Exploring the assortativity-clustering space of a network's degree sequence

Petter Holme
Department of Computer Science, University of New Mexico, Albuquerque, New Mexico 87131, USA

Jing Zhao
School of Life Sciences & Technology, Shanghai Jiao Tong University, Shanghai 200240, China; Shanghai Center for Bioinformation and Technology, Shanghai 200235, China; and Department of Mathematics, Logistical Engineering University, Chongqing 400016, China

(Received 6 November 2006; published 19 April 2007)

Nowadays there is a multitude of measures designed to capture different aspects of network structure. To be able to say if a measured value is expected or not, one needs to compare it with a reference model (null model). One frequently used null model is the ensemble of graphs with the same set of degrees as the original network. Here, we argue that this ensemble can give more information about the original network than effective values of network structural quantities. By mapping out this ensemble in the space of some low-level network structure—in our case, those measured by the assortativity and clustering coefficients—one can, for example, study where in the valid region of the parameter space the observed networks are. Such analysis suggests which quantities (or combination of quantities) are actively optimized during the evolution of the network. We use four very different biological networks to exemplify our method. Among other things, we find that high clustering might be a force in the evolution of protein interaction networks. We also find that all four networks are conspicuously robust to both random errors and targeted attacks.

DOI: 10.1103/PhysRevE.75.046111 PACS number(s): 89.75.Fb, 82.39.Rt, 89.75.Hc

I. INTRODUCTION

Network structure [1–3] is usually defined as the way a network differs from what is expected. What "expected" means depends on the fundamental constraints on the network, and this can vary from system to system. For example, if the network is made of units that must be connected to two, and only two, others; then, it is not interesting whether or not a vertex lies on a cycle (we already know that it will). The ensemble of all networks fulfilling the fundamental constraints on the system is usually called *null model* (or *reference model*). When we have pinned down the null model we can measure the network structure by standard quantities. If the values of these quantities differs significantly from the null-model average, then we call the network structured. The baseline assumption of complex network theory is that network structure carries information about the forces that have formed the network. Ever since the studies of Barabási and co-workers [1,4], the degree distribution (or, if referring to the set of degrees of one particular network, *degree sequence*) has been regarded as the most fundamental network structure. For many networks, the degrees are related to outer factors (not emerging from the network evolution). In such cases the ensemble of all graphs with the same degree sequence as the original network is a natural null model. Another interpretation is that the network structures measured relative to this null model are of higher order than the degree—i.e., what remain after the effects of the more fundamental structure (the degree sequence) is filtered away. The usual way to use a null model is to compare a network measure with the ensemble average value of the null model. In this paper we will argue that one can glean more information about the original network by studying the null model ensemble in greater detail than just measuring averages. The particular null model we use in this paper is the above-mentioned—random graphs conditional to the same set of degrees as the original network—but this can lead straightforwardly to other more, or less, constrained null models.

We consider networks that can be modeled as a graph $G=(V,E)$, where V is the set of N vertices and E is the set of M undirected edges. We denote the ensemble of graphs with the same degree sequence as G as $\mathcal{G}(G)$. Our basic approach to study $\mathcal{G}(G)$ is to resolve its members in the space of higher order network structures. The two such higher order network structures we consider in this paper are the correlation between the degrees at either side of an edge (measured by the *assortative mixing coefficient*, r [5], or simply *assortativity*) and the fraction of triangles in the network (measured by the *clustering coefficient*, C [3,6]). By mapping out $\mathcal{G}(G)$ in the space defined by r and C one can pose questions such as the following: How large is the region in r-C space where members of $\mathcal{G}(G)$ actually exist? (This helps us answer how constrained the network evolution is if the degrees are given.) Is the real network close to $\mathcal{G}(G)$'s boundaries in r-C space? (Which would indicate whether or not r or C are actively optimized.)

The basis for our exploration of an ensemble $\mathcal{G}(G)$ is to map out its members in the space defined by some network-structural measures, in our case the assortativity and clustering. We measure the relative size of the largest connected component, the average distances within the largest connected component, the error and attack robustness for all our test networks. We explore the r-C space by successively rewire pairs of edges, (i,j) and (i',j') to (i,j') and (i',j), that takes the system in a desired direction. Rewiring techniques for studying networks are half a century old [7]. In the physics literature these techniques were first used in Refs. [8,9].

II. NETWORK STRUCTURAL MEASURES

One fundamental network structure is *assortativity*—the correlation between the degrees at either side of an edge. A simple way of measuring this structure is by the assortativity [3] r. If one use an edge list representation internally [i.e., let the edges be stored in an array of ordered pairs $(i_1,j_1),\ldots,(i_M,j_M)$] then [5]

$$r=\frac{4\langle k_1k_2\rangle-\langle k_1+k_2\rangle^2}{2\langle k_1^2+k_2^2\rangle-\langle k_1+k_2\rangle^2}, \tag{1}$$

where, for an edge (i,j), k_1 is the degree of first argument (i.e., the degree of i) and k_2 is the degree of the second argument. The range of r is $[-1,1]$ where negative values indicate a preference for high connected vertices to attach to low-degree vertices, and positive values means that vertices tend to be attached to others with degrees of similar magnitudes.

For some classes of real-world networks there is a strong tendency for triangles (fully connected subgraphs of three vertices) to form in the network. The network measure of the density of triangles is called *clustering coefficient* and is commonly measured by [6]

$$C=3n_{\text{triangle}}/n_{\text{triple}}, \tag{2}$$

where n_{triangle} is the number of triangles and n_{triple} is the number of connected triples (subgraphs consisting of three vertices and two or three edges). The factor three is included to normalize the quantity to the interval [0,1].

Two quantities that are, perhaps more than any other, related to the functionality of dynamic processes on the network are the relative size of the largest component (connected subgraph) s, and the average distance $\langle d\rangle$. s is simply defined as the number of vertices in the largest component divided by N. The distance $d(i,j)$ between two vertices i and j is defined as the number of edges in the shortest path between these two vertices. $\langle d\rangle$ is $d(i,j)$ averaged over all vertex pairs $(i\neq j)$ in the largest component. In a network with large s and small $\langle d\rangle$, spreading processes will be fast and far-reaching. This is a good property of information networks, but bad in the context of, for example, disease spreading. For most purposes, we believe, valuable information gets lost in such a combination (a fragmented network G with short average distances can be something very different from a connected graph of large distances and the same average reciprocal distances as G).

One line of complex-network research is the study of the response of the network to attacks, errors, failures and other events that effectively change the structure. The error response problem is usually formulated as follows: How does the functionality of the network change if a random fraction of the vertices, or edges, is removed [3]? The attack problem is the same, except that the vertices are not selected randomly but according to some strategy intended to decrease the networks' functionality as rapidly as possible [10,11]. A frequently used metric for functionality is the ratio of s before and after the event [10–12]. In the error and attack robustness problems, this quantity is typically plotted as a function of the number of removed vertices. Since we aim at mapping out the r-C space of degree sequences, we would like to capture the robustness with just one number. We will use what we call the *f-robustness* R_f of a network as the expected fraction of vertices that needs to be removed for the relative size of the largest component to decrease to a fraction $f\in(0,1)$ of its original value. The way of removal can either be random (the error problem) or selective (the attack problem). For the rest of the paper we will set $f=1/2$, and refer to the $1/2$-robustness just as "robustness" R. Other f values give slightly different results, but our conclusions will hold for a range of intermediate f values.

III. THE ANALYSIS SCHEME

The fundamental idea of our method is simple: we update the network by choosing pairs of edges randomly, say (i,j) and (i',j'), and swap one end of them [forming (i,j') and (i',j)]. This guarantees that the degree sequence stays intact. We navigate in the r-C space by only accepting changes that move us in the desired direction. If an edge-swap would introduce a self-edge (i.e., if $i=j'$ or $i'=j$) or a multiple edge [i.e., if (i,j') or (i',j) belongs to E before the swapping, or *move*] it is not performed. There are many other technicalities concerning the convergence to extremes, uniformity of the sampling and more that we discuss in the Appendix.

The members of the ensemble $\mathcal{G}(G)$ do not, in general, cover the whole range of (r,C) values. Indeed, for any finite G, $\mathcal{G}(G)$ defines a set of points, rather than a continuous region, in the r-C space. We will perform a more coarse-grained analysis breaking down the r-C space into pixels and average quantities over the graphs of $\mathcal{G}(G)$ with (r,C) values within the pixel. (Thus a pixel constitutes a graph ensemble in itself, our aim is to sample its members with uniform randomness.) For a computationally tractable resolution, the pixels containing members of $\mathcal{G}(G)$ typically form contiguous regions. We will refer to the pixels that contain a member of $\mathcal{G}(G)$ as *valid pixels*, and all pixels that are valid or between valid pixels the *valid region* of $\mathcal{G}(G)$.

To trace the valid region of $\mathcal{G}(G)$ we start by finding the lowest and highest assortativity value, $r_{\min}$ and $r_{\max}$, respectively. Briefly speaking, to find $r_{\min}$ we rewire edge pairs that lower r (and vice versa for $r_{\max}$). After finding the extremal r values, we splice the region between these into L segments. Then we go through the region and for each region $n\in[1,L]$ we find the minimal and maximal C values, $C_{\min}(n)$ and $C_{\max}(n)$. The region in C-space between the lowest $C_{\min}=\min_{1\leq n\leq L}C_{\min}(n)$ and highest $C_{\max}=\max_{1\leq n\leq L}C_{\max}(n)$ observed clustering coefficient is segmented into L regions. [Note that $C_{\min}$, without argument, is the global clustering minimum, whereas $C_{\min}(n)$ is the minimum conditioned on r being in the nth segment.] Thus we (assuming our method works) obtain an $L\times L$ grid of the r-C space that contains the valid region of $\mathcal{G}(G)$. The method is illustrated in Fig. 1 and described in detail in Appendix A.

IV. NETWORKS

Our method can be applied to every kind of system that can be modeled as an undirected network. To limit ourselves,

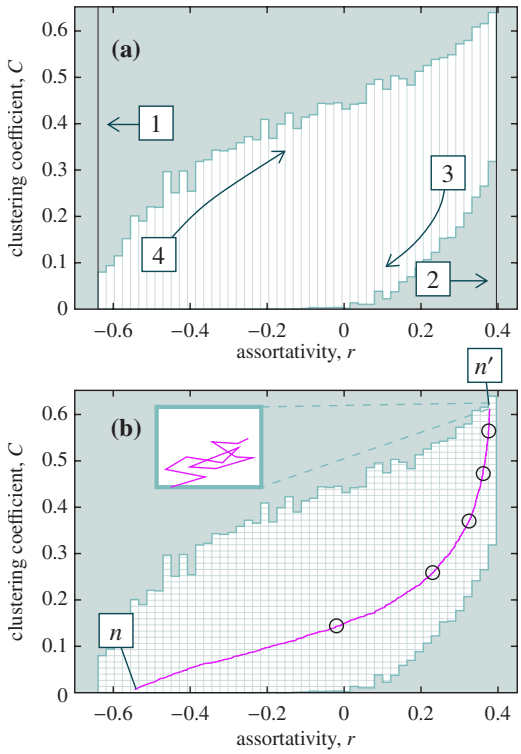

FIG. 1. (Color online) Illustration of the analysis scheme applied to the *C. elegans* neural network. Panel (a) shows how the valid region is mapped out: (1) $r_{\min}$ is located. (2) $r_{\max}$ is found and the interval $[r_{\min}, r_{\max}]$ is divided into L segments. (3) $C_{\min}(n)$ is constructed. (4) $C_{\max}(n)$ is traced and the interval $[C_{\min}, C_{\max}]$ is segmented into L regions. Panel (b) illustrates the sampling of the pixels. The next pixel to go to is chosen from a random permutation of the pixels. In this example n and n' are chosen to be far apart. The line shows the path taken by the algorithm. The circles indicate every thousandth step on the way from n to n'. The blowup illustrates the random walk within a pixel to sample the graphs of the pixel more randomly.

we use four networks from biology as examples in this paper. These networks are, nonetheless, representing fundamentally different systems.

Cancer is a disease that occurs due to changes in the genome. One important process causing such changes is gene fusion—when two genes merge to form a hybrid gene [13]. In Ref. [14] the authors construct a network of human genes that have been observed to be fused in the development of tumors in humans. Some genes can fuse with many others but most of the genes have only been observed fusing with one, or a few others. Statistics of this and the other networks are listed in Table I. The second example network represents the metabolism (the cellular biochemistry except signaling processes) of humans. It is constructed by connecting the substrates of a reaction with the products (a so called *substrate graph* [15]). Furthermore are the most common chemical substrates removed (according to the method in Ref. [16])—this is commonly done since these substances are so common that they does not put any constraint on the biochemical flow. In protein interaction networks the vertices are proteins and two proteins constitute an edge if they can interact physically. We use the ("physical interaction") data

TABLE I. Basic statistical properties of the example networks we use. The number of vertices N, number of edges M, assortativity r, clustering coefficient C, relative size of the largest cluster s, average distance in the largest cluster $\langle d \rangle$, the error robustness R_{error}, and the attack robustness R_{attack}.

	Gene fusion	Protein interaction	Metabolic	Neural
N	291	4168	1905	280
M	278	7434	3526	1973
r	−0.36	−0.13	−0.10	−0.069
C	0.0016	0.034	0.039	0.20
s	0.38	0.94	0.87	1
$\langle d \rangle$	4.2	4.8	4.5	2.6
R_{error}	0.43	0.36	0.36	0.50
R_{attack}	0.012	0.048	0.046	0.38

set from Ref. [17] of protein interaction in the budding yeast *S. cerevisiae*.

Like metabolic networks there is a question how to represent neural network—either one use a coarse-grained representation of a complex organism's nervous system [18], or the complete neuronal map of a simple organism. In this work, we take the latter approach and use the neural network of *C. elegans* [19].

In this work we assume the subject network to be accurate. To get more valid error estimates for our structural measures, one would need to take the accuracy of the edges into account.

V. NUMERICAL RESULTS

In this section we present numerical results for our four network-structural measures over the $\mathcal{G}(G)$ ensembles of the four test graphs. To get a first view, we display the valid region of the gene fusion graph in Fig. 2(a). As seen, the valid region is not covering a large part of the theoretical limits of r $(-1 \leqslant r \leqslant 1)$ and C $(0 \leqslant C < 1)$. The requirement that the graph should be simple (no multiple edges or self-edges) puts hard constraints on the actual r values that can occur (cf. Ref. [20]). Figure 2(a) shows that, considering the entire r-C plane, such constraints are even harder. The general shape of the valid region is consistent with the observations that the simple-graph constraint induce a positive correlation between r and C [20,21].

In Figs. 2(b), 2(c) and 2(d) we show three example networks of $\mathcal{G}(G)$ (where G is the gene fusion network). Figure 2(b) displays the relatively fragmented real network. Figure 2(c) is a random network G' with the almost the same r-C coordinates as the real network $[\delta(G,G') \approx 0.0026]$. Maybe the biggest visible difference between G and G' is the larger size of the largest component of G'. Is it true that the gene fusion network is unusually fragmented, given the degree sequence and r-C coordinates? If so, there might be an evolutionary pressure for gene fusion networks to be fragmented. Figure 2(d) shows, as a contrast, a network far away from G and G'. The network has a well-defined core where high-degree vertices connect to each other. There are also a

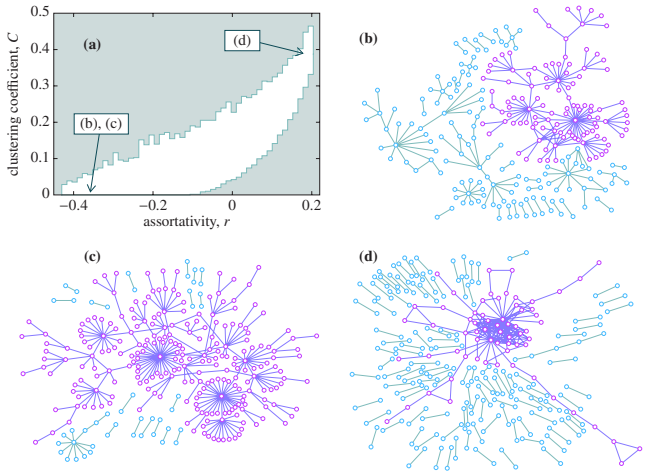

FIG. 2. (Color online) The valid region demarcated by the (a) $C_{\min}(r)$ and $C_{\max}(r)$ curves and three networks: (b) is the original gene fusion network; (c) shows a random sample with r-C coordinates close to those of the real network. Panel (d) shows a network with high clustering and high assortativity. The largest component of (b), (c), and (d) are indicated with a different color.

number of peripheral triangles, which indicates that the network evolves toward a maximal C value, given its assortativity.

A. Location in r-C space

In Fig. 3 we plot the relative size of the largest component of the four test networks. We also display the locations of the actual networks in the r-C plane, and the $\mathcal{G}(G)$ averages. [The $\mathcal{G}(G)$ averages are obtained from a rewiring sampling of $\mathcal{G}(G)$, see Appendix A.] We see that the C value of the gene fusion graph lies close to the $C_{\min}(r)$ boundary of its valid region. C averaged over the whole $\mathcal{G}(G)$ is about three times larger ($\langle C\rangle_{\mathcal{G}(G)}=0.0061\pm0.0001$) than the observed value ($C=0.0017$). Furthermore, we see that the assortativity is lower than the $\mathcal{G}(G)$ average. This kind of analysis has been used by many authors (following Ref. [8]). The interpretation is usually that the network is, effectively, disassortative and clustered (i.e., $r<\langle r\rangle_{\mathcal{G}(G)}$ and $C>\langle C\rangle_{\mathcal{G}(G)}$). However, looking at the entire valid region, we can get another perspective: If high clustering really would have been an important goal for the network (given the degree sequence) there is large room for improvement. For the assortativity, on the other hand, the observed network is rather close to the minimum. This might be telling us that assortativity is a more important factor, than clustering, in the evolution of the gene fusion networks. The protein interaction network of Fig. 3(b) is located quite far from the ensemble average—the assortativity is much lower than the $\mathcal{G}(G)$ average, and given that assortativity, the clustering is maximal. Also the metabolic [Fig. 3(c)] and neural [Fig. 3(d)] networks are more clustered than the average, but here the assortativity is slightly larger than the $\mathcal{G}(G)$ average. From Fig. 3 we also note that the density of states is very inhomogeneous distributed—the average (r,C) is close to $C=0$ and (except for the neural network) left of the middle of the assortativity spectrum. This is confirmed by a brief, unbiased sampling of $\mathcal{G}(G)$; we generate 10^5 members of $\mathcal{G}(G)$ and measure the extreme values of r and C (and repeat the procedure ten times to obtain error estimates). These intervals for the metabolic network are $\Delta r=[-0.164(4),-0.147(4)]$ and $\Delta C=[0.0075(2),0.0099(2)]$, which is one 24th and one 180th, respectively, of the full valid ranges. Similar observations hold for the other networks. This illustrates why the full valid r-C region cannot be sampled by random rewiring—the extreme networks are just too rare to be sampled—but the real values ($r=-0.101$, $C=0.0394$) are also extreme.

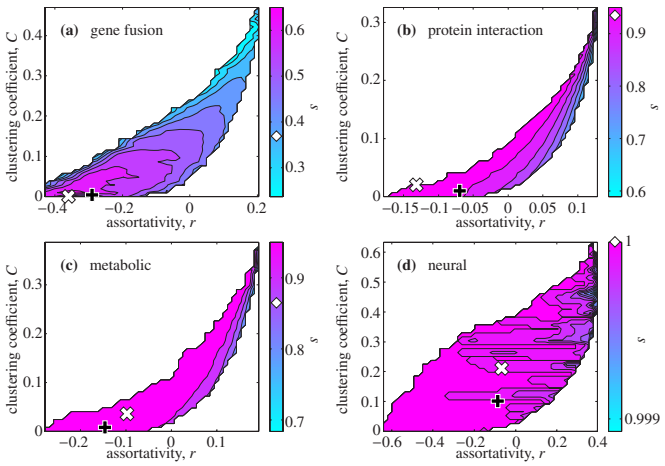

FIG. 3. (Color online) The relative size of the largest component s as a function of r and C. The networks are (a) the network of gene fusions in tumors in humans, (b) protein interaction network of *S. cerevisiae*, (c) human metabolic network and (d) the *C. elegans* neural network. The $\times$-like symbols of the main figures and the diamond symbols of the color bars indicate the values of the real networks. The plus-like symbol indicates the average (r,C) value of the $\mathcal{G}(G)$ ensemble.

The shapes of the valid regions are rather similar, with an exception for the broader region of the neural network. This can be related to the more narrow degree sequence of the neural network [22]. We have established a correlation between r and C. Reference [21] argues that such correlation occurs in social networks because of their modularity (or "community structure" as the authors call it). However, our large-r networks have no explicit bias towards high modularity, which leads us to conjecture that the correlation between r and C, or more fundamentally the sum $\Sigma_{(i,j)\in E}k_ik_j$ [which, given a degree sequence, is the only factor of Eq. (1) that can vary] is a more general phenomenon (cf. Ref. [23]). Since r is normalized by, essentially, the variance of the degree, it follows that the valid region for $\mathcal{G}(G)$ with more narrow degree sequence will appear stretched (larger).

B. Size of largest component

Turning to the average size of the largest component, we observe that the gene fusion network is indeed more fragmented than the average network of the same (r,C) coordinates [as anticipated from comparing Figs. 2(b) and 2(c)].

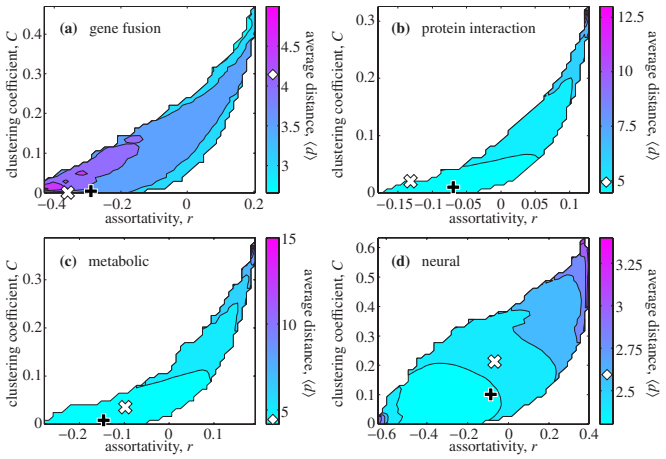

FIG. 4. (Color online) The average distance within the largest component $\langle d \rangle$ as a function of r and C. The panes and symbols correspond to those of Fig. 3.

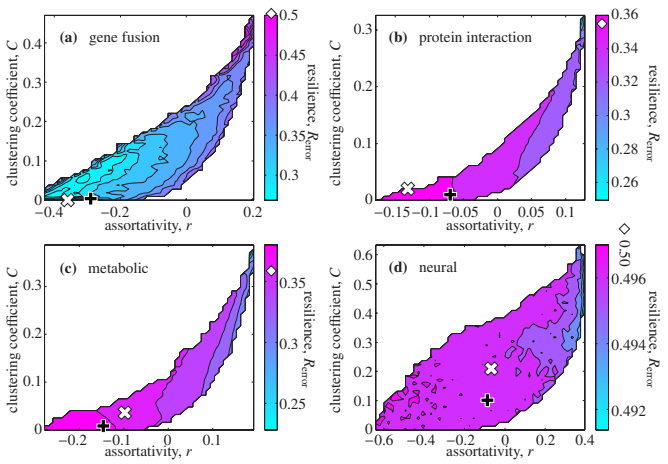

FIG. 5. (Color online) The error robustness R_{error} as a function of r and C. The panes and symbols correspond to those of Fig. 3.

The protein interaction and neural networks have no particular bias in this respect, whereas the metabolic network is more fragmented than expected. The relatively low s of the metabolic network can be attributed to the "modularity" of such networks [15,16]. Such modules are subgraphs that are densely connected within, and sparsely interconnected. Sometimes they are even disconnected from the largest component (which explains the lower s). In general, s decreases with assortativity. This is natural—in more assortative networks high degree vertices are connected to each other, forming a highly connected core and a periphery too sparse to be connected [viz. Fig. 2(c) and 2(d)]. For the denser networks (the protein interaction, metabolic, and neural networks) s increases with C (for a fixed r). For the sparser gene-fusion network s has a peak at intermediate C. We do not speculate further about combinatorial cause of these dependencies; but we note [comparing, e.g., Figs. 2(a) and 2(b)] that even though the shape of the valid regions are similar, the s behavior can be qualitatively different.

C. Distances in the largest component

In Fig. 4 we display the average distance in the largest component. As mentioned, measuring the distance can give complementary information to the $s(r,C)$ graphs of Fig. 3—while s tells us how much of the network that can be reached, $\langle d \rangle$ tells us how fast that can happen. For all networks the big picture is that large connected components have large average distances. This is expected from most network models. There is, however, more information than this in Fig. 4: For components of the same size, the average distance is (except for the gene fusion network) increasing with both r and C. That $\langle d \rangle$ should increase with C seems quite natural—if one of a triangle's edges is rewired to connect two distant vertices, the distances in the surrounding of the triangle would increase with one, but this would be more than compensated by the connection of the two previously distant areas. Disassortative networks typically lack a well-defined core. Such cores are known to keep the average distance of general power-law networks short [24]. Thus one would expect an increase of r to cause a larger $\langle d \rangle$, but apparently the clustering-related increase of the average distance outweighs this effect. In contrast to the relative size, the average distances of the real networks are close to the $\mathcal{G}(G)$ averages at the same r-C coordinates. The behavior of the gene fusion network is the opposite of the others. As seen in Sec. V B, s varies more for the gene fusion network than the other networks—it is natural that the larger-network-larger-$\langle d \rangle$ effect is dominant over the clustering-related increase mentioned above, which explains why the largest average distances of the gene fusion network is in the low-C and low-r corner of the (C,r) space.

D. Error robustness

Next, we turn to the error robustness problem. As seen in Fig. 5 the $\mathcal{G}(G)$ ensemble of the gene fusion network [Fig. 5(a)], once again, has a qualitatively different behavior than the other three networks [Figs. 5(b)–5(d)]. While the gene fusion network is most robust for high r and C values the other networks are most robust for low r. A sketchy explanation can be found in the chainlike subgraphs extending from the largest component in a large-r network (cf. Fig. 2)—with a random deletion of vertices, these subgraphs are likely to be disconnected from the core rather soon (whereas in a disassortative network alternative paths may still exist), then if the deletion-robust core is less than half of the original component size it follows that it may soon be isolated. The sparsity of the gene fusion network makes the low-r $\mathcal{G}(G)$ graphs much like trees (i.e., having few cycles), and since cycles provide redundant paths that can make a network robust, it follows that these graphs are fragile. For a fixed r, R_{error} is a decreasing function of C for the three largest networks. We believe this is an effect of the local path redundancy induced by triangles—if one vertex of a triangle is deleted, the other two are still connected. For the gene fusion network the behavior is once again opposite from the others—the most robust networks have high C and r. For this network in particular the above-mentioned treelike sub-

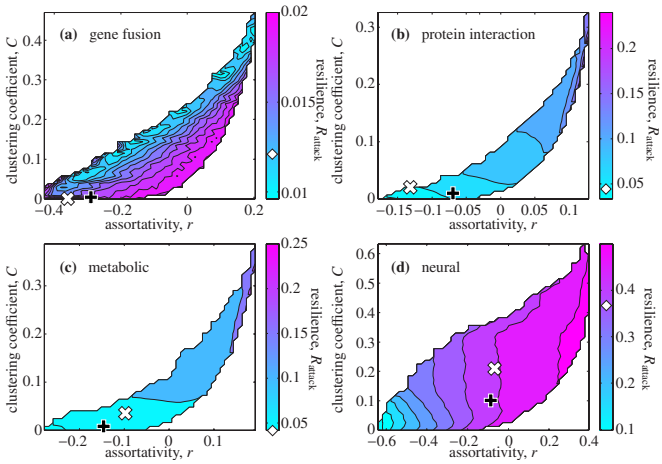

FIG. 6. (Color online) The attack robustness R_{attack} as a function of r and C. The panes and symbols correspond to those of Fig. 3.

graphs increase in frequency with C and r which explains the different behavior.

The R_{error} values for the real networks are always markedly higher than the $\mathcal{G}(G)$ averages for the same (r,C) coordinates. Networks with highly skewed degree distributions (the gene fusion, protein interaction and metabolic networks) are known to be robust to errors by virtue of degree distribution alone [11], now Fig. 5 tells us that all these networks have a yet higher error tolerance which is an indication that error robustness is an important factor in the evolution of these networks.

E. Attack robustness

The final quantity we measure is the attack robustness (see Fig. 6). R_{attack}'s functional dependence on r and C is quite different from that of R_{error}. The gene fusion $\mathcal{G}(G)$ has the highest attack robustness at high r and low C values. The other networks have higher robustness values for high assortativity, but no clear tendency in the C direction. The attack mechanism we study targets the high degree vertices. Having all high degree vertices connected to each other is probably the only way to keep the network from instantaneous fragmentation. The observed r dependence is thus rather expected. The real-world networks all have R_{attack} values of the same order of magnitude as the average values for the $\mathcal{G}(G)$ networks of the same location in r-C space. We note that for studying the attack problem of metabolic networks, the (less common) enzyme centric graph representation is more appropriate (see Sec. IV). The reason being that one can suppress an enzyme much easier than removing the substrates.

F. Comparison between the graphs

Even though all our example networks are constructed from biological data, they represent fundamentally different systems—the neural network is spatial by nature, the protein interaction and (even more so) the metabolic networks are the background topology for an active dynamic system,

TABLE II. Summary of the network structural measures of the real world networks relative to the average values of the $\mathcal{G}(G)$ a distance $\delta<0.02$ from the real network. "<" indicates that the real network have a lower value than the corresponding $\mathcal{G}(G)$ value. All results are significant with p values >0.01, except the s value of the neural network that has a p value of ~ 0.05.

	Gene fusion	Protein interaction	Metabolic	Neural
s	<	<	<	>
$\langle d\rangle$	<	<	<	<
R_{error}	>	>	>	>
R_{attack}	>	>	>	>

whereas the gene fusion network is a representation of possible but undesired events. The protein interaction, metabolic and neural networks have one thing in common—the organism needs them to be robust to errors (caused by injuries, mutations, disease etc.) [25]. As mentioned above and summarized in Table II the error robustness is indeed higher for the real networks than the $\mathcal{G}(G)$ ensemble at the same (r,C) coordinates. As mentioned above, the attack robustness of the real network is of the same order as the $\mathcal{G}(G)$ average at the same (r,C) coordinate, but actually there is a significant tendency that these network also are more robust to attacks. Furthermore, the distances in the largest component, and the relative sizes s are (with the neural network s value as the only exception) smaller in the real than the $\mathcal{G}(G)$ networks. In general, none of the networks have the same structural measures as the $\mathcal{G}(G)$ averages at their coordinates. This suggests that the degree-distribution, assortativity, and clustering coefficient are not enough to fully describe the structure of the network.

Despite these similarities between the statistics of the real-world networks the r-C space of the different degree sequences have qualitatively different network structure. Especially, the gene fusion network behaves almost the opposite of the other networks (at least for s, $\langle d\rangle$ and R_{error}. The source of this opposite behavior (as we discuss above) is probably that it is much sparser than the other networks. The neural network is the densest network and the only one that does not have a power-law-like degree distribution.

VI. DISCUSSION

Different quantities for measuring network structure are usually not independent. This is usually seen from correlations between quantities in ensembles of networks. This makes it hard to hypothesize about evolutionary favorable network structure from values of quantities alone. In this paper we suggest a method to analyze null models and the original network, in parallel, so that the constraints on the network's evolution and the correlations of the quantities are easier to infer. Using this information one can make hypotheses about the microscopics of the evolution. The particular null model we use is $\mathcal{G}(G)$—the ensemble of graphs with the same degree sequence as the subject graph G. In this work

we map out $\mathcal{G}(G)$ in the two-dimensional space defined by the clustering coefficient and the assortativity. Then we measure other network structural quantities throughout this space (or, alternatively, one can use other studies linking r and C to the performance of the system). One formal way to see our method is that we resolve $\mathcal{G}(G)$ in the (high dimensional) space of all sensible network measures. Then, for simplicity, we project to a few dimensions. (The case of projection to one dimension has been studied in a less formalized way earlier—projection to assortativity [26] or a "hierarchy" measure [27].) For example, if G is just beneath the boundary of the r-C space, and the boundary, at that location, runs in the r direction [so that C values larger than G's would fall outside of $\mathcal{G}(G)$], then a few changes can bring the network in any direction in the (r,C) space except in the positive C direction. In such a situation we can surmise that there is an evolutionary force driving the network towards high clustering. Further, one can include other network structural measures (like average largest component size, pathlength, error and attack robustness) to make the study more detailed. The method can, straightforwardly, be generalized to temporal network data. In such cases one can observe the movement of G in the (r,C) space and make yet more relevant hypotheses of the network forming forces.

We exemplify our method by studying four different biological networks. The functional characteristics of the r-C spaces varies much between the four example networks. For example, the *C. elegans* neural network covers a much larger area of the r-C space than the other networks; whereas the other networks have a valid region of similar shape and size. Despite this similarity, the gene network shows a structural dependency on r and C that is very different from the metabolic and protein interaction networks. The position of the real networks in the valid region of the r-C space adds some further information. For example, the gene network is close to the border in the r direction, but not in the C direction, suggesting that assortativity has been a more important factor, than clustering, in the evolution of the network. Furthermore we compare the network structure of the real networks with the average values of networks in $\mathcal{G}(G)$ that are close to the (r,C) coordinates of the real network. From this analysis, we conclude that all our four example networks are more robust to both random errors and targeted attacks than what can be expected from a random network constrained to the same degree distribution, assortativity and clustering coefficient. For all networks, except maybe the gene fusion network, this is in line with robustness being an important factor in the network evolution.

The analysis scheme presented in this paper can be further extended and analyzed. It would be interesting with a quantitative evaluation of the network-structural spaces, and how they depend on the degree sequence. One can also, for time-resolved data sets, incorporate dynamic information in the analysis by monitoring the network-evolutionary trajectory in the r-C space. By doing this one can observe the selective pressure, in terms of assortativity and clustering, in the evolution of the network. Yet another possibility is to identify the structure that is most relevant for the network evolution by some method similar to principal component analysis.

ACKNOWLEDGMENTS

The authors thank Mikael Huss, Martin Rosvall, and Alexei Vazquez for helpful suggestions and comments. P.H. acknowledges financial support from the Wenner-Gren Foundations and the National Science Foundation (Grant No. CCR-0331580).

APPENDIX A: DETAILS OF THE ALGORITHM

In this appendix we address details of the analysis scheme (that, for clarity, were omitted from Sec. III).

To find the $\mathcal{G}(G)$ elements of minimal and maximal assortativity is a nontrivial optimization problem. There are deterministic methods that, if they terminate, are guaranteed to give the maximal (or minimal) assortativity [26,28]. To avoid such technicalities and to simplify the program, we will use the same kind of optimization algorithm to find $r_{\max}$ and $r_{\min}$ as to find $C_{\min}(n)$ and $C_{\max}(n)$. In Appendix B we will argue that this method allows us to come as close to the optimal r values as we need. A method we find efficient is to repeat the simple edge-pair swapping procedure (where only changes in the desired direction are accepted) with different random seeds until no lower state is found during a number ν_{rep} of repetitions [29]. Each individual edge pair is terminated when no lowest state is found for ν_{same} swaps. In general, the larger the network is, the more densely distributed are the points close to the border of the valid region. If one is satisfied with finding a value a certain distance from the extrema, then ν_{rep} and ν_{same} do not need to be increased for larger N. To find $C_{\min}(n)$ and $C_{\max}(n)$ almost the same procedure is employed. First, edge pairs are swapped until the desired segment of r is found. Second, unless r is outside the segment n and the move takes the system yet further from segment, edge pairs are swapped provided the clustering would decrease [for $C_{\min}(n)$] or increase [for $C_{\max}(n)$]. When the valid region is traced out and we sample networks of different pixels, we select the pixels randomly. The idea is to sample the space of networks more randomly.

To summarize, the algorithm for finding the extremal assortativity values, $r_{\min}$ and $r_{\max}$, is as follows.

(1) Choose two undirected edges (i,j) and (i',j') at random. If the program makes a difference between the arguments of the edge, the direction of the reading of the edge also has to be randomized [so (i,j) is read as (j,i) with probability 1/2].

(2) Check if swapping these edges to (i,j') and (i',j) would introduce a self-edge or multiple edge in the network. If so, go to step (1).

(3) Let Δr be the change in r if the move in step (1) is executed. If r is to be minimized and $\Delta r<0$, then accept the change (vice versa for maximization of r).

(4) If no move has been executed during the last ν_{same} executions of step (3), then take the current r as $\tilde{r}_{\min}$ (or $\tilde{r}_{\max}$).

(5) Repeat from the beginning ν_{rep} times and return the lowest observed $\tilde{r}_{\min}$ during these iterations.

Given $r_{\min}$ and $r_{\max}$, and a division of the r space into L segments of width $(r_{\max}-r_{\min})/L$, we trace the boundaries of the valid region as follows.

(6) Go through the regions sequentially. Say the nth region is the interval $[r_n, r_{n+1})$.

(7) Perform steps (1) and (2) of the assortativity optimization algorithm.

(8) Let ΔC be the change in clustering coefficient during the previous step. If $r < r_n$ and $\Delta r > 0$, $r \geqslant r_{n+1}$ and $\Delta r < 0$ or $r_n \leqslant r < r_{n+1}$ and $\Delta C < 0$ (for minimization) or $\Delta C > 0$ (for maximization), then perform the change of step (6).

(9) If, counting from the first time the system entered the desired r segment, the minimal (maximal) C value has been repeated ν_{same} times, take this value as $\tilde{C}_{\min}(n)$ $[\tilde{C}_{\max}(n)]$.

(10) Repeat from step (6) ν_{rep} times. Let the lowest $\tilde{C}_{\min}(n)$ values and largest $\tilde{C}_{\max}(n)$ during these iterations be $C_{\min}(n)$ and $C_{\max}(n)$.

Then, when the valid region is mapped out, we split the C range (between $C_{\min}$ and $C_{\max}$) in L segments of equal width, thus forming an $L \times L$ grid enclosing the valid region. This grid is sampled as follows.

(11) Construct a random list of the valid pixels (i.e., a list where all valid pixels appear once and only once).

(12) Pick the next pixel $P_n = [r_n, r_{n+1}) \times [C_m, C_{m+1})$ from the index list of step (11). Denote the center $[(r_n + r_{n+1})/2, (C_m + C_{m+1})/2]$ of the pixel $(r_{n,0}, C_{m,0})$. Let

$$\delta(r,C) = \sqrt{\left(\frac{r - r_{n,0}}{r_{\max} - r_{\min}}\right)^2 + \left(\frac{C - C_{m,0}}{C_{\max} - C_{\min}}\right)^2} \quad \text{(A1)}$$

measure the distance in r-C space from the current position (r, C) to the center of the target pixel.

(13) Pick edge-pair candidates according to steps (1) and (2) of the assortativity optimization algorithm.

(14) Calculate $\Delta(r,C) = \delta(r',C') - \delta(r,C)$ where r and C are the current assortativity and clustering values, and r' and C' are the values after the pending move has been performed. If $\Delta(r,C) < 0$ perform the move.

(15) If the updated (r,C) belongs to P_n, then, first, make ν_{rnd} random edge swappings such that (r,C) does not leave P_n. (This is to sample the pixel more uniformly.) Then measure network structural quantities of P_n, save these values for statistics, and go to step (12).

(16) If not all pixels have been measured go to step (12).

(17) Go to step (11) until each pixel have been sampled ν_{samp} times.

The parameter values we use in this study are (unless otherwise stated) the following: $\nu_{\text{same}} = 10^5$, $\nu_{\text{rep}} = 5$, $\nu_{\text{samp}} = 100$, $\nu_{\text{rnd}} = 1000$, and $L = 50$. The choice of parameters and further considerations are discussed in Appendix B. Due to the uncertain stopping conditions of steps (4), (5), (9), and (10) it is hard to derive meaningful bounds on the computational complexity. We note, however, that the optimization is faster in r than in C direction, this probably relates to the observation in Fig. 1(b) that swapping procedure moves faster in the r than in the C direction. (The speed in the C direction is roughly the same per 1000 steps, but the speed in the r direction decreases.)

APPENDIX B: CONVERGENCE AND SAMPLING UNIFORMITY

In this appendix, we address some technical issues of our method related to the convergence of our optimization algorithm and uniformity of the sampling. We will also motivate our choice of parameters.

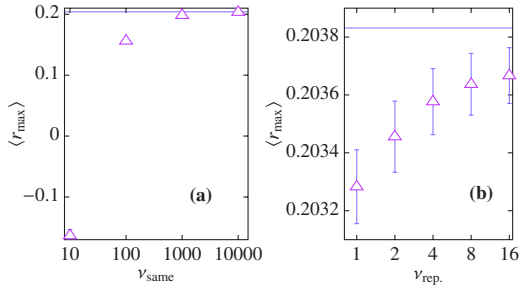

FIG. 7. (Color online) Convergence of the optimization algorithm. Panel (a) shows the average maximal assortativity $\langle r_{\max} \rangle$ with $\nu_{\text{rep}} = 1$. The horizontal line represents the result of the maximization algorithm of Ref. [28]. Panel (b) shows the further improvement by finding the maximum over many independent runs (for $\nu_{\text{same}} = 10000$). The vertical bars indicate the standard deviation of the observed maxima.

1. Assortativity and clustering extremes

To find the extremal assortativity values we use the edge-swapping algorithm described in Sec. III. To find $r_{\min}$ we start from a random member of $\mathcal{G}(G)$ and swap random edge pairs (keeping the graph simple at all times) that lower r. When no graph of lower r has been found for ν_{same} time steps, we break the iteration. To avoid the effect of being trapped in local minima, this process is repeated ν_{rep} times. The main motivation for using this method is that it is at heart the same scheme as for obtaining the extremal clustering values and sampling the valid region (and thus we can reuse the same code for many steps of the calculations). In this section, we argue that the optimization performance of this method is sufficiently good for our purpose.

There is a deterministic method to maximize the assortativity that is, if it exits properly, guaranteed to find $r_{\max}$ [28]. The method works as follows: First all vertex pairs (i,j) are ranked in decreasing order of the product of their degrees, $k_i k_j$. Then the edges are added in order of this list unless the degree of one of the vertices already is fulfilled. There are some other technicalities from the additional constraint (of the authors) that the network should be connected. Of our networks, only the neural network has such an evolutionary constraint, so we do not impose it.

In Fig. 7 we display the parameter dependence of the convergence for the gene fusion network. The horizontal line is the theoretical maximum obtained by the algorithm of Ref. [28]. When $\nu_{\text{same}} = 10000$ we obtain an average maximal assortativity within 0.001 of the theoretical maximum [Fig. 7(a)]. By increasing ν_{rep} the accuracy can be increased further [Fig. 7(b)]. The lattice spacing we use is $0.005 \lesssim r \lesssim 0.02$, so we deem a precision of 0.001 sufficient. The gene fusion network is our smallest network but the other networks are not harder to converge. When one edge pair is swapped so that r decreases, the only term of Eq. (1) that changes is $\langle k_1 k_2 \rangle$. The potential change of the sum $\Sigma_{(i,j) \in E} k_i k_j$, in the calculation of $\langle k_1 k_2 \rangle$ (close to the extrema)

is of the order of the typical degree values of the network. These values grow slower than the network itself, which means that a larger network can be closer in r, but further away in number of edge swaps to reach the global optimum, than a smaller network. Some authors [28] use $\Sigma_{(i,j)\in E} k_i k_j$ to measure the degree correlations, but since we strive for a macroscopic level of description (consistent in the large-N limit), r is a more appropriate quantity for the present work.

The optimization of the clustering to find the minima (maxima) of the segments of assortativity space follows the same pattern as the method to find the minimal (maximal) r. Changes of the parameters (ν_{same} and ν_{rep}) have the same effect as in Fig. 7, and the same values seem sufficient.

2. Sampling uniformity

The other technical issue we address in this Appendix is the uniformity of our sampling procedure. Ideally we would like all unique (i.e., non-isomorphic) members of $\mathcal{G}(G)$ to be sampled with the same probability. The most important observation is trivial—by edge-pair swapping one can go from one member of $\mathcal{G}(G)$ to any other, and thus all members of the ensemble will contribute to the averages. A much harder question is whether or not every member of $\mathcal{G}(G)$ is sampled with uniform probability. In this section, we will argue that our algorithm does a reasonably good job in the sense that there are no inconsistencies and parameter values are appropriate.

When the target pixel is found [step (15) of the algorithm] we perform ν_{rnd} additional random edge-pair swaps. The idea is to sample the $\mathcal{G}(G)$ members of the pixel more uniformly (and indeed to be able to reach into the interior of the pixel). In Fig. 8(a) we illustrate the effect of these random moves. We plot a normalized histogram of the relative largest cluster size s for 0, 100, and 10000 random moves. We see that these moves do make a difference (the $\nu_{\text{rnd}}=0$ is different from the $\nu_{\text{rnd}}=100$) but it does not matter if $\nu_{\text{rnd}}=100$ or $\nu_{\text{rnd}}=10000$. The same situation is observed for other pixels, networks and quantities. Therefore, we use $\nu_{\text{rnd}}=1000$ in this work.

Next, we will illustrate the use of the randomly permuted list in the sampling of the pixels [steps (11) and (12) of the algorithm]. The motivation for this procedure is that the network structure can depend on the direction from which the search arrives to the pixel. In Fig. 8(b) we illustrate the test procedure—we sample separate histograms from four starting points in the four cardinal directions with respect to the central $(r,C)=(-0.1,0.1)$ pixel. In Fig. 8(c) we see that the histograms from the W and S pixels are different. There appears to be two regions of $\mathcal{G}(G)$ contributing to these histograms (one with $s\approx 0.65$, one with $s\approx 0.75$). Searches starting from W seem to arrive at the $s\approx 0.75$ region more

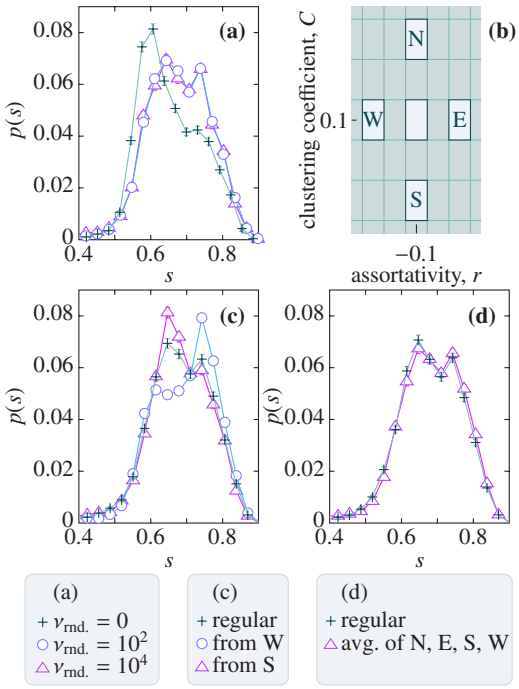

FIG. 8. (Color online) Histograms of s for the discussion of sampling uniformity. All the histograms are from the gene fusion network and a pixel centered around $r=-0.1$, $C=0.1$ (the dimensions of a pixel are $\Delta r=0.013$, $\Delta C=0.0096$). The error bars represent standard errors. Lines are guides for the eyes. Panel (a) shows the histograms with different numbers of random edge-pair swappings ν_{rnd} within the pixel before the measurements of quantities. Panel (b) illustrates the location of the starting point pixels used in panels (c) and (d). Panel (c) compares histograms for swapping processes starting at W, S with the regular algorithm. Panel (d) compares the average histogram of walks starting in the four peripheral points of (b) with the result of the regular algorithm. In panels (c) and (d) $\nu_{\text{rnd}}=1000$. The whole range of the histograms is not shown, which is why the areas under the curves appear different.

frequently, and searches staring at S ends up around $s\approx 0.65$ more frequently. The curve of the actual algorithm weighs the two peaks more equal. The curves from N and E coincides almost completely the curve for the regular algorithm (and are therefore omitted for clarity). The impression we get is that the search from one direction can induce a bias in the network structure [symbolically speaking, the graphs have a preference for ending up in a certain region of $\mathcal{G}(G)$]. However, from other directions, or by the random sampling of pixels [step (11)], the bias is reduced. This picture is further strengthened in Fig. 8(d) where we show that the average value of the histograms from the four starting points are overlapping with the histogram of the regular algorithm.

[1] R. Albert and A.-L. Barabási, Rev. Mod. Phys. **74**, 47 (2002).
[2] S. N. Dorogovtsev and J. F. F. Mendes, *Evolution of Networks: From Biological Nets to the Internet and WWW* (Oxford University Press, Oxford, 2003).
[3] M. E. J. Newman, SIAM Rev. **45**, 167 (2003).
[4] A.-L. Barabási and R. Albert, Science **286**, 509 (1999).
[5] M. E. J. Newman, Phys. Rev. Lett. **89**, 208701 (2002).
[6] A. Barrat and M. Weigt, Eur. Phys. J. B **13**, 547 (2000).
[7] D. Gale, Pac. J. Math. **7**, 1073 (1957).
[8] S. Maslov and K. Sneppen, Science **296**, 910 (2002).
[9] S. Shen-Orr, R. Milo, S. Mangan, and U. Alon, Nat. Genet. **31**, 64 (2002).
[10] P. Holme, B. J. Kim, C. N. Yoon, and S. K. Han, Phys. Rev. E **65**, 056109 (2002).
[11] R. Albert, H. Jeong, and A.-L. Barabási, Nature (London) **406**, 378 (2000).
[12] A. E. Motter, Phys. Rev. Lett. **93**, 098701 (2004).
[13] F. Mitelman, B. Johansson, and F. Mertens, Nat. Genet. **36**, 331 (2004).
[14] M. Höglund, A. Frigyesi, and F. Mitelman, Oncogene **25**, 2674 (2006).
[15] J. Zhao, H. Yu, J. Luo, Z. W. Cao, and Y.-X. Li, Chin. Sci. Bull. **51**, 1529 (2006).
[16] M. Huss and P. Holme, IET Systems Biology, arXiv:q-bio/0603038 (unpublished).
[17] P. Holme and M. Huss, J. R. Soc., Interface **2**, 327 (2005).
[18] O. Sporns, G. Tononi, and G. M. Edelman, Cereb. Cortex **10**, 127 (2000).
[19] J. G. White, E. Southgate, J. N. Thompson, and S. Brenner, Philos. Trans. R. Soc. London, Ser. A **314**, 1 (1986).
[20] S. Maslov, K. Sneppen, and A. Zaliznyak, Physica A **333**, 529 (2004).
[21] M. E. J. Newman and J. Park, Phys. Rev. E **68**, 036122 (2003).
[22] L. A. N. Amaral, A. Scala, M. Barthélémy, and H. E. Stanley, Proc. Natl. Acad. Sci. U.S.A. **97**, 11149 (2000).
[23] S. N. Soffer and A. Vazquez, Phys. Rev. E **71**, 057101 (2005).
[24] F. Chung and L. Lu, Proc. Natl. Acad. Sci. U.S.A. **99**, 15879 (2002).
[25] A. Wagner, *Robustness and Evolvability in Living Systems* (Princeton University Press, Princeton, NJ, 2005).
[26] J. Zhao, L. Tao, H. Yu, J.-H. Luo, Z.-W. Cao, and Y.-X. Li, e-print arXiv:physics/0611078.
[27] J. B. Axelsen, S. Bernhardsson, M. Rosvall, K. Sneppen, and A. Trusina, Phys. Rev. E **74**, 036119 (2006).
[28] L. Li, D. Alderson, J. C. Doyle, and W. Willinger, Internet Math. **2**, 431 (2005).
[29] L. R. Walker and R. E. Walstedt, Phys. Rev. B **22**, 3816 (1980).