

Design of Probabilistic Boolean Networks Under the Requirement of Contextual Data Consistency

Edward R. Dougherty, *Member, IEEE*, and Yufei Xiao, *Student Member, IEEE*

Abstract—A key issue of genomic signal processing is the design of gene regulatory networks. A probabilistic Boolean network (PBN) is composed of a family of Boolean networks. It stochastically switches between its constituent networks (contexts). For network design, connectivity and transition rules must be inferred from data via some optimization criterion. Except rarely, the optimal rule for a gene will not be a perfect predictor because there will be inconsistencies in the data. It would be natural to model these inconsistencies to reflect changes in PBN contexts. If we assume inconsistencies result from the data arising from a random function, then design involves finding the realizations of a random function and the probability mass on those realizations so that the resulting random function best fits the data relative to the expectation of its output and does so using a minimal number of realizations. We propose PBN design satisfying the biological assumption that data are consistent within a context, for which the distribution of the network agrees with the empirical distribution of the data, and such that this is accomplished with a minimal number of contexts. The design also satisfies the biological constraint that, because the network spends the great majority of time in its attractors, all data states should be attractor states in the model.

Index Terms—Data consistency, gene regulatory network, graphical model, network inference.

I. INTRODUCTION

PROBABILISTIC Boolean networks (PBNs) represent an interface between the determinism of Boolean networks and the probabilistic nature of Bayesian networks by incorporating rule-based uncertainty [15]. This compromise is important because rule-based dependencies between genes are biologically meaningful, while mechanisms for handling uncertainty are conceptually and empirically necessary. The binary (Boolean) nature of PBNs has been assumed so as to model ON-OFF switching behavior, but their structure extends easily to any discrete (multi-valued) setting, thereby yielding a general framework for probabilistic gene regulatory networks (PGRNs), which, owing to the multivariate logical character of their rules, are also typically referred to as PBNs. The dynamics of these networks can be studied in the probabilistic context of Markov

chains, thereby facilitating steady-state analysis. PBNs offer the potential to design treatment strategies based on the application of external control variables to drive network dynamics [16], [12].

A key issue is network design (inference) from data [1], [11]. Network connectivity and transition rules must be inferred, with perhaps the imposition of biological constraints [19]. When building function-based gene networks from expression data, the functions are typically derived via some optimization-based criterion. This requires determining, for each gene g , the genes that will serve as input to the function giving the value of g and the structure of the function. The original method proposed for PBNs is based on the coefficient of determination [15]. Other methods have been proposed, including optimizing the connectivity of the network according to the data in a Bayesian framework [22].

Except in rare circumstances, the optimal function for a gene will not be a perfect predictor because there will be inconsistencies in the data. This means that a specific vector of values for a set of regulatory genes will not necessarily correspond to a single value of the target gene. Thus, network design is inherently probabilistic. In this paper, we model inconsistencies in a way that reflects context changes in regulation. The network can be in any of a number of contexts. Within a context, the network behaves deterministically, and the generated data is consistent. If a regulatory set takes on a specific vector of values, then the target gene associated with the regulatory set must take on a single value.

Formally, a *probabilistic gene regulatory network* is composed of a set of n genes, g_1, g_2, \dots, g_n , each taking values in a finite set V (containing d values), and a set of vector-valued *network functions*, $\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_r$, governing the state transitions of the genes. There is a set of state vectors $S = \{\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^m\}$, with $m = d^n$ and $\mathbf{x}^k = (x_{k1}, x_{k2}, \dots, x_{kn})$, where x_{ki} is the value of gene g_i in state k . Each network function \mathbf{f}_j is composed of n functions $\psi_{j1}, \psi_{j2}, \dots, \psi_{jn}$, and the value of gene g_i at time $t + 1$ is given by $g_i(t + 1) = \psi_{ji}[g_1(t), \dots, g_{i-1}(t), g_{i+1}(t), \dots, g_n(t)]$. The choice of which network function \mathbf{f}_j to apply is governed by a selection procedure. At each time point, a random decision is made as to whether to switch the network function for the next transition, with a probability q of a change being a system parameter. If a decision is made to change the network function, then a new function is chosen from among $\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_r$, with the probability of choosing \mathbf{f}_j being the selection probability c_j . Finally, at each time point, there is a probability p of any gene changing its value uniformly randomly to another value in V . Whereas a network switch corresponds to a change in a latent variable

Manuscript received July 5, 2005; revised September 27, 2005. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Jan C. de Munck. This work was supported in part by the National Human Genome Research Institute, the National Cancer Institute, and the National Science Foundation under grant CCF-0514644.

E. R. Dougherty is with the Department of Electrical Engineering, Texas A&M University, College Station, TX 77843 USA, and also with the Computational Biology Division, Translational Genomics Research Institute, Phoenix, AZ 85004 USA (e-mail: e-dougherty@tamu.edu).

Y. Xiao is with the Department of Electrical Engineering, Texas A&M University, College Station, TX 77843 USA (e-mail: fei@neo.tamu.edu).

Digital Object Identifier 10.1109/TSP.2006.877641

causing a structural change in the functions governing the network, for instance, in the case of a gene outside the network model that participates in the regulation of a gene in the model, a random perturbation corresponds to a transient value flip that leaves the network wiring unchanged, as in the case of activation or inactivation owing to external stimuli such as mutagens, heat stress, etc.

The state space S of the network together with the set of network functions, in conjunction with transitions between the states and network functions, determine a Markov chain, the states of the Markov chain being of the form $(\mathbf{x}^i, \mathbf{f}_j)$. The random perturbation makes the Markov chain ergodic, meaning that it has the possibility of reaching any state from another state and that it possesses a steady-state distribution. In the special case when $q = 1$, a network function is randomly chosen at each time point, the Markov chain consists only of the PGRN states, and the PGRN is said to be *instantaneously random*. When $q < 1$, the PGRN is said to be *context sensitive*, the idea being that network changes result from the genes responding to *latent* variables external to the model network.

We confine ourselves to the binary setting, in which case the preceding description characterizes a probabilistic Boolean network. Each state vector consists of “0”s and “1”s, and each network function consists of a set of n Boolean functions that can be represented by a truth table. One can view a PBN as a collection of Boolean networks, each defined by a network function, with the probability q and the selection probabilities governing how the PBN switches between Boolean networks. The Boolean setting simplifies the analysis, but it is not restrictive since the analysis goes through for any finite valuation set.

Attractors play a key role in Boolean networks. Given a starting state, within a finite number of steps, the network will transition into a cycle of states, called an *attractor*, and absent perturbation will continue to cycle thereafter. Each attractor is a subset of a *basin* composed of those states that lead to the attractor if chosen as starting states. The basins form a partition of the state space for the network. Nonattractor states are transient. They are visited at most once on any network trajectory. When modeling genetic regulatory networks, attractors are often identified with phenotypes [8]. Real biological systems are typically assumed to have short attractor cycles, with singleton attractors being of special import. Much of our interest concerns singleton attractors.

By definition, the attractors of a PBN are the attractors of its constituent Boolean networks. Once in an attractor cycle, the network will remain in the cycle unless thrown out by a random perturbation or network switch. Assuming these to be infrequent, when observed, the network will, with high probability, be in an attractor state. The probability of a PBN being in a particular state can be quantitatively expressed given information on the perturbation and switching probabilities, the attractor structure, and the basin structure [3]. Thus, the probability of an observation vector being an attractor can be precisely determined for a given model.

Regarding the use of the Boolean framework, we note that the logical character of gene regulation has been recognized for some time [18], [4], [8] and the dynamical behavior of Boolean networks can be used to model many biologically phenomena,

such as cellular state dynamics possessing switchlike behavior, stability, and hysteresis [7]. From almost the inception of microarray analysis, logical relations among genes have been constructed from the data [5], [9] and recently the manifestation of logical relations in the continuous data has been analyzed [13]. Besides the fact that we are concerned in many of our applications with ON–OFF-type behavior, an important practical reason for working in the binary setting, or at least in the context of a very coarse quantization, is the exponentially increasing complexity (and therefore data requirement) with finer quantization. The general question as to whether certain genes, when quantized as binary switches can be informative in separating phenotype classes such as tumors and normal tissue, as well as different stages of tumor development, depends on the bimodality of their behavior. The potential for binary discrimination has been shown for clustering [17] and classification [21]. The former has a good discussion of binarization. When using microarray data, which integrates expression over a collection of cells, it should be recognized that we are modeling global behavior, not the activity of individual cells, so that binarization corresponds to global bimodality.

Overall, we propose an inference procedure for PBNs whose contexts model the data in such a way that they are consistent for each context, the intent being to view data inconsistencies as being due to latent variables. Separate sections are dedicated to data-consistent inference, data-consistent operator design, and data-consistent PBN design. We follow these with a discussion of the relationship between standard and data-consistent designs, the role of data filtering, application to a melanoma-related network, and some concluding remarks.

II. INFERENCE AND DATA CONSISTENCY

PBN inference has mainly been based on classical binary optimization, where the predictor variables for each target gene are selected using the *coefficient of determination* (CoD). The CoD measures the degree to which the best estimate for the value of a target gene is improved using the knowledge of the values of a set of predictor genes, relative to the best estimate in the absence of knowledge of the predictors. Formally, $\text{CoD} = (\varepsilon_0 - \varepsilon_{\text{opt}}) / \varepsilon_0$, where ε_0 is the error arising when using the best estimate of the target-gene expression level given only statistics relating to the target gene itself, and ε_{opt} is the error arising using the best estimate of the target-gene level using the levels of the predictor genes. If a predictor set can perfectly predict a target, then $\varepsilon_{\text{opt}} = 0$ and $\text{CoD} = 1$; if a predictor set provides no information about the target, then $\varepsilon_{\text{opt}} = \varepsilon_0$ and $\text{CoD} = 0$. In general, $0 \leq \text{CoD} \leq 1$.

If we fix ahead the number of predictor genes (referred to as the *connectivity*) that can compose a regulatory set for a target gene, then the original design method is to choose the regulatory set with the largest CoD and then define a binary regulatory function based on the genes in the regulatory set. For instance, suppose genes g_1 , g_2 , and g_3 have the highest collective CoD among all triples for predicting gene g . Let $wxyz$ denote a binary vector of values for (g_1, g_2, g_3, g) . If 0001 appears more often in the data than 0000, then for $xyz = 000$ the predictor function is defined by $\psi(000) = 1$; otherwise, it is defined by

$\psi(000) = 0$ (ties being broken either by convention or randomly). If both 0000 and 0001 appear in the data, then the data are inconsistent relative to predicting g via $g_1, g_2,$ and g_3 . Inconsistency means that the data are interpreted in such a way that the predictor is a random function: The same values of the predictors can yield different values of the target. This interpretation is problematic under the assumption that biological regulation is deterministically encoded in the genes.

In this paper, we address an inherent problem that leads to inconsistency. Consider a network with two contexts, C_0 and C_1 . If the regulatory genes $g_1, g_2,$ and g_3 form the vector 001 in context C_0 , then their target gene g must take on a specific value, say 0, in C_0 . This uniqueness condition holds for all vectors of values for $g_1, g_2,$ and g_3 . It may be that in context C_1 , the regulatory genes take the vector 001, while gene g has value 1, but the data are consistent so long as a single context is maintained. Unless the contexts are known when data from the network are sampled, it would appear that the network is not operating consistently. Since the context is generally not known, an experiment is likely to yield n_0 and n_1 observations of 0 and 1, respectively, meaning that 001 has been observed n_0 and n_1 times in contexts C_0 and C_1 , respectively. The regulatory function ψ for g would then be defined for 001 by $\psi(001) = 0$ if $n_0 > n_1$ and $\psi(001) = 1$ if $n_1 > n_0$, with some convention determining $\psi(001)$ if $n_1 = n_0$.

Here, we take a different approach. If the data reveal two values for a target gene for a single vector for the regulatory set, then we construct the network so that there are two distinct functions, ψ_0 and ψ_1 , such that $\psi_0(001) = 0$ and $\psi_1(001) = 1$. The functions represent different network contexts. The probabilities of the two functions being selected for regulation will be in agreement with the context probabilities.

Since the context is selected by external variables, we cannot know deterministically when the system is in a certain context, but we can infer the probability of the system being in a particular context from the data. Our basic criterion for network design is that the distribution of expected state observations for the system, if it is observed over a long period of time, agrees with the observed distribution of states for the data. As for consistency, that holds *ipso facto* because the system behaves deterministically so long as it remains in a fixed context (is determined by the unique set of functions defining that context). Whereas with the previous design methods the number of constituent networks depends on the number of high-CoD predictor sets [15] or high Bayes-score predictor sets [22], and these depend on the designer's choice of a threshold, using the approach of the present paper, the number of constituent networks is determined by the data.

Owing to their importance, it is key that attractors are properly modeled in an inferred PBN. If the switching and perturbation probabilities are very small, which is typical if the network is sufficiently self-contained not to be subject to frequent latent-variable effects, then it behaves as a single Boolean network for long periods of time. As a result, it spends the vast majority of its time in attractors.

In most experimental situations, unless a situation has been created where time-course gene-expression measurements are taken following some stimulus to the system that drives it out of

its steady-state behavior, our assumption is that measurements (or at least almost all of them) are taken in the steady state. Under the assumption that we are sampling from the steady state, biological state stability leads to the further assumption that most of the steady-state probability mass is concentrated in the attractors and that real-world attractors are most likely to be singleton attractors consisting of one state. This means that our modeling applies to cells that have very short transient durations in relation to their durations in their steady states. If each cell in a family spends a strong majority of its time in a particular state, then because we are averaging a large number of cells (in the case of microarray data), one can expect with high probability that the observed measurements correspond to that state.

This steady-state assumption has two implications for inference. First, and most importantly, since data states are, with probability near one, attractor states, we would like them to be attractors in the model. According to Proposition 1 (below), this is accomplished with the proposed inference procedure. For small samples, it is very possible that sampling misses biological attractor states in the data; however, with large samples the likelihood grows for observing biological attractor states in the data, and therefore incorporating them in the model. This is precisely what one would expect in a learning paradigm. As for the converse of the first implication, while network design can result in nondata states being attractor states in the model, Propositions 2 and 4 show that in a number of cases, a nondata state will not be an attractor. In addition, as might be expected in a learning environment, avoiding nondata states as attractors depends on design generalization beyond that immediately implied by the data.

III. DATA-CONSISTENT OPERATOR DESIGN

Since the key to network design is designing the functions, we begin by treating data consistency in the general framework of designing a single Boolean operator on random inputs. Let $S = \{\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^m\}$ be the set of $m = 2^n$ vectors associated with the binary-valued observation variables X_1, X_2, \dots, X_n , and let Y be a target binary random variable to be predicted via X_1, X_2, \dots, X_n . A data set D composed of observations of the form (\mathbf{x}^k, y) is said to be *consistent* if $(\mathbf{x}^k, 0)$ and $(\mathbf{x}^k, 1)$ are not both in D . Going the other way, a random predictor-target pair (\mathbf{X}, Y) is said to be *consistent* with the data D if D is consistent relative to the observation pairs resulting from (\mathbf{X}, Y) . In such a case, there exists a predictor ψ for Y via \mathbf{X} , defined on S , possessing zero error on the data. ψ is said to be *consistent relative to D* . ψ may not be unique, since for any vector \mathbf{x}^k for which neither $(\mathbf{x}^k, 0)$ nor $(\mathbf{x}^k, 1)$ appears in the data, ψ can be defined arbitrarily.

Consider a random operator Ψ on S . Every realization ψ of Ψ defines a function on the random vector \mathbf{X} , or, equivalently, on the state space S endowed with the probability measure corresponding to \mathbf{X} . If D is any data set generated by ψ , then, *ipso facto*, ψ is consistent relative to D . The number of observations in the data corresponding to any vector \mathbf{x}^k is related to the probability of \mathbf{x}^k in S , not ψ . Specifically, letting $\nu(\mathbf{x}^k)$ denote the number of observations of \mathbf{x}^k in an arbitrary data set of

size N , then $E[\nu(\mathbf{x}^k)] = N\pi(\mathbf{x}^k)$, where $\pi(\mathbf{x}^k)$ is the probability of \mathbf{x}^k in S . In accordance with the empirical distribution of $\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^m$ for D , define the probability measure π_D on S by $\pi_D(\mathbf{x}^k) = \nu_D(\mathbf{x}^k)/N$, where $\nu_D(\mathbf{x}^k)$ is the number of observations of \mathbf{x}^k in D . Then $\pi_D(\mathbf{x}^k)$ is an estimate of $\pi(\mathbf{x}^k)$. A key to operator design is the following observation: if ψ_0 and ψ_1 are two realizations of Ψ and ψ_0 and ψ_1 agree on all vectors except \mathbf{x}^i , for which $\psi_0(\mathbf{x}^i) = 0$ and $\psi_1(\mathbf{x}^i) = 1$, then both pairs $(\mathbf{x}^i, 0)$ and $(\mathbf{x}^i, 1)$ may lie in a data set generated by ψ_0 and ψ_1 , but the data will be consistent for all $\mathbf{x}^k \neq \mathbf{x}^i$.

A. Operator Construction

Case 1: Suppose data set D has the property that there is a single vector \mathbf{x}^i possessing different Y values, and all other vectors possess a single Y value in D . Suppose there are $\nu_D(\mathbf{x}^i, 0)$ and $\nu_D(\mathbf{x}^i, 1)$ pairs $(\mathbf{x}^i, 0)$ and $(\mathbf{x}^i, 1)$, respectively. Define two functions ψ_0 and ψ_1 that agree on all vectors besides \mathbf{x}^i and are thereon defined by $\psi_0(\mathbf{x}^i) = 0$ and $\psi_1(\mathbf{x}^i) = 1$. Define the probability structure for the random function Ψ_D possessing the realizations ψ_0 and ψ_1 by $P(\Psi_D = \psi_a) = \nu_D(\mathbf{x}^i, a)/\nu_D(\mathbf{x}^i)$ for $a = 0, 1$. ψ_0 is consistent relative to the data set $D_i(0)$ consisting of the original data set D with all pairs $(\mathbf{x}^i, 1)$ removed, and ψ_1 is consistent relative to the data set $D_i(1)$ consisting of D with all pairs $(\mathbf{x}^i, 0)$ removed.

Case 2: Suppose the data set D has the property that there exist two vectors \mathbf{x}^i and \mathbf{x}^j possessing different Y values, and all other vectors possess a single Y value in D . Let there be $\nu_D(\mathbf{x}^i, 0)$, $\nu_D(\mathbf{x}^i, 1)$, $\nu_D(\mathbf{x}^j, 0)$, and $\nu_D(\mathbf{x}^j, 1)$ pairs $(\mathbf{x}^i, 0)$, $(\mathbf{x}^i, 1)$, $(\mathbf{x}^j, 0)$, and $(\mathbf{x}^j, 1)$, respectively. Define four functions ψ_{00} , ψ_{01} , ψ_{10} , and ψ_{11} that agree on all vectors besides \mathbf{x}^i and \mathbf{x}^j , and are thereon defined by $\psi_{00}(\mathbf{x}^i) = 0$, $\psi_{00}(\mathbf{x}^j) = 0$, $\psi_{01}(\mathbf{x}^i) = 0$, $\psi_{01}(\mathbf{x}^j) = 1$, $\psi_{10}(\mathbf{x}^i) = 1$, $\psi_{10}(\mathbf{x}^j) = 0$, $\psi_{11}(\mathbf{x}^i) = 1$, and $\psi_{11}(\mathbf{x}^j) = 1$. Define the probability structure for the random function Ψ_D possessing the realizations ψ_{00} , ψ_{01} , ψ_{10} , and ψ_{11} according to $P(\Psi_D = \psi_{ab}) = \nu_D(\mathbf{x}^i, a)\nu_D(\mathbf{x}^j, b)/\nu_D(\mathbf{x}^i)\nu_D(\mathbf{x}^j)$ for $a, b = 0, 1$. ψ_{ab} is consistent relative to the data set $D_{ij}(ab)$ consisting of the original data set D with all pairs $(\mathbf{x}^i, 1-a)$ and $(\mathbf{x}^j, 1-b)$ removed.

Case k: The preceding definition and probability structure can be inductively defined for any k vectors possessing different Y values, with all the other vectors possessing a single Y value. We say that the resulting random function is *order- k consistent* relative to the data set D .

We now state the basic theorem for consistent-data operator design.

Theorem: If the random function Ψ_D is order- k consistent relative to the set D , then 1) when restricted to any of its realizations, Ψ_D produces consistent data, 2) the estimate of the expected distribution of the data generated by Ψ_D using π_D in place of π agrees with the distribution of the data in D , and 3) the latter condition cannot be accomplished with less than 2^k functions, the number of realizations of Ψ_D .

Proof: We first prove the case 1. For a random data set D of size N generated by the random function Ψ_D , let $\eta(\mathbf{x}^i, 0)$ and $\eta(\mathbf{x}^i, 1)$ be the random variables giving the number of times \mathbf{x}^i is 0 and 1, respectively, in D . Since Ψ_D is designed from the given data set D and thereafter applied to random data sets, the

probability $P(\Psi_D = \psi_0)$ is fixed upon the design of Ψ_D and is independent of the probability of observing any particular state vector in D . Thus

$$\begin{aligned} E[\eta(\mathbf{x}^i, 0)] &= NP(\Psi_D(\mathbf{x}^i) = 0)\pi(\mathbf{x}^i) \\ &= N\pi(\mathbf{x}^i)[P(\Psi_D = \psi_0)P(\psi_0(\mathbf{x}^i) = 0) \\ &\quad + P(\Psi_D = \psi_1)P(\psi_1(\mathbf{x}^i) = 0)] \\ &= NP(\Psi_D = \psi_0)\pi(\mathbf{x}^i) \\ &= N\frac{\nu_D(\mathbf{x}^i, 0)}{\nu_D(\mathbf{x}^i)}\pi(\mathbf{x}^i). \end{aligned} \quad (1)$$

If we replace $\pi(\mathbf{x}^i)$ by its estimate $\pi_D(\mathbf{x}^i)$ based upon the data set D , then we obtain the estimate

$$\hat{E}[\eta(\mathbf{x}^i, 0)] = \nu_D(\mathbf{x}^i, 0) \quad (2)$$

of the expectation $E[\eta(\mathbf{x}^i, 0)]$. Equation (2) states that the estimate of the expectation of the number of times that \mathbf{x}^i has the label 0, based on the estimate π_D equals the number of times \mathbf{x}^i has the label 0 in the data. Similarly, $\hat{E}[\eta(\mathbf{x}^i, 1)] = \nu_D(\mathbf{x}^i, 1)$. For $k \neq i$, $\hat{E}[\eta(\mathbf{x}^k, 0)]$ is either 0 or $\nu_D(\mathbf{x}^k)$, depending on the common value of $\psi_0(\mathbf{x}^k)$ and $\psi_1(\mathbf{x}^k)$. Clearly, this could not have been accomplished by a single realization.

For case 2, for an arbitrary data set D of size N generated by Ψ_D , let $\eta(\mathbf{x}^i, 0)$, $\eta(\mathbf{x}^i, 1)$, $\eta(\mathbf{x}^j, 0)$, and $\eta(\mathbf{x}^j, 1)$ be random variables giving the number of times \mathbf{x}^i is 0, \mathbf{x}^i is 1, \mathbf{x}^j is 0, and \mathbf{x}^j is 1, respectively, in D . Then

$$\begin{aligned} E[\eta(\mathbf{x}^i, 0)] &= NP(\Psi_D(\mathbf{x}^i) = 0)\pi(\mathbf{x}^i) \\ &= N\pi(\mathbf{x}^i)[P(\Psi_D = \psi_{00})P(\psi_{00}(\mathbf{x}^i) = 0) \\ &\quad + P(\Psi_D = \psi_{01})P(\psi_{01}(\mathbf{x}^i) = 0) \\ &\quad + P(\Psi_D = \psi_{10})P(\psi_{10}(\mathbf{x}^i) = 0) \\ &\quad + P(\Psi_D = \psi_{11})P(\psi_{11}(\mathbf{x}^i) = 0)] \\ &= N\pi(\mathbf{x}^i)[P(\Psi_D = \psi_{00}) + P(\Psi_D = \psi_{01})] \\ &= N\pi(\mathbf{x}^i)\left(\frac{\nu_D(\mathbf{x}^i, 0)\nu_D(\mathbf{x}^j, 0)}{\nu_D(\mathbf{x}^i)\nu_D(\mathbf{x}^j)} + \frac{\nu_D(\mathbf{x}^i, 0)\nu_D(\mathbf{x}^j, 1)}{\nu_D(\mathbf{x}^i)\nu_D(\mathbf{x}^j)}\right). \end{aligned} \quad (3)$$

If we replace $\pi(\mathbf{x}^i)$ by its estimate $\pi_D(\mathbf{x}^i)$ based on the data set D , then we obtain the estimate

$$\hat{E}[\eta(\mathbf{x}^i, 0)] = \nu_D(\mathbf{x}^i, 0) \left(\frac{\nu_D(\mathbf{x}^j, 0)}{\nu_D(\mathbf{x}^j)} + \frac{\nu_D(\mathbf{x}^j, 1)}{\nu_D(\mathbf{x}^j)} \right) = \nu_D(\mathbf{x}^i, 0) \quad (4)$$

of the expectation $E[\eta(\mathbf{x}^i, 0)]$. Similarly, $\hat{E}[\eta(\mathbf{x}^i, 1)] = \nu_D(\mathbf{x}^i, 1)$, $\hat{E}[\eta(\mathbf{x}^j, 0)] = \nu_D(\mathbf{x}^j, 0)$, and $\hat{E}[\eta(\mathbf{x}^j, 1)] = \nu_D(\mathbf{x}^j, 1)$. For $k \notin \{i, j\}$, $\hat{E}[\eta(\mathbf{x}^k, 0)]$ is either 0 or $\nu_D(\mathbf{x}^k)$, depending on the common value of $\psi_0(\mathbf{x}^k)$ and $\psi_1(\mathbf{x}^k)$ for k . In sum, when restricted to either ψ_{00} , ψ_{01} , ψ_{10} , or ψ_{11} , the estimate of the expected distribution of the data, using the estimate π_D , agrees with the data distribution. This cannot be accomplished with less than four functions. Indeed, since any function must agree with the single value for vectors other than \mathbf{x}^i and \mathbf{x}^j , were there only three functions, these would be a subset of $\{\psi_{00}, \psi_{01}, \psi_{10}, \psi_{11}\}$ and there would still be four equations of the kind in (3). These would require solution with only three variables of kind $P(\Psi_D = \psi_{ab})$ instead of the four variables $P(\Psi_D = \psi_{00})$, $P(\Psi_D = \psi_{01})$, $P(\Psi_D = \psi_{10})$, and $P(\Psi_D = \psi_{11})$.

TABLE I
DATA SET AND PREDICTOR FUNCTIONS

(a)

| xyz | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
|--------|-----|-----|-----|-----|-----|-----|-----|-----|
| Counts | 4 | 0 | 6 | 6 | 2 | 6 | 0 | 4 |

(b)

| xy | Ψ_{00}^z | Ψ_{01}^z | Ψ_{10}^z | Ψ_{11}^z |
|------|---------------|---------------|---------------|---------------|
| 00 | 0 | 0 | 0 | 0 |
| 01 | 0 | 0 | 1 | 1 |
| 10 | 0 | 1 | 0 | 1 |
| 11 | 1 | 1 | 1 | 1 |

(c)

| xz | Ψ_{00}^y | Ψ_{01}^y | Ψ_{10}^y | Ψ_{11}^y |
|------|---------------|---------------|---------------|---------------|
| 00 | 0 | 0 | 1 | 1 |
| 01 | 1 | 1 | 1 | 1 |
| 10 | 0 | 0 | 0 | 0 |
| 11 | 0 | 1 | 0 | 1 |

(d)

| yz | Ψ_{00}^x | Ψ_{01}^x | Ψ_{10}^x | Ψ_{11}^x |
|------|---------------|---------------|---------------|---------------|
| 00 | 0 | 0 | 1 | 1 |
| 01 | 1 | 1 | 1 | 1 |
| 10 | 0 | 0 | 0 | 0 |
| 11 | 0 | 1 | 0 | 1 |

The proof for case 2 extends directly to any order k , albeit, with increased notational complexity. ♦

The third part of the theorem is critical because it says that the constructed random function solves the problem with which we are concerned in an optimal way relative to minimizing the number of its realizations. By addressing data inconsistency under the assumption that inconsistencies result from the data arising from a random function of the state space, optimal operator design becomes one of finding the realizations of a random function and the probability mass on those realizations so that the resulting random operator best fits the data relative to the expectation of its output and does so using a minimal number of randomizations. In effect, we have presented an algorithm to solve this optimization problem.

To illustrate the design methodology, we consider two predictor variables X and Y , the target variable Z , and the data in Table I(a), where *count* is the number of times xyz is observed in the data. In the data, the observations $xy = 00$ and $xy = 11$ are consistent, whereas $xy = 01$ and $xy = 10$ are inconsistent. Hence, four functions are required for prediction of Z . These are shown in Table I(b). The selection probabilities are $P(\Psi_D = \psi_{00}^z) = 1/8$, $P(\Psi_D = \psi_{01}^z) = 3/8$, $P(\Psi_D = \psi_{10}^z) = 1/8$, and $P(\Psi_D = \psi_{11}^z) = 3/8$. Notice what happens if we change the count of 111 to 0. The number of functions remains 4; however, the data do not provide inference of $\psi_{ab}^z(11)$. Therefore, it must be decided by generalization. We will return to this issue in the context of PBNs.

It is important in understanding Theorem 1 to recognize that the third part of the theorem refers to the second part, that is, the number of realizations required to accomplish the distributional requirement is 2^k . If we were not concerned with the expected concordance between the expected distribution of data generated by the random function Ψ_D and the distribution of the data in D , then we would need only two realizations to achieve consistent design. To see this, suppose in D there exist m vectors, $\mathbf{x}^{i1}, \mathbf{x}^{i2}, \dots, \mathbf{x}^{im}$ possessing different Y values and for any other vector \mathbf{x} there is a single observed Y value $a_{\mathbf{x}}$. Define $\psi_0(\mathbf{x}^{ij}) = 0$ and $\psi_1(\mathbf{x}^{ij}) = 1$ for $j = 1, 2, \dots, m$, and $\psi_0(\mathbf{x}) = \psi_1(\mathbf{x}) = a_{\mathbf{x}}$ for any \mathbf{x} . These two realizations can account for all of the inconsistencies; however, the expected dis-

tribution of data generated by Ψ_D will not be concordant with the data distribution in D .

IV. DATA-CONSISTENT DESIGN OF PROBABILISTIC BOOLEAN NETWORKS

Adaptation of consistent-data design to PBNs is straightforward, but there are some issues regarding generalization and attractors that need to be addressed. Consider designing a PBN from a data set of elements from the set $\mathcal{S} = \{\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^m\}$ of $m = 2^n$ binary vectors. For a PBN, each gene is taken as a function of the remaining genes. Consistent-data design is applied to each gene in turn. A network function for the PBN is defined by taking one function for each gene. For a network with n genes, if there are m_k functions for gene k , then there are $m_1 m_2 \dots m_n$ network functions. Each network function defines a *context* of the network in which the data are consistent. This means that, so long as a network is in the context of a network function, it will generate consistent data. Each context defines a standard (constituent) Boolean network. The selection probability of a network function is the product of the selection probabilities for the individual functions composing the network function. Note that, as with other proposed PBN design methods, the perturbation and switching probabilities cannot be estimated from the steady-state data. The implementation of the design algorithm is outlined in the Appendix .

To illustrate, for the data of Table I(a), we have three function sets shown in parts (b), (c), and (d). For both xz and yz , the observations 01 and 10 are consistent, whereas 00 and 11 are inconsistent. The PBN has 64 network functions determining the same number of contexts.

Attractors are key to understanding a PBN. Relative to attractors, there is a fundamental difference between data states and nondata states. Before giving formal definitions, we consider some situations.

For the PBN resulting from the data of Table I(a), consider the data state 000. It is a singleton attractor for any context $(\psi_{ab}^x, \psi_{cd}^y, \psi_{ef}^z)$ in which $\psi_{ab}^x(00) = \psi_{cd}^y(00) = \psi_{ef}^z(00) = 0$. There are $2 \times 2 \times 4 = 16$ such contexts (out of a total of 64 contexts). Each of the six data states is a singleton attractor for

TABLE II
PREDICTOR FUNCTIONS

| yz | ψ^x | xz | ψ^y | xy | ψ^z |
|----|----------|----|----------|----|----------|
| 00 | 0 | 00 | 0 | 00 | 0 |
| 01 | x | 01 | x | 01 | x |
| 10 | x | 10 | x | 10 | x |
| 11 | x | 11 | x | 11 | x |

some number of contexts. On the contrary, consider the nondata state 001. Since $\psi_{ab}^x(01) = 1$, $\psi_{cd}^y(01) = 1$, and $\psi_{ef}^z(00) = 0$ for any ab, cd , and ef , $001 \rightarrow 110$, its complement, in every context (the arrow denoting transition). 110 is also a nondata state, and $110 \rightarrow 001$, its complement, in every context. Hence, $\{110, 001\}$ is a two-state attractor cycle in every context. Note that if 110 were a data state, then it would be a singleton attractor in some contexts and the nondata state 001 would not be in an attractor cycle in those contexts.

Now, consider a data set with a single data state, 000. All predictor–target pairs are consistent relative to the data, and only one function is required for each gene (Table II). Each function requires three of its four values to be determined by generalization (arbitrarily relative to the data). The result is a Boolean network in which 000 is a singleton attractor.

We say that a nondata state $\mathbf{x} = x_1x_2 \cdots x_n$ is *partially mapped* by the data if there exists at least one subvector, $x_1x_2 \cdots x_{k-1}x_{k+1} \cdots x_n$, which has been observed in the data, so that there exists a function ψ^k for x_k for which $\psi^k(x_1x_2 \cdots x_{k-1}x_{k+1} \cdots x_n)$ has been determined by the data, not by generalization. For the single observation 000 and the Boolean network of Table II, the states 001, 010, and 100 are partially mapped. A nondata state $\mathbf{x} = x_1x_2 \cdots x_n$ is *fully unmapped* by the data if no subvector $x_1x_2 \cdots x_{k-1}x_{k+1} \cdots x_n$ has been observed in the data. For the single observation 000, the states 011, 101, 110, and 111 are fully unmapped. A nondata state is *fully mapped* if all subvectors have been observed in the data, which was the case for 110 in the data of Table I(a).

Continuing with the single observation 000 and the network of Table II, for which the nondata states 001, 010, and 100 are partially mapped by the data, the single network function yields $001 \rightarrow xx0$, $010 \rightarrow x0x$, and $100 \rightarrow 0xx$. The actual transitions depend on the generalization; nevertheless, these partially determined nondata states are not singleton attractors. The remaining data states, 011, 101, 110, and 111, are fully unmapped by the data, so that their transitions depend totally on generalization, which can yield singleton nondata attractors. In this example, 011 becomes a singleton attractor if and only if we define $\psi^{xx}(11) = 0$, $\psi^{yy}(01) = 1$, and $\psi^{zz}(01) = 1$; 101 becomes a singleton attractor if and only if we define $\psi^{xx}(01) = 1$, $\psi^{yy}(11) = 0$, and $\psi^{zz}(10) = 1$; 110 becomes a singleton attractor if and only if we define $\psi^{xx}(10) = 1$, $\psi^{yy}(10) = 1$, and $\psi^{zz}(11) = 0$; and 111 becomes a singleton attractor if and only if we define $\psi^{xx}(11) = 1$, $\psi^{yy}(11) = 1$, and $\psi^{zz}(11) = 1$. Note that 110 and 111 cannot simultaneously be singleton attractors, nor can 011 and 111 simultaneously be singleton attractors.

We now provide some formal propositions.

Proposition 1: A data state is a singleton attractor in at least one context.

Proof: If $\mathbf{x} = x_1x_2 \cdots x_n$ is a data state, then for each gene x_k , there is at least one function ψ^k inferred from the data for which $\psi^k(x_1x_2 \cdots x_{k-1}x_{k+1} \cdots x_n) = x_k$. \mathbf{x} is a singleton attractor for the context $\{\psi^1, \psi^2, \dots, \psi^n\}$. \blacklozenge

Proposition 2: A fully or partially mapped nondata state is not a singleton attractor in any context.

Proof: If $\mathbf{x} = x_1x_2 \cdots x_n$ is a fully or partially mapped nondata state, then there exists a gene x_k determined from the data relative to $x_1x_2 \cdots x_n$. Suppose $x_1x_2 \cdots x_n \rightarrow x_1x_2 \cdots x_n$ in some context $\{\psi^1, \psi^2, \dots, \psi^n\}$. Then $x_k = \psi^k(x_1x_2 \cdots x_{k-1}x_{k+1} \cdots x_n)$. Since this relationship has been determined from the data, $x_1x_2 \cdots x_n$ must be a data state, which is a contradiction. \blacklozenge

Proposition 3: If a nondata state and its complement are both fully mapped, then they form a two-state attractor cycle in every context.

Proof: If $\mathbf{x} = x_1x_2 \cdots x_n$ is fully mapped, then in any context $\{\psi^1, \psi^2, \dots, \psi^n\}$,

$$\mathbf{x} \rightarrow \psi^1(x_2x_3 \cdots x_n)\psi^2(x_1x_3 \cdots x_n) \cdots \psi^n(x_1x_2 \cdots x_{n-1}).$$

It must be that $\psi^k(x_1x_2 \cdots x_{k-1}x_{k+1} \cdots x_n) = x_k^c$, since otherwise the fact that $x_1x_2 \cdots x_{k-1}x_{k+1} \cdots x_n$ has been observed in the data, which it has not. Hence, $\mathbf{x} \rightarrow \mathbf{x}^c$. The same argument applied to \mathbf{x}^c shows that $\{\mathbf{x}, \mathbf{x}^c\}$ is a two-state attractor. \blacklozenge

Proposition 4: Generalization can always make a given fully unmapped nondata state be or not be a singleton attractor.

Proof: If $\mathbf{x} = x_1x_2 \cdots x_n$ is a fully unmapped nondata state, then there are no data-determined functions $\psi^k(x_1x_2 \cdots x_{k-1}x_{k+1} \cdots x_n)$. To make \mathbf{x} an attractor, define $\psi^k(x_1x_2 \cdots x_{k-1}x_{k+1} \cdots x_n) = x^k$ for all k ; to make \mathbf{x} not a singleton attractor, define ψ^k in any other manner. \blacklozenge

An attractor composed solely of nondata states will be called an *artificial attractor*. As noted previously, it may not be possible to make two fully unmapped nondata states into singleton attractors. According to Proposition 2, artificial singleton attractors are fully unmapped. Every singleton attractor is either a data state or an artificial attractor. According to Proposition 3, if a nondata state and its complement are both fully mapped, then they form an artificial two-state attractor cycle in every context.

The state transitions for a PBN produce an ergodic Markov chain possessing a steady-state distribution. When a PBN is designed from non-time-course data, the implicit assumption is that the data have been obtained in the steady state. This means that the state transitions of the designed PBN do not correspond to transitions in biological time but to synthetic (mathematical) time. Hence, there is no direct correspondence between transient states of the PBN and data states. There should be, however, correspondence between steady-state behavior and the data states. Since we expect network switching to be infrequent in a real system, most of the steady-state mass should belong to the attractors, and since the data have been drawn from the steady state, we would expect it to be highly likely that the data states are attractors. In this sense, Proposition 1 provides support for the context-switching model. Proposition 2 is also encouraging relative to steady-state and data distribution correspondence.

Proposition 3, while not encouraging, posits the strong requirement that a nondata state be fully mapped. Proposition 4 only asserts existence and says nothing about the consequences of a reasonable generalization.

V. REFLECTIONS ON STANDARD AND CONTEXTUAL DESIGN

Whereas a Boolean network is assured for a single observed data state, two data states may require a PBN. At the other extreme, only a Boolean network is required for consistency if the data states are 001, 010, 100, and 111, and all four states would be singleton attractors. The issue of the number of contexts is related to the deeper issue of learning predictors for a dynamical system from steady-state data [22].

Consider a three-gene Boolean network with vectors xyz and data set $\{000, 001\}$. If we observe 000 more often than 001, why define the prediction $\psi^z(00) = 0$? After all, in the real system, 000 might transition to another state, and therefore $xy = 00$ may predict z being 1. For instance, if in the actual system $000 \rightarrow 001$, then would it not be better to predict z by $\psi^z(00) = 1$? Perhaps it would be, but we lack the dynamical data to make such a determination. The original use of prediction for gene expression was to measure multivariate gene interaction [9]: based on the data, if $xy = 00$ is observed in the steady state, then what is the best prediction for z . If we observe 000 in the data more often than 001, then the best prediction on observing $xy = 00$ in a future observation would be to predict $z = 0$. This approach has been adopted for network inference, and represents a kind of generalization because a network involves dynamical behavior. Nonetheless, under the assumption that the data come from the steady state, and assuming that when in the steady state the network spends the great majority of its time in its attractors, when choosing between the singleton attractor 000 [$\psi^z(00) = 0$] and the singleton attractor 001 [$\psi^z(00) = 1$], a majority decision based on the data indicates the singleton attractor 000.

The situation becomes more flexible with the use of PBNs. We reconsider the three-gene situation with data set $\{000, 001\}$. At first glance, it may appear that we have three possibilities: 1) 000 and 001 compose an attractor cycle in the same Boolean network; 2) they are singleton attractors in a single Boolean network; 3) or they are singleton attractors in different contexts. However, the first situation is not possible because $000 \rightarrow 001$ requires $\psi^z(00) = 1$, and $001 \rightarrow 000$ requires $\psi^z(00) = 0$. As for the second possibility, it involves the choice just discussed. If we choose $\psi^z(00) = 0$, then to have the network remain in an attractor, we must have $\psi^x(00) = 0$ and $\psi^y(00) = 0$, in which case 000 is an attractor and 001 is a transient state; if we choose $\psi^z(00) = 1$, then to have the network remain in an attractor, we must have $\psi^x(01) = 0$ and $\psi^y(01) = 0$, in which case 001 is an attractor and 000 is a transient state. Thus, we choose $\psi^z(00)$ based on the majority decision. The third possibility occurs by using context: 000 and 001 are singleton attractors in different contexts, in which case we have $\psi^z(00) = 0$ in one context and $\psi^z(00) = 1$ in the other, with all conflicts being resolved. Note that this same analysis applies whenever there are two data points and they differ only for a single gene.

For another situation, consider the data set $\{000, 111\}$. The same three apparent possibilities appear, but now they are all truly possible. We could have the cycle $000 \leftrightarrow 111$. This would not create a conflict in any predictor definitions: $\psi^x(00) =$

$\psi^y(00) = \psi^z(00) = 1$ and $\psi^x(11) = \psi^y(11) = \psi^z(11) = 0$. They could also form two singleton attractors in the same Boolean network, with $\psi^x(00) = \psi^y(00) = \psi^z(00) = 0$ and $\psi^x(11) = \psi^y(11) = \psi^z(11) = 1$. Finally, they could be singleton attractors in different contexts of a PBN. Using either noncontextual or contextual design, they can appear as singleton attractors in a single Boolean network. Were the data actually reflective of a cycle in a real regulatory system, then the inference would be erroneous. Because steady-state data are insufficient to infer dynamics, a learning assumption has been made (here and in the past) that favors short cycles over long, in this case favoring singleton attractors. Moreover, the number of contexts is minimized by assuming them to be singleton attractors in a Boolean network. Note that the same analysis applies whenever there are two data points and they differ by more than a single gene.

To help clarify the issue, we define two states to be *neighbors* if they differ by a single gene. A data state is said to be *isolated* if it has no neighbors in the data and *nonisolated* otherwise. If two data states are neighbors, as are 000 and 001, then they require two contexts to avoid data inconsistency. Since context selection depends on the data frequencies, the frequencies of 000 and 001 affect the resulting PBN probabilities. On the other hand, if a data state is isolated, as is the case of 000 for the data set $\{000, 110, 111\}$, then it does not generate contexts. When a data state is isolated, its frequency in the data does not affect PBN probabilities.

The number of constituent networks is determined by how inconsistencies appear in the data, not the number of states appearing in the data. To illustrate, a PBN with data states 000, 001, 010, and 011 has four, four, and one function(s) for x , y , and z , respectively, for a total of 16 contexts. A PBN with data states 010, 100, 101, and 110 has two, two, and two functions for x , y , and z , respectively, for a total of eight contexts.

VI. FILTERING

We have addressed data inconsistency from the perspective of biological context. The context problem is inherent to an open system, one that receives inputs from external variables that affect the system output. We have focused on system design, and as with all inference procedures, the design precision is affected by noise. Data-consistent design begins with binary state vectors (profiles), under the assumption of previous filtering, normalization, and quantization. Generally speaking, it is difficult to model the impact of various noise sources on high-level data analysis algorithms, the central problem being the large number of sources of variance inherent in the process of making these measurements—for instance, using cDNA microarrays. In many statistical papers, the measured gene expression data are assumed to have multiple noise sources: sample preparation, labeling, hybridization, background fluorescence, different arrays, fluorescent dyes, and different printing locations. As with any high-level processing, network design is influenced by lower level processing. In our case, noisy observation vectors can negatively affect design because our aim is to have the steady-state distribution of the designed network agree with the empirical distribution. In particular, noisy observations can result in spurious contexts.

Relative to data-consistent design, there is a more fundamental issue than observation noise pertaining to the number of

TABLE III
EXPRESSION PROFILES FOR MELANOMA

| Profile # | GENES | | | | | | | | | | counts |
|-----------|-------|-------|-------|-------|-------|--------|-----------|------|-------|-------|--------|
| | RET-1 | HADHB | MMP-3 | S100P | pirin | MART-1 | synuclein | STC2 | PHO-C | WNT5A | |
| 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 2 |
| 2 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| 3 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 4 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 2 |
| 5 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| 6 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 7 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 8 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 2 |
| 9 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 10 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 11 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 12 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 13 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 2 |
| 14 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 8 |
| 15 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| 16 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 17 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 18 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| 19 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 20 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |

contexts generated by the data, namely, sample heterogeneity. In many cases microarray data are obtained from heterogeneous cell populations, in particular, when tumor samples are analyzed. In fact, the entire issue of contextual modeling relates to data heterogeneity: the data relating to a specific set of genes composing a network derive from heterogeneous sources because each source is conditioned by conditions external to the network. This heterogeneity affects model design. If in the case of a Bayesian network the conditional probability of a gene given its parents is estimated across sample data arising from heterogeneous subpopulations, then the conditioning is in effect averaged across different data sources and the resulting conditional probability does not specifically apply to any of the subpopulations. The same can be said of PBN (or PGRN) design using coefficients of determination computed relative to the full sample. It is precisely our desire to make PGRN design specific to the subpopulations (contexts) arising from external latent variables that has motivated data-consistent design. Consequently, when there is excessive sample heterogeneity there can be an extraordinarily large number of contexts.

To reduce the large number of contexts arising from excessive data heterogeneity (or from observation noise) we can filter the binary profiles. Specifically, if two profiles are very close, we can join them, thereby identifying their individual contexts. Since we lack a heterogeneity model it is impossible to optimally derive this identification filter and we therefore take an intuitive approach, which has generally been how data filtering has proceeded in the context of microarrays. The filter is applied in the following manner: 1) if a profile is observed more than once in the data, then it remains invariant; 2) if a profile appears only once in the data and it is within Hamming distance 1 of a repeated profile, then it is identified with the repeated profile; 3) if an unrepeated profile is not within Hamming distance 1 of a repeated profile, then it is left invariant. The idea is straightforward. Singleton profiles that are almost identical to repeated profiles are assumed to result from either noise or statistically less important contexts very close to more important contexts. In practice, one can choose to use or not use the Hamming filter.

TABLE IV
FILTERED EXPRESSION PROFILES

| $g_1 g_2 \dots g_{10}$ | counts |
|------------------------|--------|
| 1001111101 | 2 |
| 1101110000 | 1 |
| 1010111101 | 1 |
| 1001111100 | 2 |
| 0101110011 | 1 |
| 1011111111 | 1 |
| 0101111101 | 1 |
| 0100110001 | 3 |
| 0001110001 | 1 |
| 0110100011 | 1 |
| 0100100011 | 1 |
| 1010101010 | 1 |
| 1011101110 | 2 |
| 1010001110 | 9 |
| 0010110000 | 1 |
| 0101010010 | 1 |
| 0011011010 | 1 |
| 0010001010 | 1 |

VII. MELANOMA NETWORK

We apply the contextual-design method to a gene network that has served as a model to study the external control of gene regulatory networks, in particular, for the regulatory avoidance of metastatic melanoma—for instance, in [12], where the context-sensitive PBN was constructed by the Bayesian connectivity approach.

The ten genes considered here were first identified in a study concerned with the feasibility of producing Markovian networks whose stationary distributions closely reflect the data [10]. The chosen genes arose from data in a study of metastatic melanoma [2]. In this study, the abundance of messenger RNA for the gene WNT5A was found to be highly discriminating between cells with properties typically associated with high metastatic competence versus those with low metastatic competence. These findings were validated and expanded in a second study [20]. In this study, experimentally increasing the levels of the Wnt5a protein secreted by a melanoma cell line via genetic engineering methods directly altered the metastatic competence of that cell

TABLE V
ATTRACTORS FOR MELANOMA NETWORK

| | | | | | | | | | | | | |
|----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Context 1 | 113 | 138 | 176 | 218 | 305 | 338 | 371 | 381 | 419 | 637 | 654 | 682 |
| | 701 | 702 | 750 | 767 | 880 | | | | | | | |
| Context 2 | 113 | 138 | 176 | 218 | 291 | 305 | 338 | 371 | 381 | 637 | 654 | 682 |
| | 701 | 702 | 750 | 767 | 880 | | | | | | | |
| Context 3 | 113 | 138 | 176 | 218 | 305 | 338 | 371 | 381 | 419 | 636 | 654 | 682 |
| | 701 | 702 | 750 | 767 | 880 | | | | | | | |
| Context 4 | 113 | 138 | 176 | 218 | 291 | 305 | 338 | 371 | 381 | 636 | 654 | 682 |
| | 701 | 702 | 750 | 767 | 880 | | | | | | | |
| All Attractors | 113 | 138 | 176 | 218 | 291 | 305 | 338 | 371 | 381 | 419 | 636 | 637 |
| | 654 | 682 | 701 | 702 | 750 | 767 | 880 | | | | | |
| Data Profiles | 113 | 138 | 176 | 218 | 291 | 305 | 338 | 371 | 381 | 419 | 636 | 637 |
| | 654 | 682 | 701 | 750 | 767 | 880 | | | | | | |

as measured by the standard *in vitro* assays for metastasis. A further finding of interest was that an intervention that blocked the Wnt5a protein from activating its receptor, the use of an antibody that binds Wnt5a protein, could substantially reduce Wnt5a's ability to induce a metastatic phenotype. This suggests a study of control based on interventions that alter the contribution of the WNT5A gene's action to biological regulation, since the available data suggest that disruption of this influence could reduce the chance of a melanoma metastasizing. The control objective is to externally downregulate the WNT5A gene, because WNT5A ceasing to be downregulated is strongly predictive of the onset of metastasis. Owing to computational issues relating to dynamic programming, in the control studies only seven of the original ten genes were used; here, we use the full set of ten to demonstrate network design: RET-1, HADHB, MMP-3, S100P, pirin, MART-1, synuclein, STC2, PHO-C, and WNT5A.

In the original expression study, 31 expression profiles were found for the ten genes, with some profiles repeated. Table III lists the 20 distinct profiles, along with their counts. As discussed previously, when we design a PBN, we must generalize the unspecified entries in the truth table. Here we do so by majority vote: if half or more of the entries have value 1, then set all the unspecified entries to 1; otherwise set them to 0. If we design a PBN based on the 20 profiles without any filtering, the resulting PBN has 128 contexts. The Hamming-distance filter yields 18 distinct profiles. They and their counts are shown in Table IV. Under the Hamming-distance filter and majority-vote generalization, the designed PBN has four contexts. Table V lists the attractors in each context and the data profiles (in decimal form for convenience). As must be the case, the PBN captures all the data profiles as attractors. There is only one spurious attractor point, 702.

VIII. CONCLUSION

This paper provides an inference procedure for PBNs in which the network contains contexts to model the data in such a way that it is consistent for each context. The intent is to view genomic regulation as deterministic (up to gene perturbation), with data inconsistencies due to latent variables. A key property is that every data state must be an attractor in at least one context, which is concordant with the assumption that the data states are attractor states for the real biological system. The dynamics depend on generalization. This is to be expected since the inference problem is an ill-posed inverse problem owing to a lack of dynamical data. The attractors constrain the dynamical behavior but do not determine it. Future work will

concentrate on the critical issue of generalization. Given a set of prior network properties postulated in accord with biological considerations, the aim will be to construct generalizations that yield networks possessing the desired properties.

Of particular importance is the manner in which generalization affects network connectivity. Whereas it is often assumed in PBN design that connectivity is limited and this limitation is imposed on design, the theory in the present paper depends on the possibility of full connectivity. We refer to this possibility because once the realizations are determined they can be reduced so that they only involve essential variables, thereby reducing the connectivity. The degree to which the connectivity is reduced by logic reduction depends on the generalization. Going further, one might at the outset choose to limit the connectivity. Prior limitation might make data-consistent design impossible; however, one might try to achieve close-to-data-consistent design, where the closeness is based on some objective criterion. These considerations lead to two areas of ongoing research: 1) posing a suitable definition of connectivity minimization and developing efficient algorithms to select a generalization minimizing connectivity and 2) defining an appropriate probabilistic criterion for approximate data consistency and developing efficient algorithms to optimize design relative to the criterion.

The approach of constructing generalizations that yield networks possessing the desired properties is inevitable because building a dynamical model from steady-state data is a kind of overfitting [22], especially when data are limited. This is why we view a designed network as providing a regulatory structure consistent with observed steady-state behavior. Given our main interest is in steady-state behavior, this is reasonable in that we are trying to understand dynamical regulation corresponding to observed steady-state behavior—say, for the purpose of developing control strategies. Placing constraints on the designed network or optimizing design relative to conditions such as connectivity not only produces networks with desirable properties, it also reduces the number of possible generalizations, which can be very large when data are limited. The salient point here is that many generalizations will lead to networks not possessing the required properties and therefore they are not allowed, thereby reducing the extent to which the inverse problem is ill-posed. Nonetheless, unless the constraints are sufficiently strong to produce a unique solution, there will still be a collection of networks concordant with the data and the constraints. It is for this reason that we are investigating *robust* control strategies that, while not necessarily optimal relative to a given network, provide beneficial intervention across the family of networks in the solution space.

We close with the epistemological question: How is the obtained PBN, or any such high-level system model, related to a real biological system? We comment only briefly, referring to [6] for a more comprehensive account. When considering models of genomic regulation, one might say that there is a decision-making layer and a physical (chemical) layer. PBNs, and other such functional networks, provide representation at the decision-making layer. Predictive logical relations correspond to changes in the continuous data related to the up- and downregulated character of the genes involved [13]. Since a living system is of necessity an information processing system, *ipso facto*, a genomic network corresponding specifically to control information is biological. Indeed, the desire to understand information processing within the cell is a salient motivation for network construction [4]. Dougherty and Braga-Neto [6] write, “Regarding reality, the fact that a complete biochemical description of cellular activity would likely produce the corollary description of the information processing system does not denigrate the reality of the latter.” Poincaré [14] has remarked: “What we call ‘objective reality’ is, strictly speaking, that which is common to several thinking beings and might be common to all; this common part, we shall see, can only be the harmony expressed by mathematical laws.”

APPENDIX

SUMMARY OF THE DESIGN ALGORITHM

Step 1: Let the data set be $D = \{\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^{m_D}\} \subseteq S$, $m_D \leq m$.

Step 2: Let g_i be the target gene. If there exist l_i pairs of data, $\mathbf{x}^{k_{i0}}, \mathbf{x}^{k_{i1}}, \dots, \mathbf{x}^{k_{i,l_i-1}}, \mathbf{x}^{k_{i,l_i}}$, such that the vectors in each pair differ only on the value of target gene, namely

$$\begin{aligned} \mathbf{x}^{k_{j0}} \\ &= [x_{k_j 1}, x_{k_j 2}, \dots, x_{k_j (i-1)}, 0, x_{k_j (i+1)}, \dots, x_{k_j n}] \\ \mathbf{x}^{k_{j1}} \\ &= [x_{k_j 1}, x_{k_j 2}, \dots, x_{k_j (i-1)}, 1, x_{k_j (i+1)}, \dots, x_{k_j n}]. \end{aligned}$$

$j = 1, 2, \dots, l_i$, then $m_i = 2^{l_i}$ functions can be defined for g_i , namely

$$g_i = \varphi_j^i(g_1, \dots, g_{i-1}, g_{i+1}, \dots, g_n),$$

$j = 1, 2, \dots, m_i$. Each function is assigned a probability

$$P\{\varphi_j^i\} = \prod_{h=1}^{l_i} p_{hb}^{(i)}$$

where $p_{hb}^{(i)} = \nu(\mathbf{x}^{k_{hb}}) / [\nu(\mathbf{x}^{k_{h0}}) + \nu(\mathbf{x}^{k_{h1}})]$, $b = 0$ if φ_j^i is consistent with $\mathbf{x}^{k_{h0}}$, $b = 1$ if it is consistent with $\mathbf{x}^{k_{h1}}$, and $P\{\varphi_1^i\} + \dots + P\{\varphi_{m_i}^i\} = 1$. Apart from the l_i pairs, φ_j^i is consistent with the rest of the data in D . If $l_i = 0$, then only one function is defined and it is consistent with all data in D .

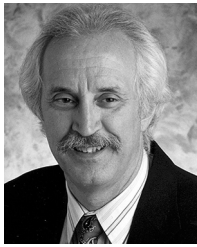
Step 3: Apply Step 2 to each gene in turn. By choosing a function for each gene and making all possible combinations, we obtain $m_1 m_2 \dots m_n$ network functions and associated selection probabilities, namely, there are $r = m_1 m_2 \dots m_n$ contexts.

Remark: As seen from the algorithm, the size of (truth table of) Boolean functions is determined by 2^n ($n =$ number of genes), and there are altogether $\sum_{i=1}^n m_i$ Boolean functions. Therefore, the complexity of the contextual design is $O(2^n \sum_{i=1}^n m_i)$; it is noteworthy that $\sum_{i=1}^n m_i$ depends primarily on the relations among data but cannot be solely accounted for by either the gene number n or data set size m_D alone.

REFERENCES

- [1] T. Akutsu, S. Miyano, and S. Kuhara, “Inferring qualitative relations in genetic networks and metabolic pathways,” *Bioinformatics*, vol. 16, pp. 727–734, 2000.
- [2] M. Bittner, “Molecular classification of cutaneous malignant melanoma by gene expression profiling,” *Nature*, vol. 406, pp. 536–540, 2000.
- [3] M. Brun, E. R. Dougherty, and I. Shmulevich, “Steady-state probabilities for attractors in probabilistic Boolean networks,” *Signal Process.*, vol. 85, no. 10, pp. 1993–2013, 2005.
- [4] E. Davidson, “A genomic regulatory network for development,” *Science*, vol. 295, pp. 1669–1678, 2002.
- [5] E. R. Dougherty, M. Bittner, Y. Chen, S. Kim, K. Sivakumar, J. Barrera, P. Meltzer, and J. Trent, “Nonlinear filters in genomic control,” in *Proc. IEEE-EURASIP Workshop Nonlinear Signal ImageProcessing*, 1999, pp. 10–15.
- [6] E. R. Dougherty and U. M. Braga-Neto, “Epistemology of computational biology: Mathematical models and experimental prediction as the basis of their validity,” *Biol. Syst.*, vol. 14, no. 1, pp. 65–90, 2005.
- [7] S. Huang, “Gene expression profiling, genetic networks, and cellular states: An integrating concept for tumorigenesis and drug discovery,” *Mol. Med.*, vol. 77, pp. 469–480, 1999.
- [8] S. A. Kauffman, *The Origins of Order: Self-Organization and Selection in Evolution*. New York: Oxford Univ. Press, 1993.
- [9] S. Kim, E. R. Dougherty, M. L. Bittner, Y. Chen, K. Sivakumar, P. Meltzer, and J. M. Trent, “A general framework for the analysis of multivariate gene interaction via expression arrays,” *Biomed. Opt.*, vol. 5, pp. 411–424, 2000.
- [10] S. Kim, H. Li, Y. Chen, N. Cao, E. R. Dougherty, M. L. Bittner, and E. B. Suh, “Can Markov chain models mimic biological regulation?,” *Biol. Syst.*, vol. 10, pp. 337–357, 2002.
- [11] H. Lahdesmaki, I. Shmulevich, and O. Yli-Harja, “On learning gene regulatory networks under the Boolean network model,” *Mach. Learn.*, vol. 52, pp. 147–167, 2003.
- [12] R. Pal, A. Datta, M. L. Bittner, and E. R. Dougherty, “Intervention in context-sensitive probabilistic Boolean networks,” *Bioinformatics*, vol. 21, pp. 1211–1218, 2005a.
- [13] R. Pal, A. Datta, A. J. Fornace, M. L. Bittner, and E. R. Dougherty, “Boolean relationships among genes responsive to ionizing radiation in the NCI 60 ACDS,” *Bioinformatics*, vol. 21, pp. 1542–1549, 2005b.
- [14] H. Poincaré, *The Foundations of Science*. Lancaster, PA: Science Press, 1946.
- [15] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang, “Probabilistic Boolean networks: A rule-based uncertainty model for gene regulatory networks,” *Bioinformatics*, vol. 18, pp. 261–274, 2002a.
- [16] I. Shmulevich, E. R. Dougherty, and W. Zhang, “Gene perturbation and intervention in probabilistic Boolean networks,” *Bioinformatics*, vol. 18, pp. 1319–1331, 2002b.
- [17] I. Shmulevich and W. Zhang, “Binary analysis and optimization-based normalization of gene expression data,” *Bioinformatics*, vol. 18, pp. 555–565, 2002c.
- [18] C.-H. Yuh, H. Bolouri, and E. H. Davidson, “Genomic cis-regulatory logic: Experimental and computational analysis of a sea urchin gene,” *Science*, vol. 279, pp. 1896–1902, 1998.
- [19] E. P. Van Someren, L. F. A. Wessels, E. Backer, and M. J. T. Reinders, “Multi-criterion optimization for genetic network modeling,” *Signal Process.*, vol. 83, pp. 763–775, 2003.

- [20] A. T. Weeraratna, Y. Jiang, G. Hostetter, K. Rosenblatt, P. Duray, M. L. Bittner, and J. M. Trent, "WNT5A signaling directly affects cell motility and invasion of metastatic melanoma," *Cancer Cell*, vol. 1, pp. 279–288, 2002.
- [21] X. Zhou, X. Wang, and E. R. Dougherty, "Binarization of microarray data based on a mixture model," *Molecul. Cancer Therap.*, vol. 2, pp. 679–684, 2003.
- [22] X. Zhou, X. Wang, R. Pal, I. Ivanov, M. L. Bittner, and E. R. Dougherty, "A Bayesian connectivity-based approach to constructing probabilistic gene regulatory networks," *Bioinformatics*, vol. 20, pp. 2918–2927, 2004.



Edward R. Dougherty (M'05) received the Ph.D. degree in mathematics from Rutgers University, Piscataway, NJ, and the M.S. degree in computer science from the Stevens Institute of Technology, Hoboken, NJ.

Currently, he is a Professor in the Department of Electrical Engineering at Texas A&M University, College Station, Director of the Genomic Signal Processing Laboratory at Texas A&M University, and Director of the Computational Biology Division of the Translational Genomics Research Institute,

Phoenix, AZ. He is author of 12 books, editor of five others, and author of more

than 180 journal papers. He has contributed extensively to the statistical design of nonlinear operators for image processing and the consequent application of pattern recognition theory to nonlinear image processing. His current research focuses on genomic signal processing, with the central goal being to model genomic regulatory mechanisms for the purposes of diagnosis and therapy.

Prof. Dougherty is a Fellow of the SPIE, was a recipient of the SPIE President's Award, and served as Editor of the *Journal of Electronic Imaging* for six years.



Yufei Xiao (S'99) was born in China. She received the B.S. degree from Zhejiang University, Hangzhou, China, in 1997 and the M.S. degree from the University of Virginia, Charlottesville, in 2002, both in electrical engineering. She is currently working towards the Ph.D. degree in electrical engineering at Texas A&M University, College Station.

From 1997 to 1999, she was a graduate student in Zhejiang University and conducted research in evolutionary computation with application to control systems. From 1999 to 2002, she was a Research

Assistant, working on nonlinear systems and robust filtering. Her current research interests are in genomic signal processing, bioinformatics and pattern recognition.