 - \bar{z_2}\|_2}, \tag{4.3}$$

where $\bar{z}$ is the mean of z, brackets denote the dot product, and $||\cdot||_2$ is the 2-norm. This distance function is designed so that $d(z_1, z_2) = 0$ if $z_1 = z_2$ and $d(z_1, z_2) = 1$ if the two signals are very different. Note that the distance function evaluates to zero for responses that differ only by a multiplicative scaling factor and an offset.

4.4.2 Maximum Error versus Number of Clusters

The maximum error is defined as the largest cluster "radius," $\max_{1 \le i \le n} d(x_i, C)$. As the number of clusters is increased, the maximum error goes down (Fig. 4.3). This value can be plotted as a function of K, the number of clusters. At $K = 1$, we will have a large maximum error value unless

the feed-forward loop responses are all nearly identical. For K = total # of responses, we will have no error since every response is associated with its own individual cluster. If the intermediate curve drops quickly then a small number of clusters can describe almost all of the data.

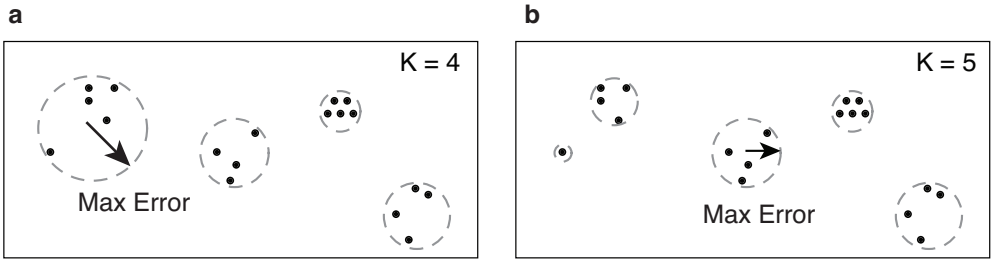

Figure 4.3: Schematic diagram of maximum error for (a) $K = 4$ and (b) $K = 5$. As the number of clusters increases, the maximum cluster error decreases.

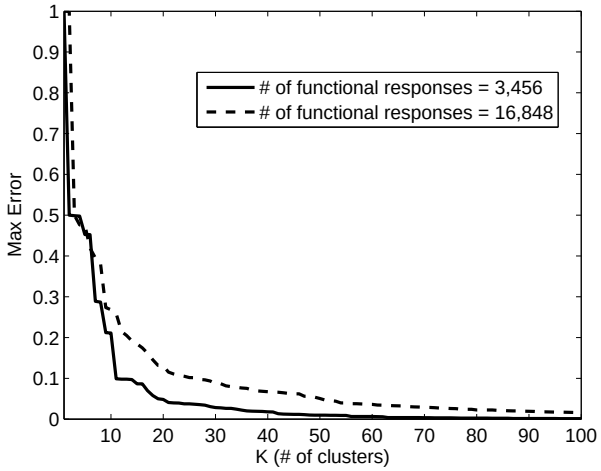

Figure 4.4: Maximum error versus the number of clusters. The results of clustering on two distinct data sets are shown.

Fig. 4.4 shows that, even for large numbers of feed-forward loop responses, the maximum error drops off rapidly with increasing cluster numbers.

4.4.3 Singular Value Decomposition

Since each cluster may contain a large number of responses – thousands in some cases – it would be convenient to have a simple way to represent data. Singular value decomposition is used to generate a representative trace that describes the most significant principal component of all of the responses in a cluster.

Singular value decomposition has been used in other biological applications to compress data into a simplified, more understandable form [48]. In this work the singular value decomposition of

a matrix $A \in \mathbb{R}^{M \times N}$ is taken:

$$A = USV^T.$$

M is the number of feed-forward loop responses we are comparing and N is the number points in time. S, U, and V come from the standard definition of singular value decomposition.

The first right singular vector (the first column of V) is the singular vector associated with the largest singular value. This vector describes the principal component of all of the response data listed in the A matrix and provides a single representative response to associate with a cluster.

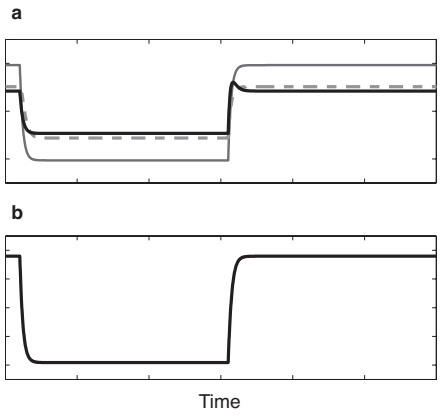

Figure 4.5: Singular value decomposition example. a) Three representative $Z(t)$ responses from one cluster. b) The first right singular vector of a matrix containing all responses from that cluster.

An example of how singular value decomposition can be used to represent many responses is shown in Fig. 4.5. Fig. 4.5a shows 3 responses plotted on top of each other. In reality, this is a small subset of all responses that fall into this cluster type. The primary singular vector associated with the complete set of responses is shown in Fig. 4.5b.

4.5 Results

The clustering approach associates the responses of the 216 feed-forward loop models with a small number of distinct patterns. These patterns can be used to classify the behavior of an individual circuit over a range of parameter values. The number of clusters it takes to describe a particular circuit configuration can be used as a measure of how robust a circuit is to parameter variation.

4.5.1 Representative Cluster Traces

A relatively simple example is presented to illustrate the utility of clustering. The data shown in Fig. 4.7 are the result of clustering on a set of 3,456 responses. All 216 possible circuit configurations

are represented. Within each configuration only parameters α_i and β_i from Eqns. (4.1)–(4.2) are varied.

$K = 11$ clusters is chosen as a cutoff point because the maximum error is acceptably small (see Fig. 4.4). Beyond this point additional cluster types represent similar responses but with differing temporal characteristics. For example, the rise times, settling times, and overshoot behavior may be different for the additional cluster types. The utility of clustering lies in its ability to segregate responses into broad class types, allowing for a qualitative understanding of possible circuit functionality. In particular, this method will be useful for considering circuits that have more complicated responses (e.g., responses to input signals that are more complicated than a step function).

Fig. 4.7 shows representative singular vectors from each of the 11 clusters. These are the responses, $Z(t)$, to the input signal shown in Fig. 4.6.

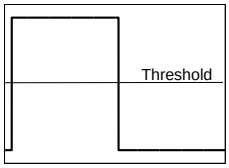

Figure 4.6: Signal level as a function of time.

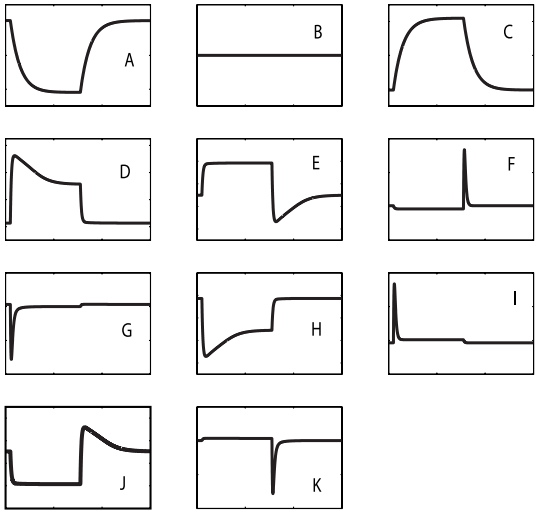

Figure 4.7: Representative responses (Z versus $time$) from each of 11 clusters. Letter labels are used for reference in Table 4.2.

When the threshold values, K_{ij}, are varied in addition to α_i and β_i the result is a large set of functional responses that do not segregate as logically into individual clusters. Even when this is the case, the clustering technique can still be applied to yield a qualitative picture of possible responses. The case with 16,848 responses shown in Fig. 4.4 corresponds to a widely explored range

of parameters, but the maximum error still drops off rapidly. If an acceptable error value is chosen, clustering can be performed to within this margin of error.

In an exploration of the more complete parameter space, the cluster types seen in Fig. 4.7 are preserved, but several additional clusters are added. For example, selecting $K = 15$ clusters produces the four additional cluster types shown in Fig. 4.8.

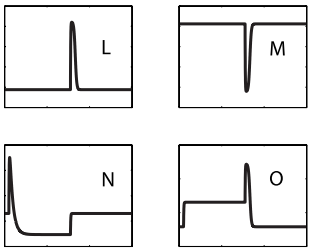

Figure 4.8: Representative responses for cluster types added when additional parameters are varied.

Even when exploring the complete parameter space, some system configurations fall into the same cluster type regardless of the parameter values selected for α_i, β_i, and K_{ij}. The responses of these genetic regulatory configurations are particularly robust to parameter changes.

4.5.2 Distributions of Responses

Table 4.2 lists cluster types for various system configurations. Each row corresponds to one particular configuration: a set of genetic regulatory and signal interaction types. The columns labeled A–O correspond to the cluster types labeled in Figs. 4.7 and 4.8. The numbers in the row tell which cluster this system's responses fall into. For some system configurations, varying parameter values causes the response to fall into different clusters. The rows are normalized by the total number of cases that were run. The entries shown in Table 4.2 are a subset from a larger table, given in Appendix C.

Table 4.2's entries do not indicate how "different" responses are within a cluster. Selecting the number of clusters (Fig. 4.4) sets the upper bound on the error within each cluster. For the 15 cluster case, all responses within a cluster are within a distance of 0.18 of each other, as measured by Eqn. (4.3).

The entropy of each response distribution in Table 4.2 is calculated by using the standard definition of Shannon entropy:

$$-\sum_{i=1}^{15} p(i)\ log_2\ p(i),$$

where $p(i)$ is the percentage of responses that fall into cluster i.

Rows which have a 1 associated with one cluster type and 0s for all the rest (entropy = 0) are

X-Y	Y-Z	X-Z	Signal X-Y	Signal Y-Z	Signal X-Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	Entropy
act	rep	rep	+	+	+	1.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
act	rep	rep	+	+	0	1.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
act	rep	rep	+	+	-	0.41	0	0.51	0.02	0	0	0	0	0.01	0	0	0	0	0.06	0	1.41
act	rep	rep	+	0	+	1.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
act	rep	rep	+	0	0	1.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
act	rep	rep	+	0	-	0.38	0	0.51	0.03	0	0	0	0	0.01	0	0	0	0	0.07	0	1.49
act	rep	rep	+	-	+	0.90	0	0	0	0.09	0	0	0	0	0	0	0	0.01	0	0	0.52
act	rep	rep	+	-	0	0	0.11	0	0	0.06	0	0	0	0	0	0.50	0	0.33	0	0	1.61
act	rep	rep	+	-	-	0	0	1.00	0	0	0	0	0	0	0	0	0	0	0	0	0
act	rep	rep	0	+	+	1.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
act	rep	rep	0	+	0	1.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
act	rep	rep	0	+	-	0.48	0	0.52	0	0	0	0	0	0	0	0	0	0	0	0	1.00
act	rep	rep	0	0	+	0.99	0.01	0	0	0	0	0	0	0	0	0	0	0	0	0	0.05

Table 4.2: Percentage of cluster types exhibited by circuit configurations. Subsection of a larger table, given in Appendix C. *act* = activator, *rep* = repressor.

particularly robust because parameter variations do not change the function of the system. Exact entries in the table are dependent upon the details of parameter selection and the distance measure used to cluster data.

Clustering provides a logical grouping of response types without prior knowledge of the behavior that a network may exhibit. The response pattern of a cluster can then be interpreted in a biologically meaningful way. For example, cluster A (Fig. 4.7) is associated with repressible circuits, where gene expression decreases upon an increase in signal level. Similarly, cluster C is associated with inducible circuits, where gene expression increases upon signal increase. For circuits associated with cluster B, gene expression is unresponsive to changes in signal. Pulsed gene expression responses, both with and without steady state changes, are seen in the remaining cluster types.

This chapter presents a method for identifying functional capabilities of a genetic network given its structure. In our analysis of feed-forward loop models, responses were organized into a relatively small number of clusters. Some feed-forward loop types show non-robust behavior, suggesting that these circuits do not have unique information processing roles. This clustering technique allows for such quick, qualitative intuition into the function of a system. Insight from clustering will be particularly helpful if the state space and parameter space are even larger than those presented in the feed-forward loop example here.

Although we consider models of feed-forward loops in isolation, in nature gene circuits are embedded within the context of the entire molecular network of the cell. Nevertheless, considering isolated gene circuit models can reveal insights into the cellular response to signals. Such models have already proved to be useful in design of synthetic gene circuits, for example, in the design of a toggle switch [49], an oscillator [40], and a circuit whose design may be selected to exhibit either toggle switch or oscillatory behavior [50]. The present technique can help to narrow down which system types and parameter ranges exhibit a desirable behavior, given a broad class of possible designs.

In the future it will be interesting to explore the implications of robustness of responses in real biological systems. In particular, is robustness necessarily a desirable trait for a genetic circuit? If the circuit is locked into one role it may not be capable of evolving in alternative environments. In addition, natural selection can act to enhance the populations of organisms that are sensitive, rather than robust, to mutations in gene circuits. This process has been used previously to explain patterns in the use of activator and repressor control in natural genetic regulatory interactions [51]. It would be interesting to consider tradeoffs involving robustness in the context of the evolution of feed-forward loop configurations and other aspects of gene circuit design.