Finding instabilities in the community structure of complex networks

David Gfeller,[1] Jean-Cédric Chappelier,[2] and Paolo De Los Rios[1]

[1]*Laboratoire de Biophysique Statistique, SB/ITP, Ecole Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland*

[2]*School of Computer and Communication Sciences, Ecole Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland*

(Received 30 March 2005; revised manuscript received 23 August 2005; published 29 November 2005)

The problem of finding clusters in complex networks has been studied by mathematicians, computer scientists, and, more recently, by physicists. Many of the existing algorithms partition a network into clear clusters without overlap. Here we introduce a method to identify the nodes lying "between clusters," allowing for a general measure of the stability of the clusters. This is done by adding noise over the edge weights. Our method can in principle be used with almost any clustering algorithm able to deal with weighted networks. We present several applications on real-world networks using two different clustering algorithms.

DOI: 10.1103/PhysRevE.72.056135 PACS number(s): 89.75.Hc, 05.10.−a, 89.20.Ff

I. INTRODUCTION

The framework of complex networks provides a remarkable tool for the analysis of complex systems consisting of many interacting entities [1,2], such as the Internet [3], the interaction map of proteins [4], social networks [5], etc. Historically a theoretical model to describe complex networks was the Erdös-Rényi graph [6]. However, this model fails to describe several features observed frequently in real-world networks. The two most famous ones are the degree distribution and the clustering coefficient. Recently different models have been proposed to give a more realistic understanding of those features [7,8].

Another characteristic of the topology of complex networks is their community structure: in real-world networks, it is common to have small sets of nodes highly connected with each other but with only a few connections to the rest of the network. Finding the clusters of a network is a crucial point in order to understand its internal structure. A large amount of clustering algorithms have been developed, each of them attempting to find a reasonably good partition of the network [9–13]. In most of the cases those algorithms partition the network into nonoverlapping clusters, assigning each node to a specific cluster ("hard-clustering"). However, the resulting clustering is sometimes questionable, especially for nodes that "lie on the border" between two clusters. We refer to such nodes as *unstable* nodes. Figure 1 shows a typical case where a node lies exactly between two clear clusters.

Defining and identifying unstable nodes is closely related to the problem of evaluating the stability of the clustering. It is indeed an important issue: since the exact solution of a clustering is generally not known, experimental perturbations are the only possible way to investigate how much a clustering algorithm is robust. An attempt was proposed by Wilkinson [14] by modifying the Girvan-Newman algorithm [9]. Recently several nondeterministic clustering algorithms have been developed [15–17]. Using the stochasticity of the output, one can probe the stability of the clustering. We finally mention the work of Palla *et al.* [18] where a clustering algorithm is designed based on the idea of clique percolation. This is an interesting, but fundamentally different answer to the problem of finding unstable nodes.

II. UNSTABLE NODES

In this work, we introduce a general method to find unstable nodes and evaluate the stability of the clusters. Instead of having a stochastic element in the algorithm, we propose to introduce stochasticity in the network itself and to use a hard-clustering algorithm. We used both the Markov clustering algorithm (MCL) [12] and the fast algorithm of Clauset *et al.* [19], but the method does not depend explicitly on these choices. The idea is to add a random noise over the edge weights in the network. In this study the noise added over the weight of the edge between nodes i and j, initially equal to w_{ij}, is equally distributed between $-\sigma w_{ij}$ and σw_{ij}, where σ is a fixed parameter, $0<\sigma<1$. Noise in this context is not only a useful tool to reveal cluster instabilities, but it has actually a deeper interpretation. In many real-world networks, edges are often provided with some intrinsic weight, but usually no information is given about the uncertainties over these values. Adding some noise could fill this lack, although arbitrarily, to take into account the possible effects of uncertainties.

In order to understand how the cluster structure changes with the noise, we compute the "*in-cluster probability*" p_{ij} of the edge between two adjacent nodes i and j. This probability is defined as the fraction of times nodes i and j have been classified in the same cluster during several occurrences of the clustering algorithm on different noisy realizations of the network. For example, in Fig. 1, node 7 has been classified 51% of the time in the same cluster as node 6 and 49% of the time in the same cluster as node 8 (p_{67}=0.51 and p_{78}=0.49). Edges with an in-cluster probability equal to one are always within a cluster and edges with an in-cluster probability close to zero connect two different clusters. We thus define edges with an in-cluster probability lower than a threshold θ as "*external edges*" (we typically choose θ=0.8).

Although it is sometimes visually obvious which nodes are unstable with respect to the cluster structure, the implementation of the algorithmic procedure has to be more precise. We start by removing all the external edges of the network, which gives a new, most of the time disconnected, network. Let us now call *initial clusters* the clusters obtained without noise, and *subcomponents* the disconnected parts of the network after the removal of the external edges. If the

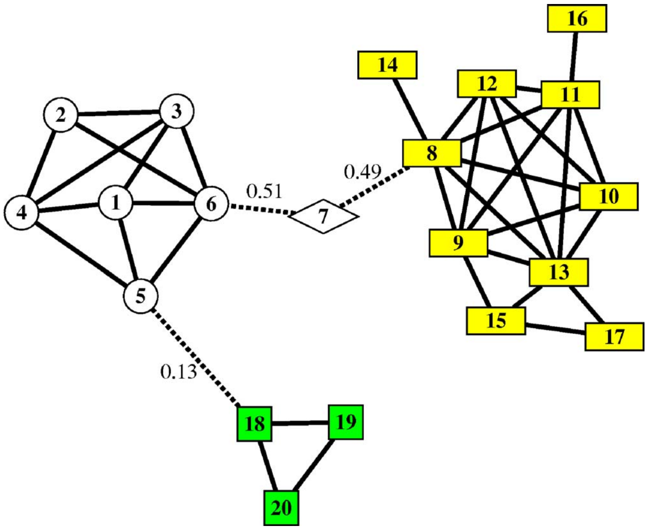

FIG. 1. (Color online) Small toy network with one unstable node (7). The clusters obtained without noise are labeled with different colors. Only in-cluster probabilities $p_{ij} < 0.8$ are shown (dashed edges). We have used MCL (described in Sec. IV) with $r = 1.6$, and $\sigma = 0.5$.

community structure of the network is stable under several repetitions of the clustering with noise, the subcomponents correspond to the initial clusters. In the opposite case, a new community structure appears with some similarities with the initial one. In order to find which subcomponents correspond to the initial clusters and which consist of unstable nodes, we introduce a similarity measure between two sets of nodes. If E_1 (E_2) is the set of initial clusters (the set of subcomponents), we use the following definition of the similarity (s_{ij}) between the initial cluster $C_{1j} \in E_1$ and the subcomponent $C_{2i} \in E_2$:

$$s_{ij} = \frac{|C_{2i} \cap C_{1j}|}{|C_{2i} \cup C_{1j}|}, \quad 1 \leq i \leq |E_2|, \quad 1 \leq j \leq |E_1|.$$

If $C_{1j} = C_{2i}$, $s_{ij} = 1$ and if $C_{1j} \cap C_{2i} = \emptyset$, $s_{ij} = 0$. For every $C_{1j} \in E_1$, we find the subcomponent C_{2i}, $1 \leq i \leq |E_2|$, with the maximal similarity and associate it with the initial cluster C_{1j} (most of the time C_{2i} corresponds to the stable core of the initial cluster C_{1j}). If there is more than one of such subcomponents, none of them will be associated with the initial cluster. In practice, this latter case is extremely rare.

For example, the network in Fig. 1 consists of three initial clusters (the three colors) and four subcomponents ({1,2,3,4,5,6}, {7}, {8,9,10,11,12,13,14,15,16,17}, {18,19,20}). Our method associates the three biggest subcomponents with the three initial clusters, while the subcomponent {7} is not associated with any cluster. In this example the same initial clusters and subcomponents are obtained with the two clustering algorithms mentioned at the beginning of this section ([12,19]).

Nodes belonging to subcomponents that have never been associated with any initial cluster will be defined as unstable nodes (those subcomponents do not always consist of only one node as in Fig. 1). We note that in some rare situations it can happen that a large initial cluster is split into two subcomponents of similar size. It is not reasonable to define one of these subcomponents as unstable and one would rather have the two subcomponents as two different clusters. This drawback can be avoided by setting a limit to the size of subcomponents made of unstable nodes.

III. CLUSTERING ENTROPY

Partitioning a network into clusters can be tricky since most of the algorithms will force a cluster structure, even on random networks where a cluster structure can arise from a purely stochastic process [20] and is thus meaningless. It is important therefore to define a suitable quantity able to assess the reliability and the robustness of the cluster structure obtained with a given algorithm.

Locally we can address the question of the stability of the clusters by looking at the in-cluster probabilities of the edges inside each cluster and around a cluster. For instance, if all edges inside the cluster have an in-cluster probability close to one and all edges connecting the cluster to its neighbors have an in-cluster probability close to zero, we can conclude that the cluster is rather stable. From a more global point of view, we propose the entropy as a measure of the stability of the cluster structure. In first approximation, we assume that the p_{ij} are independent of each other and we define the *clustering entropy* (CE) of edges as

$$S = \frac{-1}{m} \sum_{(i,j)} \{p_{ij} \log_2 p_{ij} + (1 - p_{ij}) \log_2 (1 - p_{ij})\},$$

where the sum is taken over all edges and m is the total number of edges in the network. If the network is totally unstable (i.e., in the most extreme case $p_{ij} = \frac{1}{2}$ for all edges), $S = 1$, while if the edges are perfectly stable under noise ($p_{ij} = 0$ or 1), $S = 0$.

The clustering entropy allows for comparing with a network without a predefined cluster structure. To avoid biasing the comparison, we shall always compare the CE of a network with the one of a randomized version of the network in which the degree of each node is conserved [21,22]. We first apply the randomizing step and then add the noise over the edge weights using the same σ as in the original network. The randomized network plays the role of a null-model since the clusters (if present) are destroyed by the randomizing process. Note, however, that we do not assume the randomized network to have no apparent community structure [20]. If the difference between the CE of the original network and the randomized one is important, it shows that the network has an internal cluster structure that differs fundamentally in terms of stability from a network where the nodes have been connected randomly. For this comparison, we have to restrict ourselves to unweighted networks since the rewiring process described in [21] is not designed for weighted networks.

IV. APPLICATIONS

Since it is not widely known in the physics community, we briefly describe MCL [12,23] that we used as one of the clustering algorithms, before showing applications of our

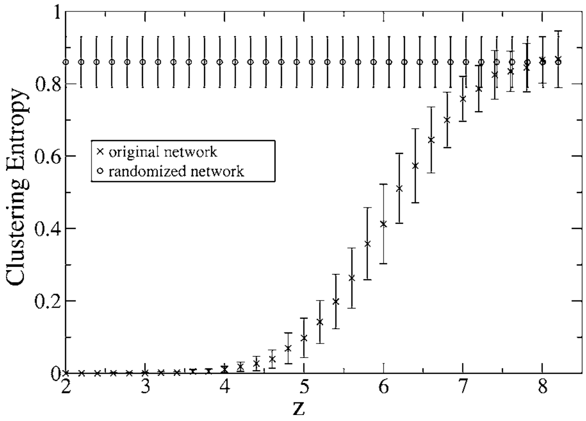

FIG. 2. CE as a function of z, the average number of edges connecting a node from a given cluster to nodes of other clusters, for a network with four communities of 32 nodes. The error bars represent the standard deviation for different networks with the same z. We have used MCL with $r=1.85$, and $\sigma=0.5$.

method to study the stability of the clusters. MCL is based on the idea that when *a random walk on a network visits a dense cluster, it will likely not leave it until many of its vertices have been visited*. However, the idea of performing a random walk on a network does not immediately lead to the clusters, since as time increases, the random walk will end up exploring the whole network. MCL favors the most probable random walks, already after a small number of steps, thereby increasing the probability of staying in the cluster. The algorithm works as follows: (1) take the adjacency matrix A of the network; add the self-edges (1's on the diagonal) and normalize each column of the matrix to one, in order to obtain a stochastic matrix W; (2) compute W^2; (3) take the rth power of every element of W^2 (typically $r \approx 1.5-2$) and normalize each column to one; and (4) go back to (2). After several iterations MCL converges to a matrix idempotent under steps (2) and (3). Only a few lines of the matrix have some nonzero entries that give the cluster structure of the network. Note that the parameter r can tune the granularity of the clustering. A small r corresponds to a few big clusters, whereas a big r to smaller ones. MCL is finding a growing consensus, especially in the bioinformatics community, thanks to its trade-off between speed, scalability, and adjustable granularity [12,24,25].

To illustrate the principle of the comparison based on the CE, we apply it on the well-known benchmark network introduced in [9]. The network consists of four communities of 32 nodes. The nodes are connected with a probability P_{in} if they belong to the same community and P_{out} if not. Typically one chooses to vary P_{in} and P_{out} keeping the average degree of the nodes constant. In Fig. 2 we plot the CE of the network. The parameter z is the average number of edges connecting a node from a given cluster to nodes of other clusters ($z=96P_{\text{out}}$). The average total degree is fixed at 16. The error bars stand for the standard deviation and give an indication of the dispersion of the values between different realizations of the network. When z is small the clusters are very well-defined and most of the algorithms correctly identify them. As z increases, the clusters become more and more fuzzy and for $z>7$ even the best currently available algorithms fail to recover the exact cluster structure of the network (actually the cluster structure tends to disappear from the network). This corresponds to the point from which the comparison of the CE does not allow one to differentiate between the network and a randomized one. We stress that the clustering entropy does not make reference to the assumed partition of the network into four clusters that, given the statistical nature of the links, cannot be guaranteed for every realization.

Let us now turn to real-world networks. As a first example, we consider the "karate club network" built by Zachary [26], where each node is an individual and edges represent social interactions. MCL correctly identifies the two communities, which correspond to the actual division of the club. The only unstable node is represented with a diamond (see Fig. 3). This node is connected to four nodes of one community and five of the other one. From a topological point of view, it is absolutely justified to consider it as an unstable node. It corresponds to an individual who still had contact with the two groups. The CE of the network is 0.14. The randomized network has an average CE of 0.27±0.1 (average and standard deviation of 100 randomized versions). Thus on average the CE is significantly larger for the randomized network.

We also studied a linguistic network based on the relation of synonymy in French [27]. The nodes are the words in a

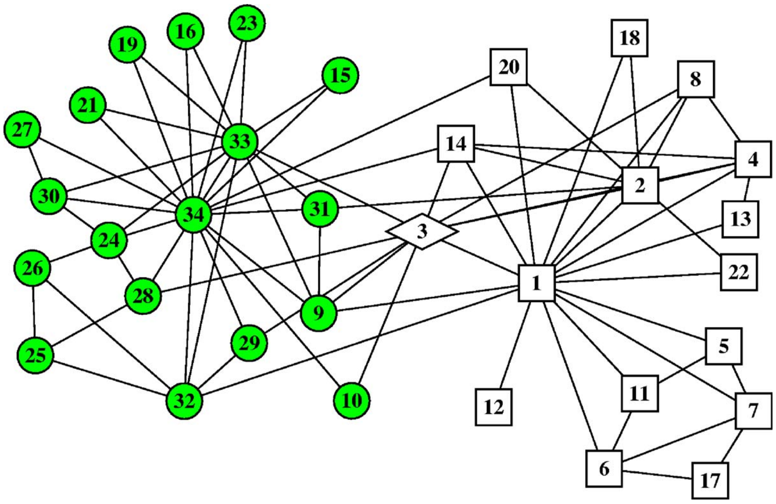

FIG. 3. (Color online) Zachary's karate club network. The two clusters are represented with two different colors. The unstable node is represented by a diamond. We have used MCL with $r=1.8$, and $\sigma=0.5$.

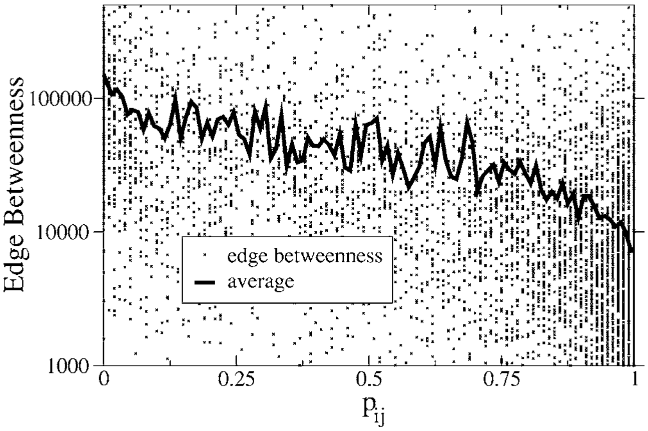

FIG. 4. Edge betweenness vs p_{ij} for a component of 9997 nodes from the synonymy network. $r=1.6$, $\sigma=0.5$.

given sense. Two nodes are connected if they are considered as synonyms. We applied MCL on the larger disconnected components of the network (up to 10 000 nodes per component) and found a much better lexical representation of the synonyms. The natural interpretation of unstable nodes in the case of a synonymy network is that they correspond to ambiguous words. As a validation of our results, we can measure the clustering coefficient and the betweenness centrality [28] of the unstable nodes. The clustering coefficient of a node i with degree k is defined as the number of 3-loops passing through i, divided by the maximal number of possible three-loops given by $\frac{1}{2}k(k-1)$ and the betweenness centrality of a node i is the number of shortest paths between all pairs of nodes that run through i. Averaging over the whole network, we have a clustering coefficient of 0.26 for the unstable nodes and 0.45 for the stable nodes and the betweenness centrality of unstable nodes is on average 1.6 times larger. The important difference was expected since unstable nodes often lie between clusters, and therefore usually do not have a large clustering coefficient, but have a large betweenness (typically node 7 in Fig. 1). Moreover, the plot of the edge betweenness [9] versus the in-cluster probability p_{ij} in Fig. 4 shows that external edges have on average a larger betweenness than the other edges, which is consistent with the Girvan-Newman clustering algorithm [9]. Yet, although this is true on average, it is not on a single edge basis.

Figure 5 shows how the CE varies with the parameter r of MCL for a component of 185 nodes from the linguistic network displayed in Fig. 6, compared with a randomized version of the same component. For $1.3<r<2$, the difference in behavior is striking. This shows that the clusters are not a by-product of the clustering algorithm, but correspond to a real community structure of the network.

We finally analyzed the protein folding network of the antiparallel β-sheet peptide developed by Rao and Caflisch [29]. The network is weighted, directed, and consists of almost 80 000 nodes. Due to the very large number of nodes we used the fast clustering algorithm of Clauset *et al.* [19]. The clustering algorithm correctly identifies the native state (or at least part of it) and other stable configurations such as the curl-like trap and the alpha-helix. In Fig. 7 we plot the network of the clusters. Each node corresponds to a cluster in the original network and two nodes are connected if there is at least one edge between the two clusters. The size of the nodes is related to the weight of the clusters and the size of the edges to the number of connections between the two clusters. As it can be seen, some of the configurations are represented by more than one cluster, which may indicate a substructure of the energy basins. Figure 7 shows also the possibilities offered by the clustering in order to visualize very large networks.

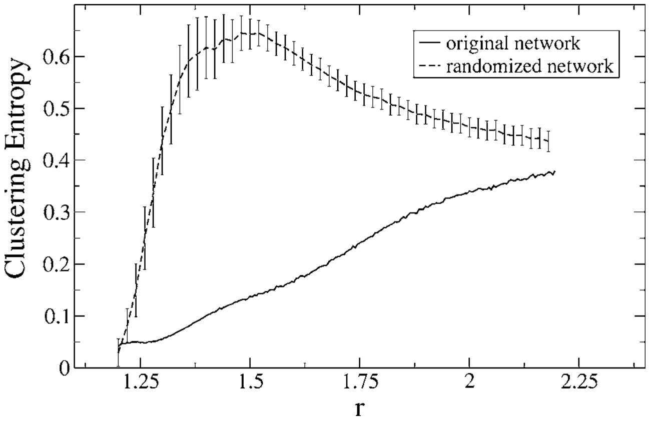

FIG. 5. CE as a function of the parameter r for a component of 185 nodes of the synonymy network. The dashed curve is the average over 50 randomized versions and the error bars correspond to the standard deviation. $\sigma=0.5$.

A possible interpretation for the unstable nodes (not represented in Fig. 7) is to consider them as transition states (as already suggested in [15]), although it is rather difficult to check our results since we do not know *a priori* which are the transition states. The unstable nodes have a clustering coefficient of about two-thirds of the average clustering coefficient of the whole network. The CE of the protein folding network is 0.08 with $\sigma=0.6$, which is low and seems to indicate that the clusters are rather robust to the noise. This was expected since the clusters often correspond to deep energy basins. In this case the comparison with a randomized network cannot be done since the network is weighted and directed, and considering it as unweighted induces important changes in the cluster structure.

V. DISCUSSION

Our method depends on two parameters, σ and θ. The parameter σ is directly related to the strength of the noise added to the network. With σ close to 0, we cannot detect the unstable nodes, while with σ close to one, the topology of the network changes dramatically. However, in many examples the results do not change significantly for a broad range of values of σ around 0.5. For instance, in the network displayed in Fig. 1, the node 7 was identified as the only unstable node for $0.15\leq\sigma\leq0.8$. If we want to strongly perturb the network or if the edge weights have a large intrinsic uncertainty, we should choose a rather high σ while if we only look for small perturbations we should choose a small

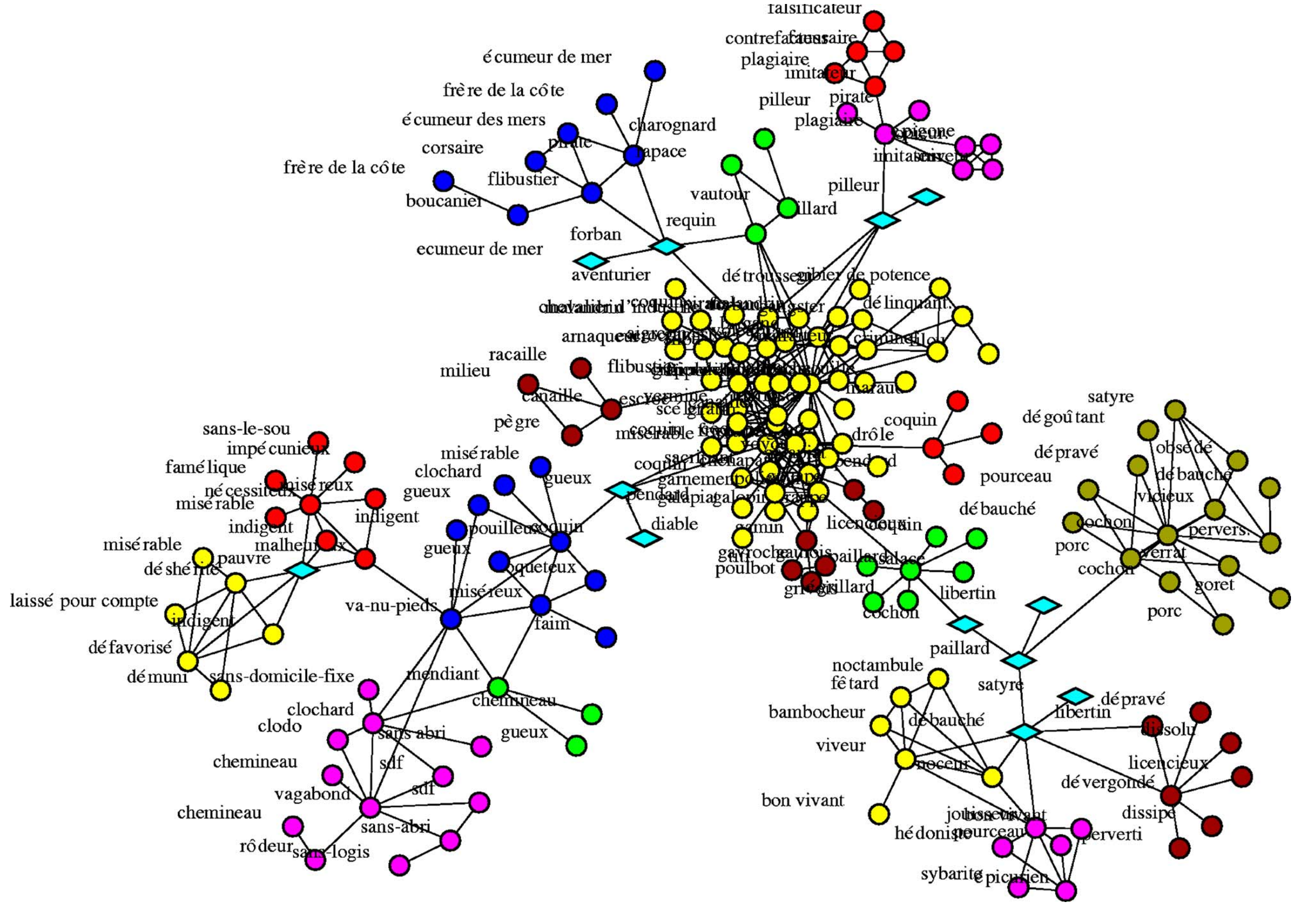

FIG. 6. (Color online) Component of 185 nodes of the linguistic network. The colors represent the different clusters found by MCL and the cyan diamonds are the unstable nodes. Some words may appear more than once since an initial distinction between different senses was sometimes already present in the original data, see [27]. $r=1.6$, $\sigma=0.5$.

σ. Moreover, very similar results are obtained using a Gaussian distribution for the noise.

Up to now we have always chosen a value of 0.8 for θ. The parameter θ has to be interpreted as a threshold such that two adjacent nodes that have been classified in the same cluster with an in-cluster probability higher than θ can be considered as belonging to the same cluster. For instance, if one wants to keep only edges that connect nodes with a very high confidence score, one should choose a rather large θ.

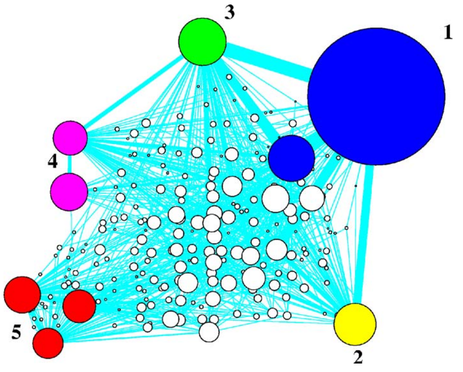

FIG. 7. (Color online) Network of the clusters of the protein folding network obtained with the clustering algorithm of Clauset *et al.* [19]. The size of the nodes is related to the total weight of the clusters and the size of the edges to the total weight of the edges between two clusters. The main clusters correspond to the following conformations. 1: native state, 2: the first hairpin is native, 3: the second hairpin is native, 4: curl-like trap, and 5: alpha-helix.

Our choice of $\theta=0.8$ was motivated by the following reason. θ should not be too close to 1 to avoid insignificant effects due to a peculiar noisy realization of the network, neither too close to 0.5, since if θ equals 0.5 the subcomponents basically correspond to the initial clusters (see Fig. 1). Thus a value of 0.8 is reasonable (as confirmed with small test networks). However, as for σ, other values of θ could be chosen without having an absolute criterion to decide which choice is the best one.

Finally the time-consuming step is the computation of p_{ij}, involving only the parameter σ, since we have to repeat the clustering several times. One can thus easily probe different values for θ without having to rerun the whole procedure.

VI. CONCLUSION

The introduction of the noise over the edges and the in-cluster probabilities p_{ij} provide a well-defined and objective way to identify unstable nodes and to deal with ambiguities in clustering. The method performs well on the small test networks presented above, and it can be applied on large real-world networks, using a fast clustering algorithm. Furthermore, it is straightforward to parallelize it in order to apply it to very large networks. As a validation of our results for large networks that can hardly be visualized, we have seen that the clustering coefficient of the unstable nodes is usually lower than the average clustering coefficient of the whole network. Moreover, these nodes have, on average, a larger betweenness, which is also expected for nodes lying between clusters. Nevertheless we could not have identified the unstable nodes only by comparing the clustering coefficient and the betweenness since very stable nodes may still have a large betweenness and a small clustering coefficient, and vice versa.

The CE allows for a quantitative comparison between a network and a null-model. We have found that in many examples the difference was clear, assuring that the clusters are neither the result of random fluctuations in the modularity of the network [20], nor an artifact of the clustering algorithm. Finally, since the method does not depend on a particular clustering algorithm, it can in principle be implemented using any clustering technique.

ACKNOWLEDGMENTS

This work was financially supported by the EU commission by contracts COSIN (FET Open IST 2001-33555) and DELIS (FET Open 001907) and the OFES-Bern under Contract No. 02.0234. We are thankful to M. E. J. Newman for providing the data of the Zachary's karate club network and to F. Rao and A. Caflisch for the protein folding network. We thank G. Salvi for a critical reading of the manuscript.

[1] R. Albert and A.-L. Barabási, Rev. Mod. Phys. **74**, 47 (2002).
[2] M. E. J. Newman, SIAM Rev. **45**, 167 (2003).
[3] M. Faloutsos, P. Faloutsos, and C. Faloutsos, Comput. Commun. Rev. **29**, 251 (1999).
[4] A. Vazquez, A. Flammini, A. Maritan, and A. Vespignani, Nature (London) **21**, 697 (2003).
[5] M. E. J. Newman and J. Park, Phys. Rev. E **68**, 036122 (2003).
[6] P. Erdös and A. Rényi, Publ. Math. (Debrecen) **6**, 290 (1959).
[7] A.-L. Barabási and R. Albert, Science **286**, 509 (1999).
[8] D. J. Watts and S. H. Strogatz, Nature (London) **393**, 440 (1998).
[9] M. Girvan and M. E. J. Newman, Proc. Natl. Acad. Sci. U.S.A. **99**, 7821 (2002).
[10] A. Capocci, V. D. P. Servedio, G. Caldarelli, and F. Colaiori, Lect. Notes Comput. Sci. **3243**, 181 (2004).
[11] M. Latapy and P. Pons, e-print cond-mat/0412368.
[12] A. J. Enright, S. Van Dongen, and C. A. Ouzounis, Nucleic Acids Res. **30**, 1575 (2002).
[13] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi, Proc. Natl. Acad. Sci. U.S.A. **101**, 2658 (2004).
[14] D. M. Wilkinson and B. A. Huberman, Proc. Natl. Acad. Sci. U.S.A. **101**, 5241 (2004).
[15] J. Reichardt and S. Bornholdt, Phys. Rev. Lett. **93**, 218701 (2004).
[16] R. Guimerà and L. A. N. Amaral, Nature (London) **433**, 895 (2005).
[17] J. Duch and A. Arenas, e-print cond-mat/0501368.
[18] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, Nature (London) **435**, 814 (2005).
[19] A. Clauset, M. E. J. Newman, and C. Moore, Phys. Rev. E **70**, 066111 (2004).
[20] R. Guimerà, M. Sales-Pardo, and L. A. N. Amaral, Phys. Rev. E **70**, 025101(R) (2004).
[21] S. Maslov and K. Sneppen, Science **296**, 910 (2002).
[22] K. A. Eriksen, I. Simonsen, S. Maslov, and K. Sneppen, Phys. Rev. Lett. **90**, 148701 (2003).
[23] S. Van Dongen, Ph.D. thesis, University of Utrecht, 2000, http://micans. org/mcl/
[24] C. Von Mering, E. M. Zdobnov, S. Tsoka, F. D. Ciccarelli, J. B. Pereira-Leal, C. A. Ouzounis, and P. Book, Proc. Natl. Acad. Sci. U.S.A. **100**, 15428 (2003).
[25] V. Prigent, J. C. Thierry, O. Poch, and F. Plewniak, Bioinformatics **21**, 1437 (2005).
[26] W. W. Zachary, J. Anthropol. Res. **33**, 452 (1977).
[27] D. Gfeller, J.-C. Chappelier, and P. De Los Rios, in *Proceedings of International Symposium on Applied Stochastic Models and Data Analysis (ASMDA'05)* (2005).
[28] M. E. J. Newman, Phys. Rev. E **64**, 016132 (2001).
[29] F. Rao and A. Caflisch, J. Mol. Biol. **342**, 299 (2004).