

Clustering assortativity, communities and functional modules in real-world networks

Lovro Šubelj* and Marko Bajec†

University of Ljubljana, Faculty of Computer and Information Science, Ljubljana, Slovenia

(Dated: February 14, 2012)

Complex networks of real-world systems are believed to be controlled by common phenomena, producing structures far from regular or random. Clustering, community structure and assortative mixing by degree are perhaps among most prominent examples of the latter. Although generally accepted for social networks, these properties only partially explain the structure of other networks. We first show that degree-corrected clustering is in contrast to standard definition highly assortative. Yet interesting on its own, we further note that non-social networks contain connected regions with very low clustering. Hence, the structure of real-world networks is beyond communities. We here investigate the concept of functional modules—groups of regularly equivalent nodes—and show that such structures could explain for the properties observed in non-social networks. Real-world networks might be composed of functional modules that are overlaid by communities. We support the latter by proposing a simple network model that generates scale-free small-world networks with tunable clustering and degree mixing. Model has a natural interpretation in many real-world networks, while it also gives insights into an adequate community extraction framework. We also present an algorithm for detection of arbitrary structural modules without any prior knowledge. Algorithm is shown to be superior to state-of-the-art, while application to real-world networks reveals well supported composites of different structural modules that are consistent with the underlying systems. Clear functional modules are identified in all types of networks including social. Our findings thus expose functional modules as another key ingredient of complex real-world networks.

PACS numbers: 89.75.Hc, 89.75.Fb, 89.20.-a, 87.18.-h

I. INTRODUCTION

Networks are the simplest representation of complex systems of interacting parts. Examples of these are ubiquitous in practice, including social networks [1], information systems [2], cooperate ownerships [3] and food webs [4], to name just a few. Despite seemingly plain form, real-world networks commonly exhibit complex structural properties that are absent from regular or random systems [5, 6]. Network complexity arises not from that of individual interactions, but rather from their intrinsic collective behavior. Thus, networked systems are believed to be controlled by common phenomena, which has been the main focus of network science in the last decade [5, 7–9]. Nevertheless, our comprehension of real-world network structure remains to be only partial [10].

Network transitivity or clustering [5, 11], degree mixing [9, 12]—degree correlations of links’ ends—and community structure [8, 13] are perhaps among most widely analyzed network properties in physics literature [14, 15]. Communities are usually seen as densely linked groups of nodes that are only sparsely linked with the rest of the network [8, 16]. These are, at least in context of social networks, considered an artifact of triadic closure [17] or homophily [18, 19], whereas communities also imply assortative—positively correlated—mixing by degree, as long as their sizes differ [20]. On the other hand, recent work suggests that network transitivity, rather than

homophily, is the cause of community structure and degree assortativity in real-world networks [21]. Regardless of the latter, there is substantial evidence that communities and assortative mixing appear concurrently with high clustering, properties also captured by many network models in the literature [12, 22–24].

However, non-social networks greatly deviate from this picture. Biological and technological networks are in fact degree disassortative—negatively correlated—whereas information networks usually exhibit no clear degree mixing [9, 25]. Moreover, many real-world networks contain connected regions of nodes with very low clustering, where classical definition of community does not apply (see Fig. 1). Although one can still partition the network into well separated groups [26], several authors have argued that clear communities emerge only in some (parts of) real-world networks [27–30]. This poses an interesting question: “Are there mesoscopic structures beyond classical communities that could explain for the properties observed in non-social networks?”.

The main purpose of this paper is to expose functional modules [30–33]—groups of regularly equivalent [34] nodes—as a possible answer. Nodes are regularly equivalent if they are linked in the same way to other equivalent nodes (e.g., multi-partite structures). Regular equivalence is a relaxed version of structural equivalence [35] that demands for the nodes to be linked exactly the same. Hence, functional modules refer to groups of nodes that are linked similarly with the rest of the network—have common neighborhoods—and thus perform the same function within the underlying system [30, 36]. Both functional modules and communities can be considered under a general concept of structural modules [37]—

* Electronic address: lovro.subelj@fri.uni-lj.si

† Electronic address: marko.bajec@fri.uni-lj.si

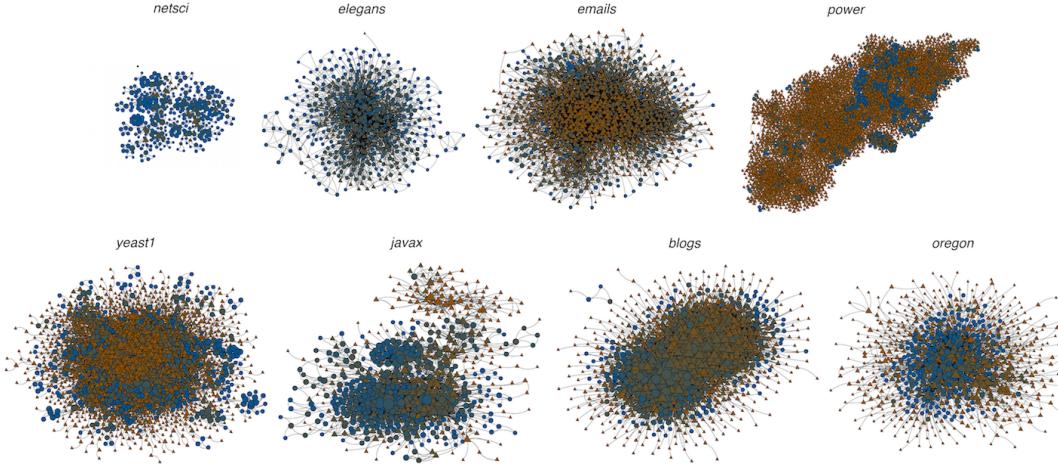


FIG. 1. (Color online) Largest connected components of different real-world networks in Table I. Node sizes are proportional to degrees, whereas symbols correspond to degree-corrected clustering d that ranges from zero (polygons) to one (circles). Observe that d is highly assortative, whereas nodes with particularly high or low d are localized in regions that are characteristic within the underlying system. For example, ignoring nodes with degree one, printing classes in *javax* software network experience average clustering of only $D = 0.176$, while $D = 0.602$ for visualization classes (above and below, respectively). Similarly, liberal and conservative blogs in *blogs* web graph experience $D = 0.435$ and $D = 0.373$ (left-hand and right-hand side, respectively).

groups of nodes with common linking patterns—although some asymmetries exist [30] (e.g., mutual independence). (Note that communities can also be seen as functional modules [31, 32]. However, most pronounced functional modules are disconnected groups of nodes, whereas best communities are, obviously, connected.)

Structure of the paper is nonstandard. We first analyze degree and clustering mixing in a large number of real-world networks of different type and origin (Sec. II). Analysis reveals that degree-corrected clustering [38] is in contrast to standard definition highly assortative in all networks (see Fig. 1)—a distinctive property of real-world networks that was previously unobserved (to our knowledge). We further show that most non-social networks contain a large number of nodes with clustering lower than expected at random, which, together with clustering assortativity, implies connected regions of nodes with extremely low transitivity. Common intuition of dense communities does not coincide with the latter, while adequate extraction of communities from these networks even increases their degree disassortativity.

Functional modules should result in lower clustering and also degree disassortativity. Thus, we propose a simple network model that implicitly introduces functional modules into the network through link copying mechanism [39, 40] that resembles triadic formation [41, 42] (Sec. III). Each node explores the network using the burning process of forest fire model [22, 43], whereas links of visited nodes are copied independently of the latter. This process has a natural interpretation in many information, technological and other networks. Model indeed generates scale-free small-world networks [5, 7] with community structure and degree (dis)assortativity, where clustering and degree mixing are controlled through pa-

rameters. This solves an open problem in network science—what simple local process produces degree disassortativity in real-world networks (to our knowledge). We further show that clustering assortativity is related to the extent of overlap between communities and functional modules. Higher values correspond to clearer separation between different structural modules, which appears to be realized through localization of communities.

The above introduces ‘structural-world’ conjecture: “Real-world networks are composed of functional modules characterizing different roles within the underlying system, and locally overlaid by communities based on some assortative property of the nodes.”. The former explain degree disassortativity and efficient long-range navigation—strength of weak ties [17]—whereas the latter increase the overall clustering and degree assortativity, and provide for efficient local navigation—weakness of strong ties [17]. Note that conjecture deviates from classical comprehension of small-world phenomena [5].

We proceed by presenting an algorithm for detection of arbitrary structural modules without any prior knowledge like the number of modules (Sec. IV). Algorithm exploits label propagation [30, 44] to partition the network into modules, whereas each module is further refined independently of others. Algorithm is shown to be comparable to state-of-the-art in community detection, and superior in detection of structural modules (App. A).

Sec. V first validates the algorithm on various synthetic networks with planted partition and random graphs. Next, application to real-world networks reveals well supported composites of different structural modules that are consistent with characteristics of the underlying system, and structural-world conjecture. Although (almost) all networks contain communities, the network structure

is more accurately predicted by considering also other structural modules. In the case of functional modules, most apparent examples are found in technological, biological, and, surprisingly, also classical social networks.

We further use techniques presented in the paper to also conduct an exploratory analysis of a larger information network (Sec. VI). We extract a citation network from *Cora* dataset [45] that includes computer science publications collected from the web. Analysis reveals skewed size distribution only in the case of functional modules, which is inconsistent with some earlier work [46, 47]. Most pronounced functional modules else mainly arrange in bipartite structures, however, the complexity of linking patterns is much higher than expected.

We conclude the paper in Sec. VII.

II. DEGREE AND CLUSTERING MIXING

Let a network be represented by a simple undirected graph, with $N = \{1, \dots, n\}$ being the set of its nodes and L being the set of its links (denote $m = |L|$). Also, let k_i be degree of node i and k the average degree.

Table I shows common statistic for 24 real-world networks of different size and origin. We consider most types of networks usually found in the literature, whereas detailed description is omitted here. Networks are ordered with respect to degree mixing coefficient r [9]. r measures degree correlations in a network and is just a Pearson correlation coefficient of degrees at links' ends.

$$r = \frac{1}{2m\sigma_k} \sum_{ij} (k_i - k)(k_j - k), \quad (1)$$

where σ_k is standard deviation and the sum goes through all linked pairs (i, j) , $r \in [-1, 1]$ [48]. Positive correlation is indicated by $r \gg 0$, which is known as assortative mixing by degree. Similarly, disassortative mixing refers to negative correlation or, equivalently, $r \ll 0$.

In scale-free networks, which most real-world networks are, r can be seen as a tendency of hubs [49]—high degree nodes—to link between themselves. Observing values of r in Table I one can conclude that the latter is indeed not the case, as most real-world networks are degree disassortative. Only social networks show strong assortativity, whereas most information and some technological networks exhibit no clear degree mixing with $r \approx 0$.

We proceed with an introduction of node clustering coefficient c [5]. c_i measures transitivity around i and is defined as the fraction of linked neighbors.

$$c_i = \frac{t_i}{\binom{k_i}{2}}, \quad (2)$$

where t_i is the number of links among neighbors of i —the number of closed triads—and $\binom{k_i}{2}$ is the number of all possible links, $c_i \in [0, 1]$. For $k_i \leq 1$, $c_i = 0$ by definition. Transitivity of the entire network can be estimated by simply averaging over all the nodes, which is known as (network) clustering coefficient C [5], $C \in [0, 1]$.

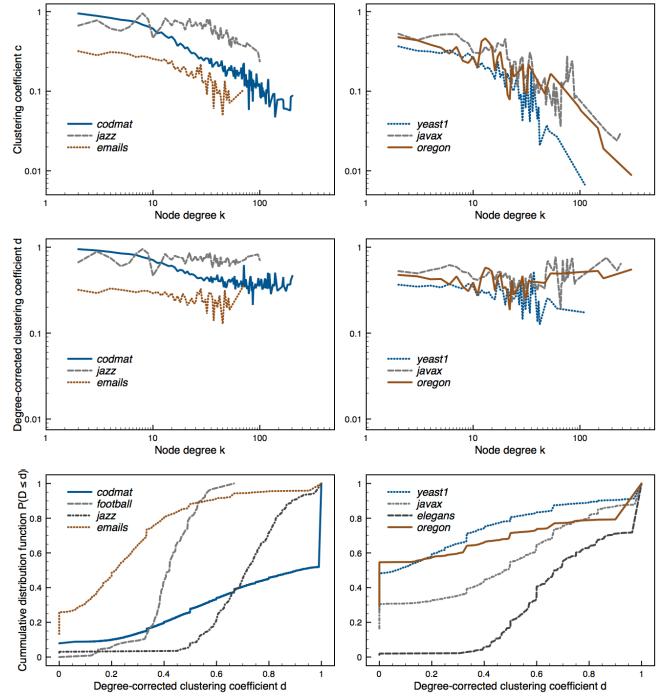


FIG. 2. (Color online) Scaling of clustering in degree assortative and disassortative real-world networks (left and right, respectively). (Nodes with degree at most one are ignored.)

Real-world networks are characterized by much higher C than expected by chance (see Table I). For example, random graph à la Erdős-Rényi [50], where links are laid between nodes with probability $p_r = k/(n-1)$, exhibits only $C = p_r$ in the limit of large n . This is less than 0.02 for almost all networks considered here. Still, in the case of configuration model [11, 51], where graphs are sampled from an ensemble with the same degree sequence as the network, $C \approx p_c$ for large enough n [20, 52].

$$p_c = \frac{(\sum_i k_i^2 - nk)^2}{n^3 k^3} \quad (3)$$

Although p_c scales as n^{-1} , it is not necessarily negligible for networks of moderate size [53] (see below).

Fig. 2 (top) shows scaling of c with respect to node degree in different real-world networks (Table I). Note that c decays with degree resembling a power-law form. Actually, different authors have observed that $c \sim k^{-\alpha}$ with $\alpha \in [0.75, 1]$ in Internet, metabolic, collaboration and other networks [54–56], whereas the same behavior also emerges in a hierarchical network [56]. However, Soffer and Vázquez [38] have shown that this scaling is actually due to node degree mixing. This can be motivated by an observation that hubs would always have low clustering, as the opposite implies a very large clique. Notice, for example, that power-law decays are much steeper for degree disassortative networks than for assortative ones.

Particularly, denominator in Eq. (2) implicitly assumes that every two nodes can form a link between themselves.

| Type | Network | Description | n | m | k | C | D | r | r_c | r_d | $d < p_r$ | $d < p_c$ |
|---------------|------------------------------|-------------------------------------|-------|--------|------|-------|------------|--------|--------|-------|-----------|-----------|
| Collaboration | <i>netsci</i> | Network scientists [57] | 1589 | 2742 | 3.5 | 0.638 | 0.690 | 0.462 | 0.442 | 0.679 | 1% | 1% |
| | <i>condmat</i> | <i>Cond. Mat.</i> archive [58] | 27519 | 116181 | 8.4 | 0.655 | 0.722 | 0.166 | 0.116 | 0.291 | 1% | 1% |
| | <i>comsci</i> | Slovenian computer sci. [59] | 239 | 568 | 4.8 | 0.479 | 0.561 | -0.044 | 0.123 | 0.355 | 6% | 6% |
| Online social | <i>pgp</i> | PGP web of trust [60] | 10680 | 24316 | 4.6 | 0.266 | 0.317 | 0.238 | 0.497 | 0.632 | 27% | 27% |
| | <i>football</i> | American football [8] | 115 | 613 | 10.7 | 0.403 | 0.419 | 0.162 | 0.369 | 0.385 | 0% | 0% |
| Social | <i>jazz</i> | Jazz musicians [61] | 198 | 2742 | 27.7 | 0.617 | 0.703 | 0.020 | 0.008 | 0.198 | 1% | 1% |
| | <i>dolphins</i> | Bottlenose dolphins [62] | 62 | 159 | 5.1 | 0.259 | 0.319 | -0.044 | 0.192 | 0.234 | 15% | 15% |
| | <i>karate</i> | Zachary's karate club [63] | 34 | 78 | 4.6 | 0.571 | 0.666 | -0.476 | -0.229 | 0.277 | 3% | 6% |
| Communication | <i>emails</i> | Emails at university [64] | 1133 | 5451 | 9.6 | 0.220 | 0.253 | 0.078 | 0.214 | 0.317 | 14% | 15% |
| | <i>enron</i> | Emails at <i>Enron</i> [43] | 36692 | 183831 | 10.0 | 0.497 | 0.530 | -0.111 | 0.185 | 0.379 | 4% | 4% |
| Road network | <i>euro</i> | European highways [65] | 1039 | 1305 | 2.5 | 0.019 | 0.025 | 0.090 | 0.395 | 0.499 | 91% | 91% |
| Power grid | <i>power</i> | Western US grid [5] | 4941 | 6594 | 2.7 | 0.080 | 0.100 | 0.003 | 0.469 | 0.653 | 74% | 74% |
| Citation | <i>hepart</i> | <i>H.-E. Part.</i> archive [66] | 27770 | 352285 | 25.4 | 0.312 | 0.353 | -0.030 | 0.132 | 0.370 | 6% | 6% |
| Documentation | <i>javadoc</i> | Javadoc (<i>javax</i>) [67] | 2089 | 7934 | 7.6 | 0.373 | 0.433 | -0.070 | 0.090 | 0.440 | 9% | 9% |
| Protein | <i>yeast1</i> | Yeast <i>S. cerevisiae</i> [47] | 2445 | 6265 | 5.1 | 0.215 | 0.250 | -0.101 | 0.372 | 0.534 | 29% | 29% |
| | <i>yeast2</i> | Yeast <i>S. cerevisiae</i> [68] | 2114 | 2203 | 2.1 | 0.059 | 0.072 | -0.162 | 0.576 | 0.675 | 68% | 68% |
| | <i>javax</i> | Java language (<i>javax</i>) [2] | 1595 | 5287 | 6.6 | 0.381 | 0.440 | -0.120 | -0.041 | 0.545 | 17% | 17% |
| Software | <i>jung</i> | JUNG graph library [2] | 317 | 719 | 4.5 | 0.366 | 0.423 | -0.190 | 0.092 | 0.443 | 21% | 21% |
| | <i>guava</i> | Guava core libraries | 174 | 355 | 4.1 | 0.320 | 0.375 | -0.218 | 0.075 | 0.734 | 34% | 34% |
| | <i>java</i> | Java language (<i>java</i>) [2] | 1516 | 10049 | 13.3 | 0.685 | 0.731 | -0.283 | -0.574 | 0.536 | 1% | 100% |
| Web graph | <i>blogs</i> | Blogs on US politics [69] | 1490 | 16715 | 22.4 | 0.263 | 0.293 | -0.221 | -0.057 | 0.308 | 8% | 13% |
| Metabolic | <i>elegans</i> | Nematode <i>C. elegans</i> [70] | 453 | 2025 | 8.9 | 0.646 | 0.710 | -0.226 | -0.240 | 0.183 | 1% | 3% |
| Internet | <i>oregon</i> | Aut. systems (<i>oregon</i>) [43] | 767 | 1734 | 4.5 | 0.293 | 0.317 | -0.299 | -0.231 | 0.262 | 35% | 70% |
| Bipartite | <i>women</i> | Southern women club [71] | 32 | 89 | 5.6 | 0.000 | 0.000 | -0.337 | | | 100% | 100% |
| Random | Erdős-Rényi graph [50] | | | | | p_r | $\geq p_r$ | 0 | | | | |
| | Configuration model [11, 51] | | | | | p_c | $\geq p_c$ | 0 | | | | |

TABLE I. Common statistics for different real-world networks gathered from the literature and random graph models. (Networks are treated as simple undirected graphs, while module detection algorithms consider multi graphs. Results for random models are valid in the limit of large n (see text for details). Percentages in last two columns ignore nodes with degree at most one.)

Although, this might be true in, e.g., online social networks, where links are generated for 'free', it indeed does not hold for other networks, where node degrees are subjected to, e.g., practical or technological constraints. Degree constraints thus introduce biases into c that are particularly apparent in degree disassortative networks.

Alternative definition of clustering that filters out degree biases has been proposed in the form of degree-corrected node clustering coefficient d [38].

$$d_i = \frac{t_i}{\omega_i}, \quad (4)$$

where ω_i is the number of all possible links between neighbors of i with respect to their degrees, $d_i \in [0, 1]$. (Note that ω can be computed from neighbors degree sequence using a simple algorithm presented in [38].) For $k_i \leq 1$, $d_i = 0$ by definition. Again, clustering of the entire network can be estimated by simply averaging over all the nodes, which is denoted D [38], $D \in [0, 1]$.

Since $\omega \leq \binom{k}{2}$, obviously, $d \geq c$ and $D \geq C$. The latter can be clearly observed in Table I. Fig. 2 (middle) also

shows scaling of d with respect to node degree. No characteristic form occurs, whereas d is rather constant across several scales. Actually, under pseudofractal model introduced in [72], $c \sim 1/k$ implies $d \sim 1/\log k$ [38].

Next, Fig. 1 shows d in different real-world networks in Table I. Interestingly, d looks highly assortative in all networks, whereas nodes with similar d are localized in regions that are characteristic within the underlying system. Note that although communities of highly clustered nodes can be clearly observed, remaining structure does not appear to be captured well by classical models.

Formally, we analyze clustering correlations in these networks by defining mixing coefficient r_d . As in Eq. (1), we adopt Pearson correlation coefficient to measure mixing of clustering at links' ends. (Similarly for r_c .)

$$r_d = \frac{1}{2m\sigma_d} \sum_{ij} (d_i - D)(d_j - D), \quad (5)$$

where σ_d is standard deviation, $r_d \in [-1, 1]$.

Values of r_d reported in Table I reveal that degree-corrected clustering d is indeed highly assortative in all

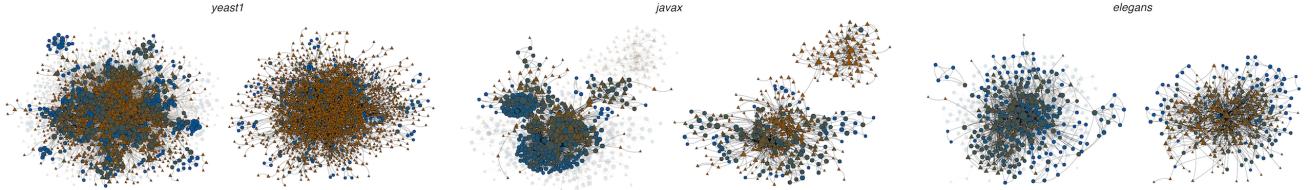


FIG. 3. (Color online) Communities extracted from different real-world networks overlaid over originals, and networks after extraction (left-hand and right-hand side, respectively). Although communities span throughout the networks, many of the links remain unaccounted for. (The number of communities equals 13, 14 and 14, while node symbols are consistent with Fig. 1.)

types of real-world networks. Although obvious to some extent, the latter cannot be considered an artifact of chance, since r_d of a respective Erdős-Rényi random graph is significantly lower [73] (due to no constraints on links). Moreover, same property is absent from standard definition of clustering c , whereas r_c is even negative in some degree disassortative networks (see above). r_d is also consistently much higher than r_c , where correlations above 0.5 are note rare. Thus, in contrast to degree mixing, where $|r| < 0.25$ in most cases, one can in fact accurately predict d based on that of the neighbors. This gives an intuitive explanation of why there exist so many (local) approaches for community detection [74, 75].

Clustering assortativity or, equivalently, $r_d \gg 0$ can thus be regarded as another common property of real-world networks that distinguishes them from mere random world. The latter has not been previously observed and therefore offers various possibilities for future work. Actually, as shown in Sec. III, synthetic generation of networks with $r_d \gg 0$, alongside with other common properties, is not straightforward.

Fig. 2 (bottom) also plots distribution of d in different degree assortative and disassortative networks. One can observe a relatively fast transition in the case of assortative networks, as most nodes share similar high value of d (e.g., *football* and *jazz* networks). This is an expected behavior under community structure model that characterizes social networks. However, in contrast to the latter, 30–55% of nodes in disassortative networks have d close to zero, whereas the distribution is else rather homogeneous (e.g., *yeast1* and *oregon* networks). Thus, despite some exceptions (e.g., *elegans* network), structure of non-social networks goes beyond communities, which also casts doubts on the structure of social networks.

We further investigate peculiar clustering of non-social networks. Table I reports fractions of nodes with d lower than predicted by random models discussed above—lower than p_r and p_c . Although nodes with degree at most one, where d equals zero by definition, are ignored, many networks still contain a large number of nodes with d lower than expected by chance. Notice also a subtle difference between p_r and p_c . For example, p_r for *oregon* Internet map is below 0.01, whereas $p_c = 0.87$. Thus, degree distributions alone can explain for the clustering observed in, e.g., scale-free networks [20].

Note that existence of nodes with very low clustering is

not particularly surprising by itself (see below). However, due to high clustering assortativity of real-world networks, the latter actually implies entire regions of nodes with very low clustering. These were mainly neglected in the past literature, as most authors were focused on nodes with high clustering, and hence community structure [74, 76]. Nevertheless, most networks analyzed here still contain some communities, which we address next.

Zhao *et al.* [29] have already stressed the importance of absence of communities from certain (parts of) real-world networks. They have proposed community extraction framework that guards against the latter. Communities are extracted one by one, where each is selected from a candidate pool according to a quality measure W (see Eq. (6)). When W drops below the value one can expect under the same procedure in a corresponding random graph, the process terminates [77]. Thus, communities are extracted only until statistically significant.

Quality of community W is simply a difference between the links within the community and the links towards the rest of the network normalized appropriately [29]. Let S be a community and S^C its complement (denote $s = |S|$).

$$W = s(n-s) \left(\frac{\sum_{i \in S} k_i^S}{s^2} - \frac{\sum_{i \in S} k_i - k_i^S}{s(n-s)} \right), \quad (6)$$

where k_i^S and $k_i - k_i^S$ are internal and external degree of node i . ($k_i^S = |\Gamma_i \cap S|$, where Γ_i is the set of neighbors of i .) Factor $s(n-s)$ is an adjustment in the spirit of ratio cut [74, 78]. Since $s(n-s)$ is maximized at $s = n/2$, factor penalizes very small or large communities and thus produces more balanced cuts (see [29] for details).

Eq. (6) can be rewritten into more convenient form as

$$W = \sum_{i \in S} \frac{k_i^S n}{s} - k_i. \quad (7)$$

Since work in [29] was focused on community structure alone, each time a community S was extracted, procedure was applied to its complement S^C . However, only the links between the nodes in S are accounted for, whereas those towards the rest of the network—between S and S^C —should not be simply disregarded as random noise. (These links in fact constitute functional modules.)

Framework we adopt here therefore removes only the links within each extracted community. If a node thus

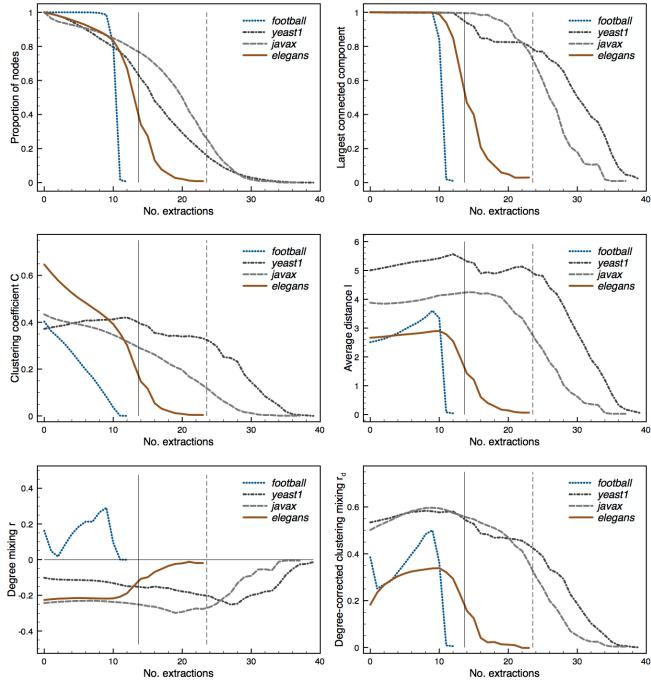


FIG. 4. (Color online) Common statistics during community extraction from different real-world networks. (Results are estimates of the mean over 100 runs, while horizontal lines correspond to termination criteria proposed in [29].)

becomes isolated, it is also removed. Note that this naturally deals with possibly overlapping communities [47] and network hierarchical structure [79, 80]. Hence, same framework might be preferable in different scenarios.

Pool of candidate communities can be generated by directly optimizing Eq. (6), or by applying some community detection algorithm [74, 75]. Due to simplicity, we adopt the latter. Thus, at each step, candidate pool consists of communities identified by the algorithm proposed in [81]. Although not the best community detection approach in the literature [74, 82], it is not limited to only community links, which is in our case essential.

Fig. 4 shows common statistics during extraction of communities from different real-world networks (Table I). Note that, in the case of non-social networks—*yeast1*, *javax* and *oregon* networks—the number of nodes decreases only gradually, whereas the largest connected components consist of almost all nodes until the network suddenly dissolves (Fig. 4 (top)). Structure below communities is thus more complex than commonly presumed and does not consist of merely, e.g., individual links between communities (see Fig. 3). On the other hand, *football* social network appears to contain only communities.

Fig. 4 also indicates the termination criteria proposed in [29] for *javax* and *elegans* networks. However, one can observe notable changes in network properties even before the latter. As dense modules are being extracted from the network, r and C , obviously, decrease (Sec. I). Still, r decreases only until pronounced communities are

found, and starts to increase, when the rest of the structure is bothered. Similarly, one can observe initial increase in r_d , indicating better separated structural modules (see Sec. III), until r_d decreases rather quickly. Due to absence of communities, the algorithm identifies larger groups of isolated nodes, and thus the size of the network also drops suddenly. Any modules beyond communities are therefore characterized by low clustering, degree disassortativity and clustering assortativity.

Observe also that average distance between the nodes denoted l [83]—the number of links in the shortest path—increases only slightly during extraction of communities. Hence, structure below communities in fact provides efficient global navigation throughout the network, whereas communities improve navigation only locally. Thus, l is only slightly higher in the network after extraction. Similar effect was famously observed by Granovetter [17] as the strength of weak ties—links below communities—and the weakness of strong ties—links within communities—and also by other authors more recently [84, 85].

Fig. 3 shows the networks obtained after applying the proposed framework. Communities are extracted as long as the largest connected component consists of more than 99% of the nodes. (Criteria based on, e.g., r_d might be more appropriate for larger networks.) Note that although communities span over large regions of these networks, much of the structure remains unexplained.

We here restate the structural-world conjecture given in Sec. I. Networks can be seen as composed out of two layers. Below, the structure is characterized by functional modules based on the roles played by the nodes within the underlying systems (see Secs. V, VI). As shown in Sec. III, this can explain observed clustering, degree disassortativity and efficient global navigation. Above, networks consist of communities that are overlaid over functional modules, and coincide with some assortative property of the nodes [18, 19]. Communities increase the overall clustering and degree assortativity in the network, and provide for efficient local navigation.

Distinction between layers is merely artificial, adopted to ease the comprehension. Although this naturally relates the conjecture to the concepts of layered or coupled networks [86–88]. Also, as already stressed in [30], semantics behind the links of the two layers necessarily differ. (For incorporation of semantics see, e.g., [89].) Since assortativity coexists with transitivity [21], a relation that produces communities cannot generate, e.g., disconnected functional modules. Thus, different structural modules are expected only in heterogeneous networks. Nevertheless, most networks are heterogeneous [30].

Despite all claims in the paper being investigated thoroughly, we still pose the above as a conjecture. The reason is that low clustering and degree disassortativity characterizing functional modules are already expected properties of, e.g., scale-free networks [20, 90]. More precisely, if a network is reduced to a simple graph, and the largest degree is at least of order \sqrt{n} , the network is likely to be degree disassortative [91] (although only a part can

be accounted for [90]). Moreover, heterogeneous scale-free networks with maximal entropy are degree disassortative networks [92], whereas disassortativity also results from high branching or low clustering [93]. On the other hand, networks are expected to be highly clustered only in the case of some explicit process that introduces transitivity [12, 20]. Hence, preferential attachment model [7] that generates scale-free networks can already interpret at least degree mixing and clustering below communities.

However, the above explains network dynamics only at a level of individual nodes, or gives a macroscopic description of the system. Functional modules, together with structural-world conjecture, provide a mesoscopic view. For example, under scale-free model of Barabási and Albert [7], nodes preferentially link to hub nodes. The same effect can be encountered under structural-worlds, while functional modules give further knowledge about which hubs are likely to be linked simultaneously. The conjecture thus extends the scale-free phenomena. Similarly, small-world model of Watts and Strogatz [5] demonstrates that introduction of long-range links into else highly clustered network significantly decreases the distances between the nodes. Again, functional modules describe how these links are distributed throughout the network, while high clustering is a result of overlaid communities. Structural-world conjecture thus encloses scale-free and small-world phenomena [5, 7] by explaining network dynamics through its mesoscopic structures—functional modules and communities.

Similar characteristics of real-world networks have most notably been investigated in the case of Internet [91, 94], software systems [2, 30], web graphs [30, 95] and biological networks [36, 96], while authors have also considered ensembles of graphs [97, 98] and directed networks [95]. Hao and Li [25] have already observed an apparent dichotomy in degree mixing of biological networks, which is very similar to our structural-worlds. However, their comprehension of the phenomena was limited to communities. Authors have also analyzed robustness [99] and resilience [100], and dynamical processes including spreading [101] and communication flows [102]. Finally, Park and Barabási [103] have shown that two independent parameters are needed to capture network (dis)assortativity, which nicely coincides with our differentiation between two layers of the structural-world.

III. MODEL

Proposed network model is based on the burning process of forest fire (FF) model [22, 43], which we introduce first. Due to simplicity, the model is presented in the case of undirected networks [104].

Let p be the burning probability, $p \in [0, 1]$ (see below). Initially, network consists of a single node, while for each new node i , the burning process proceeds as follows.

- (1) i chooses an ambassador node a uniformly at random, and links to it.

- (2) One samples x_p from a geometric distribution with mean $\frac{p}{1-p}$. i selects x_p random neighbors of a that were not yet visited, j_1, \dots, j_{x_p} , and links to them.

- (3) One recursively applies step (2) to each j_1, \dots, j_{x_p} , where the latter are taken as ambassadors of i .

(If the number of neighbors in step (2) is smaller than x_p , i selects as many neighbors as it can.) Since each node can be visited at most once, the burning process surely converges. Thus, to generate a network with n nodes, FF model repeats the above procedure $n - 1$ times.

The model produces densification and shrinking diameter effects observed in temporal real-world networks, while the networks also exhibit skewed degree distribution, short distances between the nodes and community structure [22, 43]. Hence, the model generates networks with high clustering and degree assortativity (see Fig. 7).

It also has a natural interpretation in, e.g., citation networks. Burning process mimics the author of a paper including references into bibliography. Author first reads a related paper, or selects a paper that triggered the research, and includes it into bibliography (step (1)). Author then considers bibliography of the latter, or additional resources, for other related papers (step (2)). Some of thus discovered papers are further considered and cited, while the author continues similarly as before (step (3)). Despite the above, FF model fails to reproduce the properties observed in citation networks.

Note that described process implicitly assumes that authors read, or at least consider, all the papers they cite. However, this is indeed not the case. For example, seminal work on random graphs conducted by Erdős and Rényi [50] is perhaps among most widely cited papers in the network science literature. Although, presumably, only a small number of authors have actually read the paper. As the work is widely discussed elsewhere, most authors have just copied the reference from another paper. On the other hand, authors also do not cite all papers they read, although directly related to their work. The latter can be simply due to page limitations or, in the case of papers that appear at about the same time, related paper can limit, question or even contradict the work of the author. Nevertheless, such paper would be read thoroughly, while, presumably, many of its references will be further considered and also cited.

Examples suggest that the papers that authors read or cite are selected based on two, not necessarily dependent, processes. We thus propose structural forest fire (SFF) model that adopts the same burning procedure as FF model to traverse the network, whereas links are formed according to another independent process.

Let q be the linking probability, $q \in [0, 1]$ (see below). Initially, network consists of a single link, while for each new node i , the model proceeds as follows.

- (1) i chooses an ambassador node a uniformly at random.
- (2) One samples x_p from a geometric distribution with mean $\frac{p}{1-p}$. i selects x_p random neighbors of a that

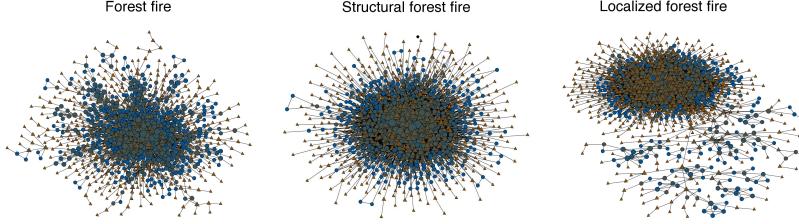


FIG. 5. (Color online) Networks containing 1000 nodes generated with different network models. FF model generates degree assortative networks, while those generated with SFF and LFF models show degree disassortativity (r equals 0.167, -0.189 and -0.124). Clustering C is 0.404, 0.411 and 0.338, while r_d is 0.347, 0.170 and 0.285. (Node symbols are consistent with Fig. 1.)

were not yet visited, j_1, \dots, j_{x_p} .

- (3) One samples x_q from a geometric distribution with mean $\frac{q}{1-q}$. i selects x_q random neighbors of a that were not yet linked, l_1, \dots, l_{x_q} , and links to them.
- (4) One recursively applies step (2) to each j_1, \dots, j_{x_p} , where the latter are taken as ambassadors of i .

(Details are the same as above. [105]) Again, the process converges, whereas the entire procedure is repeated $n-2$ times. Step (3) ensures that no multiple links are formed.

Denote v to be mean number of ambassador nodes visited within the burning process, and let $p < 0.5$. Then,

$$v \leq \sum_i \left(\frac{p}{1-p} \right)^i \leq \frac{1-p}{1-2p}, \quad (8)$$

while the expected degree in the network is

$$k \leq \frac{2vq}{1-q}. \quad (9)$$

Although valid only in the limit of large n , the bounds appear rather tight for small enough p and q (see Sec. VI).

Note that, as node does not necessarily link to the ambassador node, it will fail to form any link—become isolated—with probability $(1-q)^v$. Although the latter is negligible in most cases, this is not the case when one models very sparse networks that imply smaller p and q (e.g., power grids). Isolated nodes are else a common property of real-world networks. However, in practice, these are often ignored in the analysis, or the network is even reduced to the largest connected component.

To obtain a connected network with desired number of nodes, one could simply repeat the burning procedure until the property is reached (for other alternatives see [22, 106]). However, according to the above interpretation, isolated nodes can be seen as authors who failed to find any paper that is indeed similar to their own. The latter can indicate seminal work. In that case, an author would, presumably, attempt to relate the work with existing literature as best as possible. As such process is better imitated by FF model, for the analysis here, a node that remains isolated in the final network is re-incorporated according to the dynamics of FF model

(with same p). This ensures a network with n nodes. (Note that the above refers to only $(1-q)^v(n-2)$ nodes.)

SFF model can generate networks that exhibit most properties observed in real-world networks (see Fig. 5). Clustering assortativity r_d introduced in Sec. II is usually around 0.2, which nicely coincides with, e.g., *elegans* metabolic network that exhibits $r_d = 0.183$. However, most other networks in Table I have r_d much larger than that. By observing these networks in Fig. 1 on can realize that most of them show an interesting phenomena. Communities of highly clustered nodes are localized in certain regions, although these may be scattered across the networks (e.g., *power*, *yeast1* and *javax* networks). (Localization phenomena appears to relate to some property that may vary between the underlying systems.)

We thus also analyze a small variation of the model denoted localized forest fire (LFF) model. Due to simplicity, localization in the model is realized by using isolated nodes. When such node is re-incorporated into the network, the ambassador in step (1) is chosen only between isolated nodes, and not between all the nodes in the network as originally (omitted for first node). As isolated nodes are re-introduced using FF model, communities of highly clustered nodes emerge, while the above variation forces them to localize in certain regions of the network.

LFF model indeed generates networks with higher clus-

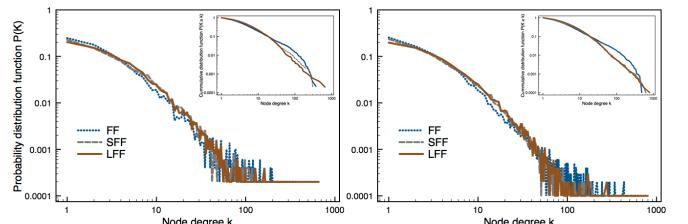


FIG. 6. (Color online) Degree distributions of networks containing 5000 and 10000 nodes generated with different network models (left and right, respectively). Scale-free exponents α for smaller networks generated with SFF and LFF models are 2.61 and 2.85, while 2.82 and 2.88 for larger ones. (Power-law $p_k \propto k^{-\alpha}$ is a plausible fit at p -value = 0.1 in all cases [107].) Average degrees range from 8.1 to 8.7, and thus coincide with density of real-world networks [59, 108].

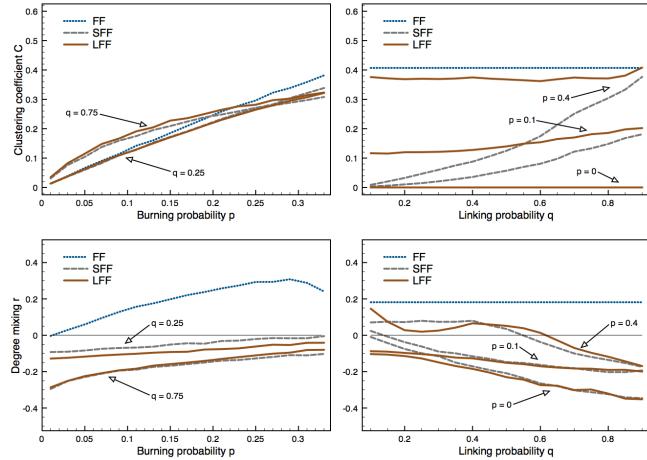


FIG. 7. (Color online) Clustering and degree mixing in networks containing 1000 nodes generated by varying the parameters of different network models. (Results are estimates of the mean over 100 network realizations.)

tering assortativity, where r_d is around 0.3 in most cases. Localization can also be clearly observed in Fig. 5, however, since our intention is merely a statistical analysis of the localization effect, LFF model is not expected to generate networks that would visually resemble real-world networks (like FF and SFF models). Results further indicate that r_d measures the overlap between the nodes with high and low clustering, or, more precisely, how well are the communities separated from the rest of the network. Note also that all models still seriously underestimate r_d .

Both SFF and LFF models generate networks that show scale-free degree distributions (Fig. 6) and small-world phenomena (results omitted). In Fig. 7 we also analyze network clustering and degree mixing by varying the probabilities p and q . p governs clustering in the network for all forest fire models, where clustering increases monotonically with p . Also, p increases degree assortativity in the case of FF model, whereas p has little effect on degree mixing in the case of SFF and LFF models. However, degree mixing in the models is governed by q , where degree disassortativity increases with q . q also does not affect the clustering for LFF model.

The above can be summarized as follows. Notice that, due to a special treatment of isolated nodes, LFF model is in fact identical to FF model for $q = 0$ [104]. On the other hand, the model generates bipartite networks for $p = 0$. (Or k -partite networks if the initial network is a full graph on k nodes.) Hence, for large p and small q , generated networks experience large clustering, degree assortativity and clear community structure. While, for small p and large q , networks show lower clustering and high degree disassortativity. Furthermore, such networks also contain pronounced functional modules, since multipartite networks are the most prominent examples of the latter. In other words, p and q vary the number of nodes on each layer of structural-world networks (Sec. II), while

one can thus generate clustering and degree mixing that resembles almost arbitrary network in Table I [109].

Although not directly discussed above, one can imagine that copying the links of ambassador node—linking to its neighbors—within the proposed models produces functional modules. We stress that the key factor here is that the node does not necessarily link to the ambassador itself. The latter is also the cause for degree disassortativity, whereas linking to the ambassador gives degree assortative networks in any case. Still, degree mixing coefficient r should not be taken as a measure of how pronounced the functional modules are, but is rather related to the difference in the size of the dependent modules. One can expect larger modules in networks that show greater degree disassortativity, whereas low clustering, alongside large average degree, is perhaps the most prominent indicator of functional modules in a network.

Note that functional modules are an implicit result of our models, while this is rather different from some other techniques in the literature. For example, blockmodels [110], which were widely studied in social networks literature in the past, can generate networks with arbitrary structural modules. However, one has to specify how different modules are interlinked between, which is often hard in practice [30, 111].

Other authors have also proposed models very similar to ours [41, 42, 112, 113]. However, these either do not adopt the burning process to traverse the network [106, 114] or the model necessarily links the nodes with their ambassadors [23, 24], which results in degree assortativity. Nevertheless, the main difference in these models is that the set of the nodes that are linked with the concerned node is always a subset of the nodes that are visited. For the models proposed here, these two sets can intersect arbitrarily, while they can also be disjoint. The latter in fact produces rich dynamics observed above.

Last, we also note that, although the interpretation of the models was limited to citation networks, the same process accurately resembles the dynamics in many other networks. For example, in the case of software networks, the process exactly mimics the actions of a developer that is adding classes of code into unfamiliar project. Similarly, in air transportation networks, an airport might offer the same flights as a nearby airport—and would thus copy its links—while it will, presumably, not offer a flight towards the latter. Finally, the same process can also be undertaken by a person that is being familiarized with friends of an acquaintance. Although the person does not become a friend of the acquaintance, he or she may still befriend with some of the acquaintance's friends.

IV. ALGORITHM

Despite the discussion in Sec. II, proposed algorithm directly partitions the network into structural modules without first extracting communities. The algorithm is based on a label propagation framework proposed in [30],

$$g_i = \operatorname{argmax}_g \left[\nu_g \sum_{j \in \Gamma_i} f_j \cdot \delta(g_j, g) + (1 - \nu_g) \sum_{j \in \Gamma_i} \sum_{l \in \Gamma_j \setminus \Gamma_i} \tilde{f}_l / k_j \cdot \delta(g_l, g) \right] \quad (10)$$

which we briefly introduce below. Due to simplicity, the framework is presented in the case of simple graphs.

Denote Γ_i to be the set of neighbors of node i and let g_i be an unknown module (or group) label. Furthermore, let \mathcal{G}_i be the set of nodes sharing label g_i . Propagating labels between the nodes was first introduced for community detection. Raghavan *et al.* [44] have proposed a simple algorithm that exploits the following procedure. Initially, each node is labeled with a unique label, $g_i = i$. Then, at each iteration, a node adopts the label shared by most of its neighbors (updates occur in a random order). Hence,

$$g_i = \operatorname{argmax}_g \sum_{j \in \Gamma_i} \delta(g_j, g), \quad (11)$$

where δ is the Kronecker delta (ties are broken uniformly at random [115]). Due to many links within communities, relative to the number of links towards the rest of the network, nodes in communities form a consensus on some label after a few iterations. Thus, when an equilibrium is reached, $\{\mathcal{G}\}$ contains communities that are clearly depicted in the network structure. Note that Eq. (11) is actually equivalent to a simple kinetic Potts model [116].

Due to extremely fast structural inference of label propagation, the algorithm exhibits near linear complexity $\mathcal{O}(m^{1.2})$, while the expected number of iterations on a network with a billion links is only 113 [117].

The basic algorithm can be further improved by also applying node preferences [118]. Preferences adjust the strength of propagation from certain nodes (e.g., hubs), and thus force the propagation process towards more desirable partitions. Let f_i be a preference of node i . Then,

$$g_i = \operatorname{argmax}_g \sum_{j \in \Gamma_i} f_j \cdot \delta(g_j, g). \quad (12)$$

Note that preferences can be set to an arbitrary node property (e.g., degree). We adopt the same preferences f and \tilde{f} (see Eq. (10)) as in [30], whereas detailed description is omitted here (due to space limitations). However, f , \tilde{f} are in fact composed out of two factors. First factor corresponds to balanced propagation [65] that, at each iteration, decreases the propagation strength from the nodes that are considered first, and increases the strength from the nodes that are considered last. This counteracts for the randomness introduced through random update orders, which stabilizes the propagation process [119].

Second factor corresponds to defensive propagation [117] that further increases the propagation strength from the core of each \mathcal{G} or, equivalently, decreases the strength from its border. The latter forces the algorithm to gradually reveal the network structure and improves its detection strength in real-world networks [67, 117].

The algorithm in Eq. (12) is comparable to current state-of-the-art in community detection, while Šubelj and Bajec [30] have extended the same principle to also functional modules. Rather than propagating the labels between the neighboring nodes, one propagates the labels between the nodes at distance two—through common neighbors. Since nodes in functional modules share many common neighbors, similarly as before, they form a consensus on some particular label. Thus, when the process unfolds, $\{\mathcal{G}\}$ contains most pronounced functional modules in the network. Generalization is identical to a standard propagation on a network with the same set of nodes, while the links represent (link-disjoint) paths of length two between the nodes in the original network.

The algorithm for functional modules is shown in the right-hand side of Eq. (10). \tilde{f}_i is preference of node i (see above), whereas k_j is a normalization. Since the sum in Eq. (12) has k_i terms, whereas the sums in the right-hand side of Eq. (10) contain up to $\sum_{j \in \Gamma_i} k_j$ terms, this introduces biases in the detection of modules. Dividing each term by k_j aggregates the contributions at each neighbor of i , which makes all sums proportional to k_i .

Eq. (10) shows complete algorithm for detection of arbitrary structural modules denoted generalized propagation [30]. Left-hand side is the same as Eq. (12), while ν_g are module dependent factors that represent adopted network modeling, $\nu_g \in [0, 1]$. Setting $\nu_g = 1$ for all g is identical to community detection algorithm in Eq. (12), whereas, for all ν_g equal to zero, the algorithm reveals only functional modules. When $\nu_g = 0.5$, identified modules depend on community and functional links.

General propagation framework in Eq. (10) can detect communities and functional modules even when only weakly depicted in the structure of the network [30, 120]. However, module factors ν_g have to be defined accordingly (see App. B). Note that ν_g are set apriori, when one initializes node labels $\{g\}$. As most labels disappear during propagation, one only has to provide sufficient number of labels with $\nu_g \approx 1$ in regions of the network, where communities could exist, and enough labels with $\nu_g \approx 0$ in regions, where one expects functional modules.

Network modeling proposed in [30] was based on conductance [121], while the algorithm in [120] uses clustering coefficients c and C [5]. Due to the analysis in Sec. II, we propose a much simpler model based on degree-corrected clustering coefficients d and D [38].

$$\nu_g = \begin{cases} 1 & \text{for } D \geq \tau \wedge d \geq \tau \\ 0 & \text{for } D < \tau \wedge d < \tau \\ 0.5 & \text{otherwise} \end{cases} \quad (13a)$$

$$\nu_g = \begin{cases} 1 & \text{for } D \geq \tau \wedge d \geq \tau \\ 0 & \text{for } D < \tau \wedge d < \tau \\ 0.5 & \text{otherwise} \end{cases} \quad (13b)$$

$$\nu_g = \begin{cases} 1 & \text{for } D \geq \tau \wedge d \geq \tau \\ 0 & \text{for } D < \tau \wedge d < \tau \\ 0.5 & \text{otherwise} \end{cases} \quad (13c)$$

Parameter τ is expected clustering in a corresponding

random graph, where Erdős-Rényi [50] graph with clustering p_r and configuration model [11, 51] with p_c are the most obvious choices (Sec. II). For the analysis here, we set τ to p_c , to reveal modules that go beyond (scale-free) degree distributions (if not stated otherwise).

Eq. (13) should be seen as follows. Since most real-world networks have $D \geq \tau$, the algorithm identifies communities in regions with high clustering (Eq. (13a)). Eq. (13b) properly models, e.g., bipartite networks, where functional modules are revealed in regions with very low clustering. Otherwise, $\nu_g = 0.5$ (Eq. (13c)).

Note that, according to structural-world conjecture, functional modules also emerge below communities, in regions with high clustering. The model in Eq. (13) thus apparently ignores most functional modules in the network. However, different modules commonly become obscure in the presence of communities. Let a network contain a complete k -partite subgraph on n nodes. Obviously, the network contains clear functional modules. Still, when k goes to n , the corresponding subgraph actually becomes a clique, and thus a well defined community. Similar effect occurs, when the size of subgraph decreases or, equivalently, n goes to k . Hence, functional modules often cannot be revealed alongside communities.

The above is directly related to the following question. Assume that networks truly contain functional modules as structural-world predicts; how that there exist numerous studies in the literature, where authors have identified communities in almost any type of real-world networks? Analysis in Sec. V shows that community detection algorithms commonly identify dependent functional modules as a single group of nodes. Nonetheless, the latter can still be a well defined community. Actually, some of the communities in the famous *football* social network (Table I) are in fact multi-partite graphs (see Fig. 9).

Although this indicates a rather undesirable behavior of community detection algorithms, it can be employed for detection of other modules. The model in Eq. (13) thus first tries to identify dependent functional modules as communities, which are then (possibly) refined into functional modules. Let $\{\mathcal{G}\}$ be the initial partition revealed by the algorithm. For each \mathcal{G} , the algorithm is further applied to a subnetwork induced by the nodes in \mathcal{G} . As this refinements proceed recursively, an entire sub-hierarchy of modules is obtained. Similarly, one can reveal a super-hierarchy by applying the algorithm to a super-network induced by $\{\mathcal{G}\}$ (agglomeration). Here (super-)nodes represent modules that are linked, when a link between their nodes also exists in the network. Final result of such divisions and agglomerations is a complete hierarchy of modules denoted \mathcal{H} [79, 80] (see Fig. 9).

Leafs of \mathcal{H} represent nodes in the network, while inner nodes $\{\mathcal{I}\}$ correspond to modules $\{\mathcal{G}\}$ that were obtained over several applications of the algorithm. Let \mathcal{I} correspond to module \mathcal{G} and let $\mathcal{I}_1, \dots, \mathcal{I}_t$ be the ancestors of \mathcal{I} . Furthermore, let $\mathcal{H}_{\mathcal{I}}$ be a sub-hierarchy rooted at \mathcal{I} . Thus, leafs of $\mathcal{H}_{\mathcal{I}}$ are nodes in module \mathcal{G} , while $\mathcal{G}_1, \dots, \mathcal{G}_t$ is a partition of the subnetwork induced by \mathcal{G} . Let each

inner node \mathcal{I} (or module \mathcal{G}) also be associated with value θ , $\theta \in [0, 1]$, which is defined as the probability that two nodes in \mathcal{G}_i and \mathcal{G}_j are linked, $i \neq j$. Let there be m such links in \mathcal{G} and denote s to be the number of all possible links, $s = \sum_{i < j} |\mathcal{G}_i||\mathcal{G}_j|$. Then, $\theta = m/s$. (If ancestors of \mathcal{I} are leafs of \mathcal{H} , θ is just the density of module \mathcal{G} .)

Probabilities θ are in fact maximum likelihood estimators for hierarchy \mathcal{H} [122], while the posterior probability of \mathcal{H} —likelihood \mathcal{L} given the network observed—is

$$\mathcal{L}(\mathcal{H}) = \prod_{\mathcal{I} \in \mathcal{H}} (\theta_{\mathcal{I}})^{m_{\mathcal{I}}} (1 - \theta_{\mathcal{I}})^{s_{\mathcal{I}} - m_{\mathcal{I}}}. \quad (14)$$

Analysis in Sec. V reports log-likelihoods $-\log \mathcal{L}$, where smaller values are better. Note that $\log \mathcal{L}$ of $\{\mathcal{G}\}$ is the entropy of the corresponding blockmodel [123].

When a network contains only communities, initial partition $\{\mathcal{G}\}$ would commonly already identify them. Thus, $\{\mathcal{G}\}$ should not be further refined, which requires an extra criteria. A promising approach is to refine only very sparse modules \mathcal{G} that clearly do not coincide with the definition of a community, e.g., when a subnetwork induced by \mathcal{G} have $D < \tau$ (results omitted). For the analysis here, \mathcal{G} is refined when $|\mathcal{G}| > 3$, while only refinements with $\mathcal{L}(\mathcal{H}_{\mathcal{I}}) > \mathcal{L}(\mathcal{G})$ are accepted— $\mathcal{H}_{\mathcal{I}}$ is the revealed hierarchy for \mathcal{G} , and $\mathcal{L}(\mathcal{G})$ is the likelihood of a hierarchy with a single (inner) node and leafs from \mathcal{G} .

Proposed algorithm thus constructs an entire hierarchy of modules \mathcal{H} and is denoted hierarchical propagation (HP) algorithm. When only a partition of the network is required, modules represented by the bottom-most (inner) nodes in \mathcal{H} are reported (no agglomerations are needed). We also consider an algorithm without module refinements, which is, for consistency with [30], denoted generalized propagation (GP) algorithm.

Complexity of HP algorithm is near ideal. Each propagation of labels (Eq. (10)) requires $\mathcal{O}(km)$, where m is the number of links and k the average degree. The number of iterations required for the propagation process to converge can be, according to [117] and above analogy, estimated to $\mathcal{O}((km)^{0.2})$. Since the number of module refinements is usually negligible, the total complexity becomes $\mathcal{O}((km)^{1.2})$. When an entire hierarchy is required, the latter refers to a single level. (For the analysis here, we limit the maximum number of iterations to 100.)

HP algorithm is first validated on synthetic benchmark networks, and applied to different real-world networks (Sec. V). Next, the algorithm is adopted for the analysis of a citation network (Sec. VI), where communities have been extracted apriori. Comparison with 12 state-of-the-art algorithms is conducted in App. A, while App. B also analyses different network models (Eq. (13)).

V. STRUCTURAL MODULES

Results in the following (and in Apps. A, B) are reported in terms of $\log \mathcal{L}$ (Eq. (14)), normalized mutual information [124] (NMI) and adjusted rand index [125]

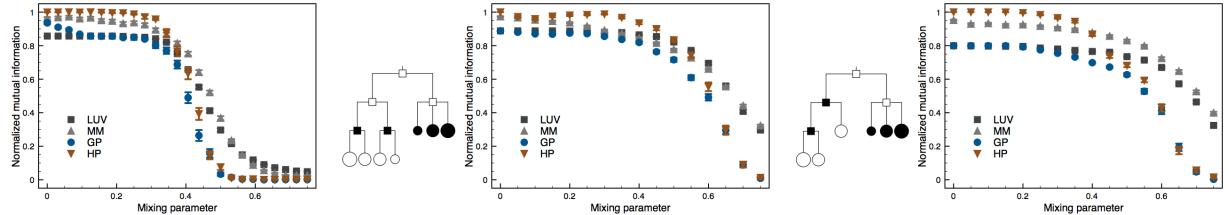


FIG. 8. (Color online) Comparison of module detection algorithms on GN2, HN7 and HN6 synthetic networks (left to right, respectively). (Results are estimates of the mean over 100 network realizations, while bars show standard error of the mean.)

(ARI). NMI equals to the mutual information of the true network partition, and partition of modules revealed by the algorithm, that is normalized by their entropies, $NMI \in [0, 1]$. ARI measures the fraction of node pairs that are classified the same in both partitions, subjected to the expected value in a null model, $ARI \in [0, 1]$. For both measures, identical partitions experience 1, while the expected value for independent partitions is zero.

Proposed HP algorithm is compared against a variation without module refinements denoted GP algorithm (Sec. IV), and a variation limited to merely communities denoted CP algorithm (see App. B). We also consider two other approaches. First is a community detection algorithm known as Louvain method [126] that optimizes modularity [127] by multi-level aggregation (LUV algorithm). Second is a (general) structural module detection algorithm, which fits a predefined mixture model using expectation-maximization [128] technique [31] (MM algorithm). Among 13 state-of-the-art algorithms in App. A, both represent the second best approach in their category (after that in [129] and HP algorithm).

MM algorithm demands the number of modules to be known beforehand, which is set to the true value in all cases. (Due to different stopping criteria, results for MM algorithm are slightly better than in [30, 120].)

A. Synthetic networks

Algorithms are first applied to three synthetic benchmark networks with planted partitions of structural modules. All partitions consist of communities and functional modules, while the structure is controlled by a mixing parameter μ , $\mu \in [0, 1]$. When μ equals zero, all links in the network are placed according to the predefined partition, whereas the structure degenerates with increasing μ .

Fig. 8 (left) plots the results for GN2 synthetic networks [36] that are a generalization of a classical community detection benchmark [8]. Networks consist of four modules with 32 nodes, where two modules are classical communities, while the other two form a bipartite structure of functional modules. Average degree is fixed to 16.

Only HP algorithm can accurately reveal the planted structure in these networks. Observe that performance of a community detection algorithm (e.g., LUV algorithm) is only slightly worse compared to a general module de-

tection algorithm (e.g., MM algorithm). The former identifies functional modules in the network as a single group of nodes, however, this is only weakly expressed by standard measures used in the literature [74, 124]. As already discussed in Sec. IV, HP algorithm first reveals communities in these networks, which are then refined into functional modules (see GP algorithm).

Next, algorithms are applied to two synthetic networks that are based on a hierarchical model [79, 80] (Sec. IV). Respective hierarchies are shown in Fig. 8, where shades of nodes correspond to values of θ that are set to either μ or $1 - \mu$. For $\mu = 0$, $\theta \in \{0, 1\}$. Leafs of hierarchies represent actual planted modules with 8, 16 or 24 nodes. Networks thus contain 7 and 6 modules, and are denoted HN7 and HN6 networks. Note that both networks consist of three communities, whereas functional modules form two bipartite structures in the case of HN7 networks, and a tripartite structure in the case of HN6 networks.

Results for HN7 networks are shown in Fig. 8 (middle), while Fig. 8 (right) plots results for HN6 networks. Again, only HP algorithm can accurately detect the true structure planted into these networks, while the performance of other algorithms is else similar as above.

HP and GP algorithms are further applied to Erdős-Rényi [50] random graphs and Barabási-Albert [7] scale-free networks that, presumably, contain no structural modules. Number of nodes is fixed to 128 in both cases, while average degree is varied from 2 to 32. When degree exceeds a certain threshold, none of the algorithms reveal any structure in these networks. The transition occurs between 8 and 16. The algorithms also do not suffer from the resolution limit problem [130] (results omitted).

We conclude that HP algorithm can detect arbitrary structural modules—communities or functional modules—when they are present in the network. Also, comparative analysis in App. A shows that the algorithm outperforms all state-of-the-art approaches considered. As the number of modules is an implicit result of the propagation process employed, the value does not have to be set apriori as for most other algorithms [31, 111]. Hence, proposed algorithm represents a promising approach for exploratory analysis of real-world networks.

Next, we adopt HP algorithm for the analysis of real-world networks, with special emphasis on the context of different modules within the underlying systems.

B. Real-world networks

We first compare the algorithms on four real-world networks from Table I. We adopt three classical social networks—*football*, *karate* and *women* networks—with known sociological classification of the nodes that results from earlier studies [8, 63, 71]. For *football* and *karate* networks, corresponding network modules are communities that coincide with some assortative property of the nodes, whereas, in the case of *women* network, the partition corresponds to functional modules that represent different roles nodes play within the underlying domain—women and events attended by the former. We also adopt *jung* software network, where nodes represent classes of code that constitute JUNG library [131]. Here software packages decided by the developers are taken as the true network partition. Respective structural modules are thus communities of classes implementing common functionality, and functional modules representing the roles of classes within JUNG project [2, 30] (see below).

The results are shown in Table II. Proposed HP algorithm most accurately reveals the true partition in all cases except for *karate* network. More precisely, the algorithm identifies three communities (on average), while sociological partitioning contains two. Nevertheless, partition with three communities is somewhat more consistent with the structure of the network, and thus commonly reported by the algorithms in the literature [132, 133].

| Network | NMI | | | | ARI | | | |
|-----------------|-------|--------------|-------|--------------|-------|--------------|-------|--------------|
| | LUV | MM | CP | HP | LUV | MM | CP | HP |
| <i>football</i> | 0.876 | 0.823 | 0.905 | 0.909 | 0.771 | 0.683 | 0.841 | 0.850 |
| <i>karate</i> | 0.629 | 0.912 | 0.834 | 0.866 | 0.510 | 0.912 | 0.823 | 0.861 |
| <i>jung</i> | 0.605 | 0.662 | 0.650 | 0.684 | 0.269 | 0.276 | 0.218 | 0.280 |
| <i>women</i> | 0.309 | 0.825 | 0.217 | 0.932 | 0.174 | 0.716 | 0.119 | 0.936 |

TABLE II. Comparison of module detection algorithms on different real-world networks with known partitioning. (Results are estimates of the mean over 100 runs.)

Although structural modules in, e.g., *football* and *women* networks are fundamentally different in many basic module properties, HP algorithm accurately detects the structure that is present in each network. We thus also apply the algorithm to a larger number of real-world networks, to analyze, whether general modules—communities and functional modules—better model the network structure than communities alone. For a fair analysis, we compare modules revealed under the same framework, exemplified by HP and CP algorithms. (For comparison, we also report results for other approaches.)

Table III shows $\log \mathcal{L}$ of hierarchies of modules identified with different algorithms (as described in Sec. IV). We consider 16 real-world networks from Table I that are reduced to the largest connected components. Observe that general modules better predict the structure present

| Network | Groups | – log \mathcal{L} | | | |
|-----------------|--------|---------------------|---------|----------------|----------------|
| | | LUV | MM | CP | HP |
| <i>netsci</i> | | 2622.3 | | 2024.7 | 2152.0 |
| <i>comsci</i> | | 2399.7 | | 2011.0 | 3326.5 |
| <i>football</i> | 12 | 1184.3 | 1138.5 | 1351.7 | 1096.0 |
| <i>karate</i> | 2 | 196.3 | 178.0 | 195.8 | 188.2 |
| <i>emails</i> | | 25903.6 | | 24287.7 | 24402.8 |
| <i>euro</i> | | 5954.1 | | 4202.1 | 4072.3 |
| <i>power</i> | | 33874.6 | | 21132.9 | 20678.8 |
| <i>javadoc</i> | 22 | 23011.5 | 19943.4 | 25361.7 | 19405.7 |
| <i>yeast1</i> | | 32621.3 | | 28280.9 | 28689.4 |
| <i>yeast2</i> | | 15766.8 | | 12826.9 | 12181.3 |
| <i>javax</i> | 21 | 16140.7 | 13728.9 | 17299.2 | 13497.5 |
| <i>jung</i> | 38 | 3081.6 | 2583.8 | 2997.5 | 2497.4 |
| <i>guava</i> | 4 | 1418.1 | 1020.4 | 1295.8 | 1012.1 |
| <i>elegans</i> | | 8673.5 | | 8955.7 | 8856.9 |
| <i>oregon</i> | | 9351.5 | | 9865.6 | 10456.6 |
| <i>women</i> | 3 | 193.3 | 204.2 | 163.6 | 207.4 |

TABLE III. Analysis of hierarchical structures revealed with module detection algorithms in different real-world networks. (Results are estimates of the mean over at least 100 runs, while ‘groups’ corresponds to a known partitioning.)

in most types of real-world networks including information (e.g., *javadoc* network), biological (e.g., *yeast2* and *elegans* networks), technological (e.g., *power* and *jung* networks) and, surprisingly, also classical social networks (e.g., *football* and *karate* networks). On the other hand, collaboration networks appear to be the most prominent examples of networks with only community structure (e.g., *netsci* and *comsci* networks).

Note that, e.g., *karate* and *elegans* networks are in fact best modeled by considering merely network modularity (see LUV algorithm). However, the latter is actually due to increased complexity, when one considers hierarchies

| Network | Runs | – log \mathcal{L} and no. levels | | | | |
|-----------------|--------|------------------------------------|----|-----------------|------|----------------|
| | | CP | HP | p_r and p_c | [80] | |
| <i>football</i> | 10^4 | 1010.9 | 3 | 954.8 | 5 | 1004.1 |
| <i>karate</i> | 10^5 | 174.1 | 3 | 172.3 | 3 | 173.9 |
| <i>euro</i> | 10^3 | 4108.9 | 6 | 3883.2 | 8 | 3924.4 |
| <i>yeast2</i> | 10^2 | 12495.0 | 6 | 11611.2 | 7 | 11596.4 |
| <i>javax</i> | 10^2 | 13020.7 | 4 | 12894.1 | 4 | 11512.2 |
| <i>jung</i> | 10^3 | 2354.5 | 5 | 2312.5 | 4 | 2272.9 |
| <i>elegans</i> | 10^2 | 8734.1 | 5 | 8640.9 | 6 | 8243.3 |
| <i>women</i> | 10^4 | 193.9 | 2 | 163.6 | 1 | 163.6 |

TABLE IV. Analysis of hierarchical structures revealed with module detection algorithms in different real-world networks. (Results are peak values estimated over at least 100 runs, while last column corresponds to (binary) hierarchies in [80].)

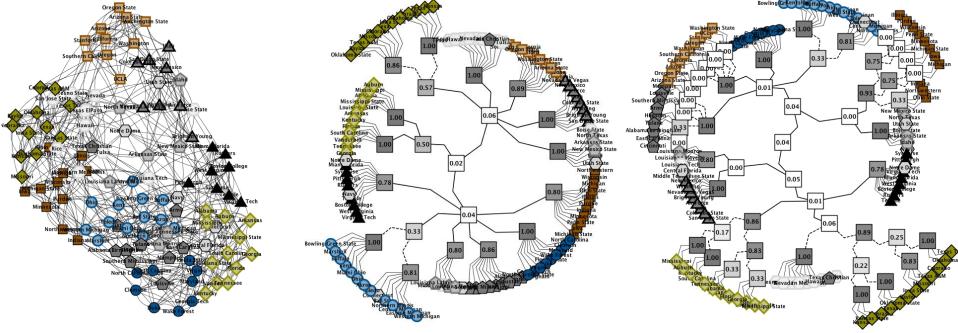


FIG. 9. (Color online) Hierarchies of *football* social network revealed with CP and HP algorithms (left-hand and right-hand side, respectively). (Node symbols represent NCAA college (American) football conferences and are consistent among figures. Values in inner nodes of hierarchies equal θ , whereas solid and dashed links correspond to agglomeration and division steps.)

of general structural modules. For example, assume that the algorithm reveals several functional modules at some level of the hierarchy. In order to adequately model the structure present, dependent functional modules must necessarily be detected as a single group on the next level.

Table IV thus also shows $\log \mathcal{L}$ for the best hierarchies revealed in 8 real-world networks from Table III. Despite the discussion above, HP algorithm still most accurately predicts the hierarchical structures in these networks. Results also reveal that p_c —expected clustering in a configuration model [11, 51]—more adequately distinguishes between different types of modules than p_r —clustering of an Erdős-Rényi random graph [50] (see Eq. (13)). Nevertheless, since p_c is merely an upper bound for C [52], p_r proves to be more appropriate for smaller or very sparse networks (e.g., *karate* and *euro* networks).

Table IV further shows the number of (non-trivial) levels that constitute each hierarchy—height of the hierarchy. Observe that hierarchies of communities generally consist of much larger number of levels than those that include also other structural modules. Since the complexity of the hierarchy increases exponentially with its height, the difference is substantial. Table IV also reports binary hierarchies from [80] that were revealed by Monte Carlo sampling scheme. Although values of $\log \mathcal{L}$ are better than those obtained with, e.g., HP algorithm, heights of respective hierarchies are again considerably larger.

We conclude that general structural modules more adequately model the mesoscopic structure of networks than communities alone. Next, we consider modules identified in different types of real-world networks in greater detail.

Fig. 9 shows hierarchies of *football* social network revealed with CP and HP algorithms. Nodes in the network represent college (American) football teams that are linked, when a game was played during the NCAA 2000 season. Known partitioning of the network corresponds to a division into conferences. Observe that both hierarchies identify conferences at some intermediate level, whereas many are further partitioned into different structural modules. More precisely, although some conferences correspond to a clique in the network, others

are in fact multi-partite structures (see Fig. 9 (right)). The latter can be directly related to the schedule of the games played, and thus the role of different teams during the 2000 season. Hence, despite the fact that the network represents a classical benchmark for community detection, hierarchy of merely communities fails to identify most of the structure present (see Fig. 9 (left)).

Fig. 11 (right) also shows hierarchy of *jung* software network (see above) revealed with HP algorithm. Modules again coincide with the known classification of the nodes—packages of software classes—whereas different modules show characteristic features of JUNG project. For example, classes implementing the same functionality (e.g., graph implementations) commonly correspond to a community in the network, while classes with the same role within the project (e.g., parsers or plug-ins), which depend on a common set of other classes, are usually expressed as functional modules. Again, much of the functionality of JUNG library would remain obscure under the framework limited to communities (see also [30]).

Last, Fig. 10 shows structural modules in *javax* software and *elegans* metabolic networks revealed with CP and HP algorithms. Observe that most prominent functional modules in *javax* network are revealed in regions with lower clustering—indicated by strong off-diagonal structure—while communities mostly exist in regions with higher clustering. However, the latter is not

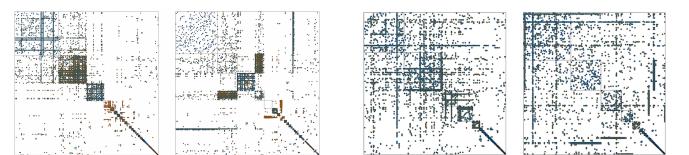


FIG. 10. (Color online) Structural modules—blockmodels—of *javax* software and *elegans* metabolic networks revealed with CP and HP algorithms (left-hand and right-hand side, respectively). (Dots represent links, whereas shades correspond to average d at link ends and range between 0 (mocha) and 1 (blue). Dots are enlarged five times for better visibility.)

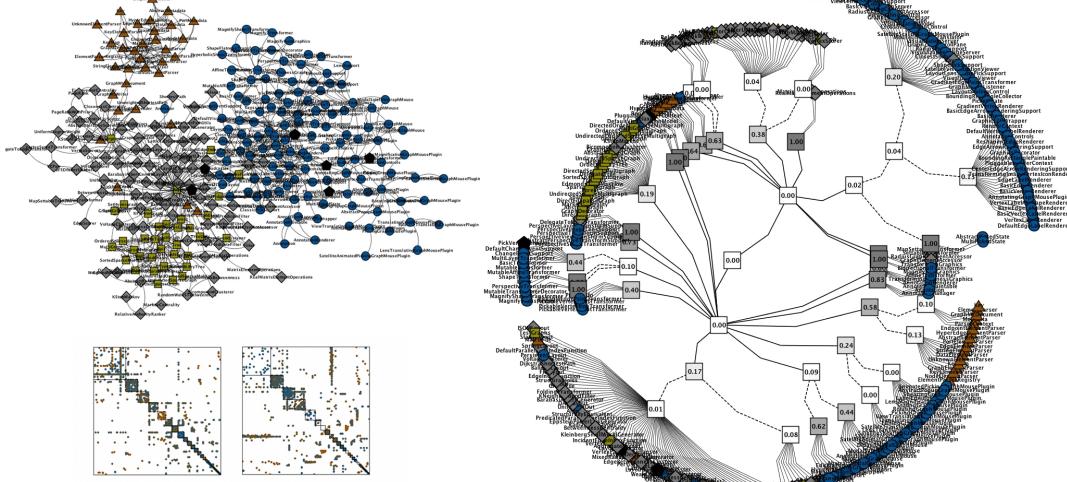


FIG. 11. (Color online) (left) Structural modules—blockmodels—of *jung* software network revealed with CP and HP algorithms (left-hand and right-hand side, respectively). (Symbols are consistent with Fig. 10.) (right) Hierarchy of *jung* software network revealed with HP algorithm. (Node symbols represent high-level packages of JUNG library [131]—*jung.visualization* (circles), *jung.io* (triangles), *jung.graph* (squares) and *jung.algorithms* (diamonds)—while hierarchy is else consistent with Fig. 9.)

the case for *elegans* network, where all nodes exhibit very high clustering. Note that both structures revealed with HP algorithm are still consistent with the structural-world conjecture, although different types of modules are better separated in the case *javax* network (Sec. II). This is further indicated by higher clustering assortativity r_d , which equals 0.545 for *javax* network, while only 0.183 for *elegans* network. Nevertheless, structure of *elegans* network might be modeled more adequately, when communities are extracted from the network apriori.

VI. CORA CITATION NETWORK

In the following we conduct a more detailed analysis of a larger information network using techniques presented in the paper. We adopt a citation network extracted from the famous *Cora* dataset [45] that includes computer science publications collected from the web, and publications automatically parsed from the bibliographies of the latter. Complete network reduced to the largest connected component contains 23166 nodes and 89157 links, while other common statistics are given in Table V.

| Description | n | m | k | C | r | r_d |
|----------------------|-------|-------|-----|-------|--------|-------|
| <i>Cora</i> network | 23166 | 89157 | 7.7 | 0.266 | -0.055 | 0.394 |
| Community extraction | 14602 | 29003 | 4.0 | 0.143 | -0.083 | 0.547 |
| Functional modules | 6832 | 10345 | 3.0 | 0.073 | -0.143 | 0.489 |

TABLE V. Common statistics for *Cora* citation network before and after community extraction, and network induced by functional modules identified in the latter. (Networks are treated as simple undirected graphs.)

We first employ SFF and LFF network models introduced in Sec. III to analyze citation dynamics depicted in the network. Table VI shows the networks generated by varying the burning and linking probabilities p and q within the models. Observe that, for $p \approx 0.3$ and $p \approx 0.75$, both models match common statistics of *Cora* network almost precisely. However, as already discussed in Sec. III, clustering assortativity r_d is underestimated.

| Model | p | q | n | m | k | C | r | r_d |
|-------|-------|-------|-------|--------------|------------|--------------|---------------|--------------|
| SFF | 0.275 | 0.75 | 23166 | 79886 | 6.9 | 0.263 | -0.060 | 0.213 |
| | 0.725 | | | 78724 | 6.8 | 0.270 | -0.056 | 0.217 |
| | 0.3 | 0.75 | | 87547 | 7.6 | 0.269 | -0.057 | 0.192 |
| | 0.775 | | | 97485 | 8.4 | 0.265 | -0.057 | 0.179 |
| LFF | 0.325 | 0.75 | 23166 | 96729 | 8.4 | 0.272 | -0.049 | 0.179 |
| | 0.25 | 0.775 | | 82057 | 7.1 | 0.265 | -0.059 | 0.277 |
| | 0.75 | | | 79581 | 6.9 | 0.275 | -0.051 | 0.285 |
| | 0.275 | 0.775 | | 88870 | 7.7 | 0.272 | -0.053 | 0.273 |
| | 0.8 | | | 100820 | 8.7 | 0.266 | -0.054 | 0.270 |
| | 0.3 | 0.775 | | 97222 | 8.4 | 0.277 | -0.049 | 0.274 |

TABLE VI. Statistics for networks containing 23166 nodes generated with SFF and LFF network models. (Results are estimates of the mean over 10 realizations of network models.)

According to Eq. (8), the number of visited nodes v by each newly added node within the models—the number of publications considered by the authors—can be estimated between 1.61 and 1.75 (simulations give similar results). As the average number of publications within each bibliography equals $k/2 = 3.84$, the fraction of publications considered by the authors, relative to the number

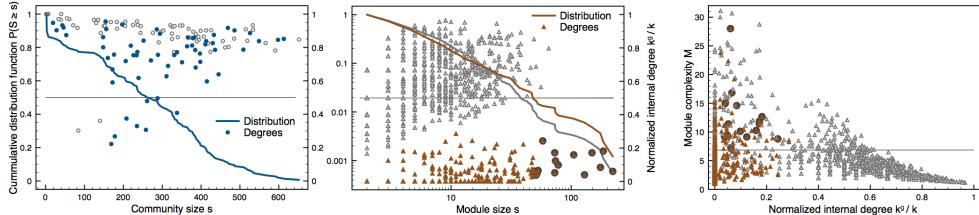


FIG. 12. (Color online) Communities and functional modules within *Cora* citation network (left-hand and right-hand side, respectively). (Filled symbols correspond to extracted communities and identified functional modules, where highlighted modules contain more than 48 nodes. Horizontal line in the right-most plot represents average M for functional modules.)

of publications cited, is $2v/k \approx 0.45$. Hence, according to citation dynamics behind *Cora* network, more than two times as many publications are cited than actually read.

Note that above results are somewhat influenced by (automatic) network sampling procedure (see [45] for details). The latter can be clearly observed in much lower average degree k than in other citation networks (see, e.g., *hepart* network in Table I). Furthermore, common density of real-world networks also estimates a much higher k [59, 108]. It ought to be mentioned that scale-free exponents α for degree distributions of the networks generated with SFF and LFF models are 2.78 and 2.67, whereas *Cora* network exhibits $\alpha = 3.3$. (Power-laws $p_k \propto k^{-\alpha}$ are plausible fits at $p\text{-value} = 0.01$ [107].)

We next also reveal different structural modules expressed in the structure of *Cora* network. Identification proceeds as follows. First, dense structural modules are identified in the network based on the extraction framework presented in Sec. II. The final pool consists of 146 modules, whereas 61 of these are extracted as communities according to a criteria based on a local maximum of r_d . Remaining network contains 14602 nodes, while other common statistics are given in Table V. Fig. 12 (left) further shows different properties of identified modules.

Second, remaining structural modules in the network are revealed using the proposed HP algorithm. The algorithm detects 1819 groups of nodes, where 827 of these are identified as functional modules according to Fig. 12 (middle). Common statistics for the network induced by functional modules are given in Table V.

As predicted by the structural-world conjecture, identified communities and functional modules overlap considerably. 43% of nodes in communities also appear in functional modules, whereas 41% of nodes in functional modules are overlaid by communities. Each node is else in 1.25 communities. Fig. 12 also shows distributions of module sizes s (note different scales). Observe that distribution for communities is rather uniform (Fig. 12 (left)), which is inconsistent with some earlier work [46, 47]. On the other hand, size distribution of functional modules shows a plausible fit to a power-law $p_s \propto s^{-\alpha}$ with $\alpha = 2.24$ at $p\text{-value} = 0.1$ [107] (Fig. 12 (middle)).

Fig. 12 (right) plots the complexity M for structural modules identified in *Cora* network after extraction of communities. M measures the complexity of linking be-

tween different structural modules. Let p_g be the probability that a neighbor of a node in some module is in module G and let G to be the corresponding random variable. Then, M of concerned module is defined as

$$M = b^{H(G)}, \quad (15)$$

where $H(G)$ is the entropy of G , $H(G) = -\sum_g p_g \log p_g$, and b is the base of the logarithm, $M \geq 1$. Note that Eq. (15) is equivalent to Shannon's coding theorem [134] (up to a constant). M of a module is thus the expected number of dependent modules in the network. Hence, M is close to one for communities and, e.g., functional modules that form bipartite structures, whereas higher values correspond to more complex configurations.

Values of M shown in Fig. 12 (right) reveal than complexity of linking patterns between different modules is much higher than expected. For example, average M for functional modules equals 6.85, however, the result is influenced by a large number of smaller modules identified in the network (many small communities remain). Indeed, when the structure is reduced to 31 modules that represent more than 48 nodes in the original network, and only inter-dependencies that are supported by more than 48 links are considered, average M decreases to 1.84. Nevertheless, functional modules still arrange in configurations that go beyond, e.g., simple bipartite structures.

Although not directly discussed above, general structural modules more accurately model *Cora* network after extraction than a framework limited to communities. For example, average height of a hierarchy revealed with HP algorithm equals 4.7, whereas 7.0 for CP community detection algorithm (Sec. V). Fig. 13 also shows corresponding module overlays. Observe that functional modules recognize artificial intelligence and operating systems as two rather independent fields of computer science, which are related through interdisciplinary fields like data structures, algorithms and programming (Fig. 13 (middle)). However, community structure of the network fails to acknowledge the latter (Fig. 13 (left)).

Last, Table VII also shows research topic distributions [45] and publications representing hub nodes for selected functional modules in Fig. 13. Similarly as in Sec. V, the structure can be related to different roles of publications within a certain field. For example, largest functional modules represent publications ad-

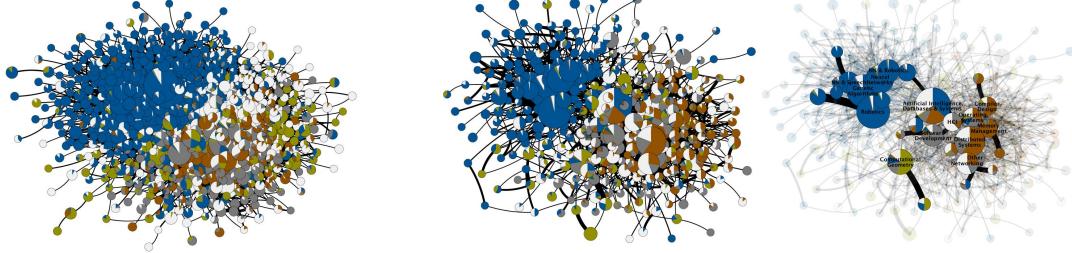


FIG. 13. (Color online) Largest connected components of community and functional module overlays upon *Cora* citation network after extraction (left-hand and right-hand side, respectively). Modules in the right-most overlay correspond to more than 48 nodes. (Node sizes are proportional to module sizes, whereas histograms represent high-level research topic classification in [45]—*artificial intelligence* (blue), *operating systems* (mocha), *data structures and algorithms* (asparagus), *programming* (gray) and other—and are consistent among figures. Due to simplicity, modules representing four nodes or less are ignored.)

| Modules | s | $k(n)$ | Reference / Topic |
|--------------------|------|---|--|
| Neural networks | 20 | 136 | Hertz et al., <i>Introduction to the theory of neural computation</i> (Addison-Wesley, 1991). |
| | (73) | | <i>Artificial intelligence – machine learning – neural networks.</i> |
| | 129 | (11) | <i>Artificial intelligence – machine learning – probabilistic methods.</i> |
| | (6) | | <i>Artificial intelligence – machine learning – genetic algorithms.</i> |
| Genetic algorithms | 37 | 182 | Goldberg, <i>Genetic algorithms in search, optimization and machine learning</i> (Addison-Wesley, 1989). |
| | (90) | | <i>Artificial intelligence – machine learning – genetic algorithms.</i> |
| | 175 | (15) | <i>Artificial intelligence – machine learning – neural networks.</i> |
| | (12) | | <i>Artificial intelligence – games and search.</i> |
| Graph drawing | 18 | Kant, <i>Drawing planar graphs using the lmc-ordering</i> , in <i>Proc. of SFCS '92</i> , pp. 101-110. | |
| | 9 | 18 Schnyder, <i>Embedding planar graphs on the grid</i> , in <i>Proc. of SODA '90</i> , pp. 138-148. | |
| | 7 | Kant et al., <i>Area requirement of visibility representations of trees</i> , in <i>Proc. of CCCG '93</i> , pp. 333-356. | |
| | 11 | Chrobak et al., <i>Convex drawings of graphs in two and three dim.</i> , in <i>Proc. of SCG '96</i> , pp. 319-328. | |
| | 23 | 9 Garg et al., <i>Planar upward tree drawings with optimal area</i> , <i>Int. J. Comput. Geom. Ap.</i> , 6 , 333 (1996). | |
| | 8 | Garg, <i>Where to draw the line</i> , PhD thesis, Brown University (1996). | |

TABLE VII. Research topic classification [45] and publications corresponding to hub nodes within selected functional modules in Fig. 13. Neural networks and genetic algorithms modules represent equally labeled bipartite structures, whereas graph drawing corresponds to lower-most pronounced bipartition (right-hand and left-hand side of Fig. 13 (right), respectively). Results show hub nodes within smaller modules and topic distribution of larger modules (below and above, respectively), while only high degree nodes are reported for graph drawing. Note that highest degrees in the larger neural networks and genetic algorithms modules are else only 19 and 26. Also, interestingly, Roberto Tamassia who was also a PhD advisor of Ashim Garg has coauthored all papers on graph drawing with multiple authors. (References have been abbreviated to fit page width.)

dressing similar problems that depend on a smaller set of prior seminal publications or, e.g., book reviews. The latter commonly correspond to hubs in the network. Nevertheless, many of the revealed configurations of functional modules include no hub nodes. (See caption of Fig. 13.)

VII. CONCLUSIONS

Findings in the paper expose functional modules as another key ingredient of complex real-world networks. These are together with communities combined into a structural-world conjecture, which provides a mesoscopic view on the structure of networks. We propose a natural

model based on the latter that generates networks with most common properties, whereas we also introduce a simple algorithm that outperforms state-of-the-art in detection of structural modules. We further propose several other techniques, valid for exploratory network analysis.

Future work will focus on the analysis of larger networks with millions of nodes and links, to devise a more detailed classification of structural-world networks.

ACKNOWLEDGMENTS

This work has been supported by Slovene Research Agency ARRS within Research Program No. P2-0359.

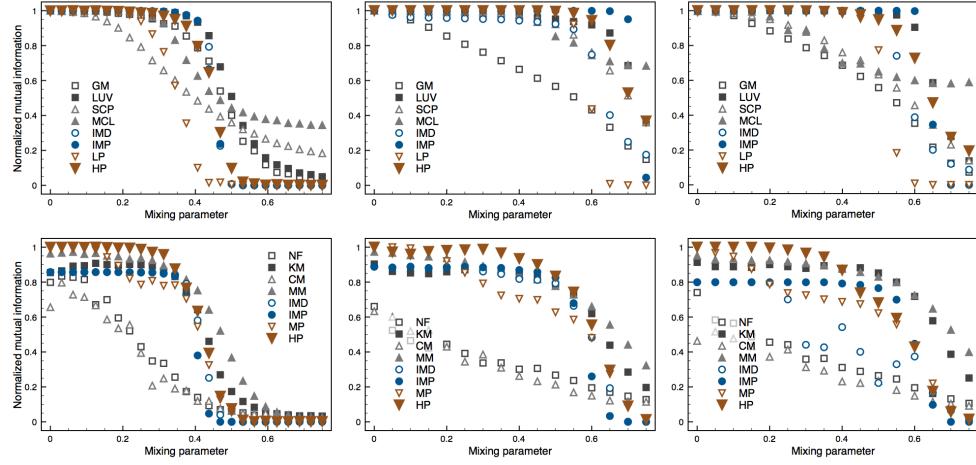


FIG. 14. (Color online) Comparison of module detection algorithms on GN and LFR synthetic networks with smaller and larger communities (top left to right, respectively), and on GN2, HN7 and HN6 synthetic networks (bottom left to right, respectively). (Results are estimates of the mean over 100 network realizations, and 10 realizations for MCL and CM algorithms.)

Appendix A: State-of-the-art

Proposed HP algorithm is compared against different state-of-the-art algorithms for community detection, and for detection of structural modules. We adopt the following community detection approaches: greedy optimization of modularity [46, 135] (GM algorithm), multi-stage optimization of modularity [126] (LUV algorithm), sequential clique percolation method [136] (SCP algorithm), Markov clustering algorithm [137] (MCL algorithm), structural compression method known as Infomod [81] (IMD algorithm), random walk based compression known as Infomap [129] (IMP algorithm), and a label propagation algorithm [44] (LP algorithm). (SCP algorithm returns overlapping communities; thus, each node in multiple communities is classified into a random one.)

The algorithms are analyzed on three synthetic benchmark networks with planted communities. First is a classical community detection benchmark [8], where the networks consist of four communities with 32 nodes (GN networks). We also consider two variations of more realistic synthetic networks with scale-free degree and community size distributions [138] (LFR networks). The number of nodes in the networks is set to 1000, while community sizes vary between 10 and 50, and 20 and 100 nodes. For all networks, the structure is controlled by a mixing parameter μ , $\mu \in [0, 1]$. When μ equals zero, all links are placed according to the predefined partition, whereas for μ equal to one, the networks are completely random.

The results are shown in Fig. 14 (top). HP algorithm is proven to be at least comparable to most of the approaches considered, yet it is outperformed by LUV and IMP algorithms. Note that the latter are among the best community detection algorithms in the literature, while their performance on these networks cannot be further improved [74, 82]. On the other hand, there is also no guaranty that the construction of networks does not im-

plicitly introduce functional modules (that are detected by HP algorithm). Nonetheless, both LUV and IMP algorithms are still limited to communities and cannot detect other structural modules, which we address next.

We adopt the following general module detection algorithms: symmetric nonnegative matrix factorization [139] (NF algorithm), k -means data clustering [140] based on [141] (KM algorithm), mixture model using expectation-maximization [31] (MM algorithm), mixture model with degree corrections [111] (CM algorithm), structural compression method [81] (IMD algorithm), model-based propagation algorithm [120] (MP algorithm), and the best community detection algorithm considered above [129] (IMP algorithm). (NF and CM algorithms are applied to each network for ten times, while the best revealed partition is reported. NF, KM, MM and CM algorithms demand the number of modules apriori.)

The algorithms are compared on GN2, HN7 and HN6 synthetic networks (Sec. V). All networks contain communities and functional modules, where the latter are connected into bipartite and tripartite structures. Again, the links in the networks are placed according to μ .

Fig. 14 (bottom) shows the results of the comparison. Only HP algorithm can accurately detect the modules planted into these networks, and MP algorithm for small enough μ , while most other approaches fail. MM mixture model also performs relatively well, whereas CM model that incorporates additional constraints is relatively unstable on these networks (due to increased complexity). Interestingly, simple KM data clustering performs much better than many other state-of-the-art algorithms [141].

The analysis does not include approaches that group nodes based on their network properties [28]. However, these also reveal groups that are not structural modules.

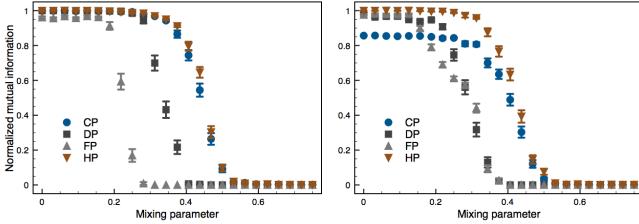


FIG. 15. (Color online) Comparison of network modeling on GN and GN2 synthetic networks (left and right, respectively). (Results are estimates of the mean over 100 network realizations, while bars show standard error of the mean.)

Appendix B: Network modeling

The section considers different network modeling techniques represented by module factors ν_g (Eq. (13)). We compare the strategy adopted with the HP algorithm with three alternatives, where ν_g equal to 0, 0.5 or 1 for all modules g (denoted CP, DP and FP algorithms). Hence, CP algorithm is based on classical propagation between

neighboring nodes that can detect merely communities. FP algorithm propagates labels through common neighbors, which reveals functional modules. DP algorithm is the default approach, where labels are simultaneously propagated between and through (common) neighbors.

The strategies are compared on GN and GN2 synthetic benchmark networks (App. A and Sec. V). GN networks consist of only communities, while GN2 networks contain communities and also functional modules. For both networks, the structure is controlled by a mixing parameter μ , $\mu \in [0, 1]$. For $\mu = 0$, all links are placed according to the planted modules, whereas the networks are completely random for $\mu = 1$.

The results are shown in Fig. 15. CP algorithm can accurately detect communities, while functional modules are identified as a single group of nodes. Interestingly, FP algorithm can reveal functional modules and also communities, however, the stability of the algorithm is rather challenged. DP algorithm performs reasonably well on both networks, although the results can be improved considerably (see HP algorithm).

-
- [1] G. Facchetti, G. Iacono, and C. Altafini, P. Natl. Acad. Sci. USA, **108**, 20953 (2011).
[2] L. Šubelj and M. Bajec, Physica A, **390**, 2968 (2011).
[3] S. Vitali, J. B. Glattfelder, and S. Battiston, PLoS ONE, **6**, e25995 (2011).
[4] R. J. Williams and N. D. Martinez, Nature, **404**, 180 (2000).
[5] D. J. Watts and S. H. Strogatz, Nature, **393**, 440 (1998).
[6] M. Faloutsos, P. Faloutsos, and C. Faloutsos, Comput. Commun. Rev., **29**, 251 (1999).
[7] A. L. Barabási and R. Albert, Science, **286**, 509 (1999).
[8] M. Girvan and M. E. J. Newman, P. Natl. Acad. Sci. USA, **99**, 7821 (2002).
[9] M. E. J. Newman, Phys. Rev. Lett., **89**, 208701 (2002).
[10] M. E. J. Newman, Phys. Today, **61**, 33 (2008).
[11] M. E. J. Newman, S. H. Strogatz, and D. J. Watts, Phys. Rev. E, **64**, 026118 (2001).
[12] M. E. J. Newman, Phys. Rev. E, **67**, 026126 (2003).
[13] G. W. Flake, S. Lawrence, and C. L. Giles, in *Proc. of KDD '00* (Boston, MA, USA, 2000) pp. 150–160.
[14] M. Newman, *Networks: An introduction* (Oxford University Press, Oxford, 2010).
[15] M. E. J. Newman, SIAM Rev., **45**, 167 (2003).
[16] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi, P. Natl. Acad. Sci. USA, **101**, 2658 (2004).
[17] M. S. Granovetter, Am. J. Sociol., **78**, 1360 (1973).
[18] M. McPherson, L. Smith-Lovin, and J. M. Cook, Annu. Rev. Sociol., **27**, 415 (2001).
[19] M. E. J. Newman and M. Girvan, Phys. Rev. E, **67**, 026126 (2003).
[20] M. E. J. Newman and J. Park, Phys. Rev. E, **68**, 036122 (2003).
[21] D. V. Foster, J. G. Foster, P. Grassberger, and M. Paczuski, Phys. Rev. E, **84**, 066117 (2011).
[22] J. Leskovec, J. Kleinberg, and C. Faloutsos, ACM Trans. Knowl. Discov. Data, **1**, 1 (2007).
[23] Z. Wu and P. Holme, Phys. Rev. E, **80**, 037101 (2009).
[24] F. Ren, X. Cheng, and H. Shen, e-print arXiv:11044209v1 (2011).
[25] D. Hao and C. Li, PLoS ONE, **6**, e28322 (2011).
[26] A. Lancichinetti, M. Kivela, J. Saramaki, and S. Fortunato, PLoS ONE, **5**, e11976 (2010).
[27] A. Lancichinetti, F. Radicchi, and J. Ramasco, Phys. Rev. E, **81** (2010).
[28] T. Nishikawa and A. E. Motter, Sci. Rep., **1**, 151 (2011).
[29] Y. Zhao, E. Levina, and J. Zhu, P. Natl. Acad. Sci. USA, **108**, 7321 (2011).
[30] L. Šubelj and M. Bajec, Eur. Phys. J. B, **85**, 32 (2012).
[31] M. E. J. Newman and E. A. Leicht, P. Natl. Acad. Sci. USA, **104**, 9564 (2007).
[32] J. Reichardt and D. R. White, Eur. Phys. J. B, **60**, 217 (2007).
[33] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing, J. Mach. Learn. Res., **9**, 1981 (2008).
[34] M. G. Everett and S. P. Borgatti, J. Math. Sociol., **19**, 29 (1994).
[35] F. Lorrain and H. C. White, J. Math. Sociol., **1**, 49 (1971).
[36] S. Pinkert, J. Schultz, and J. Reichardt, PLoS Comput. Biol., **6**, e1000659 (2010).
[37] Some of the earlier work refers to functional and structural modules differently [30, 32].
[38] S. N. Soffer and A. Vázquez, Phys. Rev. E, **71**, 057101 (2005).
[39] J. M. Kleinberg, R. Kumar, P. Raghavan, S. Rajagopalan, and A. S. Tomkins, in *Proc. of COCOON '99* (Tokyo, Japan, 1999) pp. 1–17.
[40] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and E. Upfal, in *Proc. of FOCS '00* (Washington, DC, USA, 2000) p. 57–65.

- [41] P. Holme and B. J. Kim, Phys. Rev. E, **65**, 026107 (2002).
- [42] J. M. Kumpula, J. Onnela, J. Saramaki, K. Kaski, and J. Kertész, Phys. Rev. Lett., **99**, 228701 (2007).
- [43] J. Leskovec, J. Kleinberg, and C. Faloutsos, in *Proc. of ACM KDD '05* (Chicago, IL, USA, 2005) pp. 177–187.
- [44] U. N. Raghavan, R. Albert, and S. Kumara, Phys. Rev. E, **76**, 036106 (2007).
- [45] A. K. McCallum, K. Nigam, J. Rennie, and K. Seymore, Inf. Retr., **3**, 127–163 (2000).
- [46] A. Clauset, M. E. J. Newman, and C. Moore, Phys. Rev. E, **70**, 066111 (2004).
- [47] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, Nature, **435**, 814 (2005).
- [48] Definition of r is consistent with directed networks, while for more practical expression see [9]. Note that Pearson coefficient detects only linear correlations. Although one can study neighbor connectivity plots [94], correlation profiles [96] or other measures [142] instead, r is often preferable due to its convenience.
- [49] J. J. Han, N. Bertin, T. Hao, D. S. Goldberg, G. F. Berriz, L. V. Zhang, D. Dupuy, A. J. M. Walhout, M. E. Cusick, F. P. Roth, and M. Vidal, Nature, **430**, 88 (2004).
- [50] P. Erdős and A. Rényi, Publ. Math. Debrecen, **6**, 290 (1959).
- [51] M. Molloy and B. Reed, Random Structures & Algorithms, **6**, 161 (1995).
- [52] H. Ebel, L. Mielsch, and S. Bornholdt, Phys. Rev. E, **66**, 035103 (2002).
- [53] p_c is derived for alternative, but similar, definition of C that allows easier analytical consideration [11, 20].
- [54] E. Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai, and A. L. Barabási, Science, **297**, 1551 (2002).
- [55] A. Vázquez, R. Pastor-Satorras, and A. Vespignani, Phys. Rev. E, **65**, 066130 (2002).
- [56] E. Ravasz and A. L. Barabási, Phys. Rev. E, **67**, 026112 (2003).
- [57] M. E. J. Newman, Phys. Rev. E, **74**, 036104 (2006).
- [58] M. E. J. Newman, P. Natl. Acad. Sci. USA, **98**, 404 (2001).
- [59] N. Blagus, L. Šubelj, and M. Bajec, Physica A, **391**, 2794 (2012).
- [60] M. Boguná, R. Pastor-Satorras, A. Diaz-Guilera, and A. Arenas, Phys. Rev. E, **70**, 056122 (2004).
- [61] P. Gleiser and L. Danon, Adv. Complex Syst., **6**, 565 (2003).
- [62] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, Behav. Ecol. Sociobiol., **54**, 396 (2003).
- [63] W. W. Zachary, J. Anthropol. Res., **33**, 452 (1977).
- [64] R. Guimerà, L. Danon, A. Diaz-Guilera, F. Giralt, and A. Arenas, Phys. Rev. E, **68**, 065103 (2003).
- [65] L. Šubelj and M. Bajec, Eur. Phys. J. B, **81**, 353 (2011).
- [66] “KDD-Cup,” <http://www.sigkdd.org/kddcup/> (2003).
- [67] L. Šubelj and M. Bajec, in *Proc. of ACNE '10* (Barcelona, Spain, 2010) pp. 87–104.
- [68] H. Jeong, S. P. Mason, A. Barabási, and Z. N. Oltvai, Nature, **411**, 41 (2001).
- [69] L. A. Adamic and N. Glance, in *Proc. of LinkKDD '05* (Chicago, IL, USA, 2005) pp. 36–43.
- [70] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A. Barabási, Nature, **407**, 651 (2000).
- [71] A. Davis, B. B. Gardner, and M. R. Gardner, *Deep South* (Chicago University Press, Chicago, IL, 1941).
- [72] S. N. Dorogovtsev, A. V. Goltsev, and J. F. F. Mendes, Phys. Rev. E, **65**, 066122 (2002).
- [73] r_d of a random graph [50] does not go to zero in the limit of large n (results omitted). However, since $D \approx 0$, r_d captures only small random fluctuations in d .
- [74] S. Fortunato, Phys. Rep., **486**, 75 (2010).
- [75] S. E. Schaeffer, Comput. Sc. Rev., **1**, 27 (2007).
- [76] M. A. Porter, J. Onnela, and P. J. Mucha, Not. Am. Math. Soc., **56**, 1082 (2009).
- [77] No closed formula for W in a random graph exists [29]. Thus, we maximize W in 100 realizations of a corresponding random graph [50] and take the 95-th percentile as the expected value. Hence, the results are statistically significant at p -value = 0.05.
- [78] Y. Wei and C. Cheng, in *Proc. of ICCAD '89* (New York, NY, USA, 1989) pp. 298–301.
- [79] A. Clauset, C. Moore, and M. E. J. Newman, Nature, **453**, 98 (2008).
- [80] A. Clauset, C. Moore, and M. E. J. Newman, in *Proc. of SNA '06* (Pittsburgh, PA, USA, 2006) pp. 1–13.
- [81] M. Rosvall and C. T. Bergstrom, P. Natl. Acad. Sci. USA, **104**, 7327 (2007).
- [82] A. Lancichinetti and S. Fortunato, Phys. Rev. E, **80**, 056117 (2009).
- [83] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network flows: Theory, algorithms, and applications* (Prentice-Hall, Upper Saddle River, NJ, 1993).
- [84] X. Cheng, F. Ren, H. Shen, Z. Zhang, and T. Zhou, J. Stat. Mech., **P10011** (2010).
- [85] P. A. Grabowicz, J. J. Ramasco, E. Moro, J. M. Pujol, and V. M. Eguiluz, PLoS ONE, **7**, e29358 (2012).
- [86] P. J. Mucha, T. Richardson, K. Macon, M. A. Porter, and J. Onnela, Science, **328**, 876 (2010).
- [87] C. Gu, S. Zou, X. Xu, Y. Qu, Y. Jiang, D. R. He, H. Liu, and T. Zhou, Phys. Rev. E, **84**, 026101 (2011).
- [88] J. Gao, S. V. Buldyrev, H. E. Stanley, and S. Havlin, Nat. Phys., **8**, 40 (2012).
- [89] D. Lavbič, O. Vasilecas, and R. Rupnik, Technol. Econ. Dev. Eco., **16**, 327 (2010).
- [90] J. Park and M. E. J. Newman, Phys. Rev. E, **68**, 026112 (2003).
- [91] S. Maslov, K. Sneppen, and A. Zaliznyak, Physica A, **333**, 529 (2004).
- [92] S. Johnson, J. J. Torres, J. Marro, and M. A. Muñoz, Phys. Rev. Lett., **104**, 108702 (2010).
- [93] E. Estrada, Phys. Rev. E, **84**, 047101 (2011).
- [94] R. Pastor-Satorras, A. Vázquez, and A. Vespignani, Phys. Rev. Lett., **87**, 258701 (2001).
- [95] J. G. Foster, D. V. Foster, P. Grassberger, and M. Paczuski, P. Natl. Acad. Sci. USA, **107**, 10815 (2010).
- [96] S. Maslov and K. Sneppen, Science, **296**, 910 (2002).
- [97] Z. Jing, T. Lin, Y. Hong, L. Jian-Hua, C. Zhi-Wei, and L. Yi-Xue, Chinese Phys., **16**, 3571 (2007).
- [98] G. Bounova and O. de Weck, Phys. Rev. E, **85**, 016117 (2012).
- [99] T. P. Peixoto, e-print arXiv:11084341v1 (2011).
- [100] T. P. Peixoto, J. Stat. Mech., **P01006** (2012), doi: 10.1088/1742-5468/2012/01/P01006.
- [101] M. Schläpfer and L. Buzna, Phys. Rev. E, **85**, 015101(R) (2012).
- [102] F. Peruani and L. Tabourier, PLoS ONE, **6**, e28860 (2011).

- [103] J. Park and A. Barabási, *P. Natl. Acad. Sci. USA*, **104**, 17916 (2007).
- [104] Results reported under FF model correspond to the original model for directed networks [22, 43].
- [105] In practice, x_p and x_q are sampled from negative binomial distributions $\text{NB}(1, p)$ and $\text{NB}(1, q)$.
- [106] A. Vázquez, *Phys. Rev. E*, **67**, 056104 (2003).
- [107] A. Clauset, C. R. Shalizi, and M. E. J. Newman, *SIAM Rev.*, **51**, 661 (2009).
- [108] P. J. Laurienti, K. E. Joyce, Q. K. Telesford, J. H. Burdette, and S. Hayasaka, *Physica A*, **390**, 3608 (2011).
- [109] Multiple p and q will give a network with same C and r . Still, for fixed k , there is a unique solution (Eq. (9)).
- [110] H. C. White, S. A. Boorman, and R. L. Breiger, *Am. J. Sociol.*, **81**, 730 (1976).
- [111] B. Karrer and M. E. J. Newman, *Phys. Rev. E*, **83**, 016107 (2011).
- [112] X. Li and G. Chen, *Physica A*, **328**, 274 (2003).
- [113] Z. Zhang, L. Rong, B. Wang, S. Zhou, and J. Guan, *Physica A*, **380**, 639 (2007).
- [114] M. McGlohon, L. Akoglu, and C. Faloutsos, in *Proc. of KDD '08* (New York, NY, USA, 2008) p. 524–532.
- [115] To prevent oscillations of labels, i retains its current label when it is among most frequent in Γ_i [44].
- [116] G. Tibély and J. Kertész, *Physica A*, **387**, 4982 (2008).
- [117] L. Šubelj and M. Bajec, *Phys. Rev. E*, **83**, 036103 (2011).
- [118] I. X. Y. Leung, P. Hui, P. Liò, and J. Crowcroft, *Phys. Rev. E*, **79**, 066107 (2009).
- [119] Stability parameter β is set to 2, and to 0.25 for larger real-world networks [65] (to ensure convergence).
- [120] L. Šubelj and M. Bajec, in *Proc. of NEMO '11* (Athens, Greece, 2011) pp. 66–84.
- [121] B. Bollobás, *Modern graph theory* (Springer, Heidelberg, 1998).
- [122] G. Casella and R. L. Berger, *Statistical inference* (Duxbury Press, Belmont, CA, 1990).
- [123] T. P. Peixoto, e-print arXiv:11126028v1 (2011).
- [124] L. Danon, A. Díaz-Guilera, J. Duch, and A. Arenas, *J. Stat. Mech.*, **P09008** (2005).
- [125] M. Meila, *J. Multivariate Anal.*, **98**, 873 (2007).
- [126] V. D. Blondel, J. Guillaume, R. Lambiotte, and E. Lefebvre, *J. Stat. Mech.*, **P10008** (2008).
- [127] M. E. J. Newman and M. Girvan, *Phys. Rev. E*, **69**, 026113 (2004).
- [128] A. P. Dempster, N. M. Laird, and D. B. Rubin, *J. R. Statist. Soc.*, **39**, 1 (1977).
- [129] M. Rosvall and C. T. Bergstrom, *P. Natl. Acad. Sci. USA*, **105**, 1118 (2008).
- [130] S. Fortunato and M. Barthélémy, *P. Natl. Acad. Sci. USA*, **104**, 36 (2007).
- [131] J. O'Madadhain, D. Fisher, S. White, P. Smyth, and Y. Boey, *J. Stat. Softw.*, **10**, 1 (2005).
- [132] Y. Hu, H. Chen, P. Zhang, M. Li, Z. Di, and Y. Fan, *Phys. Rev. E*, **78**, 026121 (2008).
- [133] A. D. Medus and C. O. Dorso, *Phys. Rev. E*, **79**, 066111 (2009).
- [134] C. E. Shannon, *Bell. Syst. Tech. J.*, **27**, 379 (1948).
- [135] M. E. J. Newman, *Eur. Phys. J. B*, **38**, 321 (2004).
- [136] J. M. Kumpula, M. Kivelä, K. Kaski, and J. Saramäki, *Phys. Rev. E*, **78**, 026109 (2008).
- [137] S. van Dongen, *Graph clustering by flow simulation*, PhD thesis, University of Utrecht (2000).
- [138] A. Lancichinetti, S. Fortunato, and F. Radicchi, *Phys. Rev. E*, **78**, 046110 (2008).
- [139] C. Ding, T. Li, W. Peng, and H. Park, in *Proc. of KDD '06* (Philadelphia, PA, USA, 2006) pp. 126–135.
- [140] J. B. MacQueen, in *Proc. of BSMSP '67* (Berkeley, CA, USA, 1967) pp. 281–297.
- [141] C. Lin, J. Koh, and A. L. P. Chen, in *Proc. of PAKDD '10* (Hyderabad, India, 2010) pp. 56–67.
- [142] L. Li, D. Alderson, J. C. Doyle, and W. Willinger, *Internet Math.*, **2**, 431 (2005).