

# The role of certain Post classes in Boolean network models of genetic networks

Ilya Shmulevich<sup>\*†</sup>, Harri Lähdesmäki<sup>\*‡</sup>, Edward R. Dougherty<sup>§</sup>, Jaakko Astola<sup>‡</sup>, and Wei Zhang<sup>\*</sup>

<sup>\*</sup>Cancer Genomics Laboratory, University of Texas M. D. Anderson Cancer Center, Houston, TX 77030; <sup>†</sup>Institute of Signal Processing, Tampere University of Technology, 33720, Tampere, Finland; and <sup>‡</sup>Department of Electrical Engineering, Texas A&M University, College Station, TX 77840

Communicated by Leo P. Kadanoff, University of Chicago, Chicago, IL, July 28, 2003 (received for review September 3, 2002)

**A topic of great interest and debate concerns the source of order and remarkable robustness observed in genetic regulatory networks. The study of the generic properties of Boolean networks has proven to be useful for gaining insight into such phenomena. The main focus, as regards ordered behavior in networks, has been on canalizing functions, internal homogeneity or bias, and network connectivity. Here we examine the role that certain classes of Boolean functions that are closed under composition play in the emergence of order in Boolean networks. The closure property implies that any gene at any number of steps in the future is guaranteed to be governed by a function from the same class. By means of Derrida curves on random Boolean networks and percolation simulations on square lattices, we demonstrate that networks constructed from functions belonging to these classes have a tendency toward ordered behavior. Thus they are not overly sensitive to initial conditions, and damage does not readily spread throughout the network. In addition, the considered classes are significantly larger than the class of canalizing functions as the connectivity increases. The functions in these classes exhibit the same kind of preference toward biased functions as do canalizing functions, meaning that functions from this class are likely to be biased. Finally, functions from this class have a natural way of ensuring robustness against noise and perturbations, thus representing plausible evolutionarily selected candidates for regulatory rules in genetic networks.**

**M**athematical and computational modeling is becoming increasingly important for understanding the complex dynamical interactions in genetic regulatory systems. In light of recent development of high-throughput genomic technologies, modeling efforts have the potential to move from theory into practice. The understanding of the underlying mechanisms used by genetic networks not only sets the stage for a systematic view of physiology and pathophysiology but also carries tremendous practical potential in the context of model inference from real measurement data. Because the implications of the logic of genetic networks are difficult to deduce solely by means of experimental techniques, computational and mathematical modeling play a vital role (1).

Since their inception in 1969 by Kauffman, random Boolean networks (2–5) have been one of the most intensively studied models of discrete dynamical systems, enjoying a sustained interest from both the biology and physics communities. Considering their structural simplicity, these systems are capable of displaying a remarkably rich variety of complex behaviors and share many features of other dynamical system models. Analytical and numerical studies of the relationships between structural organization and dynamical behavior of Boolean networks have yielded important insights into the overall behavior of large genetic networks (5), evolutionary principles (6–8), and the development of chaos (9, 10).

Of particular interest are the relationships between local properties of the network and its global behavior. Various parameters specifying these local properties can be “tuned” such that the network is operating in one of several different regimes. In the ordered regime, the system behaves in a simple way, with

most of its components being frozen. In this regime, the transfer of information between the components is impeded by large walls of frozen components. In the chaotic regime, the system behaves in the opposite way, with a perturbation of one component propagating to many others in an avalanche-like manner, with only a few isolated islands comprised of frozen components. Thus, networks in the chaotic regime are very sensitive to initial conditions and perturbations. The boundary between order and chaos is called the complex regime or the critical phase, with the networks undergoing a kind of phase transition (11). It is in this regime that the networks are most evolvable. Kauffman (5) argues that life must exist on the edge of chaos such that the networks representing real genetic regulatory networks operate at or near this critical phase. As Kauffman (12) puts it, “a living system must first strike an internal compromise between malleability and stability. To survive in a variable environment, it must be stable to be sure, but not so stable that it remains forever static.”

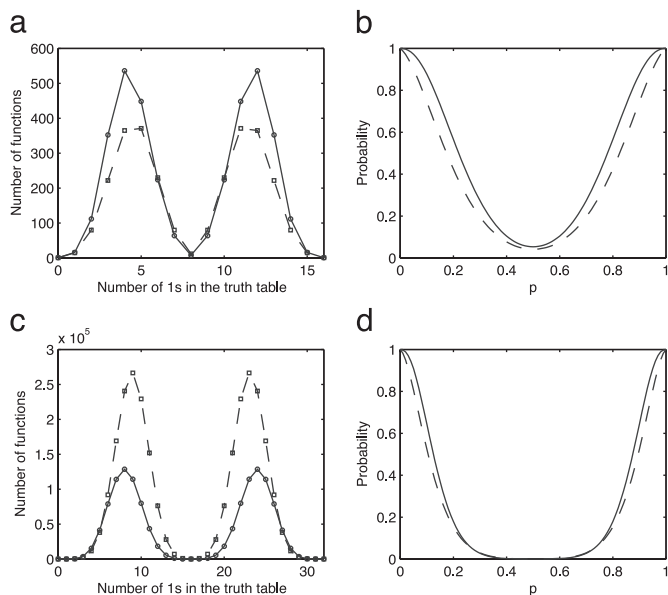
A Boolean network contains  $n$  elements (genes)  $\{x_1, \dots, x_n\}$ . Each gene  $x_i \in \{0, 1\}$  ( $i = 1, \dots, n$ ) is a binary variable, the value of which at time  $t + 1$  is completely determined by the values of some other genes  $x_{j_1(i)}, x_{j_2(i)}, \dots, x_{j_{k_i}(i)}$  at time  $t$  by means of a Boolean function  $f_i: \{0, 1\}^{k_i} \rightarrow \{0, 1\}$ . That is, there are  $k_i$  genes assigned to gene  $x_i$  and the mapping  $j_k: \{1, \dots, n\} \rightarrow \{1, \dots, n\}$ ,  $k = 1, \dots, k_i$  determines the “wiring” of gene  $x_i$ . Thus, we can write  $x_i(t + 1) = f_i(x_{j_1(i)}(t), x_{j_2(i)}(t), \dots, x_{j_{k_i}(i)}(t))$ . All genes are assumed to update synchronously in accordance with the functions assigned to them, and this process is then repeated. In a random Boolean network, the functions  $f_i$  are selected randomly, as are the genes used as its inputs.

Let us briefly consider several properties known to have an impact on the mode of operation of Boolean networks. The first two are the connectivity  $K$  and the function bias  $p$ .  $K = n^{-1} \sum_{i=1}^n k_i$  is the mean number of input variables used by the random Boolean functions  $f_i$ . We will assume in the sequel that  $k_i = K$  for all  $i = 1, \dots, n$ . The bias  $p$  of a random function  $f_i$  is the probability that it takes on the value 1. If  $p = 0.5$ , then the function is said to be unbiased. By varying these parameters, the network can be made to undergo a phase transition. For example, in the case of unbiased functions, the critical connectivity is  $K = 2$ , meaning that for  $K > 2$ , we observe chaotic behavior. In general, for a given bias  $p$ , the critical connectivity is equal to  $K_c = [2p(1 - p)]^{-1}$  (ref. 13). Alternatively, by inverting this equation, a critical bias  $p_c$  can be found for an arbitrary connectivity  $K$ . Strongly biased functions (when  $p$  is far away from 0.5) are said to have a high degree of internal homogeneity (8) and are associated with increased order in Boolean networks.

It has been known for quite some time that canalizing functions play a role in preventing chaotic behavior (2, 5, 14, 15). A canalizing function is one in which at least one of the input variables is able to determine the value of the output of the function regardless of the other variables. By increasing the percentage of canalizing functions in a Boolean network, one can

<sup>†</sup>To whom correspondence should be addressed. E-mail: is@iee.org.

© 2003 by The National Academy of Sciences of the USA



**Fig. 1.** (a and c) Subplots, with  $K = 4$  and  $5$ , respectively, show histograms of the number of canalizing (solid line) and  $A^2 \cup a^2$  (dashed line) functions versus the number of ones in their truth tables. (b and d) Subplots, with  $K = 4$  and  $5$ , respectively, show the probability that a randomly chosen Boolean function with bias  $p$  is a canalizing (solid line) or  $A^2 \cup a^2$  (dashed line) function.

move closer toward the ordered regime and, depending on the connectivity and the distribution of the number of canalizing variables in the randomly chosen canalizing functions (16), cross the phase-transition boundary. There is also strong evidence that many control rules governing transcription of eukaryotic genes are canalizing when viewed in the Boolean formalism (17).

Internal homogeneity (bias) and canalizing functions are, in fact, quite related. It is known that canalizing functions “prefer” a high degree of internal homogeneity (14). Consider the solid-line plots in Fig. 1 *a* and *c*. These plots show histograms of the number of canalizing functions of  $K = 4$  and  $5$  variables, respectively, versus the number of ones in their truth tables. Thus, for example, for  $K = 4$ , there are only 8 canalizing functions (of 3,514) that have exactly 8 ones in their truth table. The solid-line plots in Fig. 1 *b* and *d* show the probability that a randomly generated Boolean function with bias  $p$  will be a canalizing function (for  $K = 4$  and  $5$ , respectively). Again, it can be seen that for  $p = 0.5$  and  $K = 4$ , this probability is only  $\approx 0.0536$  (see ref. 14 for details). This phenomenon lends support to the observed role of canalizing functions in the formation of ordered behavior.

Although it is known that the average connectivity in real genetic networks, especially those of higher Metazoa, is considerably higher than  $K = 2$  (ref. 18), real genetic networks are clearly not chaotic. Thus, a majority of regulatory functions must either be biased or canalizing. Aldana *et al.* (19) recently posed a profound question regarding canalizing functions. Although it is known that the fraction of canalizing functions is very small as  $K$  increases, how (and, we would like to add, why) did evolution select them to act as genetic regulatory rules? The first answer that comes to mind might be that evolution acted, by means of selection, to produce regulatory functions that in their totality would ensure that the overall behavior of the network is not chaotic. We suggest that such a view is somewhat flawed. This would be akin to designing a complex digital circuit to be highly noise- and fault-tolerant by first designing the whole circuit so that it produces the correct output, then testing to see how noise-tolerant it really is, and if it is not, redesigning it again, and

repeating this process until a satisfactory circuit is produced. A more reasonable and sustainable approach would be to develop a systematic way to design noise tolerance from the bottom up, for example by building in redundancy, exactly as von Neumann (20) envisioned in 1956. Thus, the issue is not whether evolution acted by means of random selection but what criteria or fitness functions it used to arrive at canalizing rules. As Sawhill and Kauffman (21) write, “canalization would be a natural way to design in robustness against noise and uncertainty.” It thus seems more plausible that evolution acted locally (at the level of regulatory rules) to ensure robustness and reliability, whereas global order came for free, exactly as Kauffman has been asserting.

If canalizing functions indeed were found randomly and then selected, then it must have been quite a difficult task for evolution to find so few needles in such a large haystack. If a different class of functions, much more abundant in number than canalizing functions, could exhibit the same high fitness in terms of robustness and globally ordered behavior, it would have been much more likely for evolution to stumble on and select it. This possibility serves as one of the motivations of this article.

There is another issue, however. Canalizing functions, strictly speaking, constitute a class of functions in the sense that a given function either belongs or does not belong to this class. Biased functions, on the other hand, do not constitute a class. That is, because the definition of bias is a probabilistic one, given a bias  $0 < p < 1$ , any Boolean function can still be generated with a positive probability.<sup>†</sup> Thus, a  $p$ -biased random network, meaning one with functions that are all chosen randomly with a bias  $p$ , could potentially contain any Boolean function. Nonetheless,  $p$ -biased random Boolean networks possess an interesting property when the number,  $n$ , of genes is large, which is the case for real genetic regulatory networks.

Consider a transition of the network from state  $\mathbf{x}(t)$  to state  $\mathbf{x}(t + 1)$ , where  $\mathbf{x} \in \{0, 1\}^n$  is the state vector of genes. Clearly,  $n$  Boolean functions are used to make this transition. The same functions then are used again to move from state  $\mathbf{x}(t + 1)$  to state  $\mathbf{x}(t + 2)$ . This situation corresponds to the so-called quenched network model, meaning that the randomly selected functions are kept the same for the entire future operation of the network. We then could ask the question: What Boolean functions will take us directly from  $\mathbf{x}(t)$  to  $\mathbf{x}(t + 2)$ ? In other words, what Boolean functions will output  $\mathbf{x}(t + 2)$  when the input is  $\mathbf{x}(t)$ ? The answer, of course, is the composition of the Boolean functions defining the network. That is, these Boolean functions themselves are used as input variables to the same Boolean functions. Let us give a simple example for illustration.

**Example:** Suppose we have three genes,  $x_1$ ,  $x_2$ , and  $x_3$ . The corresponding three Boolean functions are given by  $f_1^{(1)}(x_1, x_2, x_3) = x_1x_2$ ,  $f_2^{(1)}(x_1, x_2, x_3) = x_1x_3 + x_2x_3$ , and  $f_3^{(1)}(x_1, x_2, x_3) = x_2x_3$ , where addition represents disjunction and multiplication represents conjunction. The superscripts (1) signify that these are one-step functions, meaning that they output  $\mathbf{x}(t + 1)$  given  $\mathbf{x}(t)$ . Let us compute the two-step functions,

$$\begin{aligned} f_1^{(2)}(x_1, x_2, x_3) &= f_1^{(1)}(f_1^{(1)}, f_2^{(1)}, f_3^{(1)}) = f_1^{(1)} \cdot f_2^{(1)} \\ &= (x_1x_2) \cdot (x_1x_3 + x_2x_3) = x_1x_2x_3 \\ f_2^{(2)}(x_1, x_2, x_3) &= f_1^{(1)} \cdot f_3^{(1)} + f_2^{(1)} \cdot f_3^{(1)} = x_2x_3 \\ f_3^{(2)}(x_1, x_2, x_3) &= f_2^{(1)} \cdot f_3^{(1)} = x_2x_3, \end{aligned}$$

<sup>†</sup>We can generate a random Boolean function with bias  $p$  by flipping a  $p$ -biased coin  $2^n$  times and thus filling in the truth table.

where we have skipped several straightforward algebraic steps.<sup>||</sup>

When  $n \rightarrow \infty$ , in the quenched model, with every step of the network, each new composition of functions results with high probability in a new Boolean function with even more inputs (taken from time 0). Because more and more functions, which were initially selected at random, are used to form the overall composition function, the latter is also random. This insight was used to propose the so-called annealed model (13), where at each step of the network a completely new random function is selected for each gene. It turns out that the quenched and annealed models exhibit very similar dynamics of state transitions (5, 13).

It is fairly straightforward to show that within the annealed approximation, if the one-step Boolean functions are random  $p$ -biased functions, then the two-step Boolean functions are also random  $p$ -biased functions. This is so because the functions at the second step are chosen independently of the state of the network after the first step. Clearly, by induction, this implies that three-step functions and beyond are all  $p$ -biased as well. Thus, it could be said that within the annealed approximation,  $p$ -biased functions have a type of closure property in the sense that compositions of  $p$ -biased random functions themselves are  $p$ -biased random functions. The implication of this result is that in an annealed random Boolean network consisting of  $p$ -biased functions, the probability that any given gene is equal to 1 is equal to  $p$  at every step of the network. The above  $p$ -biased closure property approximately holds true for large  $n$  in the quenched-model setting.

By contrast, the class of canalizing functions is not closed; that is, a composition of canalizing functions is not necessarily a canalizing function. This implies that a rule specifying a gene value several steps into the future is not necessarily a canalizing rule even if the network consists entirely of canalizing rules. Kauffman (8) recognized this issue and wrote that the necessary condition for such transitivity to hold is that “guaranteed values [of inputs and outputs] must propagate down a connected forcing structure.” In other words, the canalized value of one gene must also be its canalizing value when that gene is used as a canalizing variable in a function determining another gene. Thus, a second motivation of this article is to arrive at a class of Boolean functions that would be closed under composition analogously to the similar property of  $p$ -biased annealed networks but without requiring any additional assumptions such as the annealed approximation.

Let us summarize. We are looking for a class of Boolean functions such that:

1. it is much larger than the class of canalizing functions;
2. an abundance of functions from this class will tend to prevent chaotic behavior in networks;
3. it will be “compatible” with internal homogeneity much in the same way as are canalizing functions, meaning that functions from this class are likely to be biased;
4. functions from this class will have a natural way to ensure robustness against noise and uncertainty; and
5. it will be closed under the operation of composition, meaning that functions specifying the network state an arbitrary number of time steps into the future will also belong to this class, in the usual quenched-model setting.

### Post Classes

In 1921, the American mathematician Emil Post (22, 23) characterized all classes of Boolean functions that are closed under composition. Several well known classes of Boolean functions

such as monotone, linear, and self-dual functions fall into this category. For example, it is relatively straightforward to show that a composition of an arbitrary number of monotone Boolean functions results in a monotone Boolean function. Of course, the class of all Boolean functions is trivially a Post class as well. However, the class that we will consider is of a somewhat more mysterious nature.

To motivate the definition and make the connection with canalizing functions, recall that a function  $f: \{0,1\}^n \rightarrow \{0,1\}$  is said to be canalizing if there exists an  $i \in \{1, \dots, n\}$  and  $u, v \in \{0,1\}$  such that for all  $x_1, \dots, x_n \in \{0,1\}$ , if  $x_i = u$  then  $f(x_1, \dots, x_n) = v$ . The input variable  $x_i$  is called the canalizing variable with canalizing value  $u$  and canalized value  $v$ . Equivalently, all vectors  $x \in \{0,1\}^n$  for which  $f(x) = \bar{v}$  must have  $x_i = \bar{u}$ . When  $u = v = 0$ , this condition reduces to the Post class, \*\* which we will denote as  $A^\infty$ . Thus,  $A^\infty$  can be defined as follows.

**Definition 1:** A Boolean function  $f$  belongs to class  $A^\infty$  if all vectors on which the function takes on the value 1 have a common component equal to 1. Similarly, we define the Post class  $a^\infty$  by replacing ones by zeros in the previous sentence.

For example, the function  $f(x_1, x_2, x_3) = x_1 + x_2x_3$  belongs to  $a^\infty$ , because all the vectors on which the function is equal to 0, namely (000), (001), and (010) have a common component equal to 0 (the first one). In this example, the first variable is canalizing and  $u = v = 1$ . In other words, if  $x_1 = 1$ , then  $f(x_1, x_2, x_3) = 1$ . For simplicity, we will also say “ $f$  is  $a^\infty$ .” If we denote the set of canalizing functions by  $C$ , then it is clear from the above definitions that  $A^\infty \cup a^\infty \subseteq C$ . The class  $A^\mu$  is defined as follows.

**Definition 2:** A Boolean function  $f$  belongs to class  $A^\mu$ ,  $\mu \geq 2$ , if any  $\mu$  vectors on which the function takes on the value 1 have a common component equal to 1 (some of these  $\mu$  vectors may be repeated). The class  $a^\mu$  is defined by replacing ones by zeros in the previous sentence.

As an example, the function

$$f(x_1, x_2, x_3) = x_1x_2 + x_2x_3 + x_1x_3 \quad [1]$$

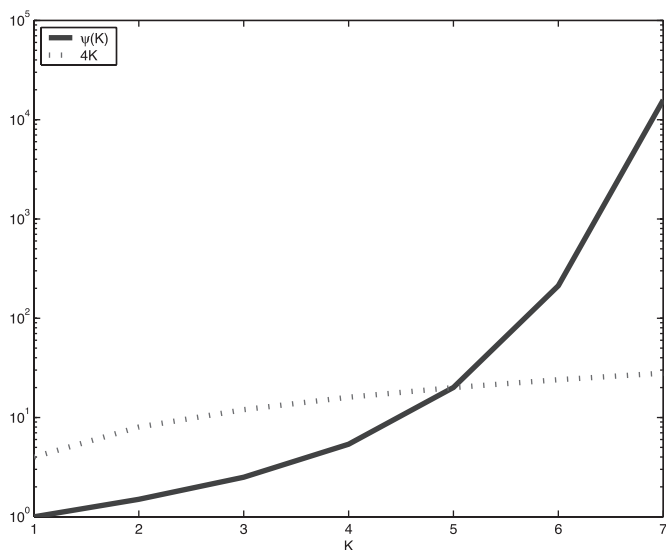
is  $A^2$ , because any two vectors on which the function is equal to 1 have a common unity component. It is easy to see that if  $f$  is  $A^\mu$ , then it is also  $A^\nu$  for any  $\nu$  satisfying  $2 \leq \nu \leq \mu$ . A similar result holds for  $a^\mu$ . Thus, these classes are nested with  $A^2$  (respectively,  $a^2$ ) being the largest and  $A^\infty$  (respectively,  $a^\infty$ ) being the smallest. Another interesting result is that if the function  $f: \{0, 1\}^n \rightarrow \{0, 1\}$  is  $A^\mu$ , then for  $\mu \geq n$  it is also  $A^\infty$  (ref. 24).

It is easy to show that if  $f$  is  $A^\mu$ , then any collection of  $m \leq \mu$  terms in the disjunctive normal form must contain a variable that belongs to each one of those terms. Thus, once again there is an interesting relationship to canalizing functions in the sense that the functions defined by those terms are themselves canalizing functions. For example, if we take any two terms from the function in Eq. 1, such as  $x_1x_2 + x_2x_3$ , then the Boolean function corresponding to these terms is canalizing, with canalizing variable  $x_2$ .

Because the classes  $a^\infty$ ,  $A^\infty$ ,  $a^\mu$ , and  $A^\mu$  are all Post classes, they are closed under the operation of composition. This implies that a Boolean network constructed from one of these classes has the remarkable property that any  $t$ -step transition function (outputting the gene value  $t$  time steps into the future) also belongs to this class much like the property of  $p$ -biased annealed networks. Let us now look at the relationship to bias more closely.

\*\*Typically, notation such as  $(A^\infty)$  is reserved to denote the property of a Boolean function. The original notation used by Post and subsequent authors (24) to denote the set of functions with such a property is, for example,  $F_{\bar{g}}$ . However, in this article we will simply use the simplified notation such as  $A^\infty$  to denote the set of functions with the property  $(A^\infty)$ .

||It is also interesting to note (details not shown) that the three-step functions are equal to the two-step functions  $f^{(3)} = f^{(2)}$ ,  $i = 1, 2, 3$ . This implies that, regardless of the starting state, a fixed-point attractor is reached in no more than three steps.



**Fig. 2.** A comparison between  $4K$  and  $\psi(K)$  for  $K = 1, \dots, 7$ . A logarithmic scale is used. Because  $\psi(K)$  grows much faster than  $4K$ , the number of  $A^2$  functions grows much faster than the number of canalizing functions.

Consider Fig. 1 once again and recall that the solid-line plots refer to canalizing functions. The dashed-line plots in the same figure have been constructed from the class  $A^2 \cup a^2$  for the same number of variables. We can notice a striking similarity between the solid- and dashed-line plots and can readily conclude that functions from  $A^2$  or  $a^2$  are also more likely to be biased, as suggested by the two-peaked histograms in Fig. 1 *a* and *c*. Similarly, Fig. 1 *b* and *d* show a marked tendency for a  $p$ -biased random Boolean function to be either  $A^2$  or  $a^2$  with increasing  $p$  (dashed line). A similar relationship exists for  $A^\mu \cup a^\mu$ ,  $\mu > 2$ . Thus, it seems that the functions in  $A^2$  or  $a^2$  behave very similarly to canalizing functions in that they both prefer internal homogeneity. This is promising from the point of view of network behavior in terms of order and chaos. Had the situation been the opposite, namely, if functions in  $A^2$  or  $a^2$  were unlikely to be biased, then networks constructed from them would have been more likely to behave chaotically, especially for  $K > 2$ . However, before discussing this, let us address the question of the cardinality of these classes, especially as it compares to that of the class of canalizing functions.

### Sizes of the Classes

As we discussed earlier, the proposed class of functions should be significantly larger than the class of canalizing functions under typical connectivity of real genetic networks for it to be a more evolutionarily plausible alternative. It is known (19) that the number of canalizing functions of  $K$  input variables is upper-bounded by  $4K \cdot 2^{2^{K-1}}$ . At the same time, the exact number of  $A^2$  functions is known to be  $\psi(K)2^{2^{K-1}}$ , where the value  $\psi(K)$  is related to the number of self-dual monotone Boolean functions with  $K$  variables (25). Thus, for us to compare the number of canalizing functions with the number of  $A^2$  functions, it suffices to compare  $4K$  with  $\psi(K)$  from the two formulas shown above. Fig. 2 compares these two values for  $K = 1, \dots, 7$  on a logarithmic scale. It is clear that although  $4K$  is only linear,  $\psi(K)$  grows at a superexponential rate.

The cardinality of  $a^2$  is equal to that of  $A^2$ . Thus, we can say that the classes  $A^2$  and  $a^2$  are asymptotically much larger than the class of canalizing functions. For  $\mu \geq 3$  (including  $\mu = \infty$ ) and  $K \rightarrow \infty$ , we have the asymptotic formula  $|A^\mu| = |a^\mu| \sim K \cdot 2^{2^{K-1}}$  (ref. 26), which is of the same order as the upper bound for the

**Table 1.** A comparison of the number of  $A^2$  and canalizing functions for  $K = 3, 4$ , and 5

	K		
	3	4	5
$A^2$ functions	40	1,376	1,314,816
Canalizing functions	120	3,514	1,292,276

number of canalizing functions.<sup>††</sup> For  $K = 3, 4$ , and 5, the exact numbers of canalizing and  $A^2$  functions are shown in Table 1. The number of  $A^2$  functions begins to exceed the number of canalizing functions already for  $K = 5$ .

### Dynamical Behavior of Networks Constructed from the Considered Post Classes

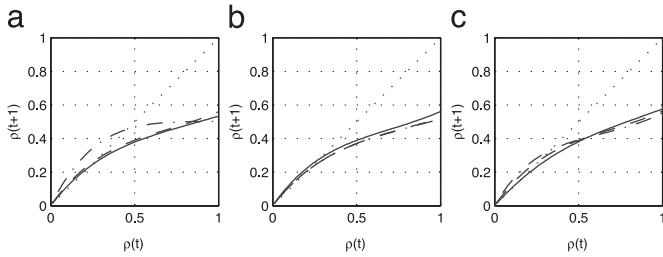
One of our requirements is that networks constructed from functions belonging to the considered Post classes should exhibit a tendency for ordered behavior. We have examined this question from several different points of view.

First, let us consider so-called Derrida curves (13), which are constructed as follows. Let  $\mathbf{x}^{(1)}(t)$  and  $\mathbf{x}^{(2)}(t)$  be two randomly chosen states at some time  $t$ . The normalized Hamming distance between them is  $\rho(t) = (1/n) \sum_{i=1}^n (x_i^{(1)}(t) \oplus x_i^{(2)}(t))$ , where  $\oplus$  is addition modulo 2. Then, given a random Boolean network realization, let  $\mathbf{x}^{(1)}(t+1)$  and  $\mathbf{x}^{(2)}(t+1)$  be the successor states of  $\mathbf{x}^{(1)}(t)$  and  $\mathbf{x}^{(2)}(t)$ , respectively. Similarly, let  $\rho(t+1)$  be the Hamming distance between these successor states. The Derrida curve consists of plotting  $\rho(t+1)$  versus  $\rho(t)$  and averaging over many random networks. If the network is operating in the chaotic regime, then small Hamming distances tend to diverge, and the Derrida curve lies above the main diagonal for small initial Hamming distances. This also implies that small gene perturbations (i.e., nearby states) tend to spread further apart and networks are sensitive to initial conditions, which is a hallmark of chaotic behavior. On the other hand, networks in the ordered regime exhibit convergence for nearby states, with the Derrida curve lying below the main diagonal. The more the Derrida curve lies above the main diagonal for small values of  $\rho(t)$ , the more chaotic the network is.

Consider Fig. 3, where we have plotted the Derrida curves corresponding to networks constructed from several different classes of functions. In Fig. 3*a*, the dash-dot line plot corresponds to random (and unbiased) Boolean networks for  $K = 4$ . As can be seen, the Derrida plot is significantly above the main diagonal for small  $\rho(t)$ , indicating that  $K = 4$  random Boolean networks behave chaotically. In the same plot, the solid line corresponds to networks constructed from canalizing functions, and the dashed line corresponds to networks constructed from  $A^2$  and  $a^2$  functions, also for  $K = 4$ . In both cases, these networks behave much less chaotically. In Fig. 3*b*, we compared the Post classes  $A^2, A^3$ , and  $A^4 = A^\infty$  for  $K = 4$ . It is evident that random networks constructed from  $A^3$  and  $A^4$  (which for the case of  $K = 4$  differ only by two functions) exhibit even more ordered behavior than those constructed from  $A^2$ . This is intuitively clear, because  $A^4$  is a more restrictive condition than  $A^2$ . Finally, in Fig. 3*c*, we compared the behavior of networks constructed from  $A^2$  functions for  $K = 3, 4$ , and 5. As expected, networks with higher values of  $K$  behave relatively more chaotically than those with lower values of  $K$ .

We also consider the effect of these Post classes on the dynamical behavior of Boolean networks in terms of percolation (28–31). Here we consider genes to be placed on a square lattice.

<sup>††</sup>There is actually an exact formula for the number of functions in  $A^\infty$ :  $\sum_{j=1}^K (-1)^{j+1} \binom{K}{j} 2^{2^{K-j}}$  for any number of variables  $K$  (27).



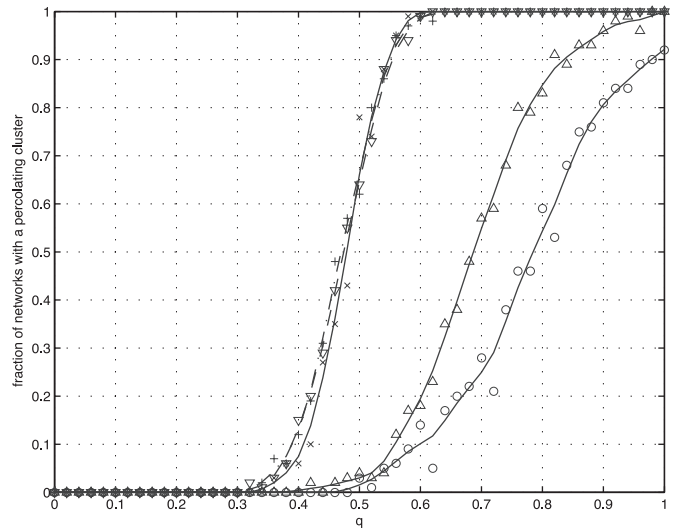
**Fig. 3.** Derrida curves corresponding to several classes. (a) Canalizing functions (solid line), the class  $A^2 \cup a^2$  (dashed line), and random Boolean functions (dash-dot line) ( $K = 4$ ). (b) Post classes  $A^2$  (solid line),  $A^3$  (dashed line), and  $A^4 = A^\infty$  (dash-dot line) ( $K = 4$ ). (c) Post classes  $A^2$  for  $K = 3$  (solid line), 4 (dashed line), and 5 (dash-dot line).

Unlike in random Kauffman networks, where the wiring for each gene is chosen randomly, on a square lattice, every gene receives inputs from its four nearest neighbors via a randomly chosen Boolean function. Thus,  $K = 4$ , as considered above. In percolation problems, one typically studies the fraction of points (sites) that must be occupied such that a continuous path of nearest neighbors is formed from one side to the other. Let  $r$  be the probability of site occupancy independent of all other sites. A percolation threshold is a critical probability  $r_c$  at which a spanning cluster of occupied sites suddenly begins to form and propagate infinitely. Thus, for  $r \geq r_c$ , a single spanning cluster of occupied sites exists, whereas for  $r < r_c$ , no spanning cluster exists. The transition from no spanning cluster to a spanning cluster is also a type of phase transition. For example, for the two-dimensional square-lattice model, the percolation threshold is  $r_c \approx 0.592746$ .

In practice, lattices are finite, and thus the typical approach is to find the probability at which a horizontally (or vertically) spanning cluster emerges. This is done by Monte Carlo simulations with many randomly generated lattices and computing what fraction of those contains a horizontally spanning (percolating) cluster. For Boolean networks on square lattices, instead of varying the occupancy probability  $r$ , one can, for instance, vary the bias  $p$  of the random Boolean functions. Then, one can select a subset of genes according to some criterion and ask whether those genes percolate through the lattice. For example, to study the dynamical behavior of networks, we can ask whether the set of frozen genes (i.e., genes that eventually stop changing values) forms a percolating cluster through the network. It has been shown that the critical bias  $p_c$  for Boolean networks on square lattices is approximately equal to 0.72 (14, 29). Thus, if each function in the random Boolean network is biased with  $p > 0.72$ , then there exists a percolating cluster of frozen genes; for  $0.5 \leq p < 0.72$ , such a cluster does not form.<sup>\*\*</sup>

We can use these ideas to study the dynamical properties of networks constructed from various classes as follows. Rather than varying the bias  $p$  of the random Boolean functions, we can vary the probability that a random Boolean function will belong to a given class. For example, we can generate random networks in which 70% of the functions are guaranteed to be canalizing and the other 30% are chosen randomly from the set of all functions. Then, we can repeat this procedure many times and record the fraction of such networks in which a frozen cluster of genes percolates. We have performed such experiments with the classes  $A^2$ ,  $A^3$ ,  $A^4 = A^\infty$ ,  $A^2 \cup a^2$ , and canalizing. The percolation results are shown in Fig. 4. The horizontal axis shows the probability  $q$  that a randomly selected function belongs to a given

<sup>\*\*</sup>We note that the situation for  $p < 0.5$  is symmetric, and the “polarity” of the bias  $p$  for each function can be different. In other words, some functions might be biased up (favoring 1), and some might be biased down (favoring 0).



**Fig. 4.** Percolation results of random networks constructed by using several classes of Boolean functions. The horizontal axis shows the fraction of functions belonging to a given class. The vertical axis shows the fraction of networks that contain a spanning (percolating) cluster of frozen genes. The classes are  $A^2$  ( $\times$ ),  $A^3$  ( $\nabla$ ),  $A^4$  ( $+$ ), canalizing ( $\circ$ ), and  $A^2 \cup a^2$  ( $\Delta$ ). The size of the lattice is  $50 \times 50$  (2,500 genes). For each  $q$  (51 equally spaced values), 100 networks were constructed for computing the fraction of those containing percolating clusters.

class. The vertical axis shows the fraction of networks for which a frozen cluster of genes percolates through the network.

In the two-dimensional square-lattice framework, we can see that the Post classes  $A^2$ ,  $A^3$ , and  $A^4 = A^\infty$  behave in the most orderly fashion. Even for  $\approx 60\%$  of functions belonging to one of these classes, the Boolean network is virtually guaranteed to contain a spanning cluster of frozen genes. The situation for canalizing functions is markedly different. Even at  $q = 1$ , when all functions are guaranteed to be canalizing, frozen genes percolate only  $\approx 92\%$  of the time. When the networks are constructed from  $A^2$  or  $a^2$  functions, they behave more orderly than canalizing networks for the same  $q$ .

An interesting observation is that  $A^2$  networks percolate at a lower value of  $q$  than  $A^2 \cup a^2$  networks. Although both classes contain mostly biased functions (as does the class of canalizing functions), the reason behind the more ordered behavior of  $A^2$  (as well as  $A^3$  and  $A^4$ ) networks relative to  $A^2 \cup a^2$  networks on the square lattice is rooted in the fact that the former represent a considerably stronger constraint than the latter in terms of “information transfer.” Consider a state  $\mathbf{x}$  of the lattice that has at least one gene ON (i.e., it is not all-zero). If all genes are governed by  $A^2$  functions, then it follows that all pairs of “parent” states (that is, those states that lead directly to  $\mathbf{x}$  in one step) must share at least one common gene that is ON. In fact, for every gene that is ON in  $\mathbf{x}$ , every pair of parent states must share at least one neighbor of that gene that is ON. The same can be said about all pairs of states at exactly two steps before  $\mathbf{x}$  due to the closure property, and so on. This represents a formidable constraint that does not need to hold for  $A^2 \cup a^2$  networks.

## Discussion

Boolean functions from the Post classes that we have considered here represent plausible evolutionarily selected candidates for regulatory rules in genetic networks. These classes are compatible with a number of natural requirements. As we have demonstrated by means of Derrida curves on random Kauffman networks and percolation simulations on square lattices, networks constructed from functions belonging to these classes have

a tendency toward ordered behavior. Thus, such networks with connectivity higher than 2, when purely random unbiased networks behave chaotically, can still avoid chaotic behavior. They are not overly sensitive to initial conditions, and damage does not spread throughout the network. On a square-lattice model, such networks exhibit a spanning cluster of frozen genes and isolated islands of twinkling genes that are functionally isolated from each other (5).

These classes, in particular  $A^2$  and its dual class  $a^2$ , are significantly larger than the class of canalizing functions for realistic  $K$ . In a sense, functions from these classes can be viewed as “softer” versions of canalizing functions. The parameter  $\mu$  in  $A^\mu$  and  $a^\mu$  then is a way of tuning the canalizing property, with  $A^\infty$  and  $a^\infty$  being the most restrictive classes that, in fact, are subsets of the class of canalizing functions.

These Post classes exhibit the same kind of preference toward biased functions as do canalizing functions. In fact, the two peaks of the histograms in Fig. 1 *a* and *c* actually correspond to  $A^2$  (left peaks) and  $a^2$  (right peaks). Thus, we can conclude that functions from the Post class  $A^2$  prefer low bias (toward 0), whereas functions from  $a^2$  prefer high bias (toward 1). It is tempting to speculate that, because the majority of genes are typically unexpressed under any given condition, the regulatory rules are preponderantly from  $A^\mu$  rather than  $a^\mu$ .

The additional remarkable property of the Post classes that we considered here is the closure property. Any gene at any number of steps in the future is guaranteed to be governed by a function from the same class, much like the composition of  $p$ -biased random Boolean functions itself is a  $p$ -biased function in the annealed approximation. Canalizing functions do not possess this property. Obviously, this property carries with it a strong constraint and, consequently, a tendency toward ordered behavior in networks. We emphasize that the closure property is intrinsic to Post classes and holds true in the usual quenched-model setting. Another interesting implication of the closure property relates to the connection between “model time” and actual physical time. In the context of genetic regulatory networks, Boolean networks, which are discrete-time systems, are idealizations of the real continuous-time systems that they represent. For instance, if we were to fit the models to real gene-expression data, we would typically sample the cell system at consecutive points in time (either uniformly or nonuniformly) and use the time-series data to infer the Boolean functions in the network. Therefore, there is an implicit albeit unspecified mapping between the discrete-model time and the actual physical time of the cells. If genes are regulated by rules belonging to the

considered Post classes, then regardless of how many discrete time steps correspond to the actual physical time interval (we emphasize that this mapping need not even be time-invariant), the overall rule specifying the gene value at the end of the physical time interval still belongs to the same Post class. Thus, the plausibility of the Post classes representing gene regulatory mechanisms does not depend on the relationship between model time and physical time.

But what of robustness against noise and uncertainty? Canalizing functions are good candidates indeed for control networks in the presence of noise or misinformation (21). Are the classes  $A^\mu$  and  $a^\mu$  ( $\mu = 2, \dots, \infty$ ) really equipped with such a mechanism, which evolution would be inclined to use as a fitness criterion? The problem of reliable computation with unreliable components in the Boolean circuit setting dates back to von Neumann (20). The relevant characteristic of a reliable circuit is redundancy. Intuitively, higher redundancy ensures reliability. The aim is to construct such circuits that, with high probability, can compute the correct function even when their components are noisy.

In the 1960s, it was discovered that the Post classes  $A^\mu$  and  $a^\mu$  play a significant role in the synthesis and reliability of control systems and self-correcting circuits (32, 33). These results rest on the concept of basis functions. Any closed class of Boolean functions contains a set of basis functions such that any function in this class can be constructed by iterative composition of these basis functions. For example, because the class of all Boolean functions is a closed class (Post class), one possible set of basis functions is  $\{x \cdot y, \bar{x}\}$  (i.e., conjunction and negation). Thus, any Boolean function can be constructed by repeated composition of these two functions. The basis set for the class  $A^2$  is  $\{x \cdot \bar{y}, xy + xz + yz\}$  (24). The second function is the well known majority (or three-point median) function. These functions are essentially building blocks from which any function in  $A^2$  can be constructed. By cascading such building blocks a sufficient number of times, in a tree-like fashion, it can be shown (33) that any Boolean function can be realized in a reliable manner, meaning that the correct output is guaranteed even when some of the components are faulty. Thus, Post classes  $A^\mu$  and  $a^\mu$  are strongly implicated in fault-tolerant Boolean circuit synthesis. It thus can be speculated that evolution used a similar mechanism to give rise to highly resilient and robust genetic networks.

We are grateful to Stuart A. Kauffman and Maximino Aldana for careful reading of this manuscript, stimulating discussions, and useful suggestions.

- Hasty, J., McMillen, D., Isaacs, F. & Collins, J. J. (2001) *Nat. Rev. Genet.* **2**, 268–279.
- Kauffman, S. A. (1969) *J. Theor. Biol.* **22**, 437–467.
- Kauffman, S. A. (1969) *Nature* **224**, 177–178.
- Kauffman, S. A. (1974) *J. Theor. Biol.* **44**, 167–190.
- Kauffman, S. A. (1993) *The Origins of Order: Self-Organization and Selection in Evolution* (Oxford Univ. Press, New York).
- Stern, M. D. (1999) *Proc. Natl. Acad. Sci. USA* **96**, 10746–10751.
- Bornholdt, S. & Sneppen, K. (2000) *Proc. R. Soc. London Ser. B* **266**, 2281–2286.
- Kauffman, S. A. (1990) *Physica D* **42**, 135–152.
- Glass, L. & Hill, C. (1998) *Europhys. Lett.* **41**, 599–604.
- Bhattacharjya, A. & Liang, S. (1996) *Phys. Rev. Lett.* **77**, 1644–1647.
- Kadanoff, L. P. (2000) *Statistical Physics: Statics, Dynamics and Renormalization* (World Scientific, Singapore).
- Kauffman, S. A. (1995) *At Home in the Universe* (Oxford Univ. Press, New York).
- Derrida, B. & Pomeau, Y. (1986) *Europhys. Lett.* **1**, 45–49.
- Stauffer, D. (1987) *J. Stat. Phys.* **46**, 789–794.
- Lynch, J. F. (1995) *Rand. Struct. Algorithms* **6**, 239–260.
- Kauffman, S. A. (2000) *Investigations* (Oxford Univ. Press, New York).
- Harris, S. E., Sawhill, B. K., Wuensche, A. & Kauffman, S. (2002) *Complexity* **7**, 23–40.
- Arnone, M. I. & Davidson, E. H. (1997) *Development (Cambridge, U.K.)* **124**, 1851–1864.
- Aldana, M., Coppersmith, S. & Kadanoff, L. P. (2002) in *Perspectives and Problems in Nonlinear Science*, eds. Kaplan, E., Marsden, J. E. & Sreenivasan, K. R. (Springer, New York), pp. 23–89.
- von Neumann, J. (1956) in *Automata Studies*, eds. Shannon, C. E. & McCarthy, J. (Princeton Univ. Press, Princeton), pp. 329–378.
- Sawhill, B. K. & Kauffman, S. A. (1997) *Santa Fe Institute Working Paper 97-05-038* (Santa Fe Inst., Santa Fe, NM).
- Post, E. (1921) *Am. J. Math.* **43**, 163–185.
- Post, E. (1941) *Two-Valued Iterative Systems of Mathematical Logic* (Princeton Univ. Press, Princeton).
- Yablonsky, S. V., Gavrilov, G. P. & Kudryavtsev, V. B. (1966) *Functions of the Algebra of Logic and Post Classes* (Nauka, Moscow).
- Pogosyan, G., Miyakawa, M., Nozaki, A. & Rosenberg, I. G. (1997) *IEICE Trans. Fundam.* **E80-A**, 1502–1507.
- Korshunov, A. D. (1988) *Trudy Inst. Mat. (Novosibirsk)* **10**, 159–204.
- Iovovich, V. & Kilibarda, G. (1999) *Diskret. Mat.* **11**, 127–138; trans. (1999) *Discrete Math. Appl.* **9**, 593–605.
- Derrida, B. & Stauffer, D. (1986) *Europhys. Lett.* **2**, 739–745.
- Stauffer, D. (1988) *J. Theor. Biol.* **135**, 255–261.
- Weisbuch, G. & Stauffer, D. (1987) *J. Phys.* **48**, 11–18.
- Hansen, A. (1988) *J. Phys. A Math. Gen.* **21**, 2481–2486.
- Muchnik, A. A. & Gindikin, S. G. (1962) *Dokl. Akad. Nauk SSSR* **144**, 1007–1010.
- Kirienko, G. I. (1964) *Probl. Kibernet.* **12**, 29–37.