Statistical Applications in Genetics and Molecular Biology

Volume 10, Issue 1

2011

Article 47

Fitting Boolean Networks from Steady State Perturbation Data

Anthony Almudevar, *University of Rochester*

Matthew N. McCall, *University of Rochester*

Helene McMurray, *University of Rochester*

Hartmut Land, *University of Rochester*

Recommended Citation:

Almudevar, Anthony; McCall, Matthew N.; McMurray, Helene; and Land, Hartmut (2011)
"Fitting Boolean Networks from Steady State Perturbation Data," *Statistical Applications in
Genetics and Molecular Biology*: Vol. 10: Iss. 1, Article 47.

DOI: 10.2202/1544-6115.1727

Available at: <http://www.bepress.com/sagmb/vol10/iss1/art47>

©2011 De Gruyter. All rights reserved.

Fitting Boolean Networks from Steady State Perturbation Data

Anthony Almudevar, Matthew N. McCall, Helene McMurray, and Hartmut Land

Abstract

Gene perturbation experiments are commonly used for the reconstruction of gene regulatory networks. Typical experimental methodology imposes persistent changes on the network. The resulting data must therefore be interpreted as a steady state from an altered gene regulatory network, rather than a direct observation of the original network. In this article an implicit modeling methodology is proposed in which the unperturbed network of interest is scored by first modeling the persistent perturbation, then predicting the steady state, which may then be compared to the observed data. This results in a many-to-one inverse problem, so a computational Bayesian approach is used to assess model uncertainty.

The methodology is first demonstrated on a number of synthetic networks. It is shown that the Bayesian approach correctly assigns high posterior probability to the network structure and steady state behavior. Further, it is demonstrated that where uncertainty of model features is indicated, the uncertainty may be accurately resolved with further perturbation experiments. The methodology is then applied to the modeling of a gene regulatory network using perturbation data from nine genes which have been shown to respond synergistically to known oncogenic mutations. A hypothetical model emerges which conforms to reported regulatory properties of these genes. Furthermore, the Bayesian methodology is shown to be consistent in the sense that multiple randomized applications of the fitting algorithm converge to an approximately common posterior density on the space of models. Such consistency is generally not feasible for algorithms which report only single models. We conclude that fully Bayesian methods, coupled with models which accurately account for experimental constraints, are a suitable tool for the inference of gene regulatory networks, in terms of accuracy, estimation of model uncertainty, and experimental design.

KEYWORDS: boolean network, gene perturbation, Bayesian modeling

1 Introduction

The reconstruction of gene regulatory networks (GRN) using gene expression data has in recent years been established as a viable and important methodology in systems biology. Typically, GRNs are represented as graphical models, in which a gene graph defines gene interactions, which are often annotated by a functional or stochastic description of the interactions (see Shmulevich and Dougherty (2007) or Koller and Friedman (2009) for recent comprehensive surveys of such models).

An important distinction can be made between static and dynamic models. The former usually take the form of connectivity graphs in which edges indicate proximity on a functional pathway. These include relevance networks and Bayesian networks. Alternatively, a dynamic model is able to predict gene state transitions in time. Although nominally a static model, Bayesian networks can be adapted to dynamic modeling by introducing a time index (termed *dynamic Bayesian networks*). These are usually fit using time course data (Sachs, Perez, Pe'er, Lauffenburger, and Nolan, 2005), but the application of this model to steady-state data has also been described in Shmulevich and Dougherty (2007). Another commonly used dynamic model is the Boolean network (BoN). Gene expression estimates are first discretized, usually (but not always) into expressed (one) and unexpressed (zero) values. For BoNs the gene graph is a directed graph annotated by a direct functional relationship between the parents and offspring. This defines an operator A on the product space of gene expression estimates. Despite the apparent simplicity of this model, it has proven to be a powerful tool in systems biology (Kauffman, 1969), particularly when the objective is to predict network response to alternative stimuli. A BoN is usually fit using gene perturbation experiments, in which the level of one or several selected genes is experimentally altered and the subsequent change in expression is measured for genes of interest. It should be noted that evidence has been compiled that discretizing gene expression estimates need not result in significant loss of information and may conform well to actual cell behavior (Shmulevich and Zhang, 2002, Zhou, Wang, and Dougherty, 2003). The ability to reduce the dynamic properties of a GRN to a single operator proves to be crucial to the methodologies proposed here, and dictates our choice of the BoN model.

1.1 Boolean Network Models of Cancer GRNs

BoNs are particularly suitable for research in cancer GRNs aiming at the identification of targets for intervention strategies downstream of oncogenic mutations. Identification of such targets is notoriously difficult and unpredictable, in part because cell regulation is inherently complex. It has also become clear that cancer

progression is associated with profound changes in the genetic networks that control the functioning of the cell. The dynamic properties of the BoN permit the modeling of network structure able to predict such catastrophic alterations of gene regulatory control.

Malignant cell transformation is a highly cooperative process that requires the interaction of a few oncogenic mutations that cause substantial reorganization of many cell features (Hanahan and Weinberg, 2000). This involves synergistic modulation of downstream signaling circuitry at multiple levels of regulation, including gene expression (Lloyd, Obermuller, Staddon, Barth, McMahon, and Land, 1997, Sewing, Wiseman, Lloyd, and Land, 1997, Xia and Land, 2007, McMurray, Sampson, Compitello, Kinsey, Newman, Smith, Chen, Klebanov, Salzman, Yakovlev, and H., 2008).

Genes that show a synergistic response to the combined action of two co-operating oncogenic mutations (e.g. activated Ras and loss-of-function p53) are referred to as *cooperation response genes* (CRG) (McMurray et al., 2008). Notably, resetting the expression of such CRGs to levels found in normal parental cells by individual gene perturbation experiments revealed that a large proportion of these CRGs (14 out of 24 tested) are critical to the malignant state in murine and human cancer cells. In contrast, perturbations of genes responding in a non-synergistic manner had a similar effect in only one out of 14 cases (McMurray et al., 2008). CRGs represent a set of 95 annotated cellular genes, many of which are involved in the regulation of cell signaling, transcription, apoptosis and metabolism, and represent key control points in many facets of cancer cell behavior (McMurray et al., 2008). CRGs are thus potentially critical nodes in gene networks underlying the malignant phenotype, providing an attractive rationale to explain why several features of cancer cells emerge simultaneously out of the interaction of a few genetic lesions (Xia and Land, 2007, McMurray et al., 2008).

1.2 Statistical and Modeling Issues

We introduce three statistical and modeling issues which the proposed methodology will address:

1. Many experiments measure only the *steady state* of a GRN – interpretable as the state reached after the indefinite operation of a dynamic system. A goodness-of-fit measure should therefore be based on the ability of a model to predict an observed steady state.
2. Many perturbations are *persistent* – they are artificially maintained at a given level regardless of the tendency of the cells' GRN. This implies that the observed steady state is a property of a GRN related to, but different from, the

one of interest. A GRN reconstruction methodology must model this effect correctly.

3. For various reasons, there are likely to be a large number of putative GRNs which can be explained by the experimental data. This need not preclude useful inference as long as model uncertainty is explicitly considered. Under this condition, Bayesian methods may be used to assign confidence levels to model features of interest.

We will argue that these issues may be addressed by incorporating BoN models into a computational Bayesian approach (Besag, Green, Higdon, and Mengersen, 1995) that constructs a posterior density on the full model space, from which a simulated sample is used to form inferential statements.

1.3 Overview of Boolean Networks

Before introducing our proposed methodology, we present a brief overview of BoNs. We begin with a formal definition of BoNs, introducing relevant concepts then extend the definition to ternary networks. We conclude with an overview of BoN model inference, using current approaches to motivate our method of inference.

1.3.1 Boolean Network Definition

Formally, a BoN of order n is completely defined by an operator $A : \mathcal{X} \mapsto \mathcal{X}$ on state space $\mathcal{X} = \mathcal{B}^n$. Let $\mathcal{A}$ denote the set of all such operators. The state $x = (x_1, \dots, x_n) \in \mathcal{X}$ represents a vector of gene expression estimates. If we assume that gene expression can be dichotomized according to whether the gene is unexpressed (0) or expressed (1), then $\mathcal{B} = \{0, 1\}$ and $x_i = 0$ if gene i is unexpressed. The operator can be specified by *transition functions*, $f_i : \mathcal{X} \mapsto \mathcal{B}$, $i = 1, \dots, n$, so that if $x = (x_1, \dots, x_n)$ and $x' = (x'_1, \dots, x'_n)$ are elements of $\mathcal{X}$ then $x' = A(x)$ is equivalent to $x'_i = f_i(x)$ for $i = 1, \dots, n$.

Although the formal definition of a BoN does not require a specific graph, this type of model is usefully regarded as a type of graphical model because many of its important properties are naturally expressed in terms of graphs. Let $G = (V, E)$ be an order n directed graph with vertices $V = \{1, \dots, n\}$ and edges E , and let $\mathcal{G}$ denote the set of all such graphs. An edge from j directed to i is indicated $j \rightarrow i \in E$, and this edge implies that j is a *parent* of i and i is a *child* of j . Next, suppose for a given operator A , for each i there is a subset $S_i \subset V$ such that the transition function f_i depends on $x \in \mathcal{B}^n$ only through components x_j for which $j \in S_i$. We then define G for which S_i are the parents of i . If we further assume that each S_i is the minimal

set with this property, then we refer to G as the *minimal graph* of A . This graph is of fundamental importance in modeling GRNs since it specifies which genes interact directly, and specifies the causal order and mathematical structure of those interactions.

Therefore, there are two complementary representations of a BoN. As described above, the BoN is characterized solely in terms of operator A , from which the minimal graph G may be constructed. Alternatively, as a graphical model we are given a graph $G \in \mathcal{G}$ which then constrains the choice of A . If S_i denotes the set of all parent nodes of $i \in V$, then an operator A *conforms* to G if f_i depends on $x \in \mathcal{B}^n$ only through components x_j for which $j \in S_i$ (G need not be the minimal graph of A). The BoN can then be considered as a pair (G, A) where A conforms to G . This permits a more natural and compact representation of a BoN. For example, if gene 4 is expressed if and only if both genes 1 and 2 are expressed, $S_4 = \{1, 2\}$, and we would set $f_4 = f_4(x_1, x_2) = x_1 \wedge x_2$ (binary operators $\wedge$ and $\vee$ denote logical operators AND and OR, while $\sim$ denotes unary logical operator NOT). Note that we only need to include in the argument of f_i those components associated with S_i . We will find both forms of the BoN useful.

1.3.2 Dynamic properties

The ability to model dynamic characteristics of a GRN is an important feature of the BoN. We define an iterative process $x(t) = A^t(x(0)) \in \mathcal{X}$ for $t = 0, 1, 2, \dots$, where A^t denotes the t th iteration of A , which determines how the network evolves from an initial network state $x(0)$ using concepts from the field of dynamic systems (by convention $x = A^t(x)$ is the identity when $t = 0$). For example, for any initial state $x(0)$ there will be a nonempty set of states F , called an *attractor*, consisting of each state occurring infinitely often in the sequence $A^t(x(0))$, $t \geq 1$. Two definitions are important:

Definition 1 We say $x \in \mathcal{X}$ is a fixed point if $x = A(x)$.

Definition 2 A finite sequence $x(1), \dots, x(m)$ is a cycle if (i) each state of the sequence is unique; (ii) $x(t+1) = A(x(t))$ for $t = 1, \dots, m$; (iii) $x(1) = A(x(m))$.

A cycle of length 1 is a fixed point. By definition, an attractor is equivalent to a cycle, and we denote the set of all attractors by $\mathcal{F}$. A state which is not a member of any attractor is called a *transient* state. Any process either remains entirely in a single attractor, or passes through a finite number of transient states, then remains indefinitely in a single attractor. For an attractor F , the *basin of attraction* is defined

as the set of states $B(F)$ from which a process can reach F . Conversely, a process beginning in state x can reach only one attractor, denoted $F(x)$. If x is not transient, then $F(x)$ identifies the attractor containing x .

Thus, the identification of a BoN model of a GRN permits a characterization of those network states which are transient, and the attractor states which represent the cells' long-term behavior. An attractor may be interpreted as a meta-stable state, so that the BoN is able to model transitions between such states. This in turn leads to the identification of those perturbations capable of altering cell behavior. Evidence for the existence of attractors in actual GRNs that can be modeled by BoNs is reported in Huang and Ingber (2000).

1.3.3 Ternary Networks

The definition of a BoN can be extended to accommodate multiple gene expression outcomes. One can replace outcomes $\mathcal{B} = \{0, 1\}$ with more than two ordered outcomes $\mathcal{B}' = \{0, 1, \dots, k-1\}$, with transition functions now mapping $(\mathcal{B}')^n$ to $\mathcal{B}'$. There are a number of ways to extend a BoN. One approach is to quantize the gene expression levels using a *quantizer function* $Q(y)$, which is a nondecreasing function of gene expression y . The levels of $Q(y)$ may be converted to ordinal values. This process is discussed in Mircean, Tabus, Astola, Kobayashi, Shiku, Yamaguchi, Shmulevich, and Zhang (2003) and is similar to the discretization of a quantitative system of equations.

A *ternary network* is created by setting $\mathcal{B} = \{-1, 0, 1\}$. This is suitable for classifying the outcomes of perturbation experiments when gene expression profiles are available for an unperturbed control sample. Then gene expression estimates from perturbation experiments can be classified as under-expressed, invariant, or over-expressed (-1, 0, 1 respectively) by comparison with the control (Kim, Dougherty, Bittner, Chen, Sivakumar, Meltzer, and Trent, 2000).

1.4 Model Inference

Inference of BoN models of GRNs usually proceeds using gene perturbation experiments, in which expression levels of specific genes are experimentally controlled and the resulting changes in the remaining genes measured. After statistical adjustment, q perturbation experiments yield observed state vectors $Y = (y_1, \dots, y_q)$, $y_k \in \mathcal{X}$. This leads to the inference problem of selecting the model $A \in \mathcal{A}$ which best explains the data Y .

There are two potential types of perturbations – *persistent* and *transient*. A persistent perturbation, denoted $e = (i, s)$, forces gene i to remain indefinitely in

state s . In the case of gene perturbations, this can be accomplished by gene knock-down by stable short hairpin RNAs (shRNA) or enforced mRNA production from gene delivery vectors, depending upon the desired final set point. In contrast, a transient perturbation forces gene i to state s temporarily, after which experimental control ceases and the gene has the potential to change to a different state. In genetics, primarily persistent perturbations are used to investigate GRNs because they allow for greater stability and control of gene expression necessary to uncover phenotypes in long term biological assays.

However, it is important to note that under persistent perturbations the cells under observation are different from the cells which are ultimately of interest, since persistent perturbations alter the cells' GRN. Define a single persistent perturbation experiment by the list of perturbations to which the cell is simultaneously subjected, $E = (e_1, \dots, e_m)$. Let A_E be the operator obtained from A by replacing f_i with $f'_i(x) \equiv s$ for each $e_k = (i, s)$ in E . This new operator conforms to the experimental conditions and describes the behavior of the perturbed cells. We define $F_E(x)$ and $B_E(F)$ to be the attractor and basin mappings corresponding to A_E . The distinction between A and A_E is crucial when considering model inference.

1.4.1 Inference methods - logical consistency

A direct approach to model inference is to identify transition functions $f_1, \dots, f_n$ which are logically consistent with the data Y . There may be no such solution, particularly if there are errors in the data leading to logical inconsistencies, or if the class of functions considered is restricted for practical or theoretical reasons. If so, we may then proceed by analogy with statistical modeling, accepting a best-fit solution that minimizes the number of misclassifications within the data, as in Boros, Ibaraki, and Makino (1998). When consistent solutions exist they will almost certainly not be unique. In this case, a parsimony criterion can be used; algorithms which determine least complex solutions are proposed in Ideker, Thorsson, and Karp (2000), Lahdesmaki, Shmulevich, and Yli-Harja (2003).

1.4.2 Inference methods - modeling attractors

In steady state experiments, it is the attractors which are observed, rather than specific Boolean functions. Therefore, it is natural to see the problem as a search for BoNs constrained by a given set of attractors. An enumerative approach to this problem was considered in Pal, Ivanov, Datta, and Dougherty (2005). Here we propose a model based approach, noting that the ability of an operator A to predict an

observed steady state may be directly evaluated by calculating iterations A^t , provided an initial state may be defined (assumptions leading to this condition will be proposed). An important advantage of this approach is that persistent perturbations may be modeled using iterations of the appropriate operator A_E (noting that the resulting score still pertains to A). Thus, the technical goal is the construction of goodness-of-fit score $D(A | Y)$ which measures the ability of A to predict observed steady states Y in a manner which correctly adjusts for the effect of the persistent perturbations.

1.4.3 Inference methods - Bayesian approaches

In Pal et al. (2005) it was shown that the search space to be considered for the enumeration of attractor-constrained BoNs is extremely large ($\geq 10^{17}$ in a four node example), so that the experimentally observed attractors will not likely be sufficient for resolution to a manageably small number of plausible models. The problem is compounded by the exponentially increasing size of the model space. As argued in Danks, Glymour, and Spirtes (2003) and Wimberly, Danks, Glymour, and Chu (2009), even when structural parsimony constraints are imposed on A (as in Ideker, Thorsson, Ranish, Christmas, Buhler, Eng, Bumgarner, Goodlett, Aebersold, and Hood (2001)), the number of perturbation experiments needed to resolve model uncertainty is too large to be practical even for small networks.

We take the view that the inability of the data Y to resolve the identity of the true model should not preclude useful inference. It may be that specific features of the network, pertaining to attractor structure or pathway relationships, can be reported with a measurable degree of confidence. We therefore propose a computational Bayesian approach involving the construction of a posterior density on the model space $\mathcal{A}$. Bayesian approaches to the inference of network structure have been applied to BoNs in Zhou, Wang, Pal, Ivanov, Bittner, and Dougherty (2004) and to large scale pedigrees in Almudevar (2007). In addition, this Bayesian approach can be used for the design of future experiments which most efficiently reduce model uncertainty (see Ideker et al. (2000), Almudevar and Salzman (2005)).

To illustrate our approach, the proposed methodology will be applied to three synthetic networks designed to highlight the strengths and limitations of using perturbation experiments to reconstruct GRNs. Furthermore, we apply our algorithm to a subset of 9 CRGs reported in McMurray et al. (2008) to demonstrate its ability to uncover potentially interesting biological features of cancer GRNs.

2 Methods

2.1 Model Assumptions

We now present three assumptions on which the inference procedure will be based.

- (A1) The wild-type cell is governed by operator A in the sense that from any initial state x , the cell will reach a steady state defined by $F(x)$. Upon reaching steady state, gene expression can be predicted from $F(x)$.
- (A2) Under perturbation experiment E , (A1) holds for operator A_E .
- (A3) There exists a single state vector X^0 which is a fixed point for any model $A \in \mathcal{A}$ not subject to persistent perturbations.

Assumption (A1) justifies the use of BoNs for modeling GRNs. It essentially states that the wild type cell possesses at least one attractor, which appears in the laboratory as a steady state with respect to gene expression, representing a form of cell stability. Furthermore, there may be more than one attractor, and an accurately fit model will be able to predict to which attractor any given gene expression state will transition and also which perturbations will force the cell to transition from one attractor to another. It is this feature that makes BoNs suitable for modeling cancer progression as a transition between meta-stable states. See Shmulevich and Dougherty (2007, Sec 2.2) for a good discussion of this issue. Assumption (A2) justifies the modeling approach proposed here and is a conjecture which is subject to experimental validation.

Assumption (A3) carries important implications for ternary networks. We assume that an outcome $0 \in \mathcal{B}$ represents an unperturbed (or wildtype) condition, and that under given experimental conditions the cell will not leave this condition. In the CRG network application described below, experimental gene perturbations are calibrated relative to a control level, represented by outcome 0. Over- or under-expression is defined as a gene expression estimate significantly above or below the control level. Note that neither outcome 0 or -1 implies that a gene is not expressed. We define state vector $X^0 = (0, \dots, 0)$, which represents the control cell (in which genes are expressed at control levels). Then (A3) states that X^0 is itself a fixed point.

2.2 Modeling Algorithm

We now present an inference procedure based on the assumptions described above. We are given persistent perturbation experiments, $E_1, \dots, E_q$, and corresponding steady state observations, $Y = (y_1, \dots, y_q)$. Although much of the following can be

easily extended to any $\mathcal{B}$, we will focus on a ternary network, where $\mathcal{B} = \{-1, 0, 1\}$ and 0 corresponds to gene expression in the unperturbed cell. We assume that observations y_j have been converted to observations in $\mathcal{X} = \mathcal{B}^n$. One method of doing this is described in Appendix II.

To score the model, we need a distance $d : \mathcal{F} \times \mathcal{X} \mapsto [0, \infty)$ such that $d(F \mid y)$ measures the compatibility of attractor F with steady state y . The distance must account for the fact that y may be a strict reduction of F . As will be discussed below, several distances are feasible.

Let X_k^0 be equal to the state vector X^0 defined in assumption (A3), modified so that for each perturbation $e = (i, s)$ in experiment E_k component i of X^0 is changed to s . If A is the correct GRN operator, then we expect attractor $F_{E_k}(X_k^0)$ to conform to observed steady state y_k for $k = 1, \dots, q$. Thus, any operator can be scored by the measure:

$$D(A \mid Y) = \sum_{k=1}^q d(F_{E_k}(X_k^0) \mid y_k). \quad (1)$$

It is important to note that to calculate $D(A \mid Y)$ we do not need to calculate the entire attractor structure of the operator A_{E_k} implied in (1). It suffices to calculate the single attractor $F_{E_k}(X_k^0)$, offering a considerable computational advantage. This is easily done by iterating A_{E_k} from state X_k^0 until a cycle is observed.

We next consider the problem of defining a score $d(F \mid x)$ for attractor $F \in \mathcal{F}$ and state vector $x \in \mathcal{X}$. When an attractor F is a fixed point, then D is a distance between two elements $x', x'' \in \mathcal{X}$. Two possible forms are the Hamming distance and L_1 distance, given respectively as:

$$\begin{aligned} d_H(x', x'') &= \sum_{i=1}^n I\{x'_i \neq x''_i\}, \\ d_1(x', x'') &= \sum_{i=1}^n |x'_i - x''_i|. \end{aligned}$$

When F is a cycle, one possibility is to reduce F to a vector and apply d_H or d_1 . For example, let $\bar{F}$ be an n -dimensional vector in which the i th component equals the mean of the i th components of each vector in the cycle. Note that if F is a fixed point, then that fixed point equals $\bar{F}$. As an example, for cycle $F = (0, -1, 1, 1), (0, 1, 0, 1)$ we have $\bar{F} = (0, 0, 1/2, 1)$. Then the score is:

$$d_M(F \mid x) = d_1(\bar{F}, x),$$

noting that d_H is not applicable here, since we may have $\bar{F} \notin \mathcal{X}$. This score must be carefully interpreted. Notice in the given example that the second cycle component

Table 1: This table gives the component-wise contribution to score $d_T(F \mid x)$ for attractor F and steady state vector x . The column indicates the set of nonzero outcomes in a component cycle of F , and the row indicates the steady state component outcome.

Steady State Outcome	Set of nonzero outcomes in attractor component			
	$\{-1\}$	$\emptyset$	$\{1\}$	$\{-1,1\}$
-1	0	1	1	1
0	1	0	1	1
1	1	1	0	1

alternates between -1 and 1, so the corresponding component of $\bar{F}$ is 0. This may be appropriate when gene expression measurements are aggregated over large numbers of cells. While this is a possibility, it would be important to confirm the existence of this effect experimentally.

Another approach is to assume that cycles containing both -1 and 1 in a single component are unlikely, if not impossible. If the attractor component contains such a cycle, it is assumed that a steady state prediction cannot be made and the component contributes a nonzero amount to the score. If a cycle component for gene i contains 0 and 1 (or 0 and -1), we assume that in the steady state observation the gene will be over-expressed (or under-expressed). The cycle can be summarized for each component by the set of nonzero outcomes observed. The score $d_T(F \mid x)$ is constructed by summing contributions from each component. These contributions are given in Table 1 by comparing the steady state outcome to the set of nonzero cycle outcomes. In the above example, if F is compared to steady state vector $x = (0, 0, 1, 0)$, the component contributions are respectively 0,1,0,1 for a total score $d_T(F \mid x) = 2$.

2.3 Search Algorithm and Estimation of Posterior Density

We now describe two related tasks: (1) minimizing $D(A \mid Y)$ over $A \in \mathcal{A}$, and (2) sampling from a posterior density on $\mathcal{A}$ representing models which are plausible given the data. As will be demonstrated below, even when one can find A such that $D(A \mid Y) = 0$, there may be many A which satisfy this. If it were feasible, a reasonable approach would be to constructively enumerate the solution set $\{A \in \mathcal{A} : D(A \mid Y) = 0\}$. If all, or almost all, models in this set shared a common model feature (for example, genes 1 and 2 jointly regulate gene 3), then that feature could then be assigned a high confidence. Of course, such an enumeration will

not generally be feasible, but we may use instead techniques from computational Bayesian methodology to approximate this type of enumeration.

It is important to note that we have no natural probabilistic models with which to construct a posterior distribution (as would exist for Bayesian network models), but we may define a density weighted by $D(A \mid Y)$ of the form:

$$\phi_{\tau}(A) \propto \exp(-D(A \mid Y)/\tau),$$

for any positive constant τ . Suppose $D^* = \min_A D(A \mid Y)$. We wish to find τ^* for which $P\{D(A \mid Y) = D^*\} \approx 1$ under ϕ_{τ^*} (allowing the probability to be slightly less than 1 would be equivalent to accepting some error tolerance). Then a sample from ϕ_{τ^*} is equivalent to an approximation of solution set $\{D(A \mid Y) = D^*\}$, since all models in this set have the same density under ϕ_{τ^*} . Simulated annealing (SA) techniques can be used to determine τ^* , following which a simulated sample from ϕ_{τ^*} can be generated using MCMC methodologies. Details of our implementation of this procedure are given in Appendix I.

Inferential statements about the true model may be constructed using *Bayesian averaging*. Let S be a statement about the model, for example, ‘gene a regulates gene b ’. Then S may be formally defined as the subset of all models in $\mathcal{A}$ for which S is true. The probability of S under density ϕ_{τ^*} is analagous to a posterior probability obtainable from a Bayesian model and is interpretable as a level of confidence for the truth of S (we therefore use the terminology of Bayesian inference in our discussion). This probability can be directly estimated using a simulated sample from ϕ_{τ^*} . A posterior probability of 1 or 0 is interpretable as certainty that statement S is true or false, while probabilities in $(0, 1)$ mean that the truth of S is compatible with, but not necessarily implied by, the data. This interpretation will be useful in the design of further experiments, of which the goal would become the conversion of posterior probabilities from values near 0.5 to 0 or 1.

The SA algorithm used here is based on one proposed in Aarts and Korst (1989). Details are given in Appendix I, but we briefly describe here the general structure. The algorithm proceeds in stages, with stage m consisting of a sample simulation of fixed length N_s from density ϕ_{τ_m} . The *temperature parameter* τ_m is updated to τ_{m+1} using the stage m sample, then $\phi_{\tau_{m+1}}$ is sampled in the subsequent stage. The parameter τ_m always decreases, which has the effect of forcing greater weight in ϕ_{τ_m} on models with smaller scores. However, this is done in a calibrated manner, ensuring that the mean value of $D(A \mid Y)$ under ϕ_{τ_m} decreases smoothly with respect to stage index m . The stages terminate after a positive test for convergence. Conditions under which convergence to the optimum is guaranteed are reported (see also Aarts, Korst, and van Laarhoven (1997)). However, even when these conditions are met, verifying true convergence may be a practical impossibility (see Aarts et al. (1997, Sec 8) for an interesting example). In our

application we may check for convergence by comparing multiple fits obtained by varying the random generator seed. It is important to note that even if convergence to the optimum score could be guaranteed, the optimal solution is unlikely to be unique. We therefore take the viewpoint that convergence to a common posterior density is the more suitable criterion.

3 Results

In this section the methodology will be demonstrated using a set of synthetic network models and gene perturbation data for 9 CRGs reported in McMurray et al. (2008). The primary motivation for this demonstration will be to validate two ideas:

- (1) We expect many network models to be compatible with a given data set; therefore, a random search algorithm cannot be expected to be consistent, that is, varying the random seed or the initial point should yield differing models. Alternatively, if the goal is to estimate a Bayesian posterior density, we should expect the algorithm to be consistent.
- (2) A Bayesian posterior density is correct if a model feature assigned a marginal posterior probability close to 0 or 1 is absent or present in the true model, respectively. A posterior probability closer to 0.5 indicates that the data is not sufficiently informative to resolve the absence or presence of that feature. This type of uncertainty should decrease with additional experimental data.

3.1 Synthetic Networks

3.1.1 Example 1

We present a detailed analysis of a synthetic network using the following steps:

1. define a ternary network, A (transition functions and topology),
2. define a list of persistent perturbations, $E_1, \dots, E_q$, with initial states, $X_1^0, \dots, X_q^0$,
3. determine the attractor, $y_k = F_{E_k}(X_k^0)$ for each k , then construct input steady state data, $Y = (y_1, \dots, y_k)$, and
4. derive an estimate, $\hat{A}$ of A , and construct a posterior density on the model space.

The simulated annealing algorithm was applied to the data Y following which a posterior density was sampled, as described in the Methods Section. Parameter settings are described in Appendix I. In this example, we assume we can

Figure 1: Gene regulatory network for Example 1. Increased (decreased) gene expression is represented by arrows (squares).

observe Y without error, so only one replication of the algorithm is presented. Issues raised by the possibility of error in Y are considered in the CRG example presented in Section 3.2.

Figure 1 shows the topology of the first synthetic network we considered. We define two types of transition functions mapping $\mathcal{B} \rightarrow \mathcal{B}$: *equality*, $f_i(x) = x_j$, represents a mapping of outcomes $(-1, 0, 1) \rightarrow (-1, 0, 1)$ from parent node j to offspring node i , whereas *negation*, $f_i(x) = \sim x_j$, represents the mapping $(-1, 0, 1) \rightarrow (1, 0, -1)$. Nodes 1-6 have one indegree, whereas nodes 7 and 8 are assumed to revert to 0 if altered. This leads to the following transition functions for genes 1-8:

$$f_1(x) = \sim x_4, \quad f_2(x) = x_1, \quad f_3(x) = x_2, \quad f_4(x) = x_3 \\ f_5(x) = x_6, \quad f_6(x) = x_5, \quad f_7(x) = 0, \quad f_8(x) = 0.$$

The transition function for gene 9 can be best represented as a table of values based on the status of genes 7 and 8 (Table 2). Attractors resulting from transient single perturbations of each gene are summarized in Table 3. Because genes 7-9 are not part of a cycle, any transient perturbation would yield a temporary change in gene expression and then reversion back to the null attractor, $\mathbf{0}$. In Table 4 columns represent the steady states resulting from the persistent single perturbations (indicated in bold type). In a real experiment, these values would be generated by comparing the gene expression estimates from the perturbed samples to control samples.

Table 2: Example 1 transition function table for Gene 9 based on the values of Genes 7 and 8.

Gene 8	Gene 7		
	-1	0	1
-1	0	1	1
0	-1	0	1
1	-1	-1	0

Table 3: Cyclic attractors resulting from transient perturbation of single genes. Null attractors, and genes within attractors remaining at outcome 0 are not listed.

Perturb 1-4 with -1 or 1								
Gene1	1	0	0	0	-1	0	0	0
Gene2	0	1	0	0	0	-1	0	0
Gene3	0	0	1	0	0	0	-1	0
Gene4	0	0	0	1	0	0	0	-1

Perturb 5-6 with 1			Perturb 5-6 with -1		
Gene5	1	0	Gene5	-1	0
Gene6	0	1	Gene6	0	-1

Table 4: Observed steady state gene expression after persistent perturbations of each gene to outcome indicated in bold.

Genes	Experiments												
	1	2	3	4	5	6	7	8	9	10	11	12	13
1	1	-1	-1	-1	0	0	0	0	0	-1	1	1	1
2	1	1	-1	-1	0	0	0	0	0	-1	-1	1	1
3	1	1	1	-1	0	0	0	0	0	-1	-1	-1	1
4	1	1	1	1	0	0	0	0	0	-1	-1	-1	-1
5	0	0	0	0	1	1	0	0	0	0	0	0	0
6	0	0	0	0	1	1	0	0	0	0	0	0	0
7	0	0	0	0	0	0	1	0	0	0	0	0	0
8	0	0	0	0	0	0	0	1	0	0	0	0	0
9	0	0	0	0	0	0	1	-1	1	0	0	0	0

Our proposed model inference algorithm was first applied to the data in columns 1-9 of Table 4. Table 5 displays the marginal posterior probabilities for each potential edge in the graph. If we report each edge with a marginal posterior probability > 0.5 , the true network topology is reproduced exactly. It must be noted, however, that a summary of the marginal edge probabilities is not necessarily an exact representation of a network topology. This would be the case if all probabilities were essentially 0 or 1, but when probabilities closer to 0.5 are present, there may be important statistical dependencies among edge events implied by the posterior density ϕ_{τ^*} . This would happen, for example, if two edges have marginal probabilities close to 0.5, but occur together in any single network under ϕ_{τ^*} only with very small probability.

Next, the attractors associated with the persistent perturbations are all correctly identified with probabilities > 0.87 . However, although the true attractors associated with transient perturbations of genes 5-9 are similarly assigned high posterior probability (> 0.86), those associated with genes 1-4 have very low posterior probability (< 0.05). We note that the true attractor for transient perturbations of genes 1-4 is quite complex and would require each connection on the circuit to be present (see Figure 1). While the marginal posterior probability for each edge is quite high (minimum 0.86, see Table 5), there exist a variety of attractors represented in the posterior density. The highest probability attractor is the null with probability ≈ 0.4 . This is the attractor that would occur if each edge in the circuit were not present (which must happen with a probability of at least $1 - 0.86 = 0.14$), but may also occur if the transition functions are not correctly estimated. An important feature of the proposed methodology is that attractor densities for the correctly resolved perturbations have much lower variability. Overall, we conclude that the topology of the network has been correctly identified, along with the attractors for transient perturbations of genes 5-9, but there is uncertainty regarding the attractors for transient perturbations of genes 1-4.

Naturally, this leads to the question of whether our method can be used to predict future perturbations that are best able to resolve model uncertainty, as in Ideker et al. (2000) or Almudevar and Salzman (2005). Our observations suggest that new perturbations of genes 1-4 would be the best choice. Accordingly, the model was refit after negative perturbation of genes 1-4 (columns 10-13 of Table 4). This resulted in a high posterior probability (> 0.84) being assigned to the correct attractor for all of the transient perturbations. This suggests that our methodology can be used to design future experiments that reduce uncertainty in the attractors.

Table 5: Marginal posterior probabilities for graph edges (Example 1). Rows are offspring and columns are parents. By considering all edges with probability > 0.5 (in boldface), we are able to reconstruct the network topology exactly.

	$\emptyset$	G1	G2	G3	G4	G5	G6	G7	G8	G9
G1	0.05	0.00	0.20	0.19	0.86	0.15	0.15	0.16	0.17	0.07
G2	0.02	1.00	0.00	0.13	0.28	0.11	0.14	0.11	0.15	0.06
G3	0.02	0.08	0.99	0.00	0.29	0.12	0.14	0.15	0.14	0.06
G4	0.06	0.09	0.15	0.90	0.00	0.17	0.19	0.17	0.17	0.10
G5	0.07	0.10	0.10	0.10	0.18	0.00	1.00	0.18	0.19	0.09
G6	0.08	0.09	0.09	0.10	0.19	1.00	0.00	0.19	0.17	0.09
G7	0.25	0.09	0.09	0.09	0.30	0.35	0.36	0.00	0.34	0.11
G8	0.25	0.08	0.08	0.09	0.29	0.31	0.32	0.29	0.00	0.27
G9	0.00	0.00	0.01	0.00	0.01	0.01	0.01	0.98	0.98	0.00

Figure 2: Gene regulatory networks for (a) Example 2 and (b) Example 3. Increased (decreased) gene expression is represented by arrows (squares).

3.1.2 Examples 2 and 3

Two additional synthetic networks were developed to further assess our methodology in the same manner as Example 1. The network topologies for Examples 2 and 3 are illustrated in Figure 2. Note that Example 3 has the same topology as Example 2; however, the transition functions differ for genes 2 and 5, introducing negative regulation. We will see that this small change greatly affects model inference.

The only ambiguity in the graphical representation of these networks is in the regulation of gene 2. The transition functions for gene 2 in Examples 2 and 3 are given in tabular form in Table 6. Attractors resulting from transient single perturbations are shown in Table 7 for Example 2 and in Table 8 for Example 3. Note that transient perturbations of genes 4, 8, and 9 result in the null attractor in both

Table 6: Transition function table for Gene 2 based on the values of Genes 1 and 3 in (a) Example 2 and (b) Example 3.

(a)

	Gene 3		
Gene 1	-1	0	1
-1	-1	-1	0
0	-1	0	1
1	0	1	1

(b)

	Gene 3		
Gene 1	-1	0	1
-1	0	-1	-1
0	1	0	-1
1	1	1	0

Table 7: Cyclic attractors resulting from transient perturbation of single genes in Example 2. Null attractors, and genes within attractors remaining at outcome 0 are not listed.

Perturb 1-3 with 1			Perturb 5-7 with 1			
Gene 1	0	0	Gene 5	1	0	0
Gene 2	1	0	Gene 6	0	1	0
Gene 3	0	1	Gene 7	0	0	1
Gene 4	0	1	Gene 8	0	0	1
			Gene 9	1	0	0

examples. In the same manner as Example 1, we treat the matrices of steady state estimates from 9 single persistent over-expression perturbations as the input to our inference algorithm.

By assuming the existence of an edge when the posterior probability is at least 0.5 (the convention adopted in Section 3.1.1), we find that the topology of Example 3 is estimated correctly; however, the same is not true in Example 2. The existence of negative regulation in Example 3 gives additional information regarding topology, relative to purely positive regulation.

We now consider estimation of attractors from transient perturbations. For both examples, whenever the null attractor was correct, it was identified as such with a minimum posterior probability of 0.896. Among the other true attractors only those of Example 2 for genes 1-3 were identified with probabilities greater than 0.5 (0.803, 0.829, and 0.828). For genes 5-7 in Example 2, the true attractors were given the highest posterior probabilities, but these were less than 0.5 (0.415, 0.414, and 0.414). The true attractors in Example 3 for genes 1-3 and 5-7 cycle between 1 and -1 for all active genes (see Table 8) and are likely to be challenging to estimate. However, it is important to note that in all cases, whenever the true attractor does not have the highest probability the posterior density is noticeably more variable than when it does.

Table 8: Cyclic attractors resulting from transient perturbation of single genes in Example 3. Null attractors, and genes within attractors remaining at outcome 0 are not listed.

Perturb 1-3 with 1					Perturb 5-7 with 1						
Gene 1	0	0	0	0	Gene 5	0	0	-1	0	0	1
Gene 2	1	0	-1	0	Gene 6	1	0	0	-1	0	0
Gene 3	0	1	0	-1	Gene 7	0	1	0	0	-1	0
Gene 4	0	1	0	-1	Gene 8	0	1	0	0	-1	0
					Gene 9	0	0	1	0	0	-1

3.2 A Cooperative Response Gene Network

In this section we report on an application of the proposed methodology to the modeling of presumed interactions between 9 CRGs. Here, interest is in deducing how CRG perturbations will alter a control GRN. Therefore, we use a ternary network with $\mathcal{B} = \{-1, 0, 1\}$ where 0 represents gene expression in the unperturbed GRN and -1 and 1 represent significant deviations from control. We consider a GRN based on the following 9 CRGs: HoxC13, Id2, Id4, Jag2, Notch3, Sfrp2, Wnt9a, Zfp385, Plac8.

3.2.1 Model Structure

Table 9 gives the observed steady states for 9 single perturbations involving the 9 CRGs (perturbations of all but Plac8 were over-expression). See Appendix II for details regarding experimental design. The direct perturbation effects are summarized in the connectivity diagram in Figure 3, in which an edge leading from gene a to b is included if there is statistical evidence that a perturbation of a changes the expression level of b . This is equivalent to the *accessibility graph* described in methodology developed in Wagner (2001, 2004). This type of model can be expected to contain redundant edges, since an edge will be included from a to all genes downstream of a , whereas the causal structure could be represented using fewer edges which model sequential, or transitive, causality. Wagner proposes replacing the *accessibility graph* with an *adjacency graph*, defined as the most parsimonious graph which preserves the pathway structure of the accessibility graph (effectively removing *shortcuts*). However, such a model is not feasible in an application in which special interest is placed on the discovery of self sustaining cyclical attractors representing alternative meta-stable states.

Table 9: Steady state summaries for the 20 edge model selected by the thresholding method described in Appendix II. The implied edges are shown in Figure 4(a). Persistent perturbations are indicated in bold type.

	Perturbation								
	hoxC13	id2	id4	jag2	notch3	sfrp2	wnt9a	zfp385	plac8-
hoxc13	1	0	0	0	0	0	0	0	0
id2	-1	1	0	0	0	0	0	0	0
id4	0	0	1	1	0	-1	0	0	0
jag2	0	1	0	1	0	0	1	1	0
notch3	0	1	0	1	1	0	1	1	1
sfrp2	1	0	-1	1	1	1	-1	-1	0
wnt9a	0	0	0	-1	0	-1	1	0	0
zfp385	0	0	0	0	0	0	0	1	0
plac8	-1	0	0	0	0	0	0	0	-1

Figure 3: Connectivity graph derived from the 9 CRG model using a threshold of 20 edges. Merging or splitting of edges is indicated by the solid circles. Increased gene expression or gene repression are represented by arrows or bars, respectively.

Figure 4: A connectivity graph for the 9 CRG model (a) constructed using edges with estimated marginal posterior probability $p(j \rightarrow i) \geq 0.6$ and (b) derived from a ternary network model fit. In both plots merging or splitting of edges is indicated by the solid circles. Edges which are in only one plot are indicated in gray.

Table 10 shows marginal edge probabilities $p(j \rightarrow i)$ obtained by averaging 10 full posterior densities estimated by our proposed algorithm (see below for more detail). All probabilities $p(j \rightarrow i) \geq 0.6$ are highlighted, which are represented by a connectivity graph shown in Figure 4(a). We note that the edges are not functionally annotated (as in Figure 3), and should be regarded as a summary of many networks, rather than a single network model, with edges representing connections of some form present in most models sampled from a posterior density. Many of the general features of the connectivity graph of Figure 3 are preserved in Figure 4a, in particular a pathway structure originating in HoxC13, then passing through Id2, then through a set of three genes Zfp386, Jag2, Notch3, and terminating in a circuit comprising Wnt9a, Sfrp2 and Id4. There is also a potential for feedback from this final circuit to the previous triplet.

Table 10: Marginal edge posterior probabilities $p(j \rightarrow i)$ obtained by consolidating 10 separate posterior densities. The *Null* parent signifies no parent.

	Edge Parent									
	Null	HoxC13	Id2	Id4	Jag2	Notch3	Sfrp2	Wnt9a	Zfp385	Plac8
HoxC13	0.392	0.000	0.214	0.040	0.319	0.307	0.041	0.041	0.185	0.203
Id2	0.063	0.977	0.000	0.082	0.205	0.151	0.099	0.080	0.165	0.174
Id4	0.001	0.001	0.001	0.000	0.532	0.327	0.625	0.511	0.001	0.001
Jag2	0.025	0.042	0.692	0.026	0.000	0.051	0.899	0.110	0.113	0.043
Notch3	0.026	0.070	0.477	0.013	0.998	0.000	0.060	0.012	0.056	0.289
Sfrp2	0.000	0.000	0.000	0.058	0.090	0.000	0.000	0.938	0.914	0.000
Wnt9a	0.032	0.130	0.103	0.189	0.157	0.041	0.824	0.000	0.415	0.109
Zfp385	0.212	0.126	0.072	0.019	0.632	0.611	0.024	0.042	0.000	0.114
Plac8-	0.114	0.671	0.373	0.079	0.188	0.180	0.114	0.076	0.169	0.000

Lastly, we examine the complete ternary network. We note that the network topology (Figure 4b) shows a similar partition into highly connected circuits (Zfp385/Jag2/Notch3 and Wnt9a/Sfrp2/Id4) as can be seen in Figures 3 and 4a. The remaining genes play no important role. In general, the networks shown in Figures 4a and 4b are very similar (edges which are not in common are indicated in gray).

The circuit Wnt9a/Sfrp2/Id4 forms an attractor which alternates over-expression of Sfrp2 with under-expression of Wnt9a and Id4. One advantage of the ternary network model is that it predicts those perturbations which are able to force a transition from one meta-stable state to another. Thus we note that Id4 does not induce feedback, so that the self sustaining behavior is due to Wnt9a and Sfrp2 alone. This circuit can be induced by setting Wnt9a or Sfrp2 individually to their attractor levels. Alternatively it can be induced by an over-expression of Jag2, which forces under-expression of Zfp385 and subsequently Wnt9a, which suffices to induce the cycle. This is predicted by applying operator A iteratively to an initial state characterized by over-expression of Jag2 (see Table 11).

Table 11: Table gives a sample pathway induced by iterating unperturbed operator A from initial state $(0, 0, 0, 1, 0, 0, 0, 0, 0)$ until the Wnt9a/Sfrp2/Id4 attractor is reached (iterations 4-6).

	Iteration					
	1	2	3	4	5	6
HoxC13	0	0	0	0	0	0
Id2	0	0	0	0	0	0
Id4	0	0	0	0	-1	0
Jag2	1	0	0	0	0	0
Notch3	0	1	0	0	0	0
Sfrp2	0	0	0	1	0	1
Wnt9a	0	0	-1	0	-1	0
Zfp385	0	-1	0	0	0	0
Plac8-	0	0	0	0	0	0

The network contains an interesting robustness property. If Wnt9a is over-expressed, or Sfrp2 is under-expressed, contrary to the predicted attractor, a regulatory action on the part of the Zfp385/Jag2/Notch3 circuit is induced which forces a return to the attractor (see Table 12).

We emphasize that this network is only one of many plausible networks. The advantage of Bayesian averaging is that any feature can be assigned a posterior probability. We may then reject or accept features with probabilities close to zero or one. Further experimentation can then resolve features possessing greater uncertainty.

In spite of any modeling uncertainty, a coherent GRN emerges, with a circuit involving genes Wnt9a, Sfrp2, Id4 predicted as a possible meta-stable state. There is some support in the literature for this model. These genes are known to regulate cell proliferation and differentiation (Lasorella, Uo, and Iavarone, 2001, Suzuki, Watkins, Jair, Schuebel, Markowitz, Chen, Pretlow, Yang, Akiyama, Van Engeland, Toyota, Tokino, Hinoda, Imai, Herman, and Baylin, 2004, Xiang, Lin, Zhang, Tan, and Lu, 2008) and have been associated with promoting or inhibiting tumor activity (McMurray et al., 2008). In addition, consistent with their proposed role as regulators in the observed network, each of the down-regulated CRGs – Jag2, Notch, and Zfp385 – behaves as a strong inhibitor of the cancer phenotype upon resetting its expression to levels found in normal parental cells (McMurray et al., 2008). In the present analysis, these genes appear to play a regulatory role with respect to the Wnt9a/Sfrp2/Id4 circuit. Consistent with this prediction, interactions between Notch and Wnt signaling have been observed in the regulation of stem cell proliferation and cell fate choice (Crosnier, Stamatakis, and Lewis, 2006).

Table 12: Table gives a sample pathway induced by iterating unperturbed operator A from initial state $(0,0,0,0,0,0,1,0,0)$ until the Wnt9a/Sfrp2/Id4 attractor is reached (iterations 8-10).

	Iteration									
	1	2	3	4	5	6	7	8	9	10
HoxC13	0	0	0	0	0	0	0	0	0	0
Id2	0	0	0	0	0	0	0	0	0	0
Id4	0	0	0	0	0	0	0	0	-1	0
Jag2	0	0	1	0	1	0	0	0	0	0
Notch3	0	0	0	1	0	1	0	0	0	0
Sfrp2	0	-1	0	-1	0	0	0	1	0	1
Wnt9a	1	0	1	0	0	0	-1	0	-1	0
Zfp385	0	0	0	-1	0	-1	0	0	0	0
Plac8-	0	0	0	0	0	0	0	0	0	0

3.2.2 Statistical Properties of Model Fits

To test the model, the algorithm was run with 30 replicates for each score type, using the parameter values described in Appendix I. We refer to scores using $d_M(F | y)$ and $d_T(F | y)$, as introduced in Section 2.2, as score types 1 and 2, respectively. 20 replications using score type 1 were run for each value of $N_s = 2,000, 10,000$ and 25,000. The type 1 score distributions grouped by N_s (Figure 5a) clearly show the dependence of the closeness of the fit on the N_s parameter. The minimum score achieved was 2.0, giving the approximate number of incorrectly predicted steady state gene expression outcomes out of 72 ($72 = 9 \text{ experiments} \times 8 \text{ unperturbed gene outcomes}$). The mean score for parameter $N_s = 100,000$ was 2.87. This is close to the minimum attainable of 0, so we are presented with the question of whether these fits are viable solutions, or whether overfitting plays a dominant role. In this algorithm, no complexity control is employed other than imposing an indegree bound of $K = 2$. It was observed that most low scoring fits reached bound K for most genes. However, the construction of a Bayesian posterior distribution will permit us to directly assess the degree of overfitting due to network complexity.

A different form of overfitting involves the appearance of extreme cycles in which expression of a single gene alternates between -1 and 1. Conceivably, such cycles introduce enough complexity to permit closer fitting of the steady states. This effect may be especially relevant for score type 1, since if the outcomes within a cycle are equally frequent then the predicted steady state value will be 0, which is the value of most observed outcomes in the data. For each of the 90 score type 1 fits, the number of extreme cycles was calculated (each of 72 predicted outcomes

Figure 5: Analysis of fit replications using score type 1. **(a)** Boxplots of score values $D(A | Y)$ grouped by N_s parameter. **(b)** Plot of extreme cycle count against score $D(A | Y)$. A running mean smoother with bandwidth 0.5 is superimposed. **(c)** A running mean smoother of indicator variable representing greater than 0 extreme cycle count, using bandwidth 0.5.

may have a distinct extreme cycle). A scatter plot of extreme cycle count against score $D(A | Y)$ is shown in Figure 5b, with a running mean smoother superimposed. In Figure 5c the proportion of fits with > 0 extreme cycles is estimated as a function of $D(A | Y)$ using a running mean smoother. The tendency for the model to fit extreme cycles with greater frequency as $D(A | Y)$ decreases is evident. From Table 1, when using score type 2 extreme cycles are permitted, but are always penalized. Figure 6 shows histograms of extreme cycle count grouped by score type (70 fits using score type 1, $N_s > 2,000$, 30 fits using score type 2, $N_s = 100,000$). The two samples were tested against equality using a permutation procedure, with significance levels $P = 0.048$ using difference in means and $P = 0.025$ using difference in zero-deleted means. We conclude that score type 2 succeeds in controlling against extreme cycles and have confined our analyses to fits using this score.

For each score type 2 fit the minimum attained score was 4. Conceivably, this minimum represents the lowest attainable score with indegree bound $K = 2$, although with no available analytical theory, this can be verified only by an exhaustive model search. We also expect many solutions to achieve exactly this score.

Table 1 defines a method of summarizing attractors as vectors of single components. Each component of the summary vector is determined by observing the sequence of that component's outcomes in the attractor cycle. The attractor summary component is set to 0 if all outcomes in the sequence are 0, set to 1 if a 1 occurs at least once but -1 does not occur, set to -1 if a -1 occurs at least once but 1

Figure 6: Histograms of extreme cycle count grouped by score type. For score type 1, 70 replications using $N_s = 10,000, 25,000, 100,000$ are used, for score type 2, 30 replications using $N_s = 100,000$.

Table 1 defines a method of summarizing attractors as vectors of single components. Each component of the summary vector is determined by observing the sequence of that component's outcomes in the attractor cycle. The attractor summary component is set to 0 if all outcomes in the sequence are 0, set to 1 if a 1 occurs at least once but -1 does not occur, set to -1 if a -1 occurs at least once but 1 does not occur, and set to 9 if both 1 and -1 occur.

It is important to emphasize that analysis involving operator A differs substantially from that involving perturbation operators A_{E_k} . The attractors associated with A are not observed, but represent the behavior of the unperturbed cell, and would therefore be of greater interest in a general study of the behavior of the cell, forming the basis of an *in silico* model of the cells' GRN. In addition, the steady states resulting from any new perturbation experiment E' can be predicted computationally using $A_{E'}$.

We examined the predicted attractor summaries for operator A for a variety of initial states. We use initial states corresponding to experiments $E_1, \dots, E_9$, except that the initial state is not forced to be persistent. For example, corresponding to E_1 , a single forced persistent over-expression of *HoxC13*, we set initial state $x_0 = (1, 0, 0, 0, 0, 0, 0, 0, 0)$, then observe iterates $A^m(x_0)$, $m \geq 1$, until a complete attractor is observed. A variety of attractors exist among the replicates, but a clear trend emerges. There are three attractors which dominate. The first is the null attractor $(0, 0, 0, 0, 0, 0, 0, 0, 0)$, essentially representing indefinite gene expression at control levels. Another highly represented attractor is $(0, 0, -1, 0, 0, 1, -1, 0, 0)$. This is of particular interest, since it represents cyclical behavior or feedback in which high levels of *Sfrp2* alternate with low levels of *Id4* and *Wnt9a*. The third attractor is $(0, 0, 1, 0, 0, 1, -1, 0, 0)$, a variation of the previous one. There are 9 other attractors represented, however, 8 of these are represented in only 1 fit, and the remaining one in 2 fits. From a Bayesian point of view, the three frequently represented attractors appear to represent a model feature with high posterior probability, while the remaining attractors can be assigned low posterior probability. In addition, there appears to be some consistency across the fits in terms of the transient structure of A . Given initial states defined by over-expression of *HoxC13*, *Id4*, *Notch3*, or under-expression of *Plac8*, the attractor is the null attractor in most fits (25/30, 29/30, 30/30, 30/30, respectively). For initial states defined by over-expression of *Jag2*, *Sfrp2*, *Wnt9a*, *Zfp385*, the attractor is a non-null attractors in most fits (27/30, 30/30, 27/30, 27/30). For the initial state defined by over-expression of *Id2* there is more uncertainty, in that 16/30 fits predict the null attractor.

3.2.3 Consistency of Bayesian Posterior Densities

Under the methodology defined in Section 2.3 a SA algorithm is first used to determine τ^* , which defines the posterior density ϕ_{τ^*} used to estimate posterior probabilities. Some flexibility can be introduced by using posterior density $\phi_{r\tau^*}$ for some positive factor r . For our analysis the posterior density was sampled for the first 10 model fits using the reported model fit as the initial point of the sampler, using $r \in 1, 3$. Our primary concern will be with consistent convergence. While we do not expect the algorithm to converge consistently to a single model, we should expect consistent convergence to a single posterior density, to within some acceptable approximation.

We first summarize the graph structure. Using Bayesian averaging the posterior probability that j is a parent of i , $p(j \rightarrow i)$, was estimated for each posterior density. A *no parent* state was included. These probabilities appear to be consistent across model fits (Figures 7 and 8). Under consistent convergence, these values should be found close to the identity. If we compile all absolute differences

$|p(j \rightarrow i) - p'(j \rightarrow i)|$ where p and p' denote distinct posterior densities, for each pair, we find 97%, 98% and 2% no greater than 0.1, no greater than 0.25 and greater than 0.95, respectively, for $r = 1$. The corresponding percentages for $r = 3$ are 87%, 95% and 0%. The numbers, along with the plots themselves, suggest a straightforward consequence of varying r – lower values yield greater overall consistency but a greater number of large differences. This tendency was also observed by the authors for other model features. It is suggestive of the bias/variance trade-off common in many forms of complex statistical modeling. In our analyses, we use the value $r = 3$, as the more cautious choice.

We noted above that when examining multiple fits, there was considerable variation in the attractor structure, but that only 3 attractors appeared consistently in most fits. Interestingly, these rarer attractors have little influence on their associated posterior densities. For example, the attractor $(1, 0, -1, 1, 9, 9, -1, 1, 1)$ appears in one model fit, but has negligible posterior probability in the associated posterior density. This is generally true of all the rarer attractors. This is an important feature of the methodology, since it identifies which features of a given model are likely to be spurious, without the need for extensive model fitting replications.

Finally, we briefly raise a technical consideration. Recall that when calculating the score $D(A | Y)$ the operators A_{E_k} are iterated from a given initial state until an attractor is reached. This means that there may be table entries which are never used. For example, in the table for Zfp385 it is predicted that the simultaneous under-expression of Jag2 and Notch3 will induce the over-expression of Zfp385. However, that particular combination of outcome levels for Jag2 and Notch3 was not encountered during the scoring of A . Although assigning values to each table entry is needed to conduct a random search, and possibly to perform further Bayesian averaging, it is still appropriate to flag these entries, since they have no effect on the score or on any of the resulting attractors or pathway structure of the perturbed operators.

Figure 7: Scatterplots of $p(j \rightarrow i)$ for posterior density pairs $(1,2), (2,3), \dots, (9,10)$, using temperature τ^* . Identity and ± 0.1 boundaries are indicated.

Posterior density at 3 x convergence temperature

Figure 8: Scatterplots of $p(j \rightarrow i)$ for posterior density pairs $(1,2), (2,3), \dots, (9,10)$, using temperature $3\tau^*$. Identity and ± 0.1 boundaries are indicated.

4 Conclusion

We have proposed a method for the modeling of GRNs as Boolean networks using gene perturbation data. The method accounts for the dynamic structure of gene perturbation experiments and, through the use of computational Bayesian methodology, permits the direct evaluation of model uncertainty. This in turn permits more useful inference statements concerning GRN features without requiring resolution to a single model, and will permit the rational and efficient design of further experiments.

The model was applied to gene perturbation data associated with 9 CRGs (McMurray et al., 2008). The method was consistent in the sense that given randomized initial points, the algorithm always converged to an approximately common posterior density. An examination of the results yielded an intuitively plausible ensemble of networks, consistent with previously reported observations. The model was also tested using a number of synthetic networks, and proved to be accurate in the sense that model features reporting posterior probabilities close to 1 were consistently accurate. In addition, it was demonstrated that experiments which resolve model uncertainty could be identified by a careful examination of the posterior density.

While significant in their own right, these results demonstrate the potential of the proposed methodology to uncover novel gene regulatory networks and to better understand the behavior of known ones. As genomic research turns away from independent tests of differential gene expression and tackles the complexity of gene networks, methods such as the one proposed here provide a crucial tool in advancing our understanding. Work is currently underway to extend the proposed methodology to larger and more complex networks and to investigate other potential gene networks.

Appendix I

In this appendix we describe in detail the Markov Chain Monte Carlo (MCMC) methodology used to search for low scoring models. Some care is needed here because the model is defined both by the network topology and the transition functions.

Reductions and extensions of transition functions

Altering an operator in some way may or may not alter its minimal graph. We now introduce some notation and definitions which will help consider this question.

Assume $\mathcal{B}$ contains a level 0. For convenience, given $x \in \mathcal{B}^I$, for integer I , let $H^S(x)$ be the subvector of x defined by index set $S \subset \{1, \dots, I\}$. Similarly, let $H^{-S}(x)$ be the subvector of x defined by removing the components of x defined by index set S . For completeness we define a null dimensioned vector $\vec{0} \in \mathcal{B}^0$, then let $\vec{0} \in \mathcal{B}^I$ is the vector in which all components equal 0 when $I \geq 1$.

Let $\mathcal{F}_I$ be the set of mappings $f: \mathcal{B}^I \rightarrow \mathcal{B}$. Then $\mathcal{F}_0$ may be interpreted as the set of constants in $\mathcal{B}$. It will be convenient to characterize a minimal graph in terms of the transition functions, which we do through the following definition:

Definition 3 Suppose $f \in \mathcal{F}_I$ for $I \geq 1$. Suppose $S \subset \{1, \dots, I\}$, $|S| = d \geq 1$. Then $f' \in \mathcal{F}_{I-d}$ is a reduction (in S) of f if $f(x) = f'(H^{-S}(x))$ for all $x \in \mathcal{B}^I$. We say f is non-reducible if it has no reduction. Also, a reduction f' of f is the maximal reduction if it is non-reducible. Conversely, an element $f' \in \mathcal{F}_{I+|S|}$ is an extension (in the components S) of $f \in \mathcal{F}_I$ if $f'(x) = f(H^{-S}(x))$ for $x \in \mathcal{B}^{I+|S|}$.

Let $\mathcal{F}_I^{nr}$ be the class of all non-reducible mappings in $\mathcal{F}_I$. If G is the minimal graph of A , with parent sets S_i , then $f_i(H^{S_i}(x)) \in \mathcal{F}_{I_i}^{nr}$, where $I_i = |S_i|$, $x \in \mathcal{X}$.

Model space

Depending on outcome set $\mathcal{B}$, and any other assumptions, restrictions may be placed on model space $\mathcal{A}$. In the example described below we impose the following restrictions. Let $G(A)$ be the minimal graph of A , with parent set S_i for the i th node. We will assume there are no self edges in $G(A)$, so that parents sets satisfy $S_i \subset V - \{i\}$. Denote node indegree $I_i = |S_i|$. We enforce bound $I_i \leq K$ (necessarily, $K \leq n - 1$).

We take the steady state vector defined in Assumption (A3) to be $X^0 = \vec{0}$, and this in turn requires that the transition function $f_i = 0$ whenever all input arguments are 0, or when $I_i = 0$. Let $\mathcal{F}_I^0$ be all mappings in $\mathcal{F}_I$ for which $f(\vec{0}) = 0$. Then denote $\mathcal{F}_I^{nr,0} = \mathcal{F}_I^{nr} \cap \mathcal{F}_I^0$. Given a set of nodes V , an operator A is constructed by defining for each $i \in V$ a parent set S_i and an element $f_i \in \mathcal{F}_{I_i}^{nr,0}$. The operator maps x to x' by setting $H^{\{i\}}(x') = f_i(H^{S_i}(x))$ for $i = 1, \dots, n$.

We denote by $\mathcal{A}^*$ the subset of $\mathcal{A}$ which satisfies these constraints.

Search Algorithm

Since an exhaustive search over $\mathcal{A}^*$ will not be feasible, we will employ a Markov chain Monte Carlo (MCMC) search, involving the simulation of a Markov chain on

state space $\mathcal{A}^*$. We use score $D(A | Y)$, and additionally require a neighborhood $\mathcal{N}(A) \subset \mathcal{A}^*$ for each $A \in \mathcal{A}^*$. Given current state A' , a *proposal* state A'' is chosen randomly from $\mathcal{N}(A')$ which becomes the subsequent state with a probability depending on the resulting change in score, otherwise the Markov chain remains at A' . The efficiency of a MCMC usually depends greatly on the choice of $\mathcal{N}(A)$ and the proposal distribution in general (see, for example, Chib and Greenberg (1995)). Ideally, selection of the proposal state does not result in too large a change in score.

We will use the simulated annealing algorithm (SAA) described in Aarts and Korst (1989) (see also Almudevar (2003) for a brief summary of the algorithm). Consider distributions of the form:

$$\phi_\tau(A) \propto \exp(-D(A | Y)/\tau) \quad (2)$$

where τ is a temperature parameter. The algorithm is structured as a sequence of stages. In the m th stage a sample of models from steady state distribution $\phi_{\tau_m}(A)$ is simulated using the Metropolis-Hastings algorithm (Hastings, 1970).

Here, τ_m is a specific stage m temperature parameter. We let μ_m and σ_m be the steady state mean and standard deviation of $D(A | Y)$ at stage m . These can be estimated directly from the stage m sample. The temperature parameter is decreased after each sample according to the formula:

$$\tau_{m+1} = \frac{\tau_m}{1 + \frac{\tau_m \ln(1+\delta)}{3\sigma_m}}, \quad (3)$$

which ensures a constant change in the variational distance of the steady state distribution towards the minimum.

The algorithm as described in Almudevar (2003) relies on 5 parameters. The initial acceptance rate χ_0 of the MCMC sampler determines the initial stage 1 temperature τ_1 , and σ_0 is the steady state standard deviation corresponding to χ_0 . At $\chi_0 = 1$ the MCMC is equivalent to a random walk. The recommended practice is to set the value close to, but less than, 1. The change in variational distance is δ . The convergence tolerance is ε , and the algorithm terminates when μ_m/σ_0 decreases by less than ε for N_ε consecutive stages. Each stage is run for a fixed number of transitions N_s (this serves the function of parameter β in Almudevar (2003)). A null model with no edges is used as the initial model of the sampler. Random replicates are generated by varying a random number seed.

A number of modifications were made to the SA algorithm as described in Almudevar (2003), permitting wider searches before convergence occurs. First, σ_m in (3) is replaced by an exponential smoother $\sigma_m^\rho = \rho\sigma_m + (1-\rho)\sigma_{m-1}^\rho$, for parameter $\rho \in (0, 1]$ (the value $\rho = 1$ represents no smoothing). Examining equation (3) we can see that a value $\sigma_m = 0$ forces $\tau_{m+1} = 0$, which will terminate the search.

Using σ_m^p permits the search to continue if no variation in $D(A | Y)$ is observed in a single stage.

We also extend the neighborhood $\mathcal{N}(A)$ to the *product neighborhood*:

$$\mathcal{N}_j(A) = \bigcup_{A' \in \mathcal{N}_{j-1}(A)} \mathcal{N}(A'); j > 1,$$

where $\mathcal{N}_1(A) = \mathcal{N}(A)$, which permits a wider search within each stage. A mixture can then be defined, selecting A' from $\mathcal{N}_j(A)$ with probability p_j^n .

Finally, we permit an adaptive truncation of stage sample length N_s . For integer N_s^{tr} we end a stage when the number of occurrences of $D(A_{i+1} | Y) > D(A_i | Y)$ equals N_s^{tr} , if this occurs before N_s models are sampled. By setting $N_s^{tr} \ll N_s$, we can select N_s large enough to adequately sample at the later stages, while avoiding needlessly large samples in the earlier stages.

It should be noted that there are a number of alternative methods for large-scale combinatorial optimization problems, for example, TABU search or genetic algorithms (Aarts and J.K., 1997). However, our interest will also be in the estimation of posterior densities on the model space, for which the use of a MCMC sampler is a natural option. The use of the SA algorithm will allow the integration of the optimization and the posterior density sampling phases of the modeling process in a way not possible with other optimization methods.

Model Neighborhood

The neighborhood $\mathcal{N}(A)$ is generated by defining a class of small modifications of A . Let S_i be the parent sets of the minimal graph G of A . Assume transition function f_i is a function of $H^{S_i}(x)$. We first select randomly a node i . Modifications may be made to f_i or S_i using some combination of the following three operations:

- (O1) Add one parent to a parent set S_i .
- (O2) Randomly change function f_i .
- (O3) Determine if a maximal reduction exists for f_i . If so, adjust f_i and S_i accordingly.

Note that the parent set of node i can be reduced by operation (O3), but not elsewhere. This ensures that any such modification is minimal. Operation (O2) is represented here by the following algorithm, which accepts as input $f \in \mathcal{F}^I$.

Perturb(f):

1. Randomly select $y \in \mathcal{B}^I - \vec{0}$.
2. Set $f' = f$.
3. If $f(y) \neq 0$ set $f'(y) = 0$.
4. If $f(y) = 0$ set $f'(y)$ to a randomly chosen value from $\mathcal{B} - 0$.
5. Return f' .

We note that in $Perturb(f)$, transitions between outcomes 1 and -1 are avoided, in order to minimize the impact of the transition. Given model A the algorithm $Proposal(A)$, defined below, generates proposal state A' . Informally, a node is chosen randomly, then one of two sets of operations is chosen at random. In the first, a parent is added to the parent set, then the node function is first extended, then perturbed. In the second, the node function is perturbed, then reduced, if possible. The algorithm requires parameter K (indegree bound), and probabilities $\alpha_0, \dots, \alpha_K$. We require $\alpha_0 = 1, \alpha_K = 0$.

Proposal(A):

1. Select node $i \in V$ at random, retrieve f_i, S_i from model A . Go to step 2a with probability α_k , and go to step 3a with probability $1 - \alpha_k$, where $k = I_i$.
- 2a. Randomly select v from $V - \{i\} - S_i$.
- 2b. Set $S'_i = S_i \cup \{v\}$.
- 2c. Determine f^e , the extension of f_i in the j th component, where v is the j th component of S'_i .
- 2d. Set $f'_i = Perturb(f^e)$.
- 2e. Go to step 4.
- 3a. Set $f'_i = Perturb(f_i)$.
- 3b. Determine if a maximal reduction in $E \neq \emptyset$ exists for f'_i . If not, set $S'_i = S_i$, then go to step 4.
- 3c. Set $S'_i = S_i - E$, and replace f'_i with it's maximal reduction.
4. Return model A' , equivalent to model A , with i th node components f_i, S_i replaced by f'_i, S'_i .

Parameter settings for search algorithms for synthetic networks

In each case the parameters for the SA algorithm were set to $N_s = N_s^{tr} = 2,000$, with $\chi_0 = 0.9, \delta = 0.1, \varepsilon = 0.001, N_\varepsilon = 3$. The initial sample, used to determine the initial temperature, is of length $m_0 = 2000$. We use a mixture of neighborhoods

$\mathcal{N}(A)$, $\mathcal{N}^2(A)$, with $p_1^n = 0.01$. Finally, in the **Proposal(A)** algorithm, the probabilities of adding a parent are $\alpha_0 = 1$, $\alpha_1 = 0.1$, $\alpha_2 = 0$. The choice of parameters reflects the fact that for these cases the algorithm was able to converge quickly to Type 2 score of 0. The posterior sampling algorithm collected 5000 model with intervals of 2000 transitions.

Parameter settings for search algorithms for CRG network

We set a cap of $K = 2$ parents. To select N_s it is necessary to estimate the size of $\mathcal{N}(A)$, which depends on A . For an n gene model, to select from $\mathcal{N}(A)$ a gene v is first chosen. If v has no parents, then a selection from $n - 1$ parents is made. There are then 4 possible transition functions which may be chosen, so set $N_0 = (n - 1)4$. If v has one parent, and we select a new parent, we choose from $(n - 2)$ genes. We then perturb one of 8 elements from the 2 dimensional table, which involves up to 2 choices. If no additional parent is chosen, the transition function is perturbed in one of up to 3 ways. Set $N_1 = (n - 2)16 + 3$. If v has 2 parents, no further parent is added, and the transition function may be perturbed in up to 16 ways so set $N_2 = 16$. Then bounds are given by:

$$|\mathcal{N}(A)| \leq n_0 N_1 + n_1 N_1 + n_2 N_2 \leq \bar{N} = 16n^2 - 29n$$

where n_i is the number of genes with i parents, and the worst case upper bound $\bar{N}$ is obtained by setting $n_1 = n$ when $n > 2$. Additionally, we must have $|\mathcal{N}^j(A)| \leq \bar{N}^j$. For $n = 9$ this gives $\bar{N} = 1,035$, with $|\mathcal{N}^2(A)| \leq 1,071,225$. However, we note that models dominated by genes with 2 parents will be much more numerous, and that the increased complexity will tend to give such models lower values of $D(A | Y)$. It will therefore be reasonable to expect most transitions to take place in neighborhoods of approximate size $|\mathcal{N}(A)| \approx nN_2 = 144$ or $|\mathcal{N}^2(A)| \approx 20,736$. The value N_s should be of the same order of magnitude as the neighborhood size, so we use an intermediate value $N_s = 100,000$, with $N_s^{tr} = 5,000$.

For the remaining parameters we take $\chi_0 = 0.9$, $\delta = 0.01$, $\varepsilon = 0.001$, $N_\varepsilon = 25$. The initial sample, used to determine the initial temperature, is of length 2000. We use a mixture of neighborhoods $\mathcal{N}(A)$, $\mathcal{N}^2(A)$, with $p_1^n = 0.01$. Finally, in the **Proposal(A)** algorithm, the probabilities of adding a parent are $\alpha_0 = 1$, $\alpha_1 = 0.1$, $\alpha_2 = 0$. Although the algorithm converges to a specific model, the current minimum scoring model is tracked, and is output in place of the convergence model.

Appendix II

The data for the 9 CRG model is of the following form. A control cell is taken to be a normal parental cell after an oncogenic transformation. Persistent perturbations were performed by individually readjusting expression of each of the 9 CRGs to levels found prior to oncogenic transformation (amounting to controlled under-expression for Plac8, and over-expression for the remainder). Gene expression profiles were obtained using four biological replications, paired with four control replicates. For each response/experiment pair involving the 9 CRGs a two-sample Mann-Whitney test was performed, resulting in a p-value indicating the statistical significance of differential expression. This results in a 9 by 9 matrix B of p-values, indicating the degree to which expression levels of the row gene are altered due to the perturbation of the column gene. Each element is annotated by a -1 or 1 indicating under- or over-expression relative to control levels. Diagonal elements of B are set to 1 (or to -1 for Plac8).

This form of data introduces the problem of selecting a p-value threshold ρ . A gene graph is constructed by selecting all matrix elements no greater than ρ , then adding an edge directed from the column gene to the row gene for each selected element. The value ρ may be selected using a multiple testing procedure (MTP) (Dudoit and van der Laan, 2008), although an *ad hoc* value such as $\rho = 0.01$ is often used in the literature. We will use a method proposed in Almudevar (2009) in which the hierarchal sequence of graphs obtained by varying ρ is first constructed, then each graph is tested against a null model with equal numbers of edges using an information theoretic score which is sensitive to general graph features, such as hub or chain structure, resulting in a sequence of graph structure p-values. The resulting sequence is shown in Figure 9. As is reported in the literature, the edge distribution of actual cellular networks tends to follow a power law distribution (Wagner, 2002), hence adopting a Erdős-Renyi random graph model (uniformly distributed edges) as the null model provides a statistical basis for detecting network structure.

The methodology may be used to select a threshold ρ , and also to provide a global significance level for the existence of general network structure, independent of the threshold selection. In Almudevar (2009) the value ρ was estimated to be the minimum graph structure p-value recorded (in this example, the 16th edge in Figure 9. Although examples in Almudevar (2009) suggest that this minimum selection rule is reasonably accurate, no formal description of its properties is currently available. We note that in Figure 9 the graph structure p-values rapidly decrease early in the sequence, then after the 20th edge an apparent steady state is reached. We therefore set the threshold at 20 edges rather than 16, noting that subsequent modeling will provide further inference regarding network structure.

The global test for network structure proposed in Almudevar (2009) yields a significance level of $P = 0.029$. It is also interesting to note that the 20 edge threshold is $\rho = 0.177$ (that is, the p-value of rank 20 in B). It is unlikely that any standard MTP would select a threshold this high. The threshold defines a 20 edge connectivity graph, shown in Figure 3, as well as the 9 experimental steady states, shown in Table 10, which represents the input data for the fitting algorithm.

Figure 9: Plot shows significance of network structure against Erdős-Renyi random graph model for increasing graph sequence indexed by p-value threshold τ . Input data consists of 9×9 p-value matrix from the 9 CRG model.

References

- Aarts, E. and L. J.K. (1997): *Local Search in Combinatorial Optimization*, Princeton, NJ: Princeton University Press.
- Aarts, E. and J. Korst (1989): *Simulated Annealing and Boltzmann Machines*, John Wiley & sons.
- Aarts, E., J. Korst, and P. van Laarhoven (1997): *Local Search in Combinatorial Optimization* (E. Aarts, J.K. Lenstra eds), Princeton, NJ: Princeton University Press, chapter Simulated Annealing, 91–120.
- Almudevar, A. (2003): “A simulated annealing algorithm for maximum likelihood pedigree reconstruction,” *Theoretical Population Biology*, 63, 63–75.
- Almudevar, A. (2007): “A graphical approach to relatedness inference,” *Theoretical Population Biology*, 71, 213–229.
- Almudevar, A. (2009): “Selection of statistical thresholds in graphical models,” *EURASIP Journal on Bioinformatics and Systems Biology*, 2009, 13 pp.
- Almudevar, A. and P. Salzman (2005): “Using a Bayesian posterior density in the design of perturbation experiments for network reconstruction,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2, 223–229.
- Besag, J., P. Green, D. Higdon, and K. Mengersen (1995): “Bayesian computation and stochastic systems,” *Statist. Sci.*, 10, 3–41.
- Boros, E., T. Ibaraki, and K. Makino (1998): “Error-free and best-fit extensions of partially defined boolean functions,” *Inform Comput*, 140, 254–283.
- Chib, S. and E. Greenberg (1995): “Understanding the metropolis-hastings algorithm,” *The American Statistician*, 49, 327–335.
- Crosnier, C., D. Stamataki, and J. Lewis (2006): “Organizing cell renewal in the intestine: stem cells, signals and combinatorial control,” *Nature Reviews Genetics*, 7, 349–359.
- Danks, D., C. Glymour, and P. Spirtes (2003): “The computational and experimental complexity of gene perturbations for regulatory network search,” in W. Hsu, R. Joehanes, and C. Page, eds., *Proceedings of IJCAI-2003 Workshop on Learning Graphical Models for Computational Genomics.*, 22–31.
- Dudoit, S. and M. J. van der Laan (2008): *Multiple Testing Procedures with Applications to Genomics*, New York: Springer.
- Hanahan, D. and R. A. Weinberg (2000): “The hallmarks of cancer,” *Cell*, 100, 57–70.
- Hastings, W. K. (1970): “Monte Carlo sampling methods using Markov chains and their applications,” *Biometrika*, 57, 97–109, URL <http://biomet.oxfordjournals.org/cgi/content/abstract/57/1/97>.

- Huang, S. and D. E. Ingber (2000): "Shape-dependent control of cell growth, differentiation, and apoptosis: switching between attractors in cell regulatory networks," *Exp Cell Res*, 261, 91–103.
- Ideker, T. E., V. Thorsson, and R. M. Karp (2000): "Discovery of regulatory interactions through perturbation: Inference and experimental design," *Pacific Symposium on Biocomputing*, 5, 302–313.
- Ideker, T. E., V. Thorsson, J. A. Ranish, R. Christmas, J. Buhler, J. K. Eng, R. Bumgarner, D. R. Goodlett, R. Aebersold, and L. Hood (2001): "Integrated genomic and proteomic analyses of a systematically perturbed metabolic network," *Science*, 292, 929–934.
- Kauffman, S. (1969): "Metabolic stability and epigenesis in randomly constructed genetic nets," *J Theor Biol.*, 22, 437–467.
- Kim, S., E. R. Dougherty, M. L. Bittner, Y. Chen, K. Sivakumar, P. Meltzer, and J. M. Trent (2000): "General nonlinear framework for the analysis of gene interaction via multivariate expression arrays," *Journal of Biomedical Optics*, 5, 411–424.
- Koller, D. and N. Friedman (2009): *Probabilistic Graphical Models: Principles and Techniques.*, Cambridge, MA: MIT Press.
- Lahdesmaki, H., I. Shmulevich, and O. Yli-Harja (2003): "On learning gene regulatory networks under the boolean network model," *Machine Learning*, 52, 147–167.
- Lasorella, A., T. Uo, and A. Iavarone (2001): "Id proteins at the cross-road of development and cancer," *Oncogene*, 20, 8326–8333.
- Lloyd, A. C., F. Obermuller, S. Staddon, C. F. Barth, M. McMahon, and H. Land (1997): "Cooperating oncogenes converge to regulate cyclin/cdk complexes," *Genes Dev*, 11, 663–677.
- McMurray, H., E. Sampson, G. Compitello, C. Kinsey, L. Newman, B. Smith, S. Chen, L. Klebanov, P. Salzman, A. Yakovlev, and L. H. (2008): "Synergistic response to oncogenic mutations defines gene class critical to cancer phenotype," *Nature*, 453, 1112–1116.
- Mircean, C., I. Tabus, J. Astola, T. Kobayashi, H. Shiku, M. Yamaguchi, I. Shmulevich, and W. Zhang (2003): *Quantization and similarity measure selection for discrimination of lymphoma subtypes under k-nearest neighbor classification*, Citeseer.
- Pal, R., I. Ivanov, A. Datta, and E. R. Dougherty (2005): "Generating boolean networks with a prescribed attractor structure," *Bioinformatics*, 21, 4021–4025.
- Sachs, K., O. Perez, D. Pe'er, D. A. Lauffenburger, and G. P. Nolan (2005): "Causal protein-signaling networks derived from multiparameter single-cell data," *Science*, 308, 523–529.

- Sewing, A., B. Wiseman, A. C. Lloyd, and H. Land (1997): “High-intensity raf signal causes cell cycle arrest mediated by p21cip1.” *Mol Cell Biol*, 17, 5588–5597.
- Shmulevich, I. and E. R. Dougherty (2007): *Genomic Signal Processing*, Princeton, NJ: Princeton University Press.
- Shmulevich, I. and W. Zhang (2002): “Binary analysis and optimization-based normalization of gene expression data.” *Bioinformatics*, 18, 555–565.
- Suzuki, H., D. Watkins, K. Jair, K. Schuebel, S. Markowitz, W. Chen, T. Pretlow, B. Yang, Y. Akiyama, M. Van Engeland, M. Toyota, T. Tokino, Y. Hinoda, K. Imai, J. Herman, and S. Baylin (2004): “Epigenetic inactivation of sfrp genes allows constitutive wnt signaling in colorectal cancer.” *Nat Genet.*, 36, 417–422.
- Wagner, A. (2001): “How to construct a large genetic network from n perturbations in fewer than n^2 easy steps.” *Bioinformatics*, 17, 1183–1197.
- Wagner, A. (2002): “Estimating coarse gene network structure from large-scale gene perturbation data,” *Genome Research*, 12, 309–315.
- Wagner, A. (2004): “Reconstructing pathways in large genetic networks from genetic perturbations.” *Journal of Computational Biology*, 11, 53–60.
- Wimberly, F., D. Danks, C. Glymour, and T. Chu (2009): *Computational methodologies in gene regulatory networks*, Hershey, PA: IGI Global Publishing.
- Xia, M. and H. Land (2007): “Tumor suppressor p53 restricts ras stimulation of rhoa and cancer cell motility.” *Nat Struct Mol Biol*, 14, 215–223.
- Xiang, Y., G. Lin, Q. Zhang, Y. Tan, and G. Lu (2008): “Knocking down wnt9a mrna levels increases cellular proliferation.” *Mol Biol Rep.*, 35, 73–79.
- Zhou, X., X. Wang, and E. R. Dougherty (2003): “Binarization of microarray data on the basis of a mixture model,” *Molecular Cancer Therapeutics*, 2, 679–684.
- Zhou, X., X. Wang, R. Pal, I. Ivanov, M. Bittner, and E. R. Dougherty (2004): “A Bayesian connectivity-based approach to constructing probabilistic gene regulatory networks,” *Bioinformatics*, 20, 2918–2927, URL <http://bioinformatics.oxfordjournals.org/content/20/17/2918.abstract>.