# Robustness of community structure in networks

Brian Karrer,[1] Elizaveta Levina,[2] and M. E. J. Newman[1,3]
[1]*Department of Physics, University of Michigan, Ann Arbor, Michigan 48109, USA*
[2]*Department of Statistics, University of Michigan, Ann Arbor, Michigan 48109, USA*
[3]*Center for the Study of Complex Systems, University of Michigan, Ann Arbor, Michigan 48109, USA*

(Received 24 September 2007; published 29 April 2008)

The discovery of community structure is a common challenge in the analysis of network data. Many methods have been proposed for finding community structure, but few have been proposed for determining whether the structure found is statistically significant or whether, conversely, it could have arisen purely as a result of chance. In this paper we show that the significance of community structure can be effectively quantified by measuring its robustness to small perturbations in network structure. We propose a suitable method for perturbing networks and a measure of the resulting change in community structure and use them to assess the significance of community structure in a variety of networks, both real and computer generated.

DOI: 10.1103/PhysRevE.77.046119 PACS number(s): 89.75.Hc, 02.10.Ox, 02.50.−r

## I. INTRODUCTION

Many networks of scientific interest decompose naturally into communities or modules, densely connected subsets of nodes with only sparser connections between them. In many cases communities have been found to correspond to behavioral or functional units within networks, such as functional modules in biochemical networks or social groups within social networks. This finding suggests that in networked systems whose function is less well understood we may be able to gain insight by discovering and examining their communities (if any), and methods for community discovery have, as a result, attracted a substantial amount of attention in the recent literature in many disciplines [1–3].

Communities are of interest for other reasons as well. Their presence can, for example, dramatically alter the behavior of dynamical processes on networks [4] (and indeed the observation of dynamical processes has been proposed as one possible method of community detection [5]). Communities can also be used as a basis for the reduction or coarse-graining of networks for visualization or other purposes [6,7]. And communities frequently display different statistics from the network as a whole, indicating that global network statistics such as degree moments or correlation functions may potentially fail to register important heterogeneities [8].

A large number of methods for finding communities have been proposed in recent years, including divisive methods based on betweenness and similar measures [9,10], methods based on searching for small cliques [11,12], information-theoretic techniques [13], statistical inference through belief propagation [14] or maximum likelihood [15], and many others.

Perhaps the most widely used technique, however, and the one on which we focus in this paper, is the maximization of the benefit function known as modularity [6,16–22], which is (to within a multiplicative constant) the difference between the number of edges within communities and the expected number of such edges under an appropriate null model. Various null models have been used but the commonest by far is the standard configuration model [23,24], which preserves the degree sequence of the original network but otherwise randomizes edge positions. The modularity is then maximized over possible divisions of the network, the optimal division being taken to be the correct partition of the network into communities.

The modularity maximization method appears, in principle, to work well [2], although it also has some limitations, notably its inability to detect small communities embedded in large networks [25,26] (though variants of the method have been proposed to circumvent this limitation [21,27]). A more fundamental problem, however, is that maximization of modularity is an NP-complete task [28] and hence is essentially intractable for all but the smallest of networks. In practical implementations of the modularity method, therefore, approximate heuristics are usually employed, such as greedy algorithms [16,17], extremal optimization [18], simulated annealing [19–21], or spectral methods [22]. These methods vary in their effectiveness and speed, the faster algorithms tending to give poorer results while the slower ones can only be applied to smaller networks if running time is to be kept to reasonable levels. In this paper we employ the spectral optimization method introduced in [22], which displays a reasonable balance between accuracy and speed, but the calculations we describe are not tied to this method. Indeed our approach is not even tied to modularity maximization in general, and could be applied to any community detection scheme with only minor modifications. For concreteness, however, we concentrate on modularity maximization in this paper.

Despite the large volume of work on community detection and its applications, one important question remains largely unaddressed, that of the significance of the results. How can we tell when the communities detected by one method or another are truly significant and when they could be merely the consequence of a chance coincidence of edge positions in the network? Clear answers to this question are crucial if the results of community analyses are to carry any real weight.

The modularity itself was originally proposed as a way of answering this question [6]: a network with strong community structure will have high modularity and hence the value of the modularity can be used as a quality function for communities. More recently, however, it has been realized that this approach is insufficient. Although it is true that networks with strong community structure have high modularity, it turns out that not all networks with high modularity have strong community structure. Indeed, there exist networks that most observers would consider to have no community structure at all that nonetheless have high modularity. Guiméra *et al.* [29] showed numerically that divisions exist of ordinary random graphs that have high modularity, even in the limit of large network size, and subsequent analytic calculations by Reichardt and Bornholdt [30,31] produced similar results. The reason for this at first peculiar finding is actually quite straightforward: the number of possible divisions of a network increases extremely fast with network size (faster than any exponential), so that although it is highly improbable that any one division will, purely by chance, have high modularity, it is, in the limit of large size, very likely that such a division will exist among the enormous number of possible candidates. As a result, high modularity is only a necessary but not sufficient condition for significant community structure.

Several authors have suggested that instead we should look for divisions of a network that have significantly higher modularity than the random graph [29,31]. For example, one could optimize the modularity for a large number of networks drawn from the random graph ensemble, calculate the mean $\mu$ and standard deviation $\sigma$ of those modularity values, and then compare the modularity $Q$ of the optimal division of the real network to those values, calculating, for instance, a $z$ score,

$$z = \frac{Q - \mu}{\sigma}, \tag{1}$$

which measures how many standard deviations the real modularity is above the mean for the random graph. If $z \gg 1$ then $Q$ is, in a precise sense, significantly greater than the modularity of the random graph.

This approach, however, has a number of problems. First, it can generate both false positives and false negatives. Some networks that do not have strong community structure in the traditional sense nonetheless have modularity significantly above that of the random graph, as shown for example in [32]. Conversely, there are also some networks that are widely agreed to show strong community structure but whose modularity is not significantly greater than the random graph. We give some examples of this type of behavior later in this paper. (To be fair, such examples appear to be rare, so that a large difference in modularities may in some situations be considered supporting, though not conclusive, evidence of community structure.)

More importantly, however, the difference in modularities does not really address the question we want to answer. In this paper we argue that the defining property of significant community structure is not a high modularity, but a community structure that is robust against small perturbations of the network. If a small change in the network—an edge added here, another deleted there—can completely change the outcome of our community finding calculations then, we argue, the communities found should not be considered trustworthy. The $z$ score is not, in general, a good measure of this type of robustness or fragility in a network, but there exist other measures that, as we will show, appear to work well.

## II. ROBUSTNESS OF COMMUNITY STRUCTURE

An interesting approach to testing the significance of community assignments has been proposed by Massen and Doye [33], who investigated the distribution of modularity values for a variety of networks, both real and computer generated, using a simulated annealing technique similar to that of Reichardt and Bornholdt [21] combined with a parallel tempering scheme of the type commonly used to equilibrate simulations of glassy systems [34]. As a function of the annealing temperature they investigated (among other things) the average modularity of divisions found, with higher temperatures favoring poor divisions (low modularity) and lower temperatures favoring better ones (high modularity).

In low-temperature systems, where only states of high modularity are sampled, they found two distinct behaviors. In most real networks they found that the states sampled correspond to roughly the same division of the network into communities, while in random graphs the states sampled correspond to a variety of quite different divisions. This suggests that real-world networks typically have a clear global modularity maximum with no other competitive maxima, while random graphs have many competing maxima. In the language of physics, the distribution of maxima has a band gap between the ground and excited states in the real networks, but no band gap in the random graph. (One can also think of the system's behavior by analogy with glassy systems, which have many competing energy minima, and nonglassy ones, which typically do not. Indeed, ideas from the theory of spin glasses, in particularly replica symmetry, have proved useful in the study of modularity [31], suggesting that the difference between the community structure of random and real-world networks may be connected with the phenomenon of replica symmetry breaking.)

One can make use of this observation to identify community structure of the kind found in random graphs that occurs purely as a result of chance fluctuations: if we observe multiple modularity maxima in a network, corresponding to distinct community assignments and having roughly equal height, we can conclude that the assignments in question are not trustworthy. This approach will reliably rule out random graphs themselves—a basic task that any significance test must certainly be capable of—but it can in principle also rule out other cases and does so in a natural way, since any network that has many different community assignments of roughly equal merit can reasonably be said not to show clear community structure.

This approach provides only a way to rule *out* candidate assignments. It allows us firmly to reject some possibilities because of the structure of the modularity maxima, but we can never guarantee that an observed community assignment is significant solely on the basis of this test. Having multiple competing modularity maxima is a good indicator that the community structure given by the highest of those maxima is not trustworthy, but it is also possible that chance fluctuations could produce a network in which the highest maximum is substantially higher than any other even if the network has no underlying community structure. In this respect, the method is similar to other significance tests in statistics. Significance tests typically only reject hypotheses (or fail to reject them) but cannot absolutely confirm a hypothesis to be correct.

Massen and Doye proposed to implement tests of this kind by using their simulated annealing method to find all or a representative subset of the assignments having greatest modularity in a network and then see if they have similar community structure. Simulated annealing, however, is computationally costly and is usually not the optimization method of choice. And the approach of Massen and Doye cannot easily be generalized to other optimization methods, such as the spectral method. We therefore adopt a different approach based on network perturbations.

Small changes to a network—the addition or removal of a few edges, for example—will in general result in small changes to the value of the modularity for particular partitions of the network. In a network with many closely competitive modularity maxima, this can change the relative heights of the maxima with the result that the global optimum may shift from one maximum to another. In a network with only a single optimum, on the other hand, this cannot happen, prevented in effect by the presence of the band gap. Thus, a simple way to determine whether the network we are looking at has just a single optimum is to perturb the network slightly and observe the resulting change in the optimal partition.

This idea is the basis for our proposed method. In effect we turn the question of the significance of a division of a network into a question about the robustness of that division against perturbations, and the latter question can in practice be answered more easily. Our method also has the substantial advantage of being entirely agnostic about the way we discover our communities. We are not even required to use a modularity optimization technique—any technique that reliably finds community structure where present will do.

We are not the first to investigate the effects of perturbations on community structure. Wu and Huberman [35], for example, examined the effect on community structure of introducing randomness into a community finding algorithm, effectively perturbing the algorithm, in this case, rather than the network. An approach closer to our own was adopted by Gfeller *et al.* [36], who examined weighted networks and added noise to the edge weights, observing the resulting change in community structure. Perturbing the network instead of the algorithm frees us of the constraint of working with any one particular algorithm. In our work we are interested in unweighted networks and we perturb the network topology rather than edge weights, but the approach is conceptually similar to that of Gfeller *et al.*

## III. QUANTIFICATION OF NETWORK ROBUSTNESS

Our approach has two key components: perturbation of the network and quantification of the resulting change in the community structure. We describe these two components in turn.

### A. Network perturbation

We wish to specify a method for perturbing an arbitrary network by an arbitrary amount. In order to make comparison of communities straightforward, we restrict our perturbed networks to having the same numbers of vertices and edges as the original unperturbed network—only the positions of the edges will be perturbed. Furthermore, we desire that a network perturbed only a small amount has just a few edges moved, while a maximally perturbed network becomes completely random and uncorrelated with the original.

There are a number of ways in which this could be achieved but one of the simplest is the following. We define a random graph with $n$ vertices and $m$ edges in standard fashion by distributing the edges between vertex pairs such that the probability of any particular edge falling between vertices $i$ and $j$ is $e_{ij}/m$. This implies that the expected number of edges between $i$ and $j$ will be equal to $e_{ij}$. (Technically, the diagonal elements of $e_{ij}$ are different: they are equal to *twice* the expected number of edges—the extra factor of 2 allows for the fact that there are two ways of choosing a vertex pair if $i$ and $j$ are distinct, but only one way if $i$ and $j$ are equal.)

This definition still leaves us a good amount of freedom since we have not chosen the form of $e_{ij}$. Except for the constraint that the total number of edges equals $m$ so that $\frac{1}{2}\Sigma_{ij}e_{ij}=m$, we are at liberty to make any choice we wish, but the obvious candidate is the so-called configuration model, which is also the null model normally used in the definition of the modularity [6] and the random graph model against which values of the modularity are usually compared [31]. The expected number of edges between vertices in the configuration model is

$$e_{ij} = \frac{k_i k_j}{2m}, \tag{2}$$

where $k_i$ is the degree of vertex $i$ in the original network.

Now we interpolate stochastically between our original network and this random graph by "rewiring" (i.e., moving) edges. Specifically, we go through each edge in the original network in turn and with probability $\alpha$ we remove it and replace it with a new edge between a pair of vertices $(i,j)$ chosen randomly with probability $e_{ij}/m$. Otherwise, with probability $1-\alpha$, we leave the edge as it is.

If $\alpha=0$, no edges are moved and this process preserves our original network. If $\alpha=1$ all edges are moved and the process generates a random graph drawn from the model ensemble. And for values of $\alpha$ in between it generates networks in which some of the edges retain their original positions while others are moved to positions drawn from the random ensemble.

With the choice (2) for $e_{ij}$, the expected number of edges between vertices $i$ and $j$ in our perturbed network is

$$e'_{ij} = (1-\alpha)A_{ij} + \alpha\frac{k_i k_j}{2m}, \tag{3}$$

where $A_{ij}$ is an element of the adjacency matrix

$$A_{ij} = \begin{cases} 1, & \text{if an edge connects node } i \text{ and } j, \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

Then the expected degree of vertex $i$ is

$$\begin{aligned}\langle k_i \rangle &= \sum_j e'_{ij} = (1-\alpha)\sum_j A_{ij} + \alpha \frac{k_i}{2m}\sum_j k_j \\ &= (1-\alpha)k_i + \alpha\frac{k_i}{2m}2m = k_i,\end{aligned} \tag{5}$$

where we have made use of $\Sigma_j A_{ij}=k_i$ and $\Sigma_j k_j = 2m$.

Thus our perturbation scheme generates networks that not only have the same number of edges as the original, but in which the expected degrees of vertices are the same as the original degrees [52].

### B. Quantifying differences in community structure

The second component of our calculation is the comparison of the optimal division of the perturbed network to the optimal division of the original network, to see if the community structure has changed significantly. A number of methods for measuring similarities or differences between partitions of a network have been proposed in the past. They can be divided roughly into three groups: methods based on pair counting, methods based on cluster matching, and information theoretic methods. We begin by reviewing some of these before we discuss our choice, the variation of information. Our discussion follows that of Meila [37].

Let $C$ and $C'$ be two divisions of the same network into communities. We will refer to such divisions as *community assignments*.

Measures of the similarity or difference between two community assignments based on *pair counting* focus on the number of pairs of vertices that are in the same or different communities in both assignments. Such measures include the *Jaccard coefficient* and the *Rand index*. We define the following four numbers:

$$a_{00} = \text{pairs in different communities in both } C \text{ and } C',$$

$$a_{11} = \text{pairs in the same communities in both } C \text{ and } C',$$

$$a_{01} = \text{pairs in different (same) communities in } C(C'),$$

$$a_{10} = \text{pairs in same (different) communities in } C(C').$$

Then, for example, the unadjusted Rand index [38] is defined to be the ratio of the number of pairs clustered in the same way in both assignments to the total number of pairs thus:

$$R(C,C') = \frac{a_{11}+a_{00}}{a_{10}+a_{01}+a_{00}+a_{11}}. \tag{6}$$

The Rand index is also sometimes used in an adjusted form in which a null-model expectation value is subtracted from the unadjusted index to give a value that is axiomatically zero in the null model. Such adjusted indices have the disadvantage, however, of nonlocality [39]: the distance between two community assignments that differ only in one region of the network depends on how the rest of the network is partitioned.

An alternative approach is *cluster matching,* as embodied in measures such as the *van Dongen metric* and the *classification error*. These measures attempt to determine the best match for each cluster in $C$ to one of the clusters in $C'$. Suppose our two community assignments $C$ and $C'$ are composed of $K$ and $K'$ communities, respectively. The individual communities we will denote $C_1 \ldots C_K$ and $C'_1 \ldots C'_{K'}$. Then let $n_k$ and $n'_{k'}$ be the sizes of communities $C_k$ and $C'_{k'}$ and $n_{kk'}$ be the number of vertices common to communities $C_k$ and $C'_{k'}$ (i.e., $n_{kk'} = |C_k \cap C'_{k'}|$). Then the normalized van Dongen metric is defined by [40]

$$D(C,C') = 1 - \frac{1}{2n}\left[\sum_{k=1}^{K} \max_{k'} n_{kk'} + \sum_{k'=1}^{K'} \max_{k} n_{kk'}\right]. \tag{7}$$

Note that such measures ignore any subdivisions of a community that is never chosen as a match to a community in the other assignment. For example, suppose

$$C = \{\{a,b,c\},\{d,e,f,g\}\}, \tag{8a}$$

$$C' = \{\{a,b,c\},\{d,e\},\{f,g\}\}, \tag{8b}$$

$$C'' = \{\{a,b,c\},\{d\},\{e\},\{f,g\}\}. \tag{8c}$$

Under the van Dongen scheme $D(C,C')=D(C,C'')$, although many would claim (and most other measures agree) that $C$ is more similar to $C'$ than to $C''$.

A third class of measures for comparing community assignments is based on information theoretic ideas [41]. In measures such as these, we regard our community assignments as “messages” and consider the Shannon information content of these messages. The most common way to do this is to define $x_i$ to be the label of the community that vertex $i$ belongs to in $C$ and $y_i$ to be the community it belongs to in $C'$. Then the messages consist simply of the ordered sets $\{x_i\}$ and $\{y_i\}$. If one knows the joint distribution from which the $x$’s and $y$’s are drawn one can then calculate various standard information measures. The usual assumption is that the joint distribution is equal simply to that of the observed community assignments. In other words, $x$ and $y$ are assumed to be values of random variables $X$ and $Y$ with joint distribution $P(X=x,Y=y)=n_{xy}/n$, where $n$ is the total number of vertices in the network. This immediately implies also that $P(X=x)=n_x/n$ and $P(Y=y)=n'_y/n$.

In a slight abuse of terminology, we can then define the *mutual information* between the assignments $C$ and $C'$ to be equal to the mutual information between the corresponding random variables:

$$I(C;C') = I(X;Y) = \sum_{x=1}^{K}\sum_{y=1}^{K'} P(x,y)\log\frac{P(x,y)}{P(x)P(y)}, \tag{9}$$

where we use the shorthand notation $P(x)$ to denote $P(X=x)$ and similarly for the other distributions. [Within physics, researchers have traditionally used the natural logarithm in expressions such as Eq. (9), while in computer science the logarithm base 2 is more common. The choice makes only the difference of a multiplicative constant, however, and has no effect on any of our results.]

The mutual information measures how much information we learn about $C'$ if we know $C$. If $C$ and $C'$ are identical, then we learn everything about $C'$ from $C$. If they are entirely uncorrelated then we learn nothing. One way to express this is to make use of $P(x,y)=P(x|y)P(y)$ to write

$$\begin{aligned} I(X;Y) &= \sum_{xy} P(x,y)\log P(x|y) - \sum_{x} P(x)\log P(x) \\ &= H(X) - H(X|Y), \end{aligned} \tag{10}$$

where $H(X)$ is the information (or entropy) of $X$ and $H(X|Y)$ is the conditional entropy, i.e., the additional information needed to describe $X$ once we know $Y$. Thus if $Y$ tells us nothing about $X$ the two terms are equal and $I(X;Y)$ is zero. In essence the mutual information tells us the same thing as the conditional entropy, but the mutual information is symmetric in $X$ and $Y$ where the conditional entropy is not, which makes the former a more attractive measure of distance than the latter.

The mutual information alone, however, is not a good measure of the difference between our community assignments. Consider, for example, the three assignments of Eq. (8). In this case the conditional entropies $H(C|C')$ and $H(C|C'')$ are both zero, because given the community assignments $C'$ and $C''$ (and the appropriate mapping of community labels from one assignment to the other) we can deduce the assignment $C$. (The mapping of labels must be given, since the labels are arbitrary and we do not want our measure to register a difference between two assignments that in fact differ only in a permutation of the labels.) Therefore $I(C,C')=I(C,C'')=H(C)$ in this case, which is clearly not a useful answer. This problem is usually dealt with by normalizing the mutual information. There are a number of ways of accomplishing this but, for example, one can define

$$I_{\text{norm}}(C,C') = \frac{2I(C,C')}{H(C)+H(C')}. \tag{11}$$

A variant of this measure has been used by Danon *et al.* [42] to define standardized tests for the performance of community finding algorithms. Although the measure works, it is quite difficult to interpret, particularly in the normalized form, which makes it hard to give a simple statement about what the values mean (other than to say they get larger as community assignments become more similar).

### C. Variation of information

In our work we make use of a different information theoretic measure, the *variation of information* [37,39,43]. The variation of information is defined by

$$\begin{aligned} V(C,C') &= V(X,Y) \\ &= H(X) + H(Y) - 2I(X;Y) \\ &= H(X|Y) + H(Y|X) \\ &= -\sum_{xy} P(x,y)\log\frac{P(x,y)}{P(y)} - \sum_{xy} P(x,y)\log\frac{P(x,y)}{P(x)}. \end{aligned} \tag{12}$$

The variation of information is the sum of the information needed to describe $C$ given $C'$ and the information needed to describe $C'$ given $C$. It has a number of desirable properties that other measures lack. It is a true metric on the space of community assignments, having all the properties of a proper distance measure. It is also a local measure in the sense described above and it returns the intuitively correct answer for the example of Eq. (8), that $V(C,C'')>V(C,C')$.

The maximum value of the variation of information is $\log n$, which is achieved when the community assignments are as far apart as possible, which in this case means that one of them places all the nodes together in a single community while the other places each node in a community on its own. The maximum value increases with $n$ because larger data sets contain more information, but if this property is undesirable one can simply normalize by $1/\log n$, as we do in the calculations presented here. In fact, since we will always be comparing networks of the same size, the normalization is irrelevant anyway.

## IV. METHODS

We now have all the components we need to describe our method as applied to a given network. First, we find the community assignment $C$ that maximizes the modularity of the network, or the best approximation to it given the optimization algorithms available. Second, we perturb the network as described in Sec. III A to create a new network, find the optimal community assignment $C'$ for that perturbed network, and measure the variation of information between $C'$ and $C$. We repeat this second step many times to derive an average value for the variation of information, and repeat the entire calculation for a range of different values of the perturbation parameter $\alpha$. For comparison, we also perform the same sequence of calculations on a set of random graphs drawn from a configuration model with the same degree sequence as the original network.

The time required to complete the calculations depends on the method used to optimize the modularity, the number of random graph samples taken, and the number of different values of $\alpha$. In our calculations, as mentioned above, we use the spectral optimization method of [22], which is reasonably fast, though certainly not the fastest available, and average over 10 or 100 random graphs depending on network size for each of 40 different values of $\alpha$ from 0 to 1. The complete calculation for the largest network studied here, with nearly 5000 vertices, took about a day on a standard desktop computer.

For applications to larger networks there are a number of approaches one might take to reduce running time. First, we note that the method is trivially parallelizable, since it involves independent repetitions of the same calculations. Thus a linear speedup could be achieved on a parallel or distributed computer. Second, significant speed gains could be achieved by replacing the spectral modularity maximization

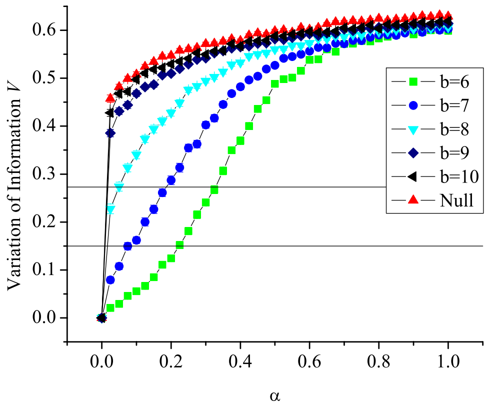

FIG. 1. (Color online) The variation of information as a function of the perturbation parameter $\alpha$ for the 128-node four-community test networks described in the text (100 networks per point).

used here with a faster method such as the near-linear greedy algorithm of Clauset *et al.* [17,44]. And third, if neither of these approaches is acceptable, improvements in running time could also be achieved by examining fewer values of $\alpha$. For the sake of clarity in presentation we have used a generous set of $\alpha$ values in our calculations, but this is not strictly necessary. The majority of the interesting information in most cases is contained in the low end of the curve of variation of information against $\alpha$. In an extreme case, even just a single value of $\alpha$ can yield significant insight.

## V. RESULTS

As a first demonstration of the method, we have applied it to a set of computer generated networks of a type proposed in [9] and used widely in the evaluation of community detection algorithms. These networks consist of 128 vertices divided into 4 communities of 32 nodes each. Each vertex pair is connected by an edge with one of two different probabilities, one for pairs in the same group and one for pairs in different groups, with values chosen so that the expected degree of each vertex remains fixed at 16. As the average number $b$ of between-group connections per vertex is increased from zero, the community structure in the network, stark at first, becomes gradually obscured until, at the point where between- and within-group edges are equally likely, the network becomes a standard Poisson random graph with no community structure at all.

Figure 1 shows the results of the application of our analysis method to graphs of this type. The figure shows the value of the normalized variation of information as a function of the parameter $\alpha$ that measures the amount of perturbation. As we can see, the variation of information starts at zero when $\alpha=0$, as we would expect for an unperturbed network, rises rapidly, then levels off as $\alpha$ approaches its maximum value of 1. Also shown is the curve for a random graph null model of the type described above.

For large values of $b$, such as $b=10$, the curve of the variation of information is essentially identical to that of the null model, indicating that whatever community structure has been found by the algorithm is no more robust against perturbation than that of a random graph. But as $b$ becomes smaller the variation of information increases slower as a function of $\alpha$ and the curves depart significantly from the null model, indicating that the community structure discovered by the algorithm is relatively robust against perturbation.

As an aid to the interpretation of the results, we have also included in the figure (and in all subsequent similar figures) horizontal lines corresponding to the value the variation of information would take if we were to randomly assign 10% and 20% of the vertices to different communities. The fact that the curves of variation of information cross these lines at larger values of $\alpha$ in some cases than others indicates that the community structure is more or less robust to perturbation. Indeed, one could simply quote the values of $\alpha$ at which the crossings occur as a single scalar measure of robustness, but to do so can mean missing interesting structure present in the full curves, so we have avoided this approach in our calculations.

Turning now to real-world networks, we have tested our method on a variety of examples including social, technological, and biological networks. A selection of results are shown in Fig. 2. Some summary statistics for the same networks are given in Table I.

Figure 2(a) shows the curve of variation of information as a function of $\alpha$ for one of the best studied examples of community structure in a social network, the "karate club" network of Zachary [45]. (The karate club has become so common an example in this context that it has almost come to the point where no publication about community structure could be complete if it failed to discuss this network.) The vertices in this network represent members of a karate club at a US university in the 1970s and the edges represent friendship between members based on independent observations by the experimenter. The network is widely believed to show strong community structure and repeated studies have upheld this view.

The black points (squares) in the figure show the variation of information for the real network while the red points (triangles) show the results for the equivalent random graph. It is clear in this case that the community structure discovered in the real network is substantially more robust against perturbation than that of the random graph. For example, the curve for the real network crosses the line representing reassignment of 20% of the vertices close to the point where $\alpha = 0.2$. Speaking loosely, we can say that about 20% of the edges must be rewired before 20% of the vertices move to different communities. For the random graph, on the other hand, only about 5% of the edges need be rewired to reach this point.

A contrasting situation is seen in Fig. 2(b), which shows results for another social network, a network of friendships among a group of first-year university students at the University of Groningen in the Netherlands [46]. Data for this network were collected by circulating questionnaires among members of the group; edges between pairs of students indicate that at least one member of the pair stated either that they were friends or that they had a "friendly relationship." Despite the similar nature of this network and the karate club

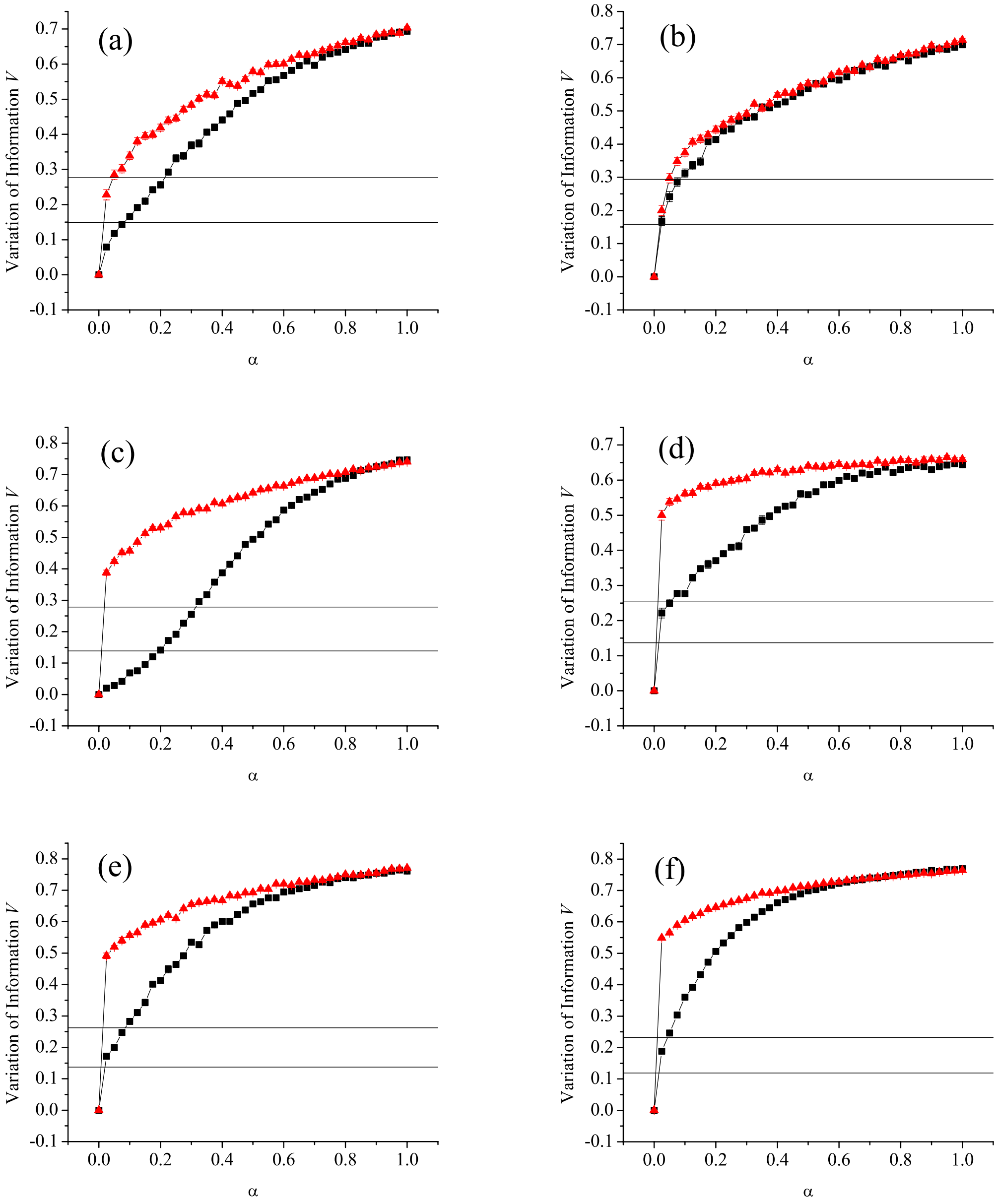

FIG. 2. (Color online) The variation of information as a function of the perturbation parameter $\alpha$ for six real-world networks as described in the text (black squares), along with equivalent results for the corresponding random graphs (red triangles). The figures are (a) Zachary's karate club, (b) social network of positive sentiments, (c) protein structure network, (d) metabolic network of *C. elegans*, (e) electronic circuit, and (f) power grid.

network (both are networks of friendship among university students), the results of the analyses are quite different. In the Groningen network, as Fig. 2(b) shows, there is essentially no difference between the variation of information for the real network and the corresponding random graph. The community structure algorithm does detect some structure in the network, finding four communities of sizes 5, 7, 9, and 11 vertices, respectively, and a respectable modularity score of 0.368, but our robustness analysis indicates that this structure is not significant and therefore should probably not be taken as indicative of the presence of any real communities in the network.

Our next two examples are both biological networks. The first [Fig. 2(c)] represents the structure of a protein (an immunoglobin), with the vertices representing $\alpha$ helices and $\beta$ sheets and an edge between any two that are less than 10 Å apart [47]. The second [Fig. 2(d)] represents known portions of the metabolic network of the nematode *C. elegans*, with vertices representing metabolites and edges representing metabolic reactions [48]. Again the two networks show contrasting behaviors. The community structure in the protein network displays substantial robustness against perturbations, with a wide gap between the variation of information curves for the true network and the random graph. A value of the variation of information equivalent to the randomization of 20% of the vertices is not reached until a perturbation strength of around $\alpha=0.3$. The metabolic network by contrast reaches the same point around $\alpha=0.05$, not much better

TABLE I. Maximum modularity and $z$ scores for each of the networks studied here. The first five lines of results are averages over computer-generated random networks as described in the text. The final six are real-world examples.

| Network | Modularity | $z$ score |
|---|---|---|
| Test $b=6$ | 0.373 | 21.0 |
| Test $b=7$ | 0.311 | 11.1 |
| Test $b=8$ | 0.248 | 2.63 |
| Test $b=9$ | 0.217 | −2.04 |
| Test $b=10$ | 0.210 | −2.99 |
| Karate club | 0.419 | 1.77 |
| University students | 0.368 | −0.19 |
| Protein structure | 0.763 | 24.5 |
| *C. elegans* metabolic | 0.434 | 25.4 |
| Electronic circuit | 0.805 | 31.2 |
| Power grid | 0.925 | 100.8 |

than the equivalent random graph. The curve of variation of information for the metabolic network does however remain distinct from that of the corresponding random graph for higher values of $\alpha$, indicating that some portion of the community structure found is relatively robust.

Our last two examples are technological networks, an electronic circuit [49,50] and a network representation of the power grid of the western United States [51]. Both of these networks show weak community structure similar to that of the metabolic network, with a variation of information that increases rapidly with $\alpha$ at first, indicating that much of the observed structure is quite fragile to perturbation, though the curves again remain distinct; we conclude that the networks show some community structure, even if the effects are not strong.

Now compare these results with those given in Table I. The final column of the table gives a $z$ score for each network calculated as described in the introduction [see Eq. (1)]. The comparison with the curves for variation of information is an interesting one. Five of the six networks have positive $z$ scores, but not all of the scores are large enough to make the results statistically significant. The most common rule of thumb is that measurements are significant if they lie more than two standard deviations from the mean of the null model, i.e., if $z>2$. By this rule, neither of our social networks have significant community structure, a surprising conclusion given that it is universally accepted that the karate club network has strong community structure, confirmed by repeated studies using many methods, and our variation of information calculation confirms this also. For the network of university students, on the other hand, the $z$ score and our calculations concur, both indicating that the community structure found is not significant, also a troubling result, since it implies that a low $z$ score may correspond either to strong community structure or to none at all.

The remaining four networks all have very large $z$ scores; the smallest of them is 24.5 and an observation 24 standard deviations from the mean will be considered significant by essentially any standard. Curiously, however, there seems to be little correlation between the $z$ scores and the robustness of the community structure. The highly robust protein structure network, for instance, has the lowest $z$ score of the four, while the power grid—one of the networks we concluded to have only rather weak community structure—has a spectacular $z=100.8$. Overall, therefore, it appears that while $z$ scores for modularity values probably do give some indication of the strength of community structure, they are in general unreliable and should not be trusted unless backed up by other calculations, such as those presented here.

## VI. CONCLUSIONS

In this paper we have examined measures of significance for network community structure that address the question of when communities found in a network can be considered believable, and could not reasonably have been the result of chance fluctuations in network structure. We have argued that high modularity scores, the conventional measure of significance, have less discriminatory power than measures that quantify the robustness of community assignments to network perturbation. We have proposed a method for perturbing networks and a measure of robustness under such perturbations based on the information-theoretic distance metric known as the variation of information. In applications to both real and computer-generated example networks, our method appears able to distinguish successfully and clearly between examples that show strong community structure and examples that do not.

In considering future directions for research, we note that all of the calculations presented here focus on the quality of partitions of an entire network. It is possible that there might be significant community structure in one part of a network and not in another, and were this the case one would like to be able to detect it. The methods described here could potentially be useful for this type of investigation: one can ask whether some communities in a network are robust under perturbation while others are not, although the global variation of information would not be suitable for this purpose and more detailed local measures would be needed. Some work along these lines has been pursued by Gfeller *et al.* [36] and it is possible that an approach similar to theirs would work in the present case.

We look forward to further developments in this area.

[1] M. E. J. Newman, Eur. Phys. J. B **38**, 321 (2004).

[2] L. Danon, J. Duch, A. Diaz-Guilera, and A. Arenas, J. Stat. Mech.: Theory Exp. (2005), P09008.

[3] S. Fortunato and C. Castellano, e-print arXiv:0712.2716.

[4] A. Arenas, A. Díaz-Guilera, and C. J. Pérez-Vicente, Phys. Rev. Lett. **96**, 114102(R) (2006).

[5] S. Boccaletti, M. Ivanchenko, V. Latora, A. Pluchino, and A. Rapisarda, Phys. Rev. E **75**, 045102(R) (2007).

[6] M. E. J. Newman and M. Girvan, Phys. Rev. E **69**, 026113 (2004).

[7] A. Arenas, J. Duch, A. Fernández, and S. Gomez, New J. Phys. **9**, 176 (2007).

[8] M. Newman, Proc. Natl. Acad. Sci. U.S.A. **103**, 8577 (2006).

[9] M. Girvan and M. E. J. Newman, Proc. Natl. Acad. Sci. U.S.A. **99**, 7821 (2002).

[10] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi, Proc. Natl. Acad. Sci. U.S.A. **101**, 2658 (2004).

[11] I. Derenyi, G. Palla, and T. Vicsek, Phys. Rev. Lett. **94**, 160202 (2005).

[12] G. Palla, I. J. Farkas, P. Pollner, I. Derenyi, and T. Vicsek, New J. Phys. **9**, 186 (2007).

[13] M. Rosvall and C. T. Bergstrom, Proc. Natl. Acad. Sci. U.S.A. **104**, 7327 (2007).

[14] M. B. Hastings, Phys. Rev. E **74**, 035102(R) (2006).

[15] A. Clauset, M. E. J. Newman, and C. Moore, in *Proceedings of the 23rd International Conference on Machine Learning* (Association of Computing Machinery, New York, 2006).

[16] M. E. J. Newman, Phys. Rev. E **69**, 066133 (2004).

[17] A. Clauset, M. E. J. Newman, and C. Moore, Phys. Rev. E **70**, 066111 (2004).

[18] J. Duch and A. Arenas, Phys. Rev. E **72**, 027104 (2005).

[19] R. Guimerà and L. A. N. Amaral, Nature (London) **433**, 895 (2005).

[20] A. Medus, G. Acuña, and C. O. Dorso, Physica A **358**, 593 (2005).

[21] J. Reichardt and S. Bornholdt, Phys. Rev. E **74**, 016110 (2006).

[22] M. E. J. Newman, Proc. Natl. Acad. Sci. U.S.A. **103**, 8577 (2006).

[23] T. Łuczak, in *Proceedings of the Symposium on Random Graphs, Poznań, 1989*, edited by A. M. Frieze and T. Łuczak (Wiley, New York, 1992), pp. 165–182.

[24] M. Molloy and B. Reed, Random Struct. Algorithms **6**, 161 (1995).

[25] S. Fortunato and M. Barthélmy, Proc. Natl. Acad. Sci. U.S.A. **104**, 36 (2007).

[26] J. M. Kumpula, J. Saramaki, K. Kaski, and J. Kertesz, Eur. Phys. J. B **56**, 41 (2007).

[27] A. Arenas, A. Fernández, and S. Gomez, e-print arXiv:physics/0703218.

[28] U. Brandes, D. Delling, M. Gaertler, R. Görke, M. Hoefer, Z. Nikoloski, and D. Wagner, in Proceedings of the 33rd International Workshop on Graph-Theoretic Concepts in Computer Science, Vol. 4769 of Lecture Notes in Computer Science, edited by A. Brandstädt, D. Kratsch, and H. Müller (Springer, Berlin, 2007).

[29] R. Guimerà, M. Sales-Pardo, and L. A. N. Amaral, Phys. Rev. E **70**, 025101(R) (2004).

[30] J. Reichardt and S. Bornholdt, Physica D **244**, 20 (2006).

[31] J. Reichardt and S. Bornholdt, Phys. Rev. E **76**, 015102(R) (2007).

[32] C. P. Massen and J. P. K. Doye, Phys. Rev. E **71**, 046101 (2005).

[33] C. P. Massen and J. P. K. Doye, e-print arXiv:cond-mat/0610077v1.

[34] D. J. Earl and M. W. Deem, Phys. Chem. Chem. Phys. **7**, 3910 (2005).

[35] F. Wu and B. A. Huberman, Eur. Phys. J. B **38**, 331 (2004).

[36] D. Gfeller, J.-C. Chappelier, and P. De Los Rios, Phys. Rev. E **72**, 056135 (2005).

[37] M. Meila, J. Multivariate Anal. **98**, 873 (2007).

[38] W. M. Rand, J. Am. Stat. Assoc. **66**, 846 (1971).

[39] M. Meila, in *ICML '05: Proceedings of the 22nd International Conference on Machine Learning* (ACM Press, New York, 2005), pp. 577–584.

[40] S. V. Dongen, Performance Criteria for Graph Clustering and Markov Cluster Experiments [CWI (Center for Mathematics and Computer Science), Amsterdam, 2000].

[41] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. (Wiley-Interscience, New York, 2006).

[42] L. Danon, A. Díaz-Guilera, J. Duch, and A. Arenas, J. Stat. Mech.: Theory Exp. (2005), P09008.

[43] M. Meila, University of Washington, Technical Report statistics No. 418, 2002 (unpublished).

[44] K. Wakita and T. Tsurumi, in Proceedings of the 16th International Conference on World Wide Web (ACM, New York, 2007), pp. 1275–1276.

[45] W. W. Zachary, J. Anthropol. Res. **33**, 452 (1977).

[46] M. A. J. van Duijn, E. P. H. Zeggelink, M. Huisman, F. N. Stokman, and F. W. Wasseur, J. Math. Sociol. **27**, 153 (2003).

[47] R. Milo, S. Iztkovitz, N. Kashtan, R. Levitt, S. Shen-Orr, I. AyzenShtat, M. Sheffer, and U. Alon, Science **303**, 1538 (2004).

[48] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A. L. Barabasi, Nature (London) **407**, 651 (2000).

[49] F. Brglez, D. Bryan, and K. Kozminski, in *Proceedings of the IEEE International Symposium on Circuits and Systems* (IEEE, Piscataway, NJ, 1989), Vol. 1929.

[50] R. F. Cancho, C. Janssen, and R. V. Sole, Phys. Rev. E **64**, 046119 (2001).

[51] D. J. Watts and S. H. Strogatz, Nature (London) **393**, 440 (1998).

[52] Note that the perturbed network may have a small number of isolated nodes. We do not discard these nodes, since that would make the perturbed network a different size from the original; instead we assign each isolated node to its own community.