

Validation of Inference Procedures for Gene Regulatory Networks

Edward R. Dougherty*

Department of Electrical and Computer Engineering, Texas A&M University; Computational Biology Division, Translational Genomics Research Institute; Department of Pathology, University of Texas M.D. Anderson Cancer Center, USA

Abstract: The availability of high-throughput genomic data has motivated the development of numerous algorithms to infer gene regulatory networks. The validity of an inference procedure must be evaluated relative to its ability to infer a model network close to the ground-truth network from which the data have been generated. The input to an inference algorithm is a sample set of data and its output is a network. Since input, output, and algorithm are mathematical structures, the validity of an inference algorithm is a mathematical issue. This paper formulates validation in terms of a semi-metric distance between two networks, or the distance between two structures of the same kind deduced from the networks, such as their steady-state distributions or regulatory graphs. The paper sets up the validation framework, provides examples of distance functions, and applies them to some discrete Markov network models. It also considers approximate validation methods based on data for which the generating network is not known, the kind of situation one faces when using real data.

Received on: September 10, 2007 - Revised on: September 20, 2007 - Accepted on: September 29, 2007

Key Words: Epistemology, gene network, inference, validation.

1. INTRODUCTION

The construction of gene regulatory networks is among the most important problems in systems biology [1-2]. Network models provide quantitative knowledge concerning gene regulation and, from a translational perspective, they provide a basis for mathematical analyses leading to systems based therapeutic strategies [3]. Network models run the gamut from coarse-grained discrete networks to the detailed description of stochastic differential equations. The availability of high-throughput genomic data has motivated the development of numerous inference algorithms. The performance, or validity, of these algorithms must be quantified. An inference algorithm takes a sample set of data as input and outputs a model network. Its validity must be evaluated relative to its ability to infer a model network close to the ground-truth network from which the data have been generated. Given a hypothetical model, generate data from the model, apply the inference procedure to construct an inferred model, and compare the hypothetical and inferred models *via* some objective function.

This paper mathematically formulates validation in terms of the distance between two networks, or the distance between two structures of the same kind deduced from the networks, such as their steady-state distributions. As a function from a sample set to a class of network models, an inference procedure is a mathematical operator and its performance must be evaluated within a mathematical framework, in this case, distance functions. The paper sets up the validation framework in general terms, provides examples of

distance functions, and applies them to some basic network models. It also considers approximate validation methods based on data for which the generating network is not known, the situation one faces when using real data.

It is hoped that this paper will help to motivate the study of network validation procedures. While we believe it describes the general setting and the basic requirements for validation, as will be pointed out, to date, there has been very little study devoted to network validation. There are many subtle statistical issues. If we are to be able to judge the worth of proposed algorithms, then these issues need to be addressed within a formal mathematical framework.

2. BACKGROUND: NETWORK MODELS

Although our aim is to consider network inference from a fairly general perspective, to give concrete examples we require some specific models. Thus, we assume the underlying network structure is composed of a finite node (gene) set, $V = \{X_1, X_2, \dots, X_n\}$, with each node taking discrete values in $[0, d - 1]$. The corresponding state space possesses $N = d^n$ states, which we denote by $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$. We express the state \mathbf{x}_j in vector form by $\mathbf{x}_j = (x_{j1}, x_{j2}, \dots, x_{jn})$. For notational convenience we write vectors in row form but treat them as columns when multiplied by a matrix. The corresponding dynamical system is based on discrete time, $t = 0, 1, 2, \dots$, with the state-vector transition $\mathbf{X}(t) \rightarrow \mathbf{X}(t + 1)$ at each time instant. The state $\mathbf{X} = (X_1, X_2, \dots, X_n)$ is often referred to as a gene activity profile (GAP).

Markov Chains

We assume that the process $\mathbf{X}(t)$ is a *Markov chain*, meaning that the probability of $\mathbf{X}(t)$ conditioned on \mathbf{X} at $t_1 <$

*Address correspondence to this author at the Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843-3128, USA; E-mail: edward@ece.tamu.edu

$t_2 < \dots < t_s < t$ is equal to the probability of $\mathbf{X}(t)$ conditioned on $\mathbf{X}(t_s)$. We also assume the process is *homogeneous*, meaning that the transition probabilities depend only on the time difference, that is, for any t and u , the u -step transition probability,

$$p_{jk}(u) = P(\mathbf{X}(t+u) = \mathbf{x}_k | \mathbf{X}(t) = \mathbf{x}_j) \quad (1)$$

depends only on u . We are not asserting that the Markov property and homogeneity are necessary assumptions for gene regulatory networks. We make these assumptions to facilitate mathematically tractable modeling for the current study. Under these assumptions, we need only consider the one-step transition probability matrix,

$$\mathbf{P} = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1N} \\ p_{21} & p_{22} & \cdots & p_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ p_{N1} & p_{N2} & \cdots & p_{NN} \end{pmatrix} \quad (2)$$

where the one-step transition probability, p_{jk} , is given by $p_{jk} = p_{jk}(1)$. We refer to \mathbf{P} simply as the *transition probability matrix*. For $t = 0, 1, 2, \dots$, the state probability structure of the network is given by the t -state probability vector

$$\mathbf{p}(t) = (p_1(t), p_2(t), \dots, p_N(t)) \quad (3)$$

where $p_j(t) = P(\mathbf{X}(t) = \mathbf{x}_j)$. $\mathbf{p}(0)$ is the initial-state probability vector.

Besides the state one-step probabilities, we can consider the gene one-step probabilities,

$$p_i(j, r) = P(X_i(t+1) = r | \mathbf{X}(t) = \mathbf{x}_j) \quad (4)$$

If, given the GAP at t , the conditional probabilities of the genes are independent, then

$$p_{jk} = p_1(x_{j1}, \mathbf{x}_k) p_2(x_{j2}, \mathbf{x}_k) \dots p_n(x_{jn}, \mathbf{x}_k) \quad (5)$$

Suppose that gene X_i at time $t+1$ depends only on values of genes (predictors) in a regulatory set, $R_i \subset V$, at time t , the dependency being independent of t . Then the gene one-step probabilities are given by

$$p_i(j, r) = P(X_i(t+1) = r | X_i(t) = x_{ij} \text{ for } X_i \in R_i) \quad (6)$$

In this form, we see that the Markov dependencies are restricted to regulatory genes.

The network has a *regulatory graph* consisting of the n genes and a directed edge from gene x_i to gene x_j if $x_i \in R_j$. There is also a *state-transition graph* whose nodes are the N state vectors. There is a directed edge from state \mathbf{x}_j to state \mathbf{x}_k if and only if $\mathbf{x}_j = \mathbf{X}(t)$ implies $\mathbf{x}_k = \mathbf{X}(t+1)$.

A homogeneous, discrete-time Markov chain with state space $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ possesses a steady-state distribution $(\pi_1, \pi_2, \dots, \pi_N)$ if, for all pairs of states \mathbf{x}_k and \mathbf{x}_j , $p_{jk}(u) \rightarrow \pi_k$ as $u \rightarrow \infty$. If there exists a steady-state distribution, then, regardless of the state \mathbf{x}_k , the probability of the Markov chain being in state \mathbf{x}_k in the long run is π_k . In particular, for any initial distribution $\mathbf{p}(0)$, $p_k(t) \rightarrow \pi_k$ as $t \rightarrow \infty$. Not all Markov chains possess steady-state distributions.

Rule-Based Networks

A basic type of regulatory model occurs when the transition $\mathbf{X}(t) \rightarrow \mathbf{X}(t+1)$ is governed by a rule-based structure, meaning there exists a state function $\mathbf{f} = (f_1, f_2, \dots, f_n)$ such that $X_i(t+1) = f_i(R_i(t))$. A classical example of a rule-based network is a Boolean network (BN), where the values are binary, 0 or 1, and the function f_i can be defined via a logic expression or a truth table consisting of 2^n rows, with each row assigning a 0 or 1 as the value for the GAP defined by the row [4-5]. As defined, the BN is deterministic and the entries in its transition probability matrix are either 0 or 1. The *connectivity* of the BN is the maximum number of predictors for a gene. If each has the same number of predictors, then we say that the network has uniform connectivity.

The model becomes stochastic if the BN is subject to perturbation, meaning that at any time point, instead of necessarily being governed by the state function $\mathbf{f} = (f_1, f_2, \dots, f_n)$, there is a positive probability $p < 1$ that the GAP may randomly switch to another GAP. There are more refined ways of characterizing perturbations, such as defining perturbations at the gene level rather than the state level, but state-level perturbation is easy to describe and is sufficient for our purposes here. For a BN with perturbation, the corresponding Markov chain possesses a steady-state distribution.

The long-run behavior of a deterministic BN depends on the initial state and the network will eventually settled down and cycle endlessly through a set of states called an *attractor cycle*. The set of all initial states that reach a particular attractor cycle forms the *basin of attraction* for the cycle. Attractor cycles are disjoint. With perturbation, in the long run the network may randomly escape an attractor cycle, be re-initialized, and then begin its transition process anew.

3. QUANTIFYING THE DIFFERENCE BETWEEN NETWORKS

Distance Functions

To discuss validity, we must first discuss the manner in which we are to compare two networks. Given networks \mathcal{H} and \mathcal{M} , we need a function, $\mu(\mathcal{M}, \mathcal{H})$, quantifying the difference between them. We require that μ be a *semi-metric*, meaning that it satisfies the following four properties:

- (1) $\mu(\mathcal{M}, \mathcal{H}) \geq 0$,
- (2) $\mu(\mathcal{M}, \mathcal{M}) = 0$,
- (3) $\mu(\mathcal{M}, \mathcal{H}) = \mu(\mathcal{H}, \mathcal{M})$ [symmetry],
- (4) $\mu(\mathcal{M}, \mathcal{H}) \leq \mu(\mathcal{M}, \mathcal{N}) + \mu(\mathcal{N}, \mathcal{H})$ [triangle inequality].

As a semi-metric, μ is called a *distance function*. If μ should satisfy a fifth condition,

- (5) $\mu(\mathcal{M}, \mathcal{H}) = 0 \Rightarrow \mathcal{M} = \mathcal{H}$,

then it is a *metric*. A distance function is often defined in terms of some characteristic, by which we mean some structure associated with a network, such as its regulatory graph, steady-state distribution, or probability transition matrix.

This is why we do not require the fifth condition, $\mu(\mathcal{M}, \mathcal{H}) = 0 \Rightarrow \mathcal{M} = \mathcal{H}$, for a network distance function.

If we want to approximate one network by another, say for reasons of computational complexity, then a distance function can be used to measure the goodness of the approximation. If \mathcal{M}_1 and \mathcal{M}_2 are two approximations of network \mathcal{H} , then \mathcal{M}_1 is a better approximation than \mathcal{M}_2 relative to μ if $\mu(\mathcal{M}_1, \mathcal{H}) < \mu(\mathcal{M}_2, \mathcal{H})$.

Because a network distance function need only be a semi-metric, one must be careful in applying propositions from the theory of metric spaces. For instance, in a metric space, if a sequence of points in the space is convergent, then the limit of the sequence is unique. When the points are networks, this is not necessarily true. A sequence of networks can converge to two distinct networks: $\{\mathcal{H}_i\}$ can converge to both \mathcal{M} and \mathcal{N} , with $\mathcal{M} \neq \mathcal{N}$.

Rule-Based Distance

For Boolean networks (with or without perturbation) possessing the same gene set, a distance is given by the proportion of incorrect rows in the function-defining truth tables. Denoting the state functions for networks \mathcal{H} and \mathcal{M} by $\mathbf{f} = (f_1, f_2, \dots, f_n)$ and $\mathbf{g} = (g_1, g_2, \dots, g_n)$, respectively, since there are n truth tables consisting of 2^n rows each, this distance is given by

$$\mu_{fun}(\mathcal{M}, \mathcal{H}) = \frac{1}{n2^n} \sum_{i=1}^n \sum_{k=1}^N I[g_i(\mathbf{x}_k) \neq f_i(\mathbf{x}_k)] \quad (7)$$

where I denotes the indicator function, $I[A] = 1$ if A is a true statement and $I[A] = 0$ otherwise [6]. If we wish to give more weight to those states more likely to be observed in the steady state, then we can weight the inner sums in Eq. 7 by the corresponding terms in the steady-state distribution, $\boldsymbol{\pi} = (\pi_1, \pi_2, \dots, \pi_N)$. For Boolean networks without perturbation, μ_{fun} is a metric. If there is perturbation, then μ_{fun} is not a metric because two distinct networks may be identical with regard to the rules but possess different perturbation probabilities.

Topology-Based Distance

If one's focus is on the topology of a network, then a straightforward approach is to construct the adjacency matrix. Given an n -gene network, for $i, j = 1, 2, \dots, n$, the (i, j) entry in the matrix is 1 if there is a directed edge from the i th to the j th gene; otherwise, the (i, j) entry is 0. If $\mathbf{A} = (a_{ij})$ and $\mathbf{B} = (b_{ij})$ are the adjacency matrices for networks \mathcal{H} and \mathcal{M} , respectively, where \mathcal{H} and \mathcal{M} possess the same gene set, then the *hamming* distance between the networks is defined by

$$\mu_{ham}(\mathcal{M}, \mathcal{H}) = \sum_{i,j=1}^n |a_{ij} - b_{ij}| \quad (8)$$

Alternatively, the hamming distance may be computed by normalizing the sum, such as by the number of genes or the number of edges in one of the networks, for instance, when one of the networks is considered as representing

ground truth. The hamming distance is a coarse measure since it contains no steady-state or dynamic information. Two networks can be very different and yet have $\mu_{ham}(\mathcal{M}, \mathcal{H}) = 0$.

If one of the networks in Eq. 8 is considered as ground truth, then the hamming distance can be reformulated in terms of the numbers of false-negative and false-positive edges. If \mathcal{H} is the ground-truth network, then a false-negative edge is a directed edge not in \mathcal{M} that is in \mathcal{H} and a false-positive edge is directed edge in \mathcal{M} that is not in \mathcal{H} . Letting FN and FP be the numbers of false-negative and false-positive edges, respectively, the hamming distance is given by $FN + FP$. Because we are considering directed graphs, an incorrectly oriented edge in \mathcal{M} between two genes is both a false-negative and false-positive edge, although one can slightly alter the definitions to avoid this kind of double counting. If we were to consider undirected graphs, then this anomaly would not occur because an edge would either be present or absent. In this case, the hamming distance is still defined by Eq. 8 but the adjacency matrix is symmetric.

Since our interest is measuring the closeness of an inferred network to the network generating the data, we concentrate on distance functions, in particular, the hamming distance, which has been used for this purpose [7, 8]. Non-distance measures related to the hamming distance have been used in the context of regulatory graphs. Again let \mathcal{H} denote the ground-truth network. A true-positive edge is a directed edge in both \mathcal{H} and \mathcal{M} , and a true-negative edge is a directed edge in neither \mathcal{H} nor \mathcal{M} (with analogous definitions holding for undirected graphs). Let TP and TN be the numbers of true-positive and true-negative edges, respectively. The *positive predictive value* is defined by $TP/(TP + FP)$, the *sensitivity* is defined by $TP/(TP + FN)$, and the *specificity* is defined by $TN/(TN + FP)$. These kinds of measures have been used in several regulatory-graph inference papers [8-12] and a study using these measures has been performed to evaluate a number of inference procedures [13].

Transition-Probability-Based Distance

Distances for Markov networks can be defined *via* their probability transition matrices by considering matrix norms. A *norm* is a function $\|\bullet\|$ on a linear (vector) space, \mathcal{L} , such that:

- (1) $\|\mathbf{v}\| \geq 0$,
- (2) $\|\mathbf{v}\| = 0 \Rightarrow \mathbf{v} = \mathbf{0}$,
- (3) $\|a\mathbf{v}\| = |a| \cdot \|\mathbf{v}\|$ [*homogeneity*],
- (4) $\|\mathbf{v} + \mathbf{w}\| \leq \|\mathbf{v}\| + \|\mathbf{w}\|$ [*triangle inequality*].

Given a norm on \mathcal{L} , a metric is defined on \mathcal{L} by $\|\mathbf{v} - \mathbf{w}\|$.

For an $n \times n$ matrix and $r \geq 1$, the r -norm is defined by

$$\|\mathbf{P}\|_r = \left(\sum_{i,j=1}^n |p_{ij}|^r \right)^{1/r} \quad (9)$$

The *supremum* norm is defined by

$$\|\mathbf{P}\|_\infty = \max \{ |p_{ij}| : i, j = 1, 2, \dots, n \} \quad (10)$$

These norms are well-studied in linear algebra. Each yields a metric defined by $\|\mathbf{P} - \mathbf{Q}\|_r$. If $\mathbf{P} = (p_{ij})$ and $\mathbf{Q} = (q_{ij})$ are the probability transition matrices for networks \mathcal{H} and \mathcal{M} , respectively, then a network distance function is defined by

$$\mu_{prob}^r(\mathcal{M}, \mathcal{H}) = \|\mathbf{P} - \mathbf{Q}\|_r \quad (11)$$

Whereas $\|\bullet\|_r$ defines a matrix metric, μ_{prob}^r is only a network semi-metric because two distinct networks may have the same transition probability matrix.

Long-Run Distance

Since steady-state behavior is of particular interest, for instance, being associated with phenotypes, a natural choice for a network distance is to measure the difference between steady-state distributions [14]. If $\boldsymbol{\pi} = (\pi_1, \pi_2, \dots, \pi_N)$ is a probability vector, then its r -norm is defined by

$$\|\boldsymbol{\pi}\|_r = \left(\sum_{i=1}^N |\pi_i|^r \right)^{1/r} \quad (12)$$

for $r \geq 1$, and its supremum norm is defined by

$$\|\boldsymbol{\pi}\|_\infty = \max \{ |\pi_i| : i = 1, 2, \dots, N \} \quad (13)$$

If $\boldsymbol{\pi} = (\pi_1, \pi_2, \dots, \pi_N)$ and $\boldsymbol{\omega} = (\omega_1, \omega_2, \dots, \omega_N)$ are the steady-state distributions for networks \mathcal{H} and \mathcal{M} , respectively, then a network distance is defined by

$$\mu_{stead}^r(\mathcal{M}, \mathcal{H}) = \|\boldsymbol{\pi} - \boldsymbol{\omega}\|_r \quad (14)$$

Other norms can be used to define the distance function.

Not all networks possess steady-state distributions. The long-run behavior of a deterministic rule-based network, such as a Boolean network, depends on the initial state. A rule-based finite-value network possesses attractor cycles that characterize its long-run behavior and we can consider comparing this long-run behavior. This can be done by considering the proportion of time spent in a state once an attractor cycle has been entered. For any initial state \mathbf{x}_k , the network eventually enters the attractor cycle, C_k , whose basin contains \mathbf{x}_k . An arbitrary state \mathbf{x}_j either lies in C_k or it does not. Let m_k denote the number of states in C_k and p_k be the probability that the initial state is \mathbf{x}_k . We define the long-run probability of \mathbf{x}_j by

$$\zeta_j = \sum_{k=1}^N \frac{p_k}{m_k} I[\mathbf{x}_j \in C_k] \quad (15)$$

Letting $\boldsymbol{\zeta} = (\zeta_1, \zeta_2, \dots, \zeta_N)$, we can proceed analogously to the steady-state case by replacing $\boldsymbol{\pi}$ by $\boldsymbol{\zeta}$ to define the r -norm, and then define the distance function $\mu_{long}^r(\mathcal{M}, \mathcal{H})$ in the usual way.

Suppose all attractor cycles are singletons, so that $m_k = 1$. Moreover, suppose we do not know the initial-state prob-

abilities and we set $p_k = 1/N$. If \mathbf{x}_k is an attractor, let b_k denote the number of states in its basin; if \mathbf{x}_k is not an attractor, let $b_k = 0$. Then Eq. 15 reduces to $\zeta_j = b_j/N$. To this point, $\boldsymbol{\zeta} = (\zeta_1, \zeta_2, \dots, \zeta_N)$ describes a probability density because its components sum to 1. Now suppose we ignore the basin sizes so that $\zeta_j = 1/N$ if \mathbf{x}_j is an attractor and $\zeta_j = 0$ otherwise. If $\boldsymbol{\zeta} = (\zeta_1, \zeta_2, \dots, \zeta_N)$ and $\boldsymbol{\xi} = (\xi_1, \xi_2, \dots, \xi_N)$ correspond to networks \mathcal{H} and \mathcal{M} , respectively, then the network distance induced by the 1-norm is given by

$$\mu_{att}(\mathcal{M}, \mathcal{H}) = \sum_{j=1}^N |\zeta_j - \xi_j| = \frac{1}{N} |A_{\mathcal{H}} \Delta A_{\mathcal{M}}| \quad (16)$$

where $A_{\mathcal{H}}$ and $A_{\mathcal{M}}$ are the attractor sets for \mathcal{H} and \mathcal{M} , respectively,

$$A_{\mathcal{H}} \Delta A_{\mathcal{M}} = (A_{\mathcal{H}} - A_{\mathcal{M}}) \cup (A_{\mathcal{M}} - A_{\mathcal{H}}) \quad (17)$$

is the symmetric difference of $A_{\mathcal{H}}$ and $A_{\mathcal{M}}$, and $|\bullet|$ denotes the number of elements in a set. The distance $\mu_{att}(\mathcal{M}, \mathcal{H})$ compares the attractor sets of the two networks. $\mu_{att}(\mathcal{M}, \mathcal{H}) = 0$ if and only the attractor sets are the same. We have derived $\mu_{att}(\mathcal{M}, \mathcal{H})$ from $\mu_{long}(\mathcal{M}, \mathcal{H})$ assuming only singleton attractors, but $\mu_{att}(\mathcal{M}, \mathcal{H})$ can be applied to any rule-based discrete network.

Trajectory-Based Distance

Continuing with rule-based finite-value networks, rather than simply focusing on the long-run probabilities, one can take a more refined perspective by considering differences in the trajectories. Continue to let m_k denote the number of states in the cycle C_k for initial state \mathbf{x}_k and let t_k be the time it takes \mathbf{x}_k to reach C_k . The time trajectory of the network is given by $\mathbf{X}(t) = (X_1(t), X_2(t), \dots, X_n(t))$. For a given initial state this trajectory is deterministic. For initial state \mathbf{x}_k , denote the trajectory by

$$\mathbf{x}^{(k)}(t) = (x_1^{(k)}(t), x_2^{(k)}(t), \dots, x_n^{(k)}(t)) \quad (18)$$

Given the initial state is \mathbf{x}_k , we define the *amplitude cumulative distribution* of gene X_i by

$$F_i(z|k) = \frac{1}{m_k} \sum_{t=t_k}^{t_k+m_k-1} I[x_i^{(k)}(t) \leq z] \quad (19)$$

This increasing function of z counts the fraction of time that $x_i^{(k)}(t) \leq z$ in the cycle C_k .

Given two attractor cycles, C_k and C_j , resulting from initializations \mathbf{x}_k and \mathbf{x}_j , respectively, we define a distance between the cycles relative to gene X_i using the amplitude cumulative distributions, $F_i(\bullet|k)$ and $F_i(\bullet|j)$, by

$$\delta_i(C_k, C_j) = \|F_i(\bullet|k) - F_i(\bullet|j)\| \quad (20)$$

for some function norm $\|\bullet\|$. For example, we could use the L_1 norm

$$\|F_i(\bullet|k) - F_i(\bullet|j)\|_1 = \int_0^\infty |F_i(z|k) - F_i(z|j)| dz \quad (21)$$

The L_1 norm possesses an interesting interpretation if gene X_i has constant amplitude values, a and b , on cycles C_k and C_j , respectively. In this case, $F_i(\bullet|k)$ and $F_i(\bullet|j)$ are unit step functions with steps at a and b , respectively. Hence, in this case the L_1 norm reduces to

$$\delta_i(C_k, C_j) = |a - b| \quad (22)$$

and gives the distance, in amplitude, between the values of gene X_i on the two cycles. For a Boolean network, $\delta_i(C_k, C_j) = 0$ if the gene is either ON or OFF on both cycles and $\delta_i(C_k, C_j) = 1$ if X_i is ON for one cycle and OFF for the other (assuming X_i is constant on both cycles).

Considering the full set of genes, we define a distance between two attractor cycles, C_k and C_j by

$$\delta(C_k, C_j) = \frac{1}{n} \sum_{i=1}^n \delta_i(C_k, C_j) \quad (23)$$

Now consider two networks, \mathcal{M} and \mathcal{H} , having the same genes. We define the distance between \mathcal{M} and \mathcal{H} as the expected distance between attractor cycles over all possible initial states:

$$\mu_{amp}(\mathcal{M}, \mathcal{H}) = E[\delta(C_k^{\mathcal{M}}, C_k^{\mathcal{H}})] = \sum_{k=1}^N p_k \delta(C_k^{\mathcal{M}}, C_k^{\mathcal{H}}) \quad (24)$$

where $C_k^{\mathcal{M}}$ and $C_k^{\mathcal{H}}$ are the attractor cycles corresponding to initialization by state \mathbf{x}_k in networks \mathcal{M} and \mathcal{H} , respectively [15].

Equivalence Classes of Networks

The previous examples of network distance functions demonstrate a common scenario: a network semi-metric is defined by a metric on some network characteristic, for instance, its regulatory graph, its transition probability matrix, etc. The metric requirement, $\mu(\mathcal{M}, \mathcal{H}) = 0 \Rightarrow \mathcal{M} = \mathcal{H}$, fails because distinct networks possess the same characteristic. To formalize the situation, let $\lambda_{\mathcal{M}}$ and $\lambda_{\mathcal{H}}$ denote the characteristic λ corresponding to networks \mathcal{M} and \mathcal{H} , respectively. If v is a metric on a space of characteristics (directed graphs, matrices, probability densities, etc.), then a semi-metric μ_v is induced on the network space according to

$$\mu_v(\mathcal{M}, \mathcal{H}) = v(\lambda_{\mathcal{M}}, \lambda_{\mathcal{H}}) \quad (25)$$

This is quite natural if our main interest is with the characteristic, not the specific network itself.

Focus on network characteristics leads to the identification of networks possessing the same characteristic. Given any set, U , a relation \sim between elements of U is called an *equivalence relation* if it satisfies the following three properties for $a, b, c \in U$:

- (1) $a \sim a$ [reflexivity],
- (2) $a \sim b \Rightarrow b \sim a$ [symmetry],
- (3) $a \sim b$ and $b \sim c \Rightarrow a \sim c$ [transitivity].

If $a \sim b$, then a and b are said to be *equivalent*. An equivalence relation on U induces a partition of U . The subsets forming the partition are defined according to a and b lie in the same subset if and only if $a \sim b$. The subsets are called *equivalence classes*. The equivalence class of elements equivalent to a is denoted by $[a]^-$. According to the definitions, $[a]^- = [b]^-$ if and only if $a \sim b$.

If v is a semi-metric on a set U and we define $a \sim b$ if and only if $v(a, b) = 0$, then

$$\mu([a]^-, [b]^-) = v(a, b) \quad (26)$$

defines a metric on the space of equivalence classes because $\mu([a]^-, [b]^-) = 0 \Leftrightarrow v(a, b) = 0 \Leftrightarrow a \sim b \Leftrightarrow [a]^- = [b]^-$.

If we define $\mathcal{M} \sim \mathcal{H}$ if $\lambda_{\mathcal{M}} = \lambda_{\mathcal{H}}$, then this is a network equivalence relation. If we focus on equivalence classes of networks rather than the networks themselves, we are in effect identifying equivalent networks. For instance, if we are only interested in steady-state distributions, then it may be advantageous to identify networks possessing the same steady-state distribution.

4. INFERENCE PERFORMANCE

An inference procedure operates on data generated by a network \mathcal{H} and constructs an inferred network \mathcal{M} to serve as an estimate of \mathcal{H} , or it constructs a characteristic to serve as an estimate of the corresponding characteristic of \mathcal{H} . For instance, the data may be used to infer a distribution that estimates the steady-state distribution of \mathcal{H} . The data could be dynamical, consisting of time-course observations, or it might be taken from the steady state, as with microarray measurements assumed to come from the steady state of some phenotypic class. In the latter case, it makes sense to consider inference accuracy relative to the steady-state distribution of \mathcal{H} , rather than \mathcal{H} itself. For full network inference, the inference procedure is a mathematical operation, a mapping from a space of samples to a space of networks, and it must be evaluated as such. There is a generated data set S and the inference procedure is of the form $\psi(S) = \mathcal{M}$. If a characteristic is being estimated, then $\psi(S)$ is a characteristic, for instance, $\psi(S) = F$, a probability distribution.

Measuring Inference Performance Using Distance Functions

Focusing on full network inference, the goodness of an inference procedure ψ is measured relative to some distance, μ , specifically, $\mu(\mathcal{M}, \mathcal{H}) = \mu(\psi(S), \mathcal{H})$, which is a function of the sample S . In fact, S is a realization of a random set process, Σ , governing data generation from \mathcal{H} . In general, there is no assumption on the nature of Σ . It might be directly generated by \mathcal{H} or it might result from directly generated data corrupted by noise of some sort. $\mu(\psi(\Sigma), \mathcal{H})$ is a random variable and the performance of ψ is characterized by the distribution of $\mu(\psi(\Sigma), \mathcal{H})$, which depends on the distribution of Σ .

The salient statistic regarding the distribution of $\mu(\psi(\Sigma), \mathcal{H})$ is its mean, $E_{\Sigma}[\mu(\psi(\Sigma), \mathcal{H})]$, where the expectation is taken with respect to Σ .

Rather than considering a single network, we can consider a distribution, H , of random networks, where, by definition, the occurrences of realizations \mathcal{H} of H are governed by a probability distribution. This is precisely the situation with regard to the classical study of random Boolean networks. Averaging over the class of random networks, our interest focuses on

$$\mu^*(H, \Sigma, \psi) = E_H[E_{\Sigma}[\mu(\psi(\Sigma), H)]] \quad (27)$$

It is natural to define the inference procedure ψ_1 better than the inference procedure ψ_2 relative to the distance μ , the random network H , and the sampling procedure Σ if

$$\mu^*(H, \Sigma, \psi_1) < \mu^*(H, \Sigma, \psi_2) \quad (28)$$

Whether an inference procedure is “good” is not only relative to the distance function, it is relative to how one views the value of the expected distance. Indeed, it is not really possible to determine an absolute notion of goodness.

In practice, the expectation is estimated by an average,

$$\hat{\mu}(H, \Sigma, \psi) = \frac{1}{m} \sum_{j=1}^m \mu(\psi(S_j), \mathcal{H}_j) \quad (29)$$

where S_1, S_2, \dots, S_m are sample point sets generated according to Σ from networks $\mathcal{H}_1, \mathcal{H}_2, \dots, \mathcal{H}_m$ randomly chosen from H .

The preceding analysis applies virtually unchanged when a characteristic is being estimated. One need only replace \mathcal{H} and H by λ and Λ , where λ and Λ are a characteristic and a random characteristic, respectively, and replace the network distance μ by the characteristic distance.

We next present three examples using previously introduced distance functions to measure inference performance. Algorithm description will be sketchy in order to avoid long digressions from the issue of distance illustration. We defer to the cited literature for details.

Example 1. The Boolean network model has been in existence for a long time and various inference procedures have been proposed [16-18]. One proposed method for Boolean networks with perturbation is based on the observation of a single dynamic realization of the network [6]. This method will be discussed in some detail in Section 5 in regard to consistent inference; for now, we are only concerned with the distance between the inferred network and the original network generating the data, where the distance function is given by $\mu_{fun}(\mathcal{M}, \mathcal{H})$ in Eq. 7. Fig. (1) shows the average (in percentage) of the distance function using 80 data sequences generated from 16 randomly generated Boolean networks with 7 genes, perturbation probability $p = 0.01$, uniform connectivity $k = 2$ or $k = 3$, and data sequence lengths varying from 500 to 40,000. The reduction in performance from connectivity 2 to connectivity 3 is not surprising because the number of truth-table lines jumps dramatically.

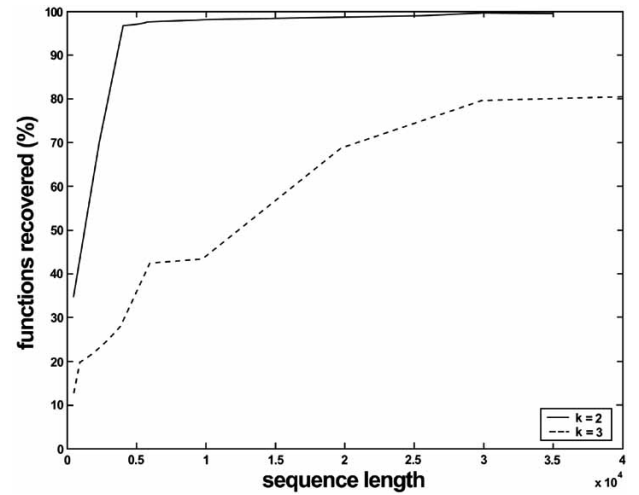


Fig. (1). Rule-based distance performance for Boolean-network inference for connectivity $k = 2, 3$.

Example 2. There have been a number of papers addressing the inference of connectivity graphs using information-theoretic approaches [9, 10, 19]. In a study proposing using the minimum description length (MDL) principle to infer regulatory graphs [8], the hamming distance was used to compare the performance of the newly proposed algorithm with an earlier information-theoretic algorithm, called REVEAL [9]. Fig. (2) compares the hamming distances between the inferred networks and the corresponding synthetic networks that generated the data relative to increasing sample size. It does so for the REVEAL algorithm and the MDL algorithm using three different settings for a user-defined parameter. The performance measures are obtained by averaging over 30 randomly generated networks, each containing 20 genes and 30 edges, with the distance function being normalized over 30, the number of edges in the synthetic networks.

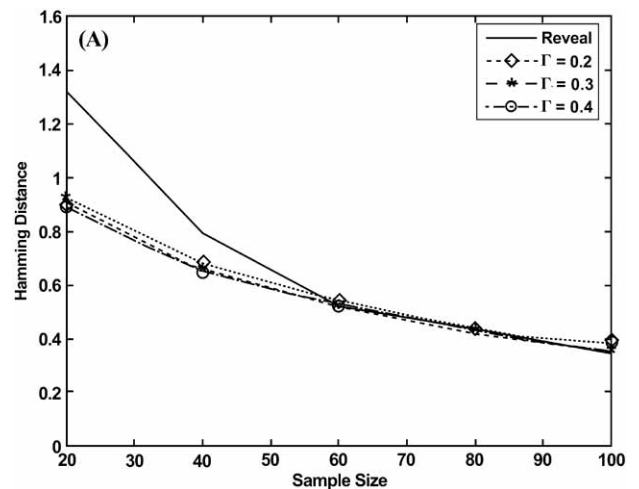


Fig. (2). Hamming distance performance for inferring regulatory graphs using information theory.

Example 3. A probabilistic Boolean network (PBN) is a network defined as a collection of discrete-valued networks such that at any point in time one of the constituent networks is controlling the network dynamics [20]. In a context-sensitive PBN there is a binary random variable determining whether there should be a switch of constituent networks at that time point, the modeling assumption being that there are latent variables outside the model network whose changes induce stochasticity into the PBN [21]. Typically, there is also a probability of permutation. This example considers a Bayesian connectivity-based inference procedure for designing PBNs from steady-state data [22]. A synthetic PBN, \mathcal{H} , composed of two constituent Boolean networks is used to generate a random sample of size 60 from its steady-state distribution and the inference procedure is used to construct a designed PBN composed of ten constituent PBNs (note that the inference procedure does not have input relating to the number of constituent BNs of the generating network). According to definition, the attractors of a PBN are the attractors of its constituent BNs. \mathcal{H} has six singleton attractors, two of which, call them \mathbf{x}_a and \mathbf{x}_b , contain 0.99 of the steady-state mass. The designed PBN has more attractors, which is not uncommon, but \mathbf{x}_a and \mathbf{x}_b appear in all ten constituent networks as singleton attractors and they contain 0.78 of the steady-state mass. Since, for PBNs with low probability of network switching almost all of the steady-state mass lies in its attractors [21], $\mu_{\text{stead}}^1(\psi(S), \mathcal{H}) \leq 0.21$ (or approximately so), the maximum 1-norm being 2.

5. CONSISTENCY

The greater the amount of data, the better inference one can expect. The hope is that, for large data sets, the inferred network will be close to the generating network. We define an inference procedure, ψ , to be *consistent* if $\mu^*(\mathcal{H}, \Sigma, \psi) \rightarrow 0$ as $|\Sigma| \rightarrow \infty$. We illustrate consistency using Boolean networks with perturbation. We use the inference procedure referred to in Example 1 that applies to a single observed time series [6] and the distance function μ_{fun} of Eq. 7.

Owing to perturbation, the network has a steady-state distribution and all states communicate with each other. Hence, given a long time series we are likely to observe most of the states and their corresponding state-to-state transitions $\mathbf{x}_k \rightarrow \mathbf{x}_{(k)}$, for $k = 1, 2, \dots, N$, where $\mathbf{x}_{(k)}$ denotes the next state following \mathbf{x}_k under the network state function. If we ignore perturbation, then using the observed state-to-state transitions we can construct a table of state-to-gene transitions of the form $\mathbf{x}_k \rightarrow x_i$, for $k = 1, 2, \dots, N$ and $i = 1, 2, \dots, n$. These define the functions f_1, f_2, \dots, f_n accordingly. Because the truth table for function f_i has 2^n rows of the form $f_i(\mathbf{x}_k)$, some rows may be empty owing to insufficient observations and these rows can be filled in randomly. As the length of the time series increases, the probability of not observing the state \mathbf{x}_k goes to 0. Indeed, for any positive integer c , if we let $\eta(\mathbf{x}_k)$ denote the number of times \mathbf{x}_k is observed in the time series, then $P(\eta(\mathbf{x}_k) \geq c) \rightarrow 1$ as $|\Sigma| \rightarrow \infty$, where the probability is with respect to the time series Σ .

With perturbation, the state-to-state transitions do not directly define functions because state \mathbf{x}_k may transition to more than one state. However, assuming a perturbation probability less than 0.5, the transitions from \mathbf{x}_k will be dominated by the single transition determined by the state function \mathbf{f} and this dominating choice can be used for inference. Letting $\eta_j(\mathbf{x}_k)$ denote the number times we observe the transition $\mathbf{x}_k \rightarrow \mathbf{x}_j$, if $\mathbf{f}(\mathbf{x}_k) = \mathbf{x}_{(k)}$ is the function-defined transition, then

$$P(\eta_{(k)}(\mathbf{x}_k) > \max\{\eta_1(\mathbf{x}_k), \eta_2(\mathbf{x}_k), \dots, \eta_N(\mathbf{x}_k)\}) \rightarrow 1 \text{ as } |\Sigma| \rightarrow \infty \quad (30)$$

Thus, if $\hat{\mathbf{f}}$ denotes the inferred state function, then $P(\hat{\mathbf{f}} = \mathbf{f}) \rightarrow 1$ as $|\Sigma| \rightarrow \infty$. Similar asymptotic statements hold for f_1, f_2, \dots, f_n . This insures that, for any $\tau > 0$, $P(\mu_{\text{fun}}(\psi(\Sigma), \mathcal{H}) < \tau) \rightarrow 1$ as $|\Sigma| \rightarrow \infty$ for any Boolean network \mathcal{H} . Since $\mu_{\text{fun}}(\psi(\Sigma), \mathcal{H}) \leq 1$, this is equivalent to $E_{\Sigma}[\mu_{\text{fun}}(\psi(\Sigma), \mathcal{H})] \rightarrow 0$ as $|\Sigma| \rightarrow \infty$. Finally, if \mathcal{H} is the class of all Boolean networks on n genes with perturbation probability p , then, since \mathcal{H} is a finite set,

$$\mu_{\text{fun}}^*(\mathcal{H}, \Sigma, \psi) = E_{\mathcal{H}}[E_{\Sigma}[\mu_{\text{fun}}(\psi(\Sigma), \mathcal{H})]] \rightarrow 0 \text{ as } |\Sigma| \rightarrow \infty \quad (31)$$

and the inference procedure is consistent relative to μ_{fun} .

The preceding argument assumes that the perturbation probability is known. A modification of the inference procedure yields an estimator for p [6]; however, if p is also being estimated, then the model space \mathcal{H} is no longer finite and the consistency proof has to be modified. We do not believe this is the proper place to go into such mathematical issues.

6. APPROXIMATION

Inference performance is evaluated based on the ability of an inference procedure to identify the network from which the data have been derived. This can only be done exactly if the data-generating network is known. Suppose we do not know the random network, \mathcal{H} , generating the data for which we want to evaluate the inference procedure, ψ , but know a network \mathcal{N} that we believe to be a good approximation to the networks in \mathcal{H} . We might then compare the inferred network to \mathcal{N} . In effect, such a comparison is approximating $\mu^*(\mathcal{H}, \Sigma, \psi)$ by $E_{\Sigma}[\mu(\psi(\Sigma), \mathcal{N})]$.

The key issue is approximation accuracy. The triangle inequality implies

$$\mu(\psi(S), \mathcal{N}) - \mu(\mathcal{N}, \mathcal{H}) \leq \mu(\psi(S), \mathcal{H}) \leq \mu(\psi(S), \mathcal{N}) + \mu(\mathcal{N}, \mathcal{H}) \quad (32)$$

for any sample set S and $\mathcal{H} \in \mathcal{H}$. Hence,

$$\begin{aligned} E_{\Sigma}[\mu(\psi(\Sigma), \mathcal{N})] - E_{\mathcal{H}}[\mu(\mathcal{N}, \mathcal{H})] \\ \leq E_{\mathcal{H}}[E_{\Sigma}[\mu(\psi(\Sigma), \mathcal{H})]] \\ \leq E_{\Sigma}[\mu(\psi(\Sigma), \mathcal{N})] + E_{\mathcal{H}}[\mu(\mathcal{N}, \mathcal{H})] \end{aligned} \quad (33)$$

If $E_{\mathcal{H}}[\mu(\mathcal{N}, \mathcal{H})] \approx 0$, meaning that $E_{\mathcal{H}}[\mu(\mathcal{N}, \mathcal{H})]$ is small, then the preceding inequality leads to the approximate inequality

$$E_{\Sigma}[\mu(\psi(\Sigma), \mathcal{N})] \leq \mu^*(H, \Sigma, \psi) \leq E_{\Sigma}[\mu(\psi(\Sigma), \mathcal{N})] \quad (34)$$

Thus, if $E_H[\mu(\mathcal{N}, H)] \approx 0$, then

$$\mu^*(H, \Sigma, \psi) \approx E_{\Sigma}[\mu(\psi(\Sigma), \mathcal{N})] \quad (35)$$

and it is reasonable to judge the performance of ψ relative to H by $E_{\Sigma}[\mu(\psi(\Sigma), \mathcal{N})]$. On the other hand, if $E_H[\mu(\mathcal{N}, H)]$ is not small, then both bounds in Eq. 33 are loose and nothing can be asserted regarding the performance of ψ relative to the data sets on which it is being applied. Therefore, unless $E_H[\mu(\mathcal{N}, H)]$ is small, the entire validation procedure is flawed because the approximation of H by \mathcal{N} is confounding the procedure. In addition, if $E_H[\mu(\mathcal{N}, H)] \approx 0$, one still has to estimate $E_{\Sigma}[\mu(\psi(\Sigma), \mathcal{N})]$, which generally means that the number of sample sets is sufficiently large that the expectation is well-estimated by the average distance.

The preceding approximation methodology is common in the literature. A proposed inference procedure is applied to one or more real data sets. The inferred network is compared, not to the unknown random network generating the data, but to a model network that has been human-constructed from the literature (and implicitly assumed to approximate the data-generating network). For instance, a directed graph (adjacency matrix), \mathbf{A} , is constructed from relations found in the literature and the hamming distance is used in the approximating expectation, $E_{\Sigma}[\mu_{ham}(\psi(\Sigma), \mathbf{A})]$, in Eq. 35. The aim is to compare the result of the inference procedure to some characteristic related to existing biological knowledge. The problem is that the constructed regulatory graph may not be a good approximation to the regulatory graph for the system generating the data. This can happen because the literature is incomplete, there are insufficiently validated connections reported in the literature, or the conditions under which connections have been discovered, or not discovered, in certain papers are not compatible with the conditions under which the current data have been derived. As a result of any of these situations, the overall validation procedure is confounded by the precision (or lack thereof) of the approximation.

7. VALIDATION FROM EXPERIMENTAL DATA

Another form of approximation results from using experimental data for validation rather than synthetic data generated from a known, ground-truth model. In this situation, there is a *test-data* sampling procedure generating data from which an estimate of the desired characteristic corresponding to the underlying physical network is formed. Validation is then *via* the random variable $\mu(\psi(\Sigma), \xi(\Omega))$, where Σ is the *training-data* sampling procedure used to design the network and Ω is a real-data *test-sampling* procedure to validate the designed network by direct construction of the characteristic *via* independent sampling. To simplify the notation we consider a single underlying network \mathcal{H} rather than a random network H . In this situation, $E_H[\mu(\mathcal{N}, H)]$ in Eq. 33 is replaced by $\mu(\xi(\Omega), \lambda_{\mathcal{H}})$, where $\lambda_{\mathcal{H}}$ is the characteristic for \mathcal{H} , and Eq. 33 takes the form

$$\begin{aligned} & E_{\Omega}[E_{\Sigma}[\mu(\psi(\Sigma), \xi(\Omega))]] - E_{\Omega}[\mu(\xi(\Omega), \lambda_{\mathcal{H}})] \\ & \leq E_{\Sigma}[\mu(\psi(\Sigma), \lambda_{\mathcal{H}})] \\ & \leq E_{\Omega}[E_{\Sigma}[\mu(\psi(\Sigma), \xi(\Omega))]] + E_{\Omega}[\mu(\xi(\Omega), \lambda_{\mathcal{H}})] \end{aligned} \quad (36)$$

If $E_{\Omega}[\mu(\xi(\Omega), \lambda_{\mathcal{H}})] \approx 0$, then

$$\mu^*(\mathcal{H}, \Sigma, \psi) \approx E_{\Omega}[E_{\Sigma}[\mu(\psi(\Sigma), \xi(\Omega))]] \quad (37)$$

If ξ is a consistent estimator of $\lambda_{\mathcal{H}}$, so that $E_{\Omega}[\mu(\xi(\Omega), \lambda_{\mathcal{H}})] \approx 0$ for large test samples, then, on average, the approximation is good.

Consider what happens if one only has data to estimate (train) the model, which may happen when data are limited on account of cost or the availability of samples. In this case, one tests on the same data, thereby having $\Omega = \Sigma$ in Eq. 36 and the *resubstitution* estimate, $E_{\Sigma}[\mu(\psi(\Sigma), \xi(\Sigma))]$, in Eq. 37. If ξ is a consistent estimator of $\lambda_{\mathcal{H}}$ and the single training sample is large, then the conclusion of Eq. 37 again holds. But we do not have a large sample. Hence, Eq. 36 cannot be used to insure good average performance. But it also cannot be used to insure good performance when there is a small independent test-data sample. In the independent case, we are concerned with the absolute difference

$$\Delta_{test} = |E_{\Sigma}[\mu(\psi(\Sigma), \lambda_{\mathcal{H}})] - E_{\Omega}[E_{\Sigma}[\mu(\psi(\Sigma), \xi(\Omega))]]| \quad (38)$$

When the same data are used for training and testing, our interest is with

$$\Delta_{train} = |E_{\Sigma}[\mu(\psi(\Sigma), \lambda_{\mathcal{H}})] - E_{\Sigma}[\mu(\psi(\Sigma), \xi(\Sigma))]| \quad (39)$$

As with classification, where resubstitution error estimation is usually biased low owing to overfitting by the classification rule, in the case of network validation, resubstitution is risky because the characteristic of the designed network is being compared to a characteristic inferred from the same data with which the network has been designed. According to Eq. 36, as in the case of classification, this is not a problem for large samples, but it can be a serious problem for small samples because overfitting can cause Δ_{train} to be much less than Δ_{test} . Whereas substantial effort has gone into studying these kinds of problems in pattern recognition, there appears to be an absence of the analogous study for network validation.

Example 4. An attractor-preserving inference method for PBNs based on steady-state data has been proposed and applied to PBNs [23]. A PBN has been designed from cDNA microarray data using 7 genes: WNT5A, pirin, S100P, RET1, MART1, HADHB, and STC2. The steady-state distribution of the designed network has been compared to the histogram of the data, the histogram serving as an estimate of the steady-state distribution of the underlying physical network. Fig. (3) illustrates the comparison of the portion of the steady-state distribution corresponding to the data states with the data histogram. Referring to Eq. 14, the 1-norm and 2-norm yield the resubstitution error distances $\mu_{stead}^1(\psi(S), \xi(S)) = 0.45$ (out of a maximum of 2) and $\mu_{stead}^2(\psi(S), \xi(S))$

= 0.1262, respectively, the latter being the root-mean-square error.

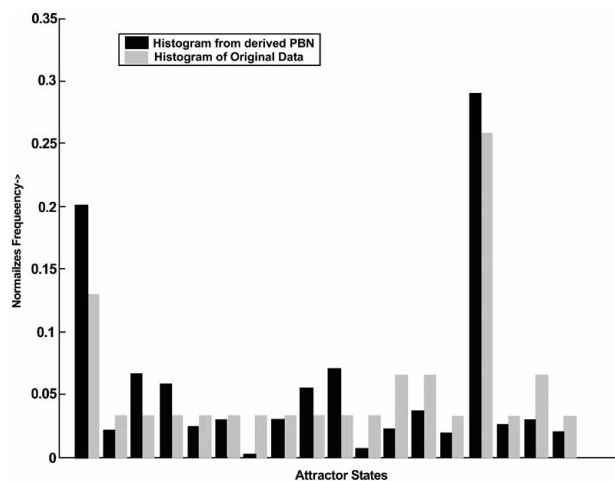


Fig. (3). Comparison of steady-state distribution for a designed network and data histogram.

8. CONCLUSION

This paper has proposed a mathematically rigorous framework for the validation of inference procedures for gene regulatory networks and has illustrated this framework employing validation methods used in the literature. Owing to the central role of regulatory networks in systems biology and the need to apply inference procedures to the massive data sets resulting from high-throughput technologies, validation cannot be left to *ad hoc* methods whose own performances are not understood. A formal framework is necessary. As should be clear from the paper, a great deal of work needs to be done to establish the properties of inference procedures under various conditions, such as the sampling procedure, model class, and validation criterion (distance function). Absent rigorous results in this regard, proposed inference procedures will remain speculative and the quality of their performances unknown. A sound epistemology will be lacking.

ACKNOWLEDGEMENTS

I would like to acknowledge the National Science Foundation (CCF-0514644) and the National Cancer Institute (2 R25CA090301-06) for supporting this work. I also wish to thank Wentao Zhao for assisting with the literature search and Barak Faryabi for proof reading the manuscript.

REFERENCES

- [1] de Jong, H. Modeling and simulation of genetic regulatory systems: a literature review. *Computat. Biol.*, **2002**, 9(1): 67-103.
- [2] Shmulevich, I., Dougherty, E. R. *Genomic Signal Processing*, Princeton University Press, Princeton, **2007**.
- [3] Datta, A., Pal, R., Dougherty, E. R. Intervention in probabilistic gene regulatory networks. *Curr. Bioinform.*, **2006**, 1(2): 167-184.
- [4] Kauffman, S. A. Metabolic stability and epigenesis in randomly constructed genetic nets. *Theoret. Biol.*, **1969**, 22: 437-467.
- [5] Kauffman, S. A. *The Origins of Order*, Oxford University Press, New York, **1993**.
- [6] Marshall, S., Yu, L., Xiao, Y., Dougherty, E. R. Inference of probabilistic Boolean networks from a single observed temporal sequence. *EURASIP J. Bioinform. Sys. Biol.*, **2007**, Article ID 32454: 15 pages.
- [7] Bernard, A., Hartemink, A. Informative structure priors: joint learning of dynamic regulatory networks from multiple types of data. *Pacific Sympo. Biocomput.*, **2005**, 10: 459-470.
- [8] Zhao, W., Serpedin, E., Dougherty, E. R. Inferring gene regulatory networks from time series data using the minimum description length principle. *Bioinformatics* **2006**, 22(17): 2129-2135.
- [9] Liang, S., Fuhrman, S., Somogyi, R. REVEAL a general reverse engineering algorithm for inference of genetic network architectures. *Pacific Sympo. Biocomput.*, **1998**, 3: 18-29.
- [10] Margolin, A., Nemenman, I., Basso, K., Klein, U., Wiggins, C., Stolovitzky, G., Favera, R. D., Califano, A. ARACNE: an algorithm for reconstruction of genetic networks in a mammalian cellular context. *BMC Bioinform.*, **2006**, 7 (Suppl. 1): S7.
- [11] Chen, X., Anantha, G., and Wang, X. An effective structure learning method for constructing gene networks. *Bioinformatics* **2006**, 22(11): 1367-1374.
- [12] Zou, M., Conzen, S. D. A new dynamic Bayesian network (DBN) approach for identifying gene regulatory networks from time course microarray data. *Bioinformatics* **2005**, 21(1), 71-79.
- [13] Bansal, M., Belcastro, V., Ambesi-Implombato, A., di Bernardo, D. How to infer gene networks from expression profiles. *Mol. Syst. Biol.*, **2006**, 3: 78.
- [14] Kim, S., Li, H., Dougherty, E. R., Chao, N., Chen, Y., Bittner, M. L., Suh, E. B. Can Markov chain models mimic biological regulation. *Biol. Syst.*, **2002**, 10(4): 447-458.
- [15] Brun, M., Kim, S., Choi, W., Dougherty, E. R. Comparison of network models via steady-state trajectories. *EURASIP J. Bioinform. Syst. Biol.*, **2007**, Article ID 82702: 11 pages.
- [16] Akutsu, T., Miyano, S., and Kuhara, S. Identification of genetic networks from a small number of gene expression patterns under the Boolean model. *Pacific Sympo. Biocomput.*, **1999**, 4: 17-28.
- [17] Lahdesmaki, H., Shmulevich, I., and Yli-Harja, O. On learning gene regulatory networks under the Boolean network model. *Machine Learn.*, **2003**, 52: 147-167.
- [18] Shmulevich, I., Saarinen, A., Yli-Harja, O., Astola, J. Inference of genetic regulatory networks via best-fit extensions. *Computat. Statist. Approaches Genom.*, **2002**, 197-210, eds. W. Zhang and I. Shmulevich, Kluwer Academic Publishers, Boston.
- [19] Nemenman, I. Information theory, multivariate dependence, and genetic network inference. *Technical Report NSF-KITP-04-54, KITP, UCSB*, **2004**.
- [20] Shmulevich, I., Dougherty, E. R., Kim, S., Zhang, W. Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks. *Bioinformatics* **2002**, 18: 261-274.
- [21] Brun, M., Dougherty, E. R., Shmulevich, I. Steady-state probabilities for attractors in probabilistic Boolean networks. *EURASIP J. Signal Process.*, **2005**, 85(10): 1993-2013.
- [22] Zhou, X., Wang, X., Pal, R., Ivanov, I., Bittner, M. L., Dougherty, E. R. A Bayesian connectivity-based approach to constructing probabilistic gene regulatory networks. *Bioinformatics* **2004**, 20(17): 2918-2927.
- [23] Pal, R., Ivanov, I., Datta, A., Dougherty, E. R. Generating Boolean networks with a prescribed attractor structure. *Bioinformatics* **2005**, 21(21): 4021-4025.