Topological generalizations of network motifs

N. Kashtan,[1,2] S. Itzkovitz,[1,3] R. Milo,[1,3] and U. Alon[1,3]
[1]*Department of Molecular Cell Biology, Weizmann Institute of Science, Rehovot, Israel 76100*
[2]*Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot, Israel 76100*
[3]*Department of Physics of Complex Systems, Weizmann Institute of Science, Rehovot, Israel 76100*
(Received 8 March 2004; published 23 September 2004)

Biological and technological networks contain patterns, termed network motifs, which occur far more often than in randomized networks. Network motifs were suggested to be elementary building blocks that carry out key functions in the network. It is of interest to understand how network motifs combine to form larger structures. To address this, we present a systematic approach to define "motif generalizations": families of motifs of different sizes that share a common architectural theme. To define motif generalizations, we first define "roles" in a subgraph according to structural equivalence. For example, the feedforward loop triad—a motif in transcription, neuronal, and some electronic networks—has three roles: an input node, an output node, and an internal node. The roles are used to define possible generalizations of the motif. The feedforward loop can have three simple generalizations, based on replicating each of the three roles and their connections. We present algorithms for efficiently detecting motif generalizations. We find that the transcription networks of bacteria and yeast display only one of the three generalizations, the multi-output feedforward generalization. In contrast, the neuronal network of *C. elegans* mainly displays the multi-input generalization. Forward-logic electronic circuits display a multi-input, multi-output hybrid. Thus, networks which share a common motif can have very different generalizations of that motif. Using mathematical modeling, we describe the information processing functions of the different motif generalizations in transcription, neuronal, and electronic networks.

DOI: 10.1103/PhysRevE.70.031909 PACS number(s): 87.10.+e, 89.75.−k

I. INTRODUCTION

A major current challenge is to understand the function of biological information-processing networks [1–13]. These networks, as well as networks from engineering, ecology, and other fields, were recently found to contain *network motifs*: small subgraphs that occur in the network far more often than in randomized networks [15,14]. Each class of networks was found to have a characteristic set of network motifs [16]. Information-processing networks, such as gene regulation networks [15,17], neuron networks, and some electronic circuits, were found to share many of the same network motifs [14,16]. Recently, in the case of the transcription network of the bacterium *E. coli*, network motifs were shown theoretically and experimentally to function as elementary building blocks of the network, each performing specific information-processing tasks [15,18,19]. For example, one of the most significant motifs shared by biological information processing networks is the feedforward loop (FFL). In transcription networks, the feedforward loop with positive regulations was shown to act as a "persistence detector" circuit that rejects transient activation signals yet allows rapid response to inactivation signals [15,18,19]. A second motif, the single-input module, was shown to generate a temporal order of gene expression, which correlates with the functional order of the genes in the pathway [15,21,22]. A third major motif, the bifan, which is the building block of dense arrays of overlapping regulation, performs hard-wired combinatorial decisions governed by the input functions of the output genes [23–25].

Network motifs can, in some cases, also be used as building blocks of a coarse-grained version of the network [54]. Here, we address the question of whether a given network motif appears independently in the network or whether instances of the motif combine to form larger structures [15,20,55,56]. If the latter occurs, what is the function of these larger structures? Do different networks that share a certain network motif also share the same structural combinations of that motif? These questions require analysis of large subgraphs, a computationally difficult problem [26–29]. Recently, efficient algorithms for counting subgraphs based on sampling have been introduced [27]. These algorithms can at present be effectively used to detect motifs of up to six to seven nodes. To go beyond this requires an approach to efficiently define and detect large structures whose architecture is based on a given motif.

To address these issues, we present an approach for uniting related groups of motifs of different sizes into families termed *motif generalizations*. This allows generalizing from small motifs to the larger complexes in which they appear. We present an efficient algorithm to detect motif generalizations. We find that networks that share the same motif can have different generalizations of that motif. For example, we find different generalizations of the FFL motif in transcription, neuronal and electronic networks. Using mathematical models we analyze the information-processing functions of the FFL generalization that is selected in each of these networks.

II. RESULTS

A. Node roles in a subgraph

We begin by defining *roles* of nodes in a subgraph. A group of nodes in a subgraph share the same role if there is a permutation of these nodes, together with their correspond-

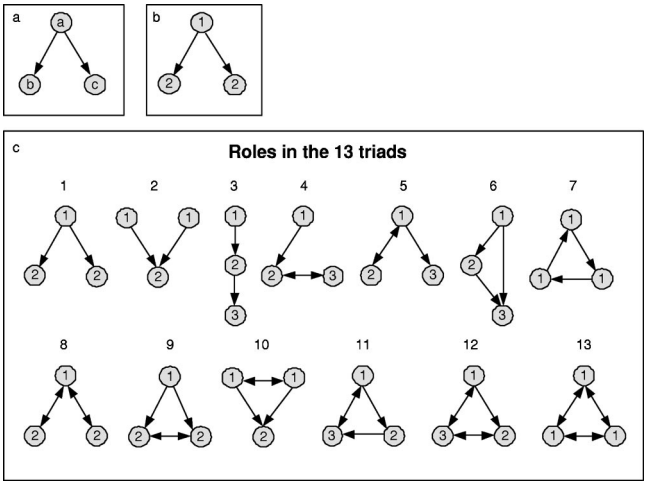

FIG. 1. (a) A directed three-node subgraph. (b) This triad has two roles. (c) Roles in all 13 types of connected triads. In each triad there are between one and three roles.

ing edges, that preserves the subgraph structure (see Appendix A for formal definitions). For example, in the *V*-shaped subgraph in Fig. 1(a), nodes (b) and (c) can be permuted, leaving the structure intact, whereas nodes (a) and (b) cannot. Thus, this subgraph has two roles, role 1 and role 2 [Fig. 1(b)]. The FFL has three roles [Fig. 1(c), triad 6], whereas the three-loop [Fig. 1(c), triad 7] has only one role (because a cyclic permutation of the three nodes preserves its structure). The 13 possible connected directed triads have between one and three roles each [Fig. 1(c)].

B. Subgraph topological generalizations

We now define subgraph topological generalizations based on node roles. Subgraph topological generalizations are extensions of a subgraph to a family of larger subgraphs which share its basic structure. Consider the FFL [Fig. 2(a)]. For this three-node subgraph we define three simple generalizations to the level of four nodes [Fig. 2(b)]. In each simple generalization a single role and its connections are duplicated. In the first simple generalization, the *X* role and its connections are duplicated. This generalization is termed double-*X* FFL or double-input FFL. The other two generalizations are obtained by duplicating the *Y* or *Z* roles. This replication process can be continued, leading to higher-order motif generalizations, the multi-*X* (multi-input), multi-*Y*, and multi-*Z* (multi-output) FFL generalizations [Fig. 2(c)].

More complex generalizations can be obtained by replicating more than one of the roles. For example, duplicating both the *X* and *Z* roles yields five-node generalizations [Fig. 2(d)]. When replicating more than one role (and in some cases replicating even a single role), one can define two kinds of generalizations: in strong generalizations, every X,Y,Z triplet forms a FFL. In weak generalizations, every node participates in at least one FFL, but not all possible FFLs are formed [Fig. 2(d)].

This procedure of generalization can be applied to any subgraph (see formal definition in Appendix B). For example simple generalizations of the four-node bi-fan are shown in

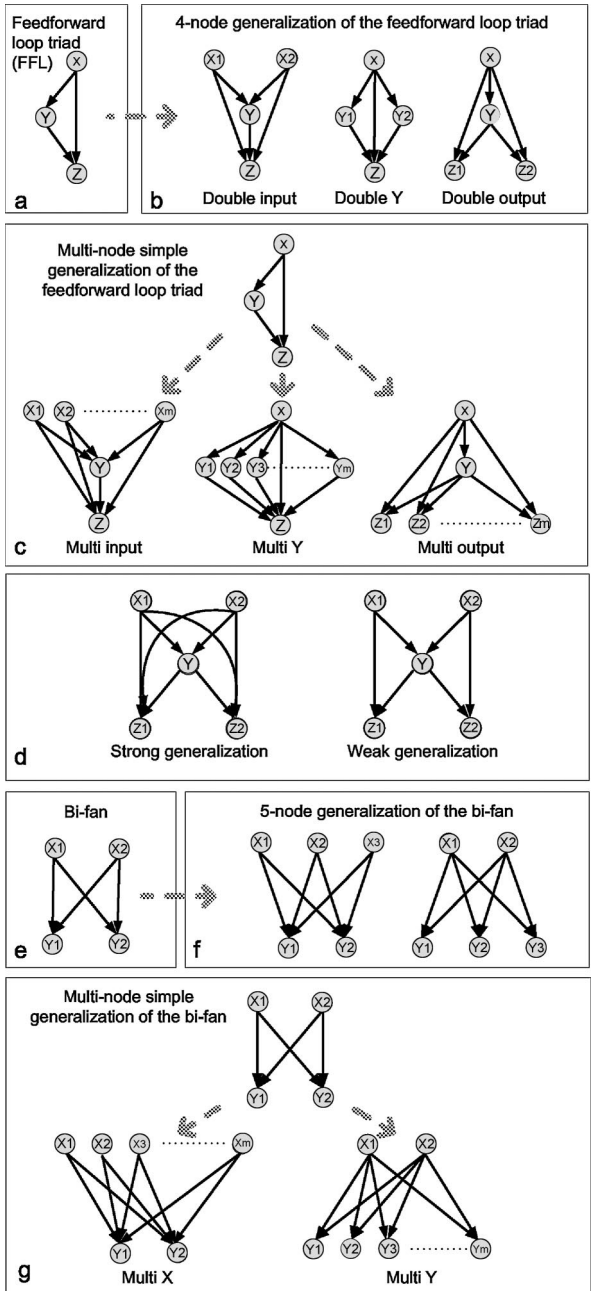

FIG. 2. (a) The feedforward loop triad has three roles: *X* (input node), *Y* (internal, secondary input) node, and *Z* (output node). (b) Four-node simple generalizations of the feedforward loop. The *X* node is duplicated to form the double-*X* generalization. The *Y* and *Z* nodes are duplicated to form the double-*Y* and double-*Z* generalizations, respectively. (c) Simple multi-node generalizations of the FFL. (d) Strong and weak generalization rules. A five-node generalization of the FFL with two *X* nodes, one *Y* node, and two *Z* nodes. In the strong generalization every combination of a X,Y,Z triplet of nodes forms a FFL. (e) The bi-fan, a four-node motif with two roles *X* (input role) and *Y* (output role). (f) Five-node simple generalizations of the bi-fan. In each of the two generalizations one of the two roles is duplicated. (g) Simple multi-node generalization of the bi-fan: an *X* or *Y* node is replicated to form the multi-input or multi-output bi-fan generalization, respectively.

Figs. 2(e)–2(g). We now describe the statistical significance of the generalizations of the motifs found in various networks.

C. Network motif topological generalizations

While enumerating all subgraphs of a given size is a difficult task, enumerating generalizations of a given subgraph can be performed efficiently by an algorithm described in Appendix C. The algorithm is based on using the appearances of the basic subgraph as nucleation points for a search for its generalizations. As an example, we applied this algorithm to networks in which the FFL and bi-fan are motifs, to ask whether any of the possible FFL or bi-fan generalizations occur significantly in the networks (Appendix C). In the transcription networks of *E. coli* [15] and *S. cerevisiae* [14] we find that the multi-*Z* FFL generalization is highly significant [Figs. 3(a) and 3(b)]. The other two possible simple generalizations are not significant (in the *E. coli* network, multi-*X*'s and multi-*Y*'s do not occur at all, in the *S. cerevisiae* network both appear only twice). An example of a multi-*Z* FFL in the *E. coli* transcription network, the maltose utilization system, is shown in Fig. 4(a). In each multi-*Z* FFL, the different genes (*Z* roles) share a common biological function (as shown in Tables II and III which list all multi-*Z* FFL complexes in the *E. coli* and *S. cerevisiae* networks).

In the network of synaptic connections between neurons in *C. elegans* [14,30,31], we find a different FFL generalization: the multi-*X* FFL [Fig. 3(c)]. This structure occurs 29 times in the network, with up to four inputs. Multi-*Y* and multi-*Z* FFL's are found in far smaller numbers (double-*X* and double-*Y* FFLs appear 3 times each) [32]. An example of a multi-*X* FFL in the locomotion control circuit of *C. elegans* is shown in Fig. 4(b).

In networks of connections between logic gates in forward-logic electronic chips [14,33,34] we find no simple generalization of the FFL. These electronic circuits do, however, show a complex FFL generalization—a structure with two *X*'s, a single *Y*, and two *Z*'s [a weak generalization, Fig. 4(c)]. In the five forward-logic electronic chips we have analyzed, 70%–100% of the FFLs are embedded in instances of this five-node structure.

The most prominent four-node network motif in these networks is the bi-fan [14] [Fig. 2(e)]. The bi-fan has two roles and therefore two simple generalizations [Fig. 2(g)]. We find that both simple generalizations of the bi-fan (multi-output and multi-input) are significant in the transcription, neuronal, and electronic networks (Table I). The multi-output bi-fan generalizations are more significant and the maximal *Y* multiplicity is higher than the maximal *X* multiplicity in all these networks. In these networks we find structures of multi-output bi-fan with ten *Y*'s and more, while multi-input bi-fans do not exceed six input *X* nodes.

D. Functions of multi-output FFL generalization in transcription networks

The function of the FFL depends on the signs of the interactions (positive or negative regulation), on their strengths, and on the functions that integrate multiple inputs into each node. In the case of positive regulation and AND-logic, the three-node FFL has been shown to function as a persistence detector [15]: it filters out short input stimuli to *X* and responds only to persistent signals. On the other hand, it responds quickly to OFF steps in the input to *X* [15,18]. With an OR-gate the FFL filters OFF pulses and reponds rapidly to ON pulses [18]. With other sign combinations, the three-node FFL can function as a pulse generator or response accelerator [18,35]. These functions apply to a wide range of interaction strengths and to both AND and OR-like input functions.

Here, we study the functions of the generalizations of the FFL. We begin with the multi-output FFL, which is the generalization that is significant in transcription networks. The multi-output FFL has a single input node *X*, a single internal node *Y* (secondary input), and a number of output nodes $Z_1, \ldots, Z_m$ [Figs. 2(c) and 4(a)]. The arrows in the FFL diagram should be assigned numbers representing the strength of the interaction of the transcription factors (TF's) *X* and *Y* with the promoters of the various *Z* genes [21]. These numbers correspond to the activation or repression coefficients of each gene (the concentration of the TF required for 50% effect [5,21,36]). Here, we consider for simplicity the most common case, that of FFLs with positive regulation and AND-logic [18]. We employ a simple model of the dynamics of this circuit [15]. $X(t)$ is the activity of the transcription factor *X*, $Y(t)$ of *Y*, and $Z_j(t)$ is the concentration of the gene product Z_j. The dynamics of transcription factor *Y* and the output gene products Z_j is given by

$$dY/dt = F(X, T_{yx}) - \alpha Y,$$

$$dZ_j/dt = F(X, T_{z_jx})F(Y, T_{z_jy}) - \alpha Z_j,$$

where α is the protein lifetime [37,38] and T_{yx}, T_{z_1x}, T_{z_2x}, T_{z_1y}, and T_{z_2y} are the activation thresholds of the various genes [Fig. 5(a)]. For simplicity we use a sharp activation function, $F(U,T)=1$ if $U>T$ and 0 otherwise. The qualitative results apply also to Michaelis-type activation functions. These equations can be solved analytically, yielding piecewise exponential dynamics in response to steplike activation profiles of *X*. We find that the multi-output FFL can encode a temporal order of expression of the *Z* genes, by means of different activation thresholds T_{z_jy} for each of the output genes [Figs. 5(a) and 5(b)]. This temporal ordering feature is shared with another common network motif, the single-input module [15,21,22]. Indeed, high-resolution expression measurements on the flagella multi-output FFL (in *E. coli*) showed that the class-2 flagella genes, which are regulated by a feedforward loop, are activated in a temporal order that corresponds to the functional order of the gene product in the assembly of the flagellar motor [39,40].

The timing of activation of gene *j* following a step activation of *X* is

$$\tau_j = -\alpha^{-1}\ln(1 - T_{z_jy}/Y_{max}).$$

The rise time of the different genes can be tuned by T_{z_jy}/Y_{max}, where Y_{max} is the maximal concentration of *Y*. Note that T_{z_jy} can be easily tuned during evolution—for ex-

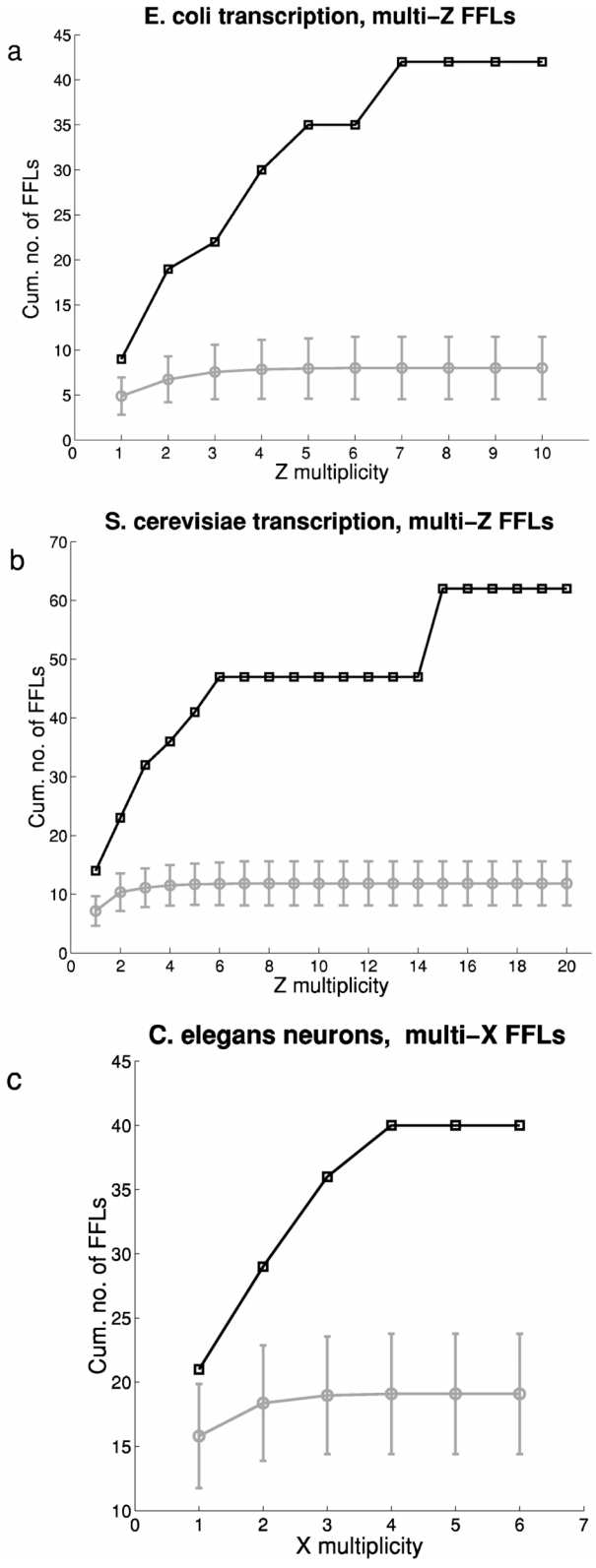

FIG. 3. Statistical significance of motif generalizations. The cumulative number of multi-*Z* FFLs in the real network (black) and randomized networks-mean ± SD (gray) in (a) *E. coli* transcription network. (b) *S. cerevisiae* transcription network. (c) The cumulative number of multi-*X* FFL's in the real and randomized networks (mean ± SD) in the *C. elegans* neuronal network.

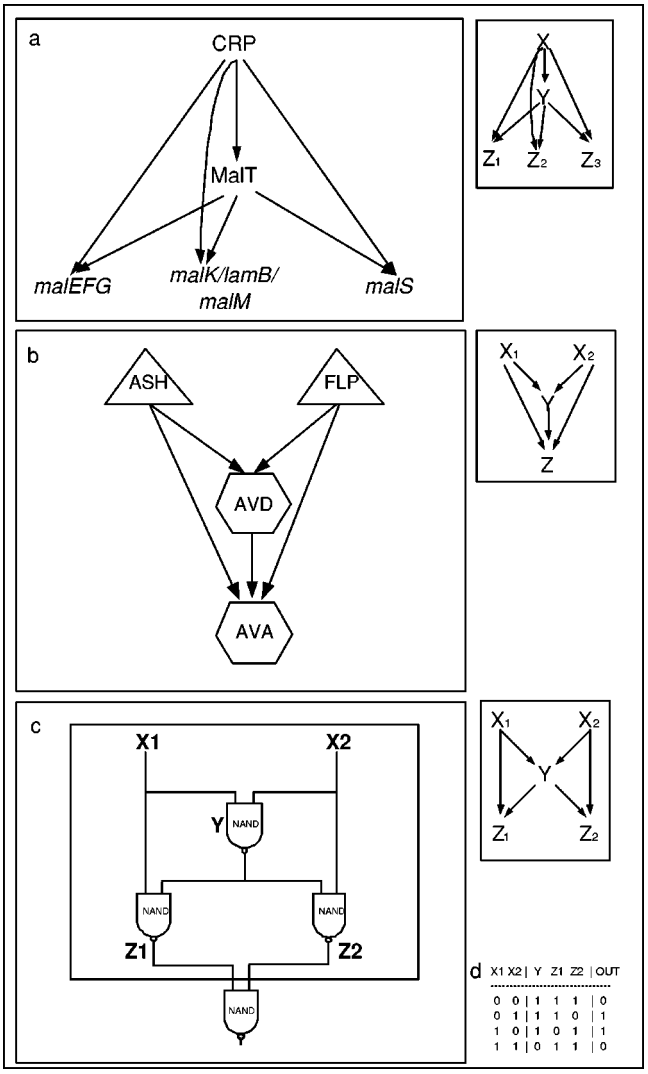

FIG. 4. The FFL generalizations found in biological and technological networks. (a) An example of a three-*Z* FFL in the transcription network of *E. coli*, maltose utilization system. The activator CRP senses glucose starvation, MalT senses maltotriose, and malEFG, malK, and malS participate in maltose metabolism and transport. (b) An example of a double-*X* FFL in the locomotion neuronal circuit of *C. elegans*. AVA and AVD are ventral cord command interneurons. AVD functions as modulator for backward locomotion. AVA functions as driver cell for backward locomotion. ASH and FLP are head sensory neurons sensitive to noxious chemicals and nose touch. (c) A generalized form of the FFL $(2X,Y,2Z)$ found in forward-logic electronic chips. This five-node structure appears as a part of a six-node module, which implements XOR (exclusive OR) using four NAND gates. (d) Truth table of the circuit described in (c) [a $(2X,Y,2Z)$ FFL generalization with additional NAND gate at the output]. There are two input bits $X1$ and $X2$ and a single output bit which is equal to ($X1$ XOR $X2$).

ample, by mutations in the binding site of Y in the Z_j promoters [25,40]. The Z gene with the lowest activation threshold is turned on first after the stimulation of X.

In addition to generating temporal order, the multi-*Z* FFL can act as a persistence detector for all of its output genes [Fig. 5(b)]: the Z genes are expressed only if the input stimu-

TABLE I. Bi-fan generalizations in different networks. (aX,bY) represents the multiplicity of each of the roles in the generalization [Fig. 2(g)]. "+": statistically significant generalizations. "−": nonsignificant generalizations. Number of appearances (N) or concentration ($\times 10^{-3}$) (C) [27] are listed.

		Transcriptional Networks		Neurons	Electronic chips
Generalization	Subgraph size	*E. coli*	Yeast	*C. elegans*	S15850
Basic bi-fan	$4(2X,2Y)$	$+(N=209)$	$+(N=1812)$	$+(N=126)$	$+(N=1040)$
Multi-output	$5(2X,3Y)$	$+(N=264)$	$+(N=14857)$	$+(N=152)$	$+(N=1990)$
	$6(2X,4Y)$	$+(C=0.015)$	$+(C=3.5)$	$+(C=0.17)$	$+(C=0.28)$
Multi-input	$5(3X,2Y)$	$+(N=20)$	$+(N=81)$	$+(N=25)$	$+(N=226)$
	$6(4X,2Y)$	$-(N=0)$	$+(N=14)$	$+(C=0.015)$	$+(C=0.001)$
Equal multi-input and -output	$6(3X,3Y)$	$+(N=6)$	$+(N=21)$	$-(N=0)$	$+(N=301)$

lus to X is present for a long enough time. The minimal time that a saturating X stimulus needs to be present to activate gene j is equal to τ_j. Thus this FFL generalization preserves the functionality of the original FFL motif.

The turn-off order of the Z genes upon a gradual decay of X activity can be separately controlled by the activation coefficients of the X TF, T_{z_jx} [40]. Thus different turn-on and turn-off orders of the Z_j genes can in principle be achieved. In summary, the multi-output FFL preserves the functionality of the simple FFL and in addition can encode temporal expression programs among the different Z genes.

E. Functions of multi-input FFL generalization in neuronal networks

A different FFL generalization, multi-input FFL, is found in the neuronal network of *C. elegans*. In general, the function of this circuit depends on the signs on the arrows and on two input functions (gates): one input function integrates the multiple X inputs to Y, and the other integrates the inputs from Y and $X_1,\ldots,X_m$ to Z [Fig. 6(a)].

We analyzed the dynamics of one possible two-input FFL, where the input function governing the Y node is an OR gate, X_1 OR X_2, and the input function of the Z node is Y AND (X_1 OR X_2) [Figs. 6(a)–6(c)]. This choice of input functions ensures that both Y and either X_1 or X_2 are needed for Z to be activated to a level that allows activation of its downstream (post synaptic) neurons or muscle cells [as is the case, for example, in the circuit of Fig. 4(b), in which ablation of the neuron AVD results in loss of sensory input to the neuron AVA [41]]. These input functions could in principle be implemented by simple neurons which integrate weighted inputs. The input function of Z, for example, represents strong synapses from Y and weaker ones from X_1 and X_2.

It is important to note that the simplest equations that describe transcription networks also describe neurons with graded potential and no spiking (as *C. elegans* neurons are thought to be [42,43]). In the case of neurons, $X_i(t)$, $Y(t)$, and $Z(t)$ represent neuron membrane potentials. The activation dynamics of the circuit in Fig. 6(a) are

$$dY/dt = F(X_1 + X_2, T_{yx}) - \alpha Y,$$

$$dZ/dt = F(Y, T_{zy})F(X_1 + X_2, T_{zx}) - \alpha Z.$$

Here α is the relaxation rate of the neurons' membrane potential, and the synaptic activation thresholds are T_{yx}, T_{zx}, and T_{zy}.

This model shows that the circuit can act as a persistence detector for both X_1 and X_2 [Fig. 6(b)]. In the locomotion neuronal circuit example [Fig. 4(b)], the FFL circuit could elicit backward motion only if the stimulation of one of the sensory neurons is longer than a threshold duration τ determined by the parameters of the circuit:

$$\tau = -\alpha^{-1}\ln(1 - T_{zy}/Y_{max}).$$

A transient stimulation would not be enough to elicit backward motion. Furthermore, we find that sufficiently closely spaced short pulses of X_1 and X_2 can elicit a response, *even if each pulse alone cannot* [Fig. 6(c)]. This highlights a "memory" function of Y, which can store information from recent stimulations over its relaxation time. In the basic three-node FFL, Y can store information about recurring pulses of X. In the multi-input FFL, Y can store information from multiple inputs [Fig. 6(c) gives an example] and increase sensitivity to one input if the other input has recently been detected. Generally, if the summed input of the input nodes X_j to node Y is $S(t)=F(X_1+X_2,T_{yx})$, Z is activated when Y activity exceeds the threshold T_{zy}:

$$Y(t) = \int_0^t S(t')e^{-\alpha(t-t')}dt' > T_{zy},$$

where $Y(t=0)=0$, showing that node Y effectively integrates the inputs over a time scale of $1/\alpha$.

F. Function of FFL generalization in electronic chips

Forward-logic electronic chips are networks in which nodes represent logic gates. These circuits are optimized to perform a hard-wired logical function between input and output nodes. Forward-logic chips, taken from an engineering database (ISCAS89), were previously found to display the FFL network motif [14]. Here we find that they display a specific generalization of the FFL, with two input and two output nodes [Fig. 4(c)]. Analyzing the appearances of this pattern, we find that this five-node generalized FFL motif is part of a commonly used module built of four NAND gates, which implements XOR (exclusive OR) logic on the two inputs [44] [see truth table in Fig. 4(d)].

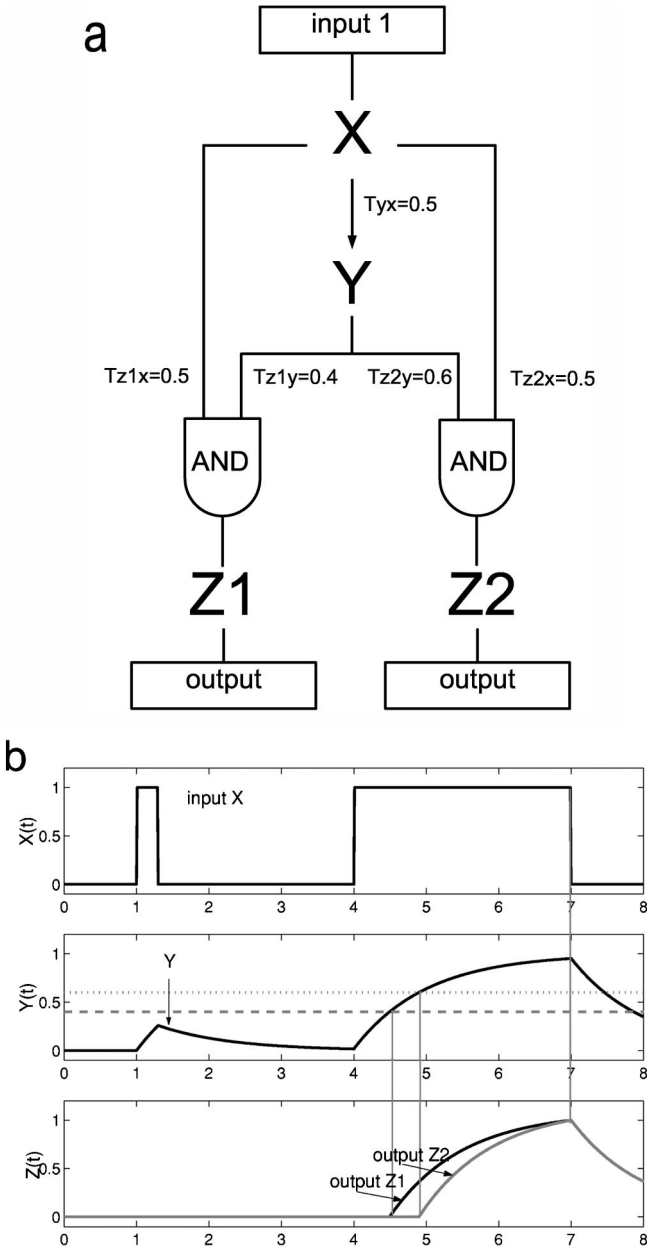

FIG. 5. Kinetics of a double-output FFL generalization following pulses of stimuli. (a) A double-output FFL with positive regulation and AND-logic input function for Z_1 and Z_2. Numbers on the arrows are activation thresholds. (b) Simulated kinetics of the double-output FFL in response to a short pulse and a long pulse of X activity. The dashed and dotted horizontal lines represent the activation thresholds T_{z_1y} and T_{z_2y}. $\alpha=1$ was used.

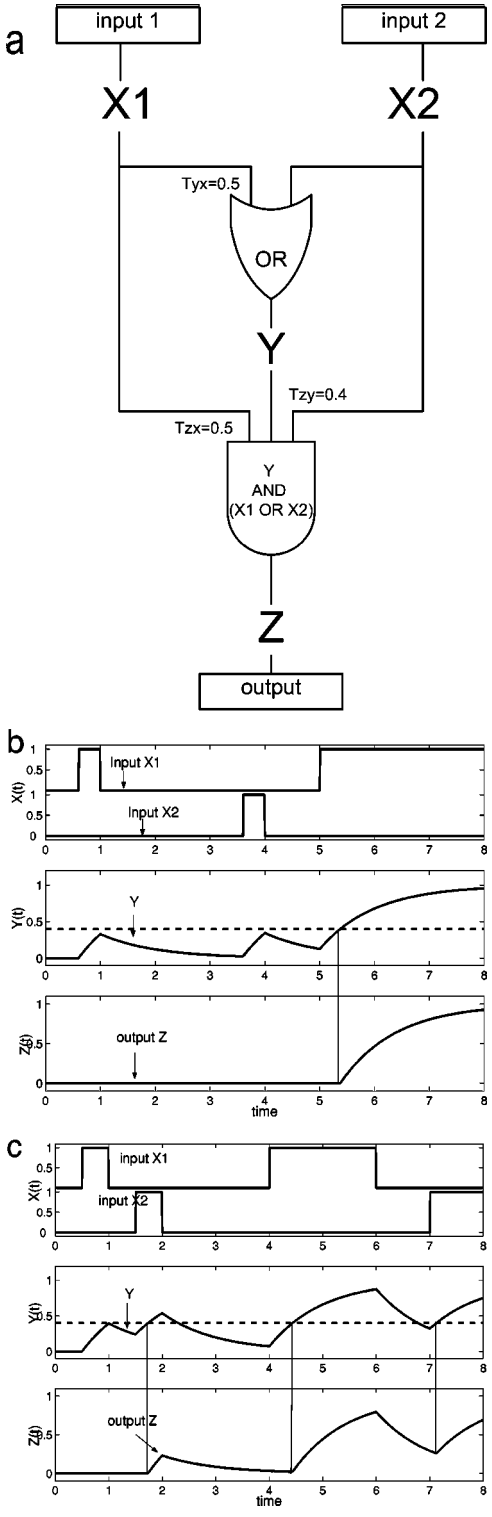

FIG. 6. Kinetics of a double-input FFL generalization following pulses of stimuli. (a) A double-input FFL. Input functions for Y and Z, and the activation thresholds, are shown as gates and numbers on the arrows. (b) Simulated kinetics of the two-input FFL, with short well-separated stimuli pulses of X_1 and X_2, followed by a persistent X_1 stimulus. (c) Simulated kinetics of the double-input FFL, with short X_1 stimulus followed rapidly by a short X_2 stimulus pulse. The dashed horizontal line corresponds to the activation threshold for Y, T_{zy}. $\alpha=1$ was used.

III. DISCUSSION

This study presented a systematic approach for defining and detecting topological generalizations of network motifs. Motif generalizations are families of subgraphs of different sizes which share a common structural theme and which appear significantly more often in the network than in randomized networks. The generalizations are produced by replicating nodes in a basic motif structure. The generalizations often preserve the functionality of the network motif on

which they are based, because they preserve the roles of nodes in the motif (for example, by replicating input or output nodes). We presented an efficient algorithm for detecting motif generalizations. We find that different networks which display the same motifs can show very different generalizations of these motifs. We also demonstrated using simple models that these generalized motifs can carry out specific information processing functions. These functions can in principle be tested experimentally in transcription and neuronal systems.

The two sensory transcription networks, from a prokaryote (*E. coli*) and a eukaryote (*S. cerevisiae*), showed the same generalization of the FFL: both networks display the multi-output FFL generalization [15,20]. The other two generalizations, multi-input and multi-*Y*, are not found significantly in these transcription networks. Multi-output FFL complexes are found throughout the transcription networks in diverse systems (Tables II and III). The *X* role is usually a global transcription factor which controls many genes, the *Y* role is usually a "local" transcription factor which controls specific gene systems, and the *Z* nodes are the regulated genes which share a specific function. Often, multi-output FFL's in *E. coli* that respond to specific stimuli have a nonhomologous multi-output FFL counterpart in yeast which responds to similar stimuli. The fact that the genes in these circuits are not evolutionary related, whereas their connectivity patterns are the same in the two organisms, suggests convergent evolution to the same regulation pattern [14,45]. Examples include systems that respond to carbon limitation, drugs, and nitrogen starvation in both organisms (Tables II and III). Multi-output FFL's can also appear in systems that make up a protein machine; for example, a multi-output FFL in *E. coli* controls genes whose products make up the flagellar basal-body motor [39] (X=flhDC, Y=fliA, Z=class-2 flagella genes). We find that the multi-output FFL can serve as a persistence detector for all the outputs. In addition it can generate temporal orders of output gene expression [40].

A different FFL generalization, the multi-input FFL, is found in the neuronal synaptic wiring of *C. elegans*. This network is found to chiefly display the multi-input FFL [Fig. 2(c)]. The multi-input FFL has a number of input nodes $X_1,\ldots,X_m$, a single internal node Y (secondary input) and a single output node Z. As an example we have mentioned the backward locomotion control circuit of the worm. This circuit is governed by two ventral-cord command interneurons AVD and AVA [41,42]. These two neurons are linked in a multi-input FFL with several input neurons, such as ASH and FLP [Fig. 4(b)], which are head sensory neurons sensitive to nose touch and noxious chemicals [41,42]. This circuit implements an avoidance reflex, eliciting backward motion in response to head stimulation. We find that the multi-input FFL can serve as a persistence detector for each input. In addition, it can serve as coincidence detector for weak inputs, firing only if short stimuli from two or more different inputs occur within a certain time of each other.

A different FFL generalization, with two inputs and two outputs, appears in a class of electronic circuits. This motif generalization functions within a XOR gate. This demonstrates that network motifs and their generalizations can be used to detect basic functional building blocks of a network without prior knowledge [54].

Motif generalizations cover a substantial portion of the high-order motifs in various biological and technological networks we have studied. However, motifs generalizations in the present form do not cover all possible types of families of structures that share similar architectural themes. It would be important to find additional rules for defining families of motifs beyond the current notion of motif generalization by role replication. Motifs and their generalizations can help us understand the design principles of complex networks by defining functional building blocks whose function can be tested experimentally.

To summarize, this study presented topological generalizations of network motifs and an efficient algorithm to detect them. We found motif generalizations in several real-world networks. Networks that share the same motif were found to exhibit different generalizations of that motif. The generalized motifs in biological networks were demonstrated theoretically to carry out information-processing functions.

ACKNOWLEDGMENTS

We thank all members of our laboratory for discussions. We thank NIH, Israel Science Foundation, and Minerva for support. N.K. was supported by Ernst and Anni Deutsch-Promotor Stiftung Foundation. R.M. was supported by Horowitz Complexity Science Foundation.

APPENDIX A: ROLES IN A SUBGRAPH—A FORMAL DEFINITION

We classify nodes in a subgraph into structurally equivalent classes. Each class represents a role. The measure of structural equivalence that we use here is automorphic equivalence [46–50]. Let $S=(V_s,E_s)$ be a subgraph. An automorphism is a one-to-one mapping, τ, from V_s to V_s, such that $(v_i,v_j)\in E_s$ if and only if $(\tau(v_i),\tau(vj))\in E_s$. Two nodes v_i and v_j are automorphically equivalent if and only if there is some automorphism τ that maps one of the nodes to the other $[\tau(v_i)=v_j]$. For each subgraph S, we classify its n nodes into roles by examining structural equivalence of all possible pairs of the nodes. By the transitivity of automorphic equivalence, one is guaranteed to partition the nodes into distinct roles. This concept can be readily generalized for networks with weights on the edges or with different types of nodes.

APPENDIX B: SUBGRAPH GENERALIZATION—A FORMAL DEFINITION

Let S be the basic subgraph where $r_1,\ldots,r_L$ are the set of roles of S with multiplicity $(d_1,\ldots,d_L)$, respectively. A simple generalization of S is a subgraph which is formed by replication of a single role r_i and its edges to preserve the role connectivity of S. Note that in a simple generalization only a single role is replicated. A generalized form of a subgraph is defined by a pair (M,V^L) where M is an $L\times L$ image matrix, which describes the connectivity between roles. $M[i,j]=1$ if there is an edge between role i and j (i is not equal to j) and $M[i,j]=0$ otherwise. $M[i,i]=0$ if there is no

TABLE II. Feedforward loops in the *E. coli* transcription network [15] classified into multi-*Z* complexes. Complex size is the number of operons (*Z*-role nodes) in the FFL generalization.

Complex size	Id.	*X*	*Y*	*Z*	Function
1	1	arcA	appY	appCBA	Anaerobic/stationary phase
	2	crp	fucPIKUR	fucAO	Fucose utilization
	3	crp	fur	cirA	Iron citrate uptake
	4	crp	galS	mglBAC	Carbon utilization
	5	crp	malI	malXY	Maltose utilization
	6	crp	melR	melAB	Melibiose utilization
	7	hns	flhDC	fliAZY	Flagella regulation
	8	metJ	metR	metA	Methionine biosynthesis
	9	ompR-envZ	csgDEFG	csgBA	Osmotic stress response
2	10	crp	caiF	caiTABCDE	Carnitine metabolism
				fixABCX	
	11	crp	nagBACD	manXYZ	Carbon utilization
				nagE	
	12	himA	ompR-envZ	ompC	Osmotic stress response
				ompF	
	13	rpoN	fhlA	fdhF	Formate hydrogen lyase system
				hycABCDEFGH	
	14	rpoN	glnALG	glnHPQ	Nitrogen utilization
				nac	
3	15	crp	malT	malEFG	Maltose utilization
				malK-lamB-malM	
				malS	
4	16	crp	araC	araBAD	Arabinose utilization
				araE	
				araFGH	
				araJ	
	17	rob	marRAB	fumC	Drug resistance
				nfo	
				sodA	
				zwf	
5	18	flhDC	fliAZY	flgBCDEFGHIJK	Flagella system
				flhBAE	
				fliE	
				fliFGHIJK	
				fliLMNOPQR	
7	19	fnr	arcA	cydAB	Anaerobic metabolism
				cyoABCDE	
				focA-pflB	
				glpACB	
				icdA	
				nuoABCDEFGHIJKLMN	
				sdhCDAB-b0725-sucABCD	

TABLE III. Feedforward loops in the *S. cerevisiae* transcription network [14] classified into multi-*Z* complexes. Complex size is the number of genes (*Z*-role nodes) in the FFL generalization.

Complex size	Id.	*X*	*Y*	*Z*	Function
1	1	TUP1	RME1	IME1	Meiosis
	2	RIM101	IME1	DIT1	Sporulation
	3	MIG1	HAP2-3-4-5	CYC1	Formation of apocytochromes
	4	MIG1	GAL4	GAL1	Galactokinase
	5	MIG1	CAT8	JEN1	Lactate uptake
	6	MIG2	CAT8	JEN1	Lactate uptake
	7	GAT1	DAL80-GZF3	GAP1	Nitrogen utilization
	8	TUP1	ALPHA1	MFALPHA1	Mating
	9	GAL11	ALPHA1	MFALPHA1	Mating
2	10	TUP1	ROX1	ANB1	Anaerobic metabolism
				CYC7	
	11	GLN3	GAT1	GAP1	Nitrogen utilization
				GLN1	Glutamate synthetase
	12	GLN3	GAT1	DAL80	Nitrogen utilization
				GLN1	Glutamate synthetase
	13	GLN3	DAL80-GZF3	GAP1	Nitrogen utilization
				UGA4	
	14	PDR1	YRR1	SNQ2	Drug resistance
				YOR1	
	15	GCN4	MET4	MET16	Methionine biosynthesis
				MET17	
3	16	HAP1	ROX1	ERG11	Anaerobic metabolism
				HEM13	
				CYC7	
	17	SPT16	SWI4-SWI6	CLN1	Cell cycle and
				CLN2	mating type switch
				HO	
4	18	GCN4	LEU3	ILV1	Leucine and branched amino
				ILV2	acid biosynthesis
				ILV5	
				LEU4	
	19	UME6	INO2-INO4	CHO1	Phospholipid biosynthesis
				CHO2	
				INO1	
				OPI3	
6	20	PDR1	PDR3	HXT11	Drug resistance
				HXT9	
				IPT1	
				PDR5	
				SNQ2	
				YOR1	

TABLE III. (*Continued.*)

Complex size	Id.	*X*	*Y*	*Z*	Function
15	21	GLN3	DAL80	CAN1	Nitrogen utilization
				DAL1	
				DAL2	
				DAL3	
				DAL4	
				DAL5	
				DAL7	
				DCG1	
				DUR1	
				DUR3	
				GDH1	
				PUT1	
				PUT2	
				PUT4	
				UGA1	

edge between every two nodes of role i, $M[i,i]=1$ if there is a single edge, and $M[i,i]=2$ if there is a mutual edge. $V^L \in N^L$ is an L-dimensional vector which defines the multiplicity of each role. The FFL which is an example of a basic subgraph, is represented by $\mathbf{(}\boldsymbol{M}_{FFL},(1,1,1)\mathbf{)}$ where

$$\mathbf{M}_{FFL}=\begin{pmatrix}0 & 1 & 1\\ 0 & 0 & 1\\ 0 & 0 & 0\end{pmatrix}$$

and the vector $(1,1,1)$ describes the role multiplicity: in the basic FFL each of the three roles X,Y,Z appears once. A FFL with two output nodes is represented by the pair $\mathbf{(}M_{FFL},(1,1,2)\mathbf{)}$. A FFL with m output nodes (m Z-role nodes) is represented by $\mathbf{(}M_{FFL},(1,1,m)\mathbf{)}$ [Fig. 2(c)]. Such a generalization has only one degree of freedom—the multiplicity of the Z role in the structure. There are cases, such as the multiplicity of more than one role, where we need additional definitions in order to distinguish between different types of structures. For this we define the generalization rule. We define two possible generalization rules: a strong generalization rule and a weak generalization rule. An example of a strong and weak $\mathbf{(}M_{FFL},(2,1,2)\mathbf{)}$ generalization is illustrated in Fig. 2(d). If S is the basic n-node subgraph with a set of L roles represented by the multiplicity vector $(d_1,\ldots,d_L)$, then a *basic n-node set* is every set of n nodes in the structure that consists of d_i nodes of role i (for all $1\leqslant i \leqslant L$). For example every set of three nodes in the multi-output FFL, consisting of the X node, Y node, and one of the Z-role nodes, is a *basic n-node set*. A strong generalization rule requires that every *basic n-node set* in the structure form the basic subgraph S. A weak generalization rule requires that every node in the structure participate in at least one *basic n-node set* [Fig. 2(d)]. Note that weak generalization can represent more than one unique structure of a given size.

APPENDIX C: ALGORITHM FOR DETECTING MOTIF GENERALIZATIONS

We begin by finding the network motifs (significant subgraphs) of size n (usually $n=3-4$) in the network as described in [14,15,27] (application and source code are available at http://www.weizmann.ac.il/mcb/UriAlon/). For each motif, for each of its roles, we prepare a list of all the nodes that play that role. We perform a search for all of the generalizations of each motif using its appearances in the network as starting point. This search reduces computation time and enables the detection of significant generalization forms of the basic motifs, which are beyond reach using algorithms that attempt to enumerate all subgraphs of a given size.

In order to compute the statistical significance of a certain generalization of a motif S, we first find for each appearance of S in the network the maximal size generalization in which it appears. Then we count the cumulative number of times S appears in the union of all the maximal generalizations (up to size k). In order to verify that the generalization significance is not due to many stand-alone appearances of the basic subgraph (e.g., a single-Z FFL in the case of multi-Z FFL generalization), we subtract the number of times S appears as a stand-alone structure in the network from the cumulative results (note that in Fig. 3 we show the results before subtractions). We compare these numbers to the corresponding numbers in randomized networks (here we used $Z_{\text{score}}>2$). It is important to note that the randomized networks preserve the incoming, outgoing and mutual edge degrees for each node. The networks are not constrained to have the same number of three-node or higher subgraphs as in the real network (in

[14] in contrast, four-node motifs were detected based on randomized networks that preserved three-node subgraph counts).

The network is described by a directed interaction graph $G=(V,E)$, where V is the set of nodes and E is the set of edges. An edge $(v_i,v_j)\in E$ represents a directed link between nodes v_i and v_j. For every n-node subgraph S which is detected as a network motif [14,15] we search for its simple generalizations (multiplicity of one of the roles). We begin by building an induced graph $G'=(V',E')$. The nodes in G' are only those that act as members (nodes) of S appearances in G, and the edges are only the edges in G between these nodes. G' is usually a much smaller graph than G, but it contains all the information we need for our purpose. For each simple generalization type j (multiplicity of the jth role of the subgraph) the following is performed: A nondirected graph $\hat{G}=(\hat{V},\hat{E})$ is built where each node represents a specific basic subgraph S in G (a specific set of nodes in G that form a subgraph of type S). The number of nodes in $\hat{G}$ equals the number of times S appears in the original graph G. Two nodes in $\hat{G}$ are connected if and only if they follow the generalization type j and the generalization rule (strong or weak). Setting the edges in $\hat{G}$ is done efficiently by using the appearances of the basic subgraph in G' as starting points. For each specific "starting point" subgraph S_1 in G' we pass through all the "neighboring" subgraphs S_2 ("neighboring" in the sense that they share all node roles excluding jth node roles) and check if the joint subgraph $(S_1\cup S_2)$ in G' forms a generalization type j. After setting all edges in $\hat{G}$, the next step is to find all maximal cliques [51] (a group of nodes in which every two are connected) in $\hat{G}$. Each maximal clique represents a maximal generalization type j of S (i.e., the generalization with maximal number of appearances of the basic subgraph). We store the size and the members (nodes in the original network) of all maximal generalizations. Complex generalizations (where more than one role is replicated) were detected in a similar way by appropriately changing the rules for setting the edges in $\hat{G}$.

APPENDIX D: NETWORK DATABASES

Transcription network of *E.coli* [15], version 1.1 ($N=423,E=519$), available at http://www.weizmann.ac.il/mcb/UriAlon/, was based on selected data from [52] and literature. Transcription network of yeast *(S. cerevisiae)* [14], version 1.3 ($N=685,E=1052$), available at http://www.weizmann.ac.il/mcb/UriAlon/, was based on selected data from [53] (N=number of nodes, E=number of edges). Self-edges were excluded. The neuronal synaptic connection network of *C. elegans* ($N=280,E=400$) was based on [30] as arranged in [31]. The network was compiled with a cutoff of at least five synapses for connections between neurons. Target muscle cells were excluded. Electronic forward-logic chips [14] were obtained by parsing the ISCAS89 benchmark data set [33] available at www.cbl.ncsu.edu/CBL_Docs/iscas89.html. Bi-fan generalizations data (Table I) are shown for chip S15850 ($N=10383,E=14240$).

[1] L. H. Hartwell, J. J. Hopfield, S. Leibler, and A. W. Murray, Nature (London) **402**, C47 (1999).
[2] C. A. Ouzounis and P. D. Karp, Genome Res. **10**, 568 (2000).
[3] H. McAdams and A. Arkin, Curr. Biol. **10**, R318 (2000).
[4] M. B. Elowitz and S. Leibler, Nature (London) **403**, 335 (2000).
[5] M. A. Savageau, Chaos **11**, 142 (2001).
[6] C. V. Rao and A. P. Arkin, Annu. Rev. Biomed. Eng. **3**, 391 (2001).
[7] S. H. Strogatz, Nature (London) **410**, 268 (2001).
[8] H. Bolouri and E. H. Davidson, BioEssays **24**, 1118 (2002).
[9] J. Hasty, D. McMillen, and J. J. Collins, Nature (London) **420**, 224 (2002).
[10] U. Alon, Science **301**, 1866 (2003).
[11] J. J. Tyson, K. C. Chen, and B. Novak, Curr. Opin. Cell Biol. **15**, 221 (2003).
[12] S. Maslov and K. Sneppen, Science **296**, 910 (2002).
[13] M. Newman, SIAM Rev. **45**, 167 (2003).
[14] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon, Science **298**, 824 (2002).
[15] S. Shen-Orr, R. Milo, S. Mangan, and U. Alon, Nat. Genet. **31**, 64 (2002).
[16] R. Milo *et al.*, Science **303**, 1538 (2004).
[17] T. I. Lee *et al.*, Science **298**, 799 (2002).
[18] S. Mangan and U. Alon, Proc. Natl. Acad. Sci. U.S.A. **100**, 11980 (2003).
[19] S. Mangan, A. Zaslaver, and U. Alon, J. Mol. Biol. **334**, 197 (2003).
[20] R. Dobrin, Q. K. Beg, A. L. Barabasi, and Z. N. Oltvai, BMC Bioinformatics **5**, 10 (2004).
[21] M. Ronen, R. Rosenberg, B. I. Shraiman, and U. Alon, Proc. Natl. Acad. Sci. U.S.A. **99**, 10 555 (2002).
[22] A. Zaslaver, A. Mayo, M. Surette, R. Rosenberg, P. Bashkin, H. Sberro, M. Tsalyuk, and U. Alon, Nat. Genet. **36**, 486 (2004).
[23] C. H. Yuh, H. Bolouri, and E. H. Davidson, Science **279**, 1896 (1998).
[24] N. Buchler, U. Gerland, and T. Hwa, Proc. Natl. Acad. Sci. U.S.A. **100**, 5136 (2003).
[25] Y. Setty, A. E. Mayo, M. G. Surette, and U. Alon, Proc. Natl. Acad. Sci. U.S.A. **100**, 7702 (2003).
[26] S. Itzkovitz, R. Milo, N. Kashtan, G. Ziv, and U. Alon, Phys. Rev. E **68**, 026127 (2003).
[27] N. Kashtan, S. Itzkovitz, R. Milo, and U. Alon, Bioinformatics **20**(11), 1746 (2004).
[28] J. Nesetril and S. Poljak, Comments Math. Univ. Carol. **26**, 415 (1985).
[29] F. Harary and E. M. Palmer, *Graphical Enumeration* (Academic Press, New York, 1973).
[30] J. White, E. Southgate, J. Thomson, and S. Brenner, Philos.

Trans. R. Soc. London, Ser. B **314**, 1 (1986).
[31] T. B. Achacoso and W. S. Yamamoto, *AY's Neuroanatomy of C. elegans for Computation* (CRC Press, Baton Rouge, 1992).
[32] We note that in the neuronal network where edges represent all synaptic connections (not only those with five or more synapses), we find also examples of the multi-*Z* and multi-*Y* FFL's, with the multi-*X* FFL the most common structure (data not shown).
[33] F. Brglez, D. Bryan, and K. Kozminski, in *Proceedings of the IEEE International Symposium on Circuits and Systems* (IEEE, New York, 1989), pp. 1929–1934.
[34] R. F. Cancho, C. Janssen, and R. V. Sole, Phys. Rev. E **64**, 046119 (2001).
[35] S. Basu, R. Mehreja, S. Thiberge, M. T. Chen, and R. Weiss, Proc. Natl. Acad. Sci. U.S.A. **101**, 6355 (2004).
[36] H. McAdams and A. Arkin, Annu. Rev. Biophys. Biomol. Struct. **27**, 199 (1998).
[37] N. Rosenfeld, M. B. Elowitz, and U. Alon, J. Mol. Biol. **323**, 785 (2002).
[38] N. Rosenfeld and U. Alon, J. Mol. Biol. **329**, 645 (2003).
[39] S. Kalir, J. McClure, K. Pabbaraju, C. Southward, M. Ronen, S. Leibler, M. G. Surette, and U. Alon, Science **292**, 2080 (2001).
[40] S. Kalir and U. Alon, Cell **117**(6), 713 (2004).
[41] M. Chalfie, J. E. Sulston, J. G. White, E. Southgate, J. N. Thomson, and S. Brenner, J. Neurosci. **5**, 956 (1985).
[42] I. A. Hope, *C. elegans: A Practical Approach* (Exford University Press, Exford, 1999).
[43] M. B. Goodman, D. H. Hall, L. Avery, and S. R. Lockery, Neuron **20**, 763 (1998).
[44] M. C. Hansen, H. Yaclin, and J. P. Hayes, IEEE Design Test **16**(3), 72 (1999).
[45] G. C. Conant and A. Wagner, Nat. Genet. **34**, 264 (2003).
[46] S. Wasserman and K. Faust, *Social Network Analysis* (Cambridge University Press, Cambridge, England, 1994).
[47] F. Lorrain and H. C. White, J. Math. Sociol. **1**, 49 (1971).
[48] C. Winship, Soc. Networks **10**, 209 (1988).
[49] C. Winship and M. Mandel, Soc. Methodol. **1983-1984**, 314 (1983).
[50] M. G. Everett, J. P. Boyd, and S. P. Borgatti, J. Math. Sociol. **15**, 163 (1990).
[51] C. Bron and J. Kerbosch, Commun. ACM **16**, 575 (1973).
[52] H. Salgado, A. Santos-Zavaleta, S. Gama-Castro, D. Millan-Zarate, E. Diaz-Peredo, F. Sanchez-Solano, E. Perez-Rueda, C. Bonavides-Martinez, and J. Collado-Vides, Nucleic Acids Res. **29**, 72 (2001).
[53] M. C. Costanzo *et al.*, Nucleic Acids Res. **29**, 75 (2001).
[54] S. Itzkovitz, R. Levitt, N. Kashtan, R. Milo, M. Itzkovitz, and U. Alon, Phys. Rev. E (to be published).
[55] E. Ziv, R. Koycheff, and C. Wiggins, e-print cond-mat/0306610.
[56] J. Berg and M. Lässig, e-print cond-mat/0308251.