Coarse-graining and self-dissimilarity of complex networks

Shalev Itzkovitz, Reuven Levitt, Nadav Kashtan, Ron Milo, Michael Itzkovitz, and Uri Alon
Departments of Molecular Cell Biology and Physics of Complex Systems, Weizmann Institute of Science, Rehovot, Israel 76100
(Received 13 May 2004; revised manuscript received 21 July 2004; published 21 January 2005)

Can complex engineered and biological networks be coarse-grained into smaller and more understandable versions in which each node represents an entire pattern in the original network? To address this, we define coarse-graining units as connectivity patterns which can serve as the nodes of a coarse-grained network and present algorithms to detect them. We use this approach to systematically reverse-engineer electronic circuits, forming understandable high-level maps from incomprehensible transistor wiring: first, a coarse-grained version in which each node is a gate made of several transistors is established. Then the coarse-grained network is itself coarse-grained, resulting in a high-level blueprint in which each node is a circuit module made of many gates. We apply our approach also to a mammalian protein signal-transduction network, to find a simplified coarse-grained network with three main signaling channels that resemble multi-layered perceptrons made of cross-interacting MAP-kinase cascades. We find that both biological and electronic networks are "self-dissimilar," with different network motifs at each level. The present approach may be used to simplify a variety of directed and nondirected, natural and designed networks.

DOI: 10.1103/PhysRevE.71.016127 PACS number(s): 89.75.Fb

I. INTRODUCTION

In both engineering and biology it is of interest to understand the design of complex networks [1–3], a task known as "reverse engineering." In electronics, digital circuits are engineered from the top down starting from functional blocks, which are implemented using logic gates, which in turn are implemented using transistors [4]. Reverse engineering of an electronic circuit means starting with a transistor map and inferring the gate and block levels. Current approaches to reverse engineering of electronic circuits usually require prior knowledge of the library of modules used for forward engineering [5,6].

In biology, increasing amounts of interaction networks are being experimentally characterized, yet there are few systematic approaches to simplify them into understandable blueprints [3,7–18].

Here we present an approach for simplifying networks by creating coarse-grained networks in which each node is a pattern in the original network. This approach is based on network motifs, significant patterns of connections that recur throughout the network [19–22]. We define coarse-graining units (CGUs), which can be used as nodes in a coarse-grained version of the network. We demonstrate this approach by coarse-graining an electronic and a biological network.

Definition of CGUs. CGUs are patterns which can optimally serve as nodes in a coarse-grained network. One can think of CGUs as elementary circuit components with defined input and output ports and internal computational nodes. The set of CGUs comprises a dictionary of elements from which a coarse-grained version of the original network is built. The coarse-grained version of the network is a new network with fewer elements, in which some of the nodes are replaced by CGUs.

Our approach to define CGUs is loosely analogous to coding principles and to dictionary text compression techniques [23,24]. The goal is to choose a set of CGUs that (a) is as small as possible, (b) each of which is as simple as possible, and which (c) make the coarse-grained network as small as possible. These three properties can be termed "conciseness," "simplicity," and "coverage." Conciseness is defined by the number of total CGU types in the dictionary set. Coverage is the number of nodes and edges eliminated by coarse-graining the network using the CGUs. To define simplicity, we describe each occurrence of the subgraph, G, as a "black box." The black box has input ports and output ports, which represent the connections of G to the rest of the network, R (Fig. 1). There can be four types of nodes in G: input nodes that receive only incoming edges from R, output nodes that have only outgoing edges to R, internal nodes with no connection to R, and mixed nodes with both incoming and outgoing edges to R. To obtain a minimal loss of information, a coarse-grained version of G includes ports, which carry out the interface to the rest of the network. The number of ports in the black box representing G is

$$H = I + O + 2M, \tag{1}$$

where I is the number of input nodes, O the number of output nodes, and M the number of mixed nodes (internal nodes do not contribute ports and each mixed node contributes two ports). The lower the number of ports, H, the more "simple" the CGU.

After defining simplicity, coverage, and conciseness, one can choose the optimal set of CGUs. To choose the optimal set of CGUs, we maximize a scoring function that combines these features:

$$S = E_{covered} + \alpha \Delta P - \beta N - \gamma \sum_{i=1}^{N} T_i, \tag{2}$$

where $E_{covered}$ is the number of edges covered by all occurrences of the CGUs and therefore eliminated in the coarse-grained network. N is the number of distinct CGUs in the set, and T_i is the number of internal nodes in the ith CGU. ΔP is

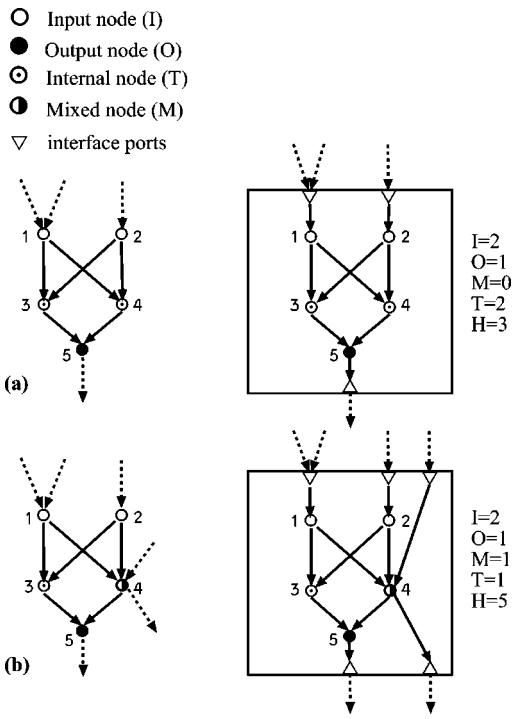


FIG. 1. Black box representation of a subgraph and the classes of nodes and ports. The nodes of the subgraph (numbered 1–5) are classified into input (I), output (O), internal (T), and mixed (M) nodes according to the edges that connect them to the rest of the network (dashed arrows). The subgraph is represented as a black box with input and output ports (right side of figure). The complexity measure H is the total number of ports, (a). Subgraph with no mixed nodes. The connectivity profile vector is (I,I,T,T,O) (b) subgraph with a mixed node. The connectivity profile vector is (I,I,T,M,O).

the difference between the number of nodes in the original network and the number of nodes and ports in the coarse-grained network:

$$\Delta P = P_{covered} - \sum_{i=1}^{N} n_i H_i, \tag{3}$$

where $P_{covered}$ is the number of nodes covered by all occurrences of the CGUs, n_i is the number of occurrences in the network of CGU i, and H_i is the number of ports of CGU i. Using this we obtain

$$S = [E_{covered} + \alpha P_{covered}] - \left[\alpha \sum_{i=1}^{N} n_i H_i + \beta N + \gamma \sum_{i=1}^{N} T_i \right]. \tag{4}$$

The scoring function has two terms: The first term, related to coverage, corresponds to the simplification gained by coarse-graining. The second term, corresponding to simplicity and conciseness, quantifies the complexity of the CGU dictionary. Maximizing S favors use of a small set of CGUs, preferentially those that appear often, with many internal nodes and few mixed nodes (since internal nodes do not contribute ports to H_i and mixed nodes contribute two ports).

The last term in the scoring function, which is the total number of internal nodes in the dictionary, prevents the trivial solution where the entire network is replaced by a single complex CGU. α, β, γ are parameters that can be set for various degrees of coarse-graining (the results below are insensitive to varying these parameters over 3 orders of magnitude).

We use simulated annealing [25,26] to find the optimal set of CGUs for coarse-graining: There is potentially a huge number of subgraphs that can serve as candidate CGUs. To reduce the number of candidate subgraphs and to focus on those likely to play functional roles, we consider only subgraphs that occur in the network significantly more often than in randomized networks: network motifs [19–22]. A candidate set of CGUs is obtained by first detecting all network motifs of 3–6 nodes (Appendix A). The nodes of every occurrence of each motif are classified to one of the four types (input, output, internal, or mixed). This defines a connectivity profile for each occurrence. For example, the two subgraphs in Fig. 1 have the profiles (I,I,T,T,O) and (I,I,T,M,O), where I, T, O, and M represent input, internal, output, and mixed nodes, respectively. The occurrences of each motif are then grouped together according to their profile to form a CGU candidate.[1] A CGU candidate of n nodes is thus characterized by its topology (an $n \times n$ adjacency matrix) and by a profile vector of length n of node classifications (Fig. 3).

In the simulated annealing optimization algorithm, each CGU candidate is assigned a random spin variable which is either 1 if all its occurrences participate in the coarse-graining or 0 otherwise. CGU candidates with spin 1 compose the "active set." At each step a spin is randomly chosen and flipped, and the coarse-graining score for the new active set is computed. The active set is updated according to a Metropolis Monte Carlo procedure [26].[2]

Once an optimal set of CGUs is found, a coarse-grained representation of the original network is formed by replacing each occurrence of a CGU with a node (Appendix B). Generally the coarse-grained representation is a hybrid in which some nodes represent CGUs and other nodes are the original nodes. The algorithm can be repeated on the coarse-grained representation to obtain higher levels of coarse-graining.

Note that the coarse-graining problem is quite different from the well-studied circuit partitioning problem [27] and from the detection of community structure in networks [28–30]. These algorithms seek to divide networks into subgraphs with minimal interconnections, usually resulting in a set of distinct and rather complex subgraphs. In contrast, coarse-graining seeks a small dictionary of simple subgraph types in order to help understand the function of the network in terms of recurring independent building blocks. An anal-

[1]Two subgraph occurrences with connectivity profile vectors V_1 and V_2 are grouped together if there exists a permutation P of the nodes that preserves the subgraph structure and for which the permuted profile vectors obey $P(V_1) = V_2$.

[2]The new active set is accepted with probability $\min\{1, e^{\Delta S/T}\}$, where ΔS is the score difference from the previous active set and T is an effective temperature, lowered by a factor of 5% between sweeps.

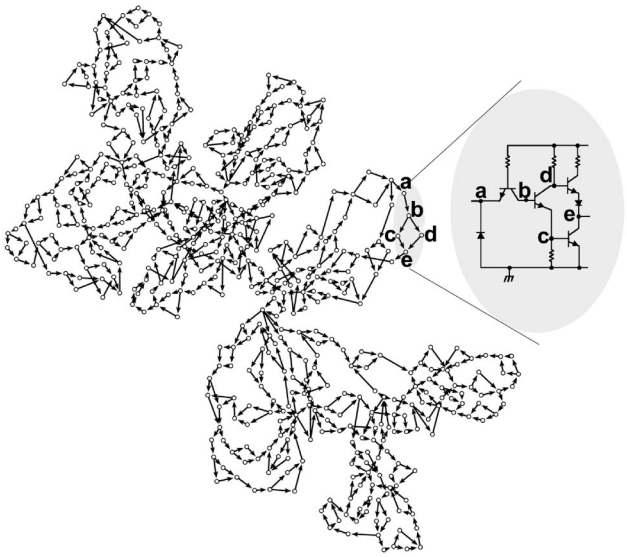


FIG. 2. Transistor level map of an 8-bit binary counter (ISCAS89 circuit S208 [31]). Nodes are junctions between transistors, and directed edges represent wire connections. Highlighted is a subgraph that represents the transistors that make up one NOT gate.

ogy is the detection of words in a text, from which spaces and punctuation marks have been removed, without prior knowledge of the language.

II. COARSE-GRAINING OF AN ELECTRONIC CIRCUIT

To demonstrate the coarse-graining approach we analyzed an electronic circuit derived from the ISCAS89 benchmark circuit set [31,32]. The circuit is a module used in a digital fractional multiplier (*S*208 [31]). The circuit is given as a netlist of five gate types (AND, OR, NAND, NOR, NOT) and a D-flipflop (DFF). To synthesize a transistor level implementation of this circuit (Fig. 2) we first replaced every DFF occurrence with a standard implementation using four NAND gates and one NOT gate [4]. All gates were then replaced with their standard transistor-transistor logic (TTL) implementation [33], where nodes represent junctions between transistors (for this purpose resistors and diodes were ignored, as were ground and Vcc). The resulting transistor network (Fig. 2) has 516 nodes and 686 edges.

Four CGUs were detected in the transistor network, each with five or six nodes [Figs. 3 and 4(a)]. These patterns correspond to the transistor implementations of the five basic logic gates AND, NAND, NOR, OR, and NOT [Fig. 4(a)]. These CGUs were used to form a coarse-grained version of the network in which each node is a CGU. In this case coverage was complete and all of the original nodes were included within CGUs. This network, termed the "gate-level network," had 99 nodes and 153 edges.

We next iterated the coarse-graining process by applying the algorithm to the gate-level network. One CGU with six nodes (gates) was detected. This CGU corresponds to a

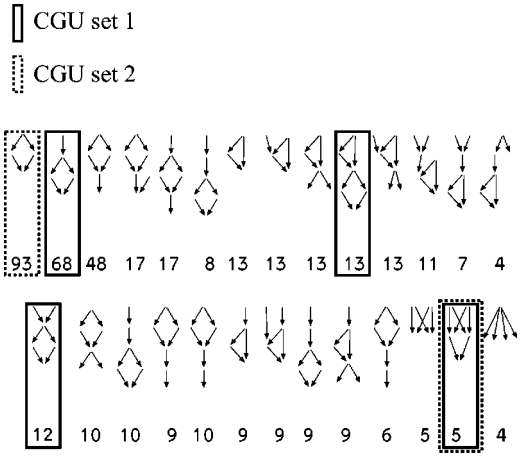


FIG. 3. A partial set of the network motif candidate CGUs for the transistor level network. The number of occurrences of each motif in the transistor network is shown. The optimal CGU dictionary consists of four units (solid boxes, CGU set 1, $\alpha=0.2$, $\beta=20$, $\gamma=0.01$). A second optimal solution consisting of two units, which is found for high values of β is also shown (dashed box, CGU set 2, $\alpha=0.2$, $\beta=500$, $\gamma=0.01$). Note that several CGU candidates share the same motif topology. They differ by their connectivity profile vectors (input/output/internal/mixed).

D-flip-flop with an additional logic gate [Fig. 4(b)]. A "flip-flop level" coarse-grained network was then formed with nodes which were either gates or flip-flops. This network had 59 nodes and 97 edges.

We applied the coarse-graining algorithm again to the flip-flop level network. Two types of CGUs were found

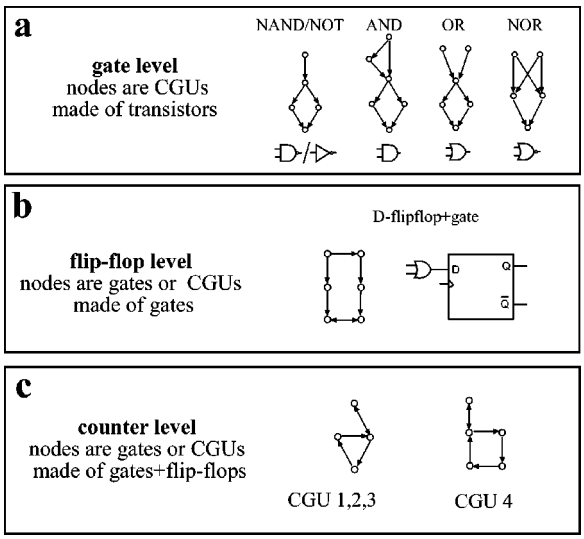


FIG. 4. The CGUs found in the different coarse-grained levels of the electronic circuit. At the gate level the CGUs are the TTL implementation of AND, OR, NAND, NOR, and NOT gates (NAND and NOT differ by the type of transistor at the input). At the flip-flop level, a single CGU, occurring 8 times is found. This CGU corresponds to the five-gate implementation of a D-flip-flop with an additional gate at the input. At the counter level, two CGU topologies are found: Seven occurrences of a three-node feedback loop + mutual edge and one occurrence of a four-node feedback loop + mutual edge, representing CGU4.

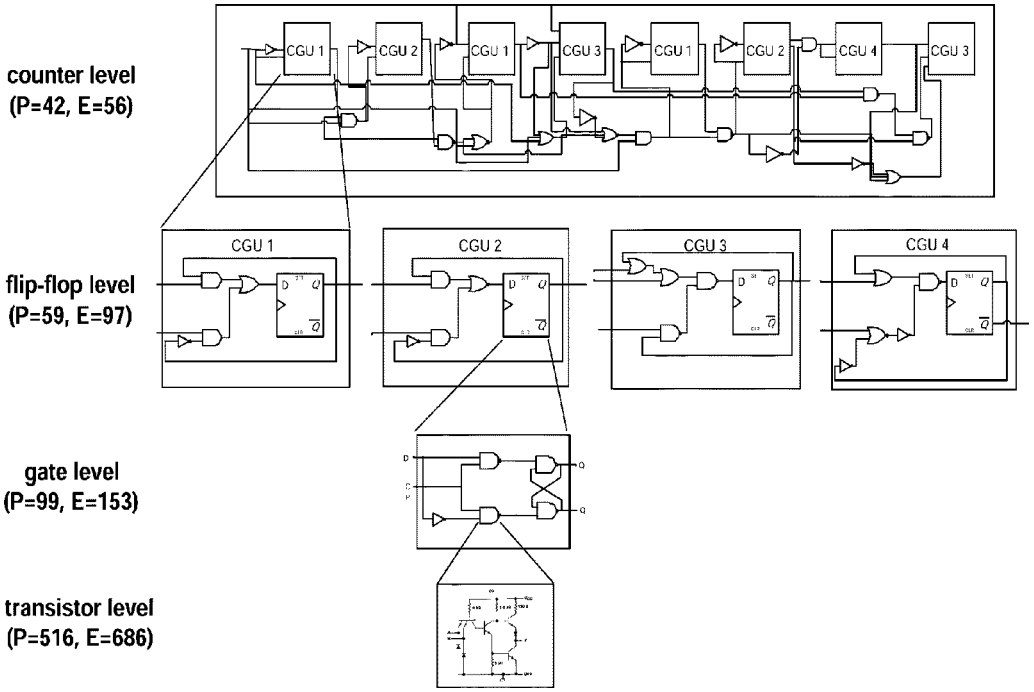


FIG. 5. Four levels of representation of the 8-bit counter electronic circuit. In the transistor level network, nodes represent transistor junctions. In the gate level, nodes are CGUs made of transistors, each representing a logic gate. Shown is the CGU that corresponds to a NAND gate. In the flip-flop level, nodes are either gates or a CGU made of gates that corresponds to a D-type flip-flop with an additional logic gate at its input. In the counter level, each node is a gate or a CGU of gates/flip-flops that corresponds to a counter subunit. Numbers of nodes (P) and edges (E) at each level are shown.

[Fig. 4(c)], which correspond to units of a digital counter. Using these CGUs, we constructed the highest-level coarse-grained network in which each node is either a CGU or a gate. This network, depicted in Fig. 5 top panel, had 42 nodes and 56 edges. Thus, the highest-level coarse-grained network has about 12-fold fewer nodes and edges than the original transistor-level network. This high-level map corresponds to sequential connections of binary counter units, each of which halves the frequency of the binary stream obtained from the previous unit. This map thus describes an 8-bit counter [34].

In other electronic circuits, we find other CGUs, including a XOR built of four NAND gates [4,22] (data not shown). The coarse-graining approach appears to automatically detect favorite modules used by electronic engineers.

III. COARSE-GRAINING OF BIOLOGICAL NETWORKS

Recent studies have shown that biological networks contain significant network motifs [19–22]. Theoretical and experimental studies have demonstrated that each network motif performs a key information processing function [3,17–19,35–40]. A coarse-grained version of biological networks is of interest because it would provide a simplified representation, focused on these important subcircuits. However, whereas electronic circuits are composed of exact copies of library units, in biology the recurring units may not have precisely the same structure. In addition, the characterization of signaling and regulatory networks is currently incomplete due to experimental limitations. Thus a more flexible definition of CGUs is needed [41]. To address these issues we modify our algorithm by allowing each CGU to represent a family of subgraphs, which share a common architectural theme. Thus, the CGUs are *probabilistically generalized network motifs (PGNMs)*: network motifs of different sizes which approximately share a common connectivity pattern.

Probabilistic generalization of network motifs. To define PGNMs, we must first discuss the concept of block models [42–44]. A block model is a compact representation of a subgraph. It consists of two elements: (1) a partition of the subgraph nodes into discrete subsets called *roles* [22] and (2) a statement about the presence or absence of a connection between roles (Fig. 6). A subgraph of n nodes can be described by an adjacency matrix G, where $G_{ij}=1$ if a directed edge exists from node i to node j and $G_{ij}=0$ if there is no connection. A block model partitions the n nodes into $m \leqslant n$ roles according to *structural equivalence*. Two nodes are structurally equivalent if they share exactly the same connections to all other nodes. The block model is an $m \times m$ matrix A, where $A_{IJ}=1$ means that all nodes which share role I have a directed connection to all nodes which share role J (Fig. 6).

In large subgraphs of real-world networks, perfect structural equivalence is not always seen. A block model can still be used as an idealized structure which can be compared to a

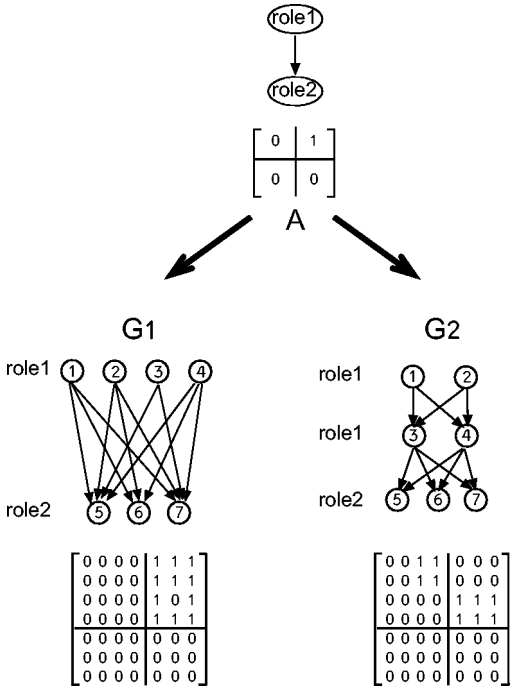


FIG. 6. A block-model (top) and two subgraphs, one which fits the block model ($G1$, bottom left) and one which does not ($G2$, bottom right). $G1$ has seven nodes and two roles (nodes 1–4 share role 1 and nodes 5–7 share role 2). Its adjacency matrix is shown below, with lines indicating the block model partition. An edge between node 3 and node 6 is missing for a perfect fit to the proposed block model. The distance between the block matrix and the adjacency matrix is $d=0.1075$. The right subgraph $G2$ does not fit the proposed block model A. The distance between the block matrix and the adjacency matrix is $d=0.7538$. An alternative block model with three roles ({1,2}, {3, 4}, {5, 6, 7}) would perfectly fit this subgraph, with $d=0$. Both of these subgraphs are aggregates of a four-node bifan subgraph (Fig. 7).

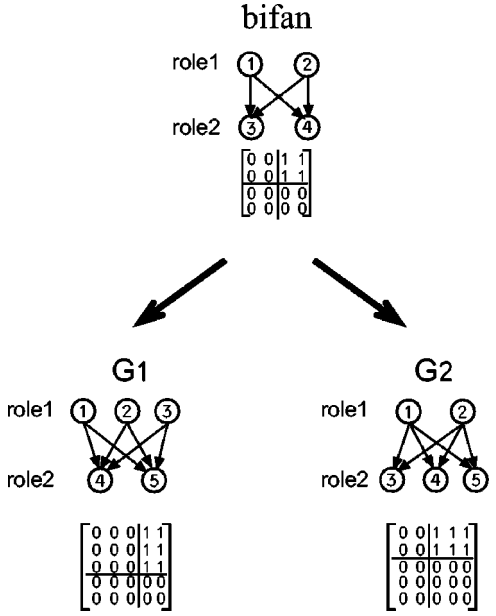


FIG. 7. Topological generalizations of the bifan [19] subgraph and their adjacency matrices. The bifan subgraph has two roles—nodes 1, 2 share role 1 and nodes 3, 4 share role 2. Lines indicate the block-model partition. Below are two generalized subgraphs obtained by role replication [22]. Subgraph $G1$ (left) is obtained by replicating the first role, with its connections. Subgraph $G2$ (right) is obtained by replicating the second role, with its connections. Adjacency matrices and block-model partitions are shown. The role-replication operation extends a subgraph while keeping a perfect fit to the block model of the original subgraph.

given subgraph. The distance between a subgraph and a proposed block model can be defined as[3] [44]

$$d = \frac{S_W}{S_T}, \tag{5}$$

where S_W is the within-block sum of squares,

$$S_W = \sum_I \sum_J \sum_{i\in I, j\in J} (G_{ij} - \langle G_{IJ}\rangle)^2 \tag{6}$$

and S_T is the total sum of squares:

$$S_T = \sum_{i,j} (G_{ij} - \langle G\rangle)^2, \tag{7}$$

where $\langle G\rangle = \Sigma G_{ij}/n^2$ is the mean value of G and $\langle G_{IJ}\rangle$ is the mean of the adjacency matrix values in block $\{I,J\}$. A subgraph with $d=0$ is perfectly described by its block model. For example, subgraph $G1$ in Fig. 6 has $n=7$ nodes. It can be described by a block model with $m=2$ roles. Nodes 1–4 are assigned the first role and nodes 5–7 are assigned the second role. The distance between the subgraph and the proposed block model is $d=0.1075$. Figure 6 also shows a subgraph $G2$, which is far from the proposed block model ($d=0.7538$).

Finding the best block model to fit arbitrary connectivity data is a combinatorially complex problem [42–44], requiring exhaustive testing of different assignments of roles to nodes. An efficient algorithm to detect PGNMs can be formed based on the fact that small network motifs in biological networks aggregate to form *network motif topological generalizations* [22,45]. Topological generalizations are subgraphs obtained from smaller network motifs, by replicating one or more of their roles, together with its connections [22] (Fig. 7). An algorithm to detect PGNMs is described in Appendix C.

To determine the optimal dictionary of CGUs, including the PGNMs, we use the following modified version of the scoring function of Eq. (2):

$$S = E_{covered} + \alpha\Delta P - \beta N - \gamma\sum_{i=1}^{N} T_i - \delta \sum_{i\in\{\mathrm{CGU}_g\}}^{N} d_i, \tag{8}$$

where N, the number of CGUs, is the number of basic motifs used. CGU_g includes the set of all PGNMs based on the

[3]This distance measure accounts for the size of the subgraph and is more appropriate than measures such as the Hamming distance (number of edges which have to be added or removed to obtain a perfect fit to a block-model).

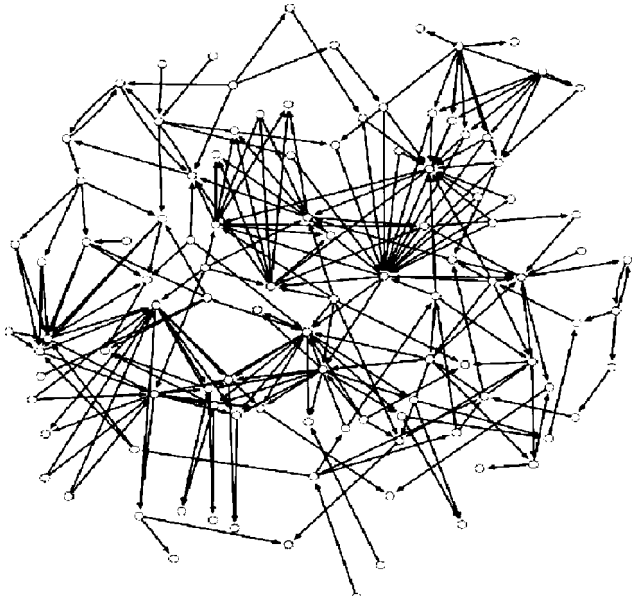


FIG. 8. A network of signal-transduction proteins in mammalian cells.

CGUs. Each CGU can give rise to several PGNMs of different sizes.

IV. CGUs IN A SIGNAL-TRANSDUCTION NETWORK

Cells process information from their environment by means of networks of protein interactions called signal-transduction networks [46–54]. We analyzed a database of mammalian signal transduction interactions based on selected data from the Signal Transduction Knowledge Environment [54] and literature [46–53]. This data set contains 94 proteins and 209 directed interactions (Fig. 8). We find that the optimal coarse graining is based on a single motif—the four-node bifan (Fig. 9). Thus $N=1$. We find nine occurrences of PGNMs based on the bifan, labeled CGU0–CGU8, which share a common design consisting of a row of input nodes with overlapping interactions to a row of output nodes (Fig. 9). The input and output rows in these CGUs sometimes represent proteins from the same subfamily (eg., JNK1, JNK2, and JNK3 in CGU 3), and in other cases they represent proteins from different sub-families (ERK and p38 in CGU 6).

Using this CGU, the signaling network can be coarse-grained [Fig. 10(a)], showing three major signaling channels [Fig 10(b)]. These channels correspond to the well-studied ERK, JNK, and p38 MAP-kinase cascades, which respond to stress signals and growth factors [46–53].

Each channel is made of three CGUs in a cascade. In each cascade, the top and bottom CGUs contain only positive (kinase) interactions, and the middle CGU contains both positive and negative (phosphatase) interactions. The p38 and ERK channels intersect at CGU 6. The MAPK phosphatase 2 (MKP2) participates in both the JNK pathway (CGU2) and the ERK pathway (CGU8), whereas MAPK phosphatase 5 (MKP5) participates in both JNK pathway (CGU2) and the P38 pathway (CGU5). The MAPKKK ASK1 and TAK1 are shared by both the JNK pathway (CGU1) and P38 pathway (CGU4) [51,52].

The structure of each CGU is similar to a single-layer perceptron, and can allow hard-wired combinatorial activation and inhibition of outputs [19,49]. Similar structures are found in transcription regulation networks ("dense overlapping regulons" [19]). However, in transcription regulation networks, these structures are not arranged in cascades. In contrast, the protein-signaling network contains CGU cascades that resemble multilayer perceptrons.

V. SELF-DISSIMILARITY OF NETWORK STRUCTURE

Interestingly, the coarse-grained signaling network displays a different set of network motifs than the original network, with prominent cascades [Fig. 10(c)]. Similarly, the electronic network displayed different CGUs at each level (Fig. 4). These networks are therefore *self-dissimilar* [55,56]: the local structure at each level of resolution is different.

VI. DISCUSSION

We presented an approach for coarse-graining networks in which a complex network can be represented by a compact and more understandable version. We defined optimal units for coarse-graining, CGUs, which allow a maximal reduction of the network, while keeping a concise and simple dictionary of elements. We demonstrated that this method can be used to fully reverse-engineer electronic circuits, from the transistor level to the highest module level, without prior knowledge of the library components used to create them.

For biological networks, where modularity may be less stringent than in electronic circuits, we modified the algorithm to seek a coarse-grained network, using a small set of structures of different sizes that form probabilistically generalized network motifs. Using this approach, a coarse-grained version of a mammalian signaling network was established, using one CGU composed of cross-activating MAP-kinases. In the coarse-grained network one can easily visualize intersecting signaling pathways and feedback loops. The present approach allows a simplified coarse-grained view of this important signaling network, showing the major signaling channels, and specifies the recurring circuit element (CGU) that may characterize protein signaling pathways in other cellular systems and organisms.

Biological and electronic networks are both self-dissimilar [55,56] showing different network motifs on different levels. This contrasts with views based on statistical physics near phase-transition points which emphasize self-similarity of complex systems.

It is important to stress that not every network can be effectively coarse-grained, only networks with particular modularity and topology. The method can readily be applied to nondirected networks. It would be interesting to apply this approach to additional biological networks, to study the systems-level function of each CGU and to study which networks evolve to have a topology that can be coarse-grained.

ACKNOWLEDGMENTS

We thank J. Doyle, H. McAdams, J. E. Ferrell, Y. Shaul, Y. Srebro, E. Dekel, and all members of our lab for valuable

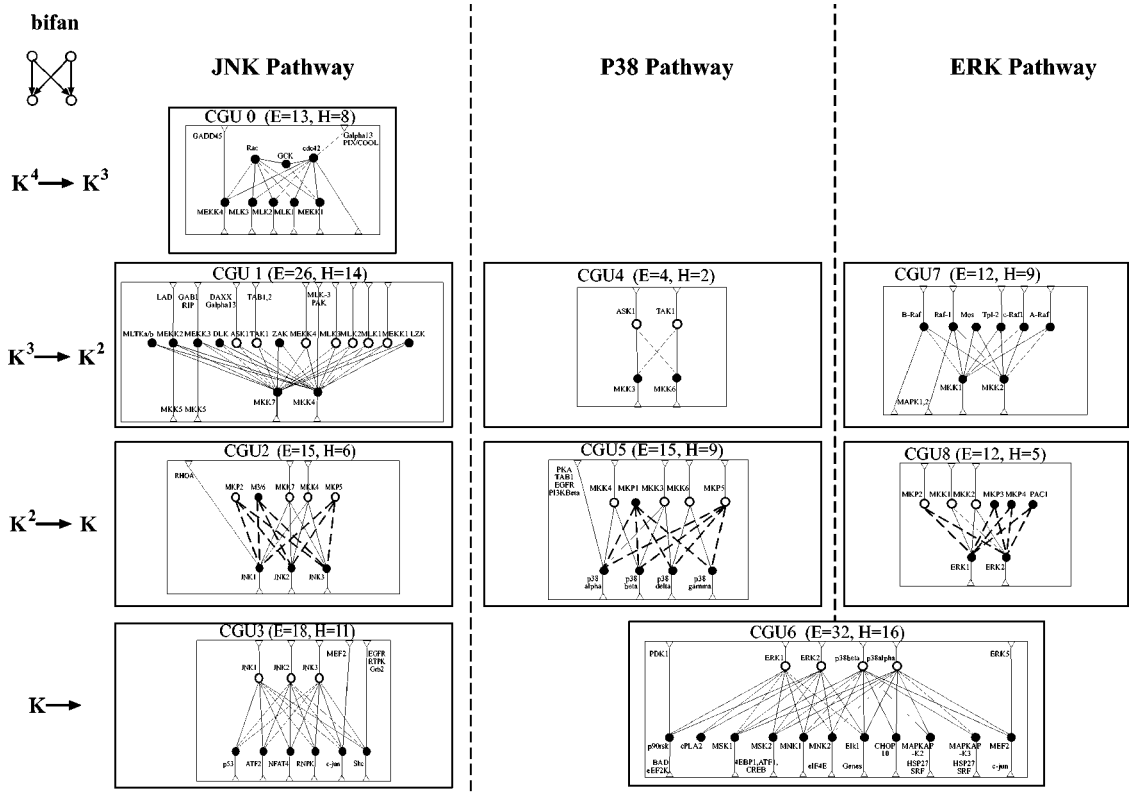
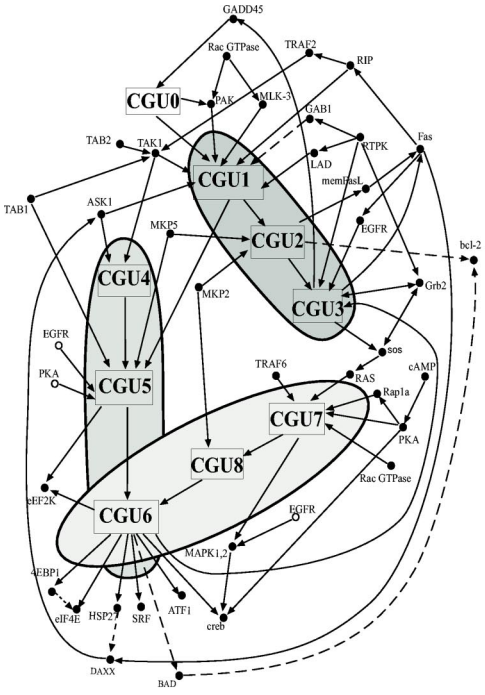


FIG. 9. CGUs in the signal-transduction network. One CGU is found, the four-node bifan with nine PGNMs, numbered CGU0–CGU8. Solid arrows represent positive (kinase) interactions; dashed arrows represent negative (phosphatase) interactions. Open circles represent duplicated nodes (nodes which participate in more than one PGNM). K, K^2, K^3, and K^4 represent MAP-kinase, kinase-kinase, kinase-kinase-kinase, etc. [46–53].



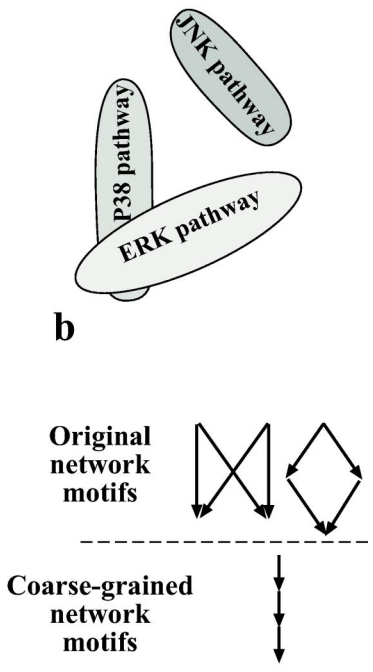


FIG. 10. (a) Coarse-grained version of the signal-transduction network. Three signaling channels made of cascades of the CGU occurrences are highlighted. Solid arrows represent positive (kinase) interactions; dashed arrows represent negative (phosphatase) interactions. EGFR and PKA have been drawn more than once for clarity. (b) The three signaling channels. (c) The network motifs [20] found at the two levels.

discussions. We thank the Israel Science Foundation, NIH, and Minerva. S.I. and R.M. acknowledge support from the Horowitz Complexity Science Foundation.

APPENDIX A: DETECTION OF NETWORK MOTIFS USING RANDOMIZED NETWORKS THAT PRESERVE CLUSTERING SEQUENCES

The set of candidate CGUs should ideally be the complete set of subgraphs of different sizes found in the network. The complete set of subgraphs is, however, too large for the optimization procedure to effectively work in practice (there are 199 four-node connected directed subgraph types, 9364 five-node subgraph types, 1 530 843 six-node subgraph types, etc., a significant fraction of which actually occur in the real networks). Due to computational limitations, we considered in the present study only a small subset of the subgraphs, those which occur significantly more often in the network than in suitably randomized networks. These subgraphs are termed network motifs [19–22].

For the detection of network motifs we considered two randomized ensembles: (1) random networks in which each node preserves the number of incoming, outgoing and mutual edges (edges that run in both direction) connected to it in the real network. (2) Random networks in which each node preserves the number of incoming, outgoing, and mutual edges connected to it in the real network, and in addition each node preserves the clustering coefficient of that node in the real network [1,2,11]. The detection of network motifs, using ensemble (1) as a null hypotheses, was described in [19,20]. The random networks created this way often have a different clustering coefficient for each node than in the real network. As a result, the number of nondirected triangles in the real network is generally different from the randomized network ensemble (either higher, as in the transistor network, or lower, as in the protein signaling network).

To assess the effect of imposing clustering constraints on the randomized networks, we preserve in the more stringent ensemble (2) also the clustering coefficient of each node [1,2,11] ("clustering sequence"), using a simulated annealing algorithm. To create such an ensemble of randomized networks we first randomize the real network with a Markov-chain Monte Carlo algorithm, which successively selects two node pairs and performs a "switch," rewiring their edges, as described in [20,57]. To define the clustering sequence of a network of N nodes, $\{C_i\}_{i=1}^N$, we treat its nondirected version [11]

$$C_i = \frac{2n_i}{K_i(K_i-1)}, \tag{A1}$$

where K_i is the number of edges connected to node i (which represent either incoming, outgoing, or mutual edges in the directed version) and n_i is the number of triangles connected to node i. Denoting the clustering sequence of the random networks by $\{C_i^R\}_{i=1}^N$ we carry out switches, again randomly selecting pairs of edges and rewiring them, but this time with probability

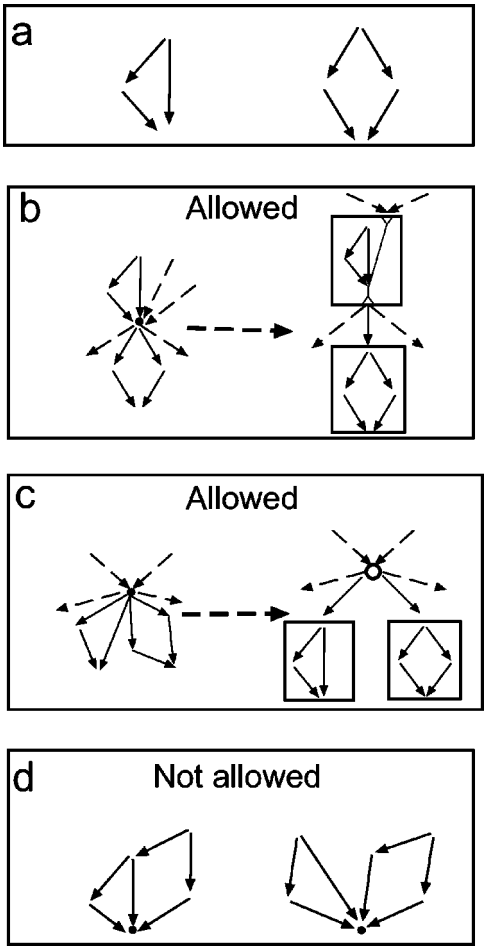


FIG. 11. Overlap rules of CGU candidates. In these examples the CGU candidates are the following: (a) A three-node feed-forward loop (left) and a four-node diamond subgraph (right). (b) Overlap of nodes which receive inputs from only one of the CGUs (left), and coarse-grained representation (right). (c) Overlap of nodes which send outputs to two CGUs (left) and coarse-grained representation (right). Note the addition of a node upstream of the two CGUs, marked with an open circle (○). (d) Two examples of disqualified cases, were a node receives inputs from both CGUs: two CGUs with a common edge (left) and without a common edge (right).

$$\min\{1, e^{-\Delta E/T}\}, \tag{A2}$$

where T is an effective temperature, lowered by a factor of 5% between sweeps, and E, the energy function, is the distance between the clustering sequences of the real and random networks:

$$E = \sum_{i=1}^{N} \frac{|C_i - C_i^R|}{C_i + C_i^R}, \tag{A3}$$

The random networks obtained have precisely the same clustering sequence and degree sequences as the real network. They are thus more constrained than in ensemble (1). In the presently studied networks, they contain almost precisely the same number of nondirected triangles as the real network. However, the numbers of directed triangle subtypes differ

from the real network. There are seven types of directed three-node triangle subgraphs [20]. The relative abundance of these seven subgraphs in the random ensemble is determined by different moments of the degree sequences [58]. Thus, three-node directed subgraphs can still be found as motifs using ensemble (2), depending on the network degree sequences. For the transistor network and signaling network studied, the two sets of network motifs of three to six nodes detected using ensembles (1) and (2) had an overlap of more than 90%. Using ensemble (2) on the transistor network results in somewhat fewer motifs that are triangles with dangling edges, and more treelike motifs than ensemble (1). Using ensemble (2) on the protein signaling network results in somewhat fewer treelike motifs. For both networks, the coarse-graining algorithm detected the same optimal sets of CGUs using either ensembles. Thus, in the present examples, coarse-graining is not affected by choice of random network ensemble.

APPENDIX B: OVERLAP RULES

The desired CGU set should have minimal overlap (shared nodes) between occurrences of the CGUs. In cases where shared nodes are necessary, the CGU partitioning should be such that the shared nodes do not affect the function of each CGU. The solutions that maximize Eq. (2) or (8) often have significant overlap between the CGUs. Here we describe rules that disqualify solutions in which overlap would interfere with the coarse grained representation. We also describe how an acceptable CGU partitioning is performed in cases where overlap is allowed.

Once a set of CGUs which maximizes the scoring function is found, it is tested for the following criterion: Allowed solutions are those in which each overlapping node receives inputs from only one CGU (Fig. 11). CGU sets which do not meet this criterion are disqualified, and a new set is sought. (Note that the overlapping nodes are allowed to send outputs to both CGUs.) In acceptable CGU sets, in every case of an overlap, the overlapping node is duplicated and appears once in each of the CGU occurrences. The acceptance criterion above ensures that the inputs to the duplicated nodes can be fully captured by one of the CGUs, thus ensuring that the function or dynamics of the coarse-grained network can be inferred from the functions of individual CGUs. Finally, in cases where the overlapping node only sends output to the CGUs and does not receive inputs from them, an additional node is created in the coarse-grained network. This node has all of the connections of the original node and sends outputs to the duplicated node in each CGU [e.g., MKP2 and MKP5 in Figs. 9, 10, and 11(c)].

APPENDIX C: ALGORITHM FOR DETECTING PGNMs

Topological generalizations of a network motif are subgraphs obtained from the network motif by replicating one or more of its roles, together with the connections [22]. The role-replication operation does not increase the number of roles in the resulting generalized subgraph, which maintains a perfect fit to the block model of the network motif. Additionally, each node has the same role in both the topological generalization and in every occurrence of the motif included in it. The role assignment is thus automatically defined. Probabilistically generalized network motifs (PGNM's) are subgraphs which have a small distance d [Eq. (5)] from its block model. To detect PGNMs we start with a network motif μ. The nodes of each occurrence of μ are partitioned into roles [22]. We then form a nondirected graph R_μ in which each node r_μ^i is an occurrence of μ in the original network R and a nondirected edge between two nodes r_μ^i and r_μ^j is set if (a) any of the nodes of these occurrences in the original network R are connected by an edge or (b) any of the nodes in the original network overlap. After establishing R_μ we start from each node r_μ^i and perform a search, consecutively adding the one node in R_μ which provides the best fit to the block model of μ (the resulting joined subgraph with the smallest increase in d). We stop when d is greater than a threshold (we use 0.3). When calculating the fit to the block model, we partition the nodes of the joined subgraph according to their role assignment in μ. If a node in R has different roles in two different occurrences of μ, when calculating d for the joined subgraph, we take the smallest distance obtained from all possible labelings of this node [for example, nodes 3, 4 in subgraph $G2$ of Fig. 6 share role 1 in the bifan (3, 4, 5, 6) and role 2 in the bifan (1, 2, 3, 4)]. We iterate this procedure by beginning with each r_μ, establishing a list of embedded subgraphs (if two embedded structures have the same d we keep only the larger one). These subgraphs are probabilistic generalizations of μ, tagged by their distance from a perfect generalization, d. In finding the optimal coarse-graining we perform a simulated annealing algorithm, sequentially generating a new active set of CGUs, recalculating the scoring function [Eq. (8)] and accepting the new active set with a Metropolis Monte Carlo probability.[4] During the optimization, we also test the resulting score from coarse-graining only subsets of the PGNMs of each CGU. For an alternative definition of probabilistic network motifs see [41].

[4]When establishing the CGU candidate set, we also ignore the connectivity profile of subgraphs. This further softens the CGU criteria. The present scoring function still favors units with few mixed nodes, because they affect the number of ports, P, in the coarse-grained network.

[1] S. H. Strogatz, Nature (London) **410**, 268 (2001); R. Albert and A. L. Barabasi, Rev. Mod. Phys. **74**, 47 (2002).
[2] M. E. J. Newman, SIAM Rev. **45**, 167 (2003).
[3] U. Alon, Science **301**, 1866 (2003).
[4] P. Horowitz and W. Hill, *The Art of Electronics* (Cambridge University Press, Cambridge, England, 1989).
[5] S. Kundu, in *Proceedings of the IEEE International Test Conference* (IEEE CS Press, New York, 1998), p. 372.

[6] M. C. Hansen, H. Yaclin, and J. P. Hayes, IEEE Des. Test **16**(3), 72 (1999).
[7] H. H. McAdams and L. Shapiro, Science **269**, 650 (1995).
[8] L. H. Hartwell *et al.*, Nature (London) **402**, C47 (1999).
[9] P. D'Haeseleer, S. Liang, and R. Somogyi, Bioinformatics **16**, 707 (2000).
[10] K. W. Kohn, Chaos **11**, 84 (2001).
[11] E. Ravasz *et al.*, Science **297**, 1551 (2002).
[12] M. E. Csete and J. C. Doyle, Science **295**, 1664 (2002).
[13] A. W. Rives and T. Galitski, Proc. Natl. Acad. Sci. U.S.A. **100**, 1128 (2003).
[14] J. J. Tyson, K. C. Chen, and B. Novak, Curr. Opin. Cell Biol. **15**, 221 (2003).
[15] T. S. Gardner *et al.*, Science **301**, 102 (2003).
[16] M. A. Beer and S. Tavazoie, Cell **117**, 185 (2004).
[17] S. Kalir and U. Alon, Cell **117**, 713 (2004).
[18] P. Eichenberger *et al.*, PLoS Biol. **2** (10) (2004).
[19] S. Shen-Orr *et al.*, Nat. Genet. **31**, 64 (2002).
[20] R. Milo *et al.*, Science **298**, 824 (2002).
[21] R. Milo *et al.*, Science **303**, 5663 (2004).
[22] N. Kashtan *et al.*, Phys. Rev. E **70**, 031909 (2004).
[23] A. R. Barron, J. Rissanen, and B. Yux, IEEE Trans. Inf. Theory **44**, 2743 (1998).
[24] T. C. Bell, J. G. Cleary, and I. H. Witten, *Text Compression*, Prentice-Hall Advanced Reference Series (Prentice-Hall, Englewood Cliffs, NJ, 1990).
[25] S. Kirkpatrick, C. Gelatt, and M. Vecchi, Science **220**, 671 (1983).
[26] M. Newman and G. Barkema, *Monte Carlo Methods in Statistical Physics* (Oxford University Press, Oxford, 1999).
[27] C. J. Alpert and A. B. Kahng, Integr., VLSI J. **19**, 1 (1995).
[28] M. Girvan and M. E. J. Newman, Proc. Natl. Acad. Sci. U.S.A. **99**, 7821 (2002).
[29] H. Zhou, Phys. Rev. E **67**, 061901 (2003).
[30] H. Zhou and R. Lipowsky, Lect. Notes Comput. Sci. **3038**, 1062 (2004).
[31] F. Brglez, D. Bryan, and K. Kozminski, in *Proceedings of the International Symposium on Circuits and Systems* (IEEE, New York, 1989), p. 1929.
[32] R. F. Cancho, C. Janssen, and R. V. Sole, Phys. Rev. E **64**, 046119 (2001).
[33] http://focus.ti.com/docs/logic/logichomepage.jhtml
[34] H. T. Nagle, B. D. Carrol, and J. D. Irwin, *An Introduction to Computer Logic* (Prentice-Hall, Englewood Cliffs, NJ, 1975).
[35] S. Mangan and U. Alon, Proc. Natl. Acad. Sci. U.S.A. **100**, 11980 (2003).
[36] M. Ronen *et al.*, Proc. Natl. Acad. Sci. U.S.A. **99**, 10 555 (2002).
[37] S. Mangan, A. Zaslaver, and U. Alon, J. Mol. Biol. **334**, 197 (2003).
[38] A. Zaslaver *et al.*, Nat. Genet. **36**, 486 (2004).
[39] G. Lahav *et al.*, Nat. Genet. **36**, 147 (2004).
[40] N. Rosenfeld, M. B. Elowitz, and U. Alon, J. Mol. Biol. **323**, 785 (2002).
[41] J. Berg and M. Lassig, Proc. Natl. Acad. Sci. U.S.A. **101**, 14 689 (2004).
[42] H. C. White, S. A. Boorman, and R. L. Breiger, Am. J. Sociol. **81**, 730 (1976).
[43] S. Wasserman and K. Faust, *Social Network Analysis* (Cambridge University Press, Cambridge, England, 1994).
[44] W. H. Panning, Soc. Networks **4**, 81 (1982).
[45] R. Dobrin, Q. K. Beg, and A. L. Barabasi, BMC Bioinf. **5**, 10 (2004).
[46] C. Y. Huang and J. E. Ferrell, Jr., Proc. Natl. Acad. Sci. U.S.A. **93**, 10 078 (1996).
[47] U. S. Bhalla and R. Iyengar, Science **283**, 381 (1999).
[48] S. J. Charette *et al.*, Mol. Cell. Biol. **20**, 7602 (2000).
[49] D. Bray, Nature (London) **376**, 307 (1995).
[50] G. Pearson *et al.*, Endocr. Rev. **22**, 153 (2001).
[51] H. Ichijo *et al.*, Science **275**, 5296 (1997).
[52] W. Wang *et al.* J. Biol. Chem. **272**, 36 (1997).
[53] J. M. Kyriakis and J. Avruch, Physiol. Rev. **81**, 807 (2001).
[54] stke.sciencemag.org
[55] D. H. Wolpert and W. G. Macready, in *Unifying Themes in Complex Systems*, edited by Y. Bar-Yam (Westview, Boulder, CO, 2000), p. 626.
[56] J. M. Carlson and J. Doyle, Proc. Natl. Acad. Sci. U.S.A. **99**, 2538 (2002).
[57] R. Milo *et al.*, e-print cond-mat/0312028; K. Sneppen and S. Maslov, Science **296**, 910 (2002).
[58] S. Itzkovitz *et al.*, Phys. Rev. E **68**, 026127 (2003).