

# Self-correcting networks: Function, robustness, and motif distributions in biological signal processing

Pablo Kaluza,<sup>1</sup> Martin Vingron,<sup>2</sup> and Alexander S. Mikhailov<sup>1</sup>

<sup>1</sup>Fritz-Haber-Institut der Max-Planck-Gesellschaft, Faradayweg 4-6, 14195 Berlin, Germany

<sup>2</sup>Max-Planck-Institut für Molekulargenetik, Ihnestr. 35, 14195 Berlin, Germany

(Received 3 March 2008; accepted 21 May 2008; published online 27 June 2008)

Statistical properties of large ensembles of networks, all designed to have the same functions of signal processing, but robust against different kinds of perturbations, are analyzed. We find that robustness against noise and random local damage plays a dominant role in determining motif distributions of networks and may underlie their classification into network superfamilies. © 2008 American Institute of Physics. [DOI: 10.1063/1.2945228]

High robustness against local damage and noise is a fundamental property of biological networks.<sup>1</sup> In gene knock-out experiments in which effects of genes are studied by making inoperative one gene per experiment, it has been found that only 13% of genes in yeast are essential, so that their absence is lethal.<sup>2</sup> Because many subsystems of the cell are small, they are subject to strong stochastic fluctuations, but this also does not prevent regular cell functioning. To suppress fluctuations and increase the stability of dynamical systems, negative feedback loops can be used (cf. Ref. 3) and such mechanisms of noise reduction are indeed employed by the cells. Alternatively, robustness against random damage can be achieved through the use of *self-correcting network architectures*. Can small self-correcting networks with only tens of elements, typical for functional modules of biological cells, be constructed? What are characteristic statistical properties of functional networks capable of self-correction? Here, these questions are addressed for a class of networks that should produce predefined responses following activation of different input elements. In the realm of biology, this class includes not only signal-transduction networks of the cell, but also genetic networks. Moreover, the entire nervous system of primitive multicellular organisms may also be viewed as belonging to such a class. In our analysis, an approach that can be described as *constructive biology* is chosen. Instead of investigating real biological networks, we consider a simple flow distribution model of signal processing where nonlinear feedbacks are excluded. By running evolutionary optimization algorithms, large ensembles of networks with the same size and the same output patterns, but robust against different kinds of local damage or distributed noise, are designed. This allows us to analyze how statistical properties of self-correcting networks depend on the kind of random perturbations against which they are robust. Special attention is paid to motif distributions of networks.<sup>4</sup> We discover that our designed networks, robust against random link deletions, have motif distributions characteristic for the second superfamily previously identified by Alon and co-workers,<sup>4</sup> which includes signal transduction and genetic developmental networks of multicellular organisms and the neural system of the

nematode *Caenorhabditis elegans*. Completely different motif distributions, which belong instead to the fourth superfamily, are characteristic for networks designed to be robust against node deletions or against noise.

## I. INTRODUCTION: BIOLOGICAL SIGNAL PROCESSING SYSTEMS AND THEIR MOTIF DISTRIBUTIONS

The nematode *Caenorhabditis elegans* (*C. elegans*) is a tiny animal whose body is formed by only 959 cells, 302 of which are neurons. The complete structure and the synaptic connectivity of its nervous system are known.<sup>5</sup> It represents a small sensory transduction neural network whose functions are mostly to generate different motor responses (such as chemotaxis) when specific external signals arrive. Generally, this nervous system consists of sensory neurons, interneurons, and motor neurons. When an external signal arrives at one of the sensory neurons, it propagates through some interneurons and eventually reaches a subset of motor neurons, thus generating a stimulus-induced response. The pool of interneurons is not further organized into layers. Each signal passes only through a few interneurons before it reaches the output.

Similar organization is characteristic for signal transduction inside the cell, but signals are generated and transmitted then at the level of single molecules. A small extracellular signal molecule arrives at a receptor protein. The chemical signal propagates through a group of intracellular signalling proteins. As the output, different target proteins are produced. Depending on their type, they can alter metabolism (metabolic enzymes), change gene expression (gene regulatory proteins), and induce cell movements or shape changes (cytoskeletal proteins). Usually, signal passes through several proteins before reaching a target.

Sensory transcription networks in bacteria or yeast control gene expression in response to external stimuli. In these networks, nodes represent genes or operons and edges represent direct transcriptional regulation. These networks are shallow, with only very short paths connecting input and output nodes. Developmental transcription networks guide evolution from an initial egg cell to the mature multicellular

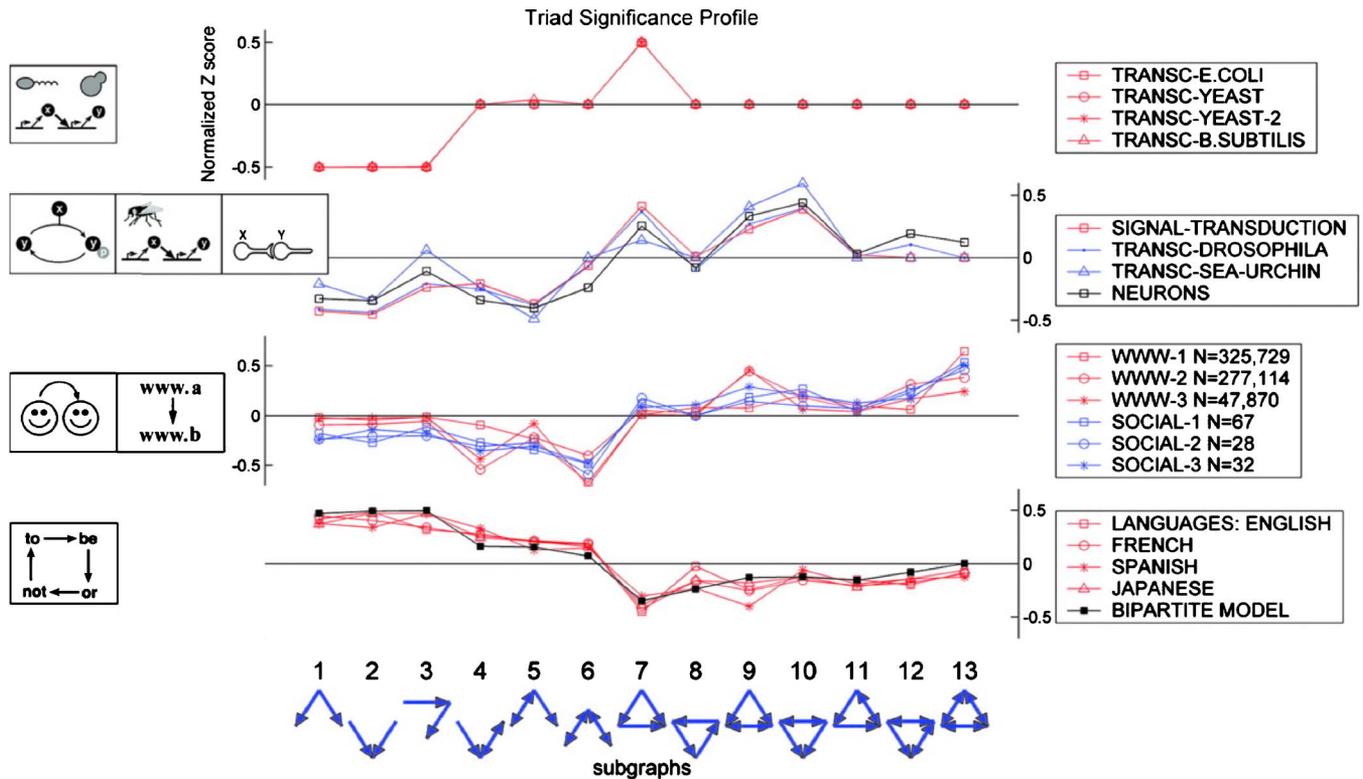


FIG. 1. (Color online) Three-node motif profiles of various networks. Networks with similar characteristic distributions are grouped into superfamilies. From Ref. 4.

organism. Once a certain stage in individual development has been reached, a signal is generated that, after processing, switches on the next genetic developmental program by activating a different combination of genes. Thus, responses are generated and, in this respect, such genetic networks are similar to the protein networks of signal transduction.

Direct comparison between these different biological signal processing systems is difficult. Indeed, completely different types of elements (proteins, genes, entire cells) are involved in them and mechanisms of interactions also differ strongly. But even abstracting from the dynamics and considering only structural organization of the networks, the comparison is not straightforward. Networks of the same kind (such as, e.g., cellular signal transduction networks) have different sizes and different numbers of connections. Standard tools of statistical analysis are not sensitive enough to grasp possible similarities. Looking for advanced statistical methods to characterize and to compare structural organizations of different networks, Alon and co-workers<sup>4</sup> have proposed to use *motif distributions*.

A three-node motif is a subgraph formed by three network elements and their connections. For directed networks, there are only 13 different motifs, all of which are shown in the bottom panel of Fig. 1. Taking a graph  $G$ , which may correspond to a pattern of synaptic connections in a neural network or to a pattern of interactions between proteins, one can count the number of times  $N_i^{\text{real}}$  some motif  $i$  is present in this graph. But is this number large or small? To answer the question, this number should be compared with the number of times  $N_i^{\text{random}}$ , the same motif is found in the randomized version of graph  $G$ .

Randomization is performed by rewiring connections in such a way that numbers of incoming and outgoing connections for each node are preserved.<sup>4</sup> It yields an ensemble of randomized versions of the original graph. Using this ensemble, the mean number  $\langle N_i^{\text{random}} \rangle$  of appearances of motif  $i$  in the random ensemble and its statistical variation  $\text{std}(N_i^{\text{random}})$  can be determined. Once this is done, the significance score  $z_i$  of each motif  $i$  in graph  $G$  can be determined as

$$z_i = \frac{N_i^{\text{real}} - \langle N_i^{\text{random}} \rangle}{\text{std}(N_i^{\text{random}})}. \quad (1)$$

When  $z_i > 0$ , motif  $i$  is over-represented, if  $z_i < 0$  it is under-represented. The motif distribution of a considered network is yielded by the vector of normalized scores

$$\bar{z}_i = \frac{z_i}{\sqrt{\sum_{i=1}^{13} z_i^2}}. \quad (2)$$

Alon and co-workers<sup>4</sup> have determined and compared motif distributions for various networks of biological, social, and informational origin. Figure 1 summarizes the results of their analysis. All networks could be divided with respect to their motif distributions into four distinct *superfamilies*. The first superfamily is formed by sensory transcription networks of bacteria and yeast. The second superfamily consists of signal-transduction networks in mammalian cells, transcription networks guiding the development in fruit fly and sea urchin, and the network of synaptic connections in *C. elegans*. The third superfamily includes networks, corresponding to WWW hyperlinks between web pages, and so

cial networks. Word-adjacency networks of texts in English, French, Spanish, and Japanese, as well as the bipartite graphs, form the fourth superfamily.

Identification of network superfamilies has been a fundamental discovery whose importance may still be not fully appreciated. Networks of the same kind can be expected to show *qualitatively* similar motif distributions. However, there is actually a *quantitative* agreement between the motif distributions of the networks inside each superfamily. The dispersion of data within each superfamily is narrow and motif distributions of different networks in each superfamily almost exactly coincide. For a physicist, this evokes analogies to “universality classes” in phase transitions and percolation phenomena, where behavior of very different physical systems becomes strikingly close.

The conjecture of Alon and his co-workers<sup>4</sup> was that the superfamilies are determined by *common functions* of the respective networks. Indeed, sensory transcription networks of microorganisms in the first superfamily need, for example, to generate fast responses and cannot therefore have long cascades. In contrast to this, biological information-processing networks of the second superfamily are much slower in their operation and typically include long cascades. Such arguments can however only explain the qualitative similarity between networks in a superfamily. Moreover, they leave, e.g., open the question whether *any* information-processing network with long cascades must belong to the second superfamily of motifs. Thus, the complete explanation of mechanisms underlying classification of networks in motif superfamilies is still missing.

Working with real biological systems, it is usually difficult to distinguish what determines their particular properties. Any organism is a product of long evolution that has led to joint optimization of its many interwoven functions. To analyze separately the effects of individual functions, methods of *constructive biology* can be used. Instead of studying real bio-organisms, one tries then to design artificial systems which intend to reproduce only specific aspects of the living cell.

In the problem of motif superfamilies of networks, the constructive approach should consist of designing large ensembles of networks with different predefined functions and comparing their statistical properties. Considering ensembles of networks with the same function, one can analyze variation of motif distributions within an ensemble. One can also look whether other criteria, such as robustness, are instead determining the motif distributions.

The constructive approach for studies of bionetworks has been followed in our recent publications.<sup>6–8</sup> Using a simplified model representing an abstraction of biological signal processing, ensembles of networks with the same output functions have been generated, which were additionally designed to be *self-correcting* against deletion of individual links or nodes, or against the application of distributed quenched noise. Our principal conclusion is that the network function *does not* yet determine its superfamily. Only the networks, robust against random link deletions, have motif distributions characteristic for the second superfamily and thus common with the biological information-processing net-

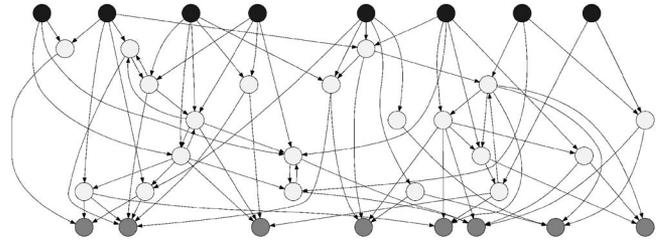


FIG. 2. Example of a flow distribution network.

works. When the same networks are designed to be self-correcting with respect to random node deletions or application of noise, they are found instead to belong to the fourth superfamily of motif distributions. The aim of this article is to provide a short review of previous publications and to present new theoretical results.

## II. FLOW DISTRIBUTION NETWORKS

Our study is performed for the *flow distribution model* of signal processing. To exclude nonlinear feedback effects, a linear model without any dynamics has been chosen. This toy model is not intended to provide a realistic description of any particular biological system. On the other hand, it retains principal features characteristic for various biological signal processing phenomena. In the model, signals are represented by fluxes that flow through a network (Fig. 2). Amplification or attenuation of signals are absent. Network nodes are flux distribution elements: each node splits its total incoming flux in equal parts and sends it further through its outgoing connections. There is no storage of flow in the nodes. External signals are applied to a set input nodes. Network responses are determined by the signals reaching a set of output nodes when one of the input nodes is activated.

The formal definition of the model is as follows. A network consists of  $N_{\text{in}}$  input nodes,  $M$  middle nodes, and  $N_{\text{out}}$  output nodes. Input nodes can be connected only with middle nodes, middle nodes can be connected among themselves or with the output nodes. The network  $G$  is defined by its adjacency matrix  $A$ .

Each directed link  $j \rightarrow i$ , corresponding to two connected elements ( $A_{ij} = 1$ ), carries some flux  $u_{ij}$ . The total flux  $x_i$  passing through node  $i$  is equal to the sum of its incoming fluxes,

$$x_i = \sum_j A_{ij} u_{ij}. \quad (3)$$

The total number of outgoing connections of a node  $j$  is  $m_j^{\text{out}} = \sum_k A_{kj}$ . The flux is divided in equal parts among all of them and therefore the flux  $u_{ij}$  through each physically present link  $j \rightarrow i$  is

$$u_{ij} = \frac{x_j}{\sum_k A_{kj}}. \quad (4)$$

Substituting Eq. (4) into Eq. (3), relationships between fluxes passing through different network nodes are obtained:

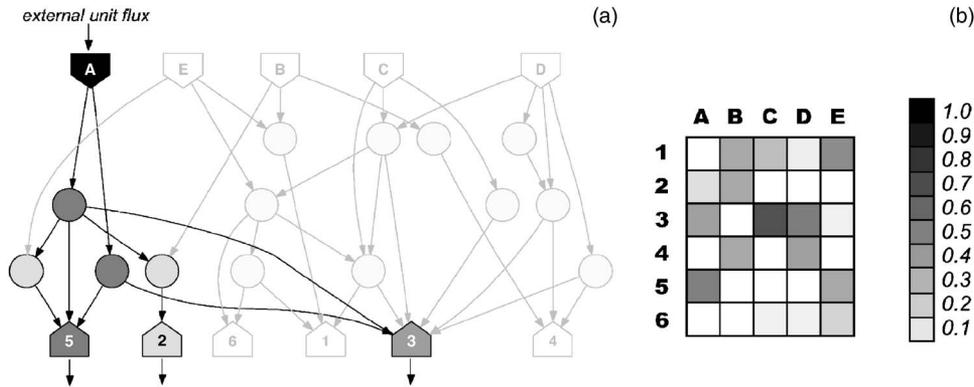


FIG. 3. (a) Subnetwork, activated when the flux is applied to node A. Total input flux in each node is displayed by gray scale, varying from black, when it is one, to white, when it is zero. (b) Output pattern matrix  $Q$  of the considered network displayed in gray scale.

$$x_i = \sum_{j=1}^N A_{ij} \frac{x_j}{\sum_{k=1}^N A_{kj}}. \quad (5)$$

These equations hold for  $M$  middle and  $N_{\text{out}}$  output nodes. The fluxes arriving at  $N_{\text{in}}$  input nodes are external and fixed.

When a unit external flux  $x_\alpha=1$  is applied to an input node  $\alpha$ , it is distributed [Fig. 3(a)] by middle nodes and reaches some output nodes  $\beta$ . The induced output fluxes can be found by solving Eq. (5). They determine the response of a network to stimulation of its input node  $\alpha$ . In the figures, we always use capital letters (A, B, C, ...) to indicate input nodes and numbers (1, 2, 3, ...) to enumerate output nodes.

Let  $Q_{\beta\alpha}$  be the flux arriving at the output node  $\beta$  when a unit flux is applied to the input node  $\alpha$ . The total flow is conserved in the model, implying that  $\sum_{\beta=1}^{N_{\text{out}}} Q_{\beta\alpha} = 1$  for any  $\alpha$ . The  $N_{\text{out}} \times N_{\text{in}}$  matrix  $Q$  with elements  $Q_{\beta\alpha}$  contains network responses to the activation of all its input nodes. Therefore, it represents the *output pattern* of the network  $G$ .

Because all elements of matrix  $Q$  are non-negative and do not exceed unity, a graphic visualization of output patterns is possible. An output pattern is displayed then as an array where each pixel is gray-shaded according to the value of the respective matrix element [Fig. 3(b)]. Thus, each column in the array represents the response of a network to the activation of one of its input nodes.

### III. EVOLUTIONARY CONSTRUCTION OF NETWORKS WITH PREDEFINED OUTPUT PATTERNS

The first problem that should be considered is how to construct a flow distribution network with a predefined output pattern. If there is no restriction on the maximal number of middle nodes, the construction is straightforward. First, one designs separate network blocks, each implementing the required response to the activation of a different input node. In each single block, the fluxes are repeatedly split by the middle nodes and arrive at the output nodes. After that, the total distribution network can be obtained by taking together separate distribution blocks. The solution becomes however far from trivial, when the number of middle nodes is restricted and relatively small. In this situation characteristic for biological signal processing, one cannot just take a sum of subnetworks each implementing a response to a different stimulus, because then the total number of employed middle nodes would be too high. Most of the available middle nodes

should be involved now in generation of network responses to several stimuli. Since competing flow distribution conditions must be simultaneously satisfied, frustration emerges, and a simple rational solution is no longer feasible.

Network construction can be seen as an optimization problem. Suppose that the actual output pattern of network  $G$  is  $Q$ , whereas the ideal (prescribed) output pattern is  $Q^0$ . The flow error  $\varepsilon(G)$  of the network  $G$  with respect to the ideal output pattern is the distance between the two matrices, i.e.,

$$\varepsilon(G) = \frac{1}{2N_{\text{in}}} \sum_{\alpha=1}^{N_{\text{in}}} \sum_{\beta=1}^{N_{\text{out}}} (Q_{\beta\alpha} - Q_{\beta\alpha}^0)^2. \quad (6)$$

The optimization task is to minimize the flow error and (possibly) bring it to zero by changing the network  $G$ .

Generally, for each network there is a (vector) function  $Q = F(G)$  that determines its particular performance. Distance between two networks  $G_1$  and  $G_2$  with respect to this kind of performance can be defined as  $\varepsilon = |F(G_1) - F(G_2)|^2$ . If the ideal performance  $Q^0$  is known, the distance of any network from such ideal performance can be defined as  $\varepsilon = |F(G) - Q^0|^2$ . This distance represents the functional error of a network.

An approximate approach to problems of complex combinatorial optimization, such as the traveling salesman problem, is provided by the method of *simulated annealing*.<sup>9</sup> A variant of this method has been used by us to design flow distribution networks.<sup>6</sup>

To construct a desired network with the chosen ideal performance  $Q^0$ , an evolutionary optimization process is run. Each iteration consists of the same steps:

- (1) Take a network  $G$  with a flow error  $\varepsilon$ .
- (2) Apply an evolutionary mutation to  $G$ , obtaining a new network  $G'$  with a different flow error  $\varepsilon'$ .
- (3) Calculate the difference  $\Delta\varepsilon = \varepsilon' - \varepsilon$ .
- (4) If  $\Delta\varepsilon < 0$ , accept the mutation and replace  $G$  by  $G'$ . If  $\Delta\varepsilon > 0$ , accept the mutation with probability  $\exp(-\Delta\varepsilon/\varepsilon\sigma)$ .
- (5) Return to step 1.

The evolution is started with a random initial network  $G_{\text{init}}$  and continued until the error becomes equal to zero or gets small enough (or for a fixed number of iterations).

This method can also be considered as the Metropolis optimization algorithm, where effective temperature  $\theta = \varepsilon\sigma$  is

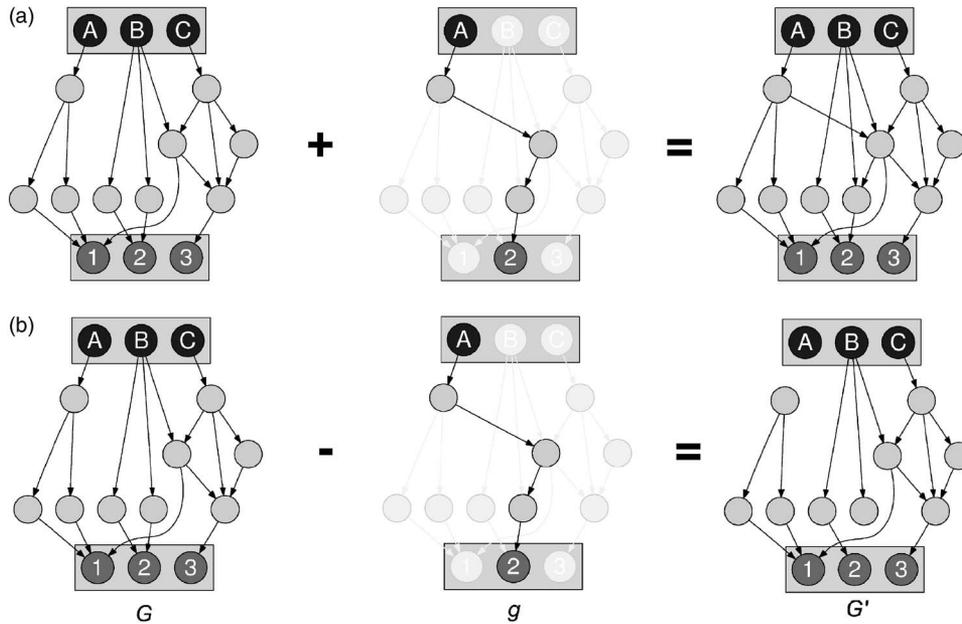


FIG. 4. The network  $G$  is mutated to network  $G'$  through the addition (a) or deletion (b) of the path  $g$ .

proportional to the error  $\varepsilon$  and gradually decreases as the error diminishes and the target is approached.

Different kinds of network mutations can be employed in the optimization process. Two mutation schemes have been tried.<sup>10</sup> In the *link mutation* scheme, a mutation consists in adding a new connection or deleting an existing one from the network. Although this scheme has been found to lead to an approximate solution, convergence was slow and final networks tended to have large numbers of connections. Therefore, a different *path mutation* scheme was usually chosen. In each elementary mutation, a path connecting an input node and an output node and passing through the middle nodes is randomly generated (Fig. 4). Subsequently, this path is either added to the network  $G$  or subtracted from it, yielding the modified network  $G'$ . During the evolution based on the path mutation scheme, the total number of middle nodes is preserved. In the final network, some middle nodes can turn out to be “dead;” i.e., having no incoming connections. Such nodes are removed from the networks, so that the number of middle nodes in the final network may become smaller.

Figure 5 shows an example of an evolutionary construction. The evolution was started with the initial network, whose output pattern was strongly different from the required “target” pattern. After  $3 \times 10^5$  iterations, a network with the output pattern very close to the target was constructed.

#### IV. ROBUSTNESS AGAINST LOCAL DAMAGE

What happens with an animal if one of its neural cells dies or one of its synaptic connections is broken? How is the performance of a genetic network changed if one gene is knocked out? How are responses of a living cell to external stimuli changed if one of its intracellular signalling proteins is absent or modified? It is a striking feature of biological signal processing systems that, in most cases, nothing dramatic would actually occur. Often, the cell or animal will

continue to function almost in the same way despite the damage. High robustness of biological systems can be achieved through the use of network architectures where, if a link or a node is destroyed, signals will pass over a different route without significantly changing the network performance. In this sense, biological signal processing networks are self-correcting.

If an electronic engineer is asked to design a self-correcting network, his solution would probably be based on the parallelization methods. By making several copies of each signal processing block and connecting them in parallel, one can ensure that the operation is not affected by breakdown of a particular block. This straightforward solution is however only applicable if there are no restrictions on the total number of processing elements. When the number of such elements is limited and must be relatively small, simple parallelization is no longer possible. Although rational solutions are not then feasible, evolutionary optimization methods can still be used to design such compact self-correcting networks.

Biological networks are characterized by the functions they play within an organism. Robustness of such networks can therefore be discussed only in connection to a particular function that given networks must exercise. A subsystem of a living cell is generally viewed as robust if it continues to operate in a satisfactory way after damage has taken place. A network is robust with respect to a particular performance if, under damage, its performance does not typically deteriorate much. Note that the same network can therefore be robust with respect to one of its functions and vulnerable with respect to another. Moreover, robustness should generally depend on the kind of damage applied.

Robustness of a network is determined by properties of the ensemble of networks obtained by applying various damage to the given network; this ensemble can be called the *damage shell* of the considered network. When robustness against node deletions is considered, the damage shell of a

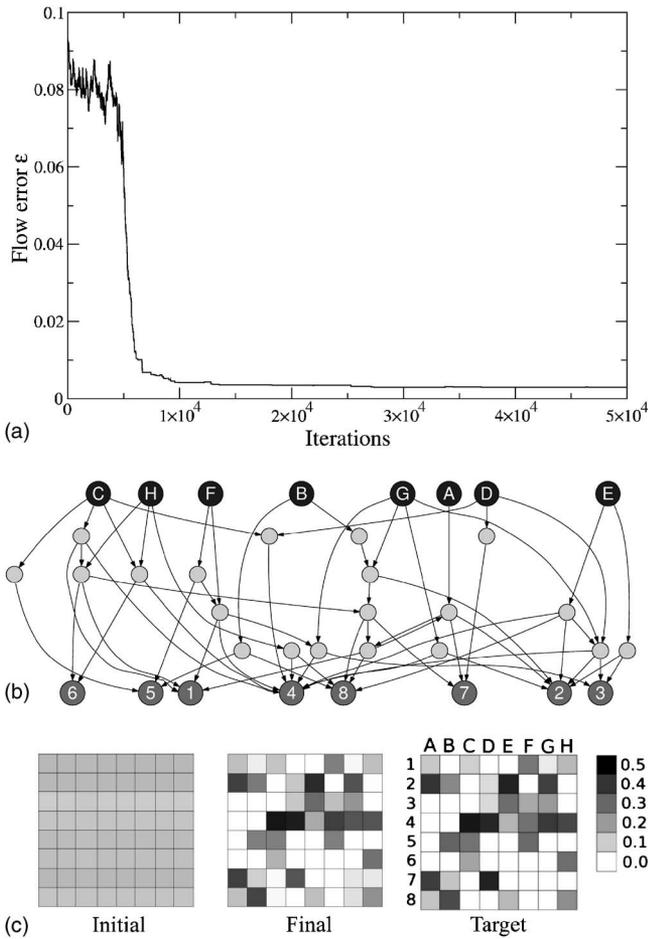


FIG. 5. Example of network construction. The temperature parameter is  $\sigma = 10^{-4}$ . The network has  $N=36$  nodes with  $N_{in}=8$ ,  $M=20$ , and  $N_{out}=8$ ; the target output pattern has  $K=4$  activated nodes. (a) Temporal evolution of the flow error  $\epsilon$ . (b) Final network obtained after  $5 \times 10^5$  iterations. (c) Initial, final, and target output patterns displayed in gray scale.

network includes all networks obtained by deleting one of its nodes. To determine robustness against link deletions, damage shells obtained by deleting one of the links in a given network should be analyzed.

Each network  $G_i$  in a damage shell has some error  $\epsilon_i$  with respect to the ideal performance. Robustness of a network is controlled by the distribution of errors in its damage shell. One may define robustness as the mean error  $\langle \epsilon_i \rangle$  of the networks in the damage shell. The disadvantage of such definition would, however, be that high statistical weights are attached to those few networks in a shell that may turn out to have exceptionally large errors. For a biological organism, any error exceeding some tolerance threshold would, however, lead to lethal consequences and its absolute magnitude is then not important. All damages that bring a system outside of its operation interval are equally dangerous.

Having this in mind, we define<sup>6</sup> robustness  $\rho(G)$  of network  $G$  with tolerance threshold  $h$  as the ratio of networks in the damage shell that have an error less than the threshold  $h$ , i.e.,

$$\rho(G) = \frac{1}{N_d(G)} \sum_{i=1}^{N_d(G)} \Theta(h - \epsilon_i), \quad (7)$$

where  $N_d(G)$  is the number of networks in the damage shell of  $G$  and  $\Theta(z)$  is the step function;  $\Theta(z)=1$  for  $z \geq 0$  and  $\Theta(z)=0$  otherwise. For flow distribution networks, the performance of any network is given by its output pattern  $\mathbf{Q}$ . The error of such a network is given by the distance equation (6) from the ideal output pattern.

## V. ROBUSTNESS AGAINST NOISE

In molecular biology, two sources of noise are generally distinguished. Internal noise is due to the stochastic nature of chemical reactions. Each reaction occurrence is a stochastic event that involves the reacting molecules. When the number of reacting molecules is small, reaction rates show substantial fluctuations. External noise is caused by random variations of external parameters, such as temperature. Usually, noise affects at the same time all components of a system. Therefore, it can be also viewed as a kind of distributed random damage. Moreover, only time-dependent fluctuations are usually classified as noise. The model of flow distribution networks is static and therefore dynamical fluctuations cannot be investigated there. It is, however, possible to study in its framework how weak distributed perturbations would affect the network performance. These distributed perturbations can be viewed as quenched noise.

We introduce noise into the model by allowing weak random modifications of the links.<sup>7</sup> If outgoing connections of a node have different weights, the flux arriving to the node is distributed over its outgoing links proportionally to their weights  $W_{ij}$ , so that Eq. (4) is replaced by

$$u_{ij} = \frac{W_{ij}x_j}{\sum_k W_{kj}}. \quad (8)$$

Accordingly, the fluxes  $x_i$  now satisfy the set of equations

$$x_i = \sum_{j=1}^N W_{ij} \frac{x_j}{\sum_{k=1}^N W_{kj}}. \quad (9)$$

If  $W_{ij}=0$ , the connection  $j \rightarrow i$  is absent.

We continue to assume that in absence of noise all connections are identical; i.e.,  $W_{ij}=1$  or  $W_{ij}=0$  for any link. Noise randomly decreases the weights of all existing connections and creates all possible new weak links. For each  $i$  and  $j$ , an independent random variable  $\xi_{ij}$  with a uniform distribution in the interval  $[0, 1]$  is defined. When noise with intensity  $S$  is applied, all link weights are modified as

$$W_{ij}^{\text{noise}} = \begin{cases} 1 - S\xi_{ij}, & \text{if } W_{ij} = 1, \\ S\xi_{ij}, & \text{if } W_{ij} = 0. \end{cases} \quad (10)$$

Applying different noise realizations to a network, statistical ensemble of various modified networks is obtained which can be viewed as the *noise shell* of the given network. When the ideal output is known, each network  $G(\xi)$  in the shell is characterized by its error  $\epsilon(\xi)$ . A particular noise realization  $\xi$  yields a dysfunctional network, if the resulting error exceeds the tolerance threshold; i.e., if  $\epsilon(\xi) > h$ . Ro-

bustness of a network against noise is defined as statistical average of the fraction of functional networks [with  $\varepsilon(\xi) < h$ ] networks inside the noise shell; i.e., as

$$\rho(G) = \langle \Theta(h - \varepsilon(\xi)) \rangle_{\xi}. \quad (11)$$

Thus defined, robustness depends on the noise intensity  $S$  and the tolerance threshold  $h$ . In practice, a finite number  $N_{\text{noise}}$  of noise realizations can only be applied. Averaging is then performed over the sample of  $N_{\text{noise}}$  networks inside the noise shell.

## VI. EVOLUTIONARY CONSTRUCTION OF ROBUST NETWORKS WITH PREDEFINED OUTPUT PATTERNS

As shown above, flow distribution networks with desired output patterns can be constructed by using an evolutionary optimization method. This method can be further extended to construct networks which, while having required output patterns, are additionally robust against a particular kind of damage or against noise. As a result of such optimization, networks that are functional themselves (i.e., have error below the tolerance threshold) and possess the maximum possible fraction of functional networks in their damage shells can be obtained.

In the modified evolutionary optimization algorithm,<sup>6</sup> the following steps are performed at each next iteration:

- (1) Take a network  $G$  with flow error  $\varepsilon$  and robustness  $\rho$ .
- (2) Apply an evolutionary mutation to  $G$  and obtain network  $G'$  with flow error  $\varepsilon'$  and robustness  $\rho'$ .
- (3) Calculate  $\Delta\varepsilon = \varepsilon' - \varepsilon$  and  $\Delta\rho = \rho' - \rho$ .
- (4) If the flow error is higher than the tolerance threshold ( $\varepsilon > h$ ), always accept the mutation if  $\Delta\varepsilon < 0$ , and accept it with probability  $\exp(-\Delta\varepsilon/\varepsilon\sigma)$  if  $\Delta\varepsilon > 0$ .
- (5) If the flow error is below the threshold ( $\varepsilon < h$ ), the mutation is always accepted if  $\Delta\rho > 0$ , and accepted with probability  $\exp(\Delta\rho/(1-\rho)\sigma_R)$  if  $\Delta\rho < 0$ .
- (6) Return to step 1.

Similar to the original evolutionary optimization algorithm, there are temperature parameters here:  $\sigma$  is used in the evolution minimizing the error and  $\sigma_R$  is used when robustness is being maximized. According to this algorithm, evolution is governed by minimization of the flow error as long as it is higher than the tolerance threshold. Once it falls below the threshold, the selection criterion is switched to robustness and the evolution proceeds to maximize this property. Depending on what damage shell is chosen when robustness of evolving networks is estimated, the optimization method can be used to construct networks robust either against the deletion of links or against the deletion of nodes. In our numerical investigations,<sup>6</sup> randomly generated networks were chosen as initial conditions for the optimization process. The path mutation scheme was applied to generate evolutionary mutations.

Figure 6 shows an example of an evolutionary optimization. During the first 5000 iterations, evolution is controlled by the decrease of the flow error. At the end of this stage, a network with the error below the threshold  $h=0.007$  is ob-

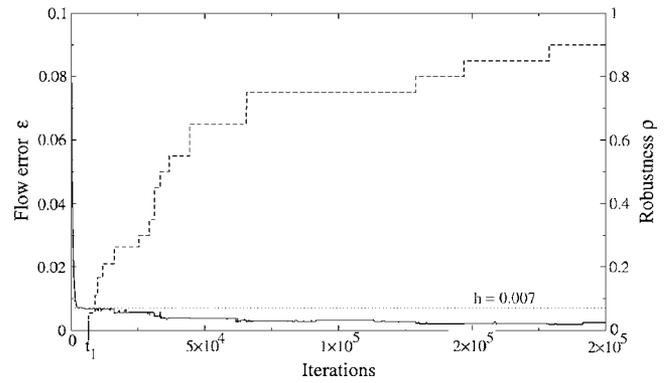


FIG. 6. Evolution of flow error (solid line) and robustness against removal of nodes (dashed line) in an optimization process. The parameters are  $h = 0.007$ ,  $N_{\text{in}} = 8$ ,  $M = 20$ ,  $N_{\text{out}} = 8$ ,  $K = 4$ , and  $\sigma = \sigma_R = 10^{-4}$ .

tained. This network is not, however, robust against node deletions and is characterized by low robustness  $\rho \approx 0.1$ . Evolution is then switched to robustness optimization. After  $4 \times 10^5$  iterations, robustness is increased to  $\rho = 0.76$ , so that the deletion of 76% of the nodes in this network still leaves its performance within the tolerance window.

Evolutionary optimization algorithms, used to construct (robust) networks with predefined output patterns and described above, represent a variant of the simulated annealing method. In these algorithms, effective temperature is introduced to prevent possible termination of an evolution when an intermediate optimum is reached. The temperature  $\theta$  changes with the evolution, vanishing as the optimization goals are reached. When output error  $\varepsilon$  is minimized,  $\theta = \sigma\varepsilon$ ; when robustness  $\rho$  is maximized,  $\theta = \sigma_R(1-\rho)$ . To determine the best values of these parameters, additional investigations have been performed.

Figure 7 displays dependence of the mean output error  $\langle \varepsilon \rangle$  on the parameter  $\sigma$ . Three data sets with different numbers  $M$  of middle nodes are shown. When  $\sigma$  is relatively large, convergence to the desired output is usually not found. Once the parameter  $\sigma$  is less than about 0.01, final output errors become however low and do not significantly depend on this parameter. A similar behavior is found when optimi-

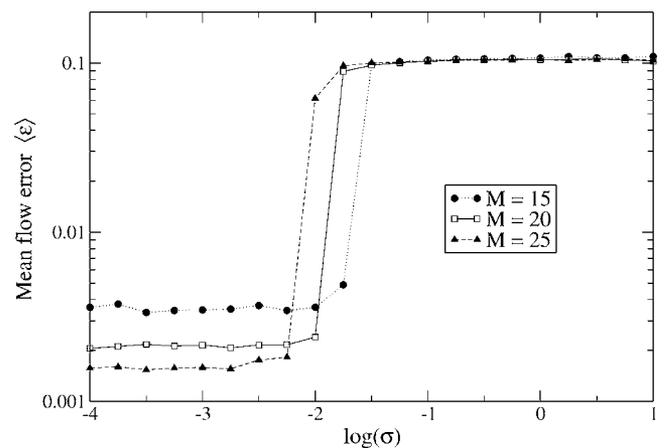


FIG. 7. Dependence of the mean flow error  $\langle \varepsilon \rangle$  on the temperature parameter  $\sigma$  for different numbers of middle nodes.

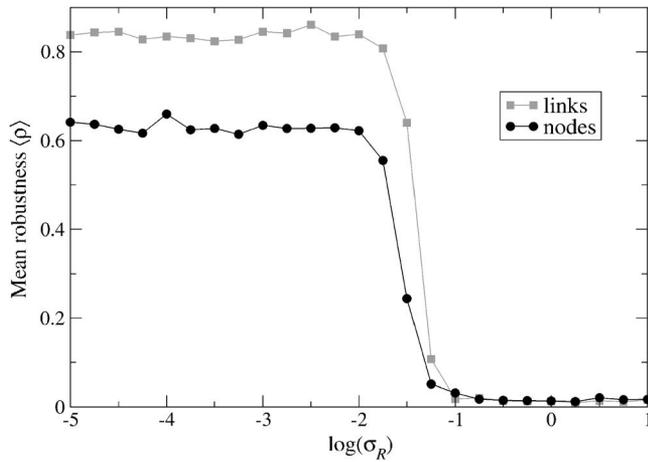


FIG. 8. Mean robustness of networks optimized against deletion of nodes (circles) or links (squares) as function of the temperature parameter  $\sigma_R$ :  $\sigma = 10^{-3}$  and the tolerance threshold is  $h=0.007$ .

zation for robustness against node or link deletion is performed (Fig. 8). Thus, for the problems considered, introduction of temperature is not essential and even gradient optimization (with  $\sigma$  or  $\sigma_R$  equal to zero) should have worked well here. In our systematic simulations, very small values of the parameters  $\sigma$  and  $\sigma_R$  have been chosen.

Evolutionary optimization algorithms were used by us to obtain large ensembles of networks with predefined output patterns, which were robust against different kinds of local damage.<sup>6,10</sup> To collect the data, each evolution was started with a different, independently generated random network. In order to eliminate the dependence of the results on a particular choice of the target output pattern, a different output pattern was randomly generated for each evolution. In the generated target patterns, only  $K=4$  output nodes received fluxes when an input node was activated. These four nodes were randomly specified for each input node and their relative activation degrees were also chosen at random.<sup>6,10</sup> All considered networks had  $N_{in}=8$  input nodes and  $N_{out}=8$  output nodes; the number of middle nodes was varied. Unless otherwise specified, ensembles of 100 independently constructed networks were used to determine statistical averages at each parameter point in all the plots shown. The evolutions were usually terminated after  $10^5$  iterations.

Black empty bars in Fig. 9 show error distributions in

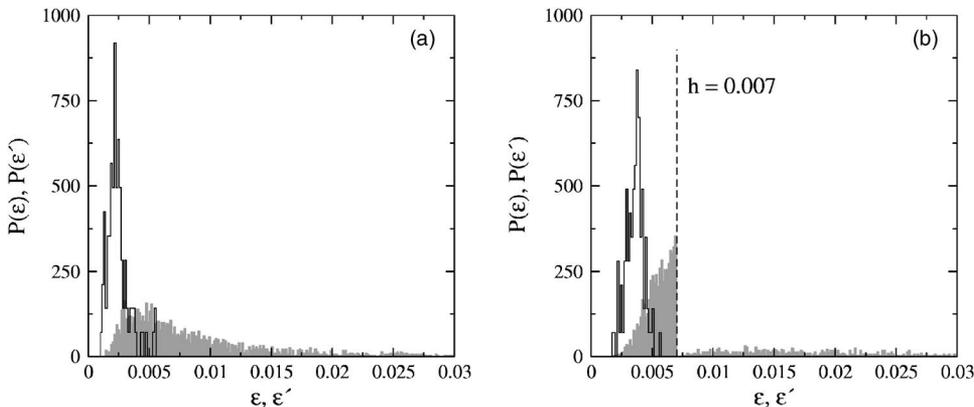


FIG. 9. Distributions of output errors  $\epsilon$  in the ensembles of 100 networks (black), optimized only for the output error (a) and optimized for the robustness against deletion of a node (b). Gray bars show distributions of errors  $\epsilon'$  of the networks in the damage shells (after removing a node).

ensembles of networks, optimized only for the output error [Fig. 9(a)] and for the robustness against deletion of a node [Fig. 9(b)]. They are similar for both ensembles. Gray filled bars display distributions of errors in damage shells of such networks (after removing a node). Here, the difference becomes obvious: The networks optimized only for the output error yield a broad distribution with a long tail extending to large errors. In contrast to this, errors exceeding the tolerance threshold are suppressed in the damage shells of networks which have been designed to be robust. The distributions in Figs. 9 and 10 are all normalized to unity.

Suppression of superthreshold errors inside the damage shells is characteristic for all robust networks. The degree of this suppression and thus the optimization efficiency depend on the tolerance threshold  $h$  used during the optimization (Fig. 10). When tolerance is large ( $h=0.01$ ), networks inside the damage shell are typically functional even in absence of additional robustness optimization. Through the robustness optimization process, only a small fraction of exceptional networks with large errors is then eliminated. As tolerance is decreased, the effects of robustness optimization become more severe. When  $h=0.007$ , a strong abrupt decrease in the relative number of networks is seen for the errors exceeding the tolerance threshold. Moreover, a number of “failed” networks forming a weak tail extending towards relatively large errors, appears in the histogram. If the threshold is further decreased ( $h=0.003$ ), optimization eventually breaks down and dysfunctional networks with  $\epsilon > h$  dominate inside the damage shell.

Similar behavior is found for the networks that have been optimized to become robust against noise. Each particular noise realization leads to a different modification of a considered network with its own output error. Generating a set of different noise realizations, all corresponding to the same noise intensity, distributions of errors in the shell of a network can be constructed.

Figure 11(a) shows error distributions for the same network, but for different noise intensities. This network has been constructed without any optimization with respect to its robustness, by applying the evolutionary optimization algorithm aimed only at decreasing the output error. We see that the distributions are roughly Gaussian. As noise intensity  $S$  is increased, the distributions get shifted towards higher mean errors and become more broad.

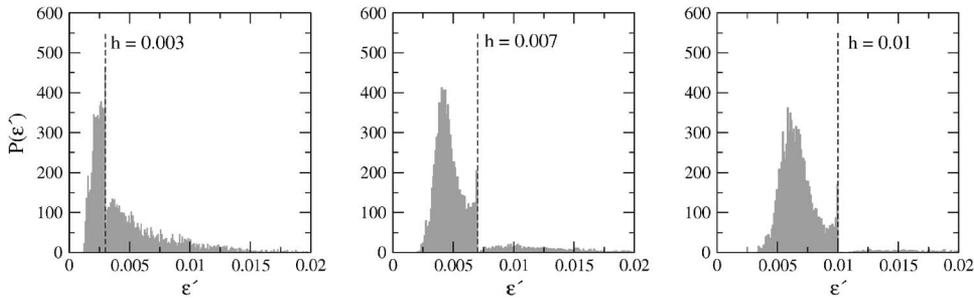


FIG. 10. Distributions of flow errors in the damage shell (removing a connection) of networks optimized with respect to link robustness, using different thresholds  $h$ .

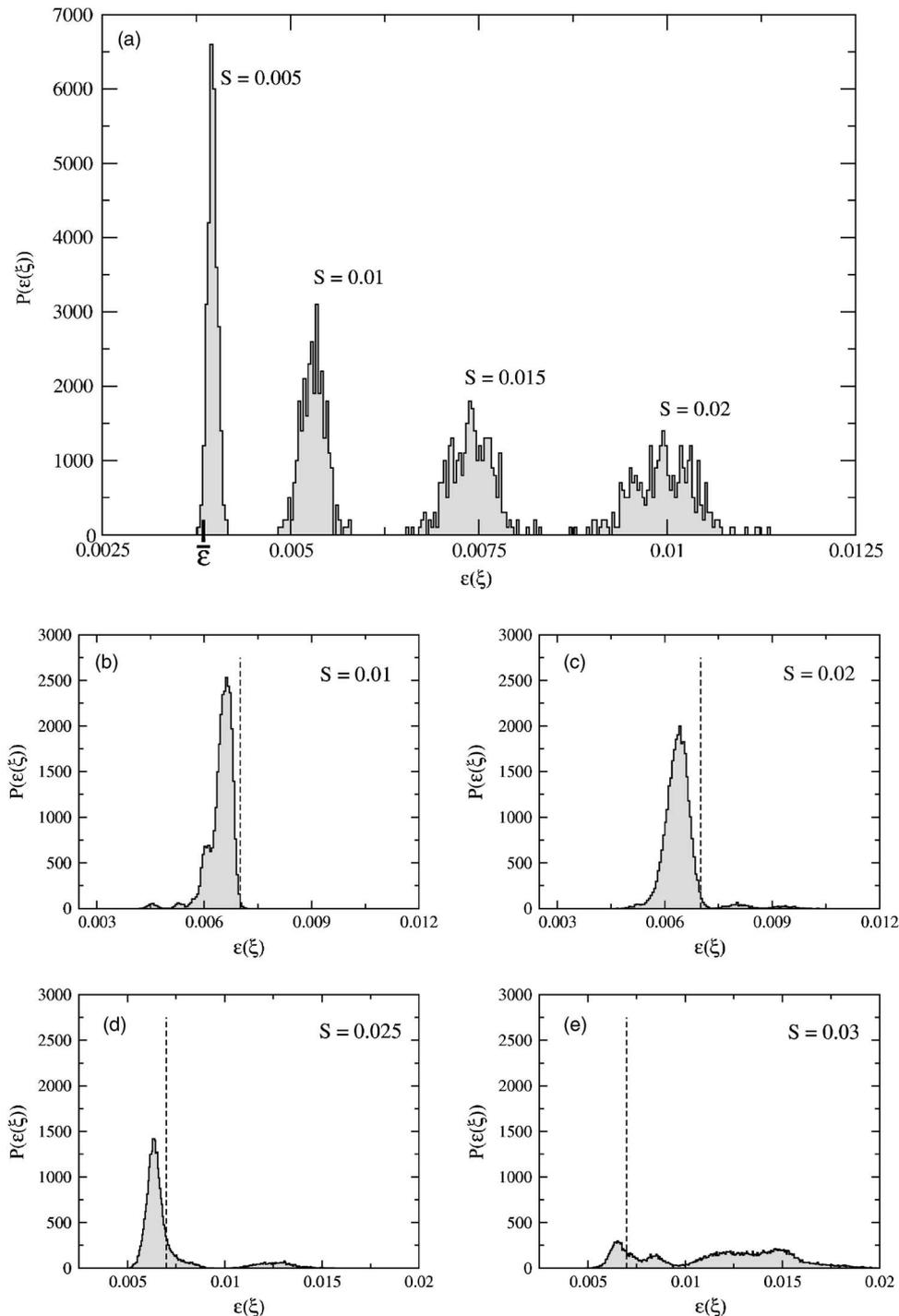


FIG. 11. (a) Distribution of flow errors for the same network, when noise of different intensity  $S$  is applied. [(b)–(e)] Flow errors of networks after application of various noise realizations for networks optimized by noise robustness using  $h=0.007$ . To compute each distribution, 300 independent noise realizations are taken. The networks have  $N=36$ , with  $N_{in}=N_{out}=8$ ,  $M=20$ , and  $K=4$ .

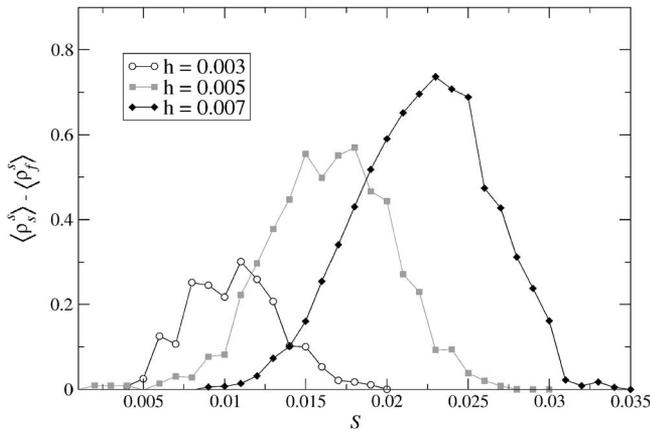


FIG. 12. Differences between the mean noise robustness of the networks optimized for noise robustness  $\langle \rho_s^s \rangle$  and that of the networks optimized only by flow error  $\langle \rho_f^s \rangle$  for different thresholds as functions of the noise intensity  $S$ .

Figures 11(b)–11(e) display analogous distributions, which have been determined for a network additionally optimized for robustness against noise of different intensities. These distributions are clearly asymmetric. The fraction of noise realizations, leading to errors exceeding the threshold  $h$ , is largely reduced. Most of the realizations have errors just below the tolerance threshold  $h$ . As the noise intensity  $S$  is increased, more noise realizations yield networks with  $\epsilon > h$  and a tail extending towards large error values is formed. At  $S=0.03$ , optimization against noise gets effectively broken down, with the majority of noise realizations having large errors.

Efficiency of optimization against noise is demonstrated in Fig. 12. Here, differences between mean noise robustness  $\langle \rho_s^s \rangle$  of the networks optimized against noise and mean noise robustness  $\langle \rho_f^s \rangle$  of the networks optimized only with respect to the output error are displayed depending on noise intensity  $S$  for three different values of the threshold  $h$ . Inside some windows of noise intensity, large gains the robustness against noise can be achieved as a result of the optimization.

Figure 13 shows distributions of noise robustness in ensembles of networks optimized against noise of different intensities. For relatively weak noise [Figs. 13(a) and 13(b)], most of the networks obtained after optimization have the (maximal possible) robustness of unity and only a few networks are less robust. For the stronger noise [Fig. 13(c)], a significant fraction of failed networks with zero robustness is present in the ensemble. At large noise [Fig. 13(d)], such failed networks constitute the majority.

Using evolutionary optimization, we have constructed three kinds of networks that are robust against the removal of nodes or links against application of noise. Cross-robustness of these networks can be considered. Does robustness against noise imply robustness against node or link deletions and vice versa?

Figure 14 displays robustness against deletion of nodes [Fig. 14(a)] and against deletion of links [Fig. 14(b)] for the ensembles of networks designed to be robust against noise. Generally, networks robust against noise are *not* robust against noise or link deletions. Additionally, curves in Fig. 15 display mean robustness against noise for the networks optimized to be robust against noise ( $\rho_s^s$ ), against node deletions ( $\rho_n^s$ ), against link deletions ( $\rho_l^s$ ), and for the ensemble of net-

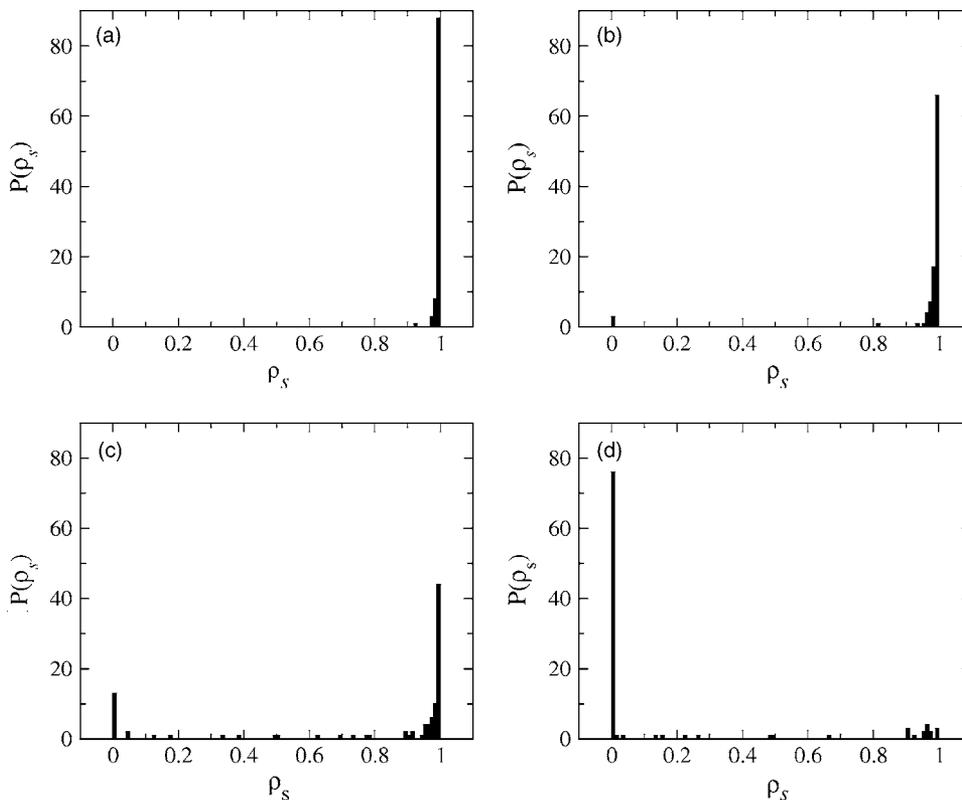


FIG. 13. Histograms of noise robustness for several ensembles of networks optimized against noise ( $h = 0.007$ ). Noise intensities  $S$  are (a) 0.01, (b) 0.02, (c) 0.025, and (d) 0.03.

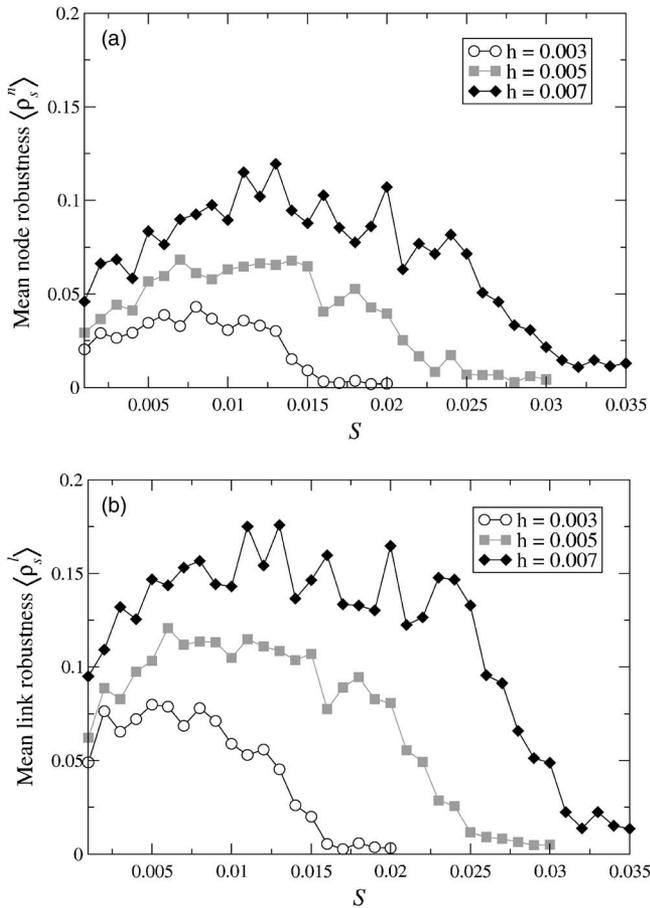


FIG. 14. (a) Mean node  $\langle \rho_s^n \rangle$  and (b) mean link  $\langle \rho_s^l \rangle$  robustness of networks optimized for noise robustness, as functions of noise intensity.

works that are merely functional ( $\rho_f^z$ ). Remarkably, the networks designed to be robust against link or node deletions have robustness against noise which is even lower than just for an ensemble of functional networks, without any robustness optimization. Thus, *robustness against noise is not compatible with robustness against local damage* in the considered network model.

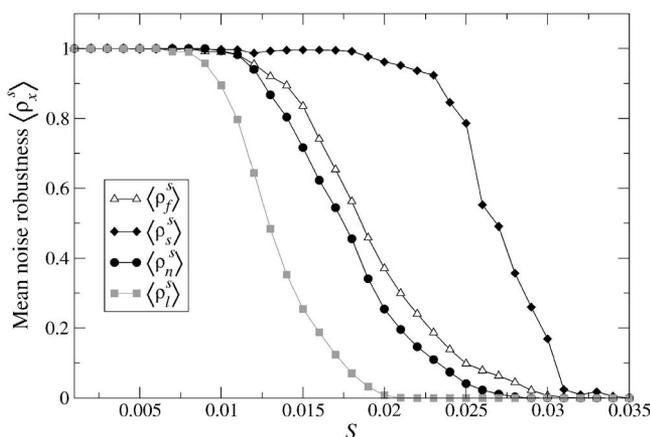


FIG. 15. Mean noise robustness  $\langle \rho_x^z \rangle$  as function of noise intensity  $S$  for networks that have been optimized using different robustness criteria ( $z = f, s, n, l$ ).

### VII. STRUCTURAL PROPERTIES OF ROBUST NETWORKS

Principal structural properties of networks, robust against different kinds of damage and noise, have been determined and analyzed. Figure 16(a) presents average mean degrees of the networks, optimized against deletion of nodes or links, in comparison to the same data for merely functional networks with the error below the tolerance threshold. As always, each data point is an average over 100 different networks obtained by running a separate evolution. The mean degrees (i.e., mean numbers of connections per a node) are strongly increased in the networks optimized for link robustness. They are decreased in the networks optimized for node robustness, but this decrease is however small and does not significantly depend on the tolerance threshold. Clustering coefficients of robust networks have been further determined [Fig. 16(b)]. They are higher for networks robust against link deletions and are decreased in the networks robust against the deletion of nodes.

An important structural property of considered networks is the average path length (APL) between input and output nodes. For each input node, shortest paths, connecting it to each of the output nodes, are determined. The APL of a network is the average length of all such paths. The mean APL of an ensemble is obtained by additional averaging over all its networks. Figure 16(c) shows mean APL for ensembles of networks optimized using different criteria and tolerance thresholds. Networks, optimized for link robustness, are characterized by longer average path lengths, as compared to the reference ensemble of merely functional networks and to the ensemble of networks optimized for node robustness.

When flux is applied to an input node, it goes only through a subset of middle nodes until it reaches the outputs. The average size of an activated subnetwork is another important property of such systems. It is defined as the mean number of middle nodes in subnetworks corresponding to activations of different input nodes. Figure 16(c) displays average subnetwork sizes for different robust networks and for the reference ensemble. In agreement with increases in connectivity and clustering coefficient, networks robust against link deletions have subnetworks larger than those robust against node deletions or functional reference networks. Since the networks have  $M=20$  middle nodes, about half of them become activated to produce one response at  $h=0.07$ . This indicates that *strong overlaps* between activated subnetworks must be present. In the networks optimized against node deletions, the average size of activated subnetworks is about twice smaller. Because we have  $N_{in}=8$  input nodes, some overlaps between these subnetworks are, however, present even in this case.

When networks are optimized against noise, their properties may depend both on the tolerance threshold and on the intensity of noise used during the optimization. Figure 17(a) shows average mean degrees of various designed networks depending on the noise intensity. As noise gets stronger, the mean degree (and thus the total number of connections) tend to decrease. This can be expected because the effects of variation of connection weights should become weaker when

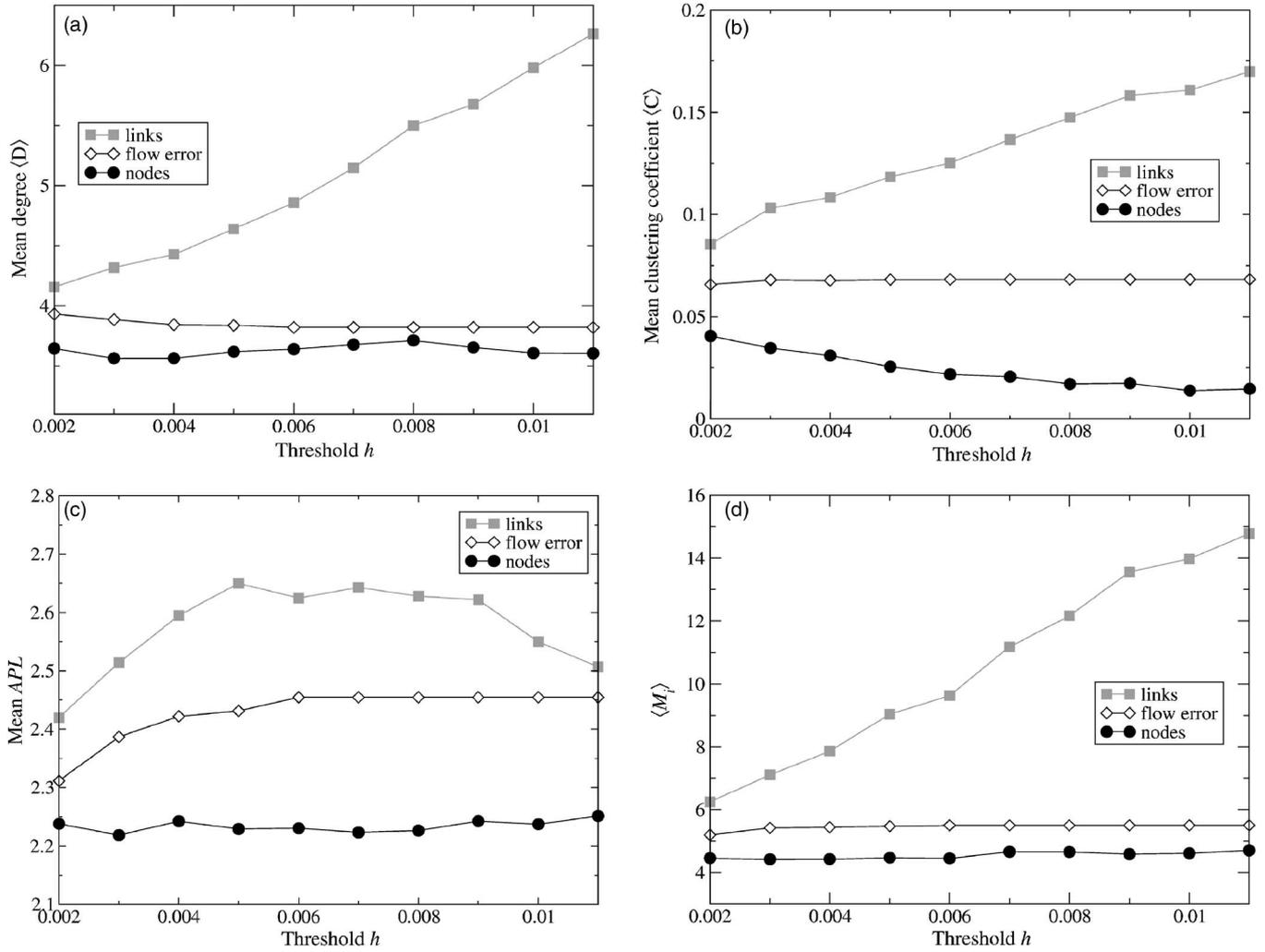


FIG. 16. Average mean degree (a), clustering coefficient (b), mean path length (c), and mean size of activated subnetworks (d) as functions of the threshold for the ensembles of networks optimized for robustness against node or link removals and only to reduce the flow error. Statistical averages are taken over ensembles of 100 designed networks at each data point.

the total number of connections is higher. The average clustering coefficient also decreases for the stronger noise, but this decrease is less pronounced [Fig. 17(b)]. Mean numbers of middle nodes in the subnetworks activated by a single input decrease as  $S$  grows [Fig. 17(c)].

**VIII. MOTIF DISTRIBUTIONS OF ROBUST NETWORKS**

To determine motif distributions of designed networks, the software package MFINDER has been used.<sup>11</sup> For each network  $G_l$ , this software determines its motif distributions with respect to motifs of different sizes. In the case of three-node motifs, the vector of normalized scores  $\bar{z}_l$  has 13 elements. To calculate this vector, the program generates an ensemble of 1000 networks, each being a different randomized version of the network  $G_l$ . Normalized scores of a network are defined by Eq. (2). For an ensemble of  $\mathcal{N}$  networks, we compute its average normalized score as

$$\langle \bar{z} \rangle = \frac{1}{\mathcal{N}} \sum_{l=1}^{\mathcal{N}} \bar{z}_l \tag{12}$$

and define its standard deviation vector as

$$\sigma = \sqrt{\frac{1}{\mathcal{N}} \sum_{l=1}^{\mathcal{N}} (\mathbf{z}_l - \langle \mathbf{Z} \rangle)^2}. \tag{13}$$

Thus defined, vectors  $\langle \mathbf{Z} \rangle$  for different network ensembles cannot, however, be compared because they are not normalized to unity. Therefore, we perform further normalization and introduce the mean normalized score vector  $\langle \mathbf{Z} \rangle$  of an ensemble with the elements

$$\langle \mathbf{Z} \rangle_\alpha = \frac{\langle \bar{z} \rangle_\alpha}{\sqrt{\sum_{\beta=1}^{13} \langle \bar{z} \rangle_\beta^2}}. \tag{14}$$

Figure 18 shows normalized mean motif distributions and their statistical dispersions for two ensembles of networks with different numbers  $M$  of middle nodes. These networks are constructed by optimization aimed only at reducing the output error. Statistical dispersion of data is high, indicating that networks with largely different motif distributions are present in these ensembles.

Mean motif distributions of the networks against node deletions, for different tolerance thresholds  $h$  used during the optimization process, are shown in Fig. 19. When tolerance

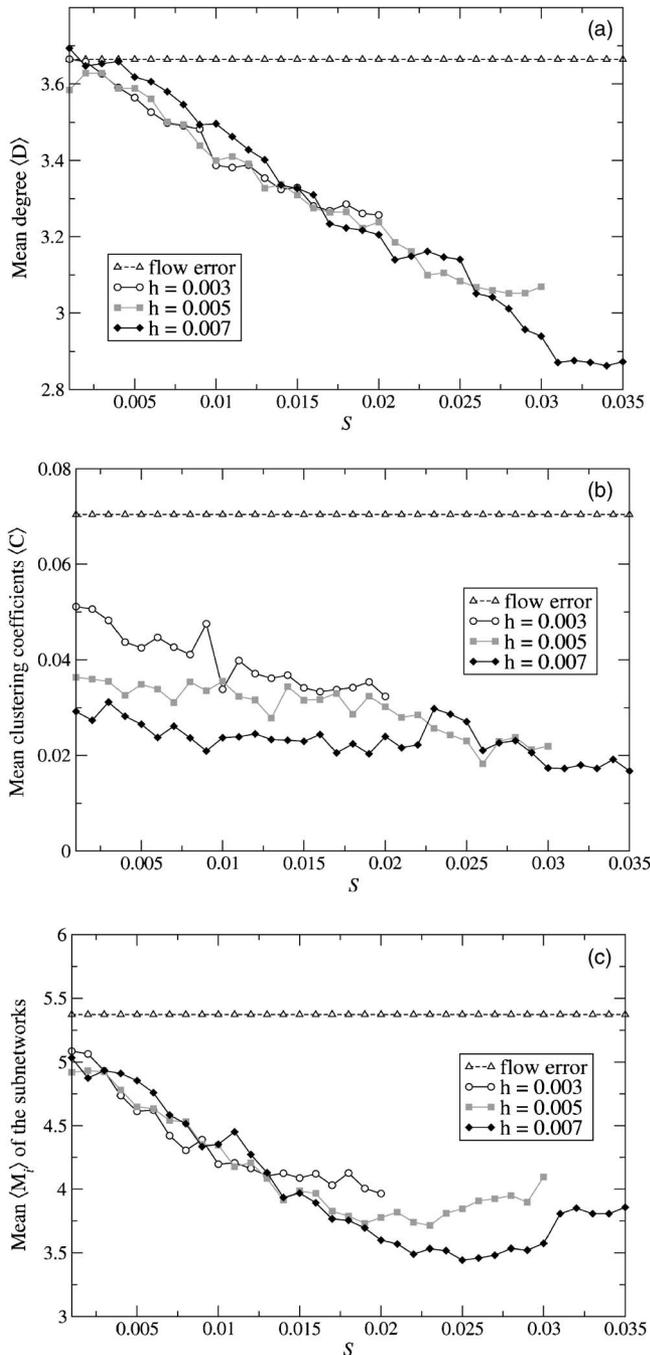


FIG. 17. Average mean degree (a), clustering coefficient (b), and mean size of activated subnetworks (c) as functions of noise intensity for ensembles of networks that have been optimized only for flow errors (empty triangles) and for noise robustness using different thresholds  $h$ . Statistical averaging over an ensemble of 100 designed networks at each data point.

is low ( $h=0.003$ ), large statistical dispersion of the data is still seen. The dispersion decreases for  $h=0.007$  and  $h=0.011$  and motif distributions are very close in these last two cases. Figure 19 (right) displays analogous data for the networks robust against link deletions.

Figure 20 displays motif distributions of the networks robust against noise of various intensities, for two different tolerance thresholds. While some dependence on the values of  $S$  and  $h$  can be noticed, the same characteristic profile of motif distributions is apparent in all cases. Statistical disper-

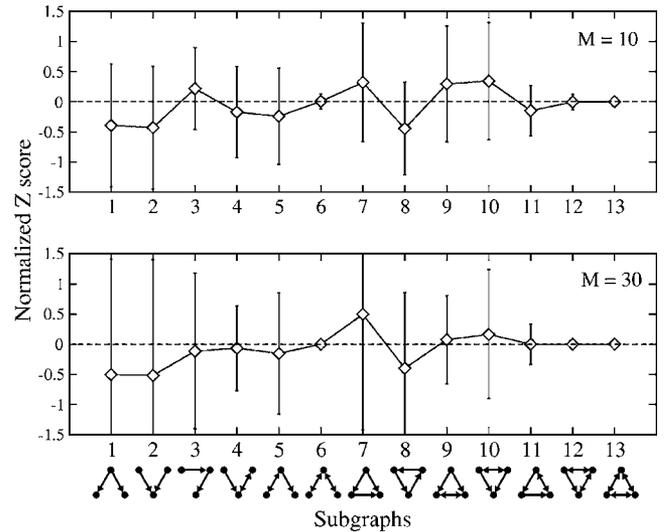


FIG. 18. Mean motif distributions of designed networks with different numbers  $M$  of middle nodes. Each ensemble contains 100 networks, obtained by running evolution with a different, randomly chosen initial condition and a different, randomly chosen target pattern. The parameters  $K=4$ ,  $N_{in}=N_{out}=8$  are the same for all networks. Bars show statistical dispersion of data.

sion of the data is, however, substantial, particularly for the networks designed to be robust at stronger noises and with the lower tolerance thresholds.

To compare motif distributions of networks with different robustness properties, we show in Fig. 21 mean normalized motif distributions for the networks designed to be robust against node or link deletions and against the introduction of noise. The same tolerance threshold  $h=0.007$  is used in the construction of all these networks. The networks have the same total sizes and the same numbers of input and output nodes. As always, the same kinds of target output patterns have been employed in the evolutionary construction. Thus, networks in all three ensembles are characterized by the same functions and they are only different with respect to their robustness properties. It is clearly seen that motif distributions of the networks robust against link deletions are strongly different from motif distributions of the networks robust against node deletions or against noise.

### IX. DISCUSSION AND CONCLUSIONS

This study, which has been performed for a simple mathematical model, allows us to make several important conclusions concerning the properties of robust signal processing networks and their motif distributions. First of all, we have clearly demonstrated that one cannot generally talk about “robust” networks, without specifying against what kind of perturbations they are robust.

All constructed networks had the same kind of output functions and differed only because they have been optimized to be robust against node or link deletions, or against noise representing weak distributed perturbation affecting all network connections. The considered model was linear and nondynamical, so that the effects of noise suppression through nonlinear feedbacks have been excluded. Therefore, robustness could develop there only through the emergence

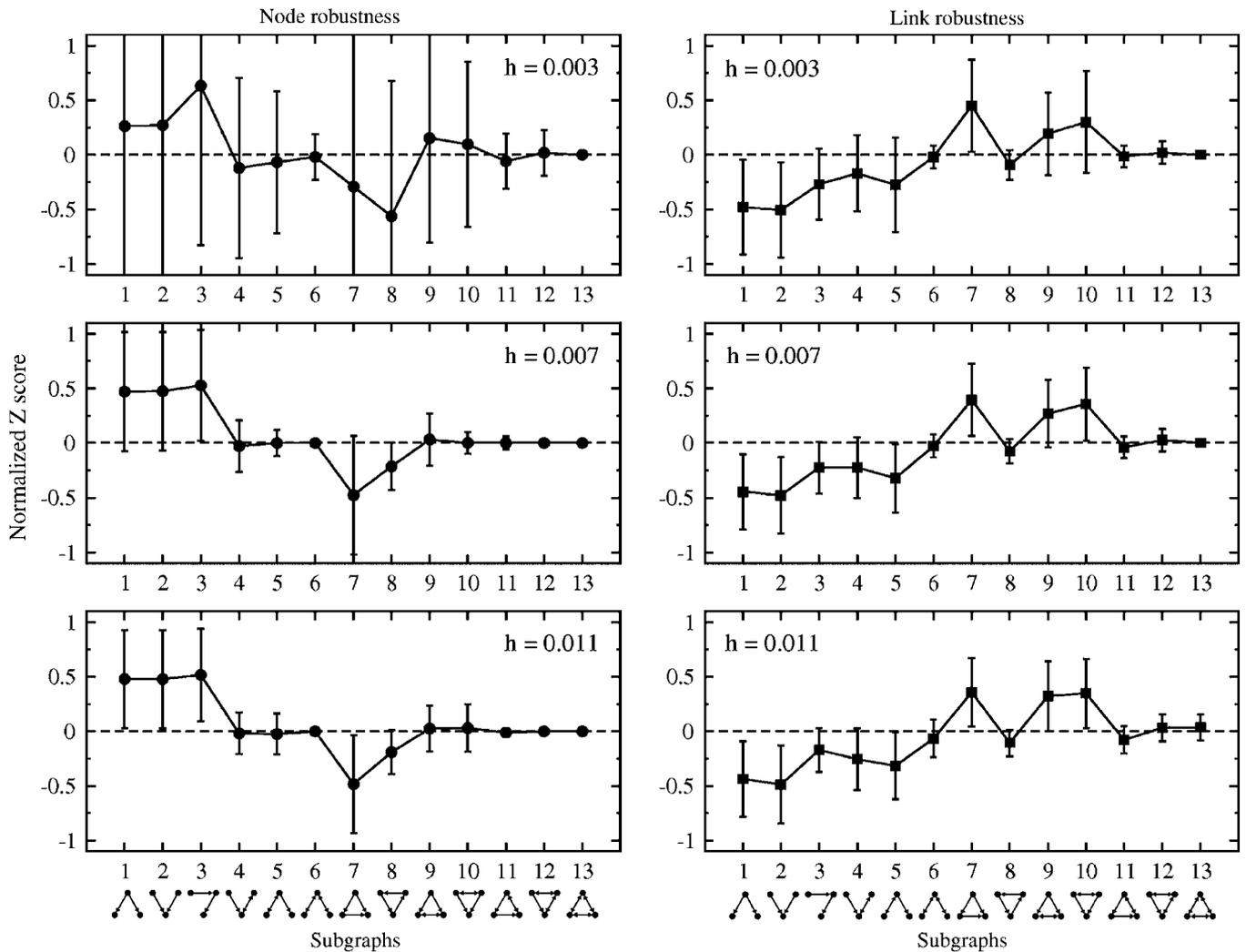


FIG. 19. Motif distributions of networks robust against removal of nodes (left, circles) and links (right, squares). Optimizations have been performed using different thresholds. Bars show statistical dispersion of data.

of special *self-correcting* network architectures, which allow to retain required functions despite local or distributed damage. Because the total number of middle processing nodes was restricted in our investigations, robustness in responses could not be achieved through a trivial parallelization of structural blocks responsible for individual outputs. Each middle processing node was typically involved in generation of several different responses, so that the network was highly interwoven. There is no rational procedure to design such self-correcting networks; in our investigations they have been constructed by running an evolutionary optimization process, similar to the method of simulated annealing in the problems of complex optimization.

By running a large number of independent evolutions for each parameter choice, extensive datasets for robust networks have been generated. This has allowed us to determine characteristic statistical properties of different robust networks with the same functions and to analyze their cross-performance. We have shown that the networks robust against different kinds of damage have different structural properties, such as the mean degree, the clustering coefficient

or the average length of a path leading to the response generation.

Using such data, we could proceed further and determine motif distributions for ensembles of constructed networks with different kinds of robustness. Previously, motif distributions were mostly analyzed for naturally occurring networks. It was found that all such networks can be divided with respect to their motif distributions into four principal superfamilies.<sup>4</sup> The origins of such superfamilies remained unclear, but it was conjectured that they should rather lie in the differences in *functions* of the respective networks. In contrast to this, we have found that our model networks with the same output functions have very different distributions, depending on the kind of local damage or noise against which they have been optimized.

Analyzing motif distributions for an ensemble of networks that have been constructed to have only the prescribed output functions, without any optimization for robustness, we could see that this ensemble is characterized by broad statistical dispersion of motifs, indicating that networks with very different motif properties are present there. On the other

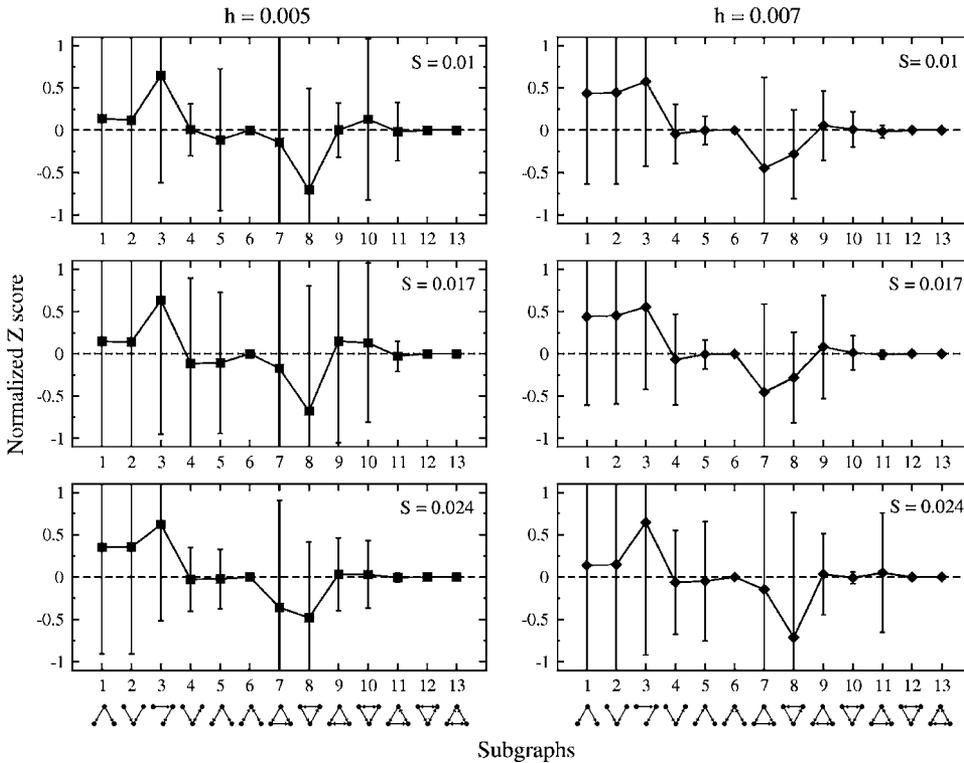


FIG. 20. Motif distributions of networks optimized for noise robustness, using different thresholds and noise intensities. Bars show statistical dispersion of data. All networks have  $N_{in} = N_{out} = 8$  and the number of middle nodes during the optimization is  $M = 20$ . Target output patterns have  $K = 4$ . All evolutions have been terminated after  $10^5$  iterations.

hand, when optimization against a particular kind of damage has been performed, network ensembles with well-defined motif distributions have always been obtained.

Thus, we conclude that, at least for the considered model of signal processing, motif distributions are determined by the specific *robustness* of self-correcting networks, rather than by their particular functions.

Remarkably, characteristic motif distribution of all constructed robust networks still belonged only to the previously known motif superfamilies. This provides further evidence that motif superfamilies, discovered by Alon and co-workers,<sup>4</sup> play a fundamental role in network classification. The networks robust against link deletions are in the second superfamily, whereas the networks robust against node deletions belong to the fourth superfamily.

Our study has been motivated by the problem of robust signal processing in biological organisms. Instead of investi-

gating real biological networks, we have used a constructive approach and generated various robust networks within a toy flow distribution model of signal processing. This model was not intended to describe any particular kind of real biological signal processing systems. In some of its properties, it was in fact very different from such real biological systems. It was linear, without any logical gates or threshold elements, and had no dynamics. Moreover, the total applied signal flux was conserved in this model, which is not typical for a biological system. Therefore, it is even more interesting to compare motif distributions of generated robust networks with those of the actual biological systems.

Figure 22 shows motif distributions of our constructed robust networks, together with those of a few characteristic biological and linguistic networks in the second and the fourth superfamilies. Surprisingly, we find that the motif distribution of the networks robust against link deletions *almost exactly coincides* with the respective distributions for the signal transduction networks and the genetic developmental network in the fruit fly and is very close to the motif distribution characteristic for the neural system of the primitive animal *C. elegans*. To the contrary, motif distributions of networks robust against node deletions are close to those of the linguistic networks.

Based on this discovery, we have suggested<sup>6</sup> that the principal role in determining motif distributions in the second superfamily is played by the condition of robustness against link removals imposed by biological evolution. Signal transduction networks and genetic developmental networks of multicellular organisms may have evolved to become robust against random disruption of interactions between proteins, rather than the removal of entire proteins (nodes) from these networks. In this respect, it should be

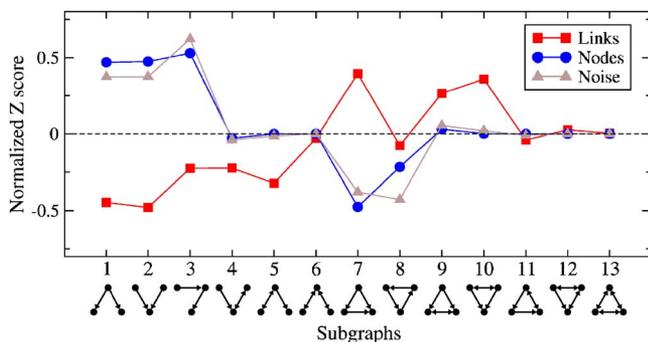


FIG. 21. (Color online) Motif distributions of the networks selected to be robust against the application of noise (triangles), the removal of individual links (squares) or nodes (circles). The noise intensity is  $S = 0.02$  and the tolerance threshold is  $h = 0.007$ .

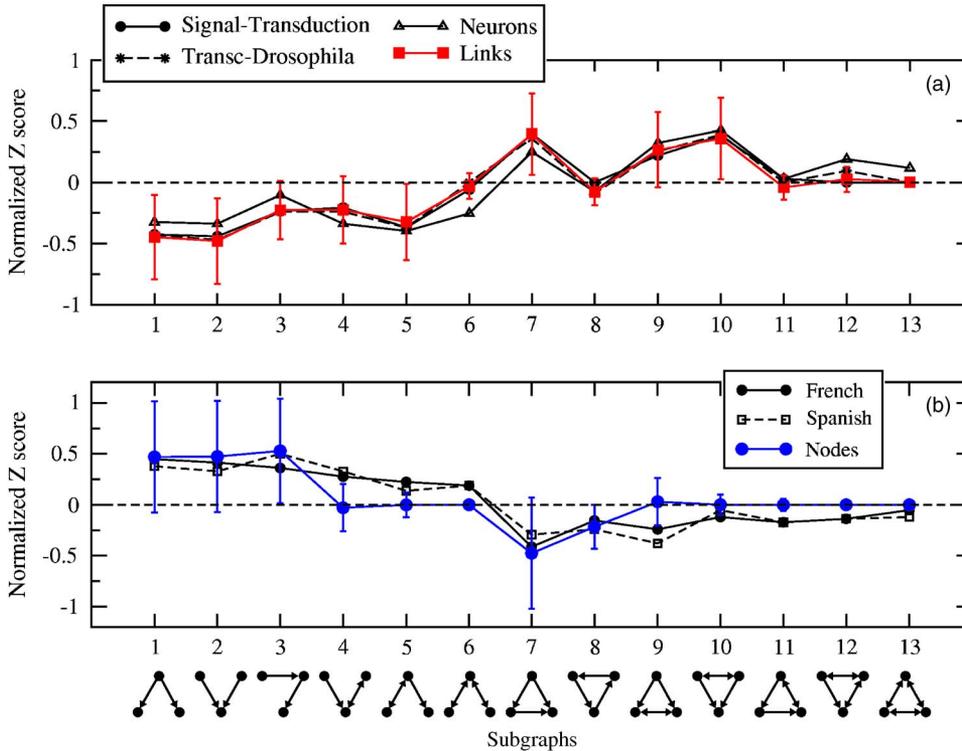


FIG. 22. (Color online) Motif distributions of networks robust against removal of (a) (squares) links and (b) (circles) nodes. For each network type, an ensemble average over a subset of functional networks with the error less than  $h=0.007$  in the ensemble of 100 independently evolved networks with different, randomly generated, output patterns is performed. Bars show statistical dispersion of data. For comparison, the data (Ref. 4) for (a) signal transduction networks, developmental genetic networks in drosophila, neural system of *C. elegans*, and (b) two linguistic networks are included (black curves).

noted that a genetic point mutation modifies only one amino acid in the protein chain and this would typically lead only to a minor change in the folded protein conformation. Therefore, it can be expected that, after such a mutation, a protein would possibly lose only some of its interactions with other proteins in the network, corresponding to breaking of individual links in the model. Moreover, the neural system of *C. elegans* may have evolved to become robust against breaking of synaptic connections, rather than against the death of whole neural cells.

It may also be that robustness against nodal deletions is the principal property of the networks in the second superfamily. Indeed, All languages are highly redundant and, if you have forgotten a word, there are many other ways to express the same meaning. Further investigations are needed to confirm or to reject this conjecture.

In addition to random local damage (i.e., due to mutations), all biological systems are also robust against internal or environmental noise. Because our model is static, we could not study effects of time-dependent fluctuations in its framework. Nonetheless, we could still consider effects of quenched noise, representing weak random and independent modifications of all network connections. It is interesting that our networks, designed to be robust against such noise, belong to the second superfamily, so that their motif distributions are strongly different from those of the actual biological signal processing networks.

It seems that network architectures providing self-correction against link removals are *not compatible* with self-correction against the effects of noise. However, biological networks must be still robust against both kinds of perturbations. The solution, found by biological evolution, may be that principally different mechanisms are employed in the

organisms to achieve robustness against link deletions (caused by mutations) and against noise. Robustness against local link damage can be ensured at the structural network level, i.e., through the use of special self-correcting architectures, as demonstrated in our study. On the other hand, fluctuations induced by noise can be suppressed dynamically, due to the presence of multiple negative feedback loops in a system. The latter effects could not, however, be investigated in framework of the considered model.

There is also another aspect related to the networks which have been designed by us to be robust against noise. Their motif distributions are in the fourth superfamily and they are similar to the motif distributions of the networks robust against deletion of nodes (see Fig. 21). Our analysis of cross-robustness of different kinds of networks (see Figs. 14 and 15) indicates, however, that the networks robust against nodal deletions are not typically robust against noise and vice versa. This implies that, within the second superfamily, network architectures with different self-correction properties can still be present. Thus, motif distributions do not yet provide a tool to distinguish robustness against noise and against some local damage.

Our reported research on the design and analysis of robust functional networks is a first step, preceding a series of further investigations. Networks with dynamical responses can also be constructed and analyzed. Moreover, similar evolutionary constructions can be performed for the nonlinear models, which are employed to describe neural systems or genetic systems and signal transduction networks. The spectacular agreement between motif distributions of real biological information processing networks and model networks, optimized to be robust against link removals, cannot be explained by purely qualitative arguments. It may indicate

the existence of a general statistical *universality class* of such networks. Perhaps, it can be here mentioned that other broad universality classes, formed by, e.g., percolation networks, are known in statistical physics.

Although the focus in these studies has been on biological signal processing, the results are more general. Similar models can be, for example, applied to describe industrial logistic networks where a set of different goods should be transported, in prescribed fractions, to a variety of destinations. We have effectively shown how such functional networks can be designed and made robust against different kinds of damage by running an artificial evolution process.

## ACKNOWLEDGMENTS

Financial support of the Volkswagen Foundation (Germany) through the program “Networks as Phenomena across Disciplines” is gratefully acknowledged.

<sup>1</sup>M. A. Savageau, *Nature (London)* **229**, 542 (1971).

<sup>2</sup>D. Deutscher, I. Meilijson, M. Kupiec, and E. Ruppin, *Nat. Genet.* **38**, 9 (2006).

<sup>3</sup>N. Wiener, *Cybernetics: Or the Control and Communication in the Animal and the Machine* (MIT Press, Cambridge, MA, 1948).

<sup>4</sup>R. Milo, S. Itzikovitz, N. Kashtan, D. Chklovskii, and U. Alon, *Science* **298**, 824 (2004).

<sup>5</sup>J. G. White, E. Sothgate, J. N. Thomson, and S. Brenner, *Philos. Trans. R. Soc. London, Ser. B* **314**, 1 (1986).

<sup>6</sup>P. Kaluza, M. Ipsen, M. Vingron, and A. S. Mikhailov, *Phys. Rev. E* **75**, 015101 (2007).

<sup>7</sup>P. Kaluza and A. S. Mikhailov, *Europhys. Lett.* **79**, 48001 (2007).

<sup>8</sup>P. Kaluza, H. Kori, and A. S. Mikhailov, in *Managing Complexity*, edited by D. Helbing (Springer, Berlin, 2007), p. 359.

<sup>9</sup>S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, *Science* **220**, 671 (1983).

<sup>10</sup>P. Kaluza, Doctoral thesis, Department of Physics, Technical University, Berlin, 2007, <http://opus.kobv.de/tuberlin/volltexte/2007/1707>.

<sup>11</sup>Software package *MFinder* is available at [www.weizmann.ac.il/mcb/UriAlon](http://www.weizmann.ac.il/mcb/UriAlon).