Stable community cores in complex networks

Massoud Seifi, Jean-Loup Guillaume, Ivan Junier, Jean-Baptiste Rouquier, and Svilen Iskrov

Abstract Complex networks are generally composed of dense sub-networks called communities. Many algorithms have been proposed to automatically detect such communities. However, they are often unstable and behave non-deterministically. We propose here to use this non-determinism in order to compute groups of nodes on which community detection algorithms agree most of the time. We show that these groups of nodes, called community cores, are more similar to Ground Truth than communities in real and artificial networks. Furthermore, we show that in contrary to the classical approaches, we can reveal the absence of community structure in random graphs.

1 Introduction

Complex networks appear in various contexts such as computer science (networks of Web pages, peer-to-peer exchanges), sociology (collaborative networks), biology (protein-protein interaction networks, gene regulatory networks). These networks can generally be represented by graphs, where vertices represent entities and edges indicate interactions between them.For example, a social network can be represented by a graph whose nodes are individuals and edges represent a form of social relationship. Likewise, a protein-protein interaction network can be modeled by a graph whose nodes are proteins and edges indicate known physical interactions between proteins.

Massoud Seifi · Jean-Loup Guillaume
Pierre and Marie Curie University, Laboratory of Computer Sciences, Paris VI (LIP6), 4, place Jussieu 75005 Paris, France, e-mail: firstname.lastname@lip6.fr

Ivan Junier · Jean-Baptiste Rouquier · Svilen Iskrov
Centre for Genomic Regulation (CRG), Barcelona, Spain
Institute of Complex Systems, Paris le-de-France, 57/59, rue Lhomond 75005 Paris, France, e-mail: firstname.lastname@iscpif.fr

An important feature of such networks is that they are generally composed of highly interconnected sub-networks called communities [6]. Communities can be considered as groups of nodes which share common properties and/or play similar roles within the graph. The automatic detection of such communities has attracted much attention in recent years and many community detection algorithms have been proposed, see [11] for a survey. Most of these algorithms are based on the maximization of a quality function known as *modularity* [14], which measures the internal density of communities. Modularity maximization is an NP-complete problem [3], and most algorithms use heuristics. For several reasons related to the modularity, as well as the non-determinism of the algorithms or randomness in initial configuration, such algorithms may produce different partitions of similar quality and there is no reason to prefer one above another. Besides, such algorithms may find communities with a high modularity in networks which have no community structure, e.g. random networks [8]. This is related to the instability of algorithms as shown in [1]: small perturbations of the input graph can greatly influence the output.

Here, we assume that, if several community detection algorithms, or multiple executions of a non-deterministic algorithm agree on certain sets of nodes, then these sets of nodes are certainly more significant. On this basis, we study the tendency of pairs of nodes to belong to the same community during multiple executions of a non-deterministic community detection algorithm. Experimental results on both artificial and real networks show the performance of this concept and we show in particular that it allows to distinguish random from non-random networks.

We provide a general description of algorithms used for detecting consensus communities in Section 2. We then present our previous contributions in Section 3. Finally, we describe the experimental results on artificial and real networks in Section 4 and on random networks in Section 5 before concluding in Section 6.

2 Algorithms for the identification of consensus communities

Two main methods have been used to combine different partitions into a set of consensus communities. One is based on network perturbations. The other one takes advantage of changing the initial configuration of the algorithms.

Network perturbations: Since most community detection algorithms are deterministic, small perturbations can be made on the network to obtain different results. Then, communities are found in each modified network and compared to the partition of the original network to obtain consensus communities. Several methods of network perturbations are proposed in the literature. For example in [9] the method involves removing a fraction of links

and putting them back between randomly selected pairs of vertices. Another technique consists in adding noise to the weight of links, i.e. slightly change them in order to influence the algorithm. For example, in [17] it is proposed to change the weight of links using a Poisson distribution whose parameter is the average weight of links in the original graph. In [5], the noise added to the weight of a link between nodes i and j, initially equal to w_{ij}, is given by a distribution between $-\sigma w_{ij}$ and σw_{ij}, where σ is a constant parameter. A weakness of this method is that it needs an additional parameter σ, whose value is in principle arbitrary. In addition, these studies consider only pairs of adjacent nodes. We will see later that we may identify nodes with a strong tendency to be in the same community even if there is no direct link between them. Also, in these studies, the comparison was made with the partition of the original network, whose significance is not obvious.

Changing the initial configuration of an algorithm: Most algorithms start with an initial partition which is modified many times until a high quality partition is obtained. In general, the algorithms are very sensitive to the initial partition and modifying it may lead to different outcomes. This method is used in [16], to identify overlapping communities by identifying stable and unstable nodes. In [10] this method is used in order to detect communities in multi-scale networks.

There are also similar methods in ensemble clustering like [19] but here we study networks, i.e. structured data, not an unordered set of vectors. In this article, we use the second approach by randomizing the order in which nodes are considered. In addition, we consider all pairs of nodes and not only connected pairs of nodes.

3 Community cores

Given a graph $G = (V, E)$ with $n = |V|$ vertices, we apply $\mathcal{N}$ times a non-deterministic community detection algorithm $\mathcal{A}$ to G. In the following we use the non-deterministic algorithm known as Louvain method [2]. At the end of an execution, each pair of nodes $(i, j) \subseteq V \times V$ can be classified either in the same community or in different communities. We keep track of this in a matrix of size $n \times n$, which we denote by $P_{ij}^{\mathcal{N}} = [p_{ij}]_{n\times n}^{\mathcal{N}}$, where p_{ij} represent the fraction of the $\mathcal{N}$ executions in which i and j were classified in the same community. Note that $p_{ij} = p_{ji}$, and we set $p_{ii} = 0$. From $P_{ij}^{\mathcal{N}}$, we create a complete weighted graph $G' = (V, E', W)$, where the weight of the link (i, j) is p_{ij}. Finally, given a threshold $\alpha \in [0, 1]$, we remove all links having $p_{ij} < \alpha$ from G' to obtain the *thresholded virtual graph*, ${G''}_{\alpha}$. The connected components in ${G''}_{\alpha}$ obtained with a given α are called α-cores, which are non-overlapping sets of nodes.. A pseudo-code version of this algorithm is given in Algorithm 1. We now analyze the impact of the parameters on the results.

Algorithm 1 Core detection

1: Input: a graph $G = (V, E)$, a threshold α, a number of executions $\mathcal{N}$, a non-deterministic community detection algorithm $\mathcal{A}$
2: Apply $\mathcal{N}$ times the algorithm $\mathcal{A}$ to G
3: Create a matrix $P_{ij}^{\mathcal{N}} = [p_{ij}]_{n\times n}^{\mathcal{N}}$ where p_{ij} is the proportion of times that i and j belonged to the same community
4: Create a complete weighted graph $G' = (V, E', W)$ with $|V|$ nodes and p_{ij} as weights
5: Remove all edges ij where $p_{ij} < \alpha$ from G' to obtain ${G''}_{\alpha}$
6: The connected components of ${G''}_{\alpha}$ are α-*cores*

Number of executions: We can estimate the variation of p_{ij} after each execution of algorithm $\mathcal{A}$ by calculating the Euclidean distance between p_{ij} values as a function of the number of executions $\mathcal{N}$. As shown in [16], the variation of p_{ij} converges when the number of executions $\mathcal{N}$ increases. It is therefore possible to terminate the iteration when the variation of p_{ij} is small enough. We derive no theoretical bound on the minimum number of executions to ensure good statistical significance on the estimators p_{ij}. However, we observe that even with an order of magnitude larger of the number of executions, the results do not change much.

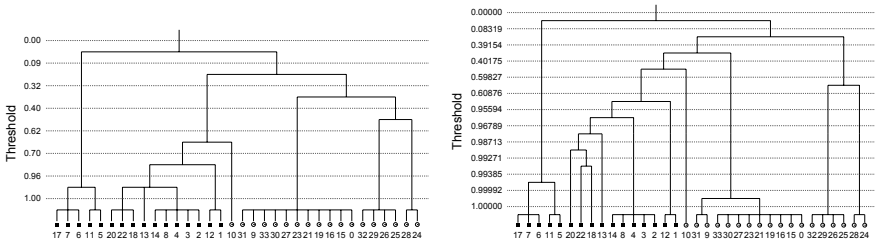
Threshold: The threshold α has a strong influence on the results of the algorithm. The proposed algorithm does not aim at finding the largest sets of nodes that are *all* connected to each other with a $p_{ij} \geq \alpha$. There are two reasons for this: (i) the calculation of cores in this case would consists in finding the largest cliques in G', which is an NP-complete problem and (ii) cores could then overlap, which is not allowed in our case. More precisely, given a threshold α, a core may contain pairs of nodes connected with a probability smaller than α.

3.1 Hierarchical structure of cores

The parameter α has a strong influence on the size of the cores, it furthermore allows to obtain a hierarchical structure of cores. Indeed α_1-cores are included in α_2-cores if $\alpha_1 > \alpha_2$, i.e. α_1-cores are sub-cores of α_2-cores. Let us discuss this on an example.

The Algorithm 1 is applied to the famous friendship network of Zachary's karate club [21]. Figure 1 shows the dendrograms of this network for $\mathcal{N} = 10^2$ and $\mathcal{N} = 10^5$, while Figure 3 shows the cores identified by our algorithm. We can see that the division found by the algorithm with $\mathcal{N} = 10^2$ and $\alpha = 0.32$ corresponds almost perfectly to the Ground Truth and only node 10 is misplaced. Note that the number and size of cores is greatly influenced by the choice of α.

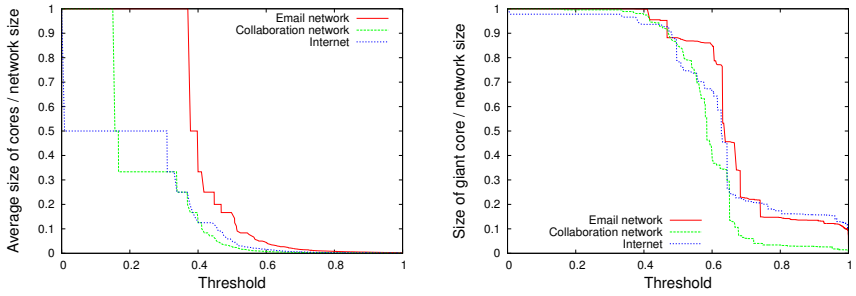
We also applied our algorithm to graphs of different sizes from different domains, including a collaboration network [13], an email network [7] and



(a) $\mathcal{N} = 10^2$ (b) $\mathcal{N} = 10^5$

Fig. 1 Hierarchical structure of cores of Zachary's network for $\mathcal{N} = 10^2$ and $\mathcal{N} = 10^5$

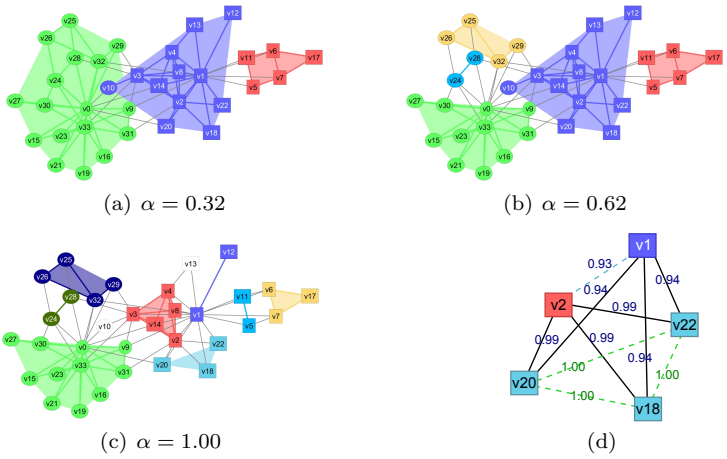
a snapshot of the Internet (created by M. Newman, unpublished). As Figure 2(a) shows, with a threshold close to zero we obtain very large cores (even larger than the communities) and a strict threshold e.g. $\alpha = 1$ will lead to tiny cores, most of which consisting in only one single node (called trivial cores). We also observe in Figure 2(b) that with an $\alpha < 0.5$, we have a giant core containing the majority of nodes. When the threshold increases, the cores will split quickly into small cores. But in the Internet or email network we still have a giant core containing 10% of the nodes even with an α equal to 1. Note that community partitions also contain a giant community.



(a) Average size of cores vs threshold α (b) Size of giant core vs threshold α

Fig. 2 Impact of threshold to the size of the cores

It must be noted that, as explained above, the nodes inside a core are not necessarily connected in the original network. For example, in Figure 3(c), a core containing the nodes 18, 20 and 22 is identified with a threshold $\alpha = 1$, however, there is no direct link between these three nodes in the original graph. As Figure 3(d) illustrates, these nodes were always together either in the community $c_x = \{1, 18, 20, 22, \ldots\}$ or in the community $c_y = \{2, 18, 20, 22, \ldots\}$. This property is interesting and shows that we can identify groups of nodes with a strong tendency to be together even if they have no direct link.

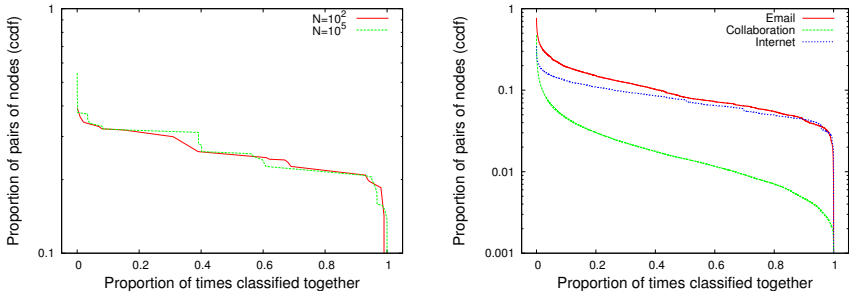


(a) $\alpha = 0.32$ (b) $\alpha = 0.62$

(c) $\alpha = 1.00$ (d)

Fig. 3 (a), (b) and (c) Cores for Zachary's network using three different thresholds. The shape of the nodes (circle/square) is the manual classification made by Zachary. (d) A subgraph of the virtual graph of Zachary's network.

We studied the distribution of p_{ij} in the matrices. The Figure 4(a) shows the p_{ij} distributions of the Zachary's network for $\mathcal{N} = 10^2$ and $\mathcal{N} = 10^5$. As we can see, most pairs are nearly always grouped or separated, but there are some pairs of nodes in the middle which are sometimes together and sometimes separated. The nodes constituting those pairs are less stable. We observe on Figure 4(b) that even on large graphs the majority of pairs are never classified together, and that a significant number of pairs of nodes are always in the same cluster. A large fraction of these pairs are linked in the original network.



(a) (b)

Fig. 4 p_{ij} distribution for (a) Zachary's network and (b) three real-world networks.

4 Significance of cores

We now apply our method to some artificial and real networks having a known community structure, to evaluate the significance of the identified cores.

A classical approach to evaluate the quality of a cluster partition consists in comparing the similarity of the clusters with known communities (or Ground Truth). Various measures of similarity between two clusterings have been proposed [15], and the most widely used is the mutual information, from information theory. It counts the number of bits shared by two random variables. Despite the popularity of mutual information, there are many ways to normalize it, which lead to different values, without definitive solution. Also, it is shown that the mutual information depends on the size of the partitions [20], therefore we used an adjusted version of this metric, called AMI [20]. We also used the edit distance presented in [1] which gives similar results and in some cases is simpler to interpret (data not shown).

Girvan and Newman artificial network [6]: Each graph is constructed with 128 vertices in 4 groups of 32 vertices each. Vertices of the same group are linked with a probability p_{in}, whereas vertices of different groups are linked with a probability p_{out}. Each subgraph corresponding to a group is therefore an Erdös-Rényi random graph [4] with connection probability p_{in}. The probabilities are chosen so as to obtain an average degree $z = 16$. With $p_{in} > p_{out}$ the intra-cluster edge density exceeds the inter-cluster edge density and the graph has a community structure. Figure 5(a) shows a comparison of the similarity of cores and communities to Ground Truth for $z_{out} = 8$ which is a value of z_{out} for which most community detection algorithms fail to identify communities. As we can see, for some α, cores are more similar to Ground Truth than communities.

American College football: This network is also a popular test network with a known community structure [6]. We compare our results with the known partitioning and we find that our algorithm reliably detects the known structure: cores are more similar to known community structure than communities for a wide range of α (see Figure 5(b)).

Another metric that we have used to evaluate the significance of cores is the p-value. The p-value is the probability of obtaining a test statistic at least as extreme as the observed one, assuming that the null hypothesis is true, i.e. assuming that the observed structure is only due to chance. The p-value varies between 0 and 1. The lower the p-value, the stronger the test rejects the null hypothesis, i.e. confirms the significance of the results.

Proteome network: We used this metric to evaluate the significance of identified cores on the Baker's yeast proteome network [18]: nodes are proteins and there is a link when two proteins have been shown to interact. Proteins can work together to achieve a particular function and we used these functions (for instance metabolism or replication) as Ground Truth: a correlation between the clustering and the functions would validate the clustering. We

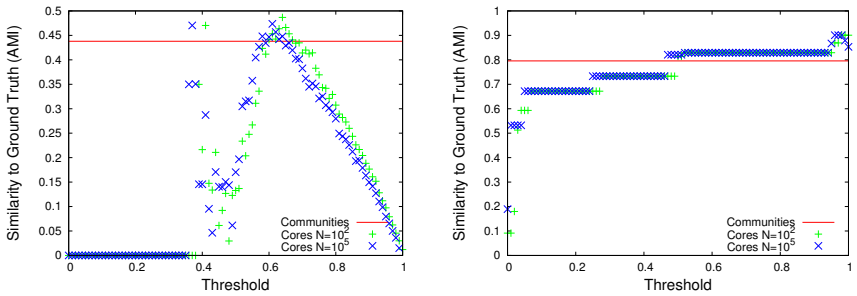


Fig. 5 (a) Girvan-Newman artificial network. (b) American College football.

define the null hypothesis as stating that a core is a random subset of the nodes, of a given size. Thus, for a given function, the number of proteins (or nodes) in the core having this function should follow a hypergeometric law. A small p-value thus denotes the fact that many more proteins than expected have the mentioned function: the nodes have not been chosen at random, but with a bias towards this function.

In Table 1, by comparing the lines having the same label, e.g. "GO:0070478", between the cores and communities, one can see that cores have smaller p-values, except for a few big groups with extremely small p-values, where our method removes some nodes from the group yielding a slightly worse p-value. Also, in cores table, the p-values are smaller when $\alpha < 1$, which means that there is a higher correlation between cores and functions. These findings show that our methodology helps to find relevant sets of cofunctional nodes.

5 Random graphs

We have shown that cores are efficient at finding a Ground Truth on real and artificial networks. In random graphs, the nodes are linked independently to each other so a strong inhomogeneity in the density of links on these graphs is not expected. Therefore random graphs should not have communities. But, as shown in [8], due to fluctuations it is possible to find a partition which has a high modularity for random networks. A good algorithm should indicate both the presence and the absence of community structure. In the following we show that cores cannot be found in random graphs, using two different random graphs model: the classical Erdös-Rényi model [4] and the configuration model [12], which is a construction model that has the degree distribution as an input but is random in all other respects.

First of all, Figure 6(a) shows the impact of the number of execution $\mathcal{N}$ on the distribution of the p_{ij} for a random graph $G(n, M)$. While there are some

(a) Cores

	function	p-value	g_s	g_f	c_s	c_f
$\alpha = 1.00$	GO:0016021	3e-134	5033	927	332	258
	GO:0055085	3e-58	5033	244	332	100
	GO:0005763	1e-45	5033	28	30	21
	GO:0005847	7e-36	5033	15	20	14
	GO:0005789	8e-36	5033	187	332	69
	GO:0016455	7e-35	5033	23	19	15
	GO:0016592	1e-34	5033	24	19	15
	GO:0051123	1e-33	5033	17	15	13
	GO:0032040	1e-33	5033	41	22	17
	GO:0000176	2e-33	5033	13	14	12
$\alpha = 0.99$	GO:0016021	4e-161	5033	927	398	307
	GO:0055085	2e-67	5033	244	398	116
	GO:0005730	1e-51	5033	180	180	65
	GO:0000398	6e-50	5033	58	164	41
	GO:0005763	1e-47	5033	28	32	22
	GO:0005762	2e-41	5033	36	31	21
	GO:0005789	3e-41	5033	187	398	80
	GO:0046540	2e-40	5033	28	164	27
	GO:0016455	3e-37	5033	23	142	23
	GO:0071004	9e-37	5033	29	164	26
$\alpha = 0.98$	GO:0016021	2e-161	5033	927	407	311
	GO:0055085	3e-66	5033	244	407	116
	GO:0005730	5e-51	5033	180	223	70
	GO:0000398	9e-51	5033	58	173	42
	GO:0005763	1e-47	5033	28	32	22
	GO:0005789	1e-42	5033	187	407	82
	GO:0005762	2e-41	5033	36	31	21
	GO:0046540	1e-39	5033	28	173	27
	GO:0071004	4e-36	5033	29	173	26
	GO:0070478	2e-35	5033	17	18	14

(b) Communities

function	p-value	g_s	g_f	c_s	c_f
GO:0016021	8e-170	5033	927	456	338
GO:0055085	9e-075	5033	244	456	129
GO:0005730	2e-050	5033	180	343	82
GO:0005789	1e-044	5033	187	456	88
GO:0000398	6e-044	5033	58	345	46
GO:0005680	3e-031	5033	16	14	12
GO:0046540	3e-031	5033	28	345	27
GO:0008054	3e-030	5033	12	14	11
GO:0031145	2e-029	5033	13	14	11
GO:0007091	2e-029	5033	13	14	11
GO:0030687	5e-029	5033	35	343	29
GO:0071004	6e-028	5033	29	345	26
GO:0045449	6e-028	5033	167	352	59
GO:0016455	1e-027	5033	23	352	23
GO:0006350	1e-026	5033	308	352	79
GO:0005847	2e-026	5033	15	107	15
GO:0016592	3e-026	5033	24	352	23
GO:0006406	5e-026	5033	53	632	40
GO:0006378	8e-026	5033	18	107	16
GO:0004298	1e-025	5033	14	92	14
GO:0000022	3e-025	5033	23	14	11
GO:0005762	7e-025	5033	36	345	27
GO:0005484	1e-024	5033	24	212	20
GO:0005763	1e-024	5033	28	356	24
GO:0000070	2e-023	5033	31	14	11
GO:0005666	7e-023	5033	18	160	16
GO:0006611	1e-022	5033	31	632	28
GO:0032040	1e-022	5033	41	343	27
GO:0005886	5e-021	5033	222	456	68
GO:0005685	1e-020	5033	17	345	17
GO:0070478	1e-020	5033	17	128	14

Table 1 Table of p-values for Baker's yeast proteome network . The parameters to compute the p-value are: g_s: total number of nodes in the network, g_f: number of proteins having this function among all the nodes of the network. c_s: size of the core c_f: number of protein in the present core having this function

high values of p_{ij}, there is a high concentration of p_{ij} at an average value (0.1 with the selected parameters: 1000 nodes, 20000 links). We obtain similar results with a wide range of parameters, see Figure 7. This means that even if we can find partitions with a good modularity, algorithms cannot choose between these partitions.

These results can be explained by the fact that using low values of threshold, our algorithm finds a single core comprising all nodes and since there is nearly no high values in random networks, there is no core with high values of the threshold, while real-world networks have high threshold cores (see Fig-

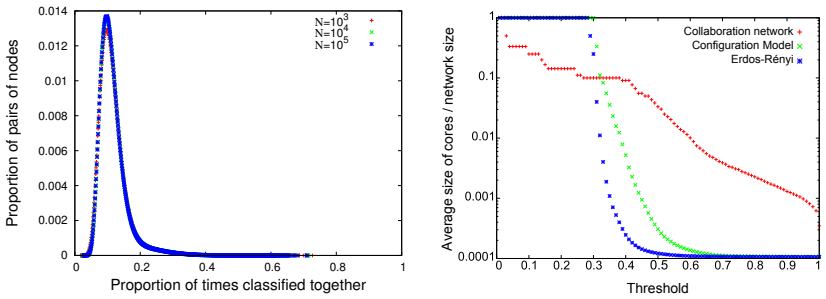


Fig. 6 p_{ij} distribution for different $\mathcal{N}$ (a). Absence of cores in random graphs (b).

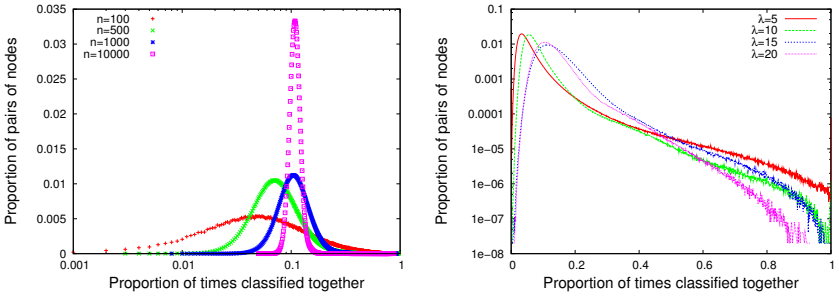


Fig. 7 Distribution of the p_{ij} averaged over 100 realizations, with $\mathcal{N} = 10^3$. Networks with different number of nodes n and an average degree of $\lambda = 20$ (left). Networks with $n = 1000$ nodes and different values of λ (right).

ure 6(b)). Interestingly, in random networks there is a sharp transition (as shown by the cusp at a threshold value around 0.3) between the situation where one single core is present and the intermediate threshold values where several cores are present, which is not present in real-world networks.

To further validate these results, we compared the cores of two real-world networks with random graphs that have the same size (and same degree distribution for the configuration model), see Figure 8. In the case of the Erdös-Rényi model, there is no pair of nodes with $p_{ij} = 0$, which means that all pairs of nodes have been grouped together at least once during 1000 execution of the Louvain algorithm. Conversely, there is nearly no pair of nodes which are always grouped together, but for the leaves (nodes of degree 1) of the network which are always grouped with their only neighbor.

All these results show that random and real-world network behave very differently from a core perspective, while both can exhibit a "classical" community structure, as measured by the modularity. This result gives a strong advantage to cores versus communities.

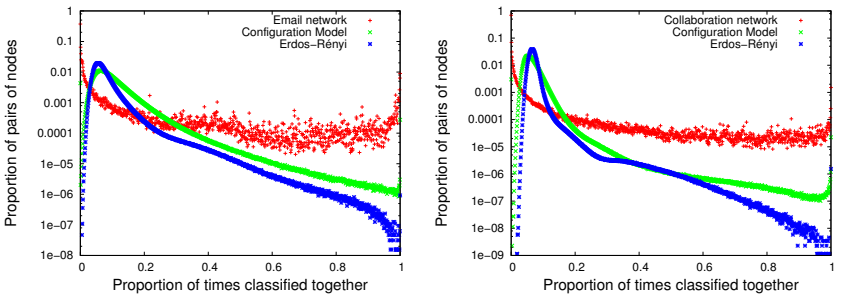


Fig. 8 p_{ij} distribution for two real-world networks together with Erdös-Rényi and configuration model random graphs with the same size.

6 Conclusion

In this paper, we have investigated community structure of complex networks, using community cores which may improve the significance and the stability of groups of nodes detected by current community detection algorithms. We showed that community detection algorithms use heuristics methods which lead to different partitions of similar quality and there is no reason to prefer one above another. Furthermore, community detection algorithms are highly unstable and can find communities in graphs that have none.

If multiple executions of a non-deterministic community detection algorithm agree on certain sets of nodes, then these sets of nodes can be considered as more significant. We showed that cores have a hierarchical structure which can be obtained using different thresholds in our proposed algorithm. We applied our method to both artificial and real networks and showed the performance of our approach when comparing cores to Ground Truth. More particularly, in random networks we find an absence of cores for high enough values of the parameter α. This might provide a robust way to distinguish random networks from real-world networks.

The perspectives of our work are to find a meaningful way to select the threshold, even if the whole hierarchy can be useful as it gives a multi-scale view of the network, and to study the dynamical networks and the evolution of cores in such networks[1].

[1] This work is supported in part by the French National Research Agency contract DynGraph ANR-10-JCJC-0202.

References

1. Aynaud, T., Guillaume, J.: Static community detection algorithms for evolving networks. In: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt), 2010 Proceedings of the 8th International Symposium on, pp. 513–519. IEEE (2010)
2. Blondel, V., Guillaume, J., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment **2008**, P10,008 (2008)
3. Brandes, U., Delling, D., Gaertler, M., Gorke, R., Hoefer, M., Nikoloski, Z., Wagner, D.: On finding graph clusterings with maximum modularity. In: Graph-Theoretic Concepts in Computer Science. Springer (2007)
4. Erdös, P., Rényi, A.: On random graphs, i. Publicationes Mathematicae (Debrecen) **6**, 290–297 (1959)
5. Gfeller, D., Chappelier, J., De Los Rios, P.: Finding instabilities in the community structure of complex networks. Physical Review E **72**(5), 056,135 (2005)
6. Girvan, M., Newman, M.: Community structure in social and biological networks. Proceedings of the National Academy of Sciences **99**(12), 7821 (2002)
7. Guimera, R., Danon, L., Diaz-Guilera, A., Giralt, F., Arenas, A.: Self-similar community structure in a network of human interactions. Physical Review E **68**(6), 065,103 (2003)
8. Guimera, R., Sales-Pardo, M., Amaral, L.: Modularity from fluctuations in random graphs and complex networks. Physical Review E **70**(2), 025,101 (2004)
9. Karrer, B., Levina, E., Newman, M.: Robustness of community structure in networks. Physical Review E **77**(4), 046,119 (2008)
10. Lambiotte, R.: Multi-scale modularity in complex networks. In: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt), 2010 Proceedings of the 8th International Symposium on. IEEE (2010)
11. Lancichinetti, A.: Community detection algorithms: a comparative analysis. Physical Review E **80**(5), 056,117 (2009)
12. Molloy, M., Reed, B.: A critical point for random graphs with a given degree sequence. Random Structures & Algorithms **6**(2-3), 161–180 (1995)
13. Newman, M.: The structure of scientific collaboration networks. Proceedings of the National Academy of Sciences **98**(2), 404 (2001)
14. Newman, M., Girvan, M.: Finding and evaluating community structure in networks. Physical review E **69**(2), 026,113 (2004)
15. Pfitzner, D., Leibbrandt, R., Powers, D.: Characterization and evaluation of similarity measures for pairs of clusterings. Knowledge and Information Systems **19**(3), 361–394 (2009)
16. Qinna, W., Fleury, E.: Detecting overlapping communities in graphs. In: European Conference on Complex Systems (ECCS 2009). Warwick Royaume-Uni (2009). URL http://hal.inria.fr/inria-00398817/en/
17. Rosvall, M., Bergstrom, C.: Mapping change in large networks. PloS one **5**(1), e8694 (2010)
18. Salwinski, L., Miller, C., Smith, A., Pettit, F., Bowie, J., Eisenberg, D.: The database of interacting proteins: 2004 update. Nucleic acids research **32**(suppl 1), D449–D451 (2004)
19. Strehl, A., Ghosh, J.: Cluster ensembles—a knowledge reuse framework for combining multiple partitions. The Journal of Machine Learning Research **3**, 583–617 (2003)
20. Vinh, N., Epps, J., Bailey, J.: Information theoretic measures for clusterings comparison: is a correction for chance necessary? In: Proceedings of the 26th Annual International Conference on Machine Learning, pp. 1073–1080. ACM (2009)
21. Zachary, W.: An information flow model for conflict and fission in small groups. Journal of anthropological research pp. 452–473 (1977)