Title
Effect of size heterogeneity on community identification in complex networks

Author
Danon, L.

Publication Date
2009-02-28

Peer reviewed

Effect of size heterogeneity on community identification in complex networks

Leon Danon,[1] Albert Díaz-Guilera,[1] and Alex Arenas[2]

[1] *Departament de Física Fonamental, Universitat de Barcelona, Marti i Franques 1, 08028 Barcelona, Spain*
[2] *Departament d'Enginyeria Informàtica i Matemàtiques, Campus Sescelades, Universitat Rovira i Virgili, 43007 Tarragona, Spain*

Identifying community structure can be a potent tool in the analysis and understanding of the structure of complex networks. Up to now, methods for evaluating the performance of identification algorithms use ad-hoc networks with communities of equal size. We show that inhomogeneities in community sizes can and do affect the performance of algorithms considerably, and propose an alternative method which takes these factors into account. Furthermore, we propose a simple modification of the algorithm proposed by Newman for community detection (Phys. Rev. E **69** 066133) which treats communities of different sizes on an equal footing, and show that it outperforms the original algorithm while retaining its speed.

PACS numbers: 89.75.Hc, 87.23.Ge

I. INTRODUCTION

Natural and artificial systems often have architectures which are best described as complex networks. The topologies of networks have been extensively studied in various disciplines in recent years, particularly within physics [1–5]. A part of that research has been directed at the study of modules or communities in networks. Communities can be defined as subsets of nodes which are densely connected to each other and loosely connected to the rest of the network. Such structures have been discovered in networks as diverse as banking networks, metabolic networks, the airport network and most notably in social networks [6–10].

Despite efforts spanning several decades in this direction [11, 12], the identification of community structure in networks remains an open problem. The space of possible partitions of even a small network is very large indeed. Several methods have been proposed for finding meaningful partitions in networks of reasonable size. These methods vary considerably from one another, not only in their general approach, but also in sensitivity and computational effort (for recent reviews, see [13, 14] and chapter 7.1 of [5]). In general, those methods which are more accurate tend to be able to explore a larger portion of the partition space, and are therefore computationally expensive (see for example [15, 16]). On the other hand, those methods which explore a smaller region of the partition space tend to be faster, but as a consequence, less accurate [17, 18]. The challenge, therefore, is to find methods which are both fast and accurate, and several attempts have been made [19–21].

In this paper we reevaluate the benchmark most commonly used at present to measure the sensitivity of a particular community identification algorithm [22]. This benchmark, although useful, does not take into account the fact that networks exhibit community structure where the community sizes are highly skewed, despite the fact that several authors have observed that distributions of community sizes seem to follow power laws in many cases [17, 18, 23–26]. In the next section we propose a benchmark for measuring algorithm sensitivity which takes this skew into account. In section III we examine Newman's Fast algorithm (NF) for community detection [17], and see that it is affected by a skew in the community size distribution, showing a tendency to find large communities at the expense of smaller ones. We propose a modification of the algorithm, in which the communities of different sizes are treated equally, and in section IV we show that it outperforms the NF algorithm in sensitivity, with no tradeoff in terms of computational effort.

II. EVALUATING ALGORITHM PERFORMANCE ON AD-HOC NETWORKS

To quantify how good a particular network partition is, the modularity measure Q was introduced in [22], and has been widely used since then. Based on a predefined set of communities i in a network, a community connection matrix e_{ij} is defined, where each member represents the proportion of links from community i to community j. Note that the matrix is normalised, that is, each of the members of the matrix $e_{ij} = \frac{L_{ij}}{L_{total}}$, L_{ij} being the number of links between community i and community j, and L_{total} is the total number of links in the network [22]. The proportion of links belonging to community i is denoted a_i and is simply the sum, $a_i = \sum_i e_{ij}$. The computation of Q is as follows:

$$Q = \sum_i (e_{ii} - a_i^2) \tag{1}$$

The modularity, Q, quantifies the difference between the intra-community links and the expected value for the same communities in a randomised network. Note that the modularity is a relative value, and while it gives an idea of how good a partition of the network is, it cannot tell us whether this partition is the best one possible. It does provide a useful way of comparing the performance of different community identification algorithms applied on one particular network.

The method most commonly used to compare the sensitivity of community identification methods was also proposed in [22], and is independent of the modularity measure. It uses a benchmark test based on networks typically containing 128 nodes grouped into four communities which contain the same number of nodes, 32, and links (on average 16 per node, $k = 16$). Pairs of nodes belonging to the same community are linked with probability p_{in}, whereas pairs belonging to different communities are joined with probability p_{out}. The value of p_{out} controls the average number of links a node has to members of any other community, z_{out}. While p_{out} (and therefore z_{out}) is varied freely, the value of p_{in} is chosen to keep the total average node degree k constant. As z_{out} is increased from zero, the communities become more and more fuzzy and harder to identify. Different community detection algorithms, when applied to these networks may give different results, reflecting their sensitivity. Since the 'real' community structure is well known in this case, it is possible to measure how well the partitions the algorithm finds compare to the original partitions.

Here we use a measure based on information theory for this purpose. The normalised mutual information, $I(A, B)$, explicitly measures the amount of information about partition A that is gained by knowing partition B [27, 28]. In other words, it is the amount of information the algorithm is able to extract from the predefined partition just from the topology. [14]. This independent measure is based on defining a *confusion matrix* $\mathbf{M}$, where rows correspond to "real" communities, and columns correspond to "found" communities. The element of $\mathbf{M}$, M_{ij} is the number of nodes in the real community i that appear in the found community j. A measure of similarity between the partitions, is then:

$$I(A,B) = \frac{-2\sum_{i=1}^{c_A}\sum_{j=1}^{c_B} M_{ij}\log\left(\frac{M_{ij}N}{M_{i.}M_{.j}}\right)}{\sum_{i=1}^{c_A} M_{i.}\log\left(\frac{M_{i.}}{N}\right) + \sum_{j=1}^{c_B} M_{.j}\log\left(\frac{M_{.j}}{N}\right)} \tag{2}$$

where the number of real communities is denoted c_A and the number of found communities is denoted c_B, N is the number of nodes, the sum over row i of matrix M_{ij} is denoted $M_{i.}$ and the sum over column j is denoted $M_{.j}$.

Because of the particular definition of these ad-hoc networks, it is tempting to think that similar networks with four communities sharing the same value of z_{out}/k will have an equivalent community structure, and that a particular method of community identification will perform equally well. This, however, is highly dependent on the number of nodes that the network has, and more importantly the number of nodes in each community. For example a network with 128 nodes with four communities each of size 32 with $k = 16$ and $z_{out} = 6$, say, will have a better defined community structure than a network with the same values of k and z_{out} which is comprised of 512 nodes with four communities each of size 128. This is simply due to the fact that the internal links are spread out over a larger number of nodes, thus making the communities less dense, in terms of proportion of actual links to possible links. In Figure 1, we can see that the same algorithm will perform significantly better on a network with 128 nodes than on one with 512 nodes with the same values of k and z_{out}.

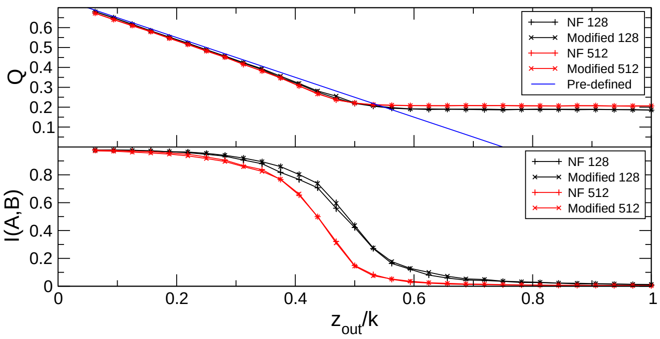

FIG. 1: (Colour online) Sensitivity of the NF algorithm and the modification described in Section III, applied to ad-hoc networks with four equal-sized communities, for two network sizes, 128 nodes and 512 nodes, with average degree $k = 16$. The top figure shows the variation of modularity found by the algorithms with z_{out}/k. For low values of z_{out}/k, the value of Q of the partitions found closely follow the expected modularity. For higher values of z_{out}/k, the partitions found show a better modularity than pre-defined partitions. There is little difference between results for different network sizes. In the bottom figure the comparison between pre-defined and found partitions using the mutual information measure $I(A, B)$ is shown. Both algorithms have similar sensitivity for both network sizes, but the sensitivity is reduced at the same value of z_{out}/k for the larger network, suggesting that communities are more fuzzy the larger they are as discussed in the text.

Furthermore, in real networks the distribution of community sizes is highly skewed, and has been observed to follow power laws in many cases [13, 18, 23–25]. We argue that this difference in sizes is important and affects different identification algorithms in different ways. To be able to evaluate the effect that a spread in community sizes will have on the performance of any algorithm, we first need to be able to create networks with controlled community structure of differing community sizes.

Consider a set of N_c communities where each community contains n_i nodes. Considering pairs of nodes, if both nodes are in the same community a link is placed between them with probability P_{in}, otherwise they are connected with probability P_e. Should P_{in} be constant for all communities, the number of links of community i would scale as the square of its size, n_i^2. To give the same weight to communities of different sizes, we propose that $P_{in} = F/n_i$ where F is a control parameter. In this way we are able to control both internal and external cohesion by varying F and P_e respectively. This method of network creation is equivalent to creating a random Erdös-Renyi network with the probability of linking being equal to P_e and then superposing N_c random networks whose sizes correspond to n_i where the probability of internal linking is F/n_i.

Figure 2(a and b) shows two networks with 5 communities each, containing one community of 64 nodes and 4 communities of 16 nodes each for two different values of P_e and F. Figure 2c shows the value of Q when the network partition corresponds exactly to the prescribed communities as a function of F and P_e. While these community sizes are chosen to be illustrative, this method of network creation is completely general and community sizes can be drawn from any given distribution.

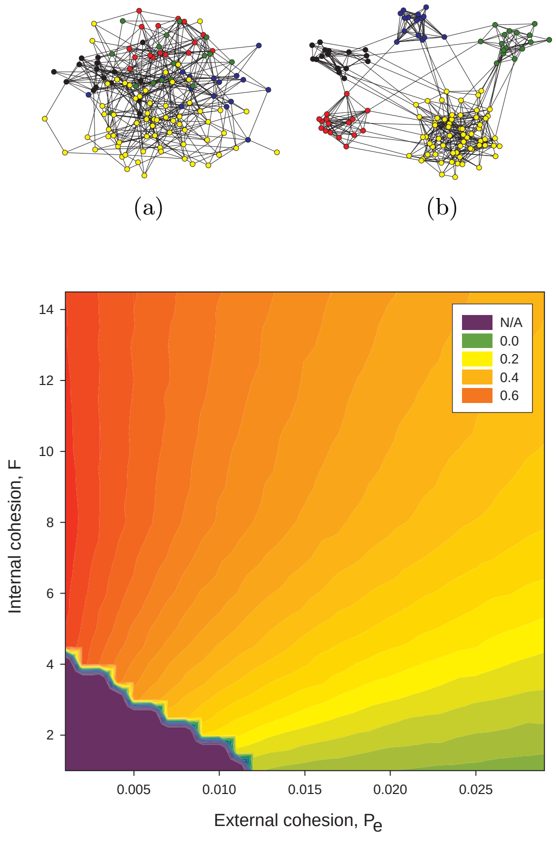

FIG. 2: (Colour online) Two examples networks created as described in the main text with 5 communities four of which have 16 nodes and one has 64, (a) has $P_e = 0.007$ and $F = 8$ and in (b) $P_e = 0.03$ and $F = 3$. (c) The modularity Q of networks as generated in the main text for values of P_e between 0.001 and 0.03, and values of F between 1 and 14. The dark zones represent parts of the parameter space where the networks constructed were disconnected for more than 1 in 100 realisations.

III. DYNAMICS OF THE FAST ALGORITHM AND ITS MODIFICATION

The performance of various community identification algorithms has recently been studied both in terms of speed and in terms of accuracy. Having a method of generation of networks with communities of differing sizes puts us in a position to test the way these sizes can affect the performances of identification algorithms. In particular we concentrate on Newman's Fast algorithm as proposed in [17]. It is dubbed fast since it runs in almost linear time for sparse networks, $O(n \log^2 n)$ [18], and while it is not the most accurate method, it remains the only algorithm able to extract community structure information from very large networks [14].

Let us consider a network that has been partitioned in some arbitrary way. Joining two neighbouring partitions i and j, would produce a change in modularity:

$$\Delta Q_{ij} = 2(e_{ij} - a_i a_j) \tag{3}$$

This can be interpreted as a measure of affinity of communities i and j, and can subsequently be used to find the two communities which are most alike (highest ΔQ). Starting with each node in the network in its own community, one can join pairs of communities with the highest ΔQ. This process can then be performed and repeated until the whole network is contained in one community. As the author states in [13], this is very similar to agglomerative hierarchical clustering methods [29, 30]. Here, "distance" measures such as single linkage or complete linkage are replaced by ΔQ. It also differs from hierarchical clustering in that not all pairs of clusters are compared, only those connected by real links in the network.

Let us analyse carefully how the algorithm proceeds when applied on the well studied karate club friendship network of Zachary [31]. Data on the network was collected over a two year period before the club split due to an internal dispute during which some of the members started their own club. The fissure is apparent in the topology of the network before the split (see Figure 3a), and this data set has become somewhat of a standard case study for community detection algorithms in the literature [10, 13, 19–21, 32–35].

Figure 3c shows the dendrogram as generated by the fast algorithm, with the different colours depicting the partition at the highest value of $Q = 0.3807$. In the first step of the algorithm, a_i is simply the degree of node i and e_{ij} is 1 for any neighbour pair. Hence, the pair of nodes that will be joined first is the neighbour pair that has the smallest product of degrees. In the case of the karate club network, these are nodes 6 and 17 with degrees 3 and 2 respectively. Note that once a community has joined with another, the resulting community tends to join again, since the first term of 4, e_{ij}, tends to be increased by the joining of neighbouring communities, especially in networks with high clustering. So, the cluster of nodes 6 and 17 absorb their common neighbour, node 7. This larger cluster now has an even larger e_{ij} to common neighbours and in the following steps absorbs nodes 1, 5 and 11, until no common neighbours exist. This process occurs in a similar fashion for nodes 24, 27, 28, 30 and 34. We observe that when choosing the pair of communities to be joined, large communities are favoured at the expense of smaller ones. In turn, this leads to the formation of a few large clusters in networks where a larger number of smaller clusters may represent the real community structure better.

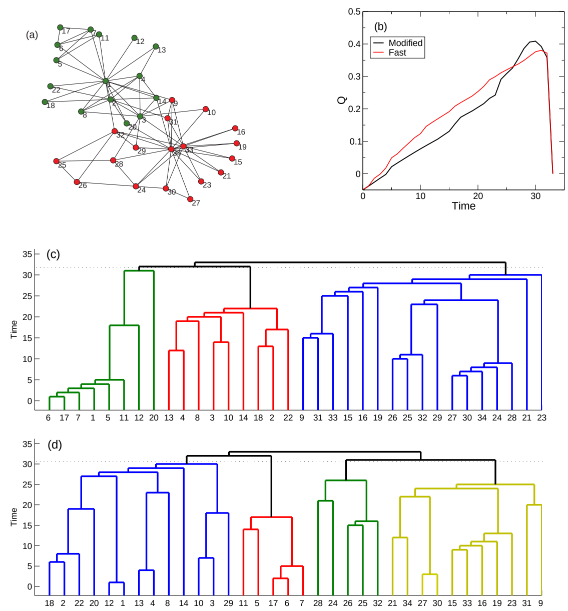

FIG. 3: (Colour online) (a) Zachary's karate club network (b) Modularity as algorithms progress (c) Dendrogram representing the progress of fast algorithm, where formation of large clusters is favoured early (d) Dendrogram representing the progress of our modification, all clusters are treated on an equal footing and individual nodes are absorbed into clusters early.

To avoid this and to treat each community as equal, we "normalise" ΔQ by the number of links:

$$\Delta Q'_{ij} = \frac{\Delta Q_{ij}}{a_i} = \frac{2}{a_i}(e_{ij} - a_i a_j) \qquad (4)$$

It is important to note that while the pair of nodes with the largest value of $\Delta Q'$ is chosen, the real value of Q must be calculated at each step using the original ΔQ, or measuring the value of Q explicitly. Note that as opposed to the original formulation, this measure is asymmetric, that is $\Delta Q'_{ij} \neq \Delta Q'_{ji}$. But, the implementation of the algorithm ensures that both $\Delta Q'_{ij}$ and $\Delta Q'_{ji}$ are considered when choosing the pair of communities to join, and, since we are interested in only the largest value of $\Delta Q'$ at each step, this poses no problem. In essence, the modified algorithm is able to take a different path in the partition space from the original, in part due to this asymmetry. For each possible merging of neighbouring communities, there exists only one value of ΔQ, whereas $\Delta Q'$ takes two different values, if the two communities have a different number of links $a_i \neq a_j$.

This normalisation insures that clusters with fewer links have the largest values of $\Delta Q'$, and therefore are joined earlier. Taking the karate club network as an example again, we see that neighbouring nodes where one neighbour has the smallest degree are joined first. This ensures that nodes with only one link are joined at the beginning of the process, such as node 12 (see Fig. 3d.). Curiously, using another method based on synchronisation recently proposed by two of us produces a very similar dendrogram [36]. We argue that this is a better way to proceed. A partition containing a single node will always contribute negatively to the value of Q, even if the degree of that node is 1. For example in [37] the authors find a partition with $Q = 0.412$ which has node 12 as a separate community, using an entirely different method for exploring the partition space. But, $Q_{i=12} = -1/78$ and the same partition, only with node 12 contained within it's neighbour community, has $Q = 0.418$ [**?**].

While the NF algorithm also ensures that single node partitions are not found in the optimal state, our modification performs this absorption much earlier. This means that in the first few steps of our algorithm will inevitably appear to performing worse than the NF algorithm. As it progresses, however, it overtakes the NF algorithm in terms of Q, as we can see in Figure 3b. Indeed, we find that when our modification does not match the performance of the NF algorithm in terms of Q, it improves it.

IV. TESTING THE MODIFICATION

To test the performance of the modification proposed, we have applied the algorithm on several networks, both ad-hoc and real. To begin with we look at networks with four equal sized communities, as described in [22].

As z_{out}/k increases, the modularity of the pre-defined partition decreases as $Q = 3/4 - z_{out}/k$ irrespective of network or community size. Figure 1a shows the expected modularity value compared with those found by the NF algorithm and our modification. For low values of z_{out}/k both algorithms find communities with the value of Q following the expected value closely. For higher z_{out}/k these values deviate from the expected value as the communities found by the algorithm do not correspond exactly to pre-defined communities. In fact, as z_{out}/k increases above 0.5 the pre-defined partitions give a lower value of Q than those found by the algorithm, which tend towards the value that random networks exhibit due to fluctuations [16]. The values of Q found by our modification is very similar to those found by the NF algorithm.

The deviation between pre-defined and found partitions is seen more clearly by looking at the mutual information measure $I(A, B)$ in the lower part of Figure 1. As z_{out}/k increases beyond the point where communities are well defined, the amount of information about community structure the algorithms are able to extract decreases. When the communities found have hardly any relation to pre-defined ones, as is the case for high z_{out}/k, $I(A, B)$ tends to zero. As network size increases however, the algorithms are able to extract less information from the network structure. This supports the suggestion that communities in these networks become more fuzzy as their size increases. Once again, our modification performs very similarly to the NF algorithm.

It seems logical that both algorithms perform with similar accuracy for these networks. As we have seen in III the NF algorithm seems to favour the formation of larger communities. However when the communities to be found are all of the same size, one would expect it to perform quite well. Our modification has little effect in this case.

The difference between the algorithms appears when communities of different sizes are present within the network. Using the network construction method proposed in Section II, we study the performance of the algorithm on networks with 21 communities. The communities are chosen by hand, with one community of 128 nodes, four communities with 32 nodes each and 16 communities containing 8 nodes each. This corresponds to a size distribution which follows a power law (with only three points), where the exponent is -1. In Figure 4 we show the difference in performance between the NF algorithm and our modification. They are compared both in terms of modularity and mutual information. Our modification performs better in all parts of the parameter space, with some regions showing up to 25% improvement over the original algorithm. The regions where the improvement is largest are those where the communities are fuzzy, that is, for high values of external cohesion P_s and low values of internal cohesion F.

This suggests that our modified algorithm will perform better in real networks, where the size of communities is highly heterogeneous and the community structure is fuzzy. To check this, we also performed tests on some real networks. Table I shows the comparison of our modified algorithms with Newman's original formulation and, where possible with the extremal optimisation algorithm. We looked at the network of Jazz bands with nodes representing the bands, and links between bands representing at least one musician that played in both [26]; the e-mail network of University Rovira i Virgili [23] where e-mail addresses are connected by exchanging messages; and the network of users of the pretty good privacy (PGP) algorithm for secure information transactions [38]. These are medium sized networks and are still tractable with the Extremal Optimisation (EO) algorithm [20], which has a larger running time scaling as $O(n^2 \log n)$. In these networks, the EO algorithm clearly performs best out of the three, which is no surprise since it explores much

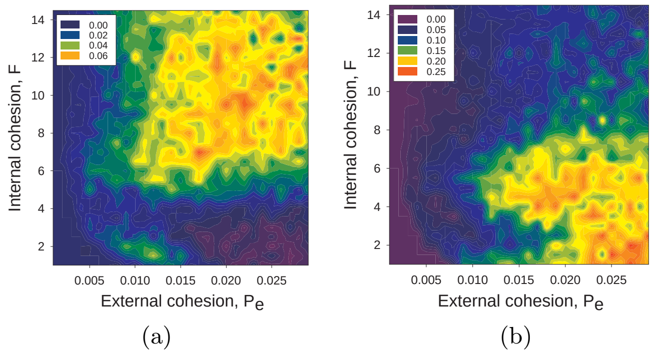

FIG. 4: (Colour online) Difference in performance between the NF algorithm and our modification (a) proportion of improvement in Q (b) proportion of improvement in I(A,B). Our modified algorithm outperforms the NF algorithm in all parts of parameter space, but the difference is most pronounced for low values of P_e and high values F, i.e. where the communities are fuzzy. Each point is an average over 100 realisations of the network.

TABLE I: Table of optimal modularity values obtained by the Extremal Optimisation algorithm, Q_{EO} [20], the NF algorithm, Q_{NF} [17], and the modification presented here, Q_{QM}.

Network	Size	Q_{EO}	Q_{NF}	Q_M
Zachary	34	0.4188	0.381	0.4087
Jazz bands	198	0.4452	0.4389	0.4409
E-mail	1144	0.5738	0.4796	0.5569
PGP	10680	0.8459	0.7329	0.7462
arXiv	44337	N/A	0.7165	0.7606
WWW	325729	N/A	0.9269	0.9403
Actor	374511	N/A	0.6829	0.7194

more of the partition space than either of the others. It is, however, impractical to use in very large networks due to running time. In large networks such as the coauthorship network of the arXiv preprint database [39], the network of web pages within the nd.edu domain [40], or the actor network [41], our algorithm is still able to run in a reasonable time. It improves on the results of the NF algorithm, finding partitions up to 16% better in terms of Q, with no tradeoff in speed.

V. CONCLUSION

To conclude, in this paper we have proposed a more realistic benchmark test for community detection algorithms in complex networks which takes into account the heterogeneity of community size observed in real networks. We have also shown that Newman's fast community detection algorithm tends to favour the creation of large communities at the expense of smaller ones. We propose a simple modification of the fast algorithm which can ensure that communities of differing sizes are treated on an equal footing, thus side-stepping this potential problem. Upon comparing the sensitivity of our modification to that of the original algorithm, we saw that they perform almost identically in ad-hoc networks with communities of equal size. However, when compared using the proposed benchmark test, the improvement in sensitivity increases. Therefore, we claim that the heterogeneity in community size should be considered when evaluating community detection algorithms.

Furthermore, we have seen that our modified algorithm gives improved results for all real networks studied. This improvement is up to 16% in some studied networks. The improvement in results comes at no extra computational cost, and a reasonable implementation of the algorithm will run in $O(n \log^2 n)$ time. We recommend the use of this simple modification for the study of community structure in very large complex networks.

VI. ACKNOWLEDGEMENTS

We thank J. Duch and M. Bogunya for useful discussions and Mark Newman for providing the arXiv network data. This work has been supported by DGES of the Spanish Government Grant No. BFM-2003-08258 and EC-FET Open Project No. IST-2001-33555. L.D. gratefully acknowledges funding from Generalitat de Catalunya. This work was supported in part by the Director, Office of Science, Computational and Technology Research, U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

[1] A. L. Barabasi and R. Albert, Review of Modern Physics **74**, 47 (2002).
[2] S. N. Dorogovtsev and J. F. F. Mendes, *Evolution of Networks: From biological nets to the internet and WWW* (Oxford University Press, Oxford, 2003).
[3] M. E. J. Newman, SIAM Review **45**, 167 (2003).
[4] S. H. Strogatz, Nature **410**, 268 (2001).
[5] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, Physics Reports p. in press (2006).
[6] M. Boss, H. Elsinger, M. Summer, and S. Thurner, cond-mat/0309582 (2003).
[7] E. Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai, and A.-L. Barabsi, Science **297**, 1551 (2002).
[8] R. Guimerà and L. N. A. Amaral, Nature **433**, 895 (2005).
[9] R. Guimerà and L. N. A. Amaral, JSTAT p. P02001 (2005).
[10] M. Girvan and M. E. J. Newman, Publications of the National Academy of Sciences USA **99**, 7821 (2002).
[11] B. W. Kernighan and S. Lin, The Bell System Tech J **49**, 291 (1970).
[12] A. Pothen, *Parallel Numerical Algorithms* (Kluwer Academic Press, 1996), chap. Graph partitioning algorithms with applications to scientific computing.
[13] M. E. J. Newman, European Physics Journal B **38**, 321 (2004).
[14] L. Danon, A. Díaz-Guilera, J. Duch, and A. Arenas, J. Stat. Mech p. P09008 (2005).
[15] J. Reichardt and S. Bornholdt, Phys. Rev. Lett. **93**, 218701 (2004).
[16] R. Guimerà, M. Sales, and L. N. A. Amaral, Physical Review E **70**, 025101 (2004).
[17] M. E. J. Newman, Physical Review E **69**, 066133 (2004).
[18] A. Clauset, M. E. J. Newman, and C. Moore, Physical Review E **70**, 066111 (2004).
[19] J. P. Bagrow and E. M. Bollt, Physical Review E **72**, 046108 (2005).
[20] J. Duch and A. Arenas, Physical Review E **72**, 027104 (2005).
[21] F. Wu and B. Huberman, Eurpean Physics Journal B **38**, 331 (2004).
[22] M. E. J. Newman and M. Girvan, Physical Review E **69**, 026113 (2004).
[23] R. Guimerá, L. Danon, A. Diaz-Guilera, F. Giralt, and A. Arenas, Physical Review E **68** (2003).
[24] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek, Nature **435**, 814 (2005).
[25] A. Arenas, L. Danon, A. Diaz-Guilera, P. M. Gleiser, and R. Guimerà, European Physical Journal B **38**, 373 (2004).
[26] P. Gleiser and L. Danon, Advances in Complex Systems **6**, 565 (2003).
[27] A. L. N. Fred and A. K. Jain, Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR, USA pp. II–128–133 (2003).
[28] L. I. Kuncheva and S. T. Hadjitodorov, Systems, Man and Cybernetics, 2004 IEEE International Conference **2**, 1214 (2004).
[29] B. Everitt, *Cluster Analysis, 3rd edition* (Edward Arnold, London, 1993).
[30] J. Scott, *Social Network Analysis, a handboook* (SAGE publications, London, 2nd edition, 2000).
[31] W. W. Zachary, Journal of Anthropological Research **33**, 452 (1977).
[32] S. Fortunato, V. Latora, and M. Marchiori, Physical Review E **70** (2004).
[33] H. Zhou, Physical Review E **67**, 041908 (2003).
[34] H. Zhou and R. Lipowsky, Lecture Notes in Computer Sciences **3038**, 1062 (2004).
[35] D. Gfeller, J.-C. Chappelier, and P. De Los Rios, Physical Review E **72**, 056135 (2005).
[36] A. Arenas, A. Diaz-Guilera, and C. J. Perez-Vicente, pp. cond–mat/0511730 (2005).
[37] L. Donetti and M. A. Muñoz, Journal of Statistical Mechanics: Theory and Experiment (2004).
[38] X. Guardiola, R. Guimerà, A. Arenas, A. Diaz-Guilera, D. Streib, and L. Amaral, cond-mat/0206240 (2002).
[39] M. E. J. Newman, Physical Review E **64**, 016132 (2001).
[40] R. Albert, H. Jeong, and A.-L. Barabási, Nature **401**, 130 (1999).
[41] A.-L. Barabási and R. Albert, Science **286**, 509 (1999).
[] Such a partition is found by a more exhaustive search of the partition space, by using for example the EO algorithm by Duch and Arenas [20].