APPLICATION AND PROPERTIES OF DYNAMICAL STRUCTURE FUNCTIONS

by

Russell E. Howes

Submitted to Brigham Young University in partial fulfillment
of graduation requirements for University Honors

Department of Computer Science

March 2008

Advisor: Sean Warnick
Signature: ____________________

Honors Representative: Bruce Collings
Signature: ____________________

ABSTRACT

APPLICATION AND PROPERTIES OF DYNAMICAL STRUCTURE FUNCTIONS

Russell E. Howes

Department of Computer Science

Bachelor of Science

This thesis describes the motivation for dynamical structure functions in systems theory, proves some of their theoretical properties, and illustrates their utility in the identification of complex dynamical systems. Dynamical structure functions offer a framework for representation of linear, time-invariant (LTI) networks that provides information on the dynamical effects of the observed states on the inputs and other observed states, either directly or through hidden states. This representation can be contrasted with the transfer function, which only describes dynamical correlations between the inputs and the observed states in a network. We review the motivation, definitions, and derivations behind dynamical structure functions and develop some deeper theoretical results on the properties of dynamical structure; specifically, that any transfer function is arbitrarily close to a transfer function for which there exists a minimal, decoupled realization. We build upon previous work relating to information necessary to fully characterize the network when given input-output data

and define a series of operations on the system, derived from common biological experiments, with which we can determine the dynamical structure.

We discuss an implementation of our predictive method, and detail future work in comparing this method's performance with that of several other, commonly-used predictive methods.

ACKNOWLEDGMENTS

I would like to thankfully recognize the assistance, vision, and support of my advisor, Sean Warnick of the Brigham Young University Computer Science Department, while working on this project. Our collaborator, Jorge Gonçalves of the Department of Engineering at Cambridge University, was instrumental in assisting with my research visit to Cambridge, where we made a lot of progress on the reconstruction problem; I thank Jorge and his students Neil Dalchau and Lee Eccleston for their knowledge, insights, and collaboration.

This research was supported by the National Science Foundation, the Brigham Young University Office of Research and Creative Activities, and the Honors Department.

Contents

Chapter 1

Introduction

One of the fundamental issues in the identification of dynamical systems is that of accurately determining a system's structure; that is, the presence or absence of causal relationships, or "connections", between any two variables in a system. The process of determining a system's structure is called reverse engineering and involves algorithms that take available data about a system's dynamics and return the algorithm's 'best guess' at system structure, usually in the form of a matrix, graph, or list of equations. There are various ways in which structural knowledge can be useful in analyzing a system. For example, a large system can sometimes be highly connected within smaller subsystems. Accordingly, we can decompose the network and look at the smaller systems. Examples in which knowledge of structure can aid in analyzing large systems include business and academic organizations (where each department can be highly interconnected but connections between departments are more sparse); terrorist or drug trafficking operations (intelligence operations can usually identify leadership and partnerships in such organizations based on frequency of communication); and population networks (epidemiologists use network structure to predict the rate a disease will spread through a given area based on its population and proximity to travel routes).

The structure identification problem has generated high interest among biologists studying genetic regulatory networks. As DNA microarray experiments become less expensive and more widely used, obtaining dynamical data from a given biochemical network has become cheaper and easier. However, accurate identification of these networks is difficult due to reasons such as inaccuracies in data measurements, lack of structural uniformity among cells of a single organism or cell type, and very high levels of system complexity. Reconstruction methods are constantly evolving to better take into account these challenges. Some of these methods emphasize predicting system dynamics (predicting reactions to system inputs) over structure (connections among states). This is adequate for some applications, but accurate knowledge of system structure can be useful–for example, in designing a drug to stimulate or inhibit a certain, undesired reaction without affecting other reactions.

The authors in [1] introduce dynamical structure functions as a rigorous description of system structure, in terms common to engineering and control theory. This thesis extends the theory introduced in [1], and illustrates a potential method for reverse engineering a system's dynamical structure function, using experiments known and available to biologists. The next section begins by stating the main points of [1] as they define dynamical structure, and relates the dynamical structure of an LTI system to its state-space representation and transfer function. We discuss concepts relating to minimality, sparsity, and decoupledness of systems, and show sufficient conditions on a transfer function for it to have a minimal realization that is completely decoupled. These sufficient conditions illustrate that any transfer function is arbitrarily close to one which has a decoupled minimal realization. We then summarize some of the popular methods used in identifying biological networks, analyzing their strong and weak points. The following section details a reconstruction method which uses properties of dynamical structure to identify the structure and dynamics of a biological network. This work concludes by discussing limitations of our work, and future directions for research in this area.

Chapter 2

Theory of Dynamical Structure Functions

The network structure of a dynamical system is a description of the causal dependencies between system variables. These dependencies are typically represented by a directed graph where variables of the system are nodes, and an arrow between nodes indicates a causal relationship between variables. (See Figure 2.1) This graph is also called the "network topology" [2] or "connection topology", and is sometimes referred to as Boolean structure. In such cases, the actual 'network structure' is described by attaching numerical parameters to each arrow in the Boolean structure.

The transfer function of a linear, time-invariant (LTI) system is an input-output representation of the system. It is not surprising that the transfer function does not fully characterize the internal structure of a system. One might think, however, that some structural information could be derived from the transfer function. Nevertheless, as shown in [1], it turns out that every transfer function G has a state-space realization that is consistent with any possible internal structure.

We illustrate this fact with a simple example. Consider a system with the following transfer function:

$$G(s) = \begin{bmatrix} \frac{1}{s+1} \\ \frac{1}{(s+1)(s+2)} \end{bmatrix}. \tag{2.1}$$

It can be shown that this transfer function is consistent with two systems with very different internal structures, given by

$$A_1 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -2 & 1 \\ 0 & 0 & -1 \end{bmatrix}, \; B_1 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \; C_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

and

$$A_2 = \begin{bmatrix} -1 & 0 \\ 1 & -2 \end{bmatrix}, \; B_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \; C_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

The networks in Figure 2.1 correspond to each of the indicated realizations of G. It is easy to show that the remaining pair of distinct internal structures ($y_1 \leftarrow y_2$ and $y_1 \rightleftharpoons y_2$) can likewise be obtained from suitable realizations of G.

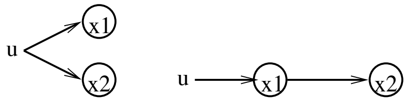

Figure 2.1 Two possible networks given the transfer function: decoupled internal structure (left) and coupled internal structure (right).

It seems plausible that specification of the system order may lead to network reconstruction. Nevertheless, order in and of itself is not enough information to reconstruct the system network from its transfer function. An exception to this is when a bijective relationship between the measured outputs and the system states is known to exist. In this case, the network structure is determined precisely by the transfer function. However, measurement of the entire state vector is unreasonable in most situations.

Not even knowledge of minimality (a system's order being the lowest possible for that transfer function) is sufficient to recover system structure from its transfer function. As the following example illustrates, minimal realizations of simple systems with known output equations can have starkly different network structures. The following transfer function

$$G(s) = \frac{1}{s+3}\begin{bmatrix} \frac{1}{s+1} \\ \frac{1}{s+2} \end{bmatrix}$$

is consistent with systems with very different internal structures, for example

$$A_1 = \begin{bmatrix} -1 & 0 & 1 \\ 0 & -2 & 1 \\ 0 & 0 & -3 \end{bmatrix}, \; B_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix},$$

$$A_2 = \begin{bmatrix} -2 & -1 & 1 \\ -1 & -3 & 1 \\ 0 & -1 & -1 \end{bmatrix}, \; B_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix},$$

and

$$C_1 = C_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

The networks in Figure 2.2 correspond to each of the indicated realizations of G. Note that both realizations are minimal.

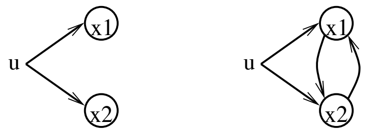

Figure 2.2 Two possible networks corresponding to minimal realisations of the transfer function: decoupled (left) and coupled (right) internal structure.

This inability of the transfer function to describe system structure, even under assumptions of minimality, is the driving motivation behind dynamical structure functions.

2.1 Derivations

We consider the LTI system given by

$$\begin{aligned} \begin{bmatrix} \dot{y} \\ \dot{x}_h \end{bmatrix} &= \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}\begin{bmatrix} y \\ x_h \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u \\ y &= \begin{bmatrix} I & 0 \end{bmatrix}\begin{bmatrix} y \\ x_h \end{bmatrix} \end{aligned} \tag{2.2}$$

where $x = [\ y' \quad x_h'\ ]' \in \mathbb{R}^n$ is the full state vector, $y \in \mathbb{R}^p$ is a partial measurement of the state, x_h are the $n-p$ "hidden" states, and $u \in \mathbb{R}^m$ is the control input. (In this work we restrict our attention to situations where output measurements constitute partial state information.) [1] Note that in many applications the most sensible description of the system is in terms of the measured outputs as states, although rarely can we measure all the states of the system.

We have seen that the state-space realization of a system completely determines structure. Nevertheless, this refined description of the structure is too detailed–it will be as hard to recover from the transfer function as the state-space description itself. We would like a notion of network structure at the resolution of our measurements, something that suppresses information about the hidden states but accurately captures the interaction structure between measured states (internal structure) and the inputs and measured states (control structure). We will now derive expressions for these structural representations.

Taking Laplace Transforms of the signals in (2.2), we find

$$\begin{bmatrix} sY \\ sX_h \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} Y \\ X_h \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} U \tag{2.3}$$

Solving for X_h gives $X_h = (sI - A_{22})^{-1} A_{21} Y + (sI - A_{22})^{-1} B_2 U$. Substituting into the first equation of (2.3) then yields $sY = WY + VU$, where $W = A_{11} + A_{12}(sI - A_{22})^{-1} A_{21}$ and $V = A_{12}(sI - A_{22})^{-1} B_2 + B_1$. Let D be the matrix with the diagonal term of W, i.e. $D = \text{diag}(W_{11}, W_{22}, ..., W_{pp})$. Then, $(sI - D)Y = (W - D)Y + VU$. Note that $W - D$ is a matrix with zeros on its diagonal, and that D is a diagonal matrix of proper rational functions. We then have

$$Y = QY + PU \tag{2.4}$$

where

$$Q = (sI - D)^{-1}(W - D) \tag{2.5}$$

and

$$P = (sI - D)^{-1} V \tag{2.6}$$

The matrix Q is a matrix of transfer functions from Y_i to Y_j, $i \neq j$, relating each measured signal to all other measured signals (note that Q is zero on the diagonal). Likewise, P is a matrix of transfer functions relating each input to each output without depending on any additional measured state Y_i.

G, as mentioned, is a matrix whose terms represent the effect of each input u_j on each output (measured state) y_i. However, this does not distinguish direct or indirect relationships, and each term includes interactions with many other states of the system, including other measured states. On the other hand, P_{ij} denotes the direct impact of the j^{th} input on the i^{th} output, where 'direct' is understood to mean exclusive of the other measured states. The terms of P are strictly proper rational functions in s, as are the terms in G, but the dynamics of P represent the action of some hidden states or that of the one corresponding measured state, but never other measured states. This precipitates the following definition:

Definition 1. *Given the system (2.2), we define the* dynamical structure function *of the system to be (Q,P), where Q and P are the* Internal Structure *and* Control Structure*, respectively, and given as in (2.5) and (2.6).*

Some important properties that follow from this definition and the preceding discussion include:

Lemma 1. *The dynamical structure function (Q,P) of any system (2.2) exists and is unique. It is related to the transfer function, G, of the system by*

$$G = (I-Q)^{-1} P. \tag{2.7}$$

Lemma 2. *Consider the system (2.2) with $p < n$. The dynamical structure function (Q,P) is invariant to changes of coordinates on the hidden states, $z_h = Tx_h$ with T invertible.*

The change of coordinates yields a new system given by

$$\begin{aligned} \begin{bmatrix} \dot{y} \\ \dot{z}_h \end{bmatrix} &= \begin{bmatrix} A_{11} & A_{12}T^{-1} \\ TA_{21} & TA_{22}T^{-1} \end{bmatrix} \begin{bmatrix} y \\ z_h \end{bmatrix} + \begin{bmatrix} B_1 \\ TB_2 \end{bmatrix} u \\ y &= \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} y \\ z_h \end{bmatrix}. \end{aligned}$$

Construction of the dynamical structure function for this system reveals it to be precisely that of the untransformed system (2.2). ■

In particular, the dynamical structure function is invariant to any change of coordinates (and corresponding change of structure) only involving the hidden states, showing that this information is suppressed in this description of the system.

The preceding two lemmas show that first, the dynamical structure function of a system contains more information than the transfer function, and second, the state-space representation contains more information than the dynamical structure function. (see Figure 2.3)

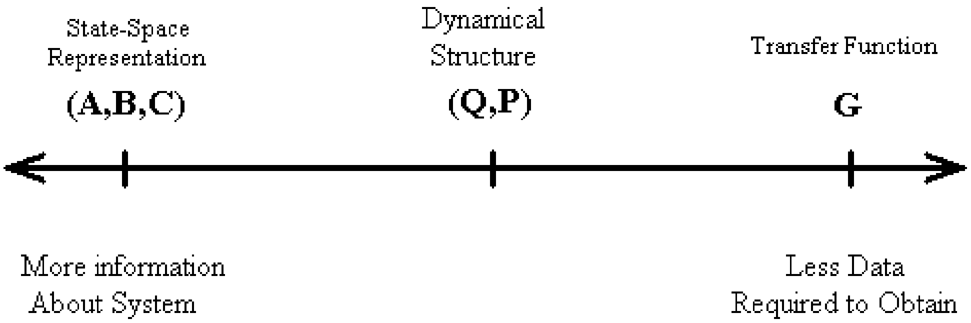
State-Space Representation
(A,B,C)
Dynamical Structure
(Q,P)
Transfer Function
G
More information About System
Less Data Required to Obtain

Figure 2.3 Relation of dynamical structure to state-space and transfer function representations of a network

2.2 Sparsity of internal structure

Real biological networks with large numbers of parameters are generally not highly connected, rather having a 'sparse' connection topology with relatively few direct relations among other observed states in the network [2]. It is of interest to learn how sparse a network might be given input-output data. We examine necessary and sufficient conditions for a transfer function to have a realization with no direct relations among the hidden states. We start by defining this in terms of dynamical structure.

Definition 2. *A system is said to be* completely decoupled *if* $Q = 0$.

We state a property of transfer function matrices, as given in [3], which we will use often in proving the next few theorems:

Definition 3. *The* characteristic polynomial *of a proper rational transfer function matrix G is the least common divisor of the denominators of all minors of G. The* degree *of G, or* $\delta(G)$*, is the degree of the characteristic polynomial of G.*

(Recall that a minor of a matrix A is the determinant of some square submatrix of A.)

Theorem 1. *A realization* (A,B,C,D) *of G is minimal if and only if* $\dim A = \delta(G)$.

The requirement that $\dim A = \delta(G)$ is then equivalent to the condition that the realization is both controllable and observable. Since we are only considering systems with form shown in 2.2, we assume that $D = 0$ and $C = [I\ 0]$, and refer to a system as (A,B).

For the remainder of the chapter, let Q and P be matrices of strictly proper rational transfer functions, of size $p \times p$ and $p \times m$ respectively, where all diagonal terms of Q are 0. (We will call such a (Q,P) a *dynamical structure pair*.) Let D be a $p \times p$ diagonal matrix of proper rational functions.

Lemma 3. *Given Q, P, and D as above, with* $W = (sI - D)Q + D$ *and* $V = (sI - D)P$*, there exists a realization* (A,B) *satisfying*

1. $W = A_{11} + A_{12}(sI - A_{22})^{-1}A_{21}$
2. $V = B_1 + A_{12}(sI - A_{22})^{-1}B_2$

Proof. Since P and Q are strictly proper and D is proper, W and V are proper. Thus there exists a realization of the transfer function

$$\begin{bmatrix} W & V \end{bmatrix} = D' + C'(sI - A')^{-1}B' = \begin{bmatrix} A_{11} & B_1 \end{bmatrix} + A_{12}(sI - A_{22})^{-1} \begin{bmatrix} A_{21} & B_2 \end{bmatrix} \tag{2.8}$$

Let (A', B', C', D') be any realization of $[W\ V]$. Then $A' = A_{22}$, $B' = [A_{21}\ B_2]$, $C' = A_{12}$, and $D' = [A_{11}\ B_1]$. □

We showed in Lemma 3 that for every D there exists a corresponding realization (A, B). In the discussion after (2.3) we derive an admissible D from any system (A, B). As a result, D parameterizes classes of systems (A, B) with fixed dynamical structure (Q, P).

Finding a realization that is *minimal* for a particular dynamical structure consists of choosing a D (not necessarily unique) that will minimize the degree of the transfer function matrix $[WV]$. This may or may not be a minimal realization of G. We wish to obtain conditions stating when a lowest-order realization of the decoupled dynamical structure $(0, G)$ is a minimal realization of G.

Lemma 4. *Given a matrix D as described above, if all denominators of D are relatively prime to all terms in P, the degree of the transfer function matrix $(sI - D)P$ is equal to the degree of $H = [D\ (sI - D)P]$.*

Proof. We will prove that the characteristic polynomials of H and $(sI - D)P$ are equal, by showing that the denominator of any minor of one matrix divides the characteristic polynomial of the other. First, any minor of $(sI - D)P$ is a minor of H, and so divides the characteristic polynomial of H.

Now let us consider any $r \times r$ minor x of H. If all r columns of H come from columns of $(sI - D)P$, x is a minor of $(sI - D)P$. Suppose that $m < r$ columns come from columns of $(sI - D)P$ (meaning the other $r - m$ columns are columns of D.) All columns of D contain at most one nonzero term, so x is either 0 or the product of $r - m$ terms of D and an $m \times m$ minor x' of $(sI - D)P$ (or 1, if $m = 0$). Since the denominator of each term in D is unique to its row, and is found in all terms in the corresponding row of $(sI - D)P$, each of the $r-m$ denominators from D divide the characteristic polynomial of $(sI-D)P$ and are relatively prime to x', which also divides the characteristic polynomial of $(sI-D)P$. Thus their product, x, divides the characteristic polynomial of $(sI-D)P$. □

We are now ready to find sufficient conditions on G for it to have a completely decoupled minimal realization.

Theorem 2 (Decoupled minimal realization)**.** *If each row of a $p \times m$ transfer function matrix G contains an element with a pole that is unique in its column, then G has a minimal realization whose dynamical structure is equal to $(0, G)$. For a single-input system ($m = 1$), the condition is necessary as well as sufficient.*

Proof. ($\Rightarrow$) Denote by $\delta(G)$ the degree of G . Suppose each row in G has a pole which is unique in its column; we'll call them $\alpha_1 \in \mathbb{C}, \alpha_2 \in \mathbb{C}, \cdots, \alpha_p \in \mathbb{C}$. We will construct a diagonal transfer matrix $D(s)$, by setting $d_{ii} = s - \frac{p_i(s)}{q_i(s)}$, where $p_i(s)$ is the minimal real-valued polynomial (of degree $\delta_i \in \{1,2\}$) for α_i and $q_i(s)$ is a polynomial of degree $\delta_i - 1$ with normalized leading coefficient, that shares no roots with other elements in G. (If α_i is real, this makes $d_{ii} = \alpha_i$.) D is a diagonal matrix of proper rational functions, as desired, and $(sI-D)_{ii} = \frac{p_i(s)}{q_i(s)}$.

Multiplying P on the left by $(sI-D)$ removes from the ith row δ_i poles unique to that row of P, replacing them with $\delta_i - 1$ poles which do not cancel and are unique to the ith row of $(sI-D)P$. As a result, $\delta((sI-D)P) = \delta(P) - p$. By Lemma 4, the degree of the transfer function matrix $(sI-D)P$ is equal to the degree of $[D\ (sI-D)P]$. It follows by Lemma 3 that there exists a realization (A,B) of G where A_{22} is a $(n-p)\times(n-p)$ matrix, which means that A is $n \times n$. Thus (A,B) is a minimal realization of G where $Q = 0$.

($\Leftarrow$) Suppose G is single-input (a column vector of transfer functions) and that some element in G (without loss of the pth element) does not contain a unique pole. If there exists a minimal realization $A, B, C = [I_p\ 0]$ so that $Q = 0$, then A_{11} is diagonal, as is $A_{12}A_{22}^{n}A_{21}$

for all values of n. It follows by induction that for any power A^m of A, the submatrix A^m_{11} is also diagonal. We now consider the transfer function $\bar{G}$ consisting of the first $p-1$ elements of G. $\bar{G}$ has the same degree as G, so the system $A, B, \bar{C}$ is a minimal realization of $\bar{G}$. However, the pth column of $\bar{C}A^k$ is uniformly zero (since the first $p-1$ elements of the pth column of A^k are zero for all k, and $\bar{C} = [I_{p-1}\ 0]$) so the pth column of the observability matrix is also uniformly zero. Since $(A, B, \bar{C})$ is not observable, it is not a minimal realization of $\bar{G}$, a contradiction. □

Theorem 2 characterizes a class of transfer functions which always have a decoupled minimal realization. Let us illustrate this result with a few examples.

Example 1. *Consider the dynamical structure*

$$Q = 0; P = G_1 = \begin{bmatrix} \frac{1}{s+1} \\ \frac{2}{s^2+2} \end{bmatrix}$$

Following Lemma 3, choose D to give us a desired $(sI-D)$*:*

$$D = \begin{bmatrix} -1 & 0 \\ 0 & -\frac{2}{s} \end{bmatrix}; (sI-D) = \begin{bmatrix} s+1 & 0 \\ 0 & \frac{s^2+2}{s} \end{bmatrix}$$

$$\begin{bmatrix} W & V \end{bmatrix} = \begin{bmatrix} D & (sI-D)P \end{bmatrix} = \begin{bmatrix} -1 & 0 & 1 \\ 0 & -\frac{2}{s} & \frac{2}{s} \end{bmatrix}$$

This transfer function has a minimal realization

$$(\bar{A}, \bar{B}, \bar{C}, \bar{D}) = \left([0], \begin{bmatrix} 0 & -1 & 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 2 \end{bmatrix}, \begin{bmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \right)$$

which gives us our minimal realization of (Q, P)*:*

$$(A, B) = \left(\begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 2 \\ 0 & -1 & 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \right)$$

This is also a minimal realization of G, which can be seen from Theorem **??** *since* $\delta(G) = 3$. *This is consistent with Theorem 2 as all elements of G contain a unique pole in their column.*

Example 2. *The transfer function* $G_2 = \begin{bmatrix} \frac{1}{(s+1)^2(s+2)} \\ \frac{1}{(s+1)(s+2)} \end{bmatrix}$ *does not admit a completely decoupled minimal realization since neither row contains a unique pole in its column, thus violating Theroem 2. We can see this as follows:*

G_2 has a minimal realization of form $(A,B) =$

$$\left(\begin{bmatrix} -1 & 1 & 0 \\ 0 & -2 & 1 \\ 0 & 0 & -1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right)$$

All possible minimal realizations of G_2 with desired form C can be found by performing a change of coordinates on the hidden state x_h, $x_h^* = T_1 y + T_2 x_h$, with T_2 invertible: In this case we set $T_1 = [t_0\ t_1]$ and $T_2 = [t_2]$, giving us

$$(\bar{A}, \bar{B}) =$$
$$\left(\begin{bmatrix} -1 & 1 & 0 \\ -\frac{t_0}{t_2} & -2 - \frac{t_1}{t_2} & \frac{1}{t_2} \\ -t_0 - \frac{(t_1 - t_2)t_0}{t_2} & t_0 - 2t_1 - \frac{(t_1 - t_2)t_1}{t_2} & \frac{t_1 - t_2}{t_2} \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ t_2 \end{bmatrix} \right)$$

From our equation for W, w_{12} never equals zero because a_{12} never equals zero. The corresponding term of Q is also always nonzero. This means that for any minimal realization of G_2, x_2 has a direct effect on x_1.

The fact that any transfer function with a unique pole on every row admits a completely decoupled realization implies that the existence of a completely decoupled minimal realization for a transfer function is not all that special, since any transfer function is arbitrarily close to one that does have a decoupled minimal realization. We show this in the following theorem.

Theorem 3. *Given the H_∞ norm on the set $\mathscr{G}$ of strictly proper, rational, stable $p \times m$ transfer functions, and any $\varepsilon > 0$, for each $G \in \mathscr{G}$ there exists a $\hat{G} \in \mathscr{G}$ so that each row of $\hat{G}$ has a pole that is unique in its column and $||\hat{G} - G|| < \varepsilon$.*

Proof. For each i, select α_i to be a pole in the ith row of G with minimal real-valued polynomial $p_i(s)$. Choose γ to be the norm of the diagonal matrix with terms $(\frac{1}{p_1(s)}, \cdots, \frac{1}{p_p(s)})$. Then let $\delta = \frac{\varepsilon}{2\gamma\|G\|}$ and find $\delta_i \in (0, \delta)$ so that no zeroes of the polynomial $p_i(s) - \delta_i$ are zeros or roots of other elements in G, and $\frac{\delta_i}{p_i(s)-\delta_i}$ is stable. Define the diagonal matrix J where $j_{ii} = \frac{p_i(s)}{p_i(s)-\delta_i}$. Let $\hat{G} = JG$. The norm of the diagonal matrix with terms $(\frac{1}{p_1(s)-\delta_1}, \cdots, \frac{1}{p_p(s)-\delta_p})$ is less than 2γ, each row of $\hat{G}$ has as unique poles the zeroes of $p_i(s) - \delta_i$ and

$$\begin{aligned} \|\hat{G} - G\| &= \|(J-I)G\| \leq \|J-I\|\|G\| \\ &< 2\delta\gamma\|G\| = \varepsilon \end{aligned}$$

□

Chapter 3

Current Network Reconstruction Methods

The dynamics governing a biological network are much more difficult to identify from data than are the dynamics of simple systems such as a circuit board, or a weight on a spring. There are several reasons for this. First, biological systems involve a large number of different chemicals/substances, usually much larger than the number of chemicals whose concentrations and behavior can be observed. Thus, any method to identify these systems is very computationally intense and requires large amounts of data describing the system [4]. Second, biochemical systems have fast reaction times and it is hard to obtain time-series data with sufficiently frequent measurements to accurately portray system dynamics. Finally, data obtained from biochemical networks is highly prone to noise [5]. This noise stems from both the limited accuracy of measurements and the uniqueness of individual cells or organisms with the same genetic structure and function. (For example, not all liver cells have exactly the same number of ribosomes.) Challenges such as these make the accurate reconstruction of genetic reaction networks very difficult, and finding ways to more accurately describe the dynamics and structure of these networks is an active research topic.

The problem of accurately reconstructing gene regulatory networks attracts researchers from many different academic and research fields, such as mathematics, computer science, electrical engineering, and statistics. Each of these disciplines favors distinct approaches for network reconstruction, and as a result, several different methods are commonly used for identifying genetic regulatory networks. This chapter does not purport to be an exhaustive survey of the different methods used, but will briefly overview several approaches from a few of the most common disciplines–Boolean networks, Bayesian methods, and differential equations models.

3.1 Boolean methods

Boolean methods of network identification involve algorithms to determine qualitative rules describing dependencies among the different states in a network. At every time step for which data is available, each state is labeled to be either on and assigned a value of 1, or off and assigned a value of 0 [6]. The effects on a given state by all other states are modeled by finding the Boolean rule that best describes the input-output data. For example, if state x_1 is on at a given time state whenever states x_1, x_2, and x_3 are off in the previous time state, an algorithm should determine that the Boolean relation for x_1 is $x_1(t+1) = NOT\{x_1(t), x_2(t), x_3(t)\}$ (see Figure 3.1). To be able to fully determine each Boolean relation (there are 2^k possible Boolean relations describing the effect on a state of k distinct states), the algorithm needs to consider data sets with as many variations of states ‘on’ or ‘off’ as possible.

Probabilistic Boolean networks [7] are an extension of Boolean networks in which a node can have more than one different Boolean rule assigned, with each of the rules carrying a probability of being the expressed rule at any time step. These networks can be

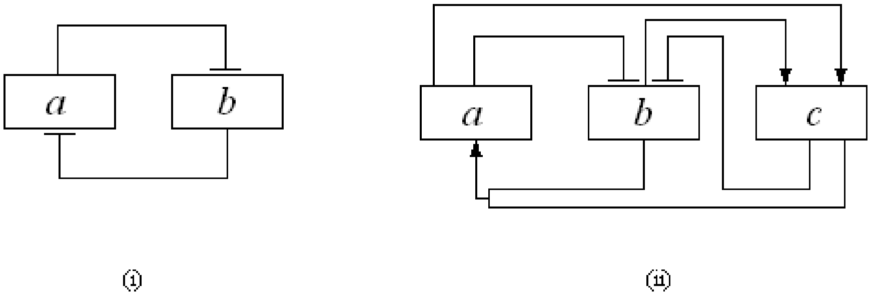

Figure 3.1 Two Boolean networks: the Boolean relation for (i) is $x_a(t+1) = NOT x_b(t), x_b(t+1) = NOT x_a(t)$. Each species degrades the other. The Boolean relation for (ii) is $x_a(t+1) = x_b(t) OR x_c(t), x_b(t+1) = x_a(t) NOR x_c(t), x_c(t+1) = x_a(t) AND x_b(t)$. In other words, x_a is produced when one of the other two are present, x_b is produced when none of the other two are present, and x_c is produced when both x_a and x_b are present.

represented as Markov chains.

One strong point of Boolean networks in discovering structural relationships between genes is their ability to predict direction of causality. Some methods are only able to determine whether a relationship exists between two states, but the Boolean rules that identify a network clearly express which variable influences and which is influenced [7]. Researchers have used Boolean methods to successfully determine qualitative relationships among genes in the development of *Drosophila*, among other organisms. Boolean networks are also computationally easy to determine from data. Boolean methods have the weakness that they cannot determine relative strengths of Boolean rules on different states, and do not easily allow for variance of the rules with reaction parameters. Generalized formalisms of Boolean networks do exist; for example, replacing the two discretized states ON and OFF with $n > 2$ states ranging from 0 (‘off’) to $n-1$ (‘fully on’). This allows for intermediate levels of expression and thus gives a better prediction of dynamics, although its utility is tempered by a sharp rise in computational and mathematical complexity [6].

3.2 Bayesian networks

Some algorithms rely on heavy use of probability laws and sometimes information theory [8] to reconstruct a gene regulatory network (GRN), by finding a data set's best probable fit within a family of possible models. These methods make heavy use of Bayes' law, which relates the conditional likelihood of an event X (given knowledge of an event Y) to the likelihoods of X, Y, and Y given X:

$$P(X|Y) = \frac{P(X)P(Y|X)}{P(Y)}$$

In other words, knowing whether statement Y is true may affect our prediction of whether statement X is true, depending on the conditional probability $P(Y|X)$.

Since static Bayesian networks are required to be acyclic, they are unable to successfully reverse engineer gene regulatory networks, as almost all interesting GRNs contain some feedback loops. Dynamic Bayesian networks assign a node to each species at each discrete time-step. In this context a feedback loop is not cyclic because all directed edges from nodes at one time step lead to nodes at later time steps. According to the author of [9], this does not unbearably increase the system complexity since the relationships among nodes are then assumed to be time-variant. Dynamic Bayesian methods can also reveal a little more information about network dynamics (as opposed to just network structure). Examples of regular and Dynamic Bayesian networks are illustrated in Figure 3.2.

Friedman et al. [10] describe a commonly-used procedure involving Bayesian methods. Their method compares different species in the network and calculates the probabilities, by use of Bayes' rule, that any two states are coexpressed. The procedure then searches through a family of different candidate networks, looking for the ones that best fit the probabilities and are thus most likely to have generated the observations. One challenge the authors mentioned in this paper was distinguishing regulation of one gene by another as opposed to non-causal coexpression of the two genes.

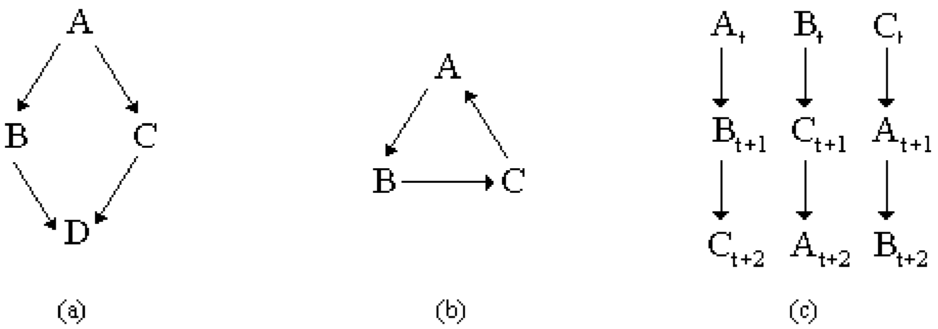

Figure 3.2 The network in (a) is an example of a Bayesian network: a directed acyclic graph. (b) is not a Bayesian network because it contains a cycle. Dynamic Bayesian Networks (c) account for feedback loops by assigning each species a node at each time step.

Besides this particular approach, other algorithms exist that rely on probability distributions. Some methods are modifications to the typical Dynamic Bayesian approach, such as [11], which builds a network one gene at a time instead of searching over complete networks. This method rates species that might affect a certain state one at a time instead of using a cumulative distribution function. Others, such as the ARCANe algorithm [12], incorporate the concept of mutual information. ARCANe identifies groups of genes with high mutual information and labels those groups as being possibly correlated. The algorithm then applies the Data Processing Inequality to identify which correlations are direct relationships and which are not. ARCANe is interesting in that it does not necessarily require data from perturbation experiments, as long as the data contains considerable ”phenotypic variations of a given cell type.” [12] Other related methods, such as Hidden Markov Models or Markov chain Monte Carlo methods (MCMC), are considered by the community to be special cases of Bayesian models [9].

One advantage Bayesian networks and mutual information methods share is the ability to incorporate almost any information available about a network. Conditional probabilities can be obtained not only from expression data, but also sources such as previous knowledge of the properties of different reactants and known relationships among species in similar networks [9]. Probability rules can have varying confidence levels, according to the type and accuracy of information on which they are based. The authors of [13] scored the accuracy of probability rules inferred by different types of experiments, such as gene expression versus protein-interaction. They were then able to weight the confidence level of these probability rules in their identification of some yeast pathways.

Methods using Bayesian networks can be very computationally intense (and are NP-hard [14]), especially given the need to calculate large numbers of probabilities and apply them to a very large search space of possible models. For example, knowing that a network is a directed acyclic graph and contains 10 nodes gives a search space on the order of 10^{18} possible models [5]. Calculating fits of different probabilities can be quite unwieldy for such a large search space, and different methods exist to reduce this search space.

3.3 Differential equation methods

Unless some of the species in a network are present in very small quantities (such as certain enzymes or nucleic acids in a cell), their respective concentrations over time can be represented fairly accurately as continuous functions. The rate of change in concentrations can then be related to the current concentration levels through a series of ordinary differential equations.

Besides relatively high concentrations, the use of differential equations to model a gene regulatory network makes several other assumptions: namely, that the reaction rates are slow compared to the rates of diffusion (‘mixing’) of a chemical. These mixing rates are usually slower for chemicals in a cell than for chemicals in a test tube, due to biochemical properties such as low diffusion rates across membranes [15].

Various approaches exist for finding a good-fit differential equation. One method [16] infers, for each concentration x_i, its likely activators and inhibitors, and takes that data into account in deciding the typical chemical kinetics of x_i based on the other states that affect it–such as self-degradation/autocatalytic production, dimerization, Michaelis-Menton kinetics, and other simple ’motifs’ (common structures in chemical reaction networks). The inferred kinetic information is used to formulate an equation approximating $\frac{dx_i}{dt}$. An example of such an inferred equation might be

$$\frac{dx_1}{dt} = V_i\left(1+\frac{x_2}{x_2^2+\theta_{1,2}}\right)\left(1+\frac{x_3}{x_3^2+\theta_{1,3}}\right)\left(1+\frac{\theta_{1,4}}{x_4+\theta_{1,4}}\right)-\lambda_1 x_1$$

where V_i is the transcription rate of x_i, $\theta_{1,2}$, $\theta_{1,3}$, and $\theta_{1,4}$ are constants relating to activation or inhibition of transcription by x_2, x_3, and x_4, and λ_1 is the degradation rate of x_1. If the equation for $\frac{dx_i}{dt}$ is known to have this form, solving for the equation reduces to finding the values for the five constants which best fit the data. However, finding the correct equation form requires knowledge of how x_2, x_3, and x_4 affect x_1. With many different chemicals involved in a network, the conjectured equation form can unintentionally disregard important reactions.

Other researchers [17] approach the reconstruction problem by assuming the network concentrations over time are given by a system of nonlinear differential equations

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x},\mathbf{p})$$

without specifying the structure of $\mathbf{f}$. $\mathbf{x}$ is the vector of the concentrations (or activity levels) of the different states, and $\mathbf{p}$ is a vector of other parameters relevant to the reaction, such as temperature, pH, rate, or other kinetic constants. Involving the terms in $\mathbf{p}$ as independent variables of the equation model makes the model robust to changes in system dynamics or structure which can result from changes to the kinetic parameters (some parameters, such as rate constants, don’t change.) The goal of setting the equation up in this manner is not to completely determine the equations of $\mathbf{f}$, but rather to identify the terms of the Jacobian of $\mathbf{f}$, $\mathbf{F} = \{\partial f_i / \partial x_j\}$. A nonzero term F_{ij} represents an effect on x_i caused by x_j. The size of the value F_{ij} represents the scale of that effect, negative terms represent degradation or inhibition, and positive terms represent production or activation.

In [17], the terms of the Jacobian are found by making perturbations to certain of the parameters p_j–for example, increasing temperature or pH of the solution. Data obtained from time-series data of the system with and without each perturbation is used in determining the best fit for the terms in F, and is compared again at a number of time steps to make sure it is robust against noise. (Steady-state data can also be used, but requires performing more experiments and perturbations.)

Other methods for system identification also use linearizations of the network. Working with a linear system greatly reduces computational complexity. [18] describes a method for reconstructing a nine-gene subsequence of the SOS pathway in *E. coli*. The authors approximate this network with a linear system

$$\frac{d\mathbf{x}}{dt} = A\mathbf{x} + \mathbf{u}$$

where A is an $n \times n$ coefficient matrix (similar to $\mathbf{F}$ previously) and $\mathbf{u}$ is an input vector representing perturbations made to the system. Using the assumption that all n states in the system are observed and that each state x_i in the system has fewer than $k < n$ connections, the authors searched for the best fit of A to the data sets $(\mathbf{x}, \mathbf{u})$. The authors optimized the network over choices of k to find a system that minimized false positives and was both consistent with the data and dynamically stable.

Recently, the authors of [14] developed a newer approach to network reconstruction from time-series data using a linear matrix inequality (LMI) algorithm and regression analysis to best fit the model to a piecewise affine (PWA) system. Piecewise affine systems offer much of the simplicity of linear systems, but are also able to incorporate more complicated dynamics such as multiple equilibrium points. This algorithm, as do many of the others

I have discussed in this chapter, exploits the suspected scale-free topology of biological networks–that is, that the majority of states in a system have relatively few connections to other states, while some states have very high numbers of connections (the number of connections for each state roughly follows a power-law distribution.) The authors tested their method on a number of 10-state sample networks, and favorably compared their results to those obtained through a method incorporating Dynamic Bayesian Networks.

One way of reducing computational complexity in fitting models using differential equations methods is by model order reduction. The authors of [14] mention that large sets of genes will often have similar expression patterns, meaning their states can be clustered into one state, This reduces the model order by eliminating variables redundant to the system dynamics. Bamieh and Giarré in [19] show a successful example of model order reduction, a three-state approximation of a ten-state model of protein levels in *Drosophila*. The approximation exhibits kinetic behavior very similar to the originial network.

A number of other approaches to model biological networks with systems of differential equations exist, each using different methods to optimize the coefficients or structure of the network. One of these is a procedure by [2] where the 1-norm of the reaction rates is minimized.

As with Bayesian methods, algorithms incorporating differential equations have their strengths and limitations. If a certain chemical in a network is present in very small quantities, that chemical's concentration is not well represented by a continuous function. Alternative methods, such as stochastic equations, are necessary to more accurately model network behavior. Networks where chemicals have slow diffusion or mixing rates with respect to reaction rates require spatial as well as temporal dynamics (partial differential equations) to yield an accurate model. Furthermore, useful data for such approaches is usually limited to expression (time-series or steady-state) data and knowledge of the mechanics of involved chemical reactions. Additional information about a system, if it cannot be well-quantified or modeled in the differential equations (such as the shape or molecular weight of a chemical), cannot really be used in a differential equation model. A Bayesian analysis can sometimes use such information to construct a probability assignment.

Advantages found in using systems of differential equations for describing biological networks include knowledge of system dynamics, not just structure. Questions about direction of causality common in Bayesian methods are resolved in differential equations models.

3.4 Why so many different approaches?

There are several reasons that no one superior method is universally accepted for determining the structure of a gene regulatory network from data. First, data sets for a regulatory network can vary greatly in both the quantity, accuracy, and type of data they contain. Sometimes, the data collected comes from a series of steady-state perturbations made to the network–concentration of one chemical is increased or decreased by a measured amount, and the system reacts to this change from the steady-state until the concentration stabilizes. The data only gives the initial and final concentrations, with no information about the dynamics of the reaction. Other measurements yield time-series data on a system, which provide information about the concentrations of the states in a network at a series of time intervals. This data can give a better idea of the dynamics of a network, but is harder to obtain, especially when short time intervals are required [2].

Also, due to network structure and scale, some methods of network identification are inapplicable in certain situations. For example, as discussed before, differential equations do not accurately model the dynamics of chemicals with very small concentrations. Discrete stochastic simulation can more effectively model such networks [20] [21].

Finally, as mentioned previously, scientists with different research and educational backgrounds have their own biases and previous knowledge about certain methods. Systems biology is an emerging field, and research in this area draws professionals in the areas of statistics, mathematics, chemical and electrical engineering, and computer science. Each field brings a different set of computational tools and methods. For example, most systems biologists who favor Bayesian methods [10] are computer scientists by training, while proponents of differential equations or regulation matrix methods are usually mathematicians or engineers [14]. A researcher highly expert in network reconstruction by one method might be negatively disposed toward using a new method with which he was relatively unfamiliar.

Among methods that only purport to find network structure–not dynamics–the commonly accepted metric of accuracy is the number of false negatives and false positives. Metrics involving network dynamics, however, differ according to author and discipline and are harder to compare. Furthermore, researchers comparing different methods are usually less familiar with other methods than their own, and do not always implement the other methods correctly. An example of this is seen in a paper comparing performance of the ARCANe algorithm to a Bayesian network method [12]. The authors compared the performance of their algorithm and the Bayesian algorithm on a synthetic network, and found their own to be more accurate. It was later pointed out in [22] that the authors used a static instead of dynamic Bayesian network; that author showed that the dynamic Bayesian method was in fact more accurate than the ARCANe method. As a result, a large-scale comparison of the predictive accuracy of different methods in identifying networks is lacking, although an paper will often describe an analysis of some sample network on which the described method favorably compares to one or two other methods [16] [12] [14] [18] [10].

Chapter 4

Using Dynamical Structure Functions to Reconstruct Genetic Networks

As stated in chapter 2, knowledge of a network's dynamical structure is knowledge of not only network topology (presence or absence of connections) but also system dynamics (whether a connection produces an activating or inhibitory effect, and the strength of that effect.) Since a given transfer function admits any internal structure, input-output data, which gives us a transfer function, is not enough information to deduce a system's dynamical structure. From [1] we know what information, in addition to input-output data, is necessary to obtain a system's dynamical structure:

Lemma 5. *Consider a system of form (2.2) and assume that the only available information on the system is its associated transfer function G, which does not have any rows or columns that are entirely zero. The dynamical structure can be reconstructed if and only if at least $p^2 - p$ transfer functions of Q or P are known.*

To accurately predict network structure, data sets that observe changes made in the structure (altering the network in some way) are preferable to data exploring diverse dynamical responses of an unaltered network, even if the data is time-series instead of steady-state [16]. We introduce a method that uses well-defined network alterations, analogous to commonly performed biological experiments, which affect network structure in a way that we can quantify and exploit to ascertain the network's dynamical structure.

4.1 Silencing and overexpression

Gene silencing and gene overexpression are two types of experiments readily available to biologists that make modifications to the structure of a biological system. Overexpression of a gene may be constitutive or inducible, through introduction of a transgene into the host which is specifically designed to increase the abundance of the desired transcript. RNA silencing involves down-regulation of one or more genes, via mutation or inhibition. The target specificity of these methods allows the control of gene expression, without directly affecting other genes in the network. Putting these modifications in terms of a general network, overexpression of a state makes the effects of other states on the overexpressed state negligible, while preserving the effects of the overexpressed state on others. Silencing removes a state from the reaction entirely. These experiments can affect the system slightly or severely depending on the state to be modified (Figure 4.1).

We model both modifications with an operator that adds an additional state variable and input to the system. For state silencing, the additional state represents a silencing RNA targeted to temporarily drive a particular measured state to zero. Inducible overexpression uses a chemical to activate the inserted transgene, temporarily driving the concentration of a particular measured state to a large value. For an LTI system with partial state observation, our modification can be represented as follows:

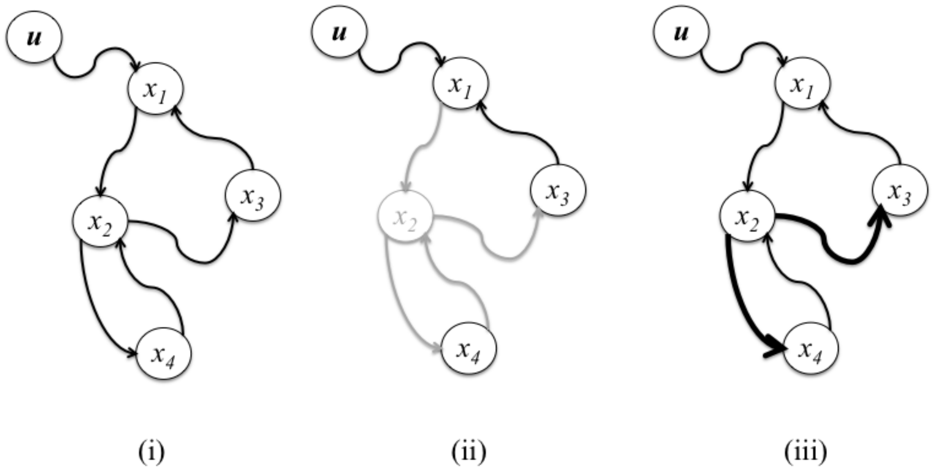

Figure 4.1 A sample network (i) with four observed states and one input, and the resulting network when x_2 is silenced (ii) or overexpressed (iii)

$$\begin{aligned}\begin{bmatrix}\dot{y}\\ \dot{x}_h\\ \dot{z}\end{bmatrix} &= \begin{bmatrix}A_{11} & A_{12} & \beta_i\delta_i\\ A_{21} & A_{22} & 0\\ 0 & 0 & -\alpha_i\end{bmatrix}\begin{bmatrix}y\\ x_h\\ z\end{bmatrix} + \begin{bmatrix}B_1 & 0\\ B_2 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}u\\ u_h\end{bmatrix}\\ y &= \begin{bmatrix}I & 0 & 0\end{bmatrix}\begin{bmatrix}y\\ x_h\\ z\end{bmatrix}\end{aligned} \tag{4.1}$$

In both silencing and overexpression, α is the coefficient for the self degradation of our added state so always takes a positive value. β is positive for overexpression and negative for gene silencing. The magnitude of β is set to be much greater than α, as we want to either quickly drive the concentration of state i to zero or rapidly induce a strong increase in the concentration.

The dynamical structure of the modified system 4.1 is equal to the dynamical structure for the original system (A,B), since this system represents the change in expression of x_i. When the system reaches its new steady state, the dynamics of the system change. [23] suggests that the modified system can be rewritten as

$$\begin{aligned}\begin{bmatrix}\dot{y}\\ \dot{x}_h\\ \dot{z}\end{bmatrix} &= \begin{bmatrix}\bar{A}_{11} & \bar{A}_{12} & \beta_i\delta_i\\ \bar{A}_{21} & A_{22} & 0\\ 0 & 0 & -\alpha_i\end{bmatrix}\begin{bmatrix}y\\ x_h\\ z\end{bmatrix}+\begin{bmatrix}B_1 & 0\\ B_2 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}u\\ u_h\end{bmatrix}\\ y &= \begin{bmatrix}I & 0 & 0\end{bmatrix}\begin{bmatrix}y\\ x_h\\ z\end{bmatrix}\end{aligned} \tag{4.2}$$

where $\bar{A}$ is A with the terms in the ith column (silencing only) and the ith row (both overexpression and silencing) set to zero. It follows directly from Equations 2.4, 2.5, and 2.6 of Section 2.1 that Q for the modified system is equal to Q for the original system, except for the ith row (for overexpression) or the ith row and the ith column (for silencing). This observation seems reasonable and consistent with our descriptions of overexpression and gene silencing, and is fantastic news because each different overexpression or silencing experiment we perform reveals more information about the terms of Q and P.

Lemma 6. *For any linear system (A,B), the internal structure of the system representing silencing of the ith state of (A,B) is equal to the internal structure of the original system, except for the terms on the ith row and column which are zero. The internal structure of the system representing overexpression of the ith state of (A,B) is equal to the internal structure of the original system, except for the terms on the ith row which are zero.*

Proof. This follows from the modifications made above, as well as Equations 2.5 and 2.6.

□

Theorem 4. *The structure for any LTI system can be completely determined by performing p overexpression or silencing experiments, one for each observed state in the network.*

Proof. Suppose our original system is (A,B) with transfer function G. Call our modified systems $(A_1,B_1),(A_2,B_2),\cdots,(A_p,B_p)$ with transfer functions $G_1,G_2,\cdots,G_p$ and dynamical structures $(Q_1,P_1),\cdots,(Q_p,P_p)$. The matrix equations $(I-Q_i)G_i = P_i$ give us $(p-1)m$ equations relating the $p+m-1$ unknowns $q_{i1},\cdots,q_{ip},p_{i1},\cdots,p_{im}$. (Recall that

$q_{ii} = 0$). Also considering the equation $(I-Q)G = P$ given by the original system, we have $(p-1)m+1$ equations, which is more than $p+m-1$ for any values of $p > 1, m \geq 1$.

□

4.2 Discussion of algorithm

An algorithm for determining the dynamical structure of single input systems is currently being implemented in MATLAB. The algorithm takes expression data from an original system and modified systems (each of the p states is silenced individually), and estimates the transfer function of each system using methods in the System Identification toolbox. The algorithm then solves for the correct dynamical structure (Q,P). Our algorithm has been tested on some small sample networks with promising results, and is currently being generalized to handle overexpression profiles and arbitrary system order.

Chapter 5

Conclusion

We have shown in this work a construction to derive, for any dynamical structure pair (Q,P), a linear system (A,B) whose dynamical structure is (Q,P). We have also found conditions on a transfer function G for the existence of a completely decoupled minimal realization, and shown as a consequence that any strictly proper transfer function is arbitrarily close to one for which a decoupled minimal realization exists.

This paper also examined several common approaches to identifying gene regulatory networks (structure and/or dynamics) from gene expression data, and explored a new reconstruction method utilizing the theoretical properties of dynamical structure.

5.1 Future work

In Section 2.2 we mentioned that finding a realization minimal for a particular dynamical structure means finding the correct D. Deriving this D for an arbitrary dynamical structure is an interesting problem, because it will yield us conditions for the existence of a minimal realization of G with any compatible dynamical structure.

We will also generalize Theorem 4 to define exactly which combinations of overexpression and silencing experiments are sufficient to obtain dynamical structure. For example, if it is not possible to perturb a particular state x_i, we can still obtain structure by performing more than p equations on the observed states (some states will be silenced in one experiment and overexpressed in another.)

Section 3.4 discussed the lack of unbiased, direct comparison of reconstruction techniques. A study is currently underway comparing a reconstruction method based on dynamical structure to some of the methods discussed above, including the sparsity method set forth in [2], the Linear Matrix Inequality method in [14], and a Monte Carlo Markov chain approach. Each algorithm will reconstruct a series of synthetic networks from 'expression' data, and the predictive accuracy of each method will be scored and compared.

Bibliography

[1] J. Gonçalves, R. Howes, S. Warnick, ”Dynamical Structure Functions for the Reverse Engineering of LTI Networks,” Proceedings of the 2007 Conference on Decision and Control, 1516-1522.

[2] A. Papachristodoulou, B. Recht, ”Determining Interconnections in Chemical Reaction Networks,” Proceedings of the 2007 American Control Conference, 4872-4877.

[3] C. T. Chen, *Linear System Theory and Design*, Revised. Saunders College Publishing, Orlando: 1984, p. 241.

[4] Z. Szallasi, J. Stelling, V. Periwal, *System Modeling in Cellular Biology: From Concepts to Nuts and Bolts*. MIT Press, Cambridge, MA: 2006.

[5] K.-H. Cho, S.-M. Choo, et al., ”Reverse engineering of gene regulatory networks,” Systems Biology, IET, May 2007, 149-163.

[6] H. de Jong, D. Ropers, ”Qualitative Approaches to the Analysis of Genetic Regulatory Networks,” chapter in *System Modeling in Cellular Biology*, p. 125.

[7] I. Shmulevich, E. Dougherty, S. Kim, W. Zhang, ”Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks,” *Bioinformatics*, Vol. 18 no. 2, pp. 261-274, 2002.

[8] P. Spirtes, C. Glymour, R. Scheines, ”Constructing Bayesian network models of gene expression networks from microarray data,” Proceedings of the Atlantic Symposium on Computational Biology, Genome Information Systems and Technology, 2000.

[9] V. Periwal, ”Bayesian Inference of Biological Systems: The Logic of Biology,” chapter in *System Modeling in Cellular Biology*, p. 125.

[10] N. Friedman. ”Inferring Cellular Networks Using Probabilistic Graphical Models,” *Science*, 6 Feb 2004, pp. 799-805.

[11] N. Barker, C. Myers, H. Kuwahara, ”Improved Algorithm for Learning of Genetic Regulatory Network Connectivity from Time Series Data,” *IEEE Transactions on Computational Biology and Bioinformatics*, No. 8, Mar 2007.

[12] K. Basso, et al, ”Reverse engineering of regulatory networks in human B cells,” *Nature Genetics* 37, 382-390, 2005.

[13] I. Lee, S. V. Date, A. T. Adai, E. M. Marcotte, ”A probabilistic functional network of yeast genes,” *Science*, 306:1555-1558, 2004.

[14] C. Cosentino, W. Curatola, F. Montefusco, M. Bansal, D. di Bernardo, F. Amato, ”Linear matrix inequalitites approach to reconstruction of biological networks,” *IET Systems Biology* 2007:1:3, 164-173.

[15] E. Conrad, J. Tyson, ”Modeling Molecular Interaction Networks with Nonlinear Ordinary Differential Equations,” chapter in *System Modeling in Cellular Biology*, p. 97.

[16] N. Soranzo, G. Bianconi, C. Altafini, ”Comparing association network algorithms for reverse engineering of large-scale gene regulatory networks: synthetic versus real data,” *Bioinformatics* 2007 23(13):1640-1647.

[17] E. Sontag, A. Kiyatkin, B. Kholodenko, ”Inferring dynamic architecture of cellular networks using time series of gene expression, protein and metabolite data,” *IEEE Bioinformatics*, 20(12):1877-1886, 2004.

[18] D. Di Bernardo, T. Gardner, J. Collins, ”Robust Identification of large genetic networks,” *Pac. Symp. Biocomput.* 2004, 9, 486-497.

[19] B. Bamieh, L. Giarr, ”On Discovering Low Order Models in Biochemical Reaction Kinetics,” Proceedings of the 2007 American Control Conference, 2702.

[20] J. Paulsson, J. Elf, ”Stochastic Modeling of Intracellular Kinetics,” chapter in *System Modeling in Cellular Biology*, p. 149.

[21] L. Petzold, ”Discrete Stochastic Simulation for Biochemical Systems–State of the Art,” talk given at the Eighth International Conference on Systems Biology, October 4, 2007.

[22] A. Hartemink, ”Reverse engineering gene regulatory networks,” *Nature Biotechnology* 23,554-555, 2005.

[23] R. Howes, S. Warnick, J. Gonçalves, N. Dalchau, ”Reconstruction of Biological Networks Through Gene Silencing and Overepxression,” Proceedings of the 2007 International Conference on Systems Biology.