RESEARCH ARTICLE

Modeling the Complexity of Genetic Networks: Understanding Multigenic and Pleiotropic Regulation

ROLAND SOMOGYI AND CAROL ANN SNIEGOSKI

Laboratory of Neurophysiology, NINDS, National Institutes of Health, Bldg. 36/2C02, Bethesda, MD 20892 USA, e-mail: rolands@helix.nih.gov

Received October 6, 1995; accepted February 6, 1996

Molecular genetics presents an increasingly complex picture of the genome and biological function. Evidence is mounting for distributed function, redundancy, and combinatorial coding in the regulation of genes. Satisfactory explanation will require the concept of a parallel processing signaling network. Here we provide an introduction to Boolean networks and their relevance to present-day experimental research. Boolean network models exhibit global complex behavior, self-organization, stability, redundancy and periodicity, properties that deeply characterize biological systems. While the life sciences must inevitably face the issue of complexity, we may well look to cybernetics for a modeling language such as Boolean networks which can manageably describe parallel processing biological systems and provide a framework for the growing accumulation of data. We finally discuss experimental strategies and database systems that will enable mapping of genetic networks. The synthesis of these approaches holds an immense potential for new discoveries on the intimate nature of genetic networks, bringing us closer to an understanding of complex molecular physiological processes like brain development, and intractable medical problems of immediate importance, such as neurodegenerative disorders, cancer, and a variety of genetic diseases. © *1996 John Wiley & Sons Inc.*

Key Words: *Boolean networks, attractors, genetic networks, gene expression, molecular signaling, development, molecular evolution, biological rhythms*

INTRODUCTION

This article is written with the intention of facilitating communication between experimental and theoretical approaches to understanding complex living systems. Far from representing a departure from reductionism, the systems approach to life science attempts to reduce observed phenomena to higher level dynamic organizational structures. Since the goal of science is to provide "algorithmic compression," we may well ask ourselves what mechanisms underlie order in complex living systems.

While this general synthetic view has been held by researchers for some time [30, 31, 46, 64, 65], the paucity of experimental data and lack of computing power to calculate complex models has slowed progress in this direction. Today, great advances in molecular biological techniques have opened the window to observing living organisms on their fundamental organizational level, the expression of genes. This perspective is especially important to the understanding of development, in which the information coded in the DNA is decompressed into the multiple molecular dimensions

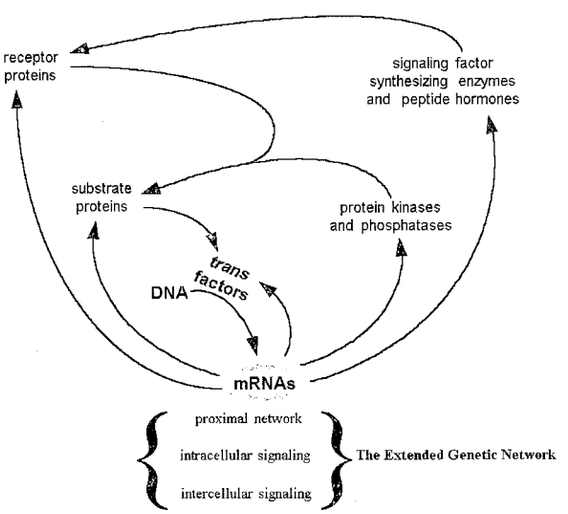
CCC 1076-2787/96/060045-19

of the phenotype. The overwhelming experimental data evidencing complex and yet opaque multigenic and pleiotropic regulation perhaps provides us with more new questions than answers. However, if we consider combining the quickly advancing technologies of computing and molecular biology, we may find ourselves challenged with the unique opportunity of making the complexity of life more transparent. Here we discuss Boolean networks[1] as a beginning framework for creating an abstract computational language that may allow us to describe the foundation of complex living systems, genetic networks. The depth of description was chosen to address a mixed audience of theoretically and experimentally oriented readers.

DEFINITION OF A GENETIC NETWORK

The information for constructing and maintaining the molecular components of a living organism essentially lies within its genes.[2] Genes directly encode the proteins which make up cells and synthesize all other building blocks and signaling molecules necessary for life. During development, while the organism is largely isolated from environmental fluctuations, we observe the unfolding of a genetic program controlling proliferation and differentiation of cells into tissues. Considering that protein function depends on its structure (gene sequence), and on its interactions with other proteins and their reaction products which are in turn dependent on their structure, the pattern of gene expression essentially determines the functional state of the system (Figure 1). Consequently, the organism could be mapped into gene expression patterns. Development may hence be viewed as a computational process governed by the structure of the genome and defined initial conditions, i.e., the physical protein and molecular environment which sets the machinery of controlled gene expression into motion.

FIGURE 1



$$\left\{\begin{array}{l} DNA = const \\ mRNA = f(protein) \\ protein = f(\mathrm{m}RNA) \end{array}\right\} \Rightarrow \{mRNA = f(\mathrm{m}RNA)\}$$

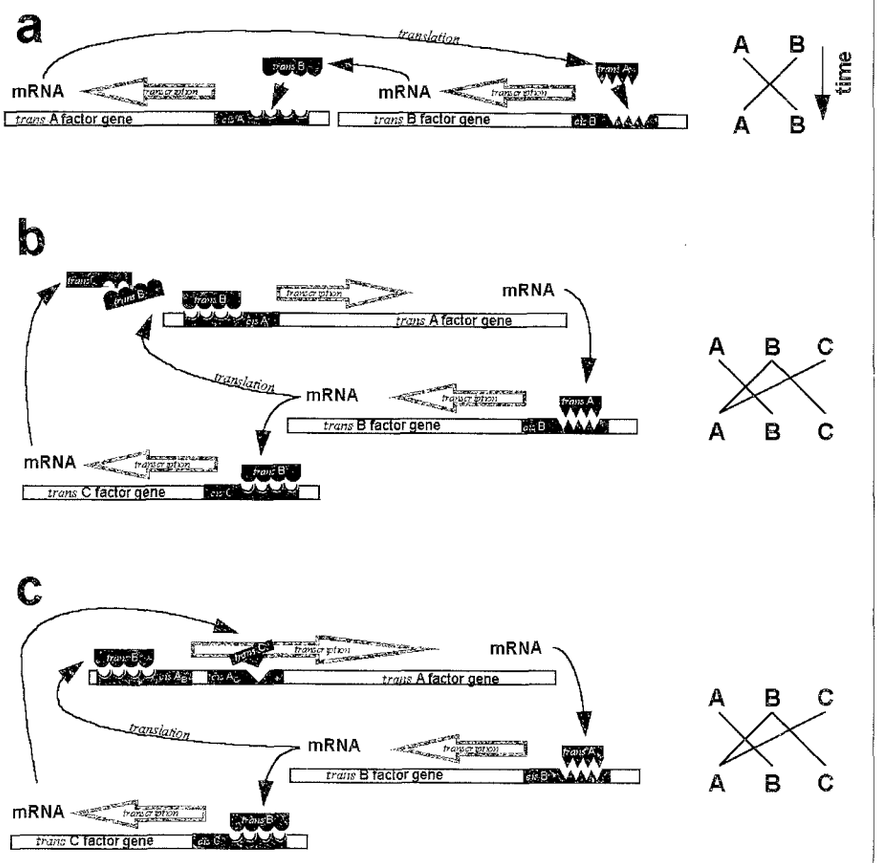
Levels of signaling in the genetic network. Genes regulate the expression of genes through their products, i.e., mRNAs and the resulting proteins. The regulatory functions involve signaling processes at different levels within and between cells. Since all of these processes are dependent on the expression of their genes, the mRNA gene expression pattern of a cell or system of cells defines the state of the signaling network.

Computation in genetic networks manifests itself in the interactions of *cis* acting regulatory DNA sequences of genes with *trans* acting elements, i.e., regulatory proteins (transcription factors, enhancers, suppressors, facilitators, etc.), the products of protein coding regions of genes. Hence, genes regulate the expression of genes, forming a genetic network. This is shown in the schematic Figure 2. Gene A codes for *trans* A factor protein, which regulates gene B by binding to the *cis* B element. Gene A is analogously regulated by gene B, forming a simple gene A-gene B feedback loop, the simplest case of a network. A wide of range of experimental evidence shows us that the regulation of gene expression is **combinatorial** (see below). This is schematized in Figure 2 as interactions between genes B and C in the regulation of A. *Trans* acting proteins may interact which each other before binding to *cis* regulatory elements [Figure 2(b)], or may bind to several *cis* elements directly leading to combinatorial structures [Figure 2(c)]. In addition, signaling interactions also take place beyond the *proximal network* of cis/trans interactions (Figure 1); we observe the levels of the *intracellular network*, involving processes such as protein phosphorylation cascades, and the *intercellular network* based on signaling factor and receptor interactions. Together, these levels of information processing form a feedback system of gene regulation (Figure 1).

Modeling Genetic Networks

To better understand the outcomes of complex interactions within genetic networks we need to find a symbolic language that reduces the network to its principal features. This step is facilitated by making the following idealizations of certain properties of genetic networks. a) The state of each gene or element can be reduced to either ***on*** (1) or ***off*** (0). Molecular interactions are often based on a sigmoidal relationship, approximating the *on/off* idealization with increasing Hill coefficient (discussed in [36]). b) As shown above (Figure 2), the combinatorial control of gene expression can be reduced to a **wiring**

FIGURE 2



Elemental interactions of gene regulation and their representation as wiring diagrams. Genes code for *trans*-acting proteins, which in turn control the expression of genes through interactions with *cis* regulatory sites located on the DNA molecule. The cybernetic foundations of such networks are represented by wiring diagrams, shown on the right, and the computational rules determining the input/output relationships (see text). **a)** Positive feedback system between genes A and B. Boolean input rules: $k = 1$ rule 2 (see Table 1) applies to both A and B. **b)** and **c)** Same as a), except that C inhibits and overrides the stimulatory action of B on A. This is accomplished by protein-protein (b) or protein-DNA (c) interactions, which are computationally equivalent (identical wiring diagrams and rules). Boolean input rules: B and C, $k = 1$ rule 2; A, $k = 2$ rule 4 (see Tables 1 and 2).

diagram causally linking the participating elements. c) The computation of these interactions can be idealized by combinatorial or **Boolean** rules, analogous to the observation that specific interactions between *cis* and *trans* elements control the expression of a gene. d) As a first approximation, all elements update their states **synchronously**. The implications of this simplifying assumption will be discussed below. Together, these principles serve as the foundation for **Boolean networks**.

DEFINITION OF A BOOLEAN NETWORK

The basis for Boolean networks was introduced by Turing and Von Neumann in the form of automata theory [67, 69, 70]. A Boolean network is a system of interconnected binary elements; any element in the network can be connected to a series of other elements. Each individual element uses a logical or Boolean rule to compute its value based on the values of the other elements it is connected to. The state of the system is defined by the pattern of states (on/off) of all of its elements. All the elements are updated synchronously, moving the system into its next state. According to the deterministic Boolean rules, each state can only have one resultant state.

The system, Σ, is defined by the number, n, of elements contained within it, each element's array, I, of k inputs and each element's Boolean function or rule, B. A particular state, S, of the system is defined as the set of values of the n elements. The total system space, Ω, is defined as all possible N combinations of the values of the n elements in S.

An element, E, may adopt the discrete value, v, of 0 or 1. A particular state of the system, S, is defined by the values of the n elements: $S = \{v_{E_1} \ldots v_{E_n}\}$. The total system space, $\Omega = \{S_1 \ldots S_N\}$ is composed of N number of states: $N = 2^n$. Each element receives an ordered array, I, of k input connections, where k may differ from element to element. Each of the input connections, I, is mapped to a selected element E of the n elements, as defined by the input array: $\mathbf{I} = \{I_1 \ldots I_k\}$.

The state defined by the values of the input elements represents a subset of total state space and may adopt 2^k permutations. The Boolean function, B, associated with each element maps each of these permutations to an output value of 0 or 1. The total number of possible Boolean functions is therefore 2^{2^k}.

The system is computed in discrete time steps. Therefore, the state of the system at any $t + 1$ is defined as follows: $S(t+1) = f\big(S(t), \{\mathbf{I}_1 \ldots \mathbf{I}_n\}, \{B_1 \ldots B_n\}\big)$.

Simplified Boolean Networks and Cellular Automata

General Boolean Networks

In general Boolean networks there are no constraints to the number of inputs, the wiring attributed to each element and its rules. Often they may be chosen at random [5, 75, 76].

Randomly Wired Boolean Networks

A simplification can be introduced by having each element in the network receive the same number of inputs, i.e., constant k. At this stage the network is still considered to be random since the inputs to a particular element may be chosen freely from among all elements [31, 36].

Cellular Automata

In cellular automata, the inputs are restricted to the immediate neighbors of a particular element. Much work has been devoted to cellular automata and the stable and metastable structures they exhibit in their space time patterns [40, 41, 73, 74, 75, 77]. However, the immediate neighbor wiring simplification of cellular automata cannot be considered to be representative for the wider variability of connections already known to exist in genetic networks.

Boolean Networks as a Language for Describing Genetic Networks

In the following sections we will discuss principal macroscopic features of Boolean networks using simple specific examples. Only larger, more complex and less penetrable networks will show all these features in combination, such as the random networks studied by Kauffman [36]. However, by identifying essential structural features of idealized Boolean networks we will gather an understanding of patterns that we may recognize in living genetic networks once data on complex, massively parallel processing genetic networks becomes available.

Genetic Networks

Input, output and computational rules for genetic networks are defined as follows:

1. **Output.** The protein encoded by a gene or its enzymatic product constitutes a *trans*-acting element or the output. In the furthest sense (extended genetic networks, Figure 1), this encompasses all regulatory interactions between proteins and their products that result in an alteration of gene expression.
2. **Input.** *Cis* acting elements (on the DNA) respond to the *trans* signal.
3. **Computation of output.** Several *trans* and *cis* acting elements are linked together resembling the input array. The input to a gene is then computed according to a combinatorial function, equivalent to the rule in Boolean networks. Whether this computation takes place on the *cis*, *trans*, or on both functional levels in a living genetic networks is irrelevant for the outcome as modeled by the rule [Figure 2(b) and (c)].

The simple example of the $n = 3$ genetic network of Figure 2(b) or (c) will help to illustrate basic genetic network behavior. Let us assume that A activates B, B activates A and C, and C inhibits A, overriding the activation of A by B. As an arbitrarily chosen initial condition element A is *on*, while B and C are *off*.

Sequential Versus Parallel Computation

By looking at this simple system **sequentially**, one may argue as follows. Since A is *on*, B must be turned *on* in the next step. B then turns *on* C. Next activated C must inhibit A. Inactive A will turn *off* B, B will then turn *off* C. In the final state all elements are *off* and will remain *off*. This apparently logical sequential argumentation is actually a fallacy which can easily be exposed by computing the network in a **parallel** fashion.

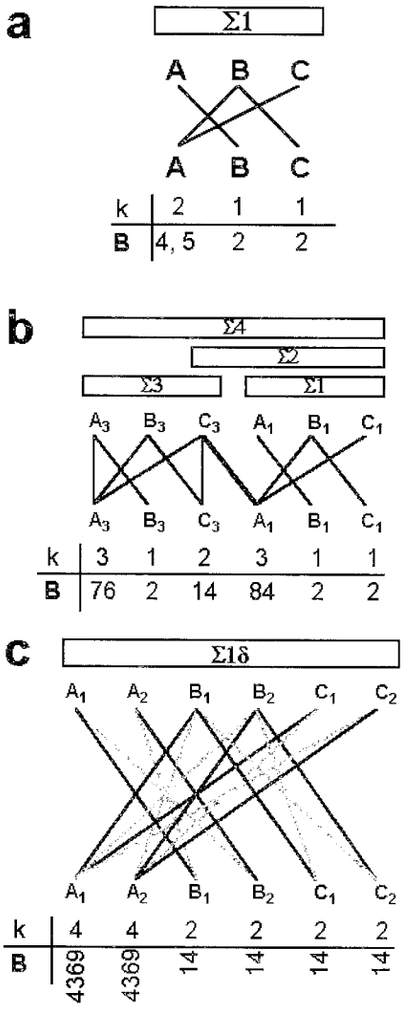
To enable **parallel** computation we will precisely formulate the $n = 3$ network according to Boolean network definitions below. The wiring diagram in Figure 3(a) [corresponding to the "biological" implementation shown in Figure 2(b) and (c)] describes the functional connections between participating elements. For A, B and C, the number of inputs, k, corresponds to 2, 1, and 1, respectively. The possibilities for rules are as follows.

1. $k = 1$ rules. There are $2^k = 2$ input permutations. A total of $2^{2^k} = 4$ combinations of input values with outputs define the $k = 1$ rule table (Table 1). Since each entry row of the rule table is expressed as a string of 1s and 0s, it corresponds to a binary number which is generally referred to as its decimal or hexadecimal equivalent. For $\Sigma 1$ [Figure 3(a)], B will be *on* only if A is *on*, and C will be *on* only if B is *on*, equivalent to $k = 1$ rule 2.
2. $k = 2$ rules. $2^{2^k} = 16$ combinations of outputs with $2^k = 4$ inputs define all the Boolean functions. In the $\Sigma 1$ example, A is turned *on* by B and turned *off* by C which overrides B, i.e., only if B is *on* and C is *off* will there be an activation of A. This corresponds to $k = 2$ rule 4 of Table 1.

$\Sigma 1$ is now completely defined [Figure 3(a)] allowing the **parallel** computation of the time course (space time patterns may be hand calculated for simple cases; more complex networks require the appropriate software such as the outstanding DDLAB by Andrew Wuensche [78]). The results shown in Figure 4a clearly differ from the fallacious, sequential attempt at predicting the outcome of this network. It illustrates a fundamental principle of Boolean networks: all network starting states will reach a cycle or attractor (see below). In this case, we observe a dynamic attractor of period 2 also displayed graphically in the right panel; the system alternates between two states that predict each other as their outcomes. Furthermore, the remaining states of this system lead to an alternative, single-state or point attractor [Figure 4(b)].

In a system like the above simple example it is not difficult to calculate the behavior of the network, although the superficial **sequential** attempt has failed to provide the correct outcome. Genetic networks as manifested in living organisms will be orders of magnitude more complex. Under-

FIGURE 3



Wiring diagrams and rules for network examples. The lines connect the upper row of output elements (time = t) to the lower row of input elements (time = $t + 1$). The no. of inputs (k) and the pertaining decimal rule (**B**) are shown underneath each wiring diagram. For an explanation of rules see text and Table 1. **a) Σ1**, identical to example shown in Figure 2. For element A, rule 4 is used in Σ1 and Σ1a, while rule 5 applies to Σ1b. **b) Σ4.** Σ1 and Σ3 are coupled through a unidirectional, nonfeedback interaction to one another in this composite network. Σ3 controls an "external" input , as depicted in Σ2, a transitional variant. c) **Σ1δ.** Redundant variant of Σ1 generated by complete duplication. Wiring and rules accommodate *or* crosslinks between the element and target pairs, as expected for functional duplication.

TABLE 1

decimal rule	hexadecimal rule	k	binary rule	binary input	decimal input
-	ff2a2a2a2a2a2a2a	6	4169697416	-	-
0	0	6	4	0	0
1	1	6	60074	1	1
2	1	6	4369	0	2
3	1	6	254	1	3
4	2	6	232	0	4
5	2	6	84	1	5
76	4c	6	76	0	6
84	54	6	84	1	7
232	e8	6	232	0	8
254	fe	6	254	1	9
4369	1111	4	4369	0	10
60074	eaaa	4	60074	1	11
4169697416	f8888888	5	4169697416	0	12
-	fefefe00fefe0000	6	-	1	13

6	5	4	3	2	1	k	decimal input
1	1	1	1	1	1	63	
1	1	1	1	1	0	62	
1	1	1	1	0	1	61	
1	1	1	1	0	0	60	
1	1	1	0	1	1	59	
1	1	1	0	1	0	58	
1	1	1	0	0	1	57	
1	1	1	0	0	0	56	
1	1	0	1	1	1	55	
1	1	0	1	1	0	54	
1	1	0	1	0	1	53	
1	1	0	1	0	0	52	
1	1	0	0	1	1	51	
1	1	0	0	1	0	50	
1	1	0	0	0	1	49	
1	1	0	0	0	0	48	
1	0	1	1	1	1	47	
1	0	1	1	1	0	46	
1	0	1	1	0	1	45	
1	0	1	1	0	0	44	
1	0	1	0	1	1	43	
1	0	1	0	1	0	42	
1	0	1	0	0	1	41	
1	0	1	0	0	0	40	
1	0	0	1	1	1	39	
1	0	0	1	1	0	38	
1	0	0	1	0	1	37	
1	0	0	1	0	0	36	
1	0	0	0	1	1	35	
1	0	0	0	1	0	34	
1	0	0	0	0	1	33	
1	0	0	0	0	0	32	
0	1	1	1	1	1	31	
0	1	1	1	1	0	30	
0	1	1	1	0	1	29	
0	1	1	1	0	0	28	
0	1	1	0	1	1	27	
0	1	1	0	1	0	26	
0	1	1	0	0	1	25	
0	1	1	0	0	0	24	
0	1	0	1	1	1	23	
0	1	0	1	1	0	22	
0	1	0	1	0	1	21	
0	1	0	1	0	0	20	
0	1	0	0	1	1	19	
0	1	0	0	1	0	18	
0	1	0	0	0	1	17	
0	1	0	0	0	0	16	
0	0	1	1	1	1	15	
0	0	1	1	1	0	14	
0	0	1	1	0	1	13	
0	0	1	1	0	0	12	
0	0	1	0	1	1	11	
0	0	1	0	1	0	10	
0	0	1	0	0	1	9	
0	0	1	0	0	0	8	
0	0	0	1	1	1	7	
0	0	0	1	1	0	6	
0	0	0	1	0	1	5	
0	0	0	1	0	0	4	
0	0	0	0	1	1	3	
0	0	0	0	1	0	2	
0	0	0	0	0	1	1	
0	0	0	0	0	0	0	

Selected rules. k corresponds to the input number. Inputs are counted from right to left in the wiring schemes (Figure 2). The rule defines the output for all possible input value combinations. All $k = 1$ rules and selected $k > 1$ rules corresponding to the discussed examples are listed. Large $k = 6$ rules are only shown as hexadecimals. Note the repetitive nature of the $k > 3$ rules, each of which includes several *and* or *or* groupings.

standing at higher levels of complexity will require clear formulations of models in a language such as Boolean networks that are dependably computable (Figure 4).

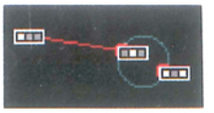
Interacting or Coupled Genetic Networks

One may also consider the alternative that there are "compartments" within a cell's genetic network which will feed critical inputs into each other that determine trajectories and attractors. This concept should be useful with respect to understanding "genetic programs" that determine specific cell types. A specific cell may share many expressed genes with other cell types. One may envision sets of genes that can be coexpressed in various combinations to define a cell type. Could genes be expressed in **modules**, required for certain functions such as basic metabolism (housekeeping genes), immune receptors, secretory apparatus, electrical excitability, contractile apparatus, etc.? **Attractors** (see below) in genetic networks would contribute stability to these units and may be controlled by a few regulatory inputs mediated by signaling factors. Consider e.g., the cell cycle attractor (discussed below), which is active in otherwise unrelated cell types, i.e., coexists with other cell-specific sets of genes in a variety of unrelated stem cells and other cell types that retain the capacity for cell division.

FIGURE 4

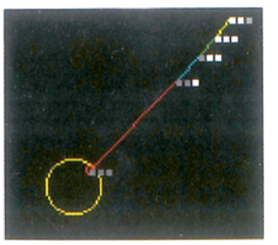
a

iteration	A	B	C
1	1	0	0
2	0	1	0
3	1	0	1
4	0	1	0



b

iteration	A	B	C
1	1	1	0
2	1	1	1
3	0	1	1
4	0	0	1
5	0	0	0
6	0	0	0



Time -space patterns and basins of attraction for network ∑1. Time space patterns are shown in the left panel. The basins of attraction of ∑1 (∑1a) include all eight possible states of the system (right panel). a) two-state dynamic attractor (repeating pattern), basin includes three states. b) point attractor, basin includes five states.

An example of hypothetical interacting network compartments illustrates coupled behavior. Element A_1 of ∑1 of the previous example may receive a regulatory input from element C_3, defining ∑2 (Figure 3). C_3 may be part of another network, ∑3. ∑3 can govern the trajectory of ∑1, together defining system ∑4. After two iterations from the initial state, the attractors of ∑3 (a two state attractor, bold) and ∑1 (a six state attractor, italics, resembles ∑1b) have been reached (Table 2). Together, they form a six state attractor in the total system ∑4 (underlined).

Salient Features of Boolean Networks

Attractors, Stability and Redundancy

One of the key features of Boolean networks is that all states, i.e., *on/off* pattern of its elements at a particular time point, lead to or are part of an **attractor**. An attractor is a distributed structure, based on a state (a point attractor) or series of states (a dynamic attractor) which repeats on itself; it can be considered a time-space map which jointly describes the oscillations of states of all pertinent variables. There is a simple rationale for explaining this property [31], as already implied by the above examples. A Boolean network occupies a limited number of states, *N*, contained in the total system space Ω. The system must eventually reach a state it has occupied be-

TABLE 2

Iteration	A_3	B_3	C_3	A_1	B_1	C_1
1	1	0	0	0	0	0
2	0	1	0	0	0	0
3	**1**	**0**	**1**	*0*	*0*	*0*
4	0	1	1	1	0	0
5	**1**	**0**	**1**	1	1	0
6	0	1	1	1	1	1
7	**1**	**0**	**1**	0	1	1
8	0	1	1	0	0	1
9	**1**	**0**	**1**	*0*	*0*	*0*

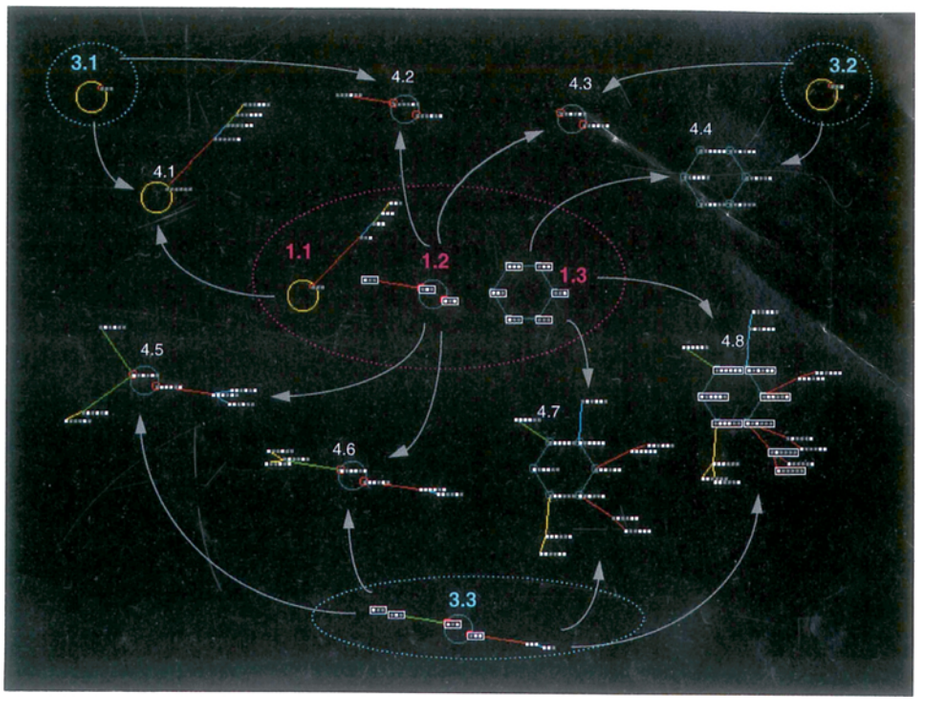
Trajectory of coupled network ∑4. The table follows one trajectory of ∑4 from the totality shown in Figure 5 (follow outlined states for this example). After two iterations the six-state attractor of the system has been reached (underlined). The main attractor is formed by coupling the attractors of subsystems ∑3 and ∑1 (see Figure 3 for breakdown). ∑3 falls into a two-state attractor (bold), while ∑1 reaches the six-state attractor (italics) of rule variant ∑1b (Figures 3 and 5). Essentially, systems ∑1 and ∑3, originally only capable of producing two-state attractors, have united to yield a six-state attractor.

fore since it cannot reach more than N states. The number of steps required to reach a repeating state can therefore be no more than N. Since each state, $S(t)$, unequivocally determines the following state $S(t + 1)$, once the system has reached a repeating state all following states must recur also in the same order, forming the **attractor**. All the states leading to or part of this attractor constitute the **basin of attraction**.

The basins of attraction for all possible states of $\Sigma 1$, $\Sigma 3$, and $\Sigma 4$ are shown in Figure 5. Three of all possible eight states of $\Sigma 1$ (same as $\Sigma 1a$) fall into the two element attractor (1.2) while the remaining five states fall into the $S = \{0, 0, 0\}$ point attractor (1.1). If randomly altered by a single bitflip, the system exhibits a propensity for falling into the "all *off*" point attractor since this basin of attraction includes 63% of Ω (Table 3). By altering the $k = 2$ rule 4 ($\Sigma 1a$) governing element A to rule 5 (A being constitutively *on* unless inhibitory C is also *on*; $\Sigma 1b$), we obtain a six element attractor (1.3), and retain the two element attractor (1.2), which lost one state in its basin of attraction (not shown; see table for basin of attraction characteristics). This demonstrates that not all attractors of a system are affected by a rule variation. $\Sigma 3$ has three basins of attraction, two point attractors each limited to itself (3.1 and 3.2), and a two state attractor which includes the remaining six elements in its basin (3.3).

The unidirectional, nonfeedback connection between $\Sigma 3$ and $\Sigma 1$ (Figure 3) leads to hybrid attractors in $\Sigma 4$ (Figure

FIGURE 5



Basins of attraction and subattractor composition of the coupled network $\Sigma 4$. All basins of attraction of $\Sigma 4$ covering the 64 possible states are shown. Each attractor of $\Sigma 4$ is a hybrid (indicated by arrows) of the attractors of $\Sigma 1$ (dotted outline in center) and $\Sigma 3$ (dotted peripheral outlines), demonstrating conservation of subnetwork characteristics. Attractor pairs 4.5 & 4.6 and 4.7 & 4.8 exhibit the same respective subattractor compositions. See Table 3 for attractor distributions. Note: attractors 1.1 and 1.2 comprise original $\Sigma 1$. Attractor 1.3 is derived form $\Sigma 1b$ (rule 5 for element A, Figure 3).

TABLE 3

				states in attractor		states in attractor basin		
System	n	N	attractor	total	% of N	total	% of N	subattractor composition
$\Sigma 1a$	3	8	1.1	1	13%	5	63%	
			1.2	2	25%	3	38%	
$\Sigma 1b$			1.3	6	75%	6	75%	
			1.4	2	25%	2	25%	
$\Sigma 3$	3	8	3.1	1	13%	1	13%	
			3.2	1	13%	1	13%	
			3.3	2	25%	6	75%	
$\Sigma 4$	6	64	4.1	1	2%	5	8%	1.1, 3.1
			4.2	2	3%	3	5%	1.2, 3.1
			4.3	2	3%	2	3%	1.2, 3.2
			4.4	6	9%	6	9%	1.3, 3.2
			4.5	2	3%	8	13%	1.2, 3.3
			4.6	2	3%	7	11%	1.2, 3.3
			4.7	6	9%	14	22%	1.3, 3.3
			4.8	6	9%	19	30%	1.3, 3.3

Attractor and basin of attraction composition of networks $\Sigma 1$-$\Sigma 4$. The number of states and percentage of total state space occupied by each of the attractors and its basin are shown for all systems (attractor numbering as in Figures 4-6). For coupled systems the subattractors, i.e., attractors of component systems, are listed. $\Sigma 1$ is divided into $\Sigma 1a$ and $\Sigma 1b$, characterized by $k = 2$ rules 4 and 5 for element A, respectively (see Figure 3). It is clear that attractors vary largely in the fraction of the total states that they capture in their basin. Furthermore, individual attractors in $\Sigma 4$ can share the same subattractor makeup (compare to Figure 5). In such cases the size of total state space occupied by these compound attractor basins increases accordingly, e.g., 4.7 and 4.8 (52%).

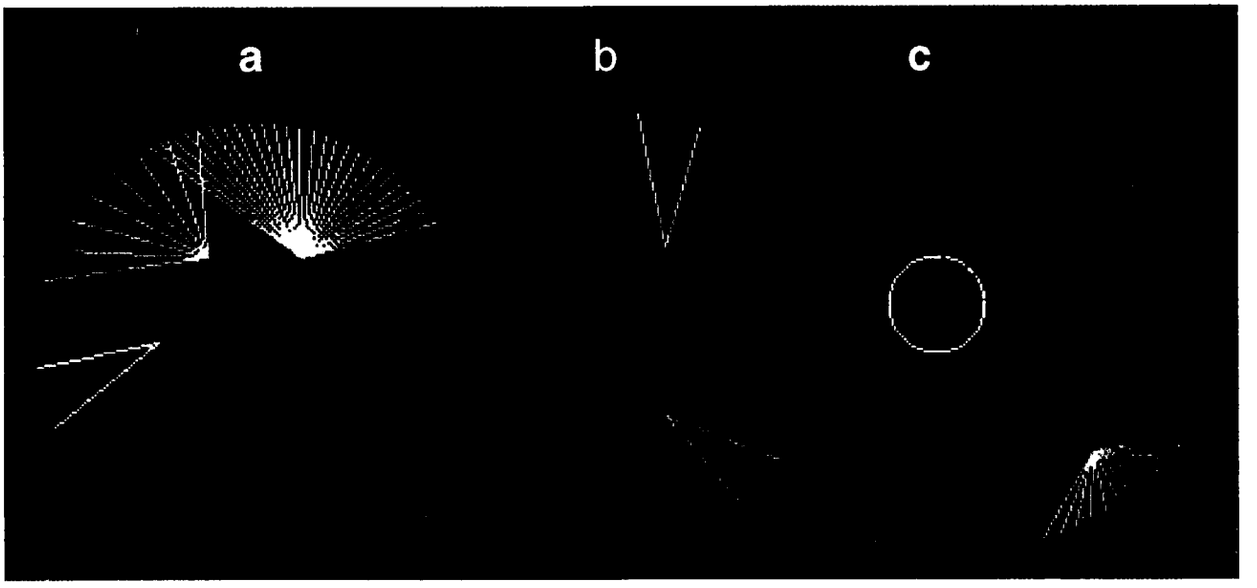
5; Table 3). All of the eight attractors of $\Sigma 4$ are combinations of the attractors 1.1-1.3 with 3.1-3.3, as also pointed out by the arrows in Figure 5. Furthermore, the attractor pairs 4.5 & 4.6, and 4.7 & 4.8 are each equivalent to a single hybrid (1.2 & 3.3, and 1.3 & 3.3, respectively), due to a synchronization phase shift (examine attractor bit-patterns in Figure 5). This model shows how small systems of elements can form a whole while retaining some of their individuality, e.g., modules of genes united in a system of coupled attractors.

We see how each network must reach one of several possible attractors depending on the initial conditions. But this has important implications, since attractors confer **stability** and resistance to perturbation, qualities fundamental to biological systems. Any state within a particular basin of attraction can be switched to any other state within this basin, without changing the global characteristics of the system, besides merely "resetting" the clock. Radical perturbations, such as randomly switching all elements of a state will cause the system to find an attractor with a probability that is solely determined by the proportion of the states lying in the basin of attraction to the total number of system states.

Despite the general resistance of attractors to point alterations in states or "bitflips," the boundaries of each basin of attraction must have at least one state in which a single bitflip will determine which basin the system will fall into. Consider that each state can reach any other state by a series of point alterations [36]. Therefore, no basin of attraction comprising fewer states than N can include all point alterations of each state. Hence, no *basin of attraction* of a system having more than one attractor can be completely resistant to point alterations in principle. However, *attractors* in selected systems may exhibit such resistance.

Besides stability these systems can demonstrate **redundancy**, i.e., removal of an element may not greatly affect the global behavior of the system. This is another important characteristic we expect from biological systems. Redundancy becomes particularly apparent for larger systems that are more abundantly wired than the examples discussed above [34]. The price of redundancy is that the size of the network must increase over the minimal number of elements required to "compute" a particular function, amounting to a reduction in short-term efficiency. Practically, this may be achieved by the parallel, independent evolution of elements overlapping in function and wiring [34], or simply by gene duplication. The latter mechanism occurs abundantly in the genome and is the primary event which leads to the evolution of gene families (discussed below). This is exemplified for $\Sigma 1$ by doubling each element and its wiring, and combining the original rules using the *or* operation to guarantee functional overlap [$\Sigma 1\delta$; Figure 3(c)]. The attractors for this $n = 6$ system are identical to $\Sigma 1$, except that the basin of attraction has increased due to a larger number of initial states (compare basin 1.3 in Figure 5 to basin a in Figure 6). By eliminat-

FIGURE 6

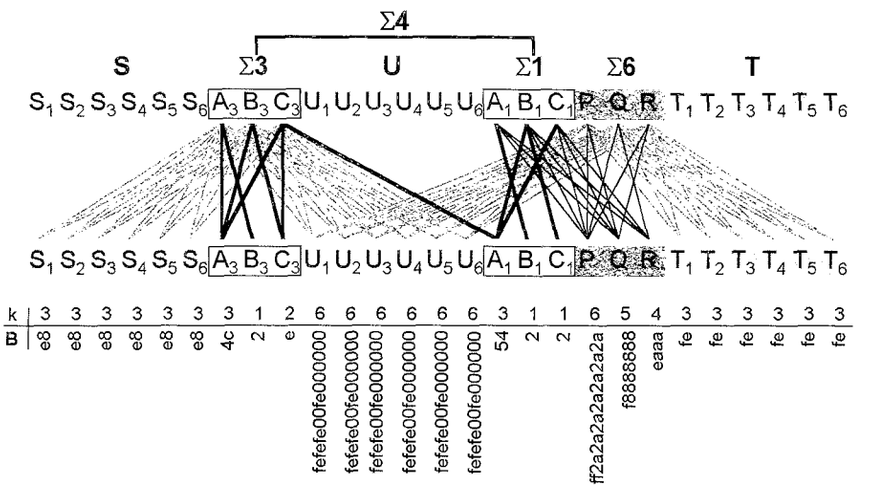


Attractor stability in the redundant network $\Sigma 1\delta$, a duplication variant of $\Sigma 1$. a) six-state basin of attraction of $\Sigma 1\delta$ ($n = 6$; see Figure 3). b) The attractor is unaffected by the deletion of elements A_2 and B_2 ($n = 4$). c) Breakdown of attractor by removal of B_1 and B_2 (one of the three possible detrimental, homogenous deletions).

ing any one element the global behavior of this system will not change. Eliminating two elements will not lead to a change in the attractor (Figure 6, basin b) except in the relatively unlikely event that a homologous pair is targeted (Figure 6, basin c). In sum, for 1, 2, 3, and 4 random deletions (which also decreases the system size!), the attractor will remain stable in 100%, 80%, 40%, and 0% of cases, respectively.

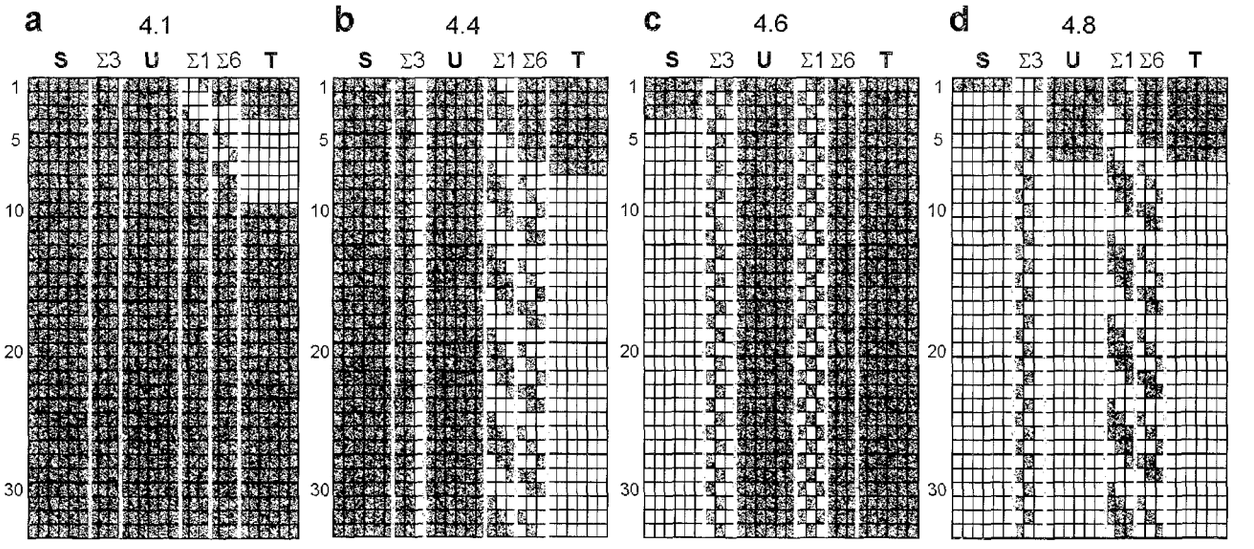
Networks may exhibit second order stability, manifested in the partitioning of **frozen** regions, akin to structural and housekeeping genes. These may be treated as dependent **modules** situated downstream of fluctuating feedback networks, either as the result of forcing structures (see below; [31]), or through radial wiring combined with homogenous rules (Σ7, Figure 7). Gene duplication could generate such structures. In example Σ7, the feedback networks Σ3, Σ1, and

FIGURE 7



Modular network of regulatory and dependent elements, Σ7. Radially, nonfeedback wired elements with redundant rules behave as dependents of regulatory elements, resembling structural or "housekeeping" genes. The no. of inputs (k) and the pertaining hexadecimal rules (**B**) are shown underneath each element. Σ4 (wiring shown in bold) forms the regulatory feedback module (identical to Figure 3 c). Elements P, Q, and R constitute Σ6 (wiring shown as thin, black lines) as a dependent feedback network, i.e., each element receives inputs from A_1-C_1 (connected by *and* operators) in addition to connections within this module. Dependent non-feedback modules **S**, **T**, and **U** (elements S1-6, T1-6, and U1-6, respectively; wiring shown as gray lines): **S** and **T** are radially wired to all elements of Σ3 and Σ6, respectively. Σ3 projects onto **S** with a rule equivalent to "at least two elements of three must be *on*". The rule "at least one element of three must be *on*" mediates activation of **T** by Σ6. **U** is analogously wired to both Σ3 and Σ6 using a rule which combines the rules governing **S** and **T** with an *and* function.

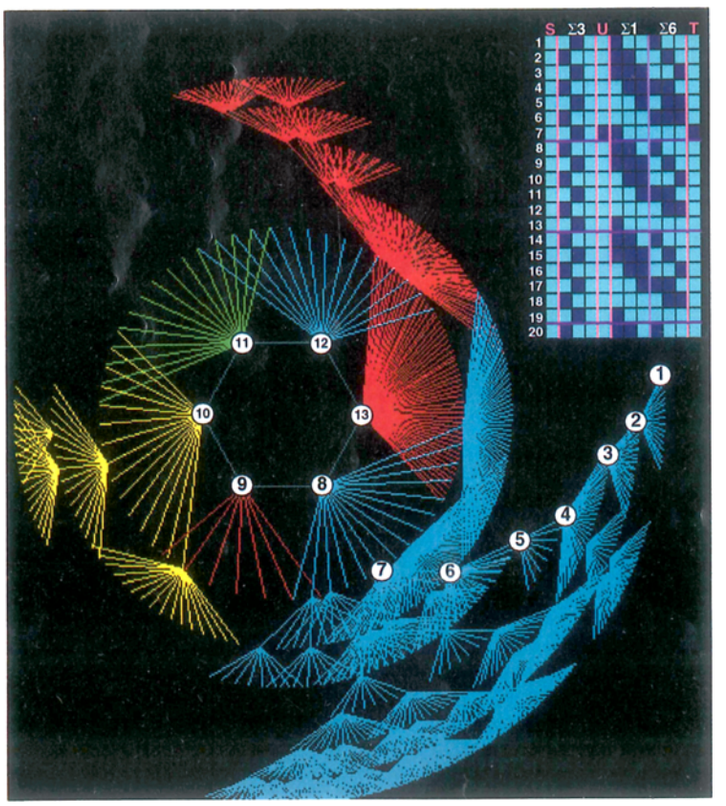
FIGURE 8



Selected trajectories of Σ7. Four different initial states were chosen, leading to attractors 4.1, 4.4, 4.6, and 4.8 of Σ4, respectively (compare to Figs. 5, 9). In each case, all elements of **S**, **T**, and **U** are initially *off*. The examples show that the states of **S**, **T**, and **U** strictly depend on the Σ4 attractors and their subattractor composition with regard to Σ1 and Σ3. **S**, **T**, and **U** result in the either all *off*, (4.1), all *on* (4.8), only **T** is *on* (4.4) or only **S** is *on* (4.6). **U** is only *on* if **S** and **T** are *on* by definition (see text).

Σ6 are hierarchically coupled to control the dependent modules **S**, **T**, and **U**. We (loosely) define *dependent* networks (Σ6, **S**, **T**, and **U**) as those highly controlled by inputs from another network, *feedback* networks as those with internal crosslinks embedded in their structures (Σ1, Σ3, and Σ6), and *nonfeedback* networks as those which solely receive inputs but do not project to other internal components (**S**, **T**, and **U**). Each element of the dependent networks Σ6, **S**, and **T** is wired to all three elements of the corresponding controlling network (see Figure 7 for rules). Control of dependent network **U** is essentially an *and* combination of **S** and **T**. The simple *or* and *and* rules are reflected in the repetitive binary and hexadecimal code for the complete $k = 4$, 5, and 6 rules (Table 1; Figure 7). Σ6, without receiving a recurring regenerative input from Σ1, will fall into the all *off* attractor, as exemplified by the trajectories in Figure 8a and c. Alternatively, the all *on* inputs of Σ1 will regeneratively activate Σ6, transducing the signal to dependent networks **T** and **U** [Figure 8(b) and (d)]. Essentially, the attractors of Σ3 and Σ1 determine the states of **S**, **T**, and **U**, either directly or as mediated by Σ6. This serves as an analogy for the generation of different cell types during development. Depending on the initial gene expression state of regulatory genes (Σ3 and Σ1), analogous to a progenitor cell, the system follows a transient pattern leading it to a cell-type specific attractor, characterized by combinations of structural genes (**S**, **T**, and **U**). Figure 9 shows a selected trajectory of Σ7 and its position in the total basin of attraction. All states in this basin will lead to permanent activation of **S**, **T**, and **U** [compare to Figure 8(d)]. Note that even for relatively large numbers of inputs ($k = 6$), a selection of simple rules from the vast rule space (Table 1) can lead to easily predictable, nonchaotic behavior. Whether such structures be generated by a) radial wiring with redundant rules due to gene duplication, b) random rules and wiring with the action of canalizing functions or homogeneity clusters [36], or c) a combination thereof, may be determined once data on parallel processing genetic networks becomes available.

FIGURE 9



Example basin of attraction for Σ7. For clarity, the dependent modules **S**, **T**, and **U** are limited to a single representative element. The inset shows the trajectory following a selected initial state (purple = *on*, green = *off*), corresponding to the marked nodes in the basin of attraction graph. This tree reaches the attractor after eight iterations. Note that **S**, **T**, and **U** are permanently *on* once the attractor is reached (compare to attractor 4.8, Figure 8).

Spontaneous Order and Evolvability in Random Boolean Networks

A number of studies by Kauffman and others have investigated Boolean networks' self-organizing behavior and drawn suggestive parallels to the behavior of biological systems (reviewed in [72]). Order in Boolean networks can be measured in terms of the smallness and stability of the state space, i.e., the attractors, that the system naturally boxes itself into out of the vast array of states available to it. Attractor cycles can be interpreted as the coordinated pattern of gene expression of a distinct cell type [32]. But, not all Boolean networks exhibit this kind of order. Kauffman's work investigates several characteristics of Boolean nets that correlate with the appearance of spontaneous order, and the mechanisms by which they work.

k = 2 Networks

The first of these characteristics is connectivity. This has been analyzed in a number of networks of various connectivities in which each element receives a constant number of k inputs. Specific wiring and functions are assigned randomly. The results show that for networks of high connectivity ($k > 5$), order does not emerge: the number of attractors increases proportionally with the size of the network (N), and attractor lengths increase exponentially with N. For networks of connectivity $k = 1$, the number and size of attractors are also unmanageably large. For $k = 2$ networks, however, the average attractor number and size are surprisingly circumscribed: both vary with the square root of the network size [35]. Interestingly, in a wide variety of organisms the number

of distinct cell types similarly varies with genome size [31].

$k = 2$ networks also show stronger resistance to perturbation than either more highly-connected or $k = 1$ nets [33, 34]. Such behavior parallels biological systems' undeniable stability in the face of random environmental noise. Additionally, when a perturbation does push a $k = 2$ system into a new basin of attraction, the number of other basins which it can fall into is fairly small. This phenomenon is suggestive of branching patterns observed in development.

Canalizing Rules and Forcing Structures

Although $k = 2$ Boolean networks may show strong self-organizing capacity, actual biological networks demand models with higher connectivity (see below). There are two basic mechanisms, both operating in similar fashion, which suffice to extend orderly behavior into networks of higher k through forcing structures and homogeneity clusters [16, 31, 71]. Forcing structures result from canalizing rules, defined as Boolean functions in which at least one particular value of at least one input location suffices to completely determine the function's output, regardless of what values any of the other inputs may receive. The Boolean rule *or* serves as an example: this function is canalizing to an output of 1 for an input value of 1 at any input location. (One should note that of the 16 possible $k = 2$ Boolean rules, 12 of them are canalizing, a greater proportion than at any other level of connectivity.)

A network with a high proportion of canalizing rules is very likely to develop forcing structures. A forcing structure has a strong tendency to fall into a state in which each of its elements permanently takes on its canalized value. Large blocks of elements thus appear which are permanently frozen into the *on* or *off* state [33]. Fixed into the *on* state, these may correspond to permanently active "housekeeping" genes in the cell. The frozen blocks effectively partition the system into smaller, functionally isolated areas consisting of active, fluctuating elements. This model suggests that cell types may differ through distinct patterns of activation of whole gene packets. Such an architecture has been shown to strongly correlate with the orderly dynamics that $k = 2$ nets exhibit: namely, small attractor sizes and resistance to perturbation.

Homogeneity Clusters, $P > P_c$

Homogeneity clusters [31] provide a second mechanism which suffices to create orderly dynamics in higher connectivity networks. As we have seen, an element with k inputs has an associated Boolean function with 2^k positions. A certain fraction of these positions in any given function takes on a 1 response. The internal homogeneity of a network, P, measures the amount that this fraction deviates from 0.5 (P assumes that each function in the network has the same level of homogeneity). Various studies on lattice networks with nearest-neighbor connectivity (cellular automata) show that if the internal homogeneity P is greater than some critical value, P_c, frozen blocks of elements similar to those created by forcing structures appear. Order emerges by the same basic mechanism as above.

Evolvability

Evolution through natural selection presupposes two basic phenomena. Clearly, there must be competing systems of various fitness levels from which to select, and the systems must be such that the mechanism of selection can increase their fitness. The theoretical results from Boolean network studies suggests a powerful potential source of initial order [31]. Given a system consisting of combinatorially acting input-output elements, such as the one implemented in genetic networks, order simply emerges for free. It need not be the infinitesimally rare product of eons of chance events. Order can occur quite easily, and, in fact, quite inevitably.

The extent to which selection modifies the spontaneous order of a Boolean network depends on how strong selectional forces are, determining to which degree biological genetic networks will resemble the random networks described above. The means by which genetic networks evolve, e.g., through gene duplication and point mutation, should also have a strong effect on the appearance of the final products as we see them today.

Evidence for Genetic Networks and the Boolean Model

Regulatory Logic of *cis* and *trans* elements

To a great extent, current research in experimental biology has been devoted to understanding the mechanisms of gene regulation, underlying the critical phenomena of cell growth and differentiation. A model of *trans* acting proteins which interact with *cis* regulatory DNA elements is now established to explain the most proximal events in gene expression control (see Figure 2). Protein *trans* elements are referred to as transcription factors acting as activators (enhancers, facilitators) or inhibitors (suppressors, silencers). *Cis* elements of DNA begin with short *trans*-factor binding sequence motifs on the promoter and have been extended to include enhancer, suppresser or silencer regions, often far removed from the promoter on the DNA molecule. Transcription factors have evolved into large families based on conserved structural protein motifs able to interact with specific segments of DNA or other regulatory proteins (reviewed in [26]). While several inputs may be required to regulate one gene (*multigenic* regulation), a particular gene may also be able to affect the expression of a wide variety of downstream genes (*pleiotropic* regulation; for a general overview of *cis/trans* mechanisms see [1]). Some examples of higher vertebrate genes allow estimates of minimum number of functional inputs, equivalent to k in network terminology, and outputs: minimum inputs - keratin, 5 [48], c-fos, 4 [54]; minimum outputs - NRSF, 10 [56].

The combinatorial nature of gene regulation as modeled

by Boolean networks has been confirmed in experiments characterizing multi-input control of gene expression in bacteria [32, 36] and, in a pertinent example, fruit fly development [1]. Further examples of this phenomenon are now being seen in higher vertebrates, such as the multigenic control of the immediate early gene, c-fos. C-fos, itself a transcription factor connecting to a wide variety of downstream genes [3], has been mapped for some of its inputs and rules in a transgenic mouse model [54]. Constructs were prepared in which four different promoter elements, SIE, SRE, FAP, and CRE have been expressed individually and in combination. The results demonstrated, contrary to previous assumptions, that these elements do not act independently. Mutation of any single element resulted in a loss of tissue specific and stimulus-evoked gene expression. IE, SRE, FAP, and CRE therefore appear to be linked by the *and* operator, leading the authors to propose the ITC (interdependent transcription complex) hypothesis. Whereas the *or* function could be computed by elements acting in isolation, a molecular complex is required to calculate the *and* function. In general, ITCs could be responsible for the physical computation of rules beyond *or* for Boolean networks.

An example for an inhibitory gene which outputs to a group or module of genes (pleiotropic control; compare to radial wiring of $\Sigma 7$, Figure 6) is found in the regulation of neural development. The (*trans*) neuron-restrictive silencer factor (NRSF) suppresses transcription of neuronal genes (neuron-specific ion channels and transporters, synaptic proteins etc.) by interacting with the (*cis*) neuron-restrictive silencer element (NRSE; [56]). NRSF is expressed in many nonneuronal tissues, glia and neuronal progenitors. The latter lose NRSF while maturing to neurons, concomitant to the induction of neuron-specific genes. The authors have proposed the role of NRSF as a "master gene," we prefer the term *focal gene*, overriding stimulatory inputs to neuron-specific genes and thereby inhibiting their expression. Inhibition of such an inhibitor, constituting activation, should then facilitate neuronal differentiation. The role of such an activator has been attributed to Hel-N1, a necessary gene for the development of the nervous system. Hel-N1 protein binds to the 3' untranslated region of the *ld* mRNA, which encodes a transcriptional repressor that is expressed in undifferentiated neural precursors [38]. It is interesting to note that the regulatory mechanism involved here is not based on a DNA-protein, or protein-protein (as shown in Figure 2), but on a protein-RNA interaction resulting in either inhibition of translation or degradation of the mRNA. From the structural network perspective, NRSF and *ld* provide examples of how a canalizing function "if *on*, then downstream element will be *off*, independent of other input states," radially wired, can control a larger module of genes. Furthermore, there are copious examples for canalyzing rules in addition to those discussed here, suggesting that this ordering principle is abundantly utilized in higher vertebrate gene regulatory architectures.[3]

Other conceivable regulatory mechanisms would satisfy feedback and control requirements for building a genetic network. When considering the basic principle that a sequence of nucleotides in DNA determine the structure of the macromolecules that in turn regulate DNA, one is led to wonder why the RNA macromolecule is excluded from directly interacting with DNA and modulating gene activity by this straightforward means. Aside from its role as an information carrying molecule, RNA can fold into complex structures such as rRNAs which make up ribosomes and tRNAs. Could mRNA or mRNA-protein complexes, analogous to ribosomes, also reasonably play a role in transcriptional regulation? RNAs should have an advantage over proteins for this function since they are also capable of base-pairing with DNA. This could provide immediate specificity for regulatory interactions. Limited experimental evidence for RNAs playing a direct regulatory role is only available on the level of translational control and in determining mRNA stability [38, 42]. Will the future show examples for more intricate RNA-gene expression interactions?

The Extended Genetic Network: Intra- and Intercellular Signaling Networks Funneling Into the Genetic Network

Genetic networks extend beyond a single cell in the control of proliferation and differentiation during development and in the coordination of cellular activity in the organism's interaction with the environment. These functions require regulatory mechanisms beyond direct *cis-trans* interactions. Proteins produce intercellular messenger molecules, which are in turn received by other cells and transduced into intracellular biochemical responses (Figure 1). Typically, a signal receptor protein either directly or via intracellular messenger molecules (Ca^{2+}, cAMP) changes the phosphorylation states of regulatory target proteins. A cascade of phosphorylation mediated protein activation and inactivation in turn causes genes to switch *on* and *off*. Signaling interactions mediated by messenger molecules and protein phosphorylation show interesting dynamic behavior, such as Ca^{2+} oscillations, which have been studied extensively as a paradigm for the application of theoretical systems analysis in biological signaling ([59]; reviewed in [61]). However, the dynamic behavior of e.g., Ca^{2+} oscillations is embedded in a general network based on crosstalk between a variety of biochemical signal transduction components [59]. This line of research is presented with the acute problem that models explaining the dynamics of these parameters will be difficult to test. The necessary experimental techniques for the massively parallel measurements of biochemical parameters are neither available nor within reach in the foreseeable future.

However, it is an arbitrary choice whether to look at biological systems as their projection in intercellular signal space,

protein activity space or gene expression space. Once the variables have been defined in a particular parameter space, other elements can be folded into the functions governing these parameters. From the view of genetic networks, all events on the level of production of signaling molecules, protein phosphorylation etc. will end up in determining which genes are activated or inactivated [27]. Based on cooperativity and threshold behavior, the binary simplification may also be extended to these biochemical interactions [4]. Therefore, the computation of extended networks could be achieved by including all higher level molecular functions in the formulation of wiring diagrams and Boolean rules. For instance, a gene encoding a signal molecule synthesizing enzyme may exert a positive feedback on its own production, provided the genes for its receptor and the necessary intracellular signal transduction molecules are active. This would correspond to *and* connections between the involved elements. Such a feedback mechanism has been considered for the regulation of GAD (glutamic acid decarboxylase; catalyzes the synthesis of the neurotransmitter GABA) in the developing rat spinal cord [62]. GAD acts through the diffusible intercellular signal GABA (γ amino butyric acid), and signaling mechanisms involving GABA receptor operated Cl^- channels, possibly Ca^{2+} channels, Ca^{2+} dependent protein kinases and phosphorylation activated transcriptional regulators. This signaling chain could lead to the activation of GAD mRNA expression. From a Boolean network standpoint it would suffice to simplify this scheme into a rule that incorporates the exact combination of expressed genes; the expression of a gene is reduced to a function of the expression of genes (Figure 1). Relative to the genetic network perspective, proteins will inadvertently execute their function, no matter how intricate, and are only relevant from the standpoint of the computation of the state as defined by the gene expression pattern.

Redundancy and Gene Knockouts

Discoveries in the molecular genetic bases of cell function has spurred the desire to reduce each function of a cell to an underlying gene and attribute a functional role to each newly discovered gene. It is compellingly argued that if a gene has been carried and conserved through millions of years of evolution it should fulfill an irreplaceable role in the organism. To this effect the approach of single-gene knockout animals, presently restricted to mice, has been introduced. This technique makes it possible to routinely introduce disrupted gene constructs into the germline to create homozygous offspring deficient for the expression of a particular gene (reviewed in [18]). Indeed, several of these mutations prevent full development of the animal (reviewed in [9]), as expected for essential genes.

However, in many instances it has been demonstrated that elimination of certain genes leads to a fully functional animal with no easily identifiable change in phenotype. Functional deletion of *c-fos*, discussed above as a transcription factor with far reaching implications for differentiation and proliferation, has not produced significant changes in the behavior of embryonic stem cells [15]. These cells were able to differentiate into a wide range of cell types in tissue culture and also in chimaeric mice. In Boolean network language *c-fos*, an immediate early gene which is transiently expressed during signaling [3], resembles a redundant regulatory element located in a tree leading to an attractor.

Structural genes can be viewed as generally redundant from a narrowly construed network perspective, since they receive inputs but provide no immediate regulatory outputs (compare to **S**, **T**, and **U**, Figure 7). However, they may provide an output in the context of the extended network by permitting aspects of cell function necessary for survival or further development. In the case of glial fibrillary acidic protein, a highly conserved intermediate cytoskeletal filament of astrocytes (a glial cell of the central nervous system), no dysfunctions in animals lacking this gene, whether in the nervous systems or otherwise, could be observed [22]. Apparently the role of this protein in organizing the glial cytoskeleton must be covered by other filaments, suggesting complete redundancy of cell and genetic network function.

Another means by which redundancy has been achieved is demonstrated in the study of the *myf-5* and *myoD* transcription factors in muscle development [55]. While deletion of either of these genes leads to no apparent changes in phenotype, the dual knockout results in a complete loss of skeletal muscle and skeletal muscle specific genes. It appears that *myf-5* and *myoD* are effectively linked by an *or* function as they form inputs to other genes that control skeletal muscle development.

Interestingly, certain genes, although active in a wide variety of cell types, appear to be necessary for only a few specialized functions limited to small systems of cells. A much studied signaling protein, Ca^{2+}/calmodulin-dependent protein kinase II (CamKII), which has been implicated in mediating many of the general effects of the ubiquitous intracellular second messenger, Ca^{2+} [25], including transcriptional regulation [24], was targeted in a gene knockout experiment. Unexpectedly, despite this protein's centrality to signaling, the mutant animals developed and survived normally and showed no direct physical abnormalities [8]. However, learning ability was slightly curtailed, and they behaved more aggressively than controls. Apparently, while the function of CamKII is redundant in most cells in which it is expressed, certain cell systems depend on it for fine-tuning of function. This example for limited redundancy can be incorporated into the genetic network model, in that an element may be redundant in trees leading to some attractors, but not to others. For example, in $\Sigma 7$ (Figures 7 and 8), element A_3 is not necessary for controlling the stimulation of $\Sigma 1$ by $\Sigma 3$, but is required for activation of **S** and **U**. In summary, a true understanding of function on a molecular level must incorporate multigenic causes and

pleiotropic effects [10, 57]. Only through a network perspective can this deeper insight be found.

Gene Duplication and the Generation of Redundancy: "Variations on a Theme" and Frozen Regions

Stability and redundancy are a characteristic feature of genetic networks. The experimental evidence suggests varying degrees of redundancy for functions of specific genes as deduced from knockouts (discussed above). Redundancy at its most basic level is already observed in random Boolean networks within certain bounds of connectivity and rules [34]. Such a network will by chance have overlapping functions as defined by the *or* operator connecting elements with shared inputs. We have shown an example in which redundancy is simply generated by identical duplications of elements [Figures 2(d) and 6]. Although the experimental evidence suggests that purely random general redundancy in living genetic networks may be limited (discussed above; gene knockouts), ample evidence exists for gene duplication in various contexts [42]. Fundamentally, example $\Sigma 1\delta$ [Figures 2(d) and 6] corresponds to a diploid organism carrying two copies of the genome. Usually diploidy is discussed in terms of sexual reproduction and generating variability during meiotic chromosome crossover, but this is relevant only to the germ line. For somatic cells, diploidy plainly offers a spare gene copy in case of a single failure due to mutation.

Further redundancy in the haploid genome is offered by gene duplication. Abundant evidence exists for the presence of identical, active copies of genes, altered gene copies which form the foundation for gene families, and nonfunctional pseudogenes (for an overview see [42]). Duplications may also result in partial copies of genes, providing opportunity for independent spreading of regulatory and protein coding domains. The significance of partial gene duplication is underlined by the high expression of retrotransposons (10% of the total genome, [1]), which are responsible for copying and transferring DNA segments. Not surprisingly, retrotransposon-mediated genetic alterations are also associated with tumors and genetic diseases [2].

A particularly interesting example for an evolving duplication has been discovered in the dual, nonallelic rat insulin I and II genes, both generating an identical protein form in the rat. These are regulated in parallel in the pancreas, reflecting true functional redundancy [20]. However, insulin expression in the brain is exclusively derived from the insulin II gene [11], suggesting that the inputs to both genes overlap only partially. Could this be the beginning of the evolution of a brain-specific insulin gene?

Regulation of the keratin gene family demonstrates how variations in the regulatory and protein coding regions of the gene have led to epithelial cell type-specific expression patterns through combinatorial computations of inputs. While common *cis* elements assure that keratins are only expressed in epithelia, variable *cis* elements direct cell-specific keratins associated with the diversity of epithelial phenotypes [48]. This regulatory constellation may be viewed as analogous to a diversified variant of the **S**, **T**, and **U** structural element clusters of $\Sigma 7$ (Figure 7). The evolution of gene families analogous to keratins has been generally implied in tissue evolution [45], providing further evidence for genetic network growth by duplication and variation as an underlying mechanism in the evolution of higher organisms.

In summary, theoretical studies have shown that order and complexity in genetic networks confined by small numbers of inputs, canalizing and $P > P_c$ rules, may resemble random genetic networks, especially in phylogenetically early organisms (see above; [31, 35]). However, more highly evolved networks reflect overlapping wiring and rule patterns based on element kinships. The high number of inputs and expansion of rule space, all potentially driving the system to disorder, may effectively be reigned in by the principle of "variations on a theme" (see above; discussed in [37]). Future studies in Boolean networks may be inspired by this interpretation.

It should be reemphasized that "redundancy" in the above context is to be understood not as "nonbeneficial," "useless" or "superfluous," but more akin to functional overlap and backup (see [10, 57]). If redundancy confers stability, it may increase the fitness of an organism and may therefore represent a selective advantage. There is nothing excessive about having a spare copy of an important functional component. Although the molecular and evolutionary mechanisms that are responsible for redundancy, gene duplication and emergence of gene families are still being debated [29, 37], the evidence for the wide-spread existence of these phenomena is compelling.

Oscillators and Attractors

We have discussed an important property of Boolean networks above: each state inexorably leads to an attractor, conferring stability to the system. In terms of genetic networks, dynamic or point attractors correspond to fully differentiated cells in which the expression of the cell type specific genes has reached its final stable state(s) (Figure 8; [36]). The detail of our knowledge on dynamic gene expression patterns in cells does not yet allow us to distinguish between attractor types. One may conjecture that dynamic attractors should be abundant in a wide variety of cell types, because point attractors occupy a much smaller region of attractor state space and thus are simply that much less likely to occur. Since there are no apparent functional benefits of point attractors over dynamic attractors, there is no reason to postulate special selection pressure for the elimination of dynamic attractors. However, dynamic attractors have been studied in cases in which they represent an integral functional property, the cell division cycle and biological rhythms.

The cell cycle attractor encompasses elements of genetic

and protein activity networks. In higher vertebrates the analogy to genetic networks is particularly useful since the regulation of gene expression plays important roles in entering and maintaining the cycle. Cells must pass the G1/S checkpoint in order to irreversibly enter the cell division program (for a general overview see [1, 42, 50]). This step requires that the network enters a state which is part of the basin of the cell cycle attractor. It is strictly regulated by positive (e.g., growth factors; [52]) and negative (e.g., retinoblastoma protein, a tumor suppresser) modulators to guard against uncontrolled growth leading to tumors. A group of genes which are induced by growth factor inputs are the D cyclins [58], strongly implicated as regulators of G1/S progression and as oncogenes (cancer-causing genes) when mutated [47]. They are part of the delayed early response to growth factors following the induction of early response genes like *jun* and *fos* [1, 47], analogous to an attractor tree. Most importantly, mRNAs coding for cyclins D1, D2 and D3 oscillate during the cell cycle [7]. D cyclins and their role in the cell cycle correspond to the general predicted behavior of elements in a dynamic genetic network attractor.

Recent work on the genetic network components underlying circadian rhythms suggests that dynamic attractor concepts will need to be invoked to gain insight into these phenomena. Although multigenic control is implied by the paucity of single gene effects in screens of circadian rhythm mutant populations, the *per* gene in Drosophila (fruit fly) and *frq* in Neurospora (a mold) satisfy the requirements for an element which lies in the attractor and is necessary for its progression (reviewed in [13]): a) the *per* and *frq* transcripts oscillate with the daily cycle, b) deletion of these genes results in mutants that loose their circadian rhythmicity, c) constitutive overexpression eliminates cycling, suggesting that the regulatory mechanism is on the level of gene expression rather than protein activity by posttranslational modification [49]. Furthermore, mutations in *per* usually either shorten or lengthen the cycle period, suggesting that altered protein forms only affect how quickly *per* completes its part in computing the next state in the attractor. Although little is understood of the number of elements, wiring and rules that participate in the circadian attractor, it has been suggested that *per* and *frq* exert a negative feedback on their transcription, either directly or through follow-up steps in the attractor [13, 23, 49].

The circadian cycle has been suggested to be the result of coupling of shorter attractors by *per* [12]. Such ultradian attractors exhibiting 5–15 h periods remain after deletion of *per* and are also present in normal flies. One may invoke principles similar to the coupling of $\Sigma 1$ to $\Sigma 3$ in $\Sigma 4$, resulting in the transformation of a pair of two state attractors into a six state attractor (Table 2; Figure 5). Could *per*, analogous to element C_3 of $\Sigma 4$ (Figure 2), play a role in coupling attractors to increase their state cycle lengths? Moreover, *per* has the ability of resetting the clock, analogous to the control of $\Sigma 1$ over $\Sigma 6$ (Figures 7–9), except that *per* acts alone and not through a combinatorial function. This is experimentally manifested in *per*'s control over the phase of the circadian oscillator [14].

Probably more cell and tissue autonomous clocks exist than are now known. These may share some genetic elements, since alterations in the circadian clock in Drosophila are known to affect the periodicity of short ultradian rhythms [13], such as the male courtship song (~1 min) and timing of luteinizing hormone pulses (~30 min). Perhaps the biological rhythms examined so far, rather than representing unusual and highly specialized functions, reflect the inherent property of deterministic genetic networks to form attractors. As more molecular components of the circadian attractor become known, combinatorial genetic network models may provide an opportunity to clearly explain how these interact to form a highly regulated and stable timing mechanism.

Binary States and Synchronization in Genetic Networks

Two main idealizations in the design of Boolean networks are the assumption of binary states, and the synchronous updating of element states in unitary time steps across the network. These simplifications allow the model to provide a transparent demonstration of the salient features of complex systems, without becoming lost in the mechanistic details and quirks of individual interactions. We have turned to Boolean networks to find answers to how a living system can coordinate the action of large numbers of genes in parallel, unerringly reaching the same result in ontogeny and maintaining order in the face of an ever fluctuating environment. Can living systems accomplish this because they have evolved features akin to the simplifications that allow Boolean networks to exhibit complex rather that chaotic behavior? Do Boolean networks, by crystallizing state and time values into discrete elements, capture essential features of living systems without oversimplifying them beyond recognition?

For one, cooperative molecular interactions with increasing Hill coefficients or in combination with positive feedbacks provide a close approximation to binary behavior (discussed in [4, 21, 28, 36]). While one may observe gradations in temporally changing parameter values with sensitive experimental techniques, the resulting threshold behavior will create a virtual *on/off* switching network. However, there is no experimental evidence, pro or contra, for tightly synchronous processing. While the discrete progression of time may be acceptable depending on the resolution provided by the time "binning" interval, absolute synchronicity is a different issue. First of all, it is not clear how large kinetic variations are in gene regulatory processes [39]. Also, since one causal link between elements may involve several steps (e.g., transcription, translation, protein phosphorylation, protein complex formation, activation of downstream genes), while another parallel interaction may operate more quickly, it would not be accu-

rate to calculate both steps in the same iteration. From a biological standpoint, there should be no advantage for gene regulation processes to march in lock-step. As long as the duration of these processes is largely reproducible, timing could be reasonably coordinated. However, considering the introduction of various levels of noise, one may seriously question how the apparent temporal coordination of genetic network processes such as ontogeny, biological clocks etc., is achieved.

A possible answer may lie in a judicious choice of network wiring and rules with respect to phase sensitivity. The attractors of $\sum 1b$ (Figure 5, attractor 1.2 and 1.3, rule 5 for element A) demonstrate the role of synchronization. If the elements of the 2 state attractor, 1.2 (Figure 5), are not processed exactly in parallel—say, if in the {1, 0, 1} state, element B were to be turned *on* (due to A) before either A or C were turned *off*, resulting in states {1, 1, 1}, {0, 1, 1} or {1, 1, 0}—attractor 1.2 will be abandoned for attractor 1.3. However, in case of the six state attractor, 1.3, the elements do not need to be updated simultaneously; they can be calculated at leisure in any order. Slight variations in timing, easily imaginable in biological systems due to noise, would not affect the six state attractor, but would drop the states of the two state attractor into the six state attractor basin. This process would be irreversible, clearly demonstrating how **noise** in Boolean networks could actually be used to **stabilize** the system or drive it into a favored conformation [31]. Could self-synchronizing rules and wiring underlie observations such as checkpoints in the cell cycle, and the stability of biological rhythms and genetic network structures in general?

On longer time scales, phase insensitivity may be granted in specialized cases, but would become absurd if taken too far. Coordination in ontogeny is required in the sense that an activator must appear while the receptor window is open. This may not always involve feedbacks and therefore requires accurate internal timing or inputs from a central timekeeper. Again, in Drosophila the *per* gene has also been implied in controlling developmental timing, but is not clear whether its function is more centralized or local in nature [13]. On the other hand, as discussed above, *per* can reset the clock, thereby overriding other clock components, and making it phase independent. One should also differentiate between phase independent inputs (coupling of $\sum 3$ to $\sum 6$, discussed above) and self-synchronizing rules and wiring, although both may be required for coordination and stability to noise. These features together offer a solution to the synchronous updating problem. If the system cannot progress to the next state without the critical update, if noise sensitive attractors fall into noise-insensitive ones, if outside inputs override internal states and become phase insensitive, the system is self-synchronizing and should essentially be modelable by Boolean networks that exhibit these features in wiring and rules. An immediate undertaking in Boolean network research would be to define such rules and wiring and test to which degree they improve the robustness of the network. Complementarily, these reflections on synchronization in Boolean network behavior have suggested interesting and potentially useful structural features that may be sought out by experimental genetic network researchers.

These conclusions will still hold up if the definition of a Boolean network were to be extended to include deterministic timing parameters in order to enable more realistic simulations of living genetic networks [53, 66, 68]. It would not be difficult to introduce a phase shift parameter in genetic network models; e.g., if $A = on$ and $B = on$ at t = 0, and $C = on$ at t = 1, then $D = on$ at t = 2. Or, taking an extended half life of e.g., C into account, the following relationship could be defined; if $A = on$ and $B = on$ at t = 0, then $C = on$ between t = 1 and t = 4. Since these modifications could be achieved in a discrete manner, the essential characteristics of Boolean networks, i.e., determinism and attractor formation will not be affected. Alternatively, phase and time lags could be modeled by introducing virtual "carrier" elements that would uphold or delay a function beyond a direct causal link. After all, Boolean networks are capable of universal computation. But this approach would disrupt the direct correlation between element and expressed gene, important for a consistent analogy between theoretical and experimental networks. The state space opened up by the introduction of phase and lag parameters is even more vast than what has been considered so far, again leading to the question of how order and complexity can be sought out of so many possibilities. However, arguing as before that original order could plausibly have been formed in small, random networks [31, 36], duplication and variations of a theme may serve as a sure foundation for network order as parameter space is increased.

Experimental Design for Extracting the Architecture of Genetic Networks

Genetic networks offer an extraordinary opportunity for the exploration of biological systems for what they are, *parallel processing molecular systems*. Boolean networks as they now stand provide us with a concise description of essential qualitative features of combinatorial networks. While we anticipate that the language of Boolean networks will evolve further as a direct means of modeling aspects of living genetic networks, much territory in experimental science may be explored now with emerging technologies for the massively parallel measurement of gene expression. On the cellular level, the sensitivity of RTPCR (reverse transcription polymerase chain reaction) combined with a fluorescent endpoint could be used to measure gene expression in single cells with FACS (fluorescent activated cell sorter) analysis [39, 63]. Distribution of the expression level of a constitutive or induced gene in a homogenous cell population will provide insight into the quantal or continuous nature of mRNA expression and the kinetics of induction, two important questions regarding the nature

of gene regulatory mechanisms. If dynamic attractors are prevalent, more genes that fluctuate on the individual cell level should be found. In an RTPCR/FACS scan, such genes would produce a ratio of expressing/non-expressing cells that corresponds to the attractor time interval during which the gene is active.

The expression level of a gene is the elementary state variable in a genetic network. In the extended interpretation of the network (Figure 1) all regulatory functions funnel into a combinatorial code that determines whether a gene will be turned *on* or *off*. The genetic network as formulated here can be experimentally accessed by measuring the expression of genes on the level of their mRNAs. While the parallel measurement of large numbers of biochemical or physiological system variables generally represents a formidable challenge, the characterization of mRNA levels is now possible and only depends on the automation of existing technologies such as the ultrasensitive RTPCR. It has been shown that quantitative RTPCR of a small gene family can provide data of sufficient quality to allow mathematical modeling of a developmental gene expression regulatory process ([62]; also discussed above). The same experimental strategy in combination with robotics could be used to measure the expression of gene families. Assays could realistically be conducted in sets of 100, e.g., a combination of 10 genes with 10 mRNA samples, requiring only ten sets of reactions for the characterization of the expression of 100 genes. Further increase in scale following improved automation and anticipating the sequencing of new genes could begin to approximate an expression fingerprint covering the total number of genes possibly expressed by a cell type (~10,000; see [1]). The contribution of robotics technology in making this approach economical would have to be matched by an equal gain in reagent efficiency, especially gene specific PCR primer oligonucleotides. This could be accomplished by creating a primer database/repository with the purpose of cataloging and synthesizing gene-specific primers in bulk. These would be made available to individual research programs in order to reduce the cost per sequence and facilitate the establishment of a standardized protocol that would be shared by many laboratories.

Gene expression maps would be beneficial for the study of paradigms that have already been shown to be interesting from the genetic network perspective, such as the cell cycle and biological rhythms. But foremost in ontogeny we find a need for systematic gene expression maps, in order that we may open a window onto the most complex genetic programs in nature. While this approach may appear mostly descriptive at first hand, a precise characterization of the natural state, where cause and effect are generated within the system, must precede any experiments designed to test an hypothesis through targeted perturbations. Starting from such maps, correlations between genes could be established to define expression groups in gene space and along the temporal dimension. Groups of genes systematically expressed in parallel could give us a clue to wiring and rules, i.e., shared *cis* and *trans* mechanisms. Time series of expressed genes, i.e., temporal groups, will help to identify causal interactions and series that could be described as attractor or tree structures. The combination of both may be modeled by Boolean networks or modified variants, leading to the formulation of competing network hypotheses. Efforts in computational reverse engineering of genetic networks are now under way.[4]

The management of these data together with results obtained from other studies of general and specific *cis/trans* regulation of gene expression would benefit by the establishment of a genetic network database akin to or associated with GenBank [6] or the Transcription Factor Database [19]. Here our knowledge of *cis* structures of genes, their combinatorial regulation, and output wiring of *trans* acting genes, could be systematically cataloged and compared to parallel gene expression maps. Such a database of gene connectivity could follow efforts that are now being undertaking in implementing a Gene Expression Database (GXD).[5]

Modeling the complete genetic network should not be a prerequisite for significant insight, since considerations of redundancy and overlap may allow a gradual approach to characterizing the whole system once a minimal number of regulatory elements has been identified to form a regulatory core. Once plausible wiring and rule propositions have been established by correlation and network analysis, targeted perturbations (e.g., manipulation of intercellular signaling factors, combinatorial transgenic knockouts, overexpression of genes etc.) of the network would be appropriate to test model predictions. While such a research program may seem ambitious at first glance, the fundamental genetic, robotics, computer and database technologies exist now and will most likely improve significantly in the near future. Considering the on-going, immense developments in molecular biological and information processing technologies, would the marriage of the two in the exploration of genetic networks not provide an unprecedented opportunity for insight into complex systems? If only a small part of this potential were realized, the implications for discovering and treating the underlying causes of multigenic dysfunctions such as cancer, genetic, autoimmune and degenerative disorders could be revolutionary [17, 44, 51]. Also in the field of regeneration, precise knowledge on the systems level will be the foundation for the controlled recapitulation of developmental programs of tissues that have a naturally limited regenerative potential, such as nerve and muscle.

ACKNOWLEDGMENTS

The authors would like to thank Andrew Wuensche for help with DDLAB [78], and Jeffery Barker, Stefanie Fuhrman, Delwood Richardson, Joerg Stucki, and Gabor Szabo for valuable discussions on the manuscript.

NOTES

1. For computation of Boolean network trajectories and basins of attraction (as shown in the figures) the authors have relied on the outstanding Discrete Dynamics Lab (DDLAB) software by A. Wuensche [75], available on the internet (http://alife.santafe.edu/alife/software/ddlab.html).
2. The authors would like to emphasize that genetic determinism as stated here reflects a conceptual approach for simplifying basic physical aspects of life, and not a philosophical conviction that an organism can be explained in such terms from the level of higher order structures which are created in an interaction with the environment, such as nervous system functions governing behavior (for a detailed discussion, see [43]).
3. Steve Harris, Andy Wuensche, Stuart Kauffman. Manuscript in preparation.
4. Manor Askenazi, Santa Fe Institute; Roland Somogyi, NIH. See WWW site "Reverse Engineering Genetic Networks", http://www.santafe.edu:80/~manor/gene.html.
5. The Gene Expression Database (GXD), a collaborative effort of The Jackson Laboratory, the MRC and University in Edinburgh, and associated laboratories in North America and Europe. See WWW site, http://www.informatics.jax.org/doc/gxdgen.html#concept

REFERENCES

1. B. Alberts, D. Bray, J. Lewis, M. Raff, K. Roberts, and J. D. Watson: Molecular biology of the cell. Garland Publishing, New York, 1994.
2. N. Amariglio and G. Rechavi: Environ. Mol. Mutagen. 21: pp. 212–218, 1993.
3. P. Angel and M. Karin: Biochim. Biophys. Acta. 1072: pp. 129–157, 1991.
4. A. Arkin and J. Ross: Biophys. J. 67: pp. 560–578, 1994.
5. W. R. Ashby: Design for a brain: The origin of adaptive behavior Chapman and Hall, London, 1952.
6. D. Benson, M. Boguski, D. J. Lipman, and J. Ostell: Genomics 6: pp. 389–391, 1990.
7. S. Bianchi, S. Fabiani, M. Muratori, A. Arnold, K. Sakaguchi, T. Miki, and M. L. Brandi: Biochem. Biophys. Res. Commun. 204: pp. 691–700, 1994.
8. C. Chen, D. G. Rainnie, R. W. Greene, and S. Tonegawa: Science 266: pp. 291–294, 1994.
9. A. J. Copp: Trends Genet. 11: pp. 87–93, 1995.
10. K. L. Crossin: Perspect. Develop. Neurobiol. 2: pp. 21–32, 1994.
11. S. U. Devaskar, B. S. Singh, L. R. Carnaghi, P. A. Rajakumar, and S. J. Giddings: Regul. Pept. 48: pp. 55–63, 1993.
12. H. B. Dowse and J. Ringo: Is the circadian clock a "metaoscillator"? Evidence from studies of circadian rhythms in Drosophila. In: The molecular biology of circadian rhythms, M. Young (Ed.) Marcel Dekker, New York, 1992, pp. 195–220.
13. J. C. Dunlap: Annu. Rev. Physiol. 55: pp. 683–728, 1993.
14. I. Edery, J. E. Rutila, and M. Rosbash: Science 263: pp. 237–240, 1994.
15. S. J. Field, R. S. Johnson, R. M. Mortensen, V. E. Papaioannou, B. M. Spiegelman, and M. E. Greenberg: Proc. Natl. Aca. Sci. USA 89: pp. 9306–9310, 1992.
16. F. Fogelman-Solie, E. Goles-Chacc, and G. Weisbuch: Bull. Math. Biol. 44: p. 715, 1982.
17. T. Friedmann: Ann. Med. 24: pp. 411–417, 1992.
18. L. A. Galli-Taliadoros, J.D. Sedgwick, S.A. Wood, and H. Korner: J. Immunol. Methods 181: pp. 1–15, 1995.
19. D. Ghosh: Trends Biochem. Sci. 16: pp. 455–457, 1991.
20. S. J. Giddins and L. R. Carnaghi: J. Biol. Chem. 263: pp. 3845–3849, 1988.
21. L. Glass and S. Kauffman: J. Theor. Biol. 34: pp. 219–237, 1972.
22. H. Gomi, T. Yokoyama, K. Fujimoto, T. Ikeda, A. Katoh, T. Itoh, and S. Itohara: Neuron 14: pp. 29–41, 1995.
23. J. C. Hall: Trends Neurosci. 18: pp. 230–240, 1995.
24. N. Hama, F. Paliogianni, B. J. Fessler, and D. T. Boumpas: J. Exp. Med. 181: pp. 1217–1222, 1995.
25. P. I. Hanson and H. Schulman: Annu. Rev. Biochem. 61: pp. 559–601, 1992.
26. X. He and M. G. Rosenfeld: Neuron 7: pp. 183–196, 1991.
27. C. S. Hill and R. Treisman: Cell 80: pp. 199–211, 1995.
28. J. J. Hopfield and D. W. Tank: Disordered Systems and Biological Organization, E. Bienenstock et al., (Ed.) Springer-Verlag, Berlin, 1986, pp. 155–170.
29. A. L. Hughes: Proc. R. Soc. Lond. B 256: pp. 119–124, 1994.
30. F. Jacob and J. Monod: 21st Symp. Soc. Study of Development and Growth. Academic Press, 1963.
31. S. A. Kauffman: J. Theoretical Biol. 22: pp. 437–467, 1969.
32. S. A. Kauffman: J. Theorectical Biol. 44: pp. 167–190, 1974.
33. S. A. Kauffman: Physica D 10 pp. 145–156, 1984.
34. S. A. Kauffman: Boolean systems, adaptive automata, evolution. In: Disordered Systems and Biological Organization, E. Bienenstock et al. (Ed.), Springer, New York, 1986, pp. 339–360.
35. S. A. Kauffman: SFI Studies in the Sciences of Complexity, Addison Wesley, Reading, MA, 1990: 8, pp. 151–192.
36. S. A. Kauffman: The origins of order, self-organization and selection in evolution. Oxford University Press, New York, 1993.
37. M. Kimura: The neutral theory of molecular evolution. Cambridge University Press, New York, 1993.
38. P. H. King, T. D. Levine, R. T. Fremeau, and J. D. Keene: J. Neurosci. 14: pp. 1943–1952, 1994.
39. M. S. H. Ko: Bio Essays 14: pp. 341–346, 1992.
40. C. Langton: Studying artificial life with cellular automata, Physica D 22: pp. 120–149, 1986.
41. C. Langton: Computation at the edge of chaos: phase transitions and emergent computation, Physica D 42: pp. 12–37, 1990.
42. B. Lewin: Genes V. Oxford University Press, New York, 1994.
43. R. C. Lewontin: Biology as ideology. Harper Collins, New York, 1993.
44. G. Manenti, G. Binelli, M. Gariboldi, F. Canzian, L. de Gregorio, F. S. Falvella, T. A. Dragani, and M. A. Pierotti: Genomics 23: pp. 118–124, 1994.
45. T. Miyata, K. Kuma, N. Iwabe, and N. Nikoh: Jpn. J. Genet. 69: pp. 473–480, 1994.
46. J. Monod and F. Jacob: Cold Spring Harb. Symp. quant. Biol. 23: pp. 389–401, 1961.
47. T. Motokura and A. Arnold: Curr. Opin. Genet. Dev. 3: pp. 5–10, 1993.
48. M. Ohtsuki, S. Flanagan, I. M. Freedberg, and M. Blumenberg: Gene Expression 3: pp. 201–213, 1993.
49. T. L. Page: Science 263: pp. 1570–1572, 1994.
50. D. S. Peeper, A. J. van der Eb, and A. Zantema: Biochim. Biophys. Acta. 1198: pp. 215–230, 1994.
51. G. G. Petranyi: Immunol. Today 13: pp. A19–20, 1992.
52. G. P. J. Reddy: Cell. Biochem. 54: pp. 379–86, 1994.
53. J. Richelle: Dynamical Systems and Cellular Automata, J. Demongeot, E. Goles, and M. Tchuente (Eds.) Academic Press, New York, 1985, pp. 247–254.
54. L. M. Robertson, T. K. Kerppola, M. Vendrell, D. Luk, R. J. Smeyne, C. Bocchiaro, J. I. Morgan, and T. Curran: Neuron 14: pp. 241–252, 1995.
55. M. A. Rudnicki, P. N. J. Schnegelsberg, R. H. Stead, T. Braun, H. -H. Arnold, and R. Jaenisch: Cell 75: pp. 1351–1359, 1993.
56. C. J. Schoenherr and D. J. Anderson: Nature 267: pp. 1360–1363, 1995.
57. B. S. Shastry: Molec. Cell. Biol. 136: pp. 171–182, 1994.
58. C. J. Sherr, H. Matsushime, and M. F. Roussel: Ciba. Found. Symp. 170: pp. 209–219, 1992.

59. R. Somogyi and J. W. Stucki: J. Biol. Chem. 266: pp. 11068–11077, 1991.
60. R. Somogyi, M. Zhao, and J. W. Stucki: Biochem. J. 286: pp. 869–877, 1992.
61. J. W. Stucki and R. Somogyi: Biochim. Biophys. Acta 1183: pp. 453–472, 1994.
62. R. Somogyi, X. Wen, W. Ma, and J. L. Barker: J. Neurosci. 15: pp. 2575–2591, 1995.
63. N. J. Sucher and D. L. Deitcher: Neuron 14: pp. 1095–1100, 1995.
64. M. Sugita: J. Theor. Biol. 42: pp. 563–585, 1963.
65. R. Thomas: J. Theor. Biol. 42: pp. 563–585, 1973.
66. R. Thomas: Dynamical systems and cellular automata, J. Demongeot, E. Goles, M. Tchuente (Eds.) Academic Press, New York, 1985, pp. 269–282.
67. A. Turing: Proc. London Math. Soc. Series 2 42: pp. 230–265, 1936.
68. Ph. Van Ham: Dynamical systems and cellular automata, J. Demongeot, E. Goles, M. Tchuente (Eds.) Academic Press, New York, 1985, pp. 283–291.
69. J. Von Neumann: The general and logical theory of automata. collected works, vol. 5. McMillan, New York, reprinted 1963, 1951.
70. J. Von Neumann: Theory of self-reproducing automata. A.W. Burks (Ed.), University of Illinois Press, 1966.
71. C. C. Walker: J. Cybernetics 1: pp. 55–67, 1971.
72. G. Weisbuch: Networks of automata and biological organization, J. Theor. Biol. 121: pp. 255–267, 1986.
73. S. Wolfram: Statistical mechanics of cellular automata, Rev. Mod. Phys. 55: pp. 601–644, 1983.
74. S. Wolfram: Universality and complexity in cellular automata, Physica D 10: pp. 1–35, 1984.
75. A. Wuensche and M. J. Lesser: The global dynamics of cellular automata, SFI studies in the sciences of complexity, vol. 1. Addison Wesley, Reading, MA, 1992.
76. A. Wuensche: The ghost in the machine: Basins of attraction in random boolean networks. In: Artificial life III. C. G. Langton (Ed.), 1993, pp. 465–501.
77. A. Wuensche: Complexity in one-D cellular automata: gliders, basins of attraction and the Z parameter, Cognitive Science Research Paper 321, U. of Sussex, 1994.
78. A. Wuensche: Discrete Dynamics Lab (DDLAB), software and documentation at MIT Press Artificial Life Online (alife.santafe.edu/alife/software/ddlab.html). 1995.