Finding Modules in Networks with Non-modular Regions

Sharon Bruckner[⋆], Bastian Kayser, and Tim O.F. Conrad

Institut für Mathematik, Freie Universität Berlin, Germany
sharonb@math.fu-berlin.de, bastiankayser84@googlemail.com, conrad@math.fu-berlin.de

Abstract. Most network clustering methods share the assumption that the network can be completely decomposed into modules, that is, every node belongs to (usually exactly one) module. Forcing this constraint can lead to misidentification of modules where none exist, while the true modules are drowned out in the noise, as has been observed e. g. for protein interaction networks. We thus propose a clustering model where networks contain both a *modular region* consisting of nodes that can be partitioned into modules, and a *transition region* containing nodes that lie between or outside modules. We propose two scores based on spectral properties to determine how well a network fits this model. We then evaluate three (partially adapted) clustering algorithms from the literature on random networks that fit our model, based on the scores and comparison to the ground truth. This allows to pinpoint the types of networks for which the different algorithms perform well.

1 Introduction

A common way of analyzing networks is to partition them into *clusters* (or *modules, communities*) where similar or interacting nodes are grouped together. This is known as *Graph Clustering*. Such a grouping can help identifying the underlying structure of the network and extract insights from it. For example, modules in a protein–protein interaction (PPI) network can correspond to protein complexes (see, e.g. [9, 27]). Accordingly, many clustering methods have been developed, varying in their definition of the optimal clustering and in the approach taken to compute it [17]. However, in most of these methods, the partition must be a *full partition*, meaning every node must belong to exactly one module. This constraint both limits the classes of networks that can be clustered, and the insights that can be gained from them.

In this work, we propose a more flexible model, where networks have two parts: a *modular region*, which can be fully partitioned into individual modules, and a *transition region*, containing nodes that cannot be assigned to any module. We call these networks "not completely clusterable" (NCC networks). Consider for example the PPI network mentioned above. It is well-known that there are

[⋆] Supported by project NANOPOLY (PITN-GA-2009–238700).

proteins that are involved in more than one protein complex [10]. Additionally, not every protein takes part in a complex, meaning that many proteins should not be assigned to *any* cluster [24]. By forcing each node into a cluster, we could fail to single out these proteins and regions and could introduce errors and meaningless clusters.

Previous work. In recent years, there has been interest in methods that challenge one part of the "full partition" assumption: Methods to find overlapping clusters, or perform *fuzzy clustering* on the network, allow each node in the network to belong to more than one module. Overlapping Clusters approaches, such as those based on clique percolation [14], variants of modularity [13, 28] or other concepts [22], identify modules that can share nodes. In fuzzy graph clustering, each node receives a probability of being assigned to each module and the modules can then be determined by thresholding the probabilities [19, 25]. Fewer methods exist, to our knowledge, that further weaken the assumptions above: methods like SCAN [26] and the one described by Feng et al. [6] define three types of nodes: nodes that belong to a single module, nodes that can belong to more than one module (*hubs*), and nodes that belong to no module at all (*outliers*). This is close to the framework that we present in this paper; however, we assume here that the nodes that do not belong in modules (either hubs or outliers, as termed by the other methods) need not necessarily comprise a small and negligible part of the network.

The MSM (Markov State Model) algorithm was proposed by some of us [20] to directly address the problem of identifying modules in NCC networks. It uses the concept of *metastability* and tries to identify metastable sets, which are then equated with modules.

Our Contributions. We discuss NCC networks exhibiting the properties defined above: (1) presence of a transition region, (2) presence of modules. In Section 2, we propose a simple characterization of such networks, *partial modularity.* Intuitively, the structures of the modular region of our networks are dense, while the transition region is relatively sparse. We propose and discuss two scores to formalize these notions, and demonstrate their behavior on simple networks.

The next natural question is, given that a network has a high partial modularity, how can we identify its modules? In Section 3, we first present a score for the quality of a clustering resulting from an algorithm by comparing it with the clustering dictated by some ground truth. We use this scoring function to compare the performance of state-of-the-art module finding methods on benchmark networks. Two of these algorithms are adaptations of existing popular algorithms, and the third, MSM, is designed for NCC networks.

2 Scoring Partial Modularity

To formalize the notion of an NCC network, we want a quantitative score for a network that evaluates the degree to which it contains modules. The natural candidate for this is Newman's modularity [12], designed to measure the strength of a full partition of a network into modules: dense connections within modules, sparse connections between the modules. However, this score has a quality that makes it not suitable for NCC networks: Networks that are tree and tree-like have a high modularity [2]. This means that sparse networks without any dense subgraphs have a high modularity, despite having no dense modules.

Since Newman's modularity is not adequate, we search in a different direction. It has been known (see, e.g. [11, 23]) that the number of eigenvalues of the transition matrix of a network that are close to 1 are linked to the number of modules in the network, and that the size of the eigenvalue gap could then indicate the difference between modules and the rest of the network. We take this idea further by looking at the gap of a different transition matrix: that of an embedded Markov chain of the continuous Markov process defined by Sarich et al. [20], where the continuous random walk is generated by a custom generator L such that the process stays for extended periods of time in dense regions of the network.The advantage is a better correspondence between eigenvalues close to 1 and dense modules, and a clearer gap. We therefore consider as a network score the size of the largest gap of this matrix $P = \exp(\alpha L)$, where α is the lag time defined in [20], which acts as a granularity parameter. Let λ_u be the eigenvalue of L that lies above the gap, and λ_l the eigenvalue below the gap. Then we define the gap score as

$$Q_\gamma := \exp(\alpha\lambda_u) - \exp(\alpha\lambda_l). \tag{1}$$

We first note that sparse networks do not have a high gap score. For example, we tested several road networks.[1] These networks have a very low average clustering coefficient (~ 0.01) and density (~ 0.0002). While their Newman modularity [12] is > 0.95, the gap score is < 0.002, considerably lower and better reflecting the absence of modules.

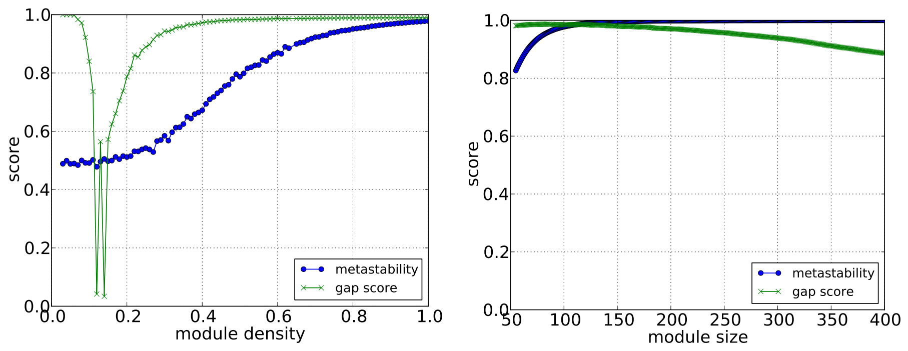
The gap score has several drawbacks. First, there is not one "true" gap in the spectrum, just as there is not one "true" clustering of the network. Different gaps induce different network partitions, and the choice of largest gap can be arbitrary. Second, there can be networks that contain modules, but do not have a clear gap. This can occur, e.g. when the density of the modules is close to that of the transition region. We will demonstrate this in Section 3.

To try and overcome these problems, we use the concept of *metastability* to define "good" NCC networks. In [20] we introduced the following definition of a metastable partition (see also [3]):

$$R := \max_{y \notin \mathcal{M}} \mathbb{E}_y(\tau(\mathcal{M})) \ll \min_{i=1,\dots,m} \mathbb{E}_i(\tau(\mathcal{M}_i)) =: W, \tag{2}$$

Here, $\mathbb{E}_y(\tau(\mathcal{M}))$ is the expected entry time of the process into an arbitrary module, if started in some node $y \in T$ in the transition region $T = V \setminus \mathcal{M}$. Likewise $\mathbb{E}_i(\tau(\mathcal{M}_i))$ denotes the expected entry time into a module M_j with $j \neq i$ if started from M_i. In other words, the *return time* R the random walk

[1] Downloaded from http://www.cs.fsu.edu/~lifeifei/SpatialDataset.htm



(a) varying module density (b) varying module size

Fig. 1. Metastability and gap scores for random networks with transition region 1000 nodes and 2 modules.

needs to enter one of the modules, if in the transition region, is small compared to its typical *waiting time* W between transitions from one module to another. Based on this, we define the metastability score as

$$Q_m := 1 - R/W. \tag{3}$$

This score, unlike the gap score above, explicitly takes the transition region into account, therefore might be better suited for NCC networks. The main drawback is that this is not a global score, but rather a score for single partition. We would have liked to continue, analogously to Newman modularity [12], by then finding the partition that minimizes R/W, and assigning the network its score. Unfortunately, this will not be useful, since every full partition ($T = \emptyset$) will set $R = 0$ and the score to 1. However, as the experiments below show, it is still indicative of the presence of modules and can be used to compare NCC networks.

Experiments. We performed a set of simple experiments to test the gap and metastability scores.

Figure 1 demonstrates the behavior of the metastability score on networks with a transition region of 1000 nodes (random Erdős–Rényi (ER) graph with density 0.05) and two modules (random graphs with given density and size). To create the network, we first generated the transition region and modules separately, and then for each module randomly identified a vertex from the module and a vertex from the transition region.

In Figure 1(a) the modules have size 100 each, and they are random graphs with density 0.03 to 1. The metastability score was computed for the planted ground truth partition. As the density of modules increases, the metastability score increases also, as the denser modules become more metastable. The transition region does not change, therefore R is the same and only W changes. The gap score increases as well with the module density, except in the cases where the density is low (< 0.18): The 3rd eigenvalue, corresponding to the transition region, is farther and farther from the 2nd eigenvalue as the density of the modules grows further from the transition region density. The errors are a result of the gap not being clear enough when the module density is close to that of the transition region, being identified between the 1st and 2nd eigenvalues. The same cannot be said when the density is constant but the module size changes: The set of networks whose scores are displayed in Figure 1(b) have modules of changing sizes, from 55 to 400, that are complete graphs. Again we see that the metastability increases with the module size, since the larger modules are more metastable. All scores are high since even the small modules, being large complete graphs, are already metastable. The gap score shows the opposite trend: The gap between the 2nd and 3rd eigenvalues decreases as the module size increases and the module size becomes closer to the transition region size. This demonstrates that the gap score does not always agree with our intuition of a modular network, and underscores the need for a better modularity score.

3 Algorithms

We select three clustering algorithms from literature, which we partially adapt for our setting. We choose standard parameters for all algorithms.

SCAN. The SCAN algorithm [26] clusters vertices together based on neighborhood similarity and reachability. It can identify vertices as hubs or outliers; we interpret both as the transition region. SCAN requires a user-defined parameter μ that determines the minimum size of a module. This we set to 10, 1% of the network size of most of the NCC networks we use for evaluation.

Markov Clustering. The idea of the MCL (Markov Clustering) algorithm [5] is to simulate random walks on the network and identify modules as regions where the random walker stays for a prolonged time. MCL always returns a full partition. It has been demonstrated (e. g. [21]) that MCL tends to produce imbalanced clusterings, consisting of a few large clusters and many small clusters of size two or three and singletons. Usually viewed as a shortcoming of the algorithm, we now interpret this tendency to our advantage: We introduce a parameter μ similar to SCAN to set a minimum size for a module. All modules with less than μ nodes are assigned to the transition region.

Markov State Model. The MSM (Markov State Model) algorithm [20] first tries to identify the modular region as the region where a random walker spends the most time. The rest is classified as transition region, and the modular region is clustered with a simple heuristic.

Ground-Truth-Based Evaluation. We evaluate the algorithms by comparing their output to the known clustering using the adjusted Rand index [16, 18], which measures how well two partitions match. We propose three versions of the score: ρ_{MT} to measure how well the modular region and transition region are distinguished, ρ_M to measure the quality of the clustering within the modular region, and ρ_c as a combined score.

3.1 Experiments on Random Networks

We now test the algorithms on random NCC networks, constructed as follows: Each node belongs either to exactly one of the modules, or to the transition region. We use a similar random graph model as in Section 2, with each module and the transition region being a random ER graph. In addition, each module and the transition region are connected by adding a random spanning tree. Then, from the transition region and from each module a node is chosen at random, and these nodes are connected by a random spanning tree to ensure that the whole graph is connected. Finally, each possible edge between a vertex from a module and a vertex from the transition region is added with a small probability.

This class has the following parameters: **Network size** (total number of nodes) N, **number of modules** M, **total number of nodes in modules** N_{mtot}, **module density** p_m, **transition region density** p_t and **inter-connection density** p_i . The inter-connection density is an indicator for the number of edges between modules and the transition region. We also define the number of nodes in transition region $N_t := N - N_{mtot}$ and number of nodes per module $N_m := N_{mtot}/M$.

The standard parameter values are as follows: $N = 1000$ nodes, $M = 5$ modules, $p_m = 0.6$ module density, $p_t = 0.01$ transition region density, and $p_i = 0.01$ interconnection density. The minimal module size under these constraints is then 10 nodes.

Since in practice the running time of the algorithms depends on implementation, and in every case the running time was < 1 minute, we focus here on the accuracy of the algorithm as determined by our evaluation measure.

Experiment 1: Varying transition region size. In this experiment, our goal is to evaluate the behavior of the different algorithms on networks where the transition region comprises between 0% and 90% of the network. In the case of 0% transition region, the modular region occupies the entire network, and the problem will again be that of full partitioning. We hypothesize that the algorithms should perform better on networks with a small transition region, as they are closer to the full partition case: SCAN looks for hubs and outliers but those are usually single nodes, not entire regions; MCL was originally designed for full partitions. Since MSM does not make assumptions about the size of the transition region, it is possible that this algorithm performs the same on the networks regardless of the transition region size.

Indeed, our experiments show that for 80% or less transition region, all algorithms perform optimally ($\rho_c = 1$ for MSM) or close to optimally ($\rho_c > 0.92$). For larger transition regions, all algorithms perform progressively worse.

Figure 2 shows the performance of the algorithms, giving the ρ_c score averaged over 5 networks for each transition region size. SCAN and MCL both identify

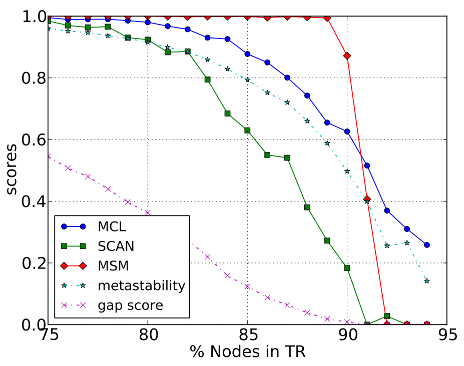


Fig. 2. Comparing the ρ_c score for SCAN, MCL, and MSM on networks with varying transition region size. All networks were generated with 1000 nodes and 5 modules, having the default densities.

only two or three modules, assigning the rest to the transition region. SCAN additionally identifies no hubs or outliers, thus the transition region is a result of small clusters, just as in the case of MCL. MSM separates the modular and transition region well ($\rho_{MT} > 0.95$), identifies five modules in the modular region, but partitions it less than optimally (average $\rho_M = 0.77$). MSM begins to deteriorate a little later than the others, at 89%, but the score decreases fast, with a score of 0 (all nodes are identified as transition region nodes) from 92%. Therefore, MSM is clearly the choice in case the transition region is large, but not too large.

We additionally plot the metastability index of these networks using the ground truth partition. This score also decreases with the size of the modular region, since there are less nodes in modules. The gap scores decreases more quickly, reaching 0.5 when the transition region comprises 75% of the network, but being close to 1 when the transition region is 10% or less.

Experiment 2: Varying module size. As we increase the size of the transition region in Experiment 1, the size of a module decreases automatically, since fewer nodes are now divided into a constant number of 5 modules. Specifically, for a transition region which covers 80% of the network, the corresponding module size is 40, and for 90% it is already 20. To test whether the difference in scores in Experiment 1 is a result of varying the transition region size or of varying the module size, we run a set of experiment where we directly vary the module size. The module size is between 20 and 200, there are 5 modules as before, and the transition region comprises 50% of the network, a value for which all algorithms in Experiment 1 performed perfectly ($\rho_c = 1$). Naturally, to preserve the same proportion of transition region to modular region while varying the total size of the modular region, the overall network size has to change as well, varying from 200 to 2000, respectively. All 3 algorithms performed perfectly for all module sizes: All three ρ scores were 1 or > 0.99. We additionally tested module size

< 20, but the results were unstable due to the small difference between the true and the minimal module size: in some cases the modules were detected correctly, in others, only 9 nodes from a 10-node module were detected, and were assigned to the transition region, causing low scores. Therefore, while a very small module size can negatively influence the algorithm, this effect disappears for slightly larger module sizes, and the low scores of Experiment 1 cannot be fully attributed to the module size, but must be due to the proportion of the transition region.

In the next set of experiments we keep the proportions between the network components constant but vary their densities.

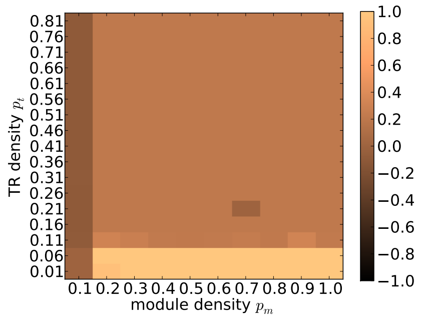
Experiment 3: Varying module and transition region densities. In this experiment, we test combinations of the module density p_m and the transition region density p_t. We take as before networks with 1000 nodes and 5 modules, with the transition region comprising 50% of the network. We additionally set $p_i = 0.01$. For these parameters and the default densities $p_t = 0.01, p_m = 0.6$ all three algorithms performed optimally in the previous experiments.

We set $p_m = 0.1, 0.2, \ldots, 0.9, 1$, and $p_t = 0.01, 0.06, 0.11, \ldots, 0.81$. Figure 3 shows a heatmap for each of the algorithms, giving the ρ_c score for each combination of transition region density and module density. Intuitively, we expect the algorithms to do well when the module density is high and the transition density is low. Indeed, we see that this is the case for all algorithms. SCAN performs the best, erring only when $p_t > 0.45$. The other two algorithms perform optimally when $p_t < 0.06$ and $p_m = 0.8$, and performance quickly deteriorates. Looking more closely at the ρ_{MT} and ρ_M scores, we see that the ρ_M score is perfect while ρ_{MT} is low: the entire transition region is detected as a single module in all these cases. The gap score follows this intuition as well, giving a low score only to networks where the module density is much lower than that of the transition region.

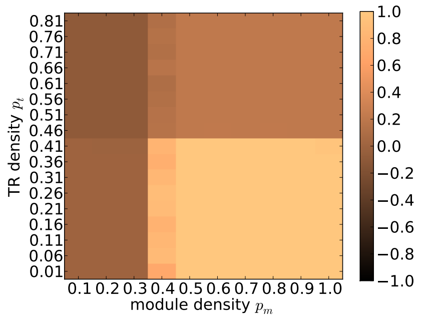
The poor performance of MSM could perhaps be attributed to the fact that the algorithm tends to reward (with a high waiting time) those nodes that have a relatively high degree. Those nodes end up being assigned to modules more often. As the density of the transition region increases, so does the average degree. Since we have fixed p_i at 0.01, and as the modules are smaller than the transition region (each module has size 100, compared to 500 for the transition region), the average degree of nodes in the module is also bounded, and for some values of p_m and p_t, the degrees are about the same, and thus MSM cannot tell them apart as well.

Discussion. Unfortunately, no algorithm comes out the clear leader in every case. MSM identifies modules even when the transition region is large, but does not perform so well when the average degree in the transition region is high. While SCAN performs better than the other algorithms whenever the densities of the transition region and modules are close, in many cases it too identifies the transition region as a module.

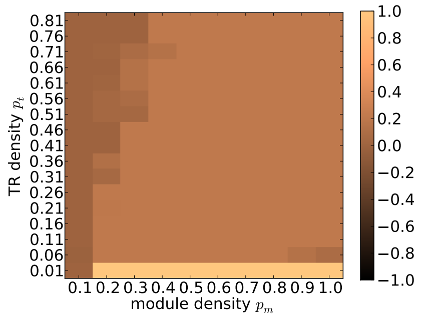
With regards to the different steps of module identification, we first note that MSM performs best the task of guessing the correct number of modules. SCAN



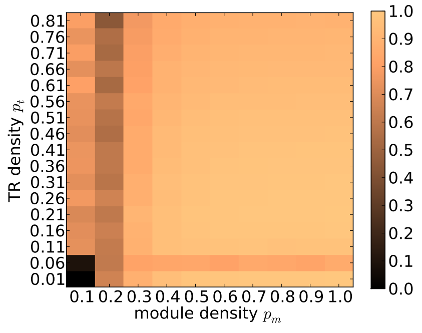
(a) Value of ρ_c for MCL.



(b) Value of ρ_c for SCAN.



(c) Value of ρ_c for MSM.



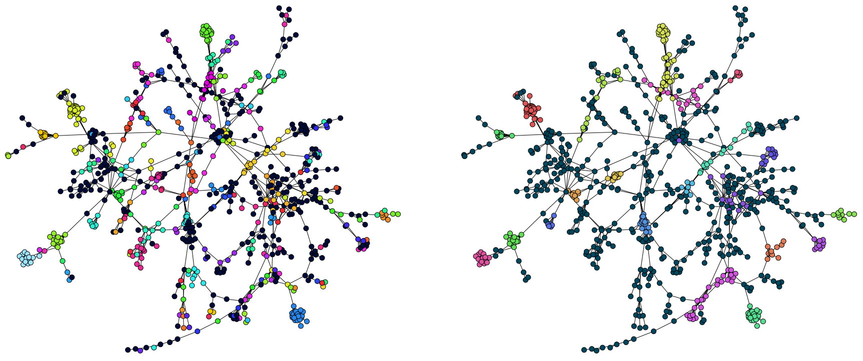
(d) Gap score

Fig. 3. Plotting the ρ_c score for the three algorithms for different combinations of p_m and p_t. All networks have 1000 nodes and 5 modules with 100 nodes each.

and MCL both under-estimate the module number, identifying modules that are too small and are therefore assigned to the transition region. No algorithm over-estimated the number of modules throughout our experiments. On the task of separating the transition region and the modular region (assessed with the ρ_{MT} measure), the three algorithms had successes and shortcomings: In Experiments 1 and 2 the errors were a result of nodes from the modular region being assigned to the transition region. In Experiment 3, the error resulted from the transition region being identified as a single module.

3.2 Experiments on Real-World Networks

We now apply MSM to a real biological network, the well-known FYI network from [7], in order to test whether the results obtained can provide insight about biological truth. The PPI network of *Saccharomyces cerevisiae* was constructed by integrating the results of several large-scale experiments. The outcome is a network whose nodes represent proteins and an edge between two nodes exists if the interaction between the corresponding proteins has been verified by multiple



(a) CYC2008 protein complexes (b) Modules found by MSM

Fig. 4. Modules in the yeast protein interaction network FYI

experiments. We note that we chose to analyze this particular network despite the abundance of more modern and complete yeast protein interaction networks, since it is unweighted and simpler. In fact, the gap score of this network is 0.17, in contrast to the Newman modularity score [12] of 0.90, suggesting that the partition with the highest Newman modularity could contain many nodes not belonging to modules.

We analyze the largest component of the FYI network, containing 778 nodes. We run the MSM algorithm on this network with the same parameters as for the benchmark networks above. Figure 4(b) shows the FYI network with the modules found by MSM in different colors. The black nodes comprise the transition region, with 556 nodes that do not belong to any modules. We identify 21 modules. The largest module contains 58 nodes, and the smallest 14 nodes.

It is a common approach in the study of PPI networks to equate network modules in PPI networks with putative proteins complexes [1, 4]. This approach can be useful for identifying previously unknown complexes, as well as in assigning previously unknown function to proteins: if a particular protein can be grouped together with a set of other proteins, it can be assumed that it has similar properties or functions to those already known about the protein set.

We therefore compare the modules we identified with the protein complexes listed in the CYC2008 [15] dataset. Figure 4 shows our modules side-by-side with the CYC2008 complexes. For this comparison we projected the complexes on the network, including very small complexes with only two proteins and also complexes comprised partially of proteins that are not a part of our network. We find that large complexes such as the 19/22S regulator complex (far left in the figure) with its 17 protein and the cytoplasmic ribosomal small subunit complex (23 proteins, far right) are identified. Many smaller complexes such as the Cytoplasmic exosome complex with 9 proteins are almost completely identified (MSM finds 8 of the proteins). We observe 38.8% of the nodes do not belong to any CYC2008 complex, and thus indeed the gap score could be said to better capture the modularity of the network than the high score given by the Newman modularity. Of course, as the network is somewhat outdated, we cannot use the CYC2008 complexes as a reliable ground truth, but these results indicate that there is promise to our approach.

4 Outlook

We introduced NCC networks, discussed two scores to evaluate their modularity, and compared the behavior of several algorithms on the task of detecting modules in such networks. There are many avenues for further research. We are currently developing a new network score to overcome the disadvantages of the two scores we presented. From the perspective of algorithms, there are many other types of clustering algorithms that might be adapted to NCC networks. One such interesting class is that of methods to identify the densest subgraph (see e.g. [8]), where it could be possible to run the algorithm repeatedly until all modules are identified. The MSM algorithm can be further improved to avoid the pitfalls of a transition region with a high average degree, as seen in Experiment 3. Finally, our random graph model is quite simple. It would be interesting to apply the three algorithms we employed and the scoring function to a richer set of networks, including more real-world examples.

References

[1] G. D. Bader and C. W. Hogue. An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinformatics*, 4, 2003. 10

[2] J. P. Bagrow. Communities and bottlenecks: Trees and treelike networks have high modularity. *Phys. Rev. E*, 85:066118, Jun 2012. 3

[3] A. Bovier, M. Eckhoff, V. Gayrard, and M. Klein. Metastability and low lying spectra in reversible markov chains. *Comm. Math. Phys.*, 228:219–255, 2002. 3

[4] S. Brohée and J. Van Helden. Evaluation of clustering algorithms for protein-protein interaction networks. *BMC Bioinformatics*, 7:488, 2006. 10

[5] A. J. Enright, S. Van Dongen, and C. A. Ouzounis. An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Research*, 30(7):1575–1584, 2002. 5

[6] Z. Feng, X. Xu, N. Yuruk, and T. Schweiger. A novel similarity-based modularity function for graph partitioning. *Data Warehousing and Knowledge Discovery*, pages 385–396, 2007. 2

[7] J.-D. J. Han, N. Bertin, T. Hao, et al. Evidence for dynamically organized modularity in the yeast protein-protein interaction network. *Nature*, 430:88–93, 2004. 9

[8] S. Khuller and B. Saha. On finding dense subgraphs. In *In ICALP '09*, pages 597–608, 2009. 11

[9] A. D. King, N. Pržulj, and I. Jurisica. Protein complex prediction via cost-based clustering. *Bioinformatics*, 20(17):3013–3020, 2004. 1

[10] R. Krause, C. von Mering, P. Bork, and T. Dandekar. Shared components of protein complexes–versatile building blocks or biochemical artefacts? *BioEssays*, 26(12):1333–1343, Dec. 2004. 2

[11] U. Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4): 395–416, 2007. 3

[12] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Pattern Recognition Letters*, 69(2):413–421, 2004. 3, 4, 10

[13] V. Nicosia, G. Mangioni, V. Carchiolo, and M. Malgeri. Extending the definition of modularity to directed graphs with overlapping communities. *Journal of Statistical Mechanics: Theory and Experiment*, 2009:22, 2008. 2

[14] G. Palla et al. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):1–10, 2005. 2

[15] S. Pu, J. Wong, B. Turner, E. Cho, and S. J. Wodak. Up-to-date catalogues of yeast protein complexes. *Nucleic Acids Research*, 37(3):825–831, Feb. 2009. 10

[16] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971. 5

[17] F. Santo. Community detection in graphs. *Physics Reports*, 486(3-5):75 – 174, 2010. ISSN 0370-1573. 1

[18] J. M. Santos and M. Embrechts. On the use of the Adjusted Rand Index as a metric for evaluating supervised classification. In *Proceedings of the 19th International Conference on Artificial Neural Networks: Part II*, ICANN '09, pages 175–184. Springer-Verlag, 2009. 5

[19] M. Sarich, C. Schuette, and E. Vanden-Eijnden. Optimal fuzzy aggregation of networks. *Multiscale Modeling and Simulation*, 8(4):1535–1561, 2010. 2

[20] M. Sarich, N. Djurdjevac, S. Bruckner, T. O. F. Conrad, and C. Schütte. Modularity revisited: A novel dynamics-based concept for decomposing complex networks. *To Appear, Journal of Computational Dynamics*, 2012. URL http://publications.mi.fu-berlin.de/1127/. 2, 3, 5

[21] V. Satuluri, S. Parthasarathy, and D. Ucar. Markov clustering of protein interaction networks with improved balance and scalability. In *Proceedings of the First ACM International Conference on Bioinformatics and Computational Biology*, BCB '10, pages 247–256. ACM, 2010. 5

[22] E. N. Sawardecker, M. Sales-Pardo, and L. A. N. Amaral. Detection of node group membership in networks with group overlap. *EPJ B*, 67:277, 2009. 2

[23] C. Schütte and M. Sarich. *Metastability and Markov State Models in Molecular Dynamics*. Submitted to Courant Lecture Notes, 2013. 3

[24] E. Sprinzak, Y. Altuvia, and H. Margalit. Characterization and prediction of protein-protein interactions within and between complexes. *PNAS*, 103(40):14718–14723, Oct. 2006. 2

[25] M. Weber, W. Rungsarityotin, and A. Schliep. Perron cluster analysis and its connection to graph partitioning for noisy data. *ZIB Report*, 04-39, 2004. 2

[26] X. Xu, N. Yuruk, Z. Feng, and T. A. J. Schweiger. SCAN: a structural clustering algorithm for networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '07, pages 824–833, New York, NY, USA, 2007. ACM. 2, 5

[27] L. Yu, L. Gao, and P. G. Sun. A hybrid clustering algorithm for identifying modules in protein protein interaction networks. *Int. J. Data Min. Bioinformatics*, 4(5): 600–615, Oct. 2010. 1

[28] S. Zhang, R. Wang, and X. Zhang. Identification of overlapping community structure in complex networks using fuzzy cc-means clustering. *Physica A: Statistical Mechanics and its Applications*, 374(1):483–490, 2007. 2