Uncovering the overlapping community structure of complex networks in nature and society

Supplementary Information

1 The k-clique-community finding algorithm

Our community definition is based on the observation that a typical member in a community is linked to many other members, but not necessarily to all other nodes in the community. In other words, a community can be interpreted as a union of smaller complete (fully connected) subgraphs that share nodes. In the mathematical literature, such complete subgraphs are called *k-cliques*, where k refers to the number of nodes in the subgraph. Therefore, we define a *k-clique-community* as the union of all k-cliques that can be reached from each other through a series of *adjacent k-cliques*, where two k-cliques are said to be adjacent if they share $k-1$ nodes. Using k-clique adjacency we can define a *k-clique chain* as the union of a sequence of adjacent k-cliques, and introduce the concept of *k-clique connectedness*: two k-cliques are k-clique-connected if they are parts of a k-clique chain. Our k-clique-communities are equivalent to the k-clique connected components of the network.

An illustration of these communities can be given by "k-clique template rolling". A k-clique template can be thought of as an object that is isomorphic to a complete graph of k nodes. Such a template can be placed onto any k-clique of the network, and rolled to an adjacent k-clique by relocating one of its nodes and keeping its other $k-1$ nodes fixed. Thus, the k-clique-communities of a graph are all those subgraphs that can be fully explored by rolling a k-clique template in them but cannot be left by this template.

The k-clique-communities of a network at $k=2$ are equivalent to the connected components, since a 2-clique is simply an edge and a 2-clique-community is the union of those edges that can be reached from each other through a series of shared nodes. Similarly, a 3-clique-community is given by the union of triangles that can be reached from one an other through a series of shared edges. As we increase k, the k-clique-communities shrink, but on the other hand become more cohesive since their member nodes have to be part of at least one k-clique.

Our experience shows that in real networks complete subgraphs of size between 10 and 100 can easily occur. Such a large complete subgraph of size s contains $\binom{s}{k}$ different k-cliques, therefore, an algorithm that tries to locate the k-cliques individually and examine the adjacency between them would be extremely slow when analysing real networks. However, a complete subgraph of size s is obviously a k-clique connected subset for any $k \leq s$, since for any pair of included smaller k-cliques, a series of adjacent k-cliques linking them can be trivially found. Furthermore, two large complete subgraphs that share at least $k-1$ nodes form one k-clique connected component as well. This implies that instead of searching for k-cliques, it is a far better strategy to locate the large complete subgraphs in the network first, and then look for the k-clique connected subsets of given k (the k-clique-communities) by studying the overlap between them.

1.1 The method

1.1.1 From cliques to k-clique-communities

To be more precise, our algorithm first extracts all complete subgraphs of the network that are not parts of larger complete subgraphs. (The details of this procedure are discussed in Sect. 1.1.2.) These maximal complete subgraphs are simply called *cliques*, and the difference between k-cliques and cliques is that k-cliques can be subsets of larger complete subgraphs. Once the cliques are located, the clique-clique overlap matrix is prepared [1]. In this symmetric matrix each row (and column) represents a clique and the matrix elements are equal to the number of common nodes between the corresponding two cliques, and the diagonal entries are equal to the size of the clique. (Note that the intersection of two cliques is always a complete subgraph.) The k-clique-communities for a given value of k are equivalent to such connected clique components in which the neighbouring cliques are linked to each other by at least $k-1$ common nodes. These components can be found by erasing every off-diagonal entry smaller than $k-1$

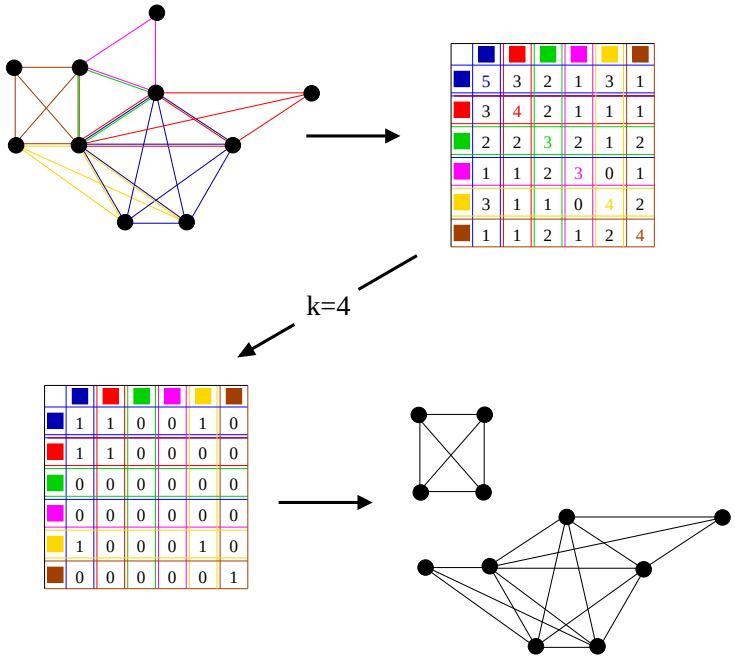

Figure 1: A simple illustration of the extraction of the k-clique-communities at $k = 4$ using the clique-clique overlap matrix. Top left picture shows the graph in which the different cliques are marked by different colours. The according clique-clique overlap matrix is shown in the top right corner. To obtain the k-clique-communities at $k = 4$, we delete the off-diagonal elements that are smaller than 3 and also the diagonal elements that are smaller than 4, resulting in the matrix shown in the bottom left of the figure. The connected components (the k-clique-communities) corresponding to this matrix are shown in the bottom right.

and every diagonal element smaller than k in the matrix, replacing the remaining elements by one, and then carrying out a component analysis of this matrix. The resulting separate components are equivalent to the different k-clique-communities. A simple illustration of the above is given in Fig. 1.

Another advantage of this method is that the clique-clique overlap matrix encodes all information necessary to obtain the communities for any value of k, therefore once the clique-clique overlap matrix is constructed, the k-clique-communities for all possible values of k can be obtained very quickly. In contrast to this, in a simple k-clique finding approach the search for the k-cliques would have to be restarted from the beginning for every single value of k.

1.1.2 Locating the cliques

As discussed in the previous section, in contrast to the k-cliques, cliques cannot be subsets of larger cliques, therefore they have to be located in a decreasing order of their size. The largest possible clique size in the studied graph is determined from the degree-sequence. Starting with this clique size, our algorithm repeatedly chooses a node, extracts every clique of this size containing that node, then deletes the node and its edges. (The deletion of the already examined nodes inhibits the finding of the same clique multiple times). When no nodes are left, the clique size is decreased by one and the clique finding procedure is restarted on the original graph. The already found cliques influence the further search since the yet unrevealed (smaller) cliques cannot be subsets of them.

The cliques of size s containing a given node v can be found by examining the interrelations of the neighbours of v. In our algorithm this is implemented in the following way: First, a set $\mathcal{A}$ is constructed

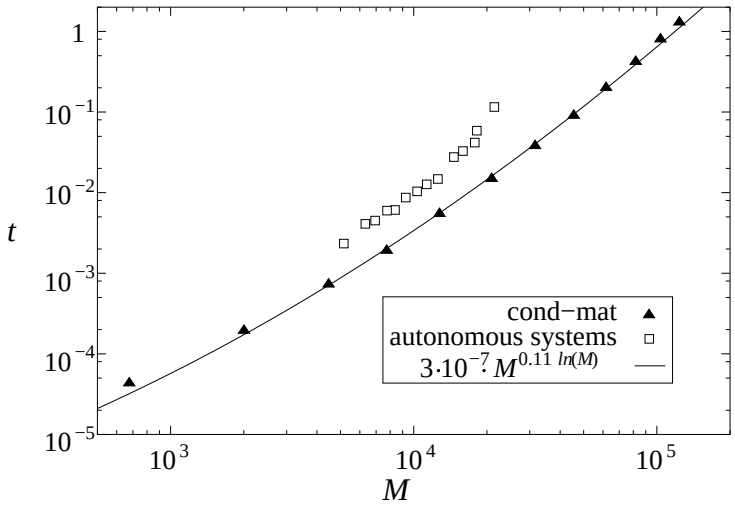

Figure 2: The time in hours on a PC needed to locate the communities as a function of the system size in the number of edges for the cond-mat archive (triangles) and for the graph of autonomous systems (squares). The former dataset is fitted with $3 \cdot 10^{-7} M^{0.11 \ln(M)}$ (solid curve).

that contains nodes all linked to each other. Initially $\mathcal{A}$ consists of v only and our goal is to enlarge this set to the actual clique-size s. Another disjunct set $\mathcal{B}$ is also determined as the set of nodes that are linked to each node in $\mathcal{A}$, but not necessarily to the nodes in $\mathcal{B}$. Initially set $\mathcal{B}$ consists of the neighbours of v.

Set $\mathcal{A}$ can be enlarged transferring nodes from $\mathcal{B}$. This is accomplished in a recursive way in order to check every possible combination of the nodes being transfered. (To avoid finding the same clique multiple times, the nodes have to be transferred from $\mathcal{B}$ to $\mathcal{A}$ in a decreasing/increasing order of their indices.) When a node w from $\mathcal{B}$ is placed into $\mathcal{A}$, the nodes that are not neighbours of w are removed from $\mathcal{B}$. (This is done in order to preserve the property that the members of $\mathcal{B}$ are all linked to each member of $\mathcal{A}$).

If $\mathcal{B}$ runs out of nodes before $\mathcal{A}$ reaches size s, or if the union of the sets $\mathcal{A}$ and $\mathcal{B}$ can be included in an already found (larger) clique, the recursion is stepped back to check other possibilities. Whenever the size of $\mathcal{A}$ reaches s, a new clique is found. After recording the clique, the algorithm is stepped back again to check the remaining possible combinations of the neighbours indices.

1.2 Efficiency of the algorithm

The determination of the full set of cliques of a graph is widely believed to be non-polynomial problem. In spite of this, our algorithm proves to be very efficient when applied to the graphs of the investigated real systems. Our experience shows that the required CPU time depends on the structure of the input data very strongly, therefore in general no closed formula can be given even to estimate the system size dependence. As an illustration of the computational speed, however, we note that a complete analysis of a co-authorship network with 127000 links takes less than 2 hours on a PC.

In Fig. 2 we display the time it took to explore the community structure (using a PC) as a function of the system size in case of the co-authorship network of the Los Alamos Condensed Matter e-print archive [2, 3] at the optimal threshold for $k = 6$ and the network of autonomous systems [4]. (In both cases the graphs of different size correspond to the state of the system at different times). As it can be seen in the figure, the curves can be fitted with $t = AM^{B \ln(M)}$ where t denotes the time needed by our algorithm, M stands for the number of edges, and A and B are fitting parameters.

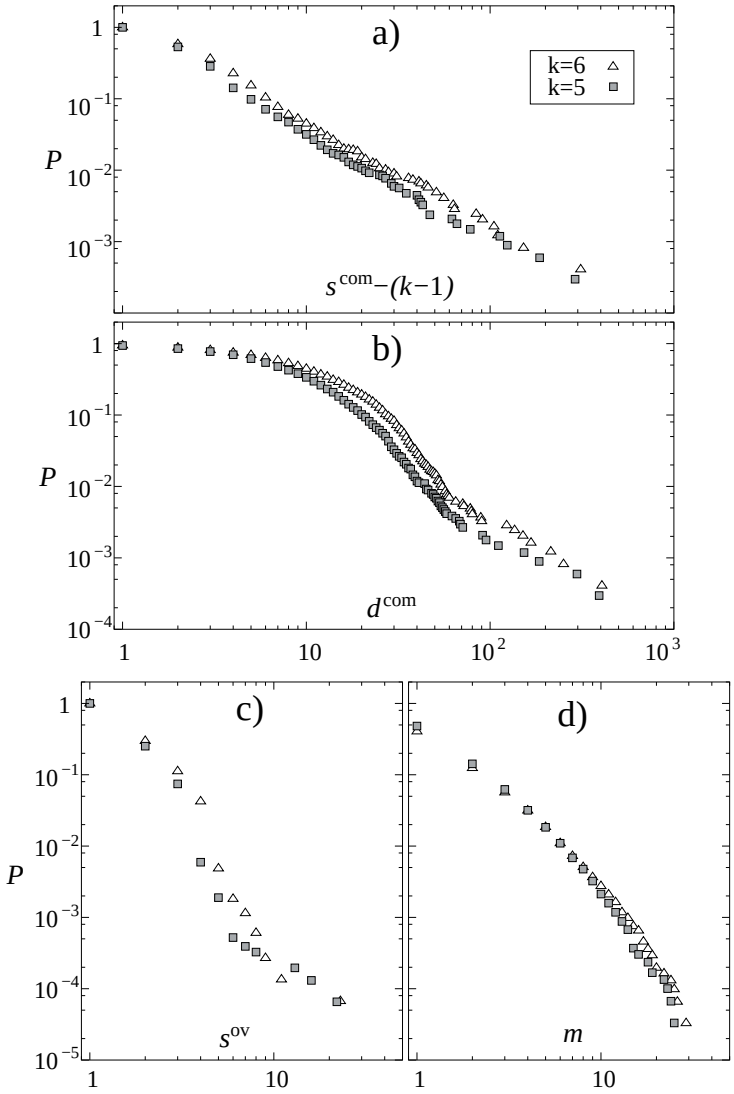

Figure 3: Statistics of the k-clique-communities for the Los Alamos Condensed Matter e-print archive at $k = 5$ (squares) and $k = 6$ (triangles). (a) the cumulative distribution function of the k-clique-community size (b) the cumulative distribution function of the k-clique-community degree (the degree distribution of the graph of communities), (c) the cumulative distribution function of the overlap size, and (d) the cumulative distribution function of the membership number of nodes.

2 Community statistics at different values of k

Our method can be directly applied to binary (undirected, unweighted) networks only. Therefore, when analysing an arbitrary system, the directionality of the links has to be ignored and if the connections are weighted, a threshold weight w^* can be introduced to prune weak links and keep those that are stronger than w^*. (If we want to keep all links, w^* is simply set to zero). If the threshold weight is increased, the number of edges is decreased and the communities shrink, however they consist of stronger links on average. Similarly, if k is increased at fixed threshold weight, the communities become smaller and more disintegrated, but at the same time also more cohesive (since every member in a community has to be part of a larger complete subgraph).

The criterion we used to fix the optimal k and w^* values is based on finding a community structure

as highly structured as possible. Usually a lower threshold weight is accompanied by a larger number of communities as more edges are left in the network. However, at a certain critical point a giant community appears which smears out the details of the community structure. Thus, for each selected value of k we adjusted the weight threshold to the point where the largest community becomes twice as big as the second largest one (just below the critical point). The restriction for the value of k we used was that at least half of the links should remain for the optimal threshold.

In case of the network representing the Los Alamos Condensed Matter e-print archive the criterions for the global k and w^* values could be matched at both $k = 5$ and $k = 6$. (In the former case the fraction f^* of the connections being kept during the application of our method was equal to $f^* = 0.75$, whereas in the latter case it turned out to be $f^* = 0.93$). In Fig. 3 we compare the relevant distributions characterising the community structure for the two values of k. In Fig. 3a the two scaling cumulative community size distributions are almost on top of each other. In case of the community degree (Fig. 3b) the scaling tails of the distribution functions are parallel similarly to the previous case. However the two distributions differ slightly at their exponential part, namely the characteristic community degree is a bit higher for $k = 6$ than for $k = 5$. There is a small difference between the two overlap size distributions as well at the middle part of the distributions (Fig. 3c). Finally, the two membership number distributions displayed in Fig. 3d match each other very well.

It can be seen from the distributions at $m = 1$ that the fraction of nodes belonging to at least one community is somewhere between 25% and 50%. The majority of the rest of the nodes fall out simply because their degree is less than $k - 1$. Nevertheless, after identifying the communities, most of these weakly connected nodes can be associated with the communities to which they are most strongly connected.

Besides this very good agreement between the relevant statistical distributions, the communities themselves show great similarities in the two cases: 44 % of the 6-clique-communities are present amongst the 5-clique-communities, and for 70 % of the 6-clique-communities one can find a corresponding 5-clique-community that differs in less than 10 % of the members. The good agreement between the results obtained for different values of k signals that the fundamental properties of the observed community structure are characteristic to the system itself and are largely independent of k.

3 Further examples

In this section we present a few more examples from the results of our community finding method. These concern both the global statistical properties of the communities determined for two additional data sets, (the Hungarian synonyms and the variables of the source code of the ftp program under Linux), as well as the local community structure around further vertices in the word association graph and in the network of the ftp program.

3.1 Community statistics

Similarly to Fig. 4 in the manuscript, the four major distributions characterising the global community structure of two further systems are plotted in Fig. 4. The triangles correspond to the network of the wu-ftp program under Linux [5] and the squares refer to the Hungarian synonym graph obtained from the OpenOffice word processor [6]. In the former network the nodes correspond to variables in the source code and are assumed to be connected if they appear together in an expression or function call, whereas in the second network two words are linked if they are synonyms of each other. The number of nodes N and links M are given by $N = 1886, 20139$ and $M = 6001, 100427$ for the network of the ftp program and the synonyms respectively. In both cases, our criterions for the global choice of the k-clique size can be matched only at $k = 5$.

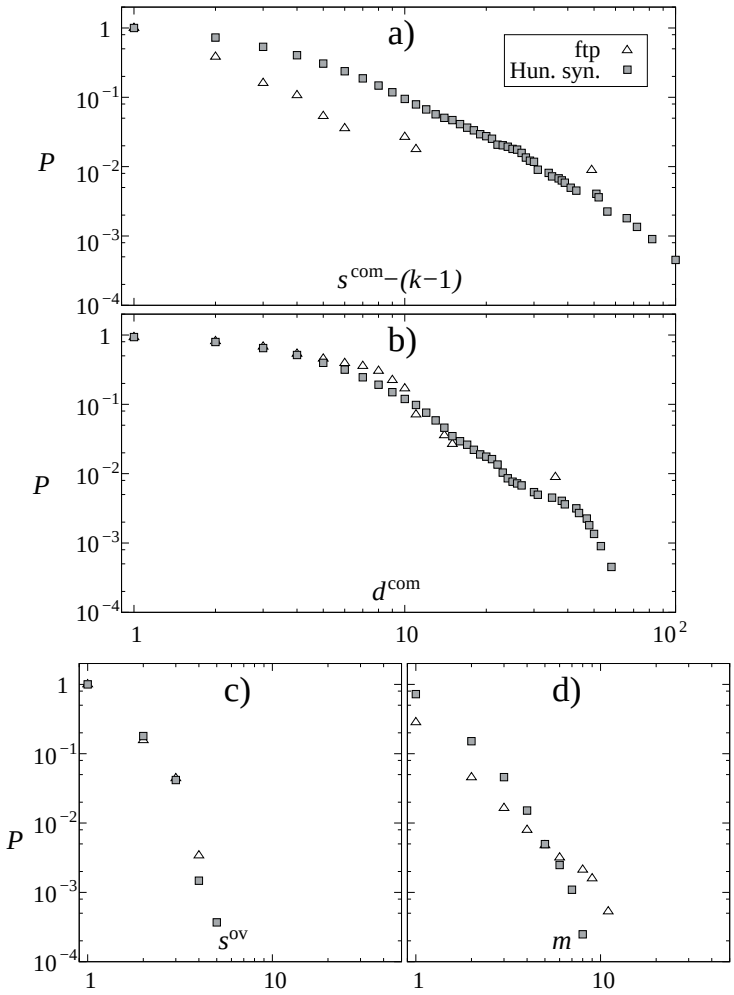

Figure 4: Statistics of the k-clique-communities for the wu-ftp program under Linux (triangles, k=4) and the graph of the Hungarian synonyms obtained from the OpenOffice word processor (squares, $k = 4$). (a) The cumulative distribution of the community size, (b) the cumulative distribution of the community degree, plot (c) is the cumulative distribution of the overlap size and (d) is that of the membership number.

Although our results for the two new data sets resemble those obtained for the data in the manuscript, there are also some deviations. In Fig. 4a the tails of the community size distributions are power-law like (however, not over such a wide range as, *i.e.*, in case of the co-authorship network). The lower part of the community degree distributions is exponential (Fig. 4b), but the extra power-law like tail present in case of the co-authorship network and the word association network is much less pronounced here. Due to the relatively small system size there is only one outstanding community degree in case of the ftp program, whereas the tail of the community degree distribution of the synonyms is somewhat like staircase. The community overlap distributions (Fig. 4c) are rather truncated, the maximal overlap size reaches just the k-clique size for the synonyms and is equal to $k - 1$ for the ftp program. In Fig. 4d, the membership number distributions decay somewhat faster than in case of the co-authorship network or the word association network.

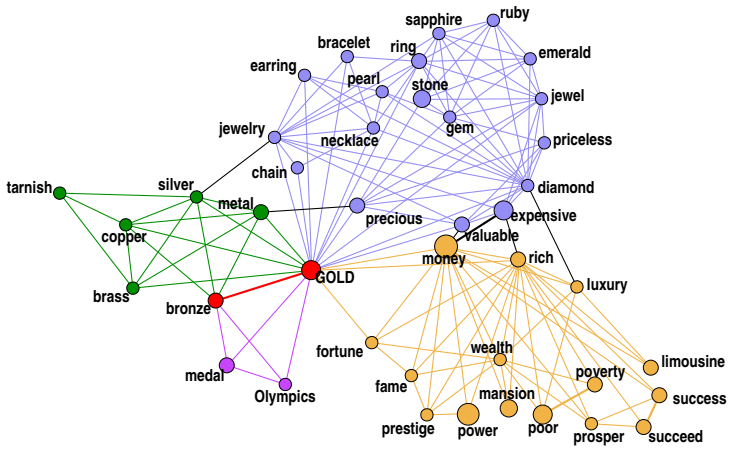

Figure 5: The k-clique communities of the word gold in the South Florida Free Association norm list for $w^* = 0.025$ and $k = 4$. The purple community is related to Olympic medals, the green one consists of metals, the blue one can be associated to jewels and finally the yellow community is related to welfare.

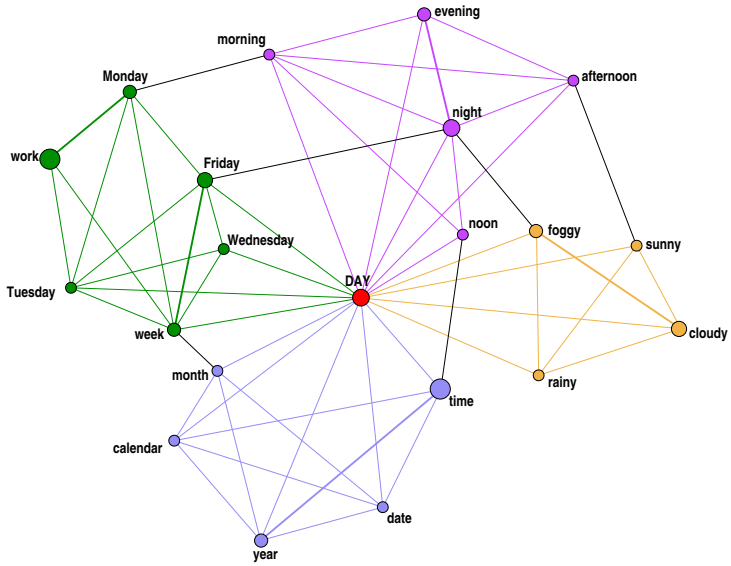

Figure 6: The k-clique communities of the word day in the South Florida Free Association norm list at $w^* = 0.025$ and $k = 4$. The green community is related to work-days, the purple one consists of day times, the yellow one can be associated to weather, whereas the blue community is related to the calendar.

3.2 Local community structure

In Figs. 5-7 the k-clique communities of three words picked from the South Florida Free Association norm list [7] are shown in a similar fashion to Fig. 2 in the manuscript. The communities are colour coded, the overlapping nodes and links between them are emphasised in red, and the size of the nodes and the width of the links are proportional to their membership numbers (the total number of communities they belong to). The threshold weight w^* and the k-clique size in these examples are set to $w^* = 0.025$ and $k = 4$.

In Fig. 5 the four communities of the word *gold* are related to Olympic medals, metals, jewels and

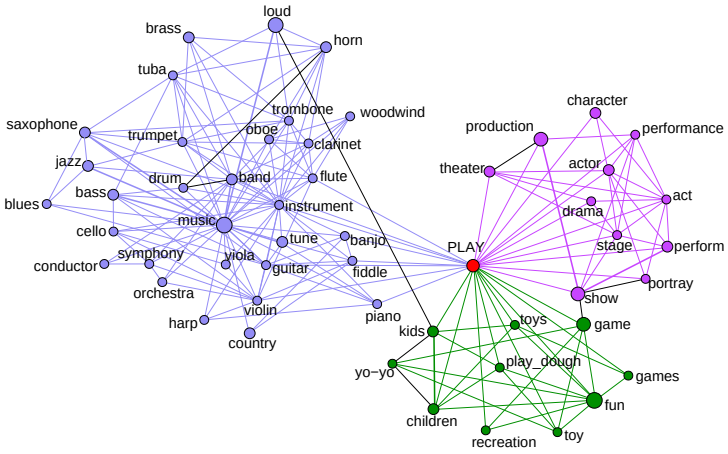

Figure 7: The k-clique communities of the word play in the South Florida Free Association norm list for $w^* = 0.025$ and $k = 4$. The blue community is associated with music, the purple one is related to theatre and the green community can be associated with children.

welfare respectively. In Fig. 6 the communities of the word *day* are shown. The green community can be associated with work days. *Thursday* has only two neighbours (*Wednesday* and *Wed*) even in the original (unpruned) network, therefore it is missing from this community, whereas *Saturday* and *Sunday* are in another community related to weekend containing *Friday*, *night*, *week* and *weekend* itself as well. The purple community of Fig. 6 consists of day times, the yellow community contains common adjectives of *day* related to weather, and the blue community can be associated with the calendar. Fig. 7 displays the three communities of the word *play*: the blue one is related to music, the purple one to theatre and the green one can be associated with children.

In Fig. 8 of we show a component from the community graph of the wu-ftp program at $k = 5$ in a fashion similar to Fig. 3 in the manuscript. The name of each node consists of two parts: the first one is specific to the variable represented by the node and the second part (separated by '@') is specific to the scope of the variable (typically a function). The names ending in '@glb' denote global variables. Since these variables have global scope, (and therefore are visible in the entire program), they may appear in several function calls and expressions throughout the entire source code. Thus, in the corresponding network the vertices representing these variables are candidates for community overlaps. Indeed, in Fig. 8, the majority of the communities are related to functions in the source code, and several community overlaps are provided by vertices representing global variables.

4 Random community statistics

The non-trivial aspects of the distributions presented in Fig. 4 of the manuscript naturally give rise to the question whether the community statistics of a random graph would significantly differ from those studied in the manuscript. In other words, what happens with the community structures if the links of the networks studied in the manuscript are reshuffled in a random way?

We calculated the major statistical distributions for two types of random graphs corresponding to the three systems studied in the manuscript. In the first case, *the degree sequences of the original graphs were preserved* during the randomisation process. We implemented this by link randomisation [8]: in each step two links were selected randomly, and then one of the endpoints of the links were swapped. This process was repeated until on average about a dozen relocations per link was reached. The other type of random graphs we tested were simple Erdős-Rényi random graphs [9] with the same number of

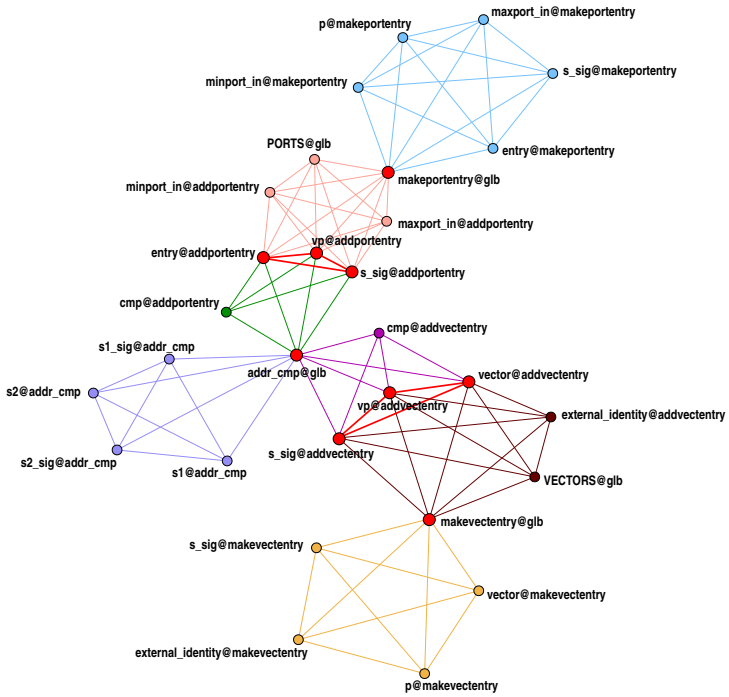

Figure 8: A component in the community graph of the wu-ftp program. Most of these communities are related to functions (sub routines) in the source code. The nodes with a name ending in '@glb' represent global variables. These are likely to appear in several function calls in the source code, hence they are likely to be members in several communities at the same time.

nodes and links as the co-authorship network at $f^* = 0.93$, the word association graph at $f^* = 0.67$ or the protein interaction graph. (The degree sequences in these cases are different from the original ones).

We have found that except for the link-randomised word association graph, *cliques of size larger than three were totally absent in the random networks, therefore, naturally, no k-clique communities for $k > 3$ can exist at all in them*. In comparison the largest clique sizes are $12, 8, 9$ and $k = 6, 4, 4$ in the original co-authorship network, word association network and protein interaction network respectively. In Fig. 9. we show the four major statistical distributions for the link-randomised word association network (triangles) compared to the original system (squares, the same as in Fig. 4 in the manuscript). In the randomised system the maximal community size is five (Fig. 9a), the maximal community degree is two (Fig. 9b), the maximal overlap size is one (Fig. 9c), and the maximal membership number is two (Fig. 9d), therefore the corresponding distributions are very truncated compared to the original ones.

In conclusion, we can say that *randomisation severely (in some cases entirely) destroys the observed community structure*. The fact that randomisation can lead to complete loss of communities also implies that they are present in the original system entirely due to specific correlations.

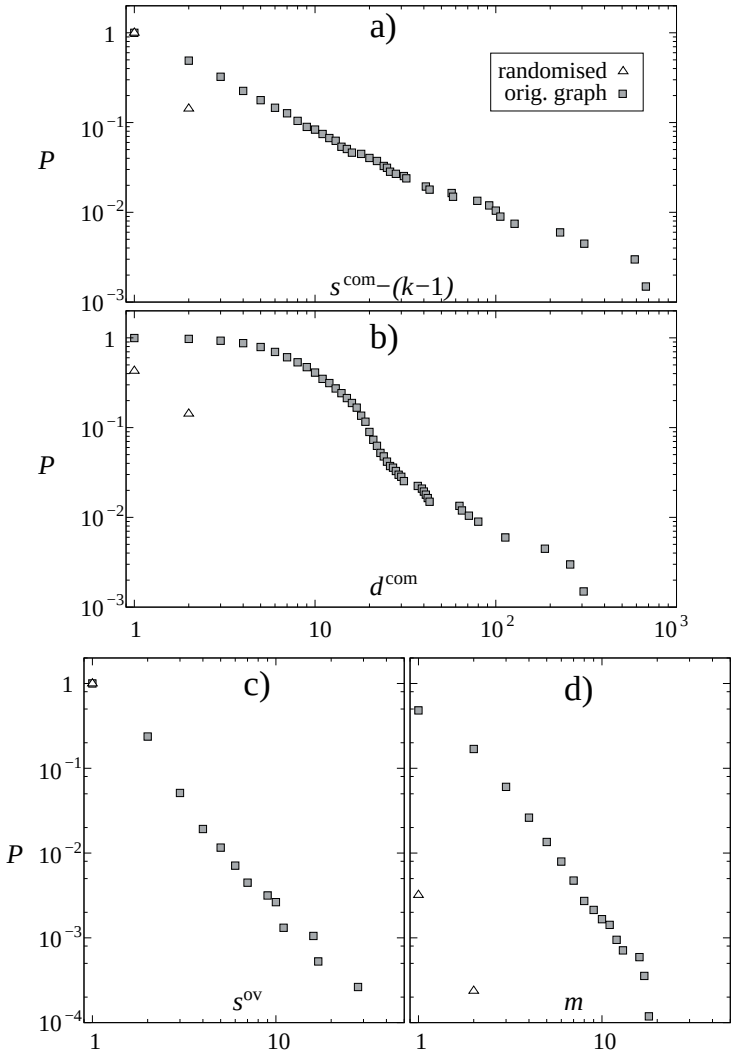

Figure 9: Statistics of the 4-clique-communities for the link-randomised word association network of the South Florida Free Association norm list at $f^* = 0.67$ (triangles), plotted together with the distributions of the original system (squares). The degree sequence was preserved during the randomisation process. (a) The cumulative distribution of the community size, (b) the cumulative distribution of the community degree, plot (c) is the cumulative distribution of the overlap size and (d) is that of the membership number.

References

[1] M. G. Everett and S. P. Borgatti, Analyzing clique overlap. *Connections* **21**, 49–61 (1998).

[2] S. Warner, E-prints and the Open Archives Initiative. *Library Hi Tech* **21**, 151–158 (2003).

[3] `http://arxiv.org/` The co-authorship data were kindly provided by Simeon Warner.

[4] The data concerning the time evolution of the network of autonomous systems was downloaded from `http://www.cosin.org/extra/data/internet/nlanr.html` .

[5] http://www.wu-ftpd.org

[6] http://www.openoffice.org/

[7] Nelson, D. L., McEvoy, C. L., & Schreiber, T. A. *The University of South Florida word association, rhyme, and word fragment norms.* `http://www.usf.edu/FreeAssociation/`.

[8] S. Maslov and K. Sneppen: Science **296**, 910 (2002)

[9] P. Erdős and A. Rényi, Publ. of the Math. Inst. of the Hung. Acad. of Sci. **5**, 17-61 (1960).