Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE @ DIRECT®

Signal Processing 83 (2003) 799–809

SIGNAL PROCESSING

www.elsevier.com/locate/sigpro

Mappings between probabilistic Boolean networks

Edward R. Dougherty[a,b], Ilya Shmulevich[b,*]

[a]*Department of Electrical Engineering, Texas A&M University, Houston, TX, USA*
[b]*Cancer Genomics Laboratory, Department of Pathology, The University of Texas M.D. Anderson Cancer Center, Box 85 1515, Holcombe Blvd., Houston, TX 77030, USA*

Received 19 June 2002; received in revised form 21 August 2002

Abstract

Probabilistic Boolean Networks (PBNs) comprise a graphical model based on uncertain rule-based dependencies between nodes and have been proposed as a model for genetic regulatory networks. As with any algebraic structure, the characterization of important mappings between PBNs is critical for both theory and application. This paper treats the construction of mappings to alter PBN structure while at the same time maintaining consistency with the original probability structure. It considers projections onto sub-networks, adjunctions of new nodes, resolution reduction mappings formed by merging nodes, and morphological mappings on the graph structure of the PBN. It places PBNs in the framework of many-sorted algebras and in that context defines homomorphisms between PBNs.

Keywords: Boolean network; Genetic network; Graphical model; Homomorphism; Many-sorted algebra; Mathematical morphology; Projection

1. Introduction

Probabilistic Boolean Networks (PBNs) have been introduced to model gene regulatory networks [10–12]. The model incorporates rule-based dependencies between genes, can cope with uncertainty, allows the systematic study of global network dynamics in the context of Markov chains, and permits the quantification of the relative influence and sensitivity of genes in their interactions with other genes. As shown in [10], PBNs exhibit a connection with Bayesian networks—another class used to model gene expression data [3,4,9]. By incorporating rule-based uncertainty, they represent an interface between the absolute determinism of Boolean networks [7,6,13] and the probabilistic nature of Bayesian networks. This compromise is important because rule-based dependencies between genes are biologically meaningful, while mechanisms for handling uncertainty are conceptually and empirically necessary.

A key aspect of any algebraic structure is the characterization of important mappings, such as homomorphisms. Owing to the large number of states often present in full networks, it is sometimes necessary to construct computationally tractable sub-networks while still carrying sufficient structure for the application at hand. Hence, among other types of mappings, we must consider projections onto sub-networks. This paper treats the construction of mappings to alter

[*] Corresponding author. Tel.: +1-713-745-1502; fax: +1-801-382-2064.
E-mail address: is@ieee.org (I. Shmulevich).

0165-1684/03/$ - see front matter © 2002 Elsevier Science B.V. All rights reserved.
doi:10.1016/S0165-1684(02)00480-2

PBN structure while at the same time maintaining consistency with the original probability structure. In particular, it considers projections, adjunctions (of new nodes), resolution reduction mappings (formed by merging nodes), and morphological mappings on the graph structure of the PBN. It also defines PBN homomorphisms in the context of many-sorted algebras.

2. Definitions and basic properties

We provide basic definitions and notations for PBNs (see [10] for details). A PBN $\mathscr{A} = \mathscr{A}(V,F)$ is defined by a set of binary-valued nodes $V = \{x_1, x_2, \ldots, x_n\}$ and a list $F = \{F_1, F_2, \ldots, F_n\}$ of sets $F_i = \{f_1^{(i)}, f_2^{(i)}, \ldots, f_{l(i)}^{(i)}\}$ of Boolean functions. Each node $x_i \in \{0,1\}$ represents the state (expression) of gene i, where $x_i = 1$ means that gene i is expressed and $x_i = 0$ means it is not expressed. The set F_i contains the possible rules of regulatory interactions for gene x_i. For $j = 1,2,\ldots,l(i)$, $f_j^{(i)} : \{0,1\}^n \to \{0,1\}$ is a possible Boolean function determining the value of x_i in terms of some other gene states. The functions $f_j^{(i)}$ are called *predictors*. Any given gene transforms its inputs (regulatory factors that bind to it) into an output, which is the state or expression of the gene itself. All genes (nodes) are updated synchronously in accordance with the functions assigned to them and this process is then repeated. At every time step, one of the predictors for x_i is selected randomly from the set F_i according to a predefined probability distribution (to be discussed).

A *realization* of a PBN at a given time is determined by a vector of Boolean functions. If there are N possible realizations, then there are N vector functions $\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_N$ of the form $\mathbf{f}_k = (f_{k_1}^{(1)}, f_{k_2}^{(2)}, \ldots, f_{k_n}^{(n)}\}$, for $k = 1,2,\ldots,N$, $1 \leqslant k_i \leqslant l(i)$, and $f_{k_i}^{(i)} \in F_i$ $(i = 1,2,\ldots,n)$. The vector function (also called *multiple-output function*) $\mathbf{f}_k : \{0,1\}^n \to \{0,1\}^n$ acts as a transition function (mapping) representing a possible realization of the entire PBN. Thus, given the values of all genes $(x_1, x_2, \ldots, x_n)$, $\mathbf{f}_k(x_1, x_2, \ldots, x_n) = (x_1^+, x_2^+, \ldots, x_n^+)$ is the state of the genes after one step of the network given by $\mathbf{f}_k$. If the predictor for each gene is chosen independently of other predictors, then $N = l(1)l(2)\cdots l(n)$. Each predictor function $f_j^{(i)}$ usually has many fictitious variables, which means that although the domain of each predictor is $\{0,1\}^n$, there are only a few input genes that actually regulate x_i at any given time. There are biological and practical justifications for probabilistically choosing one of several simple predictors for each gene [10].

Stochastically, the multiple-output function is a random vector $\mathbf{f} = (f^{(1)}, f^{(2)}, \ldots, f^{(n)})$ taking values in $F_1 \times F_2 \times \cdots \times F_n$, meaning that $\mathbf{f} = \mathbf{f}_k$ for some k. The *selection probability* that the predictor $f_j^{(i)}$ is used to determine gene i $(1 \leqslant i \leqslant l(i))$ is given by

$$c_j^{(i)} = P(f^{(i)} = f_j^{(i)}) = \sum_{k: f_{k_j}^{(i)} = f_j^{(i)}} P(\mathbf{f} = \mathbf{f}_k). \tag{1}$$

In general there needs to be no assumption that $f^{(1)}, f^{(2)}, \ldots, f^{(n)}$ are selected independently; however, here we make that assumption. Hence,

$$P(\mathbf{f} = \mathbf{f}_k) = \prod_{j=1}^{n} P(f^{(j)} = f_{k_j}^{(j)}) = \prod_{j=1}^{n} c_{k_j}^{(i)}. \tag{2}$$

In [10], the selection probabilities are obtained by using the coefficient of determination [1,8].

A PBN is a homogeneous Markov chain relative to the states $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ of the system with transition probabilities being given by

$$P(\mathbf{x} \to \mathbf{x}^+) = \sum_{k=1}^{N} P(\mathbf{f} = \mathbf{f}_k)\delta(\mathbf{f}_k(\mathbf{x}) - \mathbf{x}^+), \tag{3}$$

where the delta function is 1 or 0, depending on whether or not $\mathbf{f}_k(\mathbf{x}) = \mathbf{x}^+$.

3. Projection

To reduce the complexity of a PBN $\mathscr{A}$, one or more genes can be deleted from the network. The resulting mappings will be referred to as *projections*. To ease the notational burden, we discuss projection mappings in the context of a 4-gene network $\mathscr{A}$ consisting of the gene set $\{x, y, z, w\}$, with the Boolean function class for gene x being $F_x = \{f_k^{(x)}(x, y, z, w)\}_{k=1,2,\ldots,m}$, where without loss of generality we assume all function classes are of size m (if not, then arbitrary functions with null selection probabilities can be adjoined). The selection probability for $f_k^{(x)}$ is $c_k^{(x)}$.

First, consider deleting a single gene, where without loss of generality we delete gene y and consider the corresponding effect on x. Since y is not part of the new network, and since it can have two possible values, 0 or 1, each Boolean function in F_x for network $\mathscr{A}$ induces two Boolean functions for x in the projected network, $\pi(\mathscr{A})$:

$$f_k^{(x)}(x,y,z,w) \to \begin{cases} g_{k,1}^{(x)}(x,z,w) \equiv f_k^{(x)}(x,0,z,w), \\ g_{k,2}^{(x)}(x,z,w) \equiv f_k^{(x)}(x,1,z,w). \end{cases} \tag{4}$$

The new Boolean functions $g_{k,1}^{(x)}$ and $g_{k,2}^{(x)}$ have selection probabilities

$$\begin{aligned} d_{k,1}^{(x)} &= c_k^{(x)} P(y=0), \\ d_{k,2}^{(x)} &= c_k^{(x)} P(y=1), \end{aligned} \tag{5}$$

where we have used the model assumption that function selection is state independent. The Boolean function class, $\pi(F_x)$, for x in $\pi(\mathscr{A})$ is composed of the functions $g_{k,j}^{(x)}(x,z,w)$, for $k=1,2,\ldots,m$, and $j=1,2$. $\pi(F_z)$ and $\pi(F_w)$ are defined analogously.

Example 1. Suppose that in a 4-gene network, consisting of genes x, y, z, and w, the first predictor for gene x is $f_1^{(x)}(x,y,z,w) = (x \wedge y) \vee (y \wedge \bar{z}) \vee (\bar{x} \wedge w)$ and its selection probability is $c_1^{(x)} = 0.6$. If we delete gene y, then we form two new Boolean functions $g_{1,1}^{(x)}(x,z,w)$ and $g_{1,2}^{(x)}(x,z,w)$ as:

$$\begin{aligned} g_{1,1}^{(x)}(x,z,w) &\equiv f_1^{(x)}(x,0,z,w) \\ &= (x \wedge 0) \vee (0 \wedge \bar{z}) \vee (\bar{x} \wedge w) = \bar{x} \wedge w, \end{aligned}$$

$$\begin{aligned} g_{1,2}^{(x)}(x,z,w) &\equiv f_1^{(x)}(x,1,z,w) \\ &= (x \wedge 1) \vee (1 \wedge \bar{z}) \vee (\bar{x} \wedge w) = x \vee w \vee \bar{z}. \end{aligned}$$

Furthermore, suppose that $P(y=0) = 0.8$ and $P(y=1) = 0.2$. Then, it follows that

$$d_{1,1}^{(x)} = c_1^{(x)} P(y=0) = 0.6 \times 0.8 = 0.48$$

$$d_{1,2}^{(x)} = c_1^{(x)} P(y=1) = 0.6 \times 0.2 = 0.12.$$

In fact, there is a difficulty with the transformation of the selection probabilities because the probabilities for y are time-independent. Moreover, they may be unknown. One way to treat this problem is to replace them by the corresponding steady-state probabilities for y, if they are known (and there exists a steady-state distribution). If the steady-state distribution is unknown, or does not exist, then $P(y=0)$ and $P(y=1)$ can be estimated by running the network for some time and estimating these probabilities, recognizing that the estimates include transient behavior. Still another way is to estimate $P(y=0)$ and $P(y=1)$ using the data set from which the PBN was originally constructed.

If two genes, say y and z are deleted, then each Boolean function for x results in four Boolean functions in $\pi(F_x)$:

$$f_k^{(x)}(x,y,z,w) \to \begin{cases} g_{k,1}^{(x)}(x,w) \equiv f_k^{(x)}(x,0,0,w), \\ g_{k,2}^{(x)}(x,w) \equiv f_k^{(x)}(x,0,1,w), \\ g_{k,3}^{(x)}(x,w) \equiv f_k^{(x)}(x,1,0,w), \\ g_{k,4}^{(x)}(x,w) \equiv f_k^{(x)}(x,1,1,w). \end{cases} \tag{6}$$

The function class, $\pi(F_x)$, for x in $\pi(\mathscr{A})$ is composed of $g_{k,j}^{(x)}(x,w)$, for $k=1,2,\ldots,m$, and $j=1,2,3,4$. The new Boolean functions $g_{k,1}^{(x)}$, $g_{k,2}^{(x)}$, $g_{k,3}^{(x)}$, and $g_{k,4}^{(x)}$ have selection probabilities

$$\begin{aligned} d_{k,1}^{(x)} &= c_k^{(x)} p_{yz}(00), \\ d_{k,2}^{(x)} &= c_k^{(x)} p_{yz}(01), \\ d_{k,3}^{(x)} &= c_k^{(x)} p_{yz}(10), \\ d_{k,4}^{(x)} &= c_k^{(x)} p_{yz}(11), \end{aligned} \tag{7}$$

where $p_{yz}(ab) = P(y=a, z=b)$. More genes can be deleted in an analogous manner.

A second kind of projection results from permanently setting a gene to 0 or 1. Consider setting $y \equiv 0$. By recognizing that y no longer plays a role in the network dynamics, fixing it can be considered to be a projection. For the other genes, each Boolean function results in a new function by setting $y \equiv 0$. For instance, in the case of x,

$$f_k^{(x)}(x,y,z,w) \to g_k^{(x)}(x,z,w) = f_k^{(x)}(x,0,z,w). \tag{8}$$

The corresponding selection probability in the new network is $d_k^{(x)} = c_k^{(x)}$.

4. Adjunction

The inverse operation to projection is adjunction of a gene to the network. We begin with a 3-variable network $\{y, z, w\}$ and adjoin x. In $\{y, z, w\}$, we will assume that w has m Boolean functions $g_1^{(w)}, g_2^{(w)}, \ldots, g_m^{(w)}$ with associated selection probabilities $d_1^{(w)}, d_2^{(w)}, \ldots, d_m^{(w)}$, where m is even. If m is odd, then we can simply include an arbitrary function with zero selection probability. The Boolean functions for w in the expanded 4-gene network are constrained by their need to map into the original 3-gene network according to the rules for deletion of a gene. This constitutes the inverse requirement relative to projection (or *consistency* with projection). There are $m!/(m/2)!$ possible Boolean-function families for w. A family is generated by taking a permutation $g_{(1)}^{(w)}, g_{(2)}^{(w)}, \ldots, g_{(m)}^{(w)}$ of $g_1^{(w)}, g_2^{(w)}, \ldots, g_m^{(w)}$ and defining $m/2$ Boolean functions by

$$f_1^{(w)}(x,y,z,w) \equiv \begin{cases} f_1^{(w)}(0,y,z,w) = g_{(1)}^{(w)}(y,z,w), \\ f_1^{(w)}(1,y,z,w) = g_{(2)}^{(w)}(y,z,w), \end{cases}$$

$$f_2^{(w)}(x,y,z,w) \equiv \begin{cases} f_2^{(w)}(0,y,z,w) = g_{(3)}^{(w)}(y,z,w), \\ f_2^{(w)}(1,y,z,w) = g_{(4)}^{(w)}(y,z,w), \end{cases}$$

$$\vdots$$

$$f_{m/2}^{(w)}(x,y,z,w) \equiv \begin{cases} f_{m/2}^{(w)}(0,y,z,w) = g_{(m-1)}^{(w)}(y,z,w), \\ f_{m/2}^{(w)}(1,y,z,w) = g_{(m)}^{(w)}(y,z,w) \end{cases}$$

and there are $m/2$ such functions in each of the possible families. Since there are m permutations of the original functions and for each permutation there are $(m/2)!$ equivalent families generated, the total number of Boolean function families for w is indeed $m!/(m/2)!$.

The selection probability $c_k^{(w)}$ is determined by the deletion mapping. Specifically, if

$$f_k^{(w)}(x,y,z,w) \equiv \begin{cases} f_k^{(w)}(0,y,z,w) = g_i^{(w)}(y,z,w), \\ f_k^{(w)}(1,y,z,w) = g_j^{(w)}(y,z,w) \end{cases} \tag{10}$$

then deletion must yield

$$\begin{aligned} d_i^{(w)} &= P(x=0)c_k^{(w)}, \\ d_j^{(w)} &= P(x=1)c_k^{(w)}. \end{aligned} \tag{11}$$

Adding these equations yields $c_k^{(w)} = d_i^{(w)} + d_j^{(w)}$.

Since projection is not a one-to-one mapping between PBNs, its inverse is a multi-valued mapping. While $m!$ may seem like a large number of possible inverses, this number of possible families is greatly constrained relative to the number of all such possible families. For instance, suppose w has the four Boolean functions $g_1^{(w)}, g_2^{(w)}, g_3^{(w)}$, and $g_4^{(w)}$. Then there are 24 possible function families $\{f_1^{(w)}, f_2^{(w)}\}$. Each is of the form:

$$\begin{aligned} f_1^{(w)}(x,y,z,w) &\equiv \begin{cases} f_1^{(w)}(0,y,z,w) = g_i^{(w)}(y,z,w), \\ f_1^{(w)}(1,y,z,w) = g_j^{(w)}(y,z,w), \end{cases} \\ f_2^{(w)}(x,y,z,w) &\equiv \begin{cases} f_2^{(w)}(0,y,z,w) = g_k^{(w)}(y,z,w), \\ f_2^{(w)}(1,y,z,w) = g_l^{(w)}(y,z,w), \end{cases} \end{aligned} \tag{12}$$

where (i, j, k, l) is a permutation of $(1, 2, 3, 4)$. The selection probabilities associated with this family are $c_1^{(w)} = d_i^{(w)} + d_j^{(w)}$ and $c_2^{(w)} = d_k^{(w)} + d_l^{(w)}$.

Example 2. Suppose that in a 3-gene network consisting of genes y, z, and w, we wish to adjoin gene x. Suppose further that gene w has two Boolean functions $g_1^{(w)}(y,z,w) = y \vee (z \wedge w)$ and $g_2^{(w)}(y,z,w) = y \vee z \vee w$, with respective selection probabilities $d_1^{(w)} = 0.6$ and $d_2^{(w)} = 0.4$. Then, after adjoining gene x, one choice for the Boolean function for w would be

$$\begin{aligned} &f_1^{(w)}(x,y,z,w) \\ &\quad = (\bar{x} \wedge g_1^{(w)}(y,z,w)) \vee (x \wedge g_2^{(w)}(y,z,w)) \\ &\quad = y \vee (x \wedge z) \vee (w \wedge z) \vee (x \wedge w) \end{aligned}$$

with selection probability $c_1^{(w)} = 0.6 + 0.4 = 1$, and the other choice would be

$$\begin{aligned} &f_1^{(w)}(x,y,z,w) \\ &\quad = (x \wedge g_1^{(w)}(y,z,w)) \vee (\bar{x} \wedge g_2^{(w)}(y,z,w)) \\ &\quad = y \vee (\bar{x} \wedge z) \vee (w \wedge z) \vee (\bar{x} \wedge w) \end{aligned}$$

with the same selection probability $c_1^{(w)} = 0.6 + 0.4 = 1$. Note that after adjoining x, there is only one Boolean function in each of the two possible families; this is why the subscript in $f_1^{(w)}$ is equal to 1 in both cases.

The ways in which projection and adjunction affect the transition probabilities of the Markov chain corresponding to a PBN is related directly to their defining equations. To demonstrate this, we assume for the sake of simplicity that $\{y,z\} = \pi(\{x,y,z\}), x$ having been deleted. Without loss of generality we consider the transition $00 \rightarrow 00$ in the state space (other transition analyses being similar). Under the assumption that the Boolean functions for different genes are selected independently, $P(00 \rightarrow 00)$ is obtained by summing all products of selection probabilities for y and z for functions $g^{(y)}$ and $g^{(z)}$ such that $g^{(y)}(00) = g^{(z)}(00) = 0$. Based on Eq. (4), the function class for y is partitioned into two subclasses, $\{g_{1,1}^{(y)}, g_{2,1}^{(y)}, \ldots, g_{m,1}^{(y)}\}$ and $\{g_{1,2}^{(y)}, g_{2,2}^{(y)}, \ldots, g_{m,2}^{(y)}\}$. Similarly, the function class for z is partitioned into $\{g_{1,1}^{(z)}, g_{2,1}^{(z)}, \ldots, g_{m,1}^{(z)}\}$ and $\{g_{1,2}^{(z)}, g_{2,2}^{(z)}, \ldots, g_{m,2}^{(z)}\}$. Thus,

$$
\begin{aligned}
&P(00 \rightarrow 00)\\
&\quad = \sum d_{k,1}^{(y)} d_{j,1}^{(z)} \delta(g_{k,1}^{(y)}(00)) \delta(g_{j,1}^{(z)}(00))\\
&\qquad + \sum d_{k,1}^{(y)} d_{j,2}^{(z)} \delta(g_{k,1}^{(y)}(00)) \delta(g_{j,2}^{(z)}(00))\\
&\qquad + \sum d_{k,2}^{(y)} d_{j,1}^{(z)} \delta(g_{k,2}^{(y)}(00)) \delta(g_{j,1}^{(z)}(00))\\
&\qquad + \sum d_{k,2}^{(y)} d_{j,2}^{(z)} \delta(g_{k,2}^{(y)}(00)) \delta(g_{j,2}^{(z)}(00)),\\
&\quad = \sum c_k^{(y)} c_j^{(z)} P(x=0)^2 \delta(f_k^{(y)}(000)) \delta(f_j^{(z)}(000))\\
&\qquad + \sum c_k^{(y)} c_j^{(z)} P(x=0) P(x=1) \delta(f_k^{(y)}(000))\\
&\qquad \times \delta(f_j^{(z)}(100))\\
&\qquad + \sum c_k^{(y)} c_j^{(z)} P(x=1) P(x=0) \delta(f_k^{(y)}(100))\\
&\qquad \times \delta(f_j^{(z)}(000))\\
&\qquad + \sum c_k^{(y)} c_j^{(z)} P(x=1)^2 \delta(f_k^{(y)}(100))\\
&\qquad \times \delta(f_j^{(z)}(100)),
\end{aligned}
\tag{13}
$$

where the second equality follows from Eqs. (4) and (5). If $\{x,y,z\}$ has been obtained from $\{y,z\}$ by adjoining x, then applying Eqs. (10) and (11) to the second set of sums yields the first set.

Adjoining genes to a PBN assuming consistency with projection has benefits when it comes to estimation of Boolean functions from data. If we ignore our prior knowledge regarding the existing PBN and wish to adjoin gene x, then it is necessary to estimate from data the Boolean functions for x as well as new Boolean functions for the other genes that take into account possible dependence on x. The latter estimation is simplified by the inverse requirement.

To appreciate the savings, consider the case of w with four Boolean functions $g_1^{(w)}$, $g_2^{(w)}$, $g_3^{(w)}$, and $g_4^{(w)}$ just discussed, in which x is to be adjoined. In the absence of prior knowledge, we would have to estimate from sample data the desired predictors of y, z, and w based on observations of the four variables. For instance, the optimal predictor for w is of the form $\hat{w} = \hat{f}(x,y,z,w)$ and it is designed from sample data by defining $\hat{f}(x,y,z,w) = 1$ if $w^+ = 1$ more often than $w^+ = 0$ when (x,y,z,w) is observed, and $\hat{f}(x,y,z,w) = 0$ otherwise. The data requirement for precise estimation depends on the number of possible functions, which in this case is 2^{16}. One of the reasons for using probabilistic rather than deterministic Boolean networks is that we use more than the experimentally estimated optimal predictor and thereby make the estimated network more robust relative to estimation imprecision.

Estimation complexity is greatly reduced under the inverse requirement. For the example just considered, there are only 24 possible two-function families from which to choose. Estimation can be accomplished by choosing the family whose functions minimize the empirical error on the sample data. Without the inverse constraint, it would be necessary to choose from among all functions a certain size family of functions that minimizes the empirical error. The particular optimal case described gives just a single function possessing minimal empirical error.

5. Resolution reduction

Network resolution can be reduced by a resolution-reduction mapping that merges genes. The Boolean functions in the new PBN must reflect the integration (merging). The value of a new node must be defined as a function of the original node values, such as the median, maximum or minimum, to name some possibilities.

To mitigate notation, we explain the mapping procedure for a specific two-gene merging under the assumption of a four-gene PBN having genes x, y, z, and w. We merge y and z to form the *integrated* (maximum) gene $yz = y \vee z$. If y and z always function within the regulatory network so that if either of them is 1-valued, then a signal is sent, then nothing is lost by lowering the resolution in this way. Otherwise, there is a coarsening of the network that might lose some information, with the concomitant advantage of less complexity for computation and estimation. The advantage and disadvantage of lower resolution must be weighed. Maximum is only one of 16 possible merging functions that can be used for two genes. If y and z function within the regulatory network in such a way that only if both are 1-valued, then a signal is sent, then the minimum merging function, $y \wedge z$, is appropriate. We continue here with the maximum.

Two cases must be addressed. First, we need to define the families $\{g_k^{(x)}(x, yz, w)\}$ and $\{g_k^{(w)}(x, yz, w)\}$. We consider $g_k^{(x)}$, with $g_k^{(w)}$ being defined analogously. Each Boolean function $f_k^{(x)}(x, y, z, w)$ for x generates three Boolean functions in the new network, depending upon the outcomes of y and z. If $yz = 0$ in an induced Boolean function $g_k^{(x)}(x, yz, w)$, then it must be that $y = z = 0$; however, if $yz = 1$, then there are three possibilities: $y = 0$ and $z = 1$, $y = 1$ and $z = 0$, and $y = z = 1$. Hence, three functions are induced according to the following scheme:

$$f_k^{(x)}(x, y, z, w) \rightarrow$$

$$\left\{ \begin{array}{l} g_{k,1}^{(x)}(x, yz, w) \equiv \left\{ \begin{array}{l} g_{k,1}^{(x)}(x, 0, w) = f_k^{(x)}(x, 0, 0, w) \\ g_{k,1}^{(x)}(x, 1, w) = f_k^{(x)}(x, 0, 1, w) \end{array} \right. \\ g_{k,2}^{(x)}(x, yz, w) \equiv \left\{ \begin{array}{l} g_{k,2}^{(x)}(x, 0, w) = f_k^{(x)}(x, 0, 0, w) \\ g_{k,2}^{(x)}(x, 1, w) = f_k^{(x)}(x, 1, 0, w) \end{array} \right. \\ g_{k,3}^{(x)}(x, yz, w) \equiv \left\{ \begin{array}{l} g_{k,3}^{(x)}(x, 0, w) = f_k^{(x)}(x, 0, 0, w) \\ g_{k,3}^{(x)}(x, 1, w) = f_k^{(x)}(x, 1, 1, w). \end{array} \right. \end{array} \right. \tag{14}$$

The associated selection probabilities are

$$\begin{aligned} d_{k,1}^{(x)} &= c_k^{(x)} p_{yz}(01) T_{\max}^{-1}, \\ d_{k,2}^{(x)} &= c_k^{(x)} p_{yz}(10) T_{\max}^{-1}, \\ d_{k,3}^{(x)} &= c_k^{(x)} p_{yz}(11) T_{\max}^{-1}, \end{aligned} \tag{15}$$

where

$$T_{\max} = p_{yz}(01) + p_{yz}(10) + p_{yz}(11). \tag{16}$$

If there are m Boolean functions for x in the original PBN, then in the resolution-reduced PBN there are $3m$ Boolean functions for x. Function $g_{k,1}^{(x)}$ is selected with probability $d_{k,1}^{(x)}$, and if it is selected, then $x^+ = f_k^{(x)}(x, 0, 0, w)$ if $yz = 0$ and $x^+ = f_k^{(x)}(x, 0, 1, w)$ if $yz = 1$. Analogous comments apply to $g_{k,2}^{(x)}$ and $g_{k,3}^{(x)}$.

Example 3. Suppose that as in Example 1, $f_1^{(x)}(x, y, z, w) = (x \wedge y) \vee (y \wedge \bar{z}) \vee (\bar{x} \wedge w)$. Let us merge genes y and z by taking their maximum. For notational simplicity, let us denote the merged gene as v (this was yz above). There are four possibilities: $f_1^{(x)}(x, 0, 0, w) = \bar{x} \wedge w$, $f_1^{(x)}(x, 0, 1, w) = \bar{x} \wedge w$, $f_1^{(x)}(x, 1, 0, w) = 1$, and $f_1^{(x)}(x, 1, 1, w) = x \vee w$, depending on the values of y and z. Thus, as shown in Eq. (14), there are three choices:

$$g_{1,1}^{(x)}(x, v, w) = \bar{x} \wedge w,$$

$$g_{1,2}^{(x)}(x, v, w) = v \vee (\bar{x} \wedge w),$$

$$g_{1,3}^{(x)}(x, v, w) = (w \wedge v) \vee (\bar{x} \wedge w) \vee (v \wedge x).$$

It is easy to see that in each case, setting v to either 0 or 1 will result in one of the four functions shown above.

To define the Boolean functions for the integrated gene yz, we have, as for x, three possible cases corresponding to the ways in which yz can be 1. The difference is that the value of yz must reflect the manner in which it is defined by the maximum. Hence, for each pair of Boolean functions, $f_k^{(y)}(x, y, z, w)$ and $f_j^{(z)}(x, y, z, w)$, corresponding to y and z, three Boolean functions are generated for yz according to the following scheme:

$$\begin{pmatrix} f_k^{(y)}(x, y, z, w) \\ f_j^{(z)}(x, y, z, w) \end{pmatrix} \rightarrow$$

$$
\begin{cases}
g^{(yz)}_{k,j,1}(x, yz, w) \equiv \\
\quad \begin{cases} g^{(yz)}_{k,j,1}(x,0,w) = f^{(y)}_{k}(x,0,0,w) \vee f^{(z)}_{j}(x,0,0,w) \\ g^{(yz)}_{k,j,1}(x,1,w) = f^{(y)}_{k}(x,0,1,w) \vee f^{(z)}_{j}(x,0,1,w) \end{cases} \\
g^{(yz)}_{k,j,2}(x, yz, w) \equiv \\
\quad \begin{cases} g^{(yz)}_{k,j,2}(x,0,w) = f^{(y)}_{k}(x,0,0,w) \vee f^{(z)}_{j}(x,0,0,w) \\ g^{(yz)}_{k,j,2}(x,1,w) = f^{(y)}_{k}(x,1,0,w) \vee f^{(z)}_{j}(x,1,0,w) \end{cases} \\
g^{(yz)}_{k,j,3}(x, yz, w) \equiv \\
\quad \begin{cases} g^{(yz)}_{k,j,3}(x,0,w) = f^{(y)}_{k}(x,0,0,w) \vee f^{(z)}_{j}(x,0,0,w) \\ g^{(yz)}_{k,j,3}(x,1,w) = f^{(y)}_{k}(x,1,1,w) \vee f^{(z)}_{j}(x,1,1,w). \end{cases}
\end{cases}
\tag{17}
$$

The selection probabilities are given by

$$
\begin{aligned}
d^{(yz)}_{k,j,1} &= c^{(y)}_{k} c^{(z)}_{j} p_{yz}(01) T^{-1}_{\max}, \\
d^{(yz)}_{k,j,2} &= c^{(y)}_{k} c^{(z)}_{j} p_{yz}(10) T^{-1}_{\max}, \\
d^{(yz)}_{k,j,3} &= c^{(y)}_{k} c^{(z)}_{j} p_{yz}(11) T^{-1}_{\max}.
\end{aligned}
\tag{18}
$$

We can use any of the 16 possible two-variable binary functions. For instance, for exclusive-or, $\oplus$, there are four Boolean functions in the resolution-reduced PBN for x and w to reflect the fact that $0 \oplus 0 = 1 \oplus 1 = 0$ and $1 \oplus 0 = 0 \oplus 1 = 1$. For x,

$$
f^{(x)}_{k}(x, y, z, w) \rightarrow
$$

$$
\begin{cases}
g^{(x)}_{k,1}(x, yz, w) \equiv \begin{cases} g^{(x)}_{k,1}(x,0,w) = f^{(x)}_{k}(x,0,0,w) \\ g^{(x)}_{k,1}(x,1,w) = f^{(x)}_{k}(x,0,1,w) \end{cases} \\
g^{(x)}_{k,2}(x, yz, w) \equiv \begin{cases} g^{(x)}_{k,2}(x,0,w) = f^{(x)}_{k}(x,0,0,w) \\ g^{(x)}_{k,2}(x,1,w) = f^{(x)}_{k}(x,1,0,w) \end{cases} \\
g^{(x)}_{k,3}(x, yz, w) \equiv \begin{cases} g^{(x)}_{k,3}(x,0,w) = f^{(x)}_{k}(x,1,1,w) \\ g^{(x)}_{k,3}(x,1,w) = f^{(x)}_{k}(x,0,1,w) \end{cases} \\
g^{(x)}_{k,4}(x, yz, w) \equiv \begin{cases} g^{(x)}_{k,4}(x,0,w) = f^{(x)}_{k}(x,1,1,w) \\ g^{(x)}_{k,4}(x,1,w) = f^{(x)}_{k}(x,1,0,w). \end{cases}
\end{cases}
\tag{19}
$$

The associated selection probabilities are

$$
\begin{aligned}
d^{(x)}_{k,1} &= c^{(x)}_{k} p_{yz}(00) p_{yz}(01) T^{-1}_{\mathrm{XOR}}, \\
d^{(x)}_{k,2} &= c^{(x)}_{k} p_{yz}(00) p_{yz}(10) T^{-1}_{\mathrm{XOR}}, \\
d^{(x)}_{k,3} &= c^{(x)}_{k} p_{yz}(11) p_{yz}(01) T^{-1}_{\mathrm{XOR}}, \\
d^{(x)}_{k,4} &= c^{(x)}_{k} p_{yz}(11) p_{yz}(10) T^{-1}_{\mathrm{XOR}},
\end{aligned}
\tag{20}
$$

where

$$
\begin{aligned}
T_{\mathrm{XOR}} = {} & p_{yz}(00) p_{yz}(01) + p_{yz}(00) p_{yz}(10) \\
& + p_{yz}(11) p_{yz}(01) + p_{yz}(11) p_{yz}(10).
\end{aligned}
\tag{21}
$$

For the integrated gene yz,

$$
\begin{pmatrix} f^{(y)}_{k}(x, y, z, w) \\ f^{(z)}_{j}(x, y, z, w) \end{pmatrix} \rightarrow
$$

$$
\begin{cases}
g^{(yz)}_{k,j,1}(x, yz, w) \equiv \\
\quad \begin{cases} g^{(yz)}_{k,j,1}(x,0,w) = f^{(y)}_{k}(x,0,0,w) \oplus f^{(z)}_{j}(x,0,0,w) \\ g^{(yz)}_{k,j,1}(x,1,w) = f^{(y)}_{k}(x,0,1,w) \oplus f^{(z)}_{j}(x,0,1,w) \end{cases} \\
g^{(yz)}_{k,j,2}(x, yz, w) \equiv \\
\quad \begin{cases} g^{(yz)}_{k,j,2}(x,0,w) = f^{(y)}_{k}(x,0,0,w) \oplus f^{(z)}_{j}(x,0,0,w) \\ g^{(yz)}_{k,j,2}(x,1,w) = f^{(y)}_{k}(x,1,0,w) \oplus f^{(z)}_{j}(x,1,0,w) \end{cases} \\
g^{(yz)}_{k,j,3}(x, yz, w) \equiv \\
\quad \begin{cases} g^{(yz)}_{k,j,3}(x,0,w) = f^{(y)}_{k}(x,1,1,w) \oplus f^{(z)}_{j}(x,1,1,w) \\ g^{(yz)}_{k,j,3}(x,1,w) = f^{(y)}_{k}(x,0,1,w) \oplus f^{(z)}_{j}(x,0,1,w) \end{cases} \\
g^{(yz)}_{k,j,4}(x, yz, w) \equiv \\
\quad \begin{cases} g^{(yz)}_{k,j,4}(x,0,w) = f^{(y)}_{k}(x,1,1,w) \oplus f^{(z)}_{j}(x,1,1,w) \\ g^{(yz)}_{k,j,4}(x,1,w) = f^{(y)}_{k}(x,1,0,w) \oplus f^{(z)}_{j}(x,1,0,w). \end{cases}
\end{cases}
\tag{22}
$$

The selection probabilities are

$$
\begin{aligned}
d^{(x)}_{k,1} &= c^{(y)}_{k} c^{(z)}_{j} p_{yz}(00) p_{yz}(01) T^{-1}_{\mathrm{XOR}}, \\
d^{(x)}_{k,2} &= c^{(y)}_{k} c^{(z)}_{j} p_{yz}(00) p_{yz}(10) T^{-1}_{\mathrm{XOR}}, \\
d^{(x)}_{k,3} &= c^{(y)}_{k} c^{(z)}_{j} p_{yz}(11) p_{yz}(01) T^{-1}_{\mathrm{XOR}}, \\
d^{(x)}_{k,4} &= c^{(y)}_{k} c^{(z)}_{j} p_{yz}(11) p_{yz}(10) T^{-1}_{\mathrm{XOR}}.
\end{aligned}
\tag{23}
$$

6. Morphological mappings

If from a PBN $\mathscr{A}$ we select a subset G of genes from the total set of genes for the PBN, then two things are likely: (1) there will be genes outside of G that are essential for Boolean functions in the function families for genes in G; and (2) there will be genes outside of G for whose Boolean functions genes in G are essential. If we wish to treat G, together with the function families for genes in G, as a PBN $\mathscr{G}$, then the second condition is not an impediment because transitions of the state vectors formed by values of genes in G are fully determined by the function families for G itself. However, such is not the case for the first condition. Hence, we are faced with generating a PBN with G as its gene set while preserving to the extent possible the function structure imparted to G by the full PBN $\mathscr{A}$.

To address the issue, let $\mathscr{F}_G = \{F_x\colon x \in G\}$ be the collection of function families for genes in G, $B_G = \bigcup_{x\in G} F_x$ be the union of such function families, and G_B be the set of all genes essential for at least one function in B_G. If $G_B \subset G$, then we say G is *closed* (relative to $\mathscr{A}$). If G is closed, then $\mathscr{G} = (G, \mathscr{F}_G)$ forms a *sub-PBN* of $\mathscr{A}$. This means that no gene in G depends on a gene outside of G. It is possible to have $G_B \subset G$ properly, meaning $G_B \neq G$. This means there is a gene in G *nonessential* for all genes in G, including itself. Such a gene could be dropped from G and the new gene set together with its function families would still be a sub-PBN. If no such gene exists, then we say that G is *minimal*. Just because a gene is nonessential for G does not mean that it should be dropped. For instance, the value of a nonessential gene may reflect the state of $\mathscr{G}$.

If G is not closed ($G_B \not\subset G$), then there exists a gene $w \in G$, a function $f_w \in F_w$, and a gene $y \notin G$ such that y is essential for f_w. We say that w is *incomplete* relative to G. If a gene in G is not incomplete, then we say it is *complete* relative to G. If gene y is essential for f_w and $y \notin G$, then we can map f_w into two new mappings just as in Eqs. (4) and (5). This is because rendering y inessential for w is the same as deleting it from the network. Should two essential genes be missing from G, we proceed as in Eqs. (6) and (7), and so on. If we do this for all incomplete genes, then the resulting PBN is said to be the PBN *generated* by G. We denote it by $\Gamma[G]$. G is a sub-PBN if and only if $G = \Gamma[G]$. If one is only interested in a subset of genes within a PBN, it can be advantageous to focus on the generated PBN because this reduces the dimensionality of the state space. The newly created Boolean functions for $\Gamma[G]$ are said to be *induced*.

If G is not closed, then given a state $\mathbf{s}$ of the generated PBN $\Gamma[G]$ at time t, the next state $\mathbf{s}^+$ may be different depending on whether $\mathbf{s}$ is treated as a state of $\Gamma[G]$ or as part of the full state vector at the same time for the full PBN. To explain, relative to the full PBN, the state vector for $\Gamma[G]$ forms a part of the vector, so that the full vector takes the form $(\mathbf{s}, \mathbf{r})$, where $\mathbf{r}$ is composed of the values of genes not in G. With G not closed, the transition $\mathbf{s} \to \mathbf{s}^+$ may depend upon induced Boolean functions, and therefore may yield a different state than would be obtained by the transition $(\mathbf{s}, \mathbf{r}) \to (\mathbf{s}, \mathbf{r})^+$. On the other hand, if G is closed, then $\mathbf{s}^+$ is identical to the state vector obtained from $(\mathbf{s}, \mathbf{r})^+$ by taking the values for the genes in G. If x is the only incomplete gene in G, then this lack of consistency can be fixed, relative to x, by adjoining all of its essential genes to G to form a new gene set $G_{x\leftarrow}$, and considering the generated PBN $\Gamma[G_{x\leftarrow}]$. Of course, $G_{x\leftarrow}$ may not be closed, but at least the inconsistency is not now due to x.

This entire issue can be addressed by treating the full PBN as a directed graph. The nodes of the graph are the genes and there is an edge pointing from gene x to gene y if x is essential for y. If G is a subset of genes, then it is a sub-PBN if every edge whose front is in G has its back also in G. A gene x is incomplete if and only if there is an edge whose front is x but whose back is outside of G. Even if gene $w \in G$ is complete relative to G, it is possible that an essential gene will be absent after some finite number of transitions so long as G is not closed. All essential genes for w are in G, but if x is one of those essential genes, then the transition $w \to w^+$ is guaranteed not to require a gene outside of G but the double transition $w \to w^+ \to w^{++}$ might. We say that gene $w \in G$ is level-r complete relative to G if the r-fold transition $w \to w^+ \to \cdots \to w^{+(r)}$ is certain not to require any gene outside of G, where $w^{+(r)}$ denotes an r-fold transition. For a gene to be complete up to a high level means intuitively that it is “deep” within G. This concept can be rigorously described in the framework of mathematical morphology on graphs [15,5].

To describe the basic morphological operations on graphs, define the *distance* $d(x, y)$ between two nodes

x and y to be the shortest path from x to y (without reference to edge directionality). If there is no path from x to y, define $d(x, y) = \infty$. For any node x and positive integer r, the ball centered at x of radius r, $B_r(x)$, is the set of all nodes y such that $d(x, y) \leqslant r$. For any node set G, the size-r *dilation* of G, $\Delta^{(r)}(G)$, consists of all nodes y for which there exists a node $x \in G$ with $y \in B_r(x)$. The size-r *erosion* of G, $E^{(r)}(G)$, consists of all nodes y such that $B_r(y) \subset G$. Elementary dilation Δ and elementary erosion E are defined by $r = 1$. Dilation and erosion can be computed iteratively from the elementary operations: $\Delta^{(r)} = \Delta\Delta \cdots \Delta(r \text{ times})$ and $E^{(r)} = EE \cdots E(rtimes)$.

Returning to the PBN graph with vertices determined by function essentiality, the incomplete nodes of G are complete within $\Delta(G)$. This does not mean $\Delta(G)$ is closed. There may be genes in the *outer boundary* $\Delta(G) - G$ that are incomplete in $\Delta(G)$. More generally, all genes in G are complete up to level r in $\Delta^{(r)}(G)$. Genes in the outer r-boundary of G, $\Delta^{(r)}(G) - G$, might not be complete up to level r in $\Delta^{(r)}(G)$. Analogous considerations apply for erosion, which is dual to dilation. All genes in $E(G)$ are complete relative to G, and all genes in $E^{(r)}(G)$ are complete up to level r in G. This is not assured for genes in the inner r-boundary of G, $G - E^{(r)}(G)$.

As so far defined, dilation and erosion are independent of edge direction. Elementary dilation adjoins gene y to G if there is a gene $x \in G$ for which y is essential for x or if x is essential for y. A gene might be adjoined to G by Δ even if it is nonessential relative to G. To be adjoined, it can be related to G either as an independent or dependent variable. The latter case may be important, especially if we are interested in genes whose values reflect the values of genes in G. However, if our primary goal is to reduce dependence on genes outside the set, then it would be better to only adjoin genes essential for G. For this purpose, we need to re-define the morphological operations taking into account edge directionality.

The size-r *directional dilation* of G, $\underline{\Delta}^{(r)}(G)$, consists of all nodes y for which there exists a node $x \in G$ for which the minimal directed path from y to x has length less than or equal to r. The size-r *directional erosion* of G, $\underline{E}^{(r)}(G)$, consists of all nodes $y \in G$ such that there does not exist a node $x \notin G$ for which the minimal directed path from x to y has length less than or equal to r. The elementary directional dilation, $\underline{\Delta}$, and erosion, $\underline{E}$, are defined by letting $r = 1$. $\underline{\Delta}^{(r)} = \underline{\Delta}\underline{\Delta} \cdots \underline{\Delta}$ (r times) and $\underline{E} = \underline{E}\underline{E} \cdots \underline{E}$ (r times). As with undirected dilation and erosion, all genes in G are complete up to level r in $\underline{\Delta}^{(r)}(G)$, and all genes in $\underline{E}^{(r)}(G)$ are complete up to level r in G. The directed dilation has a property not satisfied for undirected dilation: if $\underline{\Delta}^{(r)}(G)$ is a sub-PBN, then $\underline{\Delta}^{(u)}(G) = \underline{\Delta}^{(r)}(G)$ for all $u \geqslant r$. The property does not hold for Δ because even if $\Delta^{(r)}(G)$ is closed, there may still be an edge whose back lies in $\Delta^{(r)}(G)$ but whose front lies outside $\Delta^{(r)}(G)$.

7. Probabilistic Boolean networks as many-sorted algebras

There are two graph structures associated with a PBN. Thus far we have been discussing the low-level graph consisting of genes and function-essentiality relations. The high-level graph concerns the states of the system and the transitions between the states. The present section concerns the algebraic structure of the state graph and related homomorphisms between PBNs.

Mathematical models are often composed of more than a single sort (type) of object. A rigorous understanding of the algorithmic simulation of activities within a model is enhanced by a precise delineation of the methodology by which sorts are operationally preserved and altered. Many-sorted algebras provide an algebraic framework for such delineation [14] (see [2] for a didactic introduction). In particular, the concept of a homomorphism provides a mathematical notion for the study of structure-preserving qualities of mappings between different algebras.

Formally, a *many-sorted algebra* is a seven-tuple $\mathscr{A} = (S, \alpha, \Sigma, \beta, \Omega, \gamma, G)$, where S is the set of *sorts*. The typification of both inter- and intra-sort operations is accomplished by means of signature sets. The names of the operators with n input sorts $s_1, s_2, \ldots, s_n$ and output sort s compose a *signature set*, denoted by $\Sigma(s_1, s_2, \ldots, s_n; s)$, which gives the names of n-ary operators. The names of 0-ary operators are contained in signature sets $\Sigma(\varepsilon; s)$, where ε denotes the absence of an input. Each element in a signature set is called an *operation symbol*. Σ is the class of all signature sets for $\mathscr{A}$. There is a signature set corresponding to each pair $(\mathbf{s}; s) = (s_1, s_2, \ldots, s_n; s) \in S^* \times S$; however, in practice, only a finite number of these sets will be nonempty. The mapping α maps pairs in $S^* \times S$ to corresponding signature sets: $\alpha : S^* \times S \to \Sigma$ by $\alpha(\mathbf{s}; s) = \Sigma(\mathbf{s}; s)$. The set of sorts and the class of signature sets provide a global view of structure but do not address specifics. Actual elements of the sorts are given in the *carrier sets*, there being a carrier set A_s for each sort s. Ω is the class of all carrier sets. β is the mapping from sorts to carrier sets, $\beta : S \to \Omega$ by $\beta(s) = A_s$. Finally, G is the set of all actual operators and γ maps the operational symbols (names) contained in the union of the signature sets into G:

$$\gamma : \bigcup_{S^* \times S} \Sigma(s_1, s_2, \ldots, s_n; s) \to G \tag{24}$$

such that for each $\sigma \in \Sigma(\varepsilon; s)$, $\gamma(\sigma) \in A_s$, and for $\sigma \in \Sigma(s_1, s_2, \ldots, s_n; s)$,

$$\gamma(\sigma) : A_{s_1} \times A_{s_2} \times \cdots \times A_{s_n} \to A_s. \tag{25}$$

To describe the state space of a PBN as a many-sorted algebra, we need to define the seven-tuple for the algebra. For a PBN, the set of sorts is given by

$$S = \{\text{states, inputs, values, probabilities}\}. \tag{26}$$

All signature sets are empty except the following:

$$\Sigma(\text{state, input, state; value}) = \{\text{transition}\}, \tag{27}$$

$$\Sigma(\text{input; probability}) = \{\text{selection}\}. \tag{28}$$

There is also $\Sigma(\varepsilon;\ \text{state}) = \{\text{initial}\}$, referring to initialization of the PBN, and $\Sigma(\varepsilon;\ \text{value}) = \{\text{zero, one}\}$ and Σ (ε; probability) giving the names of the values and probabilities (which is an infinite set of names), respectively. The carrier sets are given by

$$A_{\text{states}} = \{(a_1, a_2, \ldots, a_n) : a_k \in \{0, 1\}\} \tag{29}$$

$$A_{\text{inputs}} = \{(f_{1i_1}, f_{2i_2}, \ldots, f_{ni_n}) : f_{ki_k} \in B_{x_k}\} \tag{30}$$

$A_{\text{values}} = \{0, 1\}$, and $A_{\text{probabilities}} = [0, 1]$. Note that

$$\gamma(\text{transition}) : A_{\text{states}} \times A_{\text{inputs}} \times A_{\text{states}} \to A_{\text{values}} \tag{31}$$

by $\gamma(\text{transition}) = \delta$, where

$$\delta(\mathbf{x}, f_{1i_1}, f_{2i_2}, \ldots, f_{ni_n}), \mathbf{x}^+ = \begin{cases} 1 & \text{if } (f_{1i_1}, f_{2i_2}, \ldots, f_{ni_n})(\mathbf{x}) = \mathbf{x}^+, \\ 0 & \text{if } (f_{1i_1}, f_{2i_2}, \ldots, f_{ni_n})(\mathbf{x}) \neq \mathbf{x}^+. \end{cases} \tag{32}$$

Also,

$$\gamma(\text{selection}) : A_{\text{inputs}} \to A_{\text{values}} \tag{33}$$

by $\gamma(\text{selection}) = \rho$, where

$$\rho(f_{1i_1}, f_{2i_2}, \ldots, f_{ni_n}) = P(\mathbf{f} = (f_{1i_1}, f_{2i_2}, \ldots, f_{ni_n})). \tag{34}$$

Homomorphisms are structure-preserving mappings between many-sorted algebras. They are useful for recognizing when certain structures appear within other structures. A homomorphism between the many-sorted algebras $\mathscr{A}^1 = (S, \alpha, \Sigma, \beta^1, \Omega^1, \gamma^1, G^1)$ and $\mathscr{A}^2 = (S, \alpha, \Sigma, \beta^2, \Omega^2, \gamma^2, G^2)$, possessing the same set of sorts and class of signature sets, is a family $\mathscr{H} = \{h_s\}_{s \in S}$ of functions for which $h_s : A_s^1 \to A_s^2$ for any $s \in S$, and which preserves the operations. The preservation condition means that: (1) if $\sigma \in \Sigma(\varepsilon; s)$, then $h_s(\gamma^1(\sigma)) = \gamma^2(\sigma)$; and (2) if $\sigma \in \Sigma(s_1, s_2, \ldots, s_n; s)$, $(a_1, a_2, \ldots, a_n) \in A^1_{s_1} \times A^2_{s_2} \times \cdots \times A^n_{s_n}$, $\gamma^1(\sigma) = \sigma^1$, and $\gamma^2(\sigma) = \sigma^2$, then

$$\begin{aligned} & h_s(\sigma^1(a_1, a_2, \ldots, a_n)) \\ & \quad = \sigma^2(h_{s_1}(a_1), h_{s_2}(a_2), \ldots, h_{s_n}(a_n)). \end{aligned} \tag{35}$$

This equation means that the following diagram commutes:

$$\begin{tikzcd}
A^1_{s_1} \times A^1_{s_2} \times \cdots \times A^1_{s_n} \arrow[r, "\sigma^1"] \arrow[d, "h^1_{s_1} \times h^1_{s_2} \times \cdots \times h^1_{s_n}"'] & A^1_s \arrow[d, "h_s"] \\
A^2_{s_1} \times A^2_{s_2} \times \cdots \times A^2_{s_n} \arrow[r, "\sigma^2"] & A^2_s.
\end{tikzcd} \tag{36}$$

When applied to PBNs, there are two commutative diagrams corresponding to the diagram of Eq. (36). These arise from the signature sets of Eqs. (27) and (28). Letting A_s, A_i, A_v, and A_p denote the carrier sets for states, inputs, values, and probabilities, respectively, the two PBN diagrams are

$$\begin{tikzcd}
A^1_s \times A^1_i \times A^1_s \arrow[r, "\delta"] \arrow[d, "h_s \times h_i \times h_s"'] & A^1_v \arrow[d, "h_v"] \\
A^2_s \times A^2_i \times A^2_s \arrow[r, "\delta"] & A^2_v
\end{tikzcd} \tag{37}$$

$$\begin{tikzcd}
A^1_i \arrow[r, "\rho"] \arrow[d, "h_i"'] & A^1_v \arrow[d, "h_v"] \\
A^2_i \arrow[r, "\rho"] & A^2_v.
\end{tikzcd} \tag{38}$$

There are instances when projection is a PBN homomorphism. Consider deletion of gene y from the network $\{x, y, z, w\}$ when y is not essential for any gene except (perhaps) itself. Then the two induced functions of Eq. (4) reduce to a single function $g_k^{(x)}(x, z, w) = f_k^{(x)}(x, y, z, w)$. For the PBN diagrams, define h_s and h_i by

$$h_s(x, y, z, w) = (x, z, w), \tag{39}$$

$$h_i(f_k^{(x)}) = g_k^{(x)} \tag{40}$$

and let h_v and h_p be the identity. The first diagram commutes because

$$\delta((x, y, z, w), f_k^{(x)}, (x^+, y^+, z^+, w^+)) = 1 \tag{41}$$

if and only if

$$\delta((x, z, w), g_k^{(x)}, (x^+, z^+, w^+)) = 1. \tag{42}$$

The second PBN diagram commutes because

$$\rho(g_k^{(x)}) = \rho(f_k^{(x)}). \tag{43}$$

In general, projection is not a homomorphism. It is the lack of functional dependence on the deleted gene that results algebraic preservation in the present case.

8. Conclusion

Basic mappings required for the theoretical and application-oriented manipulation of PBNs have been characterized: projection, adjunction, resolution-reduction, and morphological mappings. PBN homomorphisms have been characterized in the framework of many-sorted algebras. Active research is focused on the application of these mappings to PBNs designed from gene-expression data: compressing complex networks and growing larger networks from small "seed networks" selected in accordance with biological knowledge. In addition to application-oriented research, further investigation of PBN mappings relative to both the graphical and probabilistic structures within PBNs is necessary, and, more generally, the investigation of mappings within the context of universal algebras may prove to be of benefit.

References

[1] E.R. Dougherty, S. Kim, Y. Chen, Coefficient of determination in nonlinear signal processing, Signal Processing 80 (10) (2000) 2219–2235.

[2] E.R. Dougherty, C.R. Giardina, Mathematical Methods for Artificial Intelligence and Autonomous Systems, Prentice-Hall, Englewood Cliffs, NJ, 1988.

[3] N. Friedman, M. Linial, I. Nachman, D. Pe'er, Using Bayesian networks to analyze expression data, J. Comput. Biol. 7 (2000) 601–620.

[4] A.J. Hartemink, D.K. Gifford, T. Jaakkola, R.A. Young, Using graphical models and genomic expression data to statistically validate models of genetic regulatory networks, Pacific Symposium on Biocomputing, Hawaii, January, 2001.

[5] H.J.A.M. Heijmans, P. Nacken, A. Toet, L. Vincent, Graph morphology, Visual Comm. Image Representation 3 (1) (1992) 24–38.

[6] S. Huang, Gene expression profiling, genetic networks, and cellular states: an integrating concept for tumorigenesis and drug discovery, J. Mol. Med. 77 (1999) 469–480.

[7] S.A. Kauffman, Origins of Order: Self-Organization and Selection on Evolution, Oxford University Press, Oxford, 1993.

[8] S. Kim, E.R. Dougherty, M.L. Bittner, Y. Chen, K. Sivakumar, P. Meltzer, J.M. Trent, A general framework for the analysis of multivariate gene interaction via expression arrays, Biomed. Opt. 4 (4) (2000) 411–424.

[9] K. Murphy, S. Mian, Modeling gene expression data using dynamic Bayesian networks, Technical Report, University of California, Berkeley, 1999.

[10] I. Shmulevich, E.R. Dougherty, S. Kim, W. Zhang, Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks, Bioinformatics 18 (2002) 261–274.

[11] I. Shmulevich, E.R. Dougherty, W. Zhang, Gene perturbation and intervention in probabilistic Boolean networks, Bioinformatics 18 (10) (2002) 1319–1331.

[12] I. Shmulevich, E.R. Dougherty, W. Zhang, Control of stationary behavior in probabilistic Boolean networks by means of structural intervention, J. Biol. Systems 10 (4) (2002) 431–445.

[13] R. Somogyi, C. Sniegoski, Modeling the complexity of Gene networks: understanding multigenic and pleiotropic regulation, Complexity 1 (1996) 45–63.

[14] A. Trybulec, Many-sorted algebras, Formalized Mathematics 6 (1994).

[15] L. Vincent, Graphs and Morphology, Signal Processing 16 (1989) 365–388.