EPJ manuscript No.
(will be inserted by the editor)

Self-contained algorithms to detect communities in networks

Claudio Castellano[1], Federico Cecconi[2], Vittorio Loreto[1], Domenico Parisi[2] and Filippo Radicchi[3]

[1] Dipartimento di Fisica, Università di Roma "La Sapienza" and INFM-SMC, Unità di Roma 1, P.le A. Moro 5, 00185 Roma, Italy

[2] Istituto di Scienze e Tecnologie della Cognizione, C.N.R., Viale Marx, 15, 00137, Roma, Italy

[3] Dipartimento di Fisica, Università di Roma "Tor Vergata", Via della Ricerca Scientifica 1, 00133 Roma, Italy

Received: date / Revised version: date

Abstract. The investigation of community structures in networks is an important issue in many domains and disciplines. In this paper we present a new class of local and fast algorithms which incorporate a quantitative definition of community. In this way the algorithms for the identification of the community structure become fully self-contained and one does not need additional non-topological information in order to evaluate the accuracy of the results. The new algorithms are tested on artificial and real-world graphs. In particular we show how the new algorithms apply to a network of scientific collaborations both in the unweighted and in the weighted version. Moreover we discuss the applicability of these algorithms to other non-social networks and we present preliminary results about the detection of community structures in networks of interacting proteins.

PACS. 89.75.Hc Networks and genealogical trees – 87.23.Ge Dynamics of social systems – 87.90.+y Other topics in biological and medical physics

1 Introduction

The study of complex networks has become a the fast growing field in many different domains [1,2]. Examples range from technological systems (the Internet and the web [3,4]) to biological (epidemiology [5,6], metabolic networks [7–9], food webs [10–12]) and social systems [13,14] (scientific collaborations, structure of large organizations).

One of the problems that attracted a great deal of interest very recently is the identification of the so-called community structure. The concept of community is very common and it is linked to the classification of objects in categories for the sake of memorization or retrieval of information. From this point of view the notion of community is very general and, depending on the context, can be synonymous of module, class, cohesive subgroup, cluster, etc. Among the many contexts where this notion is relevant it is worth mentioning the problem of modularity in metabolic or cellular networks [9,16] or the problem of the identification of communities in the web [17]. This last issue is relevant for the implementation of search engines of new generation, content filtering, automatic classification or the automatic realization of ontologies.

Given the relevance of the problem it is crucial to construct efficient procedures and algorithms for the identification of the community structure in a generic network. This, however, is a highly nontrivial task.

Qualitatively, a community is defined as a subset of nodes within the graph such that connections between the nodes are denser than connections with the rest of the network. The detection of the community structure in a network is generally intended as a procedure for mapping the network into a tree (Fig. 1). In this tree (called dendrogram in social sciences) the leaves are the nodes while the branches join nodes or (at higher level) groups of nodes, thus identifying a hierarchical structure of communities nested within each other.

Several algorithms exist in literature to deal with this problem and in the next section we shall give a brief overview of them. A dendrogram, i. e. a community structure, is always produced by the algorithm down to the level of single nodes, independently from the type of graph analyzed. Typically no prescription is contained in the algorithms to discriminate between networks that are actually endowed with a community structure and those that are not. So typically one needs additional, non topological, information on the nature of the network to understand which of the branches of the tree have a real significance. Without such information it is not clear at all whether the identification of a community is reliable or not. Possible ways out of this problem has been proposed by Wilkinson and Huberman [18] (limited to the lowest level of the

---

Send offprint requests to: Claudio Castellano

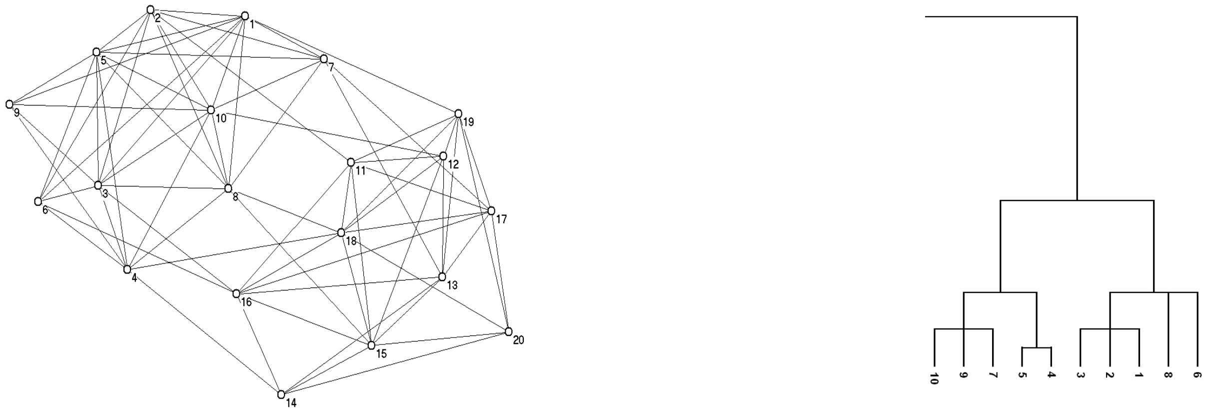

Fig. 1. A simple network (left) and the corresponding dendrogram (right).

community structure and specific to algorithms based on betweenness) and by Newman and Girvan [19] who have introduced an *a posteriori* measure of the strength of the community structure, the so-called modularity.

Another crucial issue about algorithms for the detection of community structures is the computational cost in time. This has stimulated the research about new and fast algorithms to solve the problem [20–22].

In this paper we follow the approach of [20] and we present a new class of self-contained algorithms which incorporate a quantitative definition of community. In this way the algorithms for the identification of the community structure become fully self-contained and one does not need additional non-topological information in order to evaluate the accuracy of the results. We test the performance of our algorithm on artificial and real-world graphs. In particular we show how the new algorithms apply to a network of scientific collaborations both in the unweighted and in the weighted version. Moreover we discuss the applicability of these local algorithms to other non-social networks and we present preliminary results about the detection of community structures in networks of interacting proteins.

The outline of the paper is as follows. In section 2 we present an overview of the existing algorithms to detect community structures. In section 3 we present the definitions of communities and show how these definitions can be implemented in a generic divisive algorithm in order to make it self-contained. In section 4 we present tests of the accuracy of our local algorithm (as compared to other algorithms) on some computer-generated and real networks. Section 5 is devoted to the extension of the new local algorithm to weighted graphs, focusing in particular on the network of scientific collaborations. Section 6 discusses the application of our algorithm to non-social networks. We consider in particular the example of a network of interacting proteins. We finally draw some conclusions in section 7.

2 Overview of the existing algorithms

When the size of the networks analyzed is small, it is relatively easy to check exhaustively on all network subsets whether they fulfilled some given definition of community. Nowadays, networks of many thousands or millions of vertices are investigated. It is clearly impossible to perform a complete analysis of such huge data sets. Attention has then been shifted toward a more limited goal, which nevertheless aims at extracting the key information about the community structure of a network. People have started to look for automated procedures to classify vertices in a network in a nested hierarchy of communities. The final output of such a procedure is a dendrogram, i. e. a tree which iteratively classifies the vertices (leaves) into groups, groups of groups and so on, up to the highest level (root), which contains the whole network. The intersection of the dendrogram with a line defines a subdivision of the graph into communities, which is coarser as the line is closer to the root.

There are many possible ways to build a dendrogram. Leaving for the final section the discussion about how to assess which dendrogram is best, we now concentrate on the different types of algorithms to construct a dendrogram.

Algorithms are of two types, depending on the order in which the dendrogram is built: agglomerative and divisive.

Agglomerative algorithms start from the single network vertices and group them iteratively. Traditional hierarchical clustering methods [23] belong to this class. They start by computing for each pair of nodes some quantity that measures how close the pair is in the network. Examples of such measures are suitably defined distances or correlations. Then nodes closer than a certain threshold are grouped and the procedure is iterated. A new type of agglomerative algorithm has been recently proposed by Newman [21]. At each step nodes or group of nodes are joined so that the pairing maximizes a quantity called modularity, which measures how much a division in communities is significantly different from a random one.

Opposite to agglomerative ones, divisive algorithms search for edges that are between different tightly-bound groups. By removing such edges, communities are singled out. Divisive algorithms are distinguished by the different quantities used to identify such edges between communities. Girvan and Newman (GN) [24] have recently shown that the edge betweenness is a useful quantity for this goal. To compute it one must determine the shortest path between any pair of nodes in the system [25]. The number of such shortest paths going through a certain edge is its betweenness. Clearly edges that connect dense regions of the graph tend to have higher betweenness values. Other divisive algorithms have been proposed by Zhou [26] and by Newman and Girvan [19]. As the GN algorithm, they focus on non local properties, related to diffusion of walkers on the networks. They tend to score worse than the GN method.

Very recently we have proposed a new type of divisive algorithm, which is instead based on a local property, related to the number of cycles that include a certain edge [20]. In the simplest case, when the cycles considered are triangles, the quantity one looks at is

$$\tilde{C}^{(3)}_{i,j} = \frac{z^{(3)}_{i,j} + 1}{\min[(k_i - 1), (k_j - 1)]}, \tag{1}$$

where $z^{(3)}_{i,j}$ is the number of triangles built on edge (i, j), $\min[(k_i - 1), (k_j - 1)]$ is the maximal possible number of them (k_i is the degree of node i). Imagine having two groups of nodes with a large number of connections inside them and only one link connecting them. Clearly in such case the number of triangles $z^{(3)}_{i,j}$ constructed on the edge *between* groups will be zero, while edges in the groups will have higher values of $z^{(3)}_{i,j}$. Hence we expect low values of the coefficient $\tilde{C}^{(3)}_{i,j}$ to characterize inter-community edges and higher values for edges that link nodes in densely connected regions. This expectation is confirmed empirically by the finding that in some networks edge betweenness and the coefficient $\tilde{C}^{(3)}_{i,j}$ are strongly anticorrelated [20]. Due to of the local nature of the quantity that is considered, the application of this algorithm requires a much shorter computational time than non local ones.

The definition of Eq. 1 can be straightforwardly generalized by considering higher order cycles. Coefficients of order g are defined as:

$$\tilde{C}^{(g)}_{i,j} = \frac{z^{(g)}_{i,j} + 1}{s^{(g)}_{i,j}}, \tag{2}$$

where $z^{(g)}_{i,j}$ is the number of cyclic structures of order g the edge (i, j) belongs to, while $s^{(g)}_{i,j}$ is the number of possible cyclic structures of order g that can be built given the degrees of the nodes. In this way a whole set of detection algorithms, smoothly interpolating between local and nonlocal features, can be defined [20].

3 Definitions of communities and self-contained algorithms

In the previous section we have briefly mentioned the different types of algorithm that, given a graph, construct a corresponding dendrogram. This task, however, is only the first step in the identification of the community structure of a network. This is made evident by considering the case of the Erdos-Renyi random graph [27]. By construction, such a network has no communities, since on average each group of nodes has the same density inside or outside it. However, the application of algorithms presented above always produces a dendrogram that reflects the small fluctuations with respect to the average pattern of connections.

In order to distinguish between random graphs and networks with robust community structure, we have to check whether the candidate communities detected by the algorithms are really such according to some quantitative and unambiguous definition. If the subgraph does not meet the criterion, it should not be considered as a community and the corresponding branch in the dendrogram should not be drawn.

Many definitions of community are given in the literature. Here we use two of the simplest possible of them, which correspond to rather intuitive prescriptions and are equivalent or very similar to other widely used [23].

The basic quantity we consider is k_i, the degree of a generic node i, which in terms of the adjacency matrix $A_{i,j}$ of the network G is $k_i = \sum_j A_{i,j}$. If we consider a subgraph $V \subset G$, to which node i belongs, we can split the total degree in two contributions:

$$k_i(V) = k_i^{in}(V) + k_i^{out}(V). \tag{3}$$

$k_i^{in}(V) = \sum_{j \in V} A_{i,j}$ is the number of edges connecting node i to other nodes belonging to V. $k_i^{out}(V) = \sum_{j \notin V} A_{i,j}$ is clearly the number of connections toward nodes in the rest of the network.

Definition of community in a strong sense

The subgraph V is a community in a strong sense if

$$k_i^{in}(V) > k_i^{out}(V), \ \ \forall i \in V. \tag{4}$$

In a *strong* community each node has more connections within the community than with the rest of the graph.

Definition of community in a weak sense

The subgraph V is a community in a weak sense if

$$\sum_{i \in V} k_i^{in}(V) > \sum_{i \in V} k_i^{out}(V). \tag{5}$$

In a *weak* community the sum of all degrees within V is larger than the sum of all degrees toward the rest of the network.

Clearly a community in a strong sense is also a community in a weak sense, while the converse is not true.

From the definitions given above, it is apparent that, if a network is randomly split in two parts, one very large and the other with only few nodes, the very large part almost always fulfils the definition of community. In order to deal with this problem, let us consider again the Erdös-Renyi random graph [27]. If we cut it at random in two parts containing αN and $(1-\alpha)N$ nodes, respectively, it is easy to evaluate analytically the probability $P(\alpha)$ that the subgraph containing αN nodes fulfils the weak or the strong definition.

To see this, let us consider the more general case of a graph of size N that we divide in N_{sub} sub-graphs V_j $(j = 1, ..., N_{sub})$ each of the same size $N_{in} = N/N_{sub}$. We insert edges in the graph with a probability p_{in} for pairs of nodes belonging to the same sub-graph (inward edges) and probability p_{out} of linking two nodes belonging to different sub-graphs (outward edges).

We start computing the probability that a sub-graph V_j satisfies the definition of community in the strong case. The probability that node $i \in V_j$ has m inward edges is given by

$$ {}^{in}P^{m}_{N_{in}-1} = C^{m}_{N_{in}-1} p^{m}_{in} (1-p^{m}_{in})^{N_{in}-1-m}. \tag{6}$$

where $C^{m}_{N_{in}-1}$ is the binomial coefficient. Analogously the probability that node $i \in V_j$ has n outward edges is given by

$$ {}^{out}P^{n}_{N_{out}} = C^{n}_{N_{out}} p^{n}_{out} (1-p^{n}_{out})^{N_{out}-n}, \tag{7}$$

where $N_{out} = N - N_{in}$. Therefore the probability that the node $i \in V_j$ has exactly n outward and m inward edges is

$$ W(m,n) = {}^{in}P^{m}_{N_{in}-1}\,{}^{out}P^{n}_{N_{out}}, \tag{8}$$

and the probability that node i belongs to the community V_j, in the sense of the strong definition, is

$$ P(i \in_{strong} V_j) = \sum_{n<m} W(m,n). \tag{9}$$

The probability that all nodes in subgraph V_j fulfil the strong condition is $P(i \in_{strong} V_j)^{N_{in}}$ (in the approximation that variables describing the presence of an edge between two nodes are independent) and the probability for all subgraphs to be communities is

$$ R_0 = \left[\sum_{n<m} W(m,n) \right]^{N_{in} N_{sub}}. \tag{10}$$

For the weak definition, a similar calculation yields

$$ R_0 = \left[\sum_{r<2s} \mathcal{W}(s,r) \right]^{N_{sub}} \tag{11}$$

where $\mathcal{W}(s,r) = {}^{in}P^{s}_{N_{in}(N_{in}-1)/2}\,{}^{out}P^{r}_{N_{in}N_{out}}$.

We can now use these results to compute the probability $P(\alpha)$ that a subgraph of αN nodes, randomly chosen in a random graph, fulfils the definitions. One simply sets $N_{in} = \alpha N$, $N_{out} = (1-\alpha)N$ and $p_{in} = p_{out} = p$. Then for the strong definition $P(\alpha) = \left[\sum_{n<m} W(m,n)\right]^{\alpha N}$, while for the weak definition $P(\alpha) = \left[\sum_{r<2s} \mathcal{W}(s,r)\right]$.

It is interesting to look at the limit for large N. For the weak definition the probability $P(\alpha)$ becomes

$$ P(\alpha) = \frac{1}{2}\left[1 + \mathrm{erf}(f(\alpha,N,p)\right], \tag{12}$$

with

$$ f(\alpha,N,p) = \sqrt{\frac{\alpha p}{2(1-p)}}(2\alpha - 1)N. \tag{13}$$

On the other hand, for the strong definition the probability $P(\alpha)$ becomes

$$ P(\alpha) = \left\{ \frac{1}{2}\left[1 + \mathrm{erf}(f(\alpha,N,p)\right] \right\}^{\alpha N}, \tag{14}$$

with

$$ f(\alpha,N,p) = \sqrt{\frac{p}{2(1-p)}}(2\alpha - 1)\sqrt{N}. \tag{15}$$

Hence in the limit $N \to \infty$, $P(\alpha)$ tends to a step function in both cases.

Therefore it is extremely likely that, in a random graph randomly cut in two parts, the largest one is a community according to the previous definitions. However it is extremely unlikely that both subgraphs fulfil simultaneously the definitions: therefore if we accept divisions only if both groups fulfil the definition of community, we correctly find that a random graph has no community structure. We extend this criterion to generic networks: if less than two subgraphs obtained from the cut satisfy the definitions, then the splitting is considered to be an artifact and disregarded. It is important to remark that the quantities appearing in Eqs. (4) and (5) must always be evaluated with respect to the full adjacency matrix.

In summary, in this section we have used some quantitative definitions of community to make a generic detection algorithm self-contained. This means that the procedure is fully automated, no parameter has to be tuned and the dendrogram produced consists only of subgroups that fulfil the definition of community chosen. It is straightforward to do the same using other quantitative definitions.

4 Results for unweighted networks

Figure 2 compares the computational times of our local algorithm with the ones of the GN algorithm. The task considered was the generation of the whole hierarchy of subgraphs of random graphs of increasing size N and fixed average degree $(M \sim N)$.

The new local algorithm presented in Section 3 is clearly faster than algorithms based on nonlocal quantities. In this Section we show, with some explicit examples, that it is also as accurate. More evidence can be found in Ref. [20]. One of the networks that have become a benchmark for the validation of algorithms to detect communities is the graph of acquaintances in the Karate Club studied by Zachary [28]. Results of the application of both the GN

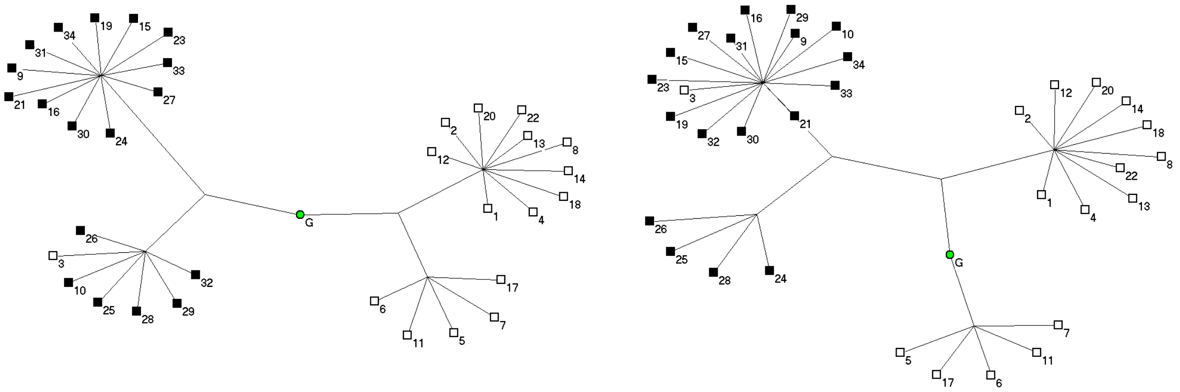

Fig. 3. Plot of the dendrograms for the Zachary's Karate Club network, obtained using the Girvan-Newman algorithm (left) and our local algorithm with $g = 4$ (right). Different symbols denote the individuals belonging to the two groups formed after the Club split.

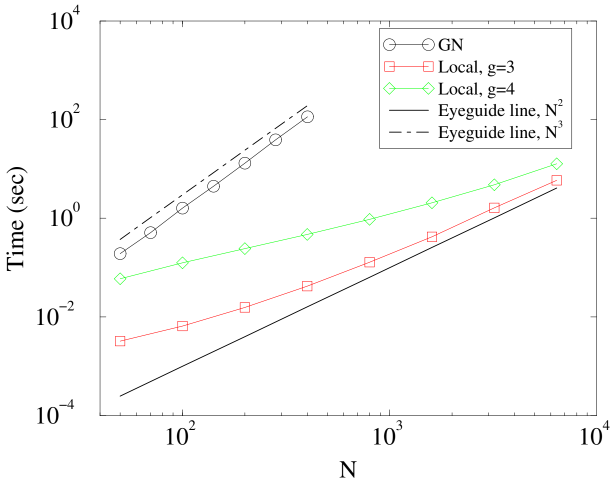

Fig. 2. Comparison of the computational times taken by our local algorithms (with $g = 3$ and $g = 4$) and by the GN algorithm to generate the whole hierarchy of subgraphs (putative communities) of random graphs. No criterion to validate the communities was imposed. While the GN algorithm features a scaling N^3 for the computational time, our local algorithms feature an asymptotic N^2 scaling with a much smaller prefactor.

and the new algorithm to such network, are presented in Fig. 3.

Some observations are in order. With both types of algorithm the first separation divides the set of nodes into two groups of roughly the same size. These two groups correspond to the eventual split of the club that followed a dispute between the manager and the trainer. Hence both algorithms perform well with respect to this main separation, since the first meaningful division into communities coincides with the pattern of the eventual breakup, with the exception of node 3, which is misclassified in both cases [29]. Interestingly, other separations in communities are found. In particular both main groups are in their turn subdivided into two sub-communities, with a remarkable overlap between the two methods. It is important to remark that the algorithms are self-contained: no additional non-topological information has been used. The structure found reflects only properties of the network topology.

The self-contained nature of the algorithms allows a more quantitative test of their accuracy. The benchmark is an artificial model with a well controlled community structure built in. It is the simple network with N nodes divided into N_{sub} groups already introduced in the previous section. As the probability p_{out} of outward connections grows from zero, the community structure in the network becomes less well defined. The analytical results presented in the previous Section are immediately applicable to this model. To test the algorithm we generate a large number of realizations of the artificial graph for several value of the probability p_{out}, keeping the average degree fixed. On each realization we apply the self-contained detection algorithm and construct the dendrogram. Then we count the fraction R_0 of times the dendrogram reproduces, at the lowest level, the 'perfect' community structure, i.e. N_{sub} communities each including exactly all the nodes. This is a quite sharp measure of success. The misclassification of a single node is considered to be a complete failure. With this prescription, formulas (10) and (11) give the expected value of R_0 for the strong or the weak definition of community, respectively.

Results are presented in Fig. (4) for $N = 120$ nodes and $N_{sub} = 6$ groups. They show that the algorithms are able to detect the community structure almost perfectly when the strong definition of community is used. The performance is not as satisfactory with respect to the weak definition. However the algorithms work better than what these plots seem to suggest. For small p_{out} it is actually the analytical estimate that is not correct, since it does not take into account the possibility that one or more of the N_{sub} communities could be, in their turn, formed by two sub-communities satisfying the definitions. This event happens for very small values of p_{out} and relatively small system sizes. For larger values of p_{out} the apparent poor

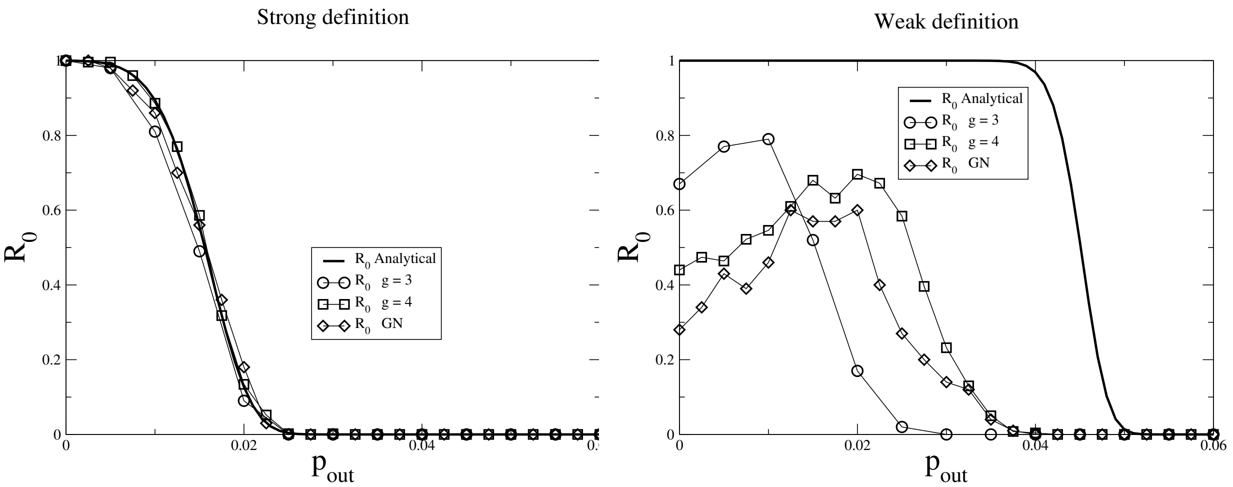

Fig. 4. Test of the efficiency of the different algorithms in the analysis of the artificial graph with six communities. The construction of the graph is described in text. Here $N = 120$ and p_{in} is changed with p_{out} in order to keep the average degree equal to 10. (Left) Strong definition: fraction of successes for the different algorithms compared with the analytical probability that six communities are actually defined. (Right) Weak definition: same quantities as in the left graph.

performance of the algorithm is due to the very restrictive definition of the success rate R_0. In the region where communities are still well defined (analytical R_0 close to 1) but R_0 for the algorithm is practically zero, what occurs is that the detection algorithms build almost perfect dendrogram with only few nodes misclassified. In any case Fig. (4) shows that the local and GN algorithms give comparable accuracy.

We then conclude that the local algorithm for the detection of the community structure performs as well as the GN in these controlled tests.

5 Extension to weighted networks

So far we have considered only unweighted (or *dichotomous*) networks, i. e. systems where all connections are exactly the same. Many interesting networks are instead weighted (or *valued*), i. e. each edge is characterized by a numerical value $A_{i,j}$ that indicates how strong the connection is. While it is always possible to neglect the weights as a first approximation in the analysis in the community structure of a network, it is clear that weights may carry crucial pieces of information. The community structure of a weighted dendrogram may be very different from the one of its unweighted counterpart.

The definitions presented in sections 2 and 3 must be suitably modified for considering weighted graphs. The generalization of the weak and strong definitions is straightforward. They remain formally the same, only the meaning of the quantities k_i^{in} and k_i^{out} changes. Rather than being the total number of connections (degree) toward nodes in the same subgroup or nodes in the rest of the graph, they measure the corresponding total weights.

We generalize the local algorithm for detecting communities by simply taking Eq. (1) and multiplying the number of triangles $z_{i,j}^{(3)}$ by the weight of the edge $A_{i,j}$. In this way strong edges will tend to have high values of $\tilde{C}_{i,j}^{(3)}$ and then will be cut later than edges with the same topological local configuration but smaller weight.

An example of nontrivial weighted graph is the network of scientific collaborations that can be inferred from papers on the cond-mat electronic repository [14]. The weight of an edge is proportional to the number of papers co-authored by the two scientists connected by it. When such a weighted network is analyzed using the local algorithm, one finds a dendrogram which differs in the details from the one obtained when the weights are neglected [20]. In particular the dendrogram is much deeper (i.e. it features a larger number of generations) and there is a larger number of small communities with size of the order of 10 nodes or less. This is evident in Fig. (4) where the number of communities of size s is plotted versus s. Apart from the increase of small communities it turns out clearly that the power law decay with exponent 2 [15] is the same with or without weight.

6 Social vs. non-social networks

Up to now we have only discussed examples of the so-called social networks. It has been shown [30] how social networks differ substantially from other types of networks, namely technological or biological networks. The origin of the difference has been shown being twofold. On the one hand they exhibit a positive correlation between adjacent vertices (also called assortativity) while most of the other non-social networks [31–33] are disassortative. On the other hand social networks show clustering coefficients well above those of the corresponding random models. In [30] Newman proposed that these differences could be explained by the presence of a community structure.

If these results are consistent with our and other's findings about community structures in social networks, the

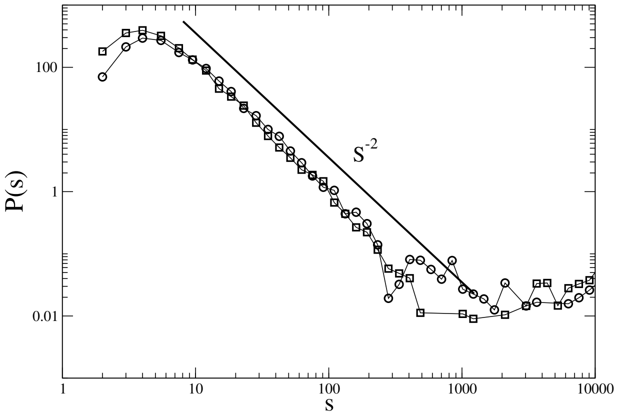

Fig. 5. Size distribution of all the communities of scientists identified in a weak sense by the algorithm described in section 2 for the unweighted (circles) and the weighted (squares) graphs of scientific collaborations. In both cases the behaviour is well reproduced by a power law with exponent -2

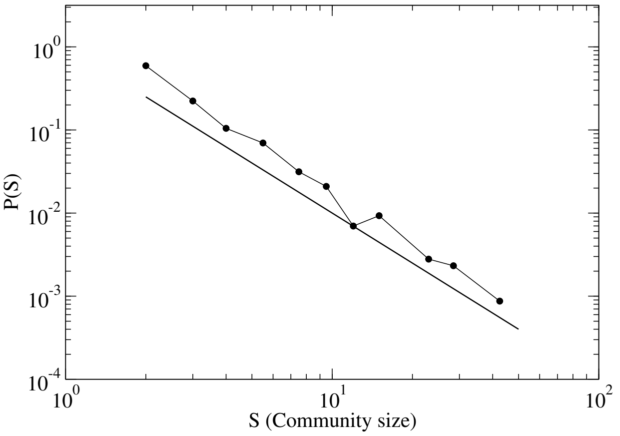

Fig. 6. Normalized size distribution of all the communities of proteins identified in a weak sense by the algorithm described in section 2 for $g = 3$. The straight line has slope -2.

question concerning the existence of a community structure in non-social networks remains wide open.

From the perspective of our local algorithm, an interesting insight comes from the study of the loops of arbitrary order [34,35]. In particular in [34] Caldarelli et *al.* study the statistics of cycles of order four (equivalent to the squares in our terminology) for four different types of networks, two of them social (and assortative) and two non-social (and disassortative). In all cases they report values for the so-called average grid coefficient (the extension of the concept of clustering coefficient to cycles of order four) which are *two to four order of magnitude larger than the corresponding coefficients of a random graph with the same average degree and size N*. They conclude arguing in favor of the presence of some sort of hierarchical structures and well-defined communities.

Given the above considerations we think it is important to check the outcomes of our algorithm, focused on the existence of local cycles of generic order n, also to non-social (and disassortative) networks.

A detailed analysis of this issue is beyond the scope of the present paper and we refer the reader to a forthcoming publication. Nevertheless we present some preliminary results obtained analyzing a network of protein interactions. We have analyzed in particular the Yeast subset of the Database of Interacting Proteins (DIP [36]) which contains all the pairs of interacting proteins identified in the budding yeast, *Saccharomyces cerevisiae.* In our dataset 4746 proteins were included with more than 15000 interactions.

We applied our algorithm to identify communities to this network and we obtained the corresponding trees where only the communities satisfying the weak or the strong definitions are drawn. An emerging difference with respect to social networks analyzed in the previous sections is that the giant component of the network is not split progressively in smaller and smaller subgroups, but only very small communities separate. In Figure 6 we report the size distribution of the small communities the algorithm recognized in a weak sense. Also in this case the distribution is compatible with a power-law with exponent -2.

The picture emerging from this analysis seems to point in the direction of the existence of modules in the network, i.e. groups of proteins that are highly interconnected with each other but with a few links outside of the module and carrying out some biological function. In [9] it has been proposed the concept of hierarchical modularity to reconcile a scale-free topology for a network with the existence of modules. This picture has been confirmed in [37] where the scaling of the clustering coefficient has been studied for several databases of interacting proteins. In this perspective it is tempting to associate the concept of community (as emerges from our or equivalent algorithms) to that of module.

In order to have a zero-th order check of this hypothesis it is important to check whether the communities our algorithm identifies are composed by group of proteins with a known equivalent functional classification. With this purpose in mind we utilized the functional classification provided by the MIPS database (Munich Information Center for Protein Sequences [38]) and reported in Table 1.

Since each protein of the DIP database can belong to more than one MIPS functional class, for each protein i belonging to the community k we have defined a vector $\boldsymbol{v_i^k}$ where each of the 18 components (one for each potential functional class) is zero when the protein does not belong to the corresponding functional class and it is equal to the MIPS code when it does. We can define in this way the an *overlap index* for each community, defined as:

$$\delta_k = \sum_{i=1}^{N_k} \sum_{j=i+1}^{N_k} \frac{2}{N_k(N_k-1)} \boldsymbol{v_i^k} \cdot \boldsymbol{v_j^k}, \tag{16}$$

where N_k is the number of proteins (not being *not yet clear-cut* (98) or *unclassified* (99)) belonging to the k-th community, and the scalar product $\boldsymbol{v_i^k} \cdot \boldsymbol{v_j^k}$ is defined as the ratio of the number of common components (different

Table 1. Functional Classification of proteins according to the MIPS database [38]

MIPS code	Functional Category
01	Metabolism
02	Energy
03	Cell cycle and DNA processing
04	Transcription
05	Protein synthesys
06	Protein fate
08	Cellular Transport and Transport Mechanisms
10	Cellular communication/signal transduction
11	Cell rescue, defense and virulence
13	Regulation/Interaction with cellular envir.
14	Cell fate
29	Transposable elements, viral and plasmid
30	Control and Cellular organization
62	Protein activity regulation
63	Protein with binding function
67	Transport facilitation
98	Classification not yet clear-cut
99	Unclassified proteins

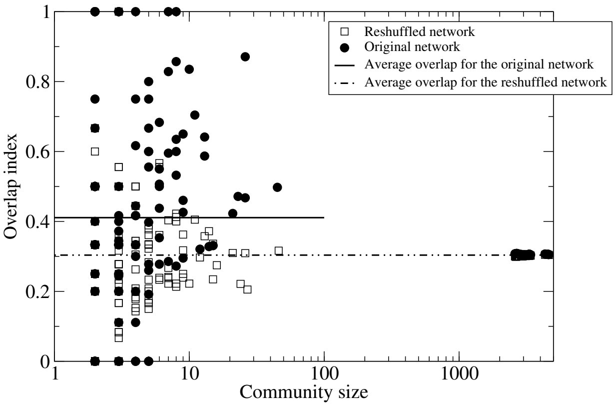

Fig. 7. Overlap index for the communities of proteins found by our local algorithm with $g = 3$. See text for details

from 0, 98 and 99) of the two proteins to the total number of different components they express. The expression (16) provides then, for each community, with a quantification of the homogeneity of the different proteins of the community with respect to the functional classification. In Fig. 7 we report the results for the overlap index as a function of the community size, for the communities identified by our algorithm (for $g = 3$) in a weak sense. For comparison we report, for each community, the same quantity computed in a reshuffled network obtained keeping exactly the same topology but reshuffling the labels of the proteins on each node.

It is evident how the communities found by our algorithm are strongly correlated from the functional point of view with respect to the corresponding random case. Notice that in the reshuffling we have restricted ourselves to the case where the identity of the proteins is maintained, i.e. we have not changed the ensemble of classes of belonging for each protein. A random assignment of a certain number of classes of belonging (always keeping the original probability distributions) to each protein would have led to much smaller values of the overlap index for the random case.

With our procedure of reshuffling, the average value of the overlap index for the communities smaller than 100 elements gives a value $\langle\delta\rangle_{random} \simeq 0.30 \pm 0.01$ for the random case and a value $\langle\delta\rangle \simeq 0.41 \pm 0.03$ for the normal case. $\langle\delta\rangle_{random}$ coincides with the value of the overlap index for the original giant component.

Following the same procedure one could check the homogeneity of each community with respect to other factors like localization of the proteins, belonging to complexes, etc. This work is still in progress.

7 Discussion and conclusions

In this paper we have discussed the introduction of two new ingredients in the algorithms for the detection of the community structure of networks. On the one hand we have presented a new type of divisive algorithm aimed at the identification of densely connected subgroups in a generic network. Such method is based on the computation of local quantities and therefore is fast and can easily be applied to large data sets. On the other hand we have pointed out a way to render any detection algorithm self-contained, i.e. an automated procedure to build a dendrogram such that each group in it fulfils an unambiguous definition of community. We have then checked the performance of the local algorithm on some networks with known community structure, extended the considerations to the case of weighted graphs and finally presented some results for a network of protein interactions, as a test of the performance of the new method for non social systems.

From the results presented, one can conclude that the local algorithm for detecting communities works well in the cases where a large number of short cycles (triangles, squares, and so on) is present in the network. This is the case of assortative networks. For non social networks, which tend to be disassortative, the algorithm tends to perform less well, and in particular many of the nodes in the graph are not associated to relatively small communities. While this is clearly a shortcoming of the algorithm, nevertheless the community structure found bears some significance also in these cases, as shown by the correlation between topological communities and functional classes in the protein interaction network.

In the end, let us remark that at present the question whether a detection algorithm is better than another one is not well posed. In all recent works about this problem, in order to compare the results of the application of an algorithm people have resorted to test on a limited number of small networks for which the 'true' community structure is in some way known. However, this is far from being systematic and objective. Only the introduction of precise measures of the quality of a dendrogram would allow establishing which of the many available algorithms is best.

Acknowledgments We thank Mark Newman for providing us the data for the social networks we have analyzed. We wish to thank Paolo De Los Rios for interesting discussions about the networks of interacting proteins, Alain Barrat and Guido Caldarelli for very stimulating discussions.

References

1. Albert, R. & Barabási, A.-L. (2002), Rev. Mod. Phys. **74**, 47.
2. Newman, M. E. J. (2003), SIAM Review **45**, 167.
3. Albert, R.,Jeong, H. & and Barabási, A.-L. (1999), Nature **401**, 130.
4. Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A. & Wiener, J. (2000), Computer Networks **33**, 309.
5. Moore, C. & Newman, M. E. J. (2000), Phys. Rev. E **61**, 5678.
6. Pastor-Satorras, R. & Vespignani, A. (2001), Phys. Rev. Lett. **86**, 3200.
7. Jeong, H., Tombor, B., Albert, R., Oltvai, Z.N. & Barabási, A.-L. (2000), Nature **407**, 651.
8. Wagner, A. & Fell, D. (2001), Proc. R. Soc. London B **268**, 1803.
9. Ravasz, E., Somera, A., Mongru, D. A., Oltvai, Z. N. & Barabási A.-L., (2002), Science **297**, 1551.
10. Dunne, J. A., Williams, R. J. & Martinez, N. D. (2002), Proc. Natl. Acad. Sci. USA **99**, 12917.
11. Camacho, J., Guimerà, R. & Amaral, L. A. N. (2002), Phys. Rev. Lett. **88**, 228102.
12. Garlaschelli, D., Caldarelli, G. & Pietronero, L. (2003), Nature **423**, 165.
13. Redner, S. (1998), Eur. Phys. J. B **4**, 131.
14. Newman, M. E. J. (2001), Proc. Natl. Acad. Sci. USA **98**, 404.
15. De Los Rios, P. (2001), Europhys. Lett. **56**, 898.
16. Rives, A. & Galitski, T. (2003), Proc. Natl. Acad. Sci. USA **100**, 1128-1133.
17. Flake, G. W., Lawrence, S. R., Giles, C. L. & Coetzee, F. M. (2002), IEEE Computer **35(3)**, 66-71.
18. Wilkinson, D. & Huberman, B.A. (2002), arXiv:cond-mat/0210147.
19. Newman, M.E J. & Girvan, M. (2003), cond-mat/0308217.
20. Radicchi, F., Castellano, C., Cecconi, F., Loreto, V. & Parisi, D. (2003) arXiv:cond-mat/0309488.
21. Newman, M.E.J. (2003) arXiv:cond-mat/0309508.
22. Wu, F. & Huberman, B.A. (2003), arXiv:cond-mat/0310600.
23. Wasserman, S. & Faust, K. (1994), Social Network Analysis (Cambridge Univ. Press, Cambridge, U.K.).
24. Girvan, M. & Newman, M. E .J. (2002), Proc. Natl. Acad. Sci. USA **99**, 7821.
25. Newman, M.E.J. (2001), Phys. Rev. **64**, 016131; Brandes, U. (2001), Journal of Mathematical Sociology **25**, 163.
26. Zhou, H. (2003), Phys. Rev. E **67**, 061901.
27. Bollobas, B. (1985), Random Graphs (Academic Press, New York).
28. Zachary, W. W. (1977), J. Anthrop. Res. **33**, 452.
29. The comparison between the community structure in the network and the splitting of the club in two groups should be taken with care. While it is reasonable that the break-up patternis correlated with the preexisting community structure, there is no reason to expect a perfect overlap. Therefore the *misclassification* of nodes is not an indication of the failure of the detection algorithm.
30. Newman, M.E.J. & Park, J. (2003) Phys. Rev. E **68**, 036122.
31. Pastor-Satorras, R., Vásquez, A. & Vespignani A. (2001) Phys. Rev. Lett. **87**, 258701 (2001).
32. Chen, Q., Chang, H., Govindn, R., Jamin, S., Shenker, S.J. & Willinger, W. (2002) Proceedings of the 21st Annaul Joint Conference of the IEEE Computer and Communications Societies, IEEE Computer Society.
33. Newman, M.E.J. (2003) Phys. Rev. E. **67**, 036126.
34. Caldarelli, G., Pastor-Satorras, R. $ Vespignani A. (2002) arXiv:cond-mat/0212026.
35. Bianconi, G. & Capocci, A. (2003) Phys. Rev. Lett. **90**, 078701.
36. Xenarios, I., Rice, D.W., Salwinski, L., Baron, M.K., Marcotte, E.M. & Eisenberg D (2000) Nuceli Acids Res. **28**, 289. The database is available at the web site `http://dip.doe-mbi.ucla.edu/`.
37. Yook, S.-H., Oltvai, Z.N. & Barabási, A.-L. (2003) accepted for publication in Proteomics.
38. Mewes, H.W., Frishman, D., Guldener, U., Mannhaupt, G., Mayer, K., Mokrejs, M., Morgenstern, B., Munsterkotter, M., Rudd, S. & Weil B.(2002) Nucleic Acids Res **30**, 31. The database is available at the web site `http://mips.gsf.de/`