VALIDATION OF DYNAMICAL STRUCTURE FUNCTIONS FOR THE RECONSTRUCTION OF BIOCHEMICAL NETWORKS

by

Taylor Southwick

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

Department of Computer Science

Brigham Young University

August 2011

Adviser: Sean Warnick

Signature: _____

Honors Representative:

Signature: _____

ABSTRACT

VALIDATION OF DYNAMICAL STRUCTURE FUNCTIONS FOR THE RECONSTRUCTION OF BIOCHEMICAL NETWORKS

Taylor Southwick
Department of Computer Science
Bachelor of Science

The past decade has seen huge leaps in the ability of biologists to gather biochemical data, but slow progress in the ability to understand what that data means. Network reconstruction algorithms aim to decipher the underlying dynamics and structure of a system. Dynamical structure functions (DSF) have been shown to work well in preliminary studies and is explored in this thesis as a viable alternative to other major methods available. Comparisons are made to two major methods, BANJO and VBSSM, using *in silico* data derived both from synthetic networks and from a mathematical model of an *in vitro* system developed in yeast (*Saccaromyces cerevisiae*) known as IRMA. Results indicate that DSF tends to outperform the other two overall, while VBSSM is slightly better at finding connections and BANJO tends to find slightly more non-connections. For general purpose *de novo* network inference, DSF should be used to ensure the highest confidence in resulting network topology.

ACKNOWLEDGMENTS

I would like to thankfully recognize the assistance of my advisor, Sean Warnick of the Brigham Young Department of Computer Sciences. His direction was essential through the whole process from research design to analysis. Dr. Guy-Bart Stan, a collaborator at Imperial College London, was key in developing some of the initial thoughts of comparison methods while providing assistance to my summer research experience in London.

TABLE OF CONTENTS

Chapter 1

Introduction

During the past decade, the production of biological data has increased to such a high point that researchers struggle to effectively use all of it. There are currently many high-throughput technologies that can yield large volumes of data. DNA sequencing, protein mass spectrometry, and other techniques to measure biological properties have seen an increase in use as they become more efficient and affordable. As technology has improved, the ability to measure dynamic processes, instead of static snapshots, has become easier and more common place. A key to unraveling the dynamic nature of biological systems is to understand the underlying structure.

Any biological application with this new data requires an understanding of the way that the components in a system interact. These applications range from curing metabolic disorders to increasing the yield of a strain of corn in new climates. Researchers use this information to build models and to plan the next phase of their research. This form of experimentation has become more common among researchers, but requires the understanding of the complex structures that coordinate the reactions in the organism.

The structure of a biochemical system is the network of interconnected chemical species. Common chemical species include DNA, proteins, and lipids. The networks

they form within a system can serve different functions, from producing new proteins via DNA transcription and translation in gene regulation networks, to sending a signal through a cell via signal transduction networks. These species form complexes (units composed of two or more individual chemical species) which then can catalyze certain chemical reactions which would not have otherwise occurred.

Complexes like this, as well as many more, run the entire biological process in every living organism. A common complex is that formed between DNA polymerase (a protein) to a strand of DNA which then initiates transcription. Another complex that is often found in signal transduction networks is a protein and a kinase or phosphatase, which will, respectively, add or remove a phosphate group an inorganic chemical marker. The connections they make are often unique and highly specific. They are key to understanding systems within an organism. The algorithms that attempt to recover, or infer, these connections are known as network reverse-engineering or reconstruction algorithms.

The problem of identifying the underlying structure of a network has interested many people in the field of biology. The number of algorithms that claim to reconstruct network structure increases every month as new ideas are tried. As this number increases, it is important to validate that an algorithm performs better than its predecessors; otherwise, there would be no novel use for it. This thesis provides an *in silico* comparison of two major network reconstruction algorithms, BANJO and VBSSM, to a novel technique called Dynamical Structure Functions, DSF. First, a review of research in the field will be provided. This will then be followed by a detailed explanation of these three methods and then a description of the model systems that were used in this comparison. Following this, the results will be shown followed by a concluding discussion to follow.

Chapter 2

Literature Review

The majority of research relating to network reconstruction has been within the past decade, as the amount of data needed before that was not available. After the sequencing of the human genome, followed by many other organisms, people wanted to know what this sequence of three billion base pairs did. Thus, dynamic data of different kinds of cells began to appear. The majority of this was attained with microarrays as this allows for thousands of gene concentrations to be measured at one time. This was followed by advancements in mass spectrometry that allowed for high resolution dynamic data of the protein concentrations in cells (known as the proteome) to be obtained. The deluge of data is enormous and much important information about biological processes can be gleaned from it.

The first attempt at network reconstruction was built off of correlation studies. Correlation became an important way of understanding possible causation almost a century ago when Pearson published his method of finding the correlation coefficient (often termed r in statistics) of a given dataset (Aldrich, 1995). One of the issues with this is that direct and indirect connections are extremely difficult to resolve (Almudevar et al., 2006; Tenenhaus et al., 2010). Some methods have found ways around this by employing certain cutoff conditions and other criteria (Rice et al., 2005), but they all rely on correlation. An inherent problem with correlation

studies is that it points to where influence may be, but it does not actually guarantee causation where a correlation is found.

Another commonly used method is to model the problem with Bayesian networks (Periwal, 2010). These models explain the probability that a node is connected to others (referred to as the node's parent) by assigning a statistical distribution to its probability. In the basic sense, this can only work for acyclic networks, which are uncommon within biology. Most biological networks are connected in various loops and other feedback structures (Milo et al., 2002), which create cyclic networks. This is generally fixed by unfolding a time-series dynamic response of a system so that each time step represents a unique set of nodes that then connects to the next step. Each step, therefore, allows for the acyclic nature required by Bayesian networks.

One of the most influential algorithms currently being used, BANJO, utilizes this principle and has gained much popularity (Smith et al., 2006). The ability for this implementation to recover network topologies is less than optimal, but this method is commonly used due to its easy user interface. Bayesian networks in general require a lot of data points over time which makes it difficult to work with biochemical networks since this would quickly become very expensive.

Dynamical models using differential equations have been gaining popularity over the years, but many different underlying models have been used. There are two main approaches to this, namely, using nonlinear functions or assuming a linearization around an equilibrium. The nonlinear approaches will generally use Hill equations, a common set of equations to describe chemical behavior (Quach et al., 2007). This approach must then fit the parameters with some algorithm. Usually, this is accomplished with a statistical approach as a heuristic to fit the equations to the time-series given. This is an issue since multiple network structures can admit the same dynamic response (Gonçalves et al., 2007). Thus it is not sufficient to

base a method on this model alone.

The second class of algorithms based on differential equations is composed of those that exploit the ability of a nonlinear system to be linearized around its equilibrium points. When this is done, the resulting approximation is valid when the states of the system remain within a specified distance of this point (a distance that is somewhat arbitrary and may be different for each system). Dynamical structure functions employ this fact, as well as VBSSM, another widely used algorithm.

Variational Bayesian state-space model (VBSSM) is a reconstruction method based on linear differential equations which then use a Bayesian network to approximate the parameters. Maximum likelihood was first used to parameterize the model, but this was found to yield excessive false positives (Rangel et al., 2005). A second approach, based on Bayesian networks, was then used to fit the model to the data (Beal, 2003). This yielded fewer false positives and works better than the maximum likelihood approach.

Since a system can be approximated by an infinite number of structures to yield the same dynamics, DSF approaches the problem not by looking at the progression through time, but by looking at its actual dynamic response (Gonçalves et al., 2007). The dynamic response can be modeled in the frequency domain, where the dynamics for a system has a unique representation. This representation is then factored in such a way to isolate the dynamic response of the system based on how each node affects the other. The mathematical foundation for this requires certain elements to be fixed in order to infer the connective structure. Thus a new experimental paradigm is imposed. The stipulation is that a perturbation must be applied that will affect one node before all others. This will then move the nodes' concentrations slightly away from the equilibrium point in accordance with the dynamics of the system. Doing this for each of the nodes will give enough data to infer the unique topology of the system.

Since many of these algorithms have been established for a while, some comparisons have been done. A 2009 study compared all of the aforementioned algorithms except DSF and found that none of them performed too well (Hache et al., 2009). Although some of them are frequently used, it is well accepted that the majority of methods fall short of reliably reconstructing the topology of the underlying network. In 2010 a synthetic yeast strain was developed to in vitro test the applicability of these algorithms (Cantone et al., 2009). This has been dubbed the "golden standard" of network reconstruction as it provides a well-defined network with known topology (Camacho and Collins, 2009).

Chapter 3

Network Reconstruction Algorithms

There are many methods currently available that offer the ability to reconstruct a network given either time-series or steady-state data. The abundance of algorithms has arisen due to the difficulty in identifying the governing dynamics of biochemical systems. These dynamics are much more complex than other systems that have been successfully modeled, such as electrical circuits. There is no defined notion of a *governing equation*, such as can be found in other disciplines. There are many attempts at mathematical formulization, but no one formulation is decidedly better than the others. Due to this, different methods to identify network dynamics will focus on certain models to help design the algorithm.

Formulations of the governing equations have, in recent years, focused on probabilistic and differential equation approaches. With the advent of computers, the Bayesian methods have become tractable and a feasible option for modelers (Bolstad, 2007). This model has been embraced because it can help account for noisy situations as well as stochastic effects of molecular interactions. Another common framework is differential equations, with the majority of algorithms focused on fitting parameters to some system of equations. Both of these have shown promise and are currently used by practicing biologists.

This section will address the methodology behind the algorithms chosen for this

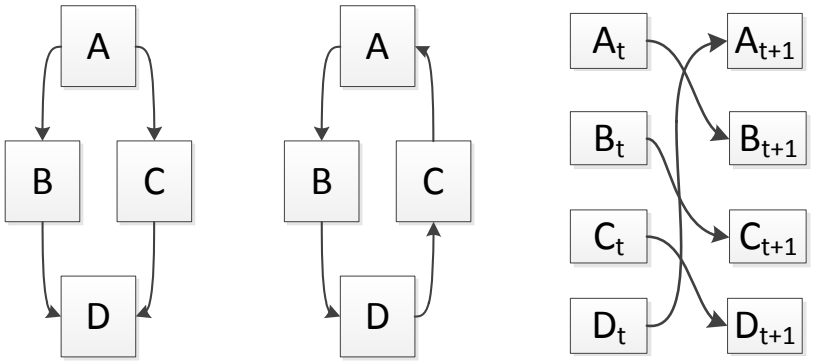
comparison. The criteria used for inclusion were that they need to be commonly used, been shown to work reasonably well, and perform adequately with small datasets. They include Dynamical Bayesian State-Space Models (VBSSM), Bayesian Network Inference with Java Objects (BANJO), and Dynamical Structure Functions (DSF). VBSSM is a combination of differential equations within a Bayesian framework, BANJO is a pure Bayesian network, and DSF is a control-theoretic approach to differential equations. The criteria were shown to be satisfied by these algorithms in Hache et al. (2009).

3.1 BANJO

Bayesian Network Inference with Java Objects (BANJO) is a popular package to reconstruct networks. This algorithm is used by many researchers from neural networks to ecological systems to biochemical systems. It incorporates both the idea of Bayesian networks and dynamic Bayesian network, the dynamic analog of pure Bayesian networks, to allow cycles to appear in the resulting network.

A Bayesian network is an acyclic directed graph (DAG) that contains probabilistic information about the state of each node dependent on those that came before it. This causes the state of a species of interest to be conditionally dependent upon the nodes that affect it. The nodes in a Bayesian network correspond to a random variable, which, in the context of network reconstruction, is the state of the species of interest. Edges are composed of directed arrows, implying a causational pair between the two. If an arrow extends from node A to node B, then it is said that A is the *parent* of B, denoted by $\pi(B) = A$. Each node X_i has a conditional probability distribution that quantifies the effect the set of parent nodes has on it. Combining these conditions, a Bayesian network can be defined.

Bayesian networks have the unique property of allowing a concise joint distri-



(a) (b) (c)

Figure 3.1: *Graphs representing (a) a directed acyclic graph (DAG), (b) a non-Bayesian network that contains a cycle, and (c) a dynamic DAG that represents the cyclic graph in (b)*

bution over all variables to be defined. Each system of n species is defined with a random variable X_i expressing the concentration of each of the species x_i as a distribution. The joint probability is the probability that $X_i = x_i$ for some concentration x_i, expressed as $P(x_1, \ldots, x_n)$. The value of this is expressed by the formula

$$P(x_1, x_2, \ldots, x_n) = \prod_{i=i}^{n} P(x_i | \pi(X_i)), \tag{3.1.1}$$

where $\pi(X_i)$ is the set of parents of the random variable X_i.

Consider a system composed of four nodes as shown in Figure 3.1(a) as a representative of an arbitrary biochemical system. In this system, A is the parent of both B and C, thus causing B and C to be independent of each other. However, D is dependent on both B and C. This is an example of a directed acyclic graph since at no point does a node affect another node that then gets propagated back to itself. The resulting joint distribution would be:

$$P(a, b, c, d) = P(a)P(b|a)P(c|a)P(d|b, c) \tag{3.1.2}$$

These probabilities can be modeled either discretely or continuously. Although continuous variables would give a better fit due to the continuous nature of biochemical concentrations, BANJO uses a discretization method to divide the continuous spectrum into a fixed set of intervals. Currently, only five levels of discretization are supported (Sladeczek et al., 2006).

These Bayesian networks work well for some applications where there are no loops, but most biological networks of interest have some sort of feedback which is prohibited in this model. An alternative formulation is to consider each node at a specific time point a separate node in the network. This is called a dynamical Bayesian networks. Unfolding a time-series dataset into a series of networks allow loops to form, as shown in the examples in Fig. 3.1(c).

The joint distribution, then, must account for all of the states dependent on the previous time points. Let x be the state vector for all those of interest (in our example, it would be $x = \begin{bmatrix} a & b & c & d \end{bmatrix}$) n the number of states (in our case $n = 4$), and the number of time points be T. The notation x_t^i implies the ith component of x at time point t. The joint distribution becomes:

$$P(x_1, x_2, \ldots, x_n) = \prod_{t=1}^{T} \prod_{i=1}^{n} P(x_t^i | \pi(x_t^i)) \tag{3.1.3}$$

Once these joint distributions are defined (either dynamic or not), BANJO will attempt to find the topology that yields the best score. This is done by hill climbing or simulated annealing. The hill climbing method attempts to find the best score by following the gradient as it rises. Simulated annealing, an analogy derived from metallurgy, is a version of hill climbing that will randomly move in a non-optimal direction in an attempt to find the global optimum. The random movement is controlled by a parameter defined as the temperature T, that cools at some rate as time increases. It has been shown that this will converge to the global opti-

mum, although this is dependent on the rate of cooling, which is difficult to define. These are similar methods, but the simulated annealing tends to find the global, rather than a local, optimum better. The network structure that yielded this optimal score is then considered the best estimate.

3.2 Variational Bayesian State-Space Models

Variational Bayesian State-Space Models (VBSSM) incorporate the idea of Bayesian networks, but add dynamics to it by using state-space models. These models are concise representations of linear dynamical systems. These models incorporate information about the dynamics of both observed and hidden variables, as well as inputs and possible noise. It is the simplest of models, but is powerful in its ability to describe system dynamics.

The algorithm employed by VBSSM views the processes in discrete time. Each time point can be affected by the dynamics of the system (represented by A), the inputs h_t to the states and u_t (with B and D defining which states or outputs receive the inputs), and noise w_t and v_t (Rangel et al., 2005).

$$x_{t+1} = Ax_t + Bh_t + w_t \tag{3.2.1}$$

$$y_t = Cx_t + Du_t + v_t \tag{3.2.2}$$

It is assumed that the distributions of the random variables are Gaussian, providing enough knowledge to begin constructing the probability densities. The conditional probabilities of the state and observable outputs are then given by

$$P(x_{t+1}|x_t, h_t) \sim N(Ax_t + Bh_t, Q) \tag{3.2.3}$$

$$P(y_t|x_t, u_t) \sim N(Cx_t + Du_t, R). \tag{3.2.4}$$

A joint probability can then be formed by constructing a Bayesian network based on the dynamics of the state-space model. The joint distribution would give a distribution that accounts for all of the states, observations, and inputs at all time points (represented by $\{x_t\}, \{y_t\}, \{h_t\}, \{u_t\}$, respectively):

$$P(\{x_t\}, \{y_t\} | \{h_t\}, \{u_t\}) = P(x_1) \prod_{t=1}^{T-1} P(x_{t+1} | x_t, h_t) \prod_{t=1}^{T} P(y_t | x_t, u_t) \tag{3.2.5}$$

The correct structure should yield the largest likelihood when applied to this model. However, the task of finding the optimal solution is difficult and, in most cases, intractable. A common technique is one called expectation-maximization (EM). The EM algorithm is a process by which the expected value of the distribution is calculated, followed by a step where the parameters are maximized. This is repeated until it converges to an answer. This form of model selection, however, was shown to infer a high number of false positives (Rangel et al., 2005). A Variational Bayesian version of the EM algorithm was used instead to address this issue.

Variational Bayesian techniques are used to approximate intractable integrals, which, in the case of Bayesian methods in general, arise often. A distribution, $q(\theta)$ is defined to be a lower bound for the true distribution with parameter θ which consists of the unknowns in the system: A,B,C,D and the covariance of the noise vectors Q and R. An EM algorithm is then applied to maximize the expectation of $q(\theta)$ in order to converge to the most likely parameters for the state space model. For full treatment of this process, please refer to Beal (2003); Beal et al. (2005); Rangel et al. (2004, 2005). During this process, the size of A and other variables are adjusted to account for an arbitrary number of hidden nodes. Since this algorithm attempts to infer all variables in θ, if A is larger than the number of nodes measured, the extra space contains information about the dynamics of the hidden states. The Variational Bayesian EM algorithm is applied in successive steps to dif-

ferent values of hidden states k in order to find which one produces the highest likelihood over all tested number of hidden states. It is assumed that the state-space system is both fully controllable and observable. This ensures that the system can be identified. These conditions imply that from any starting point x_0 the system can be taken to a state x_t with a good choice of noise vectors $\{w_t\}$ as well as given any output y_t the initial state x_0 can be determined. Once a set of parameters θ are found yield the highest likelihood, the structure of the network can be determined by considering the structure of $CB + D$.

3.3 Dynamical Structure Functions

Dynamical structure functions are a decomposition of the input-output data into two matrices: computational structure and control structure. This decomposition is derived in Gonçalves and Warnick (2008), of which an overview will be given here in order for the reader to have a basic understanding of how the reconstruction is possible.

A state-space model will provide information regarding connections between any given node in a system. However, there is an infinite number of state-space models that may give the same input-output relationship, even with the same number of states. More dynamics can be added via the hidden states (unmeasured nodes) while leaving the input-output relationship the same. Since network reconstruction utilizes input-output response dynamics to infer connective edges, a unique state-space representation is impossible to recover. However, all of these state-space models that yield the same input-output dynamics can be represented by the same transfer function, a function that relates the input to the output in the frequency domain, as opposed to the time domain, via the Laplace transformation. Transfer functions, however, do not yield any information about how a system is connected.

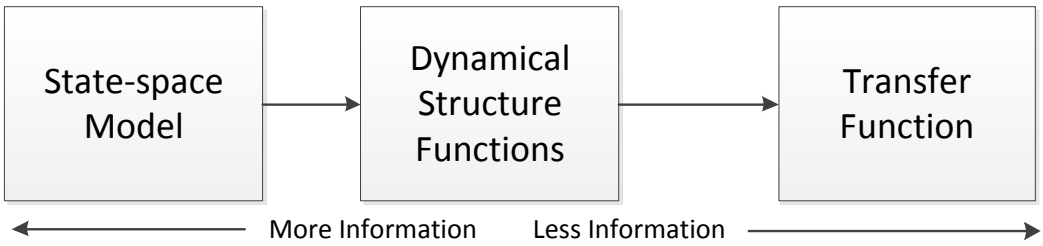


Figure 3.2: *Scale showing relative abundance of structural information available to different mathematical representations*

Dynamical structure functions provide an intermediary step in which more information about the structure is available than a transfer function, but less than in a state-space model (see Fig. 3.2). It provides information about the dynamics of a system, which is contained in both transfer functions and state-space models, and the structure of observable nodes, which is not available with transfer functions but is with state-space models. They lack the ability to provide any information regarding hidden nodes, something that state-space models contain. However, knowing the structure between the hidden nodes is useful enough, which can grow to include hidden nodes as they become known and more data is obtained.

The dynamical structure can be derived by considering a state-space model given by Eq. 3.3.1 below.

$$\dot{x} = Ax + Bu \tag{3.3.1}$$

$$y = Cx \tag{3.3.2}$$

This is then partitioned into the observed and hidden states without changing the actual system dynamics:

$$\begin{bmatrix} \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u \tag{3.3.3}$$

$$y = \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} \tag{3.3.4}$$

where $x = \begin{bmatrix} y'z' \end{bmatrix}' \in \mathbb{R}^n$ is the full state vector, $y \in \mathbb{R}^p$ are the measured states, and $z \in \mathbb{R}^{n-p}$ are the hidden states. The vector $u \in R^m$ is the control input as used earlier. When the Laplace transform is applied to this system, it yields:

$$\begin{bmatrix} sY \\ sZ \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} Y \\ Z \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} U \tag{3.3.5}$$

Solving for Z and substituting back into Eq. 3.3.3 yields:

$$sY = WY + VU \tag{3.3.6}$$

where W and V are a change of variable, $W = A_{11} + A_{12}(sI - A_{22})^{-1}A_{21}$ and $V = A_{12}(sI - A_{22})^{-1}B_2 + B_1$. If we introduce a variable $D = \text{diag}(W_{11}, W_{22}, \ldots, W_{pp})$, then

$$(sI - D)Y = (W - D)Y + VU \tag{3.3.7}$$

$$Y = QY + PU \tag{3.3.8}$$

where $Q = (sI - D)^{-1}(W - D)$ and $P = (sI - D)^{-1}V$. It is apparent that Q is a matrix of transfer functions relating Y_i to Y_j when $i \neq j$, as each diagonal entry on Q is zero. This pair (Q, P) then describes the dynamical structure function of the system. The function Q refers to the internal structure of the network and P refers to the control structure of the network.

Without any prior information, which is being assumed in order to recover a *de novo* network, there are some conditions that must hold in order to recover a

network. The transfer function of the system, given by G, will admit a dynamical structure pair (Q, P) such that

$$G = (I - Q)^{-1}P \tag{3.3.9}$$

Assuming that $m = p$, that is the number of measured states is equal to the number of input states. Assuming this is achieved by having each perturbation affect a specific measured state (effectively a diagonal P), the dynamical structure falls out very nicely as G becomes full rank. Thus, $H = G^{-1}$ characterizes (Q, P) as follows:

$$Q_{ij} = -\frac{H_{ij}}{H_{ii}} \text{ and } P_{ii} = H_{ii}^{-1} \tag{3.3.10}$$

If biological systems had no noise (either inherently or in measurements) and everything acted in a linear way, then this would suffice to recover any network with data gathered as specified above. However, this is not the case. Most data will be noisy and have nonlinear characteristics which will cause a linear representation of the system above to incorrectly identify the network. But, since the dynamical structure function is related to the transfer function of the system, the correct system can be identified through uncertainty modeling and model selection criteria.

Uncertainty modeling allows for various unmodeled dynamics, such as nonlinearities and stochastic effects, to be accounted for, thereby yielding a model that performs reasonably well. With the dynamical structure functions as currently explained, the decomposition will yield a fully coupled computational structure, implying that every state in the system interacts with every other state. This is generally not the case as species in biological networks tend to interact with only certain species as defined by their specificity, such as the lock and key model of enzyme kinetics. To account for this, a distance must be defined as the difference

between the true measured input-output dynamics given by G and the ones given by a specified $\hat{Q} \in \mathbf{Q}$ where $\mathbf{Q}$ is the set of all Boolean structure of Q obtained by zeroing out various elements in the computational structure. Since a biological system is generally loosely connected, it can be assumed that some of the values in Q are just noise as there really is not a connection between them.

This uncertainty can be modeled in many ways, of which care must be taken to ensure that the outcome will lead to a convex problem. Convexity is important as that ensures that there is a single minimum that is also the global minimum. The choice given in Yuan et al. (2011) is feedback uncertainty. This choice of uncertainty will take the transfer function G and add it to a Δ that is then recursively fed back into G. Letting the output of the true system be defined as $\hat{G}$, we can solve for its transfer function:

$$\begin{aligned} z + \Delta\, z &= Gw \\ (I + \Delta)\, z &= Gw \\ z &= (I + \Delta)^{-1} Gw \end{aligned} \tag{3.3.11}$$

Thus the transfer function of the true system from the input u to the output y is $\hat{G} = (I + \Delta)^{-1}G$. Since Δ represents the possible perturbations (due to noise or nonlinearities) that are not accounted for in the ideal model Δ, we want to minimize the size of Δ, which is done by minimizing its norm. However, an issue arises that this will always yield the fully coupled network. Thus, the number of connections is penalized if too high. This is done with Aikaike's Information Criterion which puts a weight on the number of connections; therefore, as the number of connections increase, the score increases also. In this way, a structure representing the most likely network topology is achieved in $\hat{Q}$.

Chapter 4

Network Models

This chapter presents an overview of the different methods used to perform the comparisons. First, the *in silico* synthetic models will be described, followed by a description of the mathematical model of the IRMA network.

4.1 Synthetic Networks

An *in silico* model of a network allows for a comparison to be made under controlled conditions. A downside to this, of course, is that no model is perfect, especially of biological systems. There are some basic formulas that are used to model biological processes, but these do not come close to an exact representation of the natural process, unlike the laws of physics or thermodynamics; they are more a guideline to the modeling process. Biological systems can interact in many ways with many different governing equations.

Although natural systems are inherently nonlinear, linear systems can be used to model them. Linear models are the simplest and most easily defined class of models. It can be argued that if a linear network cannot be recovered from an algorithm, then there is little chance that it will be recovered when applied to nonlinear data. The linearization can be done around the equilibria of the system assuming

that they are stable. Much insight can be gained through this approximation, as there are many tools available for the analysis of linear systems compared to the relatively few for nonlinear systems. With this in mind, simulations were conducted based on simple linear models. These simulations were constructed by applying a step input to each of the measured nodes in order to cause a perturbation that will change to which states the system equilibriates. Based on the current criteria for successful reconstruction with dynamical structure functions, these were limited to systems that would converge to a non-zero, stable equilibrium.

Thus, assuming a state-space representation as in Eq. 3.3.1, two requirements must be satifisfied for a system to be used in this simulation study. The first one is that A must be Hurwitz and the second is that $CA^{-1}B \neq 0$. The first criterion refers to the stability of the matrix. If a matrix is Hurwitz, all of the eigenvalues are strictly negative implying stability in the solution of the state-space model. The second one ensures a non-zero equilibrium.

These conditions are derived by solving the state-space model with a step input (i.e., $u(t)$ is the Heaviside function). The first step is to take the Laplace transformation of the state-space to solve for x, with the Laplacian variable s:

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{4.1.1}$$

$$sX(s) - x_0 = AX(s) + BU(s) \tag{4.1.2}$$

$$(sI - A)X(s) = x_0 + BU(s) \tag{4.1.3}$$

$$X(s) = (sI - A)^{-1}x_0 + (sI - A)^{-1}BU(s) \tag{4.1.4}$$

$$x(t) = e^{At}x_0 + \int_0^t e^{A(t-\tau)}Bu(t)d\tau \tag{4.1.5}$$

Inserting this into the state-space equation and remembering that the Heaviside function is defined as zero when $t \leq 0$ and one otherwise we get the solution at any

time t:

$$y(t) = Ce^{At}x_0 + \int_0^t Ce^{A(t-\tau)}Bu(t)d\tau \tag{4.1.6}$$

$$= Ce^{At}x_0 + CA^{-1}e^{At}B - CA^{-1}B \tag{4.1.7}$$

To find what the steady state values of the system will be after the transient parts have vanished, we take the limit of $y(t) \to \infty$ which leaves us with $-CA^{-1}B$. Since this cannot be zero, we have condition (2). The limit can only be taken if A is Hurwitz, otherwise the matrix exponential will approach infinity and the system will have no steady state, giving us condition (1).

As discussed previously, simulating allows the capabilities of an algorithm to be assessed. The focus of this study is to investigate the effect hidden states have on reconstruction. This is a common phenomenon encountered in biology, since often only a few of a species in a network are known, and only a fraction of those can be measured. The amount of hidden states of the original system may decrease as more nodes are learned.

Hidden nodes represent the nodes that are unknown or simply not measured within a system. In the Dynamical Structure Function model, this is expressed in the vector z, while in Variational Bayesian it is within the size given to k. Since hidden nodes in a given system are arbitrary and could be few to many within a system of interest, it is important that methods that claim to reconstruct network topology have ways to account for this.

In order for the simulation to represent a realistic system, artificial noise was generated and added to the signals from a Gaussian distribution. The results of simulating a system yielded a response with a pure signal. Noise was added to the signals to ensure a signal to noise ratio of around 120 dB. This is what previous experiments had shown would generally be good for reconstruction. Signal to noise

ratio is a standard measurement in signal processing that gives the ratio of the power of the signal to the noise in the signal. This was calculated with the following equation,

$$SNR = 20 \log \left(\frac{1}{ij} \sum_{i,j} \frac{d_{ij}^2}{n_{ij}^2} \right), \tag{4.1.8}$$

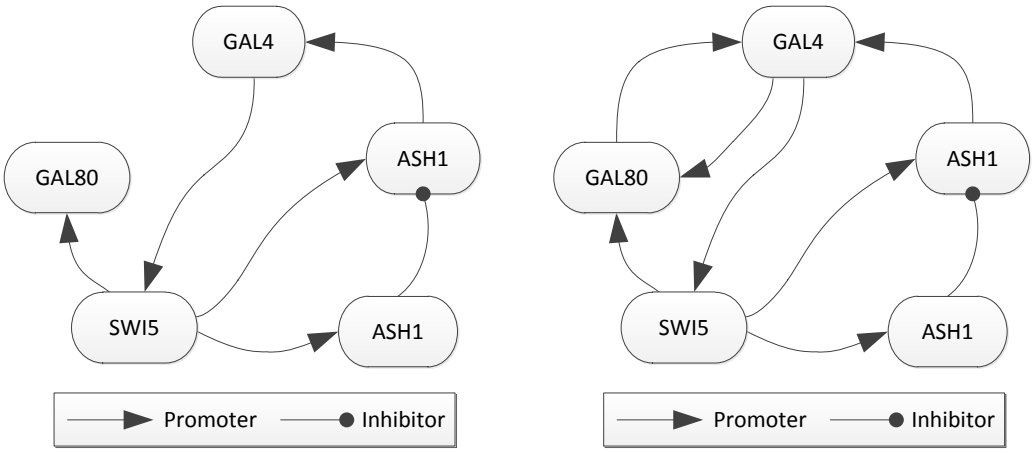
where $d \in D$ is the data, or measured signal, and $n \in N$ is the noisy signal. Both D and N are of dimension $i \times j$.

4.2 IRMA Network

The IRMA network is a synthetic network described by Cantone et al. (2009). This network was developed in order to provide a golden standard in the testing of network reconstruction methods. A dilemma arises when an algorithm is developed in that there needs to be an objective standard to which it can be compared to other methods. By creating this network, a standard now exists that can show how well an algorithm works.

The network was synthetically inserted into a strain of yeast *Saccaromyces cerevisiae* with known topology and dynamics. The network is designed as a transcriptional network of five genes. Each gene is transcribed and translated into a protein that in turn activates or inhibits the transcription of the next gene. In Fig. 4.1 the interactions between both mRNA and proteins are shown. In order to perturb nodes in the system, a set of plasmids containing a promoter and the gene of interest were created. The promoter is spliced into a plasmid (small circular strand of DNA) such that by adding a specific chemical, the gene is turned on at a semi-regular rate. This effectively allows a step input to be applied to the system to a specific node.

The IRMA network has a full mathematical formulation that can be examined



(a) (b)

Figure 4.1: *The true structure of the IRMA network for (a) mRNA and (b) proteins*

to understand its dynamics (Marucci et al., 2010). The original system of equations contains all of the mRNA and protein species and the relationships among them. When the original experiments were run by Cantone et al., the focus was on the transcriptional network, so all of the protein species were considered hidden states. A time delay was applied, since the SWI5 is delayed by about 100 s. They showed that the simulations of this system align very well with experimental results.

The model accounts for two different growth conditions that can affect the dynamics of the system. The rate of transcription changes depending on if it is grown on glucose or galactose, two forms of sugar. This is accomplished in the model by adjusting various parameters. Since this affects the dynamics of the system, it is hoped that an algorithm can recover the same structure independent of which growth medium was used.

For the purposes of the simulation study, as a preliminary step, a simulation was performed on the mathematical representation. This was accomplished with the `dde35` command to solve the delay differential equation. Parameters were assigned based on those found in Cantone et al. (2009). A step input was applied to simulate the same effect that the plasmid has in the biological system.

Chapter 5

Results

The results will be presented in two sections, with the first exploring the results of the synthetic networks followed by the results of the simulation of a real network. Any algorithm that reconstructs network topology needs to identify both connections that exist and the lack of connections between specific nodes. The analysis for the reconstruction, therefore, uses both sensitivity and specificity to identify how well an algorithm works.

Specificity is the ability to accurately discern where the absence of a reaction is, while sensitivity is the ability to detect the incidence of a reaction occurring between two species. If the number of true positives is TP, false positives is FP, true negatives TN, and false negatives FN, then these are defined as:

$$Specificity = \frac{TN}{TN + FP} \tag{5.0.1}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{5.0.2}$$

These give insight into the efficacy of the algorithm but not a comparative tool to compare how well one performs against another. A single metric is needed to accurately compare algorithms. A common tool for this is the F-Measure which is a generalization of the harmonic mean of a dataset (van Rijsbergen, 1979). The F-

Measure (Eq. 5.0.5 is defined in terms of precision (Eq. 5.0.3) and recall (Eq. 5.0.4).

$$P = \frac{TP}{TP + FP} \tag{5.0.3}$$

$$R = \frac{TP}{TP + FN} \tag{5.0.4}$$

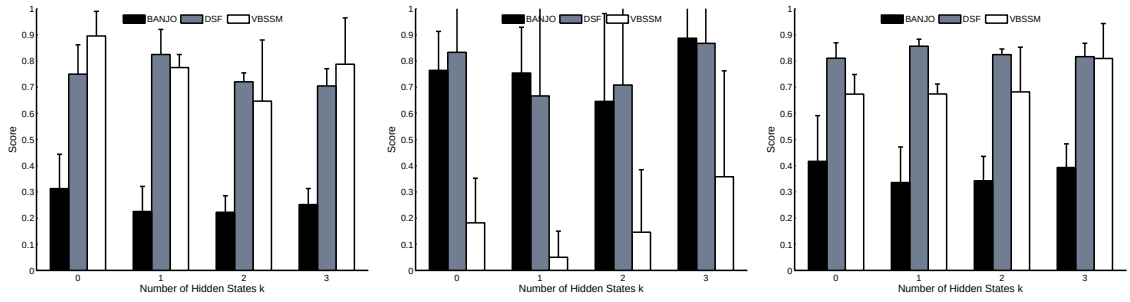
$$F = \frac{2PR}{P + R} \tag{5.0.5}$$

5.1 Synthetic Networks

The synthetic networks were created and simulated to a nonzero steady-state. Systems of four observed nodes were used, with anywhere from no hidden states to three. Artificial noise was added to simulate data more likely to represent biological processes. These statistics are represented graphically in Fig. 5.1 and in tabular form in Table 5.1.

The analysis for each of the networks were accomplished with a t-test performed via Matlab's Statistics Toolbox. This test was performed with $\alpha = 0.05$ so any p value greater than 0.05 would be statistically similar. The comparison was performed for DFS to BANJO and DFS to VBSSM on the sensitivity, specificity, and F Measure metrics.

The sensitivity of VBSSM was statistically the same as DSF when $k = \{1, 2, 3\}$.



(a) Sensitivity (b) Specificity (c) F Measure

Figure 5.1: *The recovery statistics for a random sample of synthetic networks with four observed states and varying number of hidden states k*

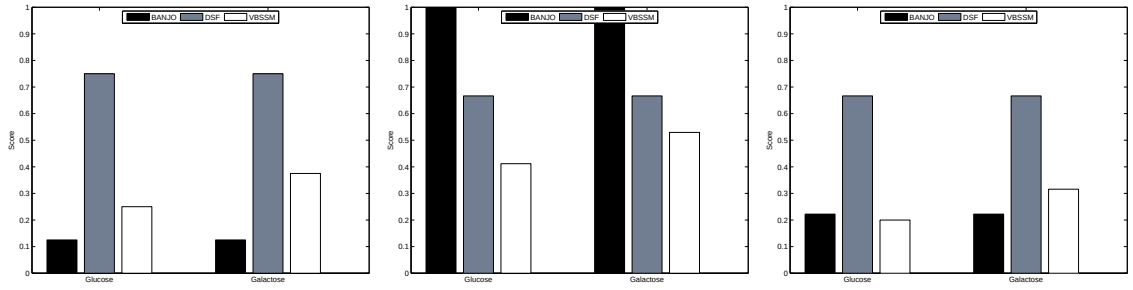
Table 5.1: *The scores and p-values for each of the experiments*

		k=0		k=1		k=2		k=3	
		Score	**p**	μ	**p**	**Score**	**p**	**Score**	**p**
BANJO	**Sensitivity**	0.313	0.000	0.225	0.000	0.222	0.000	0.251	0.000
	Specificity	0.764	0.642	0.754	0.693	0.646	0.758	0.887	0.823
	F-Measure	0.417	0.000	0.336	0.000	0.342	0.000	0.393	0.000
VBSSM	**Sensitivity**	0.896	0.035	0.775	0.390	0.647	0.391	0.788	0.079
	Specificity	0.182	0.001	0.050	0.021	0.146	0.008	0.358	0.000
	F-Measure	0.673	0.005	0.674	0.000	0.682	0.035	0.810	0.843
DSF	**Sensitivity**	0.750		0.825		0.721		0.705	
	Specificity	0.833		0.667		0.708		0.868	
	F-Measure	0.811		0.857		0.824		0.817	

When $k = 0$, VBSSM performed statistically better than DSF with a $p = 0.035$, while when $k = 1$ and $k = 2$ there was no apparent difference between the scores. When $k = 3$, VBSSM again performed better than DSF, but was not statistically better with $p = 0.079$. BANJO, for all levels of k, had p values in the range of $1e - 10$; so low that there is no statistical relationship between them.

The specificity was statistically similar between DSF and BANJO with a p value averaging 0.743 ± 0.079. In this case, though, DSF and VBSSM differed greatly with DSF having a higher score by an average 0.510 units. The p value associated with DSF and VBSSM ranged between 0.000 and 0.021.

The F-Measure yielded results that merge the sensitivity and specificity results. BANJO was always significantly worse than DSF, with p values extremely low; three significant digits show 0.000. VBSSM, however, had low p values for when



(a) Sensitivity (b) Specificity (c) F Measure

Figure 5.2: *The recovery statistics for a simulation of the IRMA network*

$k = 0$ and $k = 1$ (0.005 and 0.000 respectively). With two hidden states ($k = 2$) p rose to 0.035 and at $k = 3$, DSF and VBSSM performed statistically similarly with $p = 0.843$.

5.2 IRMA Network

The reconstruction of the IRMA network from simulated data is shown in Fig. 5.2 and Table 5.2. Since there was only one dataset for this network, there was no variation and thus no p value could be computed. Comparison will be made based on the difference between the scores for each of the algorithms.

Sensitivity analysis showed that DSF performed better than both VBSSM and BANJO. DSF recovered six times the edges of BANJO on both simulated growth mediums, while it recovered three times the edges than VBSSM on glucose and two times the edges on galactose.

The specificity showed that BANJO, which correctly identified every non-connection, performed better in this category than both DSF and VBSSM. DSF found about $\frac{2}{3}$ of the non-connections, while VBSSM identified a little less than half of the absent connections.

The F-Measure of the resulting data shows that DSF performed better than ei-

		Glucose	Galactose
BANJO	**Sensitivity**	0.125	0.125
	Specificity	1.000	1.000
	F Measure	0.222	0.222
VBSSM	**Sensitivity**	0.250	0.375
	Specificity	0.412	0.529
	F Measure	0.200	0.316
DSF	**Sensitivity**	0.750	0.750
	Specificity	0.667	0.667
	F Measure	0.667	0.667

Table 5.2: *The reconstruction statistics for IRMA pathway*

ther BANJO or VBSSM. DSF performed three times better than VBSSM on the glucose and two times better on the galactose. The results for BANJO were the same for both simulated mediums, with DSF performing three times better.

Chapter 6

Conclusion

6.1 Discussion

The results indicate that there is a strong reason to consider DSF superior to BANJO and VBSSM when trying to reconstruct network topology. However, it is apparent that sometimes the sensitivity or specificity of one of the other algorithms is better than DSF. Since one is not an indicator of high success on its own, this helps support the need to use DSF.

Although generally the ability to discern both the presence as well as the absence of a connection is vitally important to correct network inference, the importance of this may be decreased under some circumstances. If the only information needed is whether or not two species are directly or indirectly connected, then a sensitive method is needed. On the other hand, if the question pertains to whether or not two species are disconnected, then it should be fairly specific. However, if a *de novo* reconstruction is needed, both specificity and sensitivity should be high. In this case, we want as clear a picture as possible, requiring both a low false-positive rate (specificity) as well as low false-negative rate (sensitivity).

It was shown that the specificity of VBSSM is much worse than the other two methods. This might be explained by the fact that the framework upon which

VBSSM is built tended to overestimate the number of connections (Beal et al., 2005). The first method to select network topology, as discussed in Chapter 3.2, was to use a maximum likelihood expectation-maximization approach. A Bayesian approach was used in order to limit the number of false positives in the inference. However, this result indicates that even the Bayesian approach will yield too many false positives to be very useful. The sensitivity of VBSSM, however, is roughly the same, if not better, than DSF. This is a strong point in favor of VBSSM, especially if the only thing of interest is whether or not two nodes are connected.

BANJO had similar specificity when compared to DSF. Since the difference was not enough to be statistically different, the resulting inference is similar to DSF when it comes to the number of non-connective edges. A reason that BANJO may have a high specificity is because it tends to only output a few edges. Due to this, it will not incorrectly identify too many edges, which keeps the list of true non-connected species more accurate. This presents a problem when trying to infer the topological structure of an uncertain network.

Overall, the F-Measure can be used to rank the algorithms. In this category, DSF is better than both VBSSM and BANJO, except when there are three hidden states. At this point Variational Bayesian performs similar to DSF. This is an interesting anomaly, especially when the F-Measure score of VBSSM is viewed in progression, as it rises as the number of hidden states increase and approaches the score of DSF. This is interesting, and may lie in the fact that both these algorithms have models based upon state-space models. These models account for hidden nodes in the system that affect those that are observed in some unknown way. By taking this into account, they are better equipped to handle high numbers of hidden states.

6.2 Conclusion

The aim of this thesis has been to provide a comparison of Dynamical Structure Functions (DSF) against two popular network reconstruction algorithms. These algorithms, Variational Bayesian State-Space Model (VBSSM) and Bayesian Network Inference with Java Objects (BANJO), have established theoretical bases and have been used in similar fields of application with success. It would thus be useful to have a comparison of a new algorithm against ones currently in use.

In conclusion, the results have shown that for *de novo* reconstruction, Dynamical Structure Functions outperform the other two methods of interest. Although each of the methods tested has its own strong points, generally both specificity and sensitivity are required for reconstruction of any network of interest. Until now, a low threshold of ability has been the norm, with an acceptance of poor performing algorithms. Dynamical Structure Functions exceeds this threshold and performs better than the other algorithms in the study and should be used when experiments allow it to perform the reconstruction of a network.

Bibliography

Aldrich, J. (1995). Correlations genuine and spurious in Pearson and Yule. *Statistical Science*, 10(4):364–376.

Almudevar, A., Klebanov, L. B., Qiu, X., Salzman, P., and Yakovlev, A. Y. (2006). Utility of correlation measures in analysis of gene expression. *NeuroRx : the journal of the American Society for Experimental NeuroTherapeutics*, 3(3):384–95.

Beal, M. J. (2003). *Variational Algorithms for Approximate Bayesian Inference*. PhD thesis, University of Cambridge.

Beal, M. J., Falciani, F., Ghahramani, Z., Rangel, C., and Wild, D. L. (2005). A Bayesian approach to reconstructing genetic regulatory networks with hidden factors. *Bioinformatics (Oxford, England)*, 21(3):349–56.

Bolstad, W. M. (2007). *Introduction to Bayesian Statistics*. Wiley-Interscience.

Camacho, D. M. and Collins, J. J. (2009). Systems biology strikes gold. *Cell*, 137(1):24–6.

Cantone, I., Marucci, L., Iorio, F., Ricci, M. A., Belcastro, V., Bansal, M., Santini, S., di Bernardo, M., di Bernardo, D., and Cosma, M. P. (2009). A yeast synthetic

network for in vivo assessment of reverse-engineering and modeling approaches. *Cell*, 137(1):172–81.

Gonçalves, J., Howes, R., and Warnick, S. (2007). *Dynamical structure functions for the reverse engineering of LTI networks*. IEEE.

Gonçalves, J. and Warnick, S. (2008). Necessary and Sufficient Conditions for Dynamical Structure Reconstruction of LTI Networks. *IEEE Transactions on Automatic Control*, 53(7):1670–1674.

Hache, H., Lehrach, H., and Herwig, R. (2009). Reverse engineering of gene regulatory networks: a comparative study. *EURASIP journal on bioinformatics & systems biology*, 2009:1.

Marucci, L., Santini, S., di Bernardo, M., and di Bernardo, D. (2010). Derivation, identification and validation of a computational model of a novel synthetic regulatory network in yeast. *Journal of mathematical biology*.

Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., and Alon, U. (2002). Network motifs: simple building blocks of complex networks. *Science (New York, N.Y.)*, 298(5594):824–7.

Periwal, V. (2010). Bayesian Inference of Biological Systems: The Logic of Biology. In Szallasi, Z., Stelling, J., and Periwal, V., editors, *System Modeling in Cellular Biology: From Concepts to Nuts and Bolts*, chapter 4, pages 53–71. The MIT Press, Cambridge.

Quach, M., Brunel, N., and D'Alché-Buc, F. (2007). Estimating parameters and hidden variables in non-linear state-space models based on ODEs for biological networks inference. *Bioinformatics (Oxford, England)*, 23(23):3209–16.

Rangel, C., Angus, J., Ghahramani, Z., Lioumi, M., Sotheran, E., Gaiba, A., Wild, D. L., and Falciani, F. (2004). Modeling T-cell activation using gene expression profiling and state-space models. *Bioinformatics (Oxford, England)*, 20(9):1361–72.

Rangel, C., Angus, J., Ghahramani, Z., and Wild, D. L. (2005). Modeling Genetic Regulatory Networks using Gene Expression Profiling and State-Space Models. In Husmeier, D., Dybowski, R., and Roberts, S., editors, *Probabilistic Modeling in Bioinformatics and Medical Informatics*, chapter 9, pages 269–293. Springer-Verlag, London.

Rice, J. J., Tu, Y., and Stolovitzky, G. (2005). Reconstructing biological networks using conditional correlation analysis. *Bioinformatics (Oxford, England)*, 21(6):765–73.

Sladeczek, J., Hartemink, A. J., and Robinson, J. (2006). Banjo Users Guide.

Smith, V. A., Yu, J., Smulders, T. V., Hartemink, A. J., and Jarvis, E. D. (2006). Computational inference of neural information flow networks. *PLoS computational biology*, 2(11):e161.

Tenenhaus, A., Guillemot, V., Gidrol, X., and Frouin, V. (2010). Gene association networks from microarray data using a regularized estimation of partial correlation based on PLS regression. *IEEE/ACM transactions on computational biology and bioinformatics / IEEE, ACM*, 7(2):251–62.

van Rijsbergen, C. J. (1979). *Information Retrieval*. Butterworths, London.

Yuan, Y., Stan, G.-B., Warnick, S., and Goncalves, J. (2011). Robust dynamical network structure reconstruction. *Automatica*, doi:10.1016/j.automatica.2011.03.008.