SCIENTIFIC REPORTS

OPEN

SUBJECT AREAS:
CELL BIOLOGY
COMPUTER SCIENCE
COMPUTATIONAL SCIENCE

Received
17 February 2014

Accepted
13 March 2014

Published
1 April 2014

Correspondence and requests for materials should be addressed to N.P. (n.przulj@imperial.ac.uk)

Revealing the Hidden Language of Complex Networks

Ömer Nebil Yaveroğlu[1], Noël Malod-Dognin[1], Darren Davis[2], Zoran Levnajic[1,6], Vuk Janjic[1], Rasa Karapandza[3], Aleksandar Stojmirovic[4,5] & Nataša Pržulj[1]

[1]Department of Computing, Imperial College London, UK, [2]Computer Science Department, University of California, Irvine, USA, [3]Department of Finance, Accounting & Real Estate EBS Business School, Germany, [4]National Center for Biotechnology Information (NCBI), USA, [5]Janssen Research and Development, LLC, Spring House, PA, USA, [6]Faculty of Information Studies in Novo mesto, Novo Mesto, Slovenia.

Sophisticated methods for analysing complex networks promise to be of great benefit to almost all scientific disciplines, yet they elude us. In this work, we make fundamental methodological advances to rectify this. We discover that the interaction between a small number of roles, played by nodes in a network, can characterize a network's structure and also provide a clear real-world interpretation. Given this insight, we develop a framework for analysing and comparing networks, which outperforms all existing ones. We demonstrate its strength by uncovering novel relationships between seemingly unrelated networks, such as Facebook, metabolic, and protein structure networks. We also use it to track the dynamics of the world trade network, showing that a country's role of a broker between non-trading countries indicates economic prosperity, whereas peripheral roles are associated with poverty. This result, though intuitive, has escaped all existing frameworks. Finally, our approach translates network topology into everyday language, bringing network analysis closer to domain scientists.

Detecting and interpreting the patterns of change in complex networks may yield insight into their underlying function, emergent properties, and controllability[1,2]. However, this is a challenging task, since a complete comparison between complex networks has long been known to be computationally intractable[3]. Hence, simple heuristics, commonly called network *properties* or network *statistics*, such as the degree distribution, have been used to approximately say whether the structure of networks is similar[4]. The most sophisticated statistics are based on graph spectra[5,6] and small subnetworks including network motifs[7] and graphlets[8]. However, none of the current methods are sufficient for characterizing the structure and extracting information hidden in the topology of complex networks.

Real-world networks often have few types of nodes with well defined topological characteristics, also called *roles*. For example, the set of driver nodes that can control and move the networks into specific states has been identified and shown to be of low degree[1]. Also, world trade networks are proposed to have a core-periphery structure, with some countries (nodes in the network) being at the dense core, forming rich-clubs of trading countries, while others are at the sparsely connected periphery[9]. Such node roles are differently correlated in different types of networks[9]. Hence, we seek to design a method that will reveal and exploit these phenomena.

We cannot utilize graph spectra to design such a method, since spectra do not provide a direct real-world interpretation of network structure[5]. While *network motifs* and their spectra[7,10] can be used to define node roles, their interpretation is highly dependent on the choice of a network null model, which limits their usability[11]. This is because network motifs are defined as small partial subgraphs that are overrepresented in the real network compared to a chosen network null model; a partial subgraph means that once you pick a set of nodes in the large network, you can pick any subset of edges between them. *Graphlets* do not suffer from these drawbacks, can be used to define node roles and to design methods for linking the network structure with real-world function[12]. They are defined as small induced subgraphs of a large network that appear at any frequency and hence are independent of a null model (denoted by G_0 to G_{29} in Fig. 1 d); an induced subgraph means that once you pick the nodes in the large network, you must pick all the edges between them to form the subgraph. We define and utilize the correlations between graphlets (detailed below) to create a superior network measure that, unlike other simple or complex measures, makes network structure directly interpretable and provides its clear translation into everyday language. As such, our new measure can uncover novel relationships between seemingly unrelated networks from different domains. Furthermore, it can be used to track the dynamics and explain the evolution of any network, which we demonstrate on the world trade network example (Fig. 1 a-c). Our methodology is

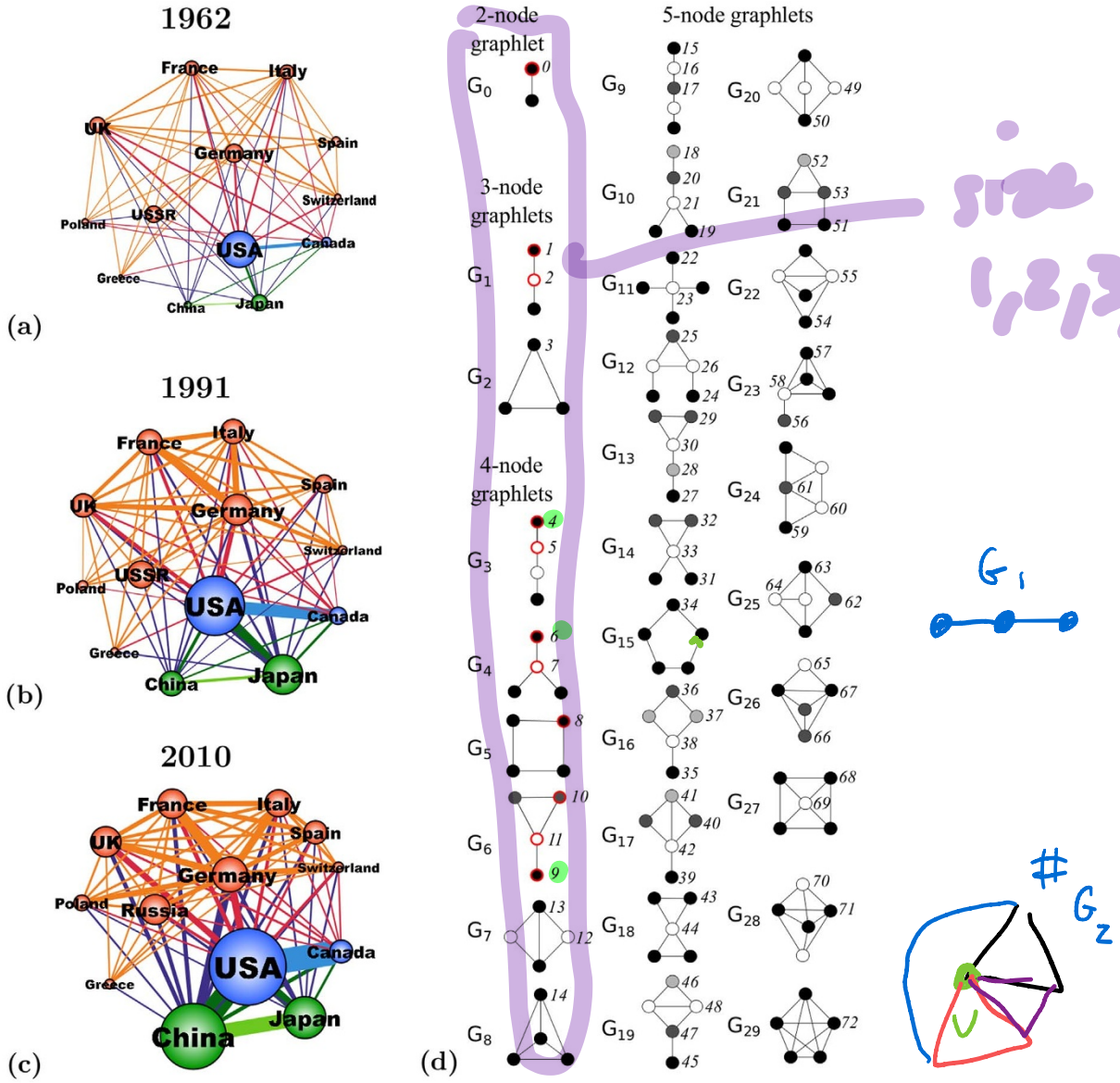
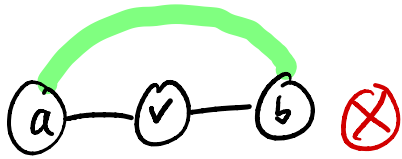


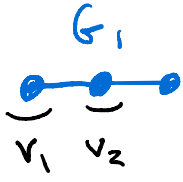
Figure 1 | Illustrations of subnetworks of the world trade networks. (a) 1962, (b) 1991, and (c) 2010. Node colors correspond to the continents: orange for Europe, green for Asia, blue for America. The node size corresponds to the GDP of the country. The edge thickness corresponds to the volume of the trade between the countries. (d) The thirty 2- to 5-node graphlets G_0, G_1, G_2, …, G_{29}. In each graphlet, nodes belonging to the same automorphism orbit are of the same shade. The 73 automorphism orbits of the 30 graphlets are labelled from 0 to 72[8]. Some orbits are redundant (their counts in a network can be derived from the counts of other orbits); the 11 red orbits illustrate the non-redundant ones for up to 4-node graphlets – there are several ways to choose non-redundant orbits, but that choice does not impact further analysis.

universal and can provide insight in all areas of science that use network theory, including biology, medicine, social sciences, and security.

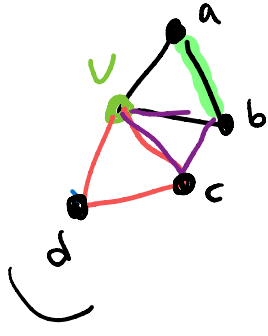
New network statistic: Graphlet Correlation Distance. The distributions of graphlet frequencies in networks have been compared in the network statistic called Relative Graphlet Frequency Distribution[13]. To increase sensitivity at the same computational cost, *symmetry groups* of nodes within graphlets, called *automorphism orbits* [For a node x of network G, the automorphism orbit of x is the set of nodes of G that can be mapped to x by an automorphism, an isomorphism of a network with itself; i.e., a bijection of nodes that preserves node adjacency. Automorphism orbits of graphlets are illustrated as 0 to 72 in Fig. 1 d.], have been used to generalize the degree distribution into the spectrum of 73 Graphlet Degree Distributions that correspond to the 73 orbits for up to 5-node graphlets: the first of these distributions is the familiar degree distribution, the second gives the number of nodes in the network that touch k orbits 1 of graphlet G_1 for all values of k, etc. for all 73 orbits. Then, Graphlet Degree Distributions of two networks are compared over all orbits in the network statistic called Graphlet Degree Distribution Agreement[8]. A related concept is that of the Graphlet Degree Vector of a node that has been used to link wiring around a node with its real-world function[12]: it has 73 coordinates, each of which measures the number of times the node is touched by a particular orbit of a graphlet (so the first coordinate is the degree of the node, the second is the number of 3-node paths that it touches at an end node etc.).

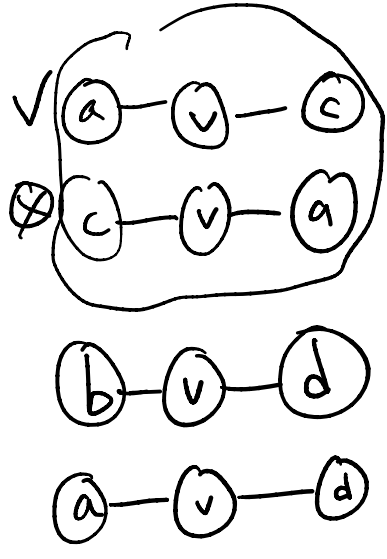
We design a superior graphlet-based measure by identifying and eliminating redundancies and exploiting dependencies between orbit counts in a network. For example, if we denote by C_i the i^{th} *graphlet degree* of a node (where $i \in \{0, 1, \ldots, 72\}$, Fig. 1 d), which is the number of times the node is touched by orbit i[8], then if we consider



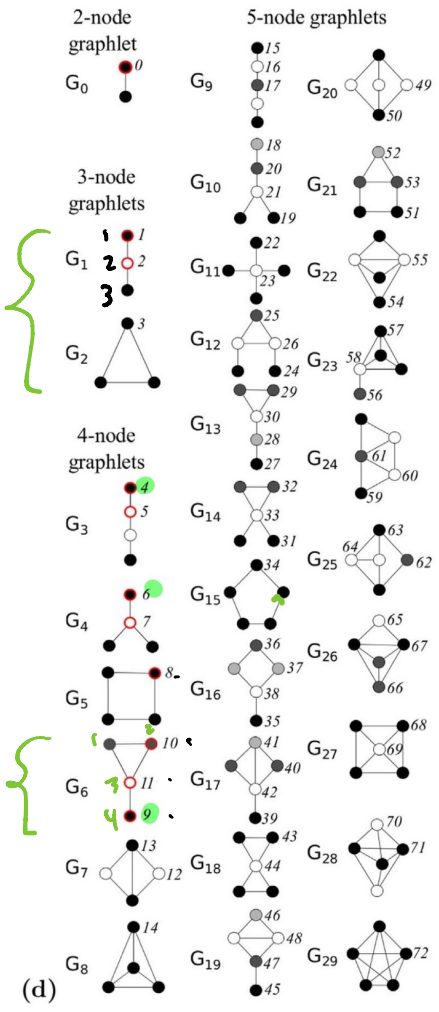


C_0 = # times it acts like v_0 in graphlet (ignoring isomorphs)





v can only act as v_2



(1,2)
(1,3)
(2,3)

"same?"

G_2

f

f(2)
f(1) f(3)

Dom Rg.

what f preserve G?

for relabeling of vertices

~orbit

3 auto. group

for G_1 =

$f(G_2) = G_2$

$f(G_1) \neq G_1$ ⊗

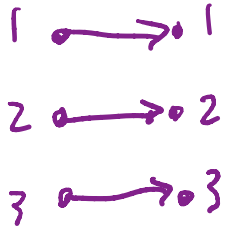
=

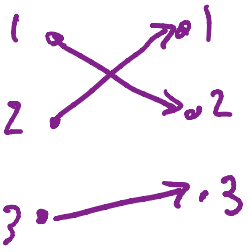
what f make

$f(G_1) = G_1$

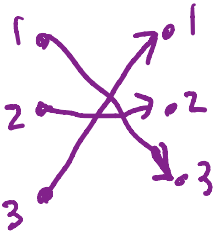
✓ automorphisms

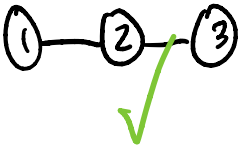
Same?

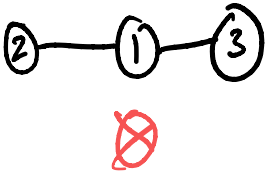


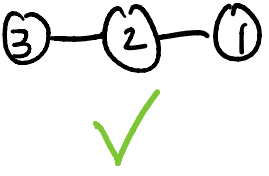


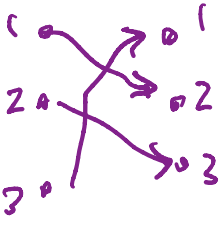
Different?

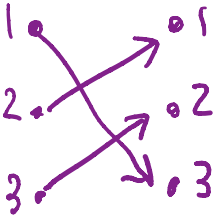


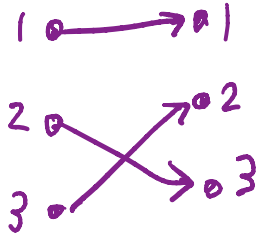


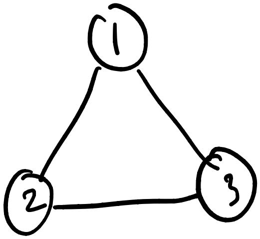








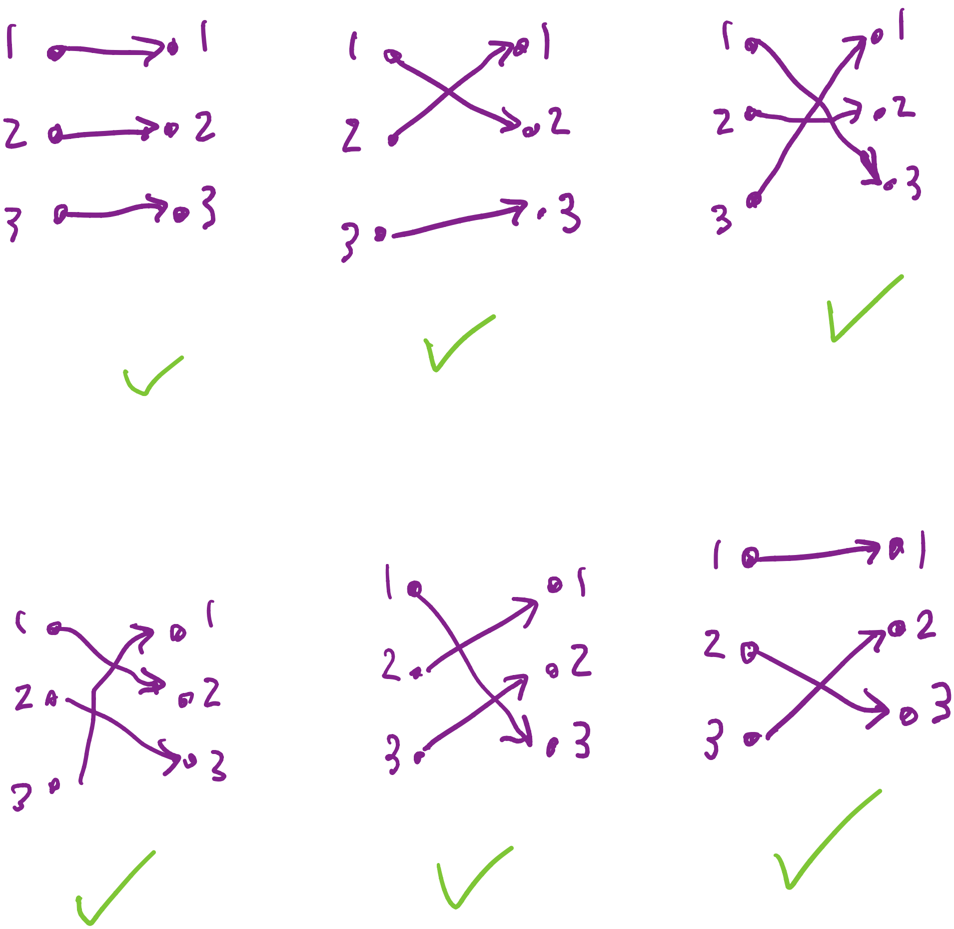




√ automorphisms

Same?

different?



the degree of the node, C_0, we can argue as follows. The neighbours of the node are either connected, or they are not: if they are connected, then they contribute to counts of triangles, C_3, that the node touches; if they are not connected, then they contribute to C_2 for the node, the number of times the node is touched by the middle of a 3-node path (orbit 2 in graphlet G_1, Fig. 1 d). Since these are the only options for connectedness of neighbours of a node, the number of ways in which C_0 neighbours of the node can be connected, $\binom{C_0}{2}$, is equal to the sum of C_2 and C_3 for the node: $\binom{C_0}{2} = C_2 + C_3$. Hence, if we know two of C_0, C_2 and C_3, we can derive the third, so one of them is redundant and does not need to be included in graphlet-based statistics. Similarly, we obtain a system of 17 linear equations describing all orbit redundancies (see Supplementary Information). When we solve it for the 73 orbits, 56 orbits remain non-redundant. There are 15 orbits for up to 4-node graphlets and 11 are non-redundant (red ones in Fig. 1 d). Similar redundancies, but in orbits of partial 4-node subgraphs have been reported[14].

We identify and exploit the dependencies (correlations) between graphlets as follows. First, we note that there are fewer dependencies between the 11 non-redundant orbits for up to 4-node graphlets than between the 56 non-redundant orbits for up to 5-node graphlets (also see below). Hence, they introduce less noise in the corresponding new network statistic, so we construct a network statistic using the 11 non-redundant orbits for up to 4-node graphlets. However, we contrast it with analogous statistics that include redundant and up to 5-node graphlet orbits as well.

Then, we devise a network statistic based on correlations between various node properties contained in non-redundant orbit counts, over all nodes, as follows. For each node in a network, first we construct its Graphlet Degree Vector consisting of 11 coordinates corresponding to the 11 non-redundant orbits. Then we construct a matrix whose rows are the above described Graphlet Degree Vectors, so the number of rows in the matrix is equal to the number of nodes in the network and it has 11 columns. The existence of correlations between non-redundant orbits over all nodes is exploited for constructing a new network statistic: for a given network N_1, we compute Spearman's Correlation coefficients between all pairs of columns of the above described matrix and present them in an 11×11 symmetric matrix that we term the *Graphlet Correlation Matrix (GCM)* of network N_1, GCM_{N_1}. In this way, we can summarize the topology of a network of any size into an 11×11 symmetric matrix with values in the interval $[-1, 1]$ (illustrated in Supplementary Fig. S2).

Different real and model networks generally have very different orbit dependencies and hence very different GCMs (Fig. 2 a–d). For example, in agreement with known properties of scale-free Barabási-Albert (SF-BA) networks[15], orbits 0, 2, 5, and 7, which are characteristic to existence of hubs, form a cluster of dependent orbits (as illustrated by their correlation coefficients being close to 1 in Fig. 2 a); also, orbits 10 and 11, which are characteristic to existence of clustering near hubs, form a cluster of dependent (i.e., correlated) orbits; and finally, orbits 1, 4, 6, and 9, characteristic to existence of a large number of degree 1 nodes, are dependent as well. The picture is quite different for geometric random graphs (GEO)[16] of the same size, which have Poisson degree distributions and hence the structure not dominated by a large fraction of degree 1 nodes and a small number of hubs (Fig. 2 b) (see Supplementary Information).

Uncovering orbit dependencies in real-world networks is much more interesting, since they can reveal currently unknown organisational principles of these networks. Indeed, the world trade network of 2010[17] (explained in Supplementary Information) contains two large clusters of dependent orbits, {0, 2, 5, 7, 8, 10, 11} and {4, 6, 9}, while there is no correlation between orbits {4, 6, 9} and orbits {0, 2, 5, 7, 8, 10, 11} (Fig. 2 c). We ask what this means and notice that orbits 4, 6 and 9 correspond to *peripheral*, degree 1 nodes that are "hanging" from graphlets G_3, G_4 and G_6 (Fig. 1 d), while members of the large cluster of correlated orbits, {0, 2, 5, 7, 8, 10, 11}, correspond to higher degree, either clustered (in a densely linked neighbourhood), or *broker*-type (*mediators* between nodes that are not directly linked) orbits. Since these two clusters are not correlated, we can conclude that countries are either clustered/brokers, or on the periphery of world trade, but not both. Hence, GCM unveils a hidden structure of this network that can be further interpreted qualitatively: through further analysis presented below, we interpret this observation on 49 world trade networks corresponding to trade data from 1962 to 2010. In contrast, the topology of the human metabolic network (see Supplementary Information) is very different from the topology of world trade networks: the correlations between all orbits are high, indicating that constituent bio-molecules can be at the same time both peripheral and clustered/broker (Fig. 2 d).

In addition to in-depth examination of network topology that can be qualitatively interpreted, the demonstrated differences in GCMs enable us to define a new measure of distance between topologies of two networks. For networks N_1 and N_2, we define their network *distance* by taking the Euclidean distance of the upper triangle values of GCM_{N_1} and GCM_{N_2} and we term it *Graphlet Correlation Distance (GCD)* between two networks. GCD is clean of redundancies and elegantly encodes much information about local network topology. We demonstrate that it outperforms other measures both on synthetic and real networks and we illustrate its utility on real-world problems (detailed below).

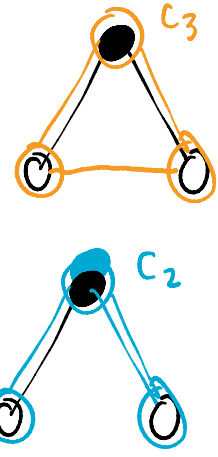
Evaluation on synthetic and real networks. To evaluate the performance of GCD for clustering networks of the same type, first we compare its results to those produced by other network statistics on synthetic networks belonging to seven different, commonly used, network models: Erdös-Rènyi random graphs (ER)[18], generalized random graphs with the same degree distribution as the data (ER-DD)[4], Barabási-Albert scale-free networks (SF-BA)[15], scale-free networks that model gene duplications and mutations (SF-GD)[19], geometric random graphs (GEO)[16], geometric graphs that model gene duplications and mutations (GEO-GD)[20], and stickiness-index based networks (STICKY)[21]. For each model, we generate 30 networks for each of the following four numbers of nodes and three edge densities that mimic the sizes and densities of real-world networks: 1000, 2000, 4000, and 6000 nodes, and 0.5%, 0.75%, and 1% edge density. Hence, the total number of synthetic networks that we compare using GCD (and other network statistics) is $7 \times 4 \times 3 \times 30 = 2,520$. Once we find GCD distances between all pairs of the 2, 520 networks, to illustrate the grouping (clustering) of these networks produced by GCD (a formal evaluation is presented below), we use the standard method of multi-dimensional scaling (MDS)[22] and embed the 2, 520 networks as points into 3-dimensional space so that their GCD distances are preserved as best as possible (Fig. 2 e). As illustrated in Fig. 2 e, networks belonging to the same model are grouped together in space regardless of size and edge density; model networks of the same size and density are grouped even better (Supplementary Fig. S5).

To illustrate its performance for grouping real networks from the same domain, we compute GCDs between all pairs of 11, 407 real-world networks from five different domains: 733 autonomous networks of routers that form the Internet, Facebook networks of 98 universities, metabolic networks of enzymes of 2, 301 organisms, 8, 226 protein structure networks, and 49 world trade networks corresponding to years 1962 to 2010 (detailed in Supplementary Information). As before, GCD-based MDS embedding of the 11, 407 networks shows clear groupings of networks from the same domain (Fig. 2 f). We interpret the grouping and the evolution of the world trade networks later in the text.

We formally evaluate the performance of GCD for clustering networks from the same model or real-world domain and systematically compare it to the performance of six other commonly used, or

the degree of the node, C_0, we can argue as follows. The neighbours of the node are either connected, or they are not: if they are connected, then they contribute to counts of triangles, C_3, that the node touches; if they are not connected, then they contribute to C_2 for the node, the number of times the node is touched by the middle of a 3-node path (orbit 2 in graphlet G_1, Fig. 1 d). Since these are the only options for connectedness of neighbours of a node, the number of ways in which C_0 neighbours of the node can be connected, $\binom{C_0}{2}$, is equal to the sum of C_2 and C_3 for the node: $\binom{C_0}{2} = C_2 + C_3$. Hence, if we know two of C_0, C_2 and C_3, we can derive the third, so one of them is redundant and does not need to be included in graphlet-based statistics. Similarly, we obtain a system of 17 linear equations describing all orbit redundancies (see Supplementary Information). When we solve it for the 73 orbits, 56 orbits remain non-redundant. There are 15 orbits for up to 4-node graphlets and 11 are non-redundant (red ones in Fig. 1 d). Similar redundancies, but in orbits of partial 4-node subgraphs have been reported[14].

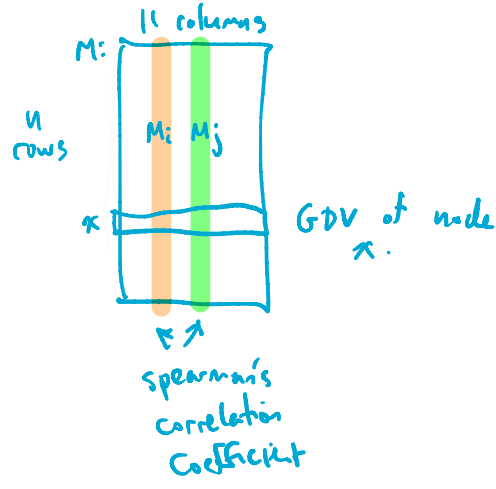
We [illegible] the dependencies (correlations) between graphlets as follows. First, we note that there are fewer [illegible]



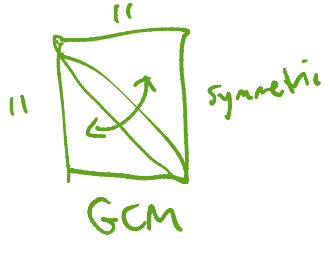
trast it with analogous statistics that include redundant and up to 5-node graphlet orbits as well.

Then, we devise a network statistic based on correlations between various node properties contained in non-redundant orbit counts, over all nodes, as follows. For each node in a network, first we construct its Graphlet Degree Vector consisting of 11 coordinates corresponding to the 11 non-redundant orbits. Then we construct a matrix whose rows are the above described Graphlet Degree Vectors, so the number of rows in the matrix is equal to the number of nodes in the network and it has 11 columns. The existence of correlations between non-redundant orbits over all nodes is exploited for constructing a new network statistic: for a given network N_1, we compute Spearman's Correlation coefficients between all pairs of columns of the above described matrix and present them in an 11×11 symmetric matrix that we term the *Graphlet Correlation Matrix (GCM)* of network N_1, GCM_{N_1}. In this way, we can summarize the topology of a network of any size into an 11×11 symmetric matrix with values in the interval $[-1, 1]$ (illustrated in Supplementary Fig. S2).

Different real and model networks generally have very different orbit dependencies and hence very different GCMs (Fig. 2 a–d). For example, in agreement with known properties of scale-free Barabási-



$C_{ij} = SCC(M_i, M_j) \in [-1, 1]$



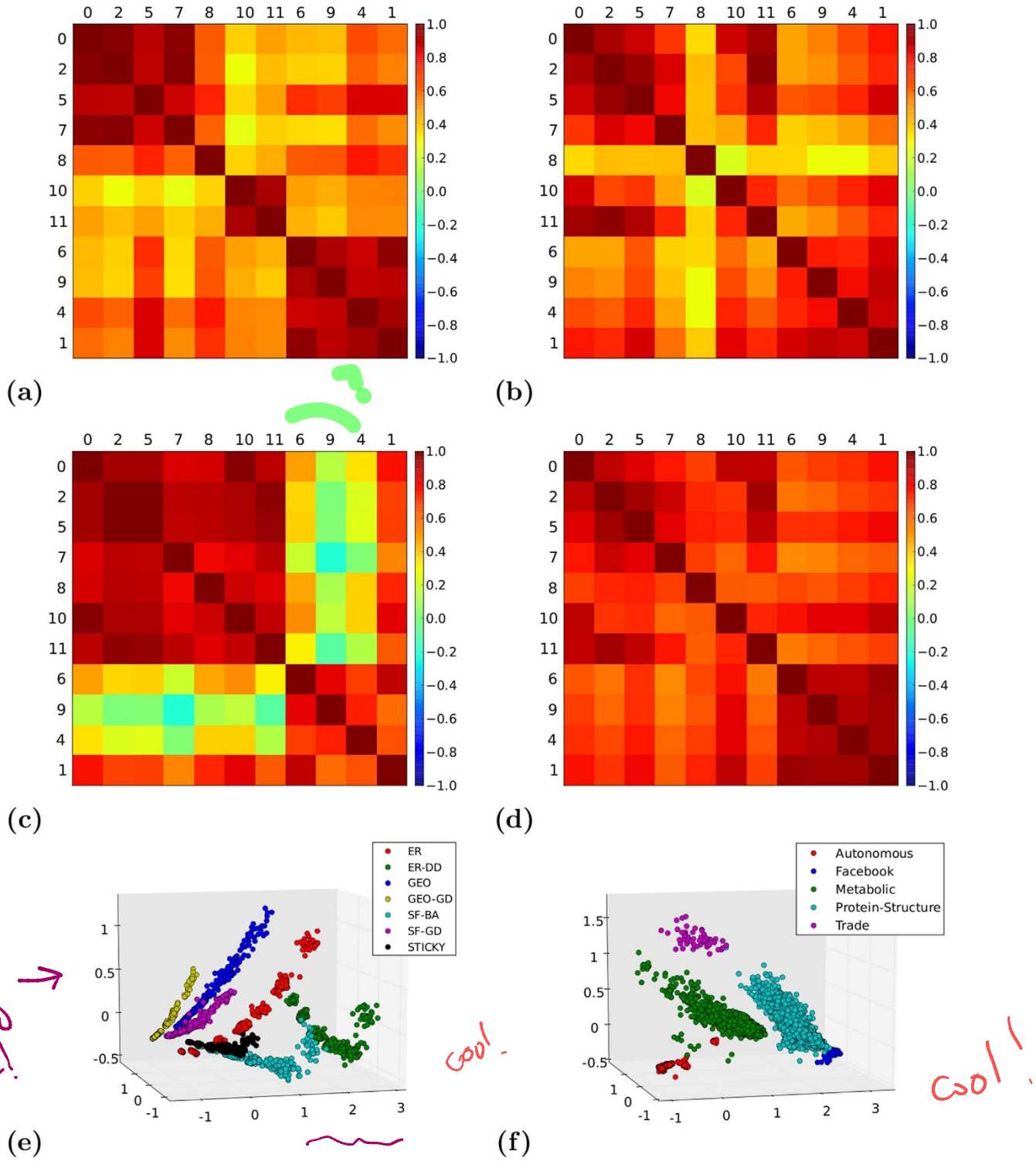


Figure 2 | Illustrations of GCMs. (a) a scale-free Barabàsi-Albert (SF-BA) network with 500 nodes and 1% edge-density; (b) a geometric random network (GEO) of the same size and density as network in (a); (c) the world trade network of 2010; and (d) the human metabolic network. Note that for SF-BA, GEO and metabolic networks, all the orbit correlations are statistically significant (p-values ≤ 0.05). This is not the case in the world trade network, where some correlations involving orbits 4 and 9 (the green cells in the GCM) have larger p-values. Illustrations of Graphlet Correlation Distance-based clustering of: (e) the 2,520 networks of various sizes and densities from 7 different random graph models: ER (red), ER-DD (green), GEO (blue), GEO-GD (yellow), SF-BA (light blue), SF-GD (purple), STICKY (black); (f) 11,407 real-world networks from 5 different domains: autonomous systems (red), Facebook (blue), metabolic networks (green), protein structure networks (light blue), and world trade networks (purple).

sensitive and robust network comparison measures (see Supplementary Information): degree distribution[4], clustering coefficient[4], network diameter[4], spectral distance[5], Relative Graphlet Frequency Distribution[13], and Graphlet Degree Distribution Agreement[8]. To make the comparison complete and evaluate what is gained by exclusion of redundant orbits or 5-node graphlets, we present comparison of the performance of GCD that includes the 11 non-redundant orbits for up to 4-node graphlets (that we term GCD-11) with the

performance of GCD constructed by using the full set of 73 orbits for all up to 5-node graphlets (termed GCD-73). Also, we make comparisons with GCDs constructed from all 15 orbits of up to 4-node graphlets (GCD-15) and from the 56 non-redundant orbits of up to 5-node graphlets (GCD-56): GCD-11 outperforms all measures for comparing networks of similar size and density, which is the most relevant for modelling network data, as models need to mimic sizes and densities of the data (see Supplementary Information).

In particular, one can test how well a distance measure groups networks of the same type by using the standard Precision-Recall curve: for small increments of parameter $\epsilon > 0$, if the distance between two networks is smaller than ϵ, then the pair of networks is retrieved. For each ϵ, precision is the fraction of correctly retrieved pairs (i.e., grouping together two networks from the same model), while recall is the fraction of the correctly retrieved pairs over all correct ones. The Area Under the Precision-Recall curve (AUPR), also called *average precision*, standardly measures the quality of the grouping by a given distance measure. We chose Precision-Recall curve analysis as it is known to be more robust to large numbers of negatives (in our case, negatives would be pairs of networks from different models that are grouped together) than Receiver Operator Characteristic (ROC) curve analysis[23].

Precision-Recall curves show that GCD-11 is the most precise among all tested measures (Fig. 3 a–b). Since the closest objects are the first to start forming clusters, we are interested in distance measures that optimize the number of correctly clustered pairs of networks that are at the shortest distance and hence are *retrieved first* by the distance measure[24]. Both GCD-11 and GCD-73 exhibit superiority in early retrieval over all other measures (beginning of the curves in Fig. 3 a). GCD-11 outperforms GCD-73 in this regard, because it contains fewer orbit dependencies and also has no redundancies, which introduce noise in GCD-73. Hence, GCD-11 is clearly the most sensitive measure for clustering networks. In addition, it is computationally efficient, since it involves counting only up to 4-node graphlets (see Supplementary Information).

Robustness to noise and missing data. Since real networks are noisy and incomplete, we evaluate the clustering quality of the above distance measures in the presence of noise. To simulate noise, we would like to randomize each of the above described 2, 520 synthetic networks 30 times (detailed below). However, if we were to randomize each of these networks 30 times, evaluating the results on the set of $2,520 \times 30 = 75,600$ networks would be computationally prohibitive. Hence, we use a subset of 280 out of the 2, 520 synthetic networks: for each of the 7 network models, we generate 10 networks for each of the following node sizes and edge densities: 1000 and 2000 nodes, and 0.5% and 1% edge density. We use these node sizes and edge densities because they correspond to networks that are more difficult to cluster than larger networks, so that if we show the methodology to be robust under these stringent conditions, we can be confident that it will be robust on real-world networks as well.

To simulate noise, we randomly rewire up to 90% of edges in the model networks of various sizes and densities described above and show that on these rewired networks, GCD-11 outperforms all other measures with respect to AUPR (Fig. 3 c; numbers on the vertical axis are not the same as those in column 2 of Fig. 3 b, since they correspond to the 280 networks described above, while those in Fig. 3 b correspond to the full set of 2, 520 networks). Similarly, it outperforms other measures on networks with missing data, which we simulate by randomly removing up to 90% of edges from model networks (Supplementary Fig. S7 a). Since many real networks are both noisy and incomplete[25,26], we ask how robust the measures are to missing edges in the presence of noise in the data. To answer that, we first randomly rewire 40% of edges in model networks to simulate noise and then randomly remove a percentage of edges to simulate missing data in the noisy networks. Again, GCD-11 outperforms all other measures even for networks with 40% of random noise that are missing up to 80% of edges at random (Fig. 3 d).

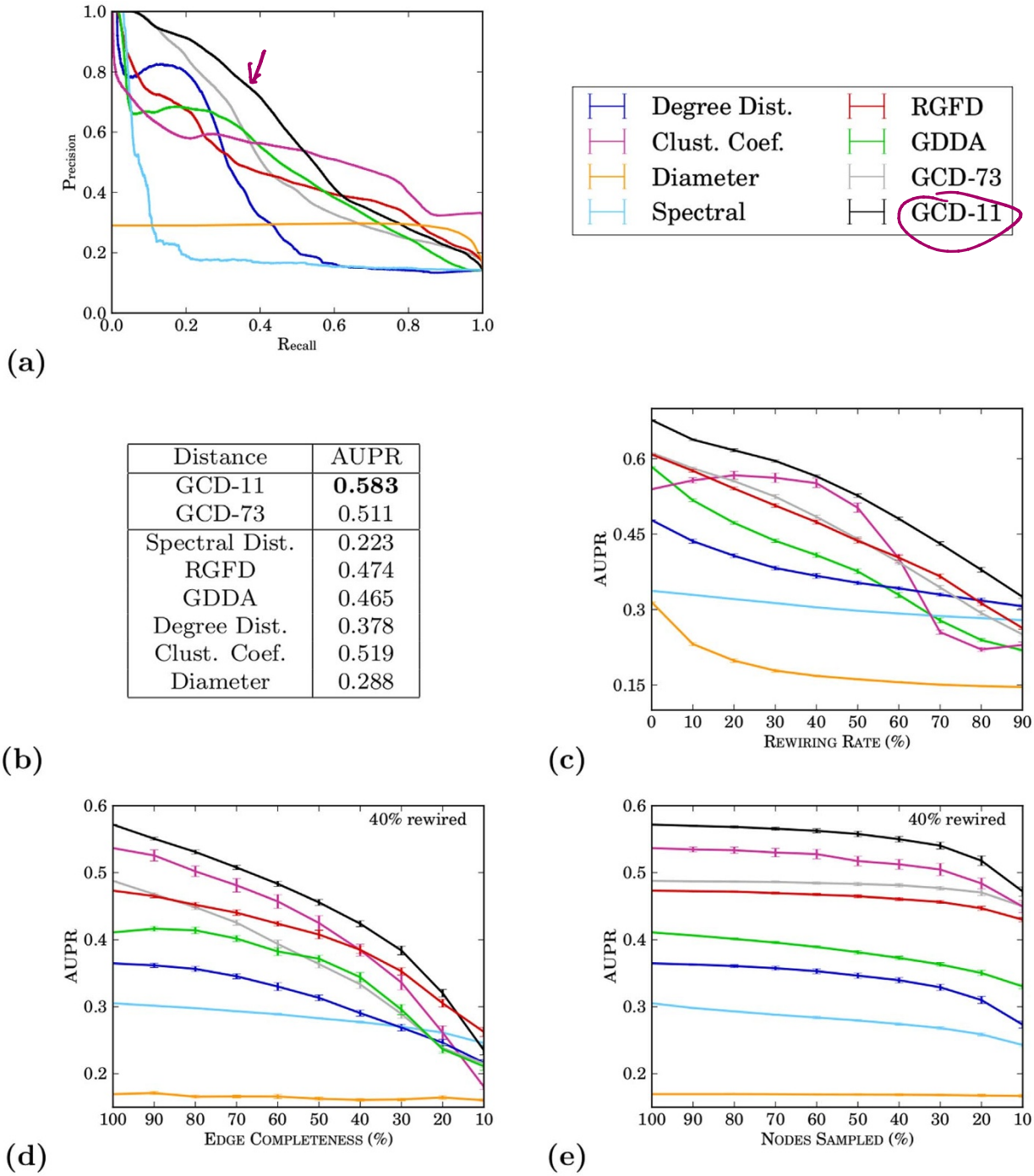
Furthermore, a surprising speed up in computational time can be obtained without loss in the clustering quality: by taking Graphlet Degree Vectors of as few as 30% of randomly chosen nodes in a model network to form GCM-11 (instead of taking Graphlet Degree Vectors of all nodes in the network), AUPR of GCD-11 only slightly decreases compared to when all nodes are used, and also it outperforms all other measures (Supplementary Fig. S7). In addition, for noisy and incomplete networks described above, the clustering obtained by GCD-11 not only outperforms those obtained by all other measures, but also it does not deteriorate even if we randomly sample as few as 30% of Graphlet Degree Vectors to form GCD-11 (Fig. 3 e) (see Supplementary Information).

These tests demonstrate robustness to noise and missing data and superiority of GCD-11 over other measures on a wide array of different network topologies, sizes and edge densities. The results improve further if we consider only networks of the same size and density (Supplementary Fig. S6).

World trade and other real network examples. Since GCM-11 is fast to compute and superior to other measures for clustering diverse networks even in the presence of large amounts of noise, we apply it to real networks in several domains.

Modelling networks from five domains. We use GCD-11 as a distance measure to evaluate the fit of network models to the above described 11, 407 real-world networks from five different domains. We use the state-of-the art non-parametric test to evaluate the fit[27,28] (Supplementary Fig. S8). Surprisingly, we find that networks from very different domains, Facebook, metabolic, and protein structure, are all best modelled by three network models: geometric random graphs (GEO), geometric graphs that mimic gene duplications and mutations (GEO-GD), and scale-free networks that also mimic gene duplications and mutations (SF-GD). While it is not difficult to explain why biological networks are the best fit by networks that model evolutionary processes, it may be surprising that Facebook networks seem to be organized by the same principles. A possible explanation is that Facebook grows as follows: when a person joins Facebook, he/she links to a group of his/her friends, which mimics a gene duplication, but he/she hardly ever has exactly the same friends as another person, which mimics the evolutionary process of divergence, or mutation. The fit of GEO to both Facebook and biological networks is perhaps more straightforward to explain, since all biological entities are subject to spatial constraints[20].

Crises and the topology of the world trade network. To gain insight into the relationship between economic crises and the world trade network (WTN), we apply GCD-11 to examine the dynamic changes of the WTN from year 1962 to 2010. We ask if rewiring of the WTN happens during crises and seek potential causes for the rewiring, or impacts of the rewiring. In particular, we test for correlations between time series of crude oil price changes and the topological changes in the WTN obtained by GCD-11. We shift these time series by up to 3 years forward and backward in time to see whether the change of WTN follows the change of oil price or vice versa and in what time interval. We test for all year shifts in $\{-3, -2, -1, 0, 1, 2, 3\}$ and report only statistically significant correlations (p-value ≤ 0.05) by using Spearman's correlation coefficients, which take into account the size of the change, and Phi correlation coefficients, which detect upward or downward trends only. Also, to cope with yearly data variability, we test the above correlations by grouping years in blocks of size 1, 2, and 3 years and report only statistically significant correlations (see Supplementary Information): for example, for changes in oil price in block sizes of 2 years for year 1990, we group year 1990 with the previous year, 1989, and find the average of the absolute



Distance	AUPR
GCD-11	**0.583**
GCD-73	0.511
Spectral Dist.	0.223
RGFD	0.474
GDDA	0.465
Degree Dist.	0.378
Clust. Coef.	0.519
Diameter	0.288

Figure 3 | Quality of clustering the 2,520 model networks using eight network distance measures (color coded and listed in the top panel). RGFD denotes Relative Graphlet Frequency Distribution and GDDA denotes Graphlet Degree Distribution Agreement. Error bars in panels (c) to (e) are one standard deviation above and below the mean. (a) Precision-Recall curves. (b) For each distance measure (the first column), the Area Under the Precision-Recall curve (AUPR, second column) achieved by a distance measure. (c) AUPR for different percentages of noise (randomly rewired edges, horizontal axis) in model networks (in 10% increments). (d) For "noisy" model networks, with 40% of edges randomly rewired, AUPR when x% of edges (horizontal axis) are kept in the network and 100 – x% are randomly removed (in 10% increments). (e) For "noisy" model networks, with 40% of edges randomly rewired, AUPR when a percentage of randomly sampled nodes (horizontal axis) is used to construct a distance measure; e.g., we obtain Graphlet Degree Distributions for x% of randomly chosen nodes to make up GCM-11 of the network and this is done for all networks before GCD-11 is computed between all pairs of networks.

values of the differences in oil prices between these two years and the two years that follow 1990, i.e., 1991 and 1992 [that is, $\frac{1}{4}(|price('91) - price('89)| + |price('91) - price('90)| + |price('92) - price('89)| + |price('92) - price('90)|)$].

We find that changes in crude oil price are correlated with changes in WTN topology and that they affect the WTN one and two years later (Fig. 4 a). Since WTN consists of trades in many commodities, different commodities are affected differently by the oil price (Supplementary Fig. S9 and Fig. S10), with the strongest and imme-

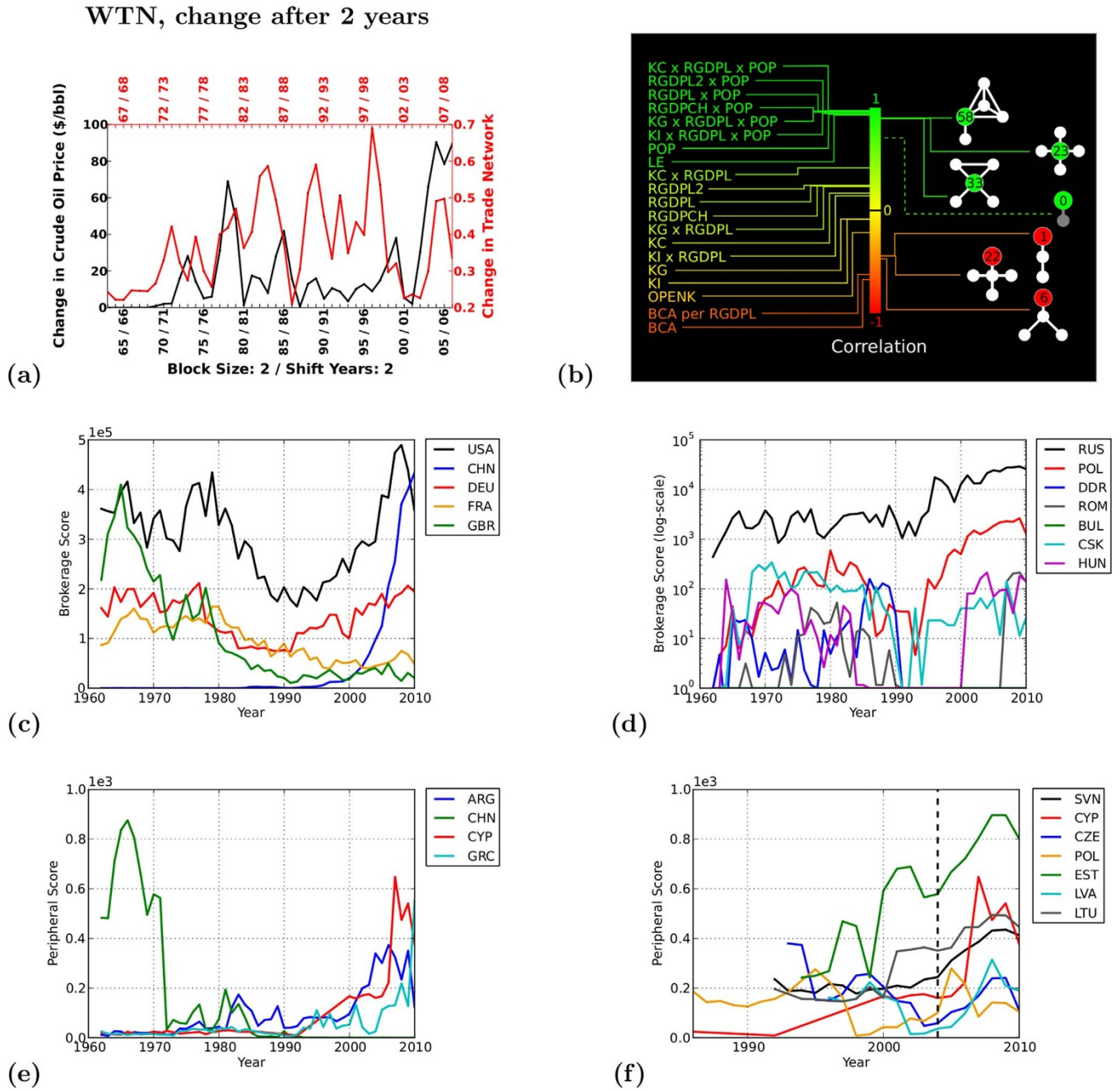


Figure 4 | Results of world trade network analysis. (a) Correlation of changes in crude oil price and changes in the structure of WTN, with the block size of two years and a two year shift; the Spearman correlation coefficient is 0.414 with the p-value of 0.005. (b) CCA correlations between economic attributes on the left (described in Supplementary Information) and graphlet degrees on the right; the middle bar is color-coded value of correlation. (c) Brokerage scores of the United States (USA), China (CHN), Germany (DEU), France (FRA), and the United Kingdom (GBR) from 1962 to 2010. (d) Brokerage scores of the Eastern Bloc from 1962 to 2010: the Soviet Union until 1991 replaced by Russia afterwards (RUS), Poland (POL), Eastern Germany (DDR), Romania (ROM), Bulgaria (BUL), Czechoslovakia until 1991 replaced by the sum of Czech Republic and Slovakia afterwards (CSK), and Hungary (HUN). (e) Peripheral scores of Argentina (ARG), China (CHN), Cyprus (CYP), and Greece (GRC) from 1962 to 2010. (f) Peripheral scores of countries that joined EU in 2004 and show an increase in their peripheral scores right before and after joining the EU: Slovenia (SVN), Cyprus (CYP), Czech Republic (CZE), Poland (POL), Estonia (EST), Latvia (LVA), and Lithuania (LTU).

diate effect (in the same year in which oil price changes) being on the trade of "Food and Live Animals" (Supplementary Fig. S10 a). This may be explained by agriculture needing oil, as well as by increase in demand for bio-fuels as oil price increases[29]. We further confirm this by observing that the correlation between oil price and the structure of the network of trade in "Food and Live Animals" increases over time, as agriculture becomes more oil dependent: Phi correlation coefficient rises from 0.31 in years 1962 to 1986, to 0.51 in years 1986 to 2007.

We ask if we can get similar results by using network similarity measures other than GCD-11. To that end, we seek for correlations between changes in crude oil price and changes in WTN structure measured by each of the above described network similarity measures: degree distribution, clustering coefficient, network diameter,

spectral distance, RGFD, and GDDA. The only relevant result is that GDDA uncovers a potentially interesting, but hard to explain correlation: a change in WTN structure (as reported by GDDA) is followed by a change in crude oil price 3 years later. Explaining this observation is a subject of future research. All other network similarity measures produce irrelevant correlations, such as "Beverage and Tobacco" trade network changes (observed by RGFD) correlating with changes in oil price two years later. The list of all correlations found by each of the similarity measures is available in the Supplementary Data.

We recall our previous observation about a country in the WTN being either peripheral or clustered/broker, but not both, and offer a qualitative explanation. In particular, we use the standard method of Canonical Correlation Analysis (CCA)[30] to correlate economic indicators of the development of a country[31,32] with its graphlet-based position in the WTN (see below). Interestingly, the indicators of economic wealth (e.g., gross domestic product, level of employment, consumption share of purchasing power parity; described in Supplementary Information) strongly correlate with a country being in a brokerage relationship (i.e., a mediator between unconnected countries), or within a cluster of densely connected countries, while the indicators of economic poverty (e.g., current account balance) correlate with a country being peripheral in the network, i.e., linked only to one other country by a trade relationship (Fig. 4 b). Since a country is either peripheral or clustered/broker, this may indicate that one of the factors that contribute to the wealth of a country could be its brokerage/clustered position in the WTN.

To evaluate if the above result linking GDP to a country's wiring in the WTN can be obtained by simpler, non-graphlet-based, previously used measures of node wiring, such as node degree, clustering coefficient, and betweenness centrality[33–35], we compute the Pearson's correlation coefficients (PCCs) between the GDPs of countries and each of these node statistics. To assess the quality of the CCA analysis described above, we measure the PCC between GDP and the graphlet degree of orbit 58, since it has the largest coefficient reported by CCA that links it to GDP. We demonstrate superiority of our method over others, since we find that orbit 58 outperforms all other statistics, achieving with GDP the PCC of 0.869, followed by betweenness centrality achieving PCC of 0.816, node degree (i.e., orbit 0) achieving PCC of 0.690, and clustering coefficient achieving PCC of -0.136. This demonstrates that our graphlet-based method finds more refined topological features than previously used methods[33–35] and that even betweenness centrality gives a more coarse-grained insight in to the function of WTN.

To quantify the strength of the brokerage position of a country in the WTN of each year, we define the *brokerage score* of the country in a particular year as the weighted linear combination of broker graphlet degrees (i.e., C_{23}, C_{33}, C_{44}, and C_{58}) using the coefficients obtained from CCA. Similarly, we quantify how *peripheral* a country is in the WTN of a particular year by using C_{15}, C_{18}, and C_{27}. Since we have demonstrated above that a country is either a broker or peripheral in each year, these brokerage and peripheral scores enable us to track changes in the position of a country in the WTN over years. We analyse if the changes in brokerage and peripheral scores of a country over years coincide with economic crises and other events impacting the economy of the country.

Indeed, we find that during 1980s, brokerage scores of the world's highest brokers fall (Fig. 4 c), for which we find support in the economics literature. For example, in the USA during the first Reagan administration, a mix of monetary policy and loose budgets sky-rocketed the dollar and sent international balances in the wrong direction. The merchandise trade deficit rose above $100 billion in 1984 and remained there throughout the decade. The ratio of the USA imports to exports during the eighties peaked at 1.64, a disproportion not seen since the War between the States. Such a drop in the export power of the USA, and thus the change of its position in the trade network (drop of its brokerage score in the WTN, black line in Fig. 4 c), had no precedent in modern USA history[36]. Another example is that of Great Britain. There is a huge drop in its brokerage score as it loses the Empire in the 1960s, seeing a small improvement in 1973 when the Conservative Prime Minister, Edward Heath, led it into the European Union (EU). However, the downward trend induced by the dissolution of the colonial superpower has continued[37]. In contrast, the reunification of Germany transformed it from being in the shadow of the Second World War a peripheral economy of Western Europe, with most of the decisions in Europe having been made by France and the UK, to being the central economy of Europe[38]. Among the countries of the former Eastern Bloc, USSR has been the most dominant broker, with both Russia and Poland sharply gaining in brokerage scores after the fall of communism (Fig. 4 d; y-axis is in logarithmic scale).

Similarly, peripheral scores (Fig. 4 e) are consistent with economic reality. China's peripheral score dropped sharply in the early 1970s, which coincides with President Nixon's international legitimization of China[39]. This was a turning point that changed China's closed economy to one deeply integrated with global financial markets[40], as evident not only by its fallen peripheral score (Fig. 4 e), but also by its increased brokerage score that has surpassed that of the USA in 2009 (Fig. 4 c). Conversely, raising peripheral scores of Argentina, Cyprus and Greece coincide with their recent economic crises. By year 2001, poor management in great part led to Argentina's real GDP shrinkage, unemployment sky-rocketed, and the international trade plunged, so Argentina turned into a peripheral economy[41]. Less than a decade later, Cyprus and Greece went the "South American way:" the similarities, starting with the fixed exchange regime followed by the bank runs, were striking[42].

Interestingly, accession of countries into the EU makes them more peripheral in the WTN, as evident by increases in their peripheral scores before and after accession (Fig. 4 f). Even though all trade within the EU is exempt from import taxes, at the time of accession new members are required to leave other advantageous free trade associations (e.g., BAFTA, CEFTA, CISFTA, EFTA). The fact that a country has to leave free trade agreements with other non-EU member countries leads to the destruction of trade connections while the positive effects of EU accession on trade need time to materialize. In other words, since trade connections are easy to break, but much more difficult to build, EU accession increases the peripheral score of a country and whether and when the country will recover remains an open question.

We assess if similar can be observed by other node measures previously applied to WTNs, such as betweenness and closeness centralities[35], and find that it cannot. In particular, we plot betweenness centrality of a country over years (and also its betweenness peripherality, that we define by subtracting from 1 the value of its betweenness centrality) and find that we cannot detect the events that can be detected by our brokerage and peripheral scores described above, such as the drop of export power of the USA, or the fall of Argentina, Cyprus and Greece (see Supplementary Fig. S11 a and b). Similarly, closeness centrality and peripherality (defined analogous to betweenness peripherality described above) can also not detect these events (Supplementary Fig. S11 c and d).

Final remarks on GCD. We have shown that by exploiting correlations between node characteristics in a network (e.g., broker/clustered and peripheral nodes in the world trade network), we can sensitively and robustly uncover the network type and track network dynamics. This is possible because real-world networks generally have few types of nodes with well defined characteristics and because the node characteristics are differently correlated in different types of networks. We have uncovered some of the node characteristics, in particular broker/clustered and peripheral ones in the world trade network, that are amenable to economic

interpretation. In particular, during a crisis, a country becomes more peripheral and less of a broker in the WTN than in economically stable periods. Accession of a country to the EU has a similar effect. The methodology promises to deliver insight in many other areas; for example, it can help detect online or telephone-based terrorist activities, because it can robustly and sensitively typify a newly formed network by identifying the most similar known network group.

1. Liu, Y.-Y., Slotine, J.-J. & Barabási, A.-L. Controllability of complex networks. *Nature* **473**, 167–173 (2011).
2. Galbiati, M., Delpini, D. & Battiston, S. The power to control. *Nat. Phys.* **9**, 126–128 (2013).
3. Cook, S. A. The complexity of theorem-proving procedures. In *Proceedings of the Third annual ACM symposium on Theory of Computing,* 151–158 (ACM, 1971).
4. Newman, M. *Networks: An Introduction* (Oxford University Press, Oxford, 2009).
5. Wilson, R. C. & Zhu, P. A study of graph spectra for comparing graphs and trees. *Pattern Recogn.* **41**, 2833–2841 (2008).
6. Thorne, T. & Stumpf, M. P. Graph spectral analysis of protein interaction network evolution. *J. R. Soc. Interface* **9**, 2653–2666 (2012).
7. Milo, R. *et al.* Network motifs: Simple building blocks of complex networks. *Science* **298**, 824–827 (2002).
8. Pržulj, N. Biological network comparison using graphlet degree distribution. *Bioinformatics* **23**, 177–183 (2007).
9. Della Rossa, F., Dercole, F. & Piccardi, C. Profiling core-periphery network structure by random walkers. *Sci. Rep.* **3**, 1467 (2013).
10. Milo, R. *et al.* Superfamilies of evolved and designed networks. *Science* **303**, 1538–1542 (2004).
11. Artzy-Randrup, Y., Fleishman, S. J., Ben-Tal, N. & Stone, L. Comment on "Network motifs: simple building blocks of complex networks" and "Superfamilies of evolved and designed networks" *Science* **305**, 1107–1107 (2004).
12. Guerrero, C., Milenković, T., Pržulj, N., Kaiser, P. & Huang, L. Characterization of the proteasome interaction network using a qtax-based tag-team strategy and protein interaction network analysis. *Proc. Nat. Acad. Sci. U.S.A.* **105**, 13333–13338 (2008).
13. Pržulj, N., Corneil, D. G. & Jurisica, I. Modeling interactome: scale-free or geometric? *Bioinformatics* **20**, 3508–3515 (2004).
14. Marcus, D. & Shavitt, Y. RAGE – a rapid graphlet enumerator for large networks. *Comput. Netw.* **56**, 810–819 (2012).
15. Barabási, A.-L. & Albert, R. Emergence of scaling in random networks. *Science* **286**, 509–512 (1999).
16. Penrose, M. *Random geometric graphs* (Oxford University Press, Oxford, 2003).
17. United Nations, United nations commodity trade statistics (COMTRADE) database., (2010) (Date of access: 15/11/2011) URL: http://comtrade.un.org.
18. Erdös, P. & Rényi, A. On random graphs. *Publ. Math. Debrecen* **6**, 290–297 (1959).
19. Vázquez, A., Flammini, A., Maritan, A. & Vespignani, A. Modeling of protein interaction networks. *Complexus* **1**, 38–44 (2002).
20. Pržulj, N., Kuchaiev, O., Stevanovic, A. & Hayes, W. Geometric evolutionary dynamics of protein interaction networks. *Pac. Symp. on Biocomput.* **2009**, 178–189 (2010).
21. Pržulj, N. & Higham, D. J. Modelling protein–protein interaction networks via a stickiness index. *J. R. Soc. Interface* **3**, 711–716 (2006).
22. Cox, T. F. & Cox, M. A. *Multidimensional Scaling* (CRC Press, Florida, 2010).
23. Davis, J. & Goadrich, M. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning (ICML '06)*, 233–240 (ACM, New York, NY, USA, 2006).
24. Yu, Y.-K., Gertz, E. M., Agarwala, R., Schäffer, A. A. & Altschul, S. F. Retrieval accuracy, statistical significance and compositional similarity in protein sequence database searches. *Nucleic Acids Res.* **34**, 5966–5973 (2006).
25. Han, J.-D. J., Dupuy, D., Bertin, N., Cusick, M. E. & Vidal, M. Effect of sampling on topology predictions of protein-protein interaction networks. *Nat. Biotechnol.* **23**, 839–844 (2005).
26. Stumpf, M. P., Wiuf, C. & May, R. M. Subnets of scale-free networks are not scale-free: sampling properties of networks. *Proc. Nat. Acad. Sci. U.S.A.* **102**, 4221–4224 (2005).
27. Rito, T., Wang, Z., Deane, C. M. & Reinert, G. How threshold behaviour affects the use of subgraphs for network comparison. *Bioinformatics* **26**, i611–i617 (2010).
28. Hayes, W., Sun, K. & Pržulj, N. Graphlet-based measures are suitable for biological network comparison. *Bioinformatics* **29**, 483–491 (2013).
29. Headey, D. & Fan, S. Anatomy of a crisis: the causes and consequences of surging food prices. *Agr. Econ.* **39**, 375–391 (2008).
30. Hair, J. F., Anderson, R. E., Tatham, R. L. & William, C. *Multivariate Data Analysis* (Prentice-Hall International, WC, 1998).
31. Heston, A., Summers, R. & Aten, B. PENN world table, (2002) (Date of access: 15/11/2011) URL: https://pwt.sas.upenn.edu/.
32. Fund, I. M. World economic outlook (WEO) database, (2006). (Date of access: 15/10/2012) URL: http://www.imf.org/external/pubs/ft/weo/2012/02/weodata/index.aspx.
33. Serrano, M. A. & Boguñá, M. Topology of the world trade web. *Phys. Rev. E* **68**, 015101 (2003).
34. Fagiolo, G., Reyes, J. & Schiavo, S. World-trade web: Topological properties, dynamics, and evolution. *Phys. Rev. E* **79**, 036115 (2009).
35. De Benedictis, L. & Tajoli, L. The world trade network. *World Econ.* **34**, 1417–1454 (2011).
36. Destler, I. US trade policy-making in the eighties. In *Politics and Economics in the Eighties*, 251–284 (University of Chicago Press, 1991).
37. Kindleberger, C. P. Government policies and changing shares in world trade. *Am. Econ. Rev.* **70**, 293–298 (1980).
38. Mundell, R. A. A reconsideration of the twentieth century. *Am. Econ. Rev.* **90**, 327–340 (2000).
39. Cukierman, A. & Tommasi, M. When does it take a Nixon to go to China? *Am. Econ. Rev.* **88**, 180–97 (1998).
40. Prasad, E. S. & Rajan, R. G. Modernizing China's growth paradigm. *Am. Econ. Rev.* **96**, 331–336 (2006).
41. Arellano, C. Default risk and income fluctuations in emerging economies. *Am. Econ. Rev.* **98**, 690–712 (2008).
42. Berka, M., Devereux, M. B. & Engel, C. Real exchange rate adjustment in and out of the eurozone. *Am. Econ. Rev.* **102**, 179–85 (2012).

Acknowledgments

We thank Michael Stumpf, Dimitris Achlioptas, and Des Higham for their comments and assistance with this work. Supported by the European Research Council (ERC) Starting Independent Researcher Grant 278212 and the USA National Science Foundation (NSF) Cyber-Enabled Discovery and Innovation (CDI) grant OIA-1028394; EU Creative Core FISNM-3330-13-500033, ARRS Program P1-0383 and Project J1-5454 (Z.L.); and the intramural program of the USA National Library of Medicine (A.S.).

Author contributions

Ö.N.Y. performed all the analyses except the canonical correlation analysis on world trade networks; N.M.D. was involved in experimental design; D.D. performed the canonical correlation analysis on world trade networks; V.J. collected the world trade network datasets; R.K. interpreted the results of the world trade network analysis; A.S. and N.P. designed and supervised the study, and analysed the results; Z.L. and N.P. wrote the manuscript. All authors discussed the results and commented on the manuscript.

Additional information

Supplementary information accompanies this paper at http://www.nature.com/scientificreports

Competing financial interests: The authors declare no competing financial interests.

How to cite this article: Yaveroğlu, Ö.N. *et al.* Revealing the Hidden Language of Complex Networks. *Sci. Rep.* **4**, 4547; DOI:10.1038/srep04547 (2014).

Journal of Statistical Software

June 2015, Volume 65, Issue 12. http://www.jstatsoft.org/

ergm.graphlets: A Package for ERG Modeling Based on Graphlet Statistics

Ömer Nebil Yaveroğlu
Imperial College London

Sean M. Fitzhugh
University of California, Irvine

Maciej Kurant
Google, Zurich

Athina Markopoulou
University of California, Irvine

Carter T. Butts
University of California, Irvine

Nataša Pržulj
Imperial College London

Abstract

Exponential-family random graph models are probabilistic network models that are parametrized by sufficient statistics based on structural (i.e., graph-theoretic) properties. The **ergm** package for the R statistical computing environment is a collection of tools for the analysis of network data within an exponential-family random graph model framework. Many different network properties can be employed as sufficient statistics for exponential-family random graph models by using the model terms defined in the **ergm** package; this functionality can be expanded by the creation of packages that code for additional network statistics. Here, our focus is on the addition of statistics based on graphlets. Graphlets are classes of small, connected, induced subgraphs that can be used to describe the topological structure of a network. We introduce an R package called **ergm.graphlets** that enables the use of graphlet properties of a network within the **ergm** package of R. The **ergm.graphlets** package provides a complete list of model terms that allows to incorporate statistics of any 2-, 3-, 4- and 5-node graphlets into exponential-family random graph models. The new model terms of the **ergm.graphlets** package enable both exponential-family random graph modeling of global structural properties and investigation of relationships between node attributes (i.e., covariates) and local topologies around nodes.

Keywords: graphlet, graphlet degree, subgraph, exponential-family random graph model, **ergm**, **statnet**, R.

1. Introduction

Networks are widely used representations of complex, relational systems from different domains such as biology, sociology, economics, and technology. A *network* (or *graph*) consists of *nodes* (or *vertices*) that represent the objects of the complex system and *edges* that represent the relationships between the objects. For example, in a friendship network, the nodes correspond to people, and an edge is drawn between two people if they are friends with each other (illustrated example in Figure 1). Networks can be further enriched with node attributes that describe various categorical features (e.g., the gender of the people in the friendship network) or numeric features (e.g., the age of the people in the friendship network) of the nodes.

Understanding the processes underlying the formation of edges in a network is one of the main challenges in network modeling. Various network models describe different rules for formation of edges; e.g., Erdös-Rényi random graph models (also known as Bernoulli graphs; Erdös and Rényi 1959), so-called "scale-free" models (Barabási and Albert 1999), geometric models (Penrose 2003), and stickiness-index-based models (Pržulj and Higham 2006). Recent work on the statistical modeling of networks has focused on the use of discrete exponential families as general representations for these and other graph distributions. Exponential-family random graph models (ERG models or ERGMs, also known as "p*" models) are probabilistic network models that are parametrized in terms of sufficient statistics based on various topological properties (Holland and Leinhardt 1981; Pattison and Wasserman 1999; Robins, Pattison, Kalish, and Lusher 2007). In ERGMs, the conditional probability of the existence of an edge given the rest of the graph is determined by the effect that the edge has on the values of these statistics (and hence topology) which are conventionally called model *term*s. Using suitable model terms, ERGMs enable statistical investigation of the importance of different structural properties on the formation of edges. For example, for a friendship network, ERGMs can answer questions such as: Are the chances of a friendship tie between two persons enhanced by having a friend in common? Is this effect stronger than would be expected due to clustering on observed characteristics (e.g., gender)? Does this effect differ based on the gender or race of the common friend? Etc.

The **ergm** package (Hunter, Handcock, Butts, Goodreau, and Morris 2008b; Handcock, Hunter, Butts, Goodreau, Krivitsky, and Morris 2014) for the R statistical computing environment (R Core Team 2015) provides a set of tools for analyzing networks within an ERGM framework. The **ergm** package allows the users to define ERGMs based on a wide range of network properties, fit ERGMs to observed networks using likelihood-based methods, simulate networks from an ERGM, perform graphical goodness-of-fit tests of the type described by Hunter, Goodreau, and Handcock (2008a) and Handcock, Hunter, Butts, Goodreau, and Morris (2003). The **ergm** package itself provides a large but limited number of model terms. Custom model terms can be introduced into the **ergm** package using the **ergm.userterms** package (Handcock, Hunter, Butts, Goodreau, and Morris 2013).

Using the functionality of the **ergm.userterms** package, we introduce a new R package called **ergm.graphlets** that enables defining ERGMs based on induced subgraph (also known as *graphlet*) properties of networks. The **ergm.graphlets** package provide model terms for an extended list of subgraph properties that capture all connected, undirected, induced subgraph patterns of size 2, 3, 4 and 5. Furthermore, the terms of the **ergm.graphlets** package differ from the available subgraph property terms of the **ergm** package as only "induced" subgraph patterns are taken into account; when evaluating a subgraph induced on a set of nodes, all

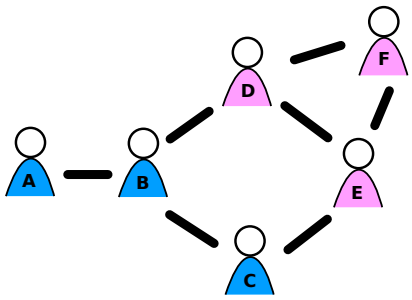


Figure 1: A hypothetical friendship network where nodes correspond to people and edges are drawn between nodes if they are friends with each other. The node colors correspond to gender (e.g., blue = male).

edges connecting the chosen set of nodes are considered.

In the remainder of this article, we proceed as follows. First, we provide some background information on various network properties, graphlets and ERGMs (Section 2). Second, we provide detailed explanations of the new model terms of the **ergm.graphlets** package in Section 3. Third, we illustrate the ERG modeling process with the new model terms on two real-world networks in Section 4. Finally, we conclude by providing a brief summary and discussing the future directions in Section 5.

2. Background

In this section, we provide a brief introduction to the graph-theoretic definitions, network properties, graphlets and exponential-family random graph models.

2.1. Definitions, network properties and graphlets

A *network* (or graph) is represented as $G = (V, E)$ where V is the set of nodes and E is the set of edges of graph G. Edges are represented by pairs of nodes, and represent ties; two node joined by an edge are said to be *adjacent*. A network G' is a *subgraph* of a network G if its nodes and edges are subsets of the nodes and edges of G. A subgraph G' is *induced* if it contains all the edges that appear between its nodes in its originating network G. Different subgraphs of a network can have very different configurations, such as a triangle, k-star, or k-cycle. A *triangle* is a complete network of three nodes (i.e., where each pair of nodes is adjacent). A *k-star* is a network of $k+1$ nodes where some node is adjacent to all other nodes. A *k-cycle* is a network of k nodes such that there exists an ordering of the nodes $v_1, v_2, \ldots, v_k$ such that each node is adjacent to the node immediately before and after it, and the first node is adjacent to the last.

For understanding complex systems, analyzing the topological properties of their network representations is crucial. Many such properties have been defined and found to be useful in various substantive contexts. The degree distribution (i.e., the distribution of the number of neighbors each node has), clustering coefficient (a measure of the tendency of edges to be contained in triangles), and diameter (i.e., the length of the maximum shortest path between any two nodes in the network) are among the most well-known examples of structural

properties (Wasserman and Faust 1994; Newman 2010). Recent work has identified many useful properties based on graphlets. *Graphlets* are isomorphic equivalence classes of small, connected induced subgraphs within a larger network (Pržulj, Corneil, and Jurisica 2004). The set of possible graphlets of a given order (number of nodes) can be enumerated, and we depict the set of 2- to 5-node graphlets in Figure 2. A range of different structural properties can be defined by reference to graphlets. The most basic graphlet properties – *graphlet counts* – are defined as the number of times that each graphlet appears in a given network. E.g., for the friendship network in Figure 1: the count of G_0 is the number of edges in the network, 7; the count of G_1 is the number of induced two-path subgraphs, 8; the count of G_2 is the number of triangles, 1, etc. More refined network properties can be defined by considering the symmetries (i.e., automorphisms) within the graphlets (Milenković and Pržulj 2008). Two nodes within a network are said to belong to the same *automorphism orbit* (or *automorphic equivalence class*) if there exists a relabeling of nodes in the graph that exchanges the two nodes while preserving the graph's adjacency structure (Wasserman and Faust 1994). Applying this notion to each 2- to 5-node graphlet yields 73 equivalence classes (i.e., *orbits*), as illustrated in Figure 2. Each orbit reflects a distinct way of participating in a graphlet structure, and counts of orbit memberships provide a node-level indicator of structural position. The *graphlet degree* of a node is the number of graphlets that the node touches at a given orbit; this generalizes the conventional notion of *degree*, which is the size of a node's neighborhood (in graphlet terms, the number of type 0 orbits that it occupies). The computation of the 73 graphlet degrees for node A in the friendship network is illustrated in Figure 3. The vector containing the 73 graphlet degrees of a node, named the *graphlet degree vector (GDV)*, provides a detailed description of network structure local to a node. Finally, the third set of graphlet properties considered here summarizes the node-level graphlet degrees by considering their distribution over the whole network. A generalization of the degree distribution, the *graphlet degree distribution* of an orbit corresponds to the distribution of the corresponding graphlet degrees of all nodes in the network (with the conventional degree distribution being the graphlet degree distribution of orbit 0). The topology of a network can be richly described with the 73 graphlet degree distributions associated with each of the 2- to 5-node graphlet automorphism orbits.

2.2. Exponential-family random graph modeling

Exponential-family random graph models (ERGMs) are probabilistic network models parametrized by sufficient statistics based on different network properties. ERGMs are specified via three elements: a vector of terms (sufficient statistics or functions thereof); a vector of real-valued parameters; and a support (often chosen to be the set of all graphs or digraphs of a given order; Kolaczyk 2009; Hunter *et al.* 2008b).[1] Sufficient statistics for an ERGM can be functions representing any topological properties of the network (and, optionally, covariates), e.g., the number of edges, the degree distribution, the number of triangles, the number of k-stars, or the number of k-cycles. In general, few constraints on model terms are required; any real-valued functions are permissible, so long as they are finite and (for identifiable models) affinely independent on the support. Model terms can also relate node or edge attributes with their topological properties, e.g., the correlation between a node's attribute value and its degree. Readers can refer to Morris, Handcock, and Hunter (2008) for a summary of model

[1] Technically, a *reference measure* is also required; for unvalued graphs on finite support, this can be taken without loss of generality to be the counting measure (Krivitsky 2012).

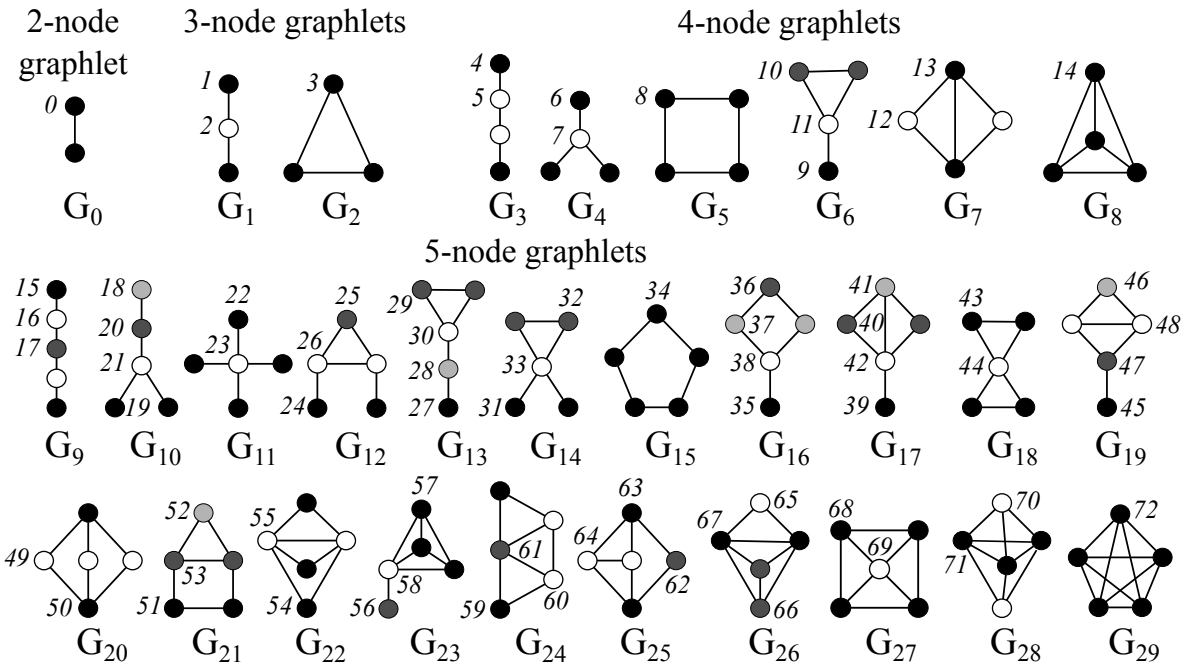
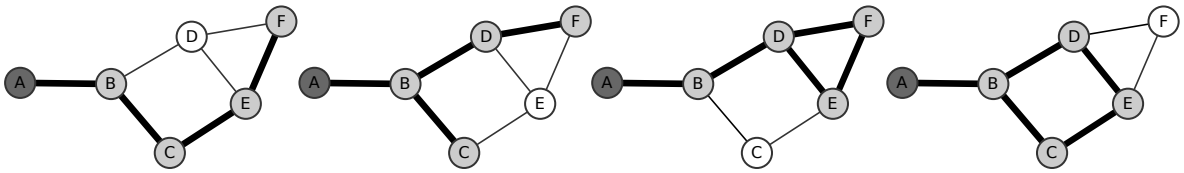


Figure 2: All 2-, 3-, 4- and 5-node graphlets, G_0, G_1, ..., G_{29}, and their automorphism orbits, 0, 1, 2, ..., 72. (Pržulj 2007)



Orbit	0	1	2...3	4	5	6	7...14	15	16...18	19	20...26	27	28...34	35	36...72
GDV(A)	1	2	0...0	3	0	1	0...0	1	0...0	1	0...0	1	0...0	1	0...0

Figure 3: Computation of the graphlet degree vector (GDV) of node A in the friendship network in Figure 1. The number of graphlets that node A touches at orbit i is the ith element of the GDV (Milenković and Pržulj 2008).

terms that are available in the **ergm** package.

ERGMs may be more formally summarized as follows. Let $\mathbf{Y}$ be a random variable that represents the n-by-n adjacency matrix of an unweighted, loopless (no self-edges), undirected network with n nodes. $\mathbf{Y}$ can have $2^{\binom{n}{2}}$ different values (configurations), where each value represents a different network having n nodes. The number of configurations arises from the fact that there are $\binom{n}{2}$ dyads in an order-n graph, each of which may here take two distinct states. The set of all possible configurations forms the *support* for $\mathbf{Y}$, denoted here by $\mathcal{Y}$. Any element of $\mathcal{Y}$ is a potential *realization* of $\mathbf{Y}$ and is represented by $\mathbf{y}$. An ERGM describes the

probability of observing a realization, $\mathbf{y}$, as a function of a vector of sufficient statistics. The probability of observing a realization is expressed in ERGM form per Equation 1:

$$\mathsf{P}_{\theta,\mathcal{Y}}(\mathbf{Y} = \mathbf{y}|\theta, t) = \frac{\exp\{\theta^\top t(\mathbf{y})\}}{\sum_{z\in\mathcal{Y}} \exp\{\theta^\top t(z)\}}, \mathbf{y} \in \mathcal{Y}, \tag{1}$$

where θ is the vector of model coefficients (i.e., the weights for the model terms) and t is the vector of sufficient statistics (i.e., model terms corresponding to network properties of interest; Frank and Strauss 1986; Wasserman and Pattison 1996). A generalization of the above to more general cases (e.g., graphs with loops, digraphs, etc.) is immediate given an alternative choice of $\mathcal{Y}$; extension to valued graphs is treated by Krivitsky (2012). Since any probability mass function for $\mathbf{Y}$ on finite $\mathcal{Y}$ can be written in this form, ERGMs are a fully general representation for random graphs of finite order.

In an inferential context, ERG models for an observed network, $\mathbf{y}$, are typically fit by estimating the model coefficients, θ, that maximize the conditional probability, $\mathsf{P}_{\theta,\mathcal{Y}}(\mathbf{Y} = \mathbf{y}|\theta, t)$ for some selected t (with statistics being chosen based on a combination of exploratory analysis and prior theory). The most common approaches to estimation are currently maximum pseudo-likelihood estimation (MPLE, generally avoided except as an approximation) and maximum likelihood estimation (MLE, implemented via one of several techniques). Since the computation of the normalizing factor (i.e., denominator) in Equation 1 is intractable, current MLE methods do not directly compute the normalizing factor, but instead, use Markov chain Monte Carlo (MCMC) algorithms to perturb the edge states of the networks one-by-one and estimate the model parameters based on the change statistics of these edge flips (for details, see Handcock *et al.* 2014). One consequence of this is that the model statistics themselves need never be directly computed: for most purposes, only the change scores of edge flips are directly necessary. This approach yields substantial savings in the computational time required for estimating the model parameters.

The **ergm** package also employs this approach for estimating the model parameters of an ERGM. For this reason, when defining new model terms with the **ergm.userterms** package, users need to focus on identifying efficient ways of computing the change statistics of the new model terms. For example, for defining "the number of edges" term, the implementation should return $+1$ when a new edge is added into the network and -1 when an edge is removed. Since these change statistics computations are likely to be performed millions of times during a typical MCMC run for parameter estimation, the computation of the change statistics should be time-optimized.

3. The ergm.graphlets package

We define graphlet statistics for ERGMs by introducing the **ergm.graphlets** package (Yaveroglu, Fitzhugh, Kurant, Markopoulou, Przulj, and Butts 2015) that is built upon the **ergm.userterms** package. The **ergm.graphlets** package is available from the Comprehensive R Archive Network (CRAN) at http://CRAN.R-project.org/package=ergm.graphlets. To install and load **ergm.graphlets**, type the following in R prompt:

```
R> install.packages("ergm.graphlets")
R> library("ergm.graphlets")
```

The **ergm.graphlets** package is open-source and released under GPL-2 and higher. The **ergm.graphlets** package introduces four graphlet based ERG modeling terms into the **ergm** package for R. These model terms are summarized as follows:

1. *Graphlet counts* – `graphletCount(g)`*:*

   Statistics for the number of times that a graphlet appears in a network can be included in an ERGM by using the `graphletCount` term. The question that the change score function of this term answers is: how does the number of graphlets of type G_i change when an edge is flipped in the network? This term has an optional argument, g. g is a vector of distinct integers representing the list of graphlets to be evaluated during the estimation of model coefficients (see Figure 2 for the list of graphlets). When this argument is not provided, all graphlets are evaluated by default. The term adds one network statistic to the model for each element in g. This term is defined for the 30 graphlets with up to 5 nodes. Therefore, g accepts values between 0 and 29.

   The `graphletCount` term shows similarity with some terms of the **ergm** package, e.g., `cycle`, `edges`, `kstar`, `threepath`, `triangle`, `twopath`. The major difference between these existing **ergm** terms and the `graphletCount` term is that the existing terms consider arbitrary subgraphs, while `graphletCount` enforces the subgraphs to be induced. For example, `graphletCount` does not count the two-path subgraphs in a three node subgraph forming a triangle, while the `twopath` term counts three different two paths in a triangle subgraph. A closer parallel is the `triadcensus` term, which counts induced subgraphs on three nodes; note, however, that the triad census includes all isomorphism classes of order 3, while the order 3 graphlets consist only of the classes corresponding to connected graphs. Thus, while there is overlap between some quantities computed by `graphletCount` and some existing **ergm** terms, the two are on the whole distinct.

2. *Graphlet orbit covariance* – `grorbitCov(attrname, grorbit)`*:*

   The correlation between a node's graphlet degree and a numeric attribute value can be included into an ERGM by using the `grorbitCov` term. The question that the change score function of this term answers is: what is the change in covariance between a vector of node attributes and graphlet degrees (for a given orbit) when an edge is changed? This term has two arguments: `attrname` and `grorbit`. The `attrname` is a character vector giving the name of a numeric node attribute. The optional `grorbit` argument is a vector of distinct integers representing the list of graphlet orbits to include into the ERGM model (see Figure 2 for the list of graphlet orbits). When `grorbit` is not provided, all graphlet orbits are evaluated by default. The term adds one network statistic to the model for each element in `grorbit`. Each term is equal to the sum given in Equation 2:

   $$grorbitCov(G, i, X) = \sum_{v \in V} GD_i(G, v) * X_v, \tag{2}$$

   where X is the vector of node attribute values, i is the queried graphlet orbit and $GD_i(G, v)$ is the number of graphlets that touch node v at orbit i. This term is defined for the 73 orbits corresponding to graphlets with up to 5 nodes. Therefore, `grorbit` accepts values between 0 and 72.

   The `grorbitCov` term can be viewed as an extension of the the `nodecov` term in the **ergm** package to higher-order structures. In fact, the `nodecov` term is a special case of

`grorbitCov` where the `grorbit` argument is set to 0.

3. *Graphlet orbit factor* – `grorbitFactor(attrname, grorbit, base)`*:*

   The `grorbitFactor` term adds a relationship between graphlet degrees and a categorical node attribute into an ERGM. The question that the change score function of this term answers is: what is the change in the total graphlet degree (for a given orbit) for those nodes with a given attribute value, for a particular edge change? This term has three arguments: `attrname`; `grorbit`; and `base`. `attrname` is a character vector giving the name of a categorical node attribute. The optional `grorbit` argument is a vector of distinct integers representing the list of graphlet orbits to include into the model (see Figure 2 for the list of graphlet orbits). When `grorbit` is not provided, all graphlet orbits are evaluated by default. The optional `base` argument is a vector of distinct integers representing the list of categories in `attrname` that are going to be omitted. When this argument is set to 0, all categories are evaluated. Otherwise, the attribute values are sorted lexicographically and the attributes that are indexed by the `base` value(s) are omitted. For example, if the "fruit" attribute has values "orange", "apple", "banana" and "pear", `grorbitFactor("fruit", 0, 2:3)` will ignore the "banana" and "orange " factors and evaluate the "apple" and "pear" factors. When the `base` argument is not provided, the argument is set to 1 by default. The `grorbitFactor` term adds $a * |\texttt{grorbit}|$ terms into the model where a represents the number of attribute values that are evaluated in the model and $|\texttt{grorbit}|$ is the number of graphlet orbits to be evaluated in the model. Each term is equal to the sum in Equation 3:

$$grorbitFactor(G, i, X_c) = \sum_{v \in V, category(v) = X_c} GD_i(G, v), \tag{3}$$

   where X_c is the category of the term, i is the queried graphlet orbit, $category(v)$ is the category that node v belongs to, and $GD_i(G, v)$ is the number of graphlets that touch node v at graphlet orbit i. This term is defined for the 73 graphlet orbits corresponding to graphlets with up to 5 nodes. Therefore, `grorbit` accepts values between 0 and 72.

   The `grorbitFactor` term extends the `nodefactor` term in the **ergm** package. In fact, the `nodefactor` term is a special case of `grorbitFactor` where the `grorbit` argument is set to 0.

4. *Graphlet degree distribution* – `grorbitDist(grorbit, d)`*:*

   The graphlet degree distributions of different graphlet orbits can be included into the ERGM by using the `grorbitDist` term. The question that the change score function of this term answers is: how do the number of nodes having graphlet degree n for orbit i change when an edge is flipped? This term has two arguments: `grorbit` and `d`. The `grorbit` argument is a vector of distinct integers representing the list of graphlet orbits to include into the model (see Figure 2 for the list of graphlet orbits). The `d` argument is a vector of distinct integers. This terms adds one network statistic to the model for each pairwise combination of the arguments in `grorbit` and `d` vectors. The statistic for the combination of (i, j) is equal to the number of nodes in the network that have graphlet degree j for orbit i. This term is defined for the 15 graphlet orbits corresponding to graphlets with up to 4 nodes. Therefore, `grorbit` accepts values between 0 and 14. Graphlets of size 5 are omitted for this term because of the high computational complexity of the change score computation of the term.

The `grorbitDist` term extends the `degree` term in the **ergm** package. In fact, the `degree` term is a special case of `grorbitDist` where the `grorbit` argument is set to 0. However, the `grorbitDist` function does not support the filtering functionalities of the `degree` term that are defined with the `by` and `homophily` arguments.

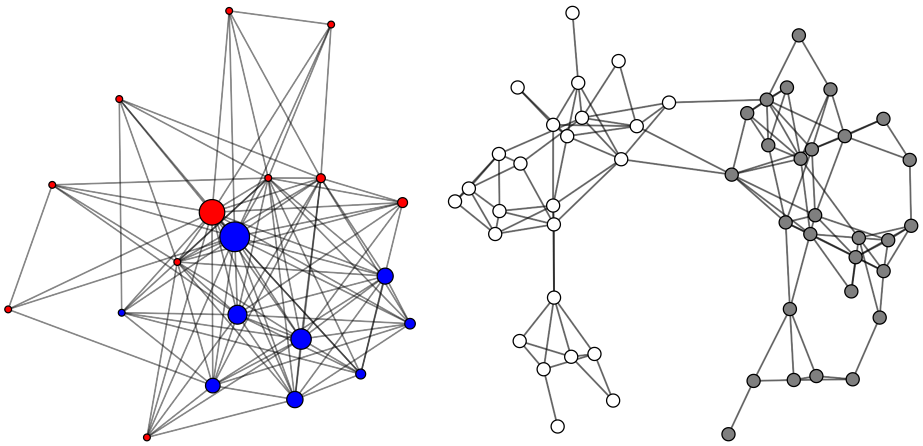
For detailed explanations and algorithmic details on the implementation of the new terms of the **ergm.graphlets** package, please refer to Appendix A.

4. Illustration: ERGMs with graphlet terms

In this section, we illustrate the use of terms from the **ergm.graphlets** package with two examples, one from the social sciences (Figure 4A) and one from the biological sciences (Figure 4B).

4.1. Lake Pomona emergent multi-organizational network (EMON)

Our first example comes from Drabek, Tamminga, Kilijanek, and Adams (1981)'s set of inter-organizational communication networks in the context of search and rescue operations. The setting for our example is the immediate aftermath of the capsizing of the Showboat Whippoorwill following its contact with a tornado near the southern shore of Lake Pomona, due south of Topeka, Kansas (Drabek *et al.* 1981). Sixty passengers and crew were stranded in the lake, prompting the immediate response of the twenty organizations whose communication ties compose our network. We use the `grOrbitFactor` and `grOrbitCov` terms of the **ergm.graphlets** package to analyze the patterns of brokerage (i.e., mediator nodes that bridge two nodes that are not directly connected as described by Gould and Fernandez 1989) in the organizational search and rescue network. Previous studies of brokerage have been limited



(A) (B)

Figure 4: (A) Lake Pomona emergent multi-organizational network (EMON) tasked with a search and rescue operation. Node size is scaled by command rank score and nodes are colored by whether they had permanent headquarters situated locally (red) or non-locally (blue). (B) Network representation of the protein structure of the two matriptase-BPTI complexes. Secondary structure elements are shaded by the complex to which they belong.

to the use of marginal tests to determine whether levels of brokerage exceed what we would expect by some baseline (Gould and Fernandez 1989; Marcum, Bevc, and Butts 2012; Lind, Tirado, Butts, and Petrescu-Prahova 2008; Spiro, Acton, and Butts 2013). The introduction of these graphlet terms enables us to examine brokerage using conditional tests in which we can identify entities' propensities to occupy brokerage roles independent of confounding factors such as degree.

Although Drabek's *emon* dataset is originally represented as a digraph, informants were asked to report on communication between organizations (without regard to directionality) and the relation is thus inherently undirected. We symmetrize the original *emon* network via union rule (Krackhardt 1987), treating a tie as present if an informant from either involved organization reports it. We include the command rank score, location and sponsorship node attributes of the original network with our undirected version. *Command rank score* is a rating of each organization's prominence in the network's chain of command, as reported by informants from all organizations participating in the search and rescue effort. When ranking those with the strongest position in the chain of command, informants were limited to the six organizations present from the early phase of the response. As a result, some organizations were not ranked and have been coded "NA" in the *emon* data. For our example, we assume those who were not ranked have the lowest possible command rank score (arriving later and being more marginal to the unfolding response) and assign them a score of 0. The *location* of each group's headquarters was also recorded; organizations were situated locally or non-locally in the Lake Pomona response. Finally, we include the *sponsorship level* of each organization: city, county, state, federal, or private. The resulting undirected network can be readily loaded from the **ergm.graphlets** package by typing:

```
R> data("emon3", package = "ergm.graphlets")
```

We illustrate the network in Figure 4A. Our network resembles a core-periphery structure with the core primarily composed of non-local organizations and organizations with high command rank scores.

In our ERGM model for the *emon* network, we begin with an `edge` term for the total number of edges (baseline density). We use dyadic independence terms (i.e., `nodefactor` and `nodecov`) for sponsorship level and command rank score. One might expect organizations at different sponsorship levels to be involved with more or fewer communication partnerships than organizations from a different sponsorship; likewise, an organization's command rank score may be associated with its propensity to be involved in more communication partnerships. Finally, we include terms related to graphlet structure. Graphlet G_6, which involves brokerage between a dyad and a pendant, is a natural choice given the core-periphery structure of the graph, and we include all its orbits (i.e., 9, 10, and 11) into our model. We incorporate the location covariate into the term to evaluate whether an organization's location is associated with its propensity to occupy these specific orbits. The results will demonstrate whether the location of an organization in this type of subgraph is associated with its role as a pendant (orbit 9), member of a dyad with ties to a broker (orbit 10), or broker between the pendant and the dyad (orbit 11). We model the network as shown below:

```
R> emon.ergm <- ergm(emon.3 ~ edges + nodefactor("Sponsorship") +
+    nodecov("Command.Rank.Score") + grorbitFactor("Location", 9:11),
+    control = control.ergm(seed = 1, MCMC.samplesize = 50000,
+    MCMC.interval = 100000, MCMC.burnin = 50000, parallel = 60))
```

```
Iteration 1 of at most 20:
Loading required package: rlecuyer
Convergence test P-value: 0e+00
The log-likelihood improved by 0.6982
Iteration 2 of at most 20:
Convergence test P-value: 0e+00
The log-likelihood improved by 0.1079
...
Iteration 9 of at most 20:
Convergence test P-value: 9e-01
Convergence detected. Stopping.
The log-likelihood improved by < 0.0001

This model was fit using MCMC.  To examine model diagnostics and check
for degeneracy, use the mcmc.diagnostics() function.
```

Before examining the coefficients we examine the MCMC diagnostics to ensure the estimation process did not exhibit any peculiar behavior (Hunter *et al.* 2008a). This model appears to have converged properly.

A summary of the model object reproduces the original formula for the model, the coefficients, deviance measures, and measures of the goodness of fit.

```
R> summary(emon.ergm)

==========================
Summary of model fit
==========================

Formula:   emon.3 ~ edges + nodefactor("Sponsorship") +
nodecov("Command.Rank.Score") + grorbitFactor("Location", c(9:11))

Iterations:  20

Monte Carlo MLE Results:
                               Estimate Std. Error MCMC %  p-value
edges                        -2.450670   0.688351      9 0.000473 ***
nodefactor.Sponsorship.County  -0.437354   0.319080      3 0.172175
nodefactor.Sponsorship.Federal -0.581708   0.606596      5 0.338852
nodefactor.Sponsorship.Private -0.041876   0.188267      1 0.824230
nodefactor.Sponsorship.State   -1.326516   0.785447      1 0.092967 .
nodecov.Command.Rank.Score      0.333315   0.075229      5  < 1e-04 ***
grorbitFactor.orb_9.attr_NL     0.009319   0.020540      0 0.650596
grorbitFactor.orb_10.attr_NL   -0.018051   0.014288      2 0.208081
grorbitFactor.orb_11.attr_NL    0.158800   0.031310      7  < 1e-04 ***
---

Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
    Null Deviance: 263.4  on 190  degrees of freedom
Residual Deviance: 144.8  on 181  degrees of freedom

AIC: 162.8    BIC: 192    (Smaller is better.)
```

The results show significant effects for our `edge` term, command rank score, and non-local organizations' occupation of orbit 11. The results show a strong, positive association between an organization's command rank score and its odds of forming a tie. Most relevant to our interests, we find that one of the automorphism orbit terms is significant. Specifically, we find a positive, significant association between an organization's being non-local (NL) and its propensity to occupy a brokerage role between a pendant and a dyad (orbit 11). Substantively, this demonstrates that non-local organizations tend to occupy this specific structure of extended brokerage in which an organization occupies a brokerage position between one organization and a pair of connected organizations. Interestingly, location is not related to occupancy of orbit 9 (a brokered pendant) or orbit 10 (a brokered cluster), which tells us that non-local organizations engaging in brokerage are not preferentially brokering between a local "core" and a non-local periphery. The role of the non-local organizations in brokerage for this response is thus richer than might be imagined at first blush.

We use the `gof` command to examine model adequacy. While the AIC and BIC demonstrate substantial improvements over a baseline model, the `gof` command measures demonstrate how well networks simulated from our model reproduce statistics from the original network. We examine the model's reproduction of four statistics: geodesic distance, degree distribution, edgewise shared partner distribution, and the triad census. We demonstrate below how we produce plots to examine these measures of fit.

```
R> EMONgof <- gof(emon.ergm, GOF = ~ degree + distance + espartners +
+    triadcensus)
R> par(mfrow = c(2, 2))
R> plot(EMONgof)
```

The plots are illustrated in Figure 5. As there are no clear discrepancies between the model-simulated networks and the original network, we find the model to be an adequate fit.

The graphlet orbit terms enable us to link local position to covariates in a model-based framework. As demonstrated, this is a useful tool for modeling brokerage, as we are able to link an entity's covariates to its propensity to occupy a specific brokerage role, whether it is a traditional (i.e., `twopath`) brokerage role or an extended brokerage role (e.g., orbit 11 in our model). Beyond brokerage, these techniques can extend to any particular automorphism orbit contained within a graphlet: pendants, clique members, or other nodes whose position may be linked to some categorical or continuous variable. Being able to incorporate these covariate-driven graphlet terms into a model-based framework will enhance our ability to understand which factors are associated with nodes' occupation of local positions within graphlets.

4.2. Protein secondary structure network

The past decade has seen a surge of interest in identifying network motifs (i.e., subgraphs that are overrepresented or underrepresented in a network, relative to chance; Milo, Shen-Orr,

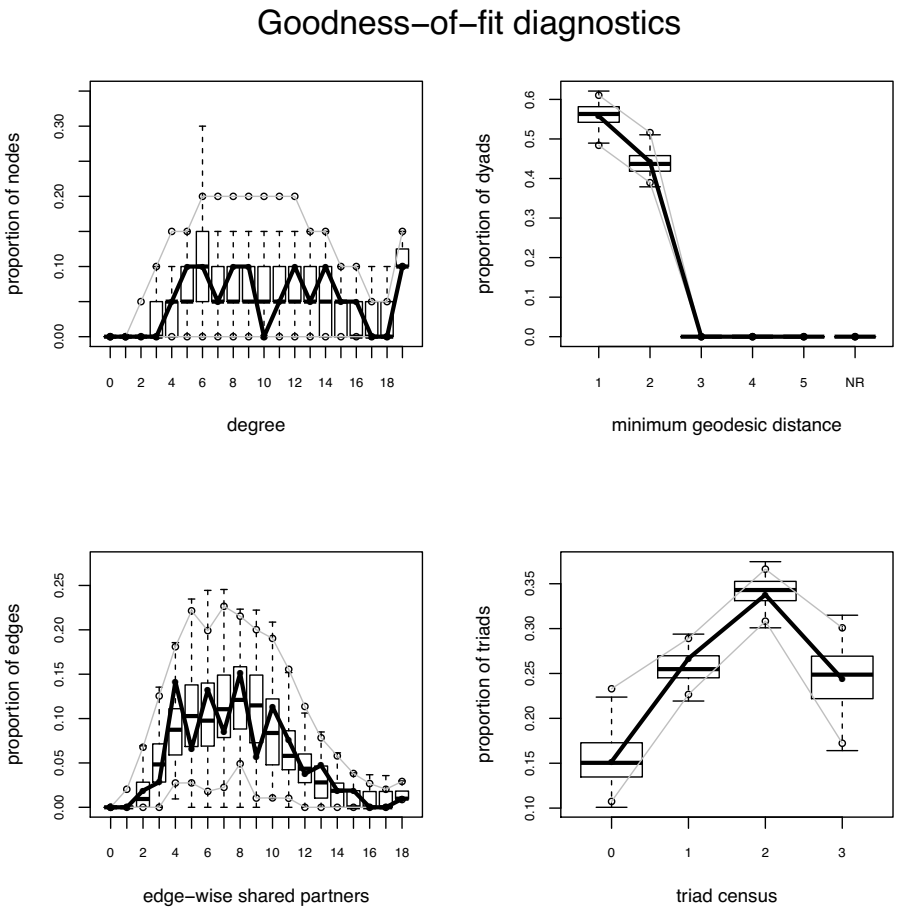


Figure 5: The solid black line in each plot represents the Lake Pomona EMON's observed statistics. The box plots illustrate the statistics for our simulated networks, as produced by the MLE.

Itzkovitz, Kashtan, Chklovskii, and Alon 2002; Milo *et al.* 2004). Typically, scholars have used marginal tests to identify how frequently these subgraphs occur relative to some baseline. In these types of tests the observed network is compared to a set of randomized networks that hold constant some statistic of the original network, often the degree distribution. While these types of marginal tests have been employed by networks scholars for decades (see, e.g., Wasserman and Faust 1994; Butts 2008, for reviews), a model-based approach allows us to examine the likelihood of observing these subgraphs, conditioned on a variety of parameters (e.g., degree, triadic closure, covariates, etc.). This is particularly important where the method of data collection itself may bias structure in particular ways; failure to account for these effects may result in spurious findings. We use the `graphletCount` terms to examine patterns of biological network motifs in an ERGM framework, while controlling for artifacts of the data collection process.

We analyse the the protein structure network of a matriptase-aprotinin complex (PDB ID: 1eaw; Friedrich *et al.* 2002) whose nodes are secondary structure elements (specifically, α helices and β sheets) which are "tied" if the distance between them is smaller than 10

Angstroms (Å) (Milo *et al.* 2004)[2]. Milo *et al.* (2004) examine the overrepresentation and underrepresentation of subgraphs in this network, by comparison to uniform random graphs conditional on the degree distribution. They find that subgraphs in the form of graphlets G_3 and G_4 are underrepresented while subgraphs in the form of G_6, G_7, and G_8 are overrepresented. We will determine whether these results hold in a model-based framework that allows us to account for potentially confounding degree, transitivity, and mixing effects, some of which represent artifacts of the data collection process.

Before modeling the protein structure network, it is important to consider how this network was obtained. Although Milo *et al.* (2004) do not report on the content of the structure[3], Friedrich *et al.* (2002) note that the asymmetric unit of the crystal structure (from which the network is constructed) contains two biological assemblies, each of which is a complex of two proteins (the catalytic domain of matriptase/MT-SP1 and a bovine pancreatic trypsin inhibitor/BPTI). The presence of multiple copies of a biologically relevant complex within a crystal structure is a common artifact of the crystallization process, and indeed the same system could potentially have been observed with more or fewer complexes in the asymmetric unit. This is of considerable importance for modeling the resulting network, as we would typically expect far more adjacencies within complexes than between them; failure to control for this effect may lead to very misleading conclusions. Indeed, as shown in Figure 4B, the network is dominated by two dense subgraphs corresponding to the two complexes, with very few ties spanning these subgraphs. To account for this, we create vertex attributes based on biological assembly membership as reconstructed from information in the Protein Data Bank (Friedrich *et al.* 2002), with polypeptide chains A and B of the structure belonging to assembly 1, and chains C and D belonging to assembly 2. By incorporating these attributes into the model, we are much better able to account for the patterns of clustering in the network than we would be if we neglected the data collection process. The protein structure network containing the assembly membership node attributes can be readily loaded by typing:

```
R> data("spi", package = "ergm.graphlets")
```

We begin by setting up our ERGM with an `edges` term, a dyadic independence term, and several dyadic dependence terms, including our graphlet terms. Because we observe very little tie formation across the sets of chains associated with each complex, we include a homophily term for protein assembly in our model. Additionally, we include a within-assembly triadic closure term (i.e., closure of triads where all members belong to the same assembly). We also include a degree term, as the original paper was concerned with graphlet counts net of the degree distribution. Of principal interest is our `graphletCount` term, which includes graphlets G_3, G_4, G_6, G_7, and G_8, the same set Milo *et al.* (2004) find to occur at greater or lesser levels than chance.

Our first model includes all terms described above. To speed up model fit, one may omit the "control" arguments, although the resulting standard errors (and accordingly, p values) will be larger than what we report.

[2]This protein structure network can be obtained from: http://www.weizmann.ac.il/mcb/UriAlon/Papers/networkMotifs/1eawInter_st.txt.

[3]The structure is not described in the paper, and is (inaccurately) summarized in the supplemental materials only as "a serine protease inhibitor" (Table S1). In fact, the structure contains two assemblies, each of which is a complex of one domain of a serine protease (MT-SP1) with an inhibitor (BPTI).

```
R> spi.ergm.34678 <- ergm(spi ~ edges + nodematch("Assembly") +
+    triangle("Assembly") + gwdegree(0.5, fixed = TRUE) +
+    graphletCount(c(3, 4, 6, 7, 8)), control = control.ergm(seed = 1,
+    MCMC.samplesize = 500000, MCMC.interval = 75000, MCMC.burnin = 300000,
+    parallel = 60))

Iteration 1 of at most 20:
Loading required package: rlecuyer
Convergence test P-value: 0e+00
The log-likelihood improved by 0.3676
Iteration 2 of at most 20:
Convergence test P-value: 0e+00
The log-likelihood improved by 0.06913
...
Iteration 10 of at most 20:
Convergence test P-value: 8.3e-01
Convergence detected. Stopping.
The log-likelihood improved by < 0.0001

This model was fit using MCMC.  To examine model diagnostics and
check for degeneracy, use the mcmc.diagnostics() function.

R> summary(spi.ergm.34678)

==========================
Summary of model fit
==========================

Formula:   spi ~ edges + nodematch("Assembly") + triangle("Assembly") +
gwdegree(0.5, fixed = T) + graphletCount(c(3, 4, 6, 7, 8))

Iterations:  20

Monte Carlo MLE Results:
                  Estimate Std. Error MCMC %  p-value
edges              -6.42760    1.22926     12  < 1e-04 ***
nodematch.Assembly  2.48031    0.74204      6 0.000852 ***
triangle.Assembly   3.87343    0.67331      1  < 1e-04 ***
gwdegree            2.40227    1.51019      5 0.111906
graphlet.3.Count    0.04962    0.02964      7 0.094298 .
graphlet.4.Count   -0.03917    0.05467      1 0.473841
graphlet.6.Count   -0.15361    0.04993      0 0.002137 **
graphlet.7.Count   -0.47295    0.17782      0 0.007910 **
graphlet.8.Count   -2.49869    0.72543      0 0.000590 ***
---

Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
     Null Deviance: 1910.3  on 1378  degrees of freedom
 Residual Deviance:  593.9  on 1369  degrees of freedom

AIC: 611.9    BIC: 658.9    (Smaller is better.)
```

Our model finds a significant, positive effect for within-assembly homophily, a positive effect for triadic closure within complexes, and a propensity for the graph to be biased against formation of graphlets G_6, G_7, and G_8, assuming all other terms are held constant. We find no significant results for graphlets G_3 and G_4.

We proceed to remove the non-significant terms to see if that improves model fit. AIC suffers slightly if we remove G_3 from the model (AIC: 612.97), while BIC improves (654.8). Both improve if we keep G_3 and remove G_4 (AIC: 610.73, BIC: 652.56). We find the best fit by removing both G_3 and G_4 (AIC: 610.7, BIC: 647.3). Accordingly, we fit our final model as follows.

```
R> spi.ergm.all <- ergm(spi ~ edges + nodematch("Assembly") +
+    triangle("Assembly") + gwdegree(0.5, fixed = TRUE) +
+    graphletCount(c(6, 7, 8)), control = control.ergm(seed = 1,
+    MCMC.samplesize = 15000, MCMC.interval = 2000, MCMC.burnin = 15000))

Iteration 1 of at most 20:
Convergence test P-value: 0e+00
The log-likelihood improved by 0.2026
Iteration 2 of at most 20:
Convergence test P-value: 0e+00
The log-likelihood improved by 0.05503
...
Iteration 8 of at most 20:
Convergence test P-value: 9.7e-01
Convergence detected. Stopping.
The log-likelihood improved by < 0.0001

This model was fit using MCMC.  To examine model diagnostics
and check for degeneracy, use the mcmc.diagnostics() function.

R> summary(spi.ergm.all)

==========================
Summary of model fit
==========================
Formula:   spi ~ edges + nodematch("Assembly") + triangle("Assembly") +
 gwdegree(0.5, fixed = T) + graphletCount(c(6, 7, 8))

Iterations:  20
```

```
Monte Carlo MLE Results:
                   Estimate Std. Error MCMC %  p-value
edges              -4.80106    0.73658      8  < 1e-04 ***
nodematch.Assembly  2.11636    0.66232      5 0.001428 **
triangle.Assembly   3.27864    0.53805      0  < 1e-04 ***
gwdegree            1.12902    1.21795      1 0.354095
graphlet.6.Count   -0.12037    0.04122      2 0.003560 **
graphlet.7.Count   -0.46225    0.16905      0 0.006330 **
graphlet.8.Count   -2.31074    0.68949      0 0.000826 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

     Null Deviance: 1910.3  on 1378  degrees of freedom
 Residual Deviance:  596.7  on 1371  degrees of freedom

AIC: 610.7    BIC: 647.3    (Smaller is better.)
```

Once again we find positive, significant effects for homophily within complexes and triadic closure within complexes. Controlling for this, we find negative, significant effects for graphlet terms G_6, G_7, and G_8.

Our final model appears to have converged without any notable issues (Hunter *et al.* 2008a). We now assess model adequacy. As Figure 6 indicates, our model closely approximates the observed network; our simulated networks show no clear deviations from the observed statistics on degree, geodesic distance, shared partners, or the triad census.

It is interesting to compare the results of our joint, multivariate analysis with the marginal tests conducted by Milo *et al.* (2004). Milo *et al.* (2004) find that the network overrepresents graphlets G_6, G_7, and G_8 and underrepresents G_3 and G_4. After controlling for other factors (particularly clustering within each complex), we find no evidence of additional underrepresentation or overrepresentation of G_3 or G_4; further, we actually find that the network appears biased *against* formation of graphlets G_6, G_7, and G_8, once other terms are accounted for. The discrepancy here is due to the use of marginal tests by Milo *et al.* (2004). To determine whether a graphlet occurs more or less often relative to chance, they compare the number of observed graphlets to the number observed in a set of random graphs conditioned on the degree distribution (a form of *conditional uniform graph test*). For this protein structure network, such random graphs bear little resemblance to the data in question (Figure 7), and in particular do not include effects related to the fact that the structure is a composite of two distinct complexes. While this does not make the results of such tests wrong per se, it does render them unable to distinguish between structural biases arising from simple features arising from the data collection process, and those arising from more subtle and informative biochemical mechanisms. The marginal approach is also unable to unravel the joint influence of multiple biases simultaneously; because graphlet structures are dependent upon one another, over- or underrepresentation of multiple graphlets (relative to a uniform baseline) may actually be the result of biases to a smaller number of features. Such complexities are difficult to unravel using marginal tests, and are more flexibly handled via the ERGM framework.

Our analysis underscores the fact that one can obtain misleading conclusions when trying to use marginal tests to assess graphlet counts, particularly when the baseline distribution being

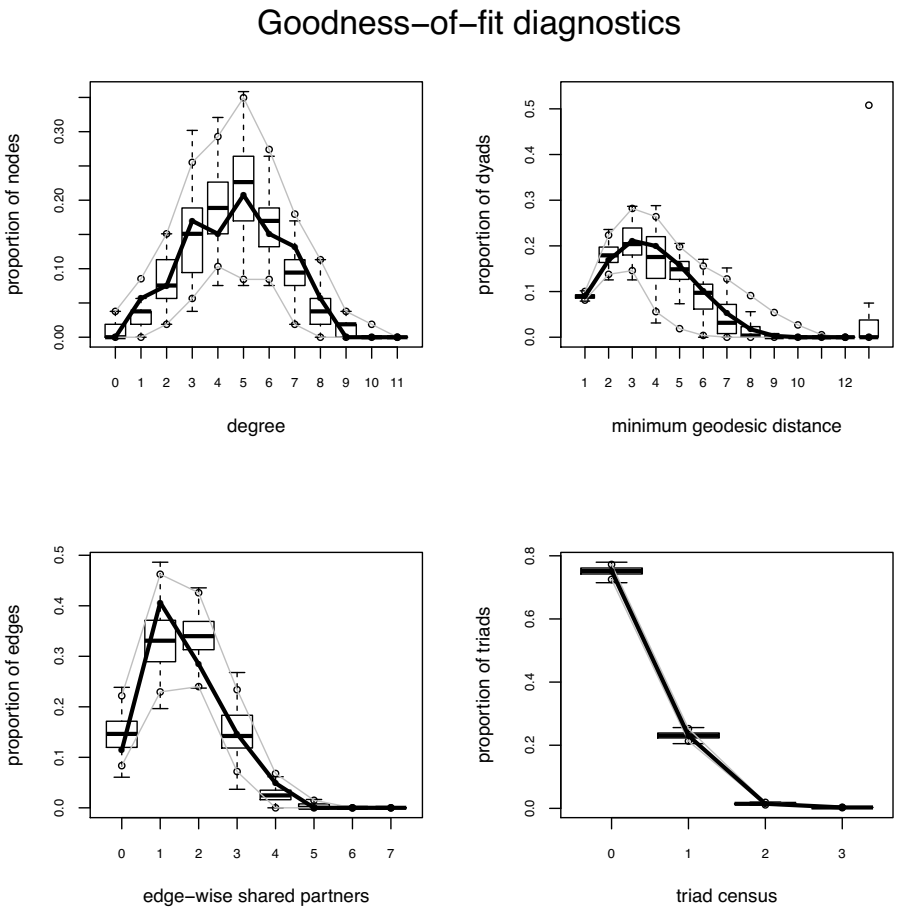


Figure 6: The solid black line in each plot represents the protein network's observed statistics. The box plots illustrate the statistics for our simulated networks, as produced by the MLE.

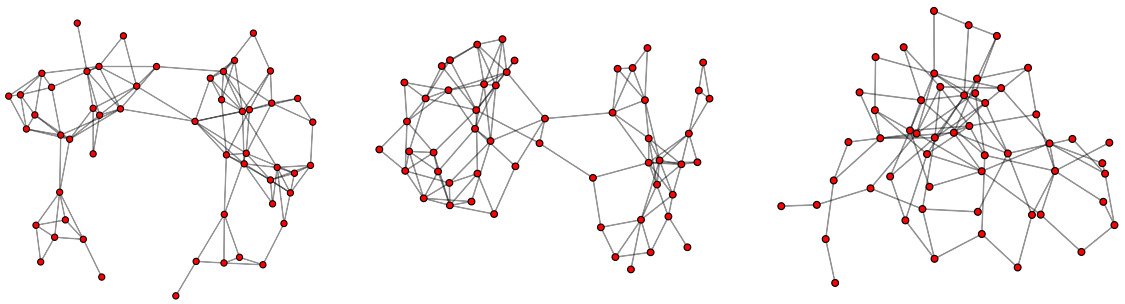


Figure 7: Observed protein network (left), typical protein network simulated by our final model (middle), and typical random graph produced by holding constant the observed network's degree distribution (right).

employed does not incorporate extremely basic features of the system being studied. While inference for complex, highly dependent systems is difficult under the best of conditions, the generative nature of the ERGM framework allows us to assess the adequacy of our models by comparison to features of the original data; given that we have identified a model that is both sensible and that successfully regenerates the important properties of the observed network, we have a stronger basis for subsequent investigation than would be obtained from simple rejection of a null hypothesis.

By using an ERGM approach and incorporating our graphlet terms, we are able to produce more sophisticated models of protein networks that include not only network motifs but also other important biological and/or chemical properties of the system in question. Scholars in a variety of biological sub-disciplines have begun to use ERGMs to model many different types of networks, including protein-protein interaction networks (Bulashevska, Bulashevska, and Eils 2010; Clark, Dannenfelser, Tan, Komosinski, and Ma'ayan 2012), neural networks (Hinne, Heskes, Beckmann, and van Gerven 2013; Simpson, Hayasaka, and Laurienti 2011; Simpson, Moussa, and Laurienti 2012), and metabolic networks (Saul and Filkov 2007). Introducing the tools from the **ergm.graphlets** package to the network community should enhance the field's ability to model graphlet counts in the context of network motifs or any other application where one is interested in counts of small, undirected, induced subgraphs.

5. Discussion

The **ergm.graphlets** package introduces four new terms into the **ergm** package which enable ERG modeling using the graphlet properties of a network. The `graphletCount` term enables defining ERGMs based on the number of graphlets in the network. `grorbitCov` term uses the relation between a numeric node attribute with a specific structural feature in order to introduce node attribute relations into a model. The `grorbitFactor` term is similar to the `grorbitCov` term except that it relates categorical node attributes with graphlet degrees. The `grorbitDist` term uses the graphlet degree distribution for ERG modeling. The `graphletCount`, `grorbitCov` and `grorbitFactor` terms are defined for graphlets with 2, 3, 4 and 5 nodes. Because of the computational complexity issues, `grorbitDist` is not defined for 5 node graphlets.

Model degeneracy, instability, and sensitivity are currently important challenges for modeling within the ERGM framework (Handcock 2003; Schweinberger 2011). For some combinations of model terms, the MCMC procedure may fail to converge within a reasonable number of iterations: this is generally because the graph distribution associated with the specified model family is ill-behaved. Like most dependence terms, the terms in the **ergm.graphlets** package sometimes suffer from these instability issues, depending on the modeled network and the other terms in the ERGM. Typically, degeneracy problems are currently handled either by using user-selected terms whose effects partially cancel (e.g., using sparse graphlets and complete graphlets together) or using curved exponential family models (Hunter and Handcock 2006; Butts 2011; Schweinberger 2011) that systematically combine large numbers of terms in a manner that balances their total effect. The former technique requires having an intuition about the structure of the data and a number of trials with different combinations of terms under this intuition. It can be hard to identify the best terms for generating an ERGM model and there is currently no general solution that works well in all settings. Our experience suggests that graphlet terms for which the change score is non-zero for most of

the steps in the MCMC procedure are good terms to start the modeling process with. For example, it is not reasonable to model a sparse network using dense graphlets, as the change score will be 0 for most of the MCMC steps. In this respect, the graphlet terms that are expected to be overrepresented in the network can also be good candidate terms to start ERG modeling. Using terms of the same graphlet size together usually improves the convergence of the MCMC process, since smaller graphlets might already be contained in a number of larger graphlets and this causes dependency issues among model terms. We have also observed that the MCMC procedure converges faster when graphlets containing closed-loop structures (e.g., triangles, cycles) are excluded from the model definition: This is mainly because of the instability of these terms, as explained in Schweinberger (2011). As more data sets are subjected to analysis using ERGMs (and models with graphlet terms in particular), better heuristics are likely to emerge.

Past work with partial (i.e., non-induced) subgraph terms has suggested that curved exponential family models can also be used for improving degeneracy issues. In curved exponential families, the parameters associated with model statistics are constrained to lie on a non-linear surface of reduced dimension, forcing them to remain in a fixed relationship with one another; this can be helpful when dealing with intrinsically correlated graph statistics, as very precise weighting may be needed to avoid the degenerate regime. Examples of curved terms include the `gwdegree`, `gwdsp`, and `gwesp` terms of the **ergm** package, as well as the closely related *alternating k-star* and *alternating path* statistics of Snijders, Pattison, Robins, and Handcock (2006). Because graphlet statistics do not "nest" in the same way as partial subgraph statistics, they may benefit from novel formal development. On the other hand, some ideas used in existing curved families – e.g., geometrically weighted degree distributions – could potentially be applied to graphlet degrees in a relatively straightforward manner. This would seem to be a promising direction for future research.

When the over- or underrepresentation of a specific graphlet statistic is of particular interest but inclusion of this statistic into one's model proves difficult, another alternative is the use of a simplified model omitting the statistic as a reference distribution against which to test the observed graphlet statistic. Specifically, let t' be the statistic of interest, and let t be the vector of statistics in the best-fitting model without t'. A test of the hypothesis that the parameter θ' associated with the joint model $(t' \cup t, \theta' \cup \theta)$ is non-zero can be conducted by examining the quantiles of $t'(y)$ in the distribution of $t(Y)$, where $Y \sim \mathrm{ERG}(\hat{\theta}, t)$ and $\hat{\theta}$ is the MLE of θ given y. This approach (which was one motivation for the original development of ERGMs) is described in more detail by Holland and Leinhardt (1981).

Although we have tried to minimize the complexity of the change score computation, there is still room for improving the graphlet counting process. We apply a brute-force algorithm, which tries to minimize the number of computations: this gives an exact solution. Further gains in efficiency may be possible. These improvements would enable the implementation of `grorbitDist` for graphlets with 5 nodes. The model coefficients for terms related with larger graphlets would also be estimated more quickly with these improvements.

In addition to their inferential value, we note that the terms in the **ergm.graphlets** package can be used for evaluating the goodness-of-fit of an ERGM model estimate based on other (non-graphlet terms). When a model (with or without graphlet related terms) is estimated, the quality of this model in explaining the structure of the data in terms of graphlet properties of the network can be assessed by simulating new networks from the model and using the `summary` function to compute the graphlet counts and graphlet degree distributions. The

graphlet properties of the network can be compared with these simulation results to evaluate whether the structure of the network fits to the structure described by the model. An example that describes how this test can be performed is explained in Goodreau, Handcock, Hunter, Butts, and Morris (2008).

In conclusion, the **ergm.graphlets** package extends the functionality of the **ergm** package by incorporating graphlet statistics. The new terms are of particular utility when modeling processes such as brokerage, functional mediation, or other phenomena that depend not only on the edges that are present within a graph, but also on those that are absent. Such processes are common in both social and biological systems, and the ability to capture them is an important goal of modern network analysis.

Acknowledgments

We thank Rachel Martin for her valuable input on protein structures, and Kai Sun and Miles Mulholland for their helpful suggestions and comments regarding the manuscript. The project was supported by ERC Starting Independent Researcher Grant 278212, NSF CDI OIA – 1028394 grant, ARRS project J1-5454, and the Serbian Ministry of Education and Science Project III44006.

References

Barabási AL, Albert R (1999). "Emergence of Scaling in Random Networks." *Science*, **286**(5439), 509–512.

Bulashevska S, Bulashevska A, Eils R (2010). "Bayesian Statistical Modelling of Human Protein Interaction Network Incorporating Protein Disorder Information." *BMC Bioinformatics*, **11**(46).

Butts CT (2008). "Social Network Analysis: A Methodological Introduction." *Asian Journal of Social Psychology*, **11**(1), 13–41.

Butts CT (2011). "Bernoulli Graph Bounds for General Random Graphs." *Sociological Methodology*, **41**(1), 299–345.

Clark NR, Dannenfelser R, Tan CM, Komosinski ME, Ma'ayan A (2012). "Sets2Networks: Network Inference from Repeated Observations of Sets." *BMC Systems Biology*, **6**(89).

Drabek TE, Tamminga HL, Kilijanek TS, Adams CR (1981). *Managing Multiorganizational Emergency Responses: Emergent Search and Rescue Networks in Natural Disaster and Remote Area Settings*. University of Colorado Intitute of Behavioral Science, Boulder, CO.

Erdös P, Rényi A (1959). "On Random Graphs." *Publicationes Mathematicae*, **6**, 290–297.

Frank O, Strauss D (1986). "Markov Graphs." *Journal of the American Statistical Association*, **81**(395), 832–842.

Friedrich R, Fuentes-Prior P, Ong E, Coombs G, Hunter M, Oehler R, Pierson D, Gonzalez R, Huber R, Bode W, Madison EL (2002). "Catalytic Domain Structures of MT-SP1/Matriptase, A Matrix-Degrading Transmembrane Serine Proteinase." *The Journal of Biological Chemistry*, **277**(2), 2160–2168.

Goodreau SM, Handcock MS, Hunter DR, Butts CT, Morris M (2008). "A **statnet** Tutorial." *Journal of Statistical Software*, **24**(9), 1–26. URL http://www.jstatsoft.org/v24/i09/.

Gould RV, Fernandez RM (1989). "Structure of Mediation: A Formal Approach to Brokerage in Exchange Networks." *Sociological Methodology*, **19**, 89–126.

Handcock MS (2003). "Assessing Degeneracy in Statistical Models of Social Networks." *Working Paper 39*, Center for Statistics and the Social Sciences, University of Washington, Seattle. URL http://www.csss.washington.edu/Papers/wp39.pdf.

Handcock MS, Hunter DR, Butts CT, Goodreau SM, Krivitsky PN, Morris M (2014). ***ergm**: Fit, Simulate and Diagnose Exponential-Family Models for Networks*. The Statnet Project (http://www.statnet.org/). R package version 3.2-4, URL http://CRAN.R-project.org/package=ergm.

Handcock MS, Hunter DR, Butts CT, Goodreau SM, Morris M (2003). ***statnet**: Software Tools for the Statistical Modeling of Network Data*. The Statnet Project (http://www.statnet.org/). R package version 2014.2.0, URL http://CRAN.R-project.org/package=statnet.

Handcock MS, Hunter DR, Butts CT, Goodreau SM, Morris M (2013). ***ergm.userterms**: User-Specified Terms for the **statnet** Suite of Packages*. The Statnet Project (http://www.statnet.org/). R package version 3.1.1, URL http://CRAN.R-project.org/package=ergm.userterms.

Hinne M, Heskes T, Beckmann CF, van Gerven MAJ (2013). "Bayesian Inference of Structural Brain Networks." *NeuroImage*, **66**, 543–552.

Holland PW, Leinhardt S (1981). "An Exponential Family of Probability Distributions for Directed Graphs." *Journal of the American Statistical Association*, **76**(373), 33–65.

Hunter DR, Goodreau SM, Handcock MS (2008a). "Goodness of Fit of Social Network Models." *Journal of the American Statistical Association*, **103**(481), 248–258.

Hunter DR, Handcock MS (2006). "Inference in Curved Exponential Family Models for Networks." *Journal of Computational and Graphical Statistics*, **15**(3), 565–583.

Hunter DR, Handcock MS, Butts CT, Goodreau SM, Morris M (2008b). "**ergm**: A Package to Fit, Simulate and Diagnose Exponential-Family Models for Networks." *Journal of Statistical Software*, **24**(3), 1–29. URL http://www.jstatsoft.org/v24/i03/.

Kolaczyk ED (2009). *Statistical Analysis of Network Data*. 1st edition. Springer-Verlag, Boston.

Krackhardt D (1987). "Cognitive Social Structures." *Social Networks*, **9**(2), 109–134.

Krivitsky PN (2012). "Exponential-Family Random Graph Models for Valued Networks." *Electronic Journal of Statistics*, **6**, 1100–1127.

Lind BE, Tirado M, Butts CT, Petrescu-Prahova M (2008). "Brokerage Role in Disaster Response: Organisational Mediation in the Wake of Hurricane Katrina." *International Journal of Emergency Management*, **5**(1/2), 75–99.

Marcum CS, Bevc CA, Butts CT (2012). "Mechanisms of Control in Emergent Interorganizational Networks." *The Policy Studies Journal*, **40**(3), 516–546.

Milenković T, Pržulj N (2008). "Uncovering Biological Network Function via Graphlet Degree Signatures." *Cancer Informatics*, **6**, 257–273.

Milo R, Itzkovitz S, Kashtan N, Levitt R, Shen-Orr S, Ayzenshtat I, Sheffer M, Alon U (2004). "Superfamilies of Evolved and Designed Networks." *Science*, **303**(5663), 1538–1542.

Milo R, Shen-Orr S, Itzkovitz S, Kashtan N, Chklovskii D, Alon U (2002). "Network Motifs: Simple Building Blocks of Complex Networks." *Science*, **298**(5594), 824–827.

Morris M, Handcock MS, Hunter DR (2008). "Specification of Exponential-Family Random Graph Models: Terms and Computational Aspects." *Journal of Statistical Software*, **24**(4), 1–24. URL http://www.jstatsoft.org/v24/i04/.

Newman M (2010). *Networks: An Introduction*. Oxford University Press.

Pattison P, Wasserman S (1999). "Logit Models and Logistic Regressions for Social Networks: II. Multivariate Relations." *British Journal of Mathematical and Statistical Psychology*, **52**(2), 169–193.

Penrose M (2003). *Random Geometric Graphs*. Oxford University Press, Oxford.

Pržulj N (2007). "Biological Network Comparison Using Graphlet Degree Distribution." *Bioinformatics*, **23**(2), 177–183.

Pržulj N, Corneil DG, Jurisica I (2004). "Modeling Interactome: Scale-Free or Geometric?" *Bioinformatics*, **20**(18), 3508–3515.

Pržulj N, Higham DJ (2006). "Modeling Protein-Protein Interaction Networks via a Stickiness Index." *Journal of the Royal Society Interface*, **3**(10), 711–716.

R Core Team (2015). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

Robins G, Pattison P, Kalish Y, Lusher D (2007). "An Introduction to Exponential Random Graph (p*) Models for Social Networks." *Social Networks*, **29**(2), 173–191.

Saul ZM, Filkov V (2007). "Exploring Biological Network Structure Using Exponential Random Graph Models." *Bioinformatics*, **23**(19), 2604–2611.

Schweinberger M (2011). "Instability, Sensitivity, and Degeneracy of Discrete Exponential Families." *Journal of the American Statistical Association*, **106**(496), 1361–1370.

Simpson SL, Hayasaka S, Laurienti PJ (2011). "Exponential Random Graph Modeling for Complex Brain Networks." *PLOS One*, **5**(6), e20039.

Simpson SL, Moussa MN, Laurienti PJ (2012). "An Exponential Random Graph Modeling Approach to Creating Group-Based Representative Whole-Brain Connectivity Networks." *NeuroImage*, **60**(2), 1117–1126.

Snijders TAB, Pattison PE, Robins GL, Handcock MS (2006). "New Specifications for Exponential Random Graph Models." *Sociological Methodology*, **36**(1), 99–153.

Solava RW, Michaels RP, Milenkovič T (2012). "Graphlet-Based Edge Clustering Reveals Pathogen-Interacting Proteins." *Bioinformatics*, **28**(18), i480–i486.

Spiro ES, Acton RM, Butts CT (2013). "Extended Structures of Mediation: Re-Examining Brokerage in Dynamic Networks." *Social Networks*, **35**, 130–143.

Wasserman S, Faust K (1994). *Social Network Analysis: Methods and Applications*. Cambridge University Press, Cambridge.

Wasserman S, Pattison P (1996). "Logit Models and Logistic Regressions for Social Networks: I. An Introduction to Markov Graphs and p*." *Psychometrika*, **61**(3), 401–425.

Yaveroglu ON, Fitzhugh SM, Kurant M, Markopoulou A, Przulj N, Butts CT (2015). ***ergm.graphlets**: A Package for ERG Modeling Based on Graphlet Properties*. R package version 1.0.3, URL http://CRAN.R-project.org/package=ergm.graphlets.

A. Algorithms and implementation

The terms in the **ergm.graphlets** package are implemented using the **ergm.userterms** package (Handcock *et al.* 2013). The **ergm.userterms** package enables users to introduce new model terms into the **ergm** package by implementing C code which calculates the change statistics of the new term. For the **ergm.graphlets** package, the change score function should answer the question: how do the graphlet counts in the network and graphlet degrees of the nodes change when an edge is flipped in the network? This question can be answered efficiently by *touching* the graphlets on the flipped edge and counting only the graphlets that are going to be affected by the edge flip. For this purpose, we identify all *edge automorphism orbits* in graphlets with 2, 3, 4 and 5 nodes. The 69 different edge automorphism orbits are in Figure 8 (Solava, Michaels, and Milenkovič 2012). In this section, we use *node orbits* for graphlet orbits that are provided in Figure 2 and *edge orbits* for edge automorphism orbits in Figure 8 for clarity.

We apply a brute-force search algorithm for computing the change score for graphlet terms. For each flipped edge, the edge orbits that are related with the queried graphlet are mapped on the flipped edge and the neighborhood of that edge is searched for nodes that complete the graphlet. For each node combination that completes the graphlet, the count of the affected graphlets is incremented by one. For identifying the change in the count of a specific graphlet, the computation is performed only for relevant edge orbits. The relations among graphlets and edge orbits are summarized in Table 1. For example, the change score for the counts of graphlet G_{11} and G_{12} can be calculated by counting $E_{19}, E_{20}, E_{21}, E_{22}, E_{27}, E_{36}, E_{40}, E_{48}$. After counting these edge orbits, the change score for G_{11} is equal to $(E_{19} - E_{27})$ and the change score for G_{12} is equal to $(E_{20} + E_{21} + E_{22} - E_{36} - E_{40} - E_{48})$ where E_x represents the number of graphlets counted by placing edge orbit x on the flipped edge. By counting the graphlet change scores based on edge orbits, we do not only restrict the counting process to graphlets that are affected from the edge flip, but also avoid repeated counting of the same edge orbit for different graphlet counts. For instance, E_3 affects the count of G_2 positively and the count of G_1 negatively. With our implementation, the number of graphlets affected by E_3 is counted only once, and this change score is computed for identifying the changes in the counts of both G_1 and G_2. The edge orbit based counting procedure is applied for computing the change scores for all the terms in the **ergm.graphlets** package.

The computational complexity of this approach is dependent on the average degree (and therefore the density) of the modelled network. The *average degree* of a network is defined as the average number of ties that a node has in the network. The *density* of a network is defined as $\frac{|E|}{\binom{|N|}{2}}$ where $|E|$ is the number of edges and $|N|$ is the number of nodes in the graph. In the average case, the computational complexity of the change counting procedure is $O(d^2)$ where d represents the average degree of a node. The worst case scenario occurs when searching for graphlet G_9 in a clique. In this case, the computational complexity of the function is $O(n^3)$ where n is the number of nodes in the network. But this situation occurs very rarely as most real-world networks are sparse.

The four terms in the **ergm.graphlets** package are all implemented using edge orbits. However, the computation of the change scores differ slightly from each other depending on the way that the graphlet counts contribute to change statistics for these terms. The computation of the four terms in the **ergm.graphlets** package are explained as follows:

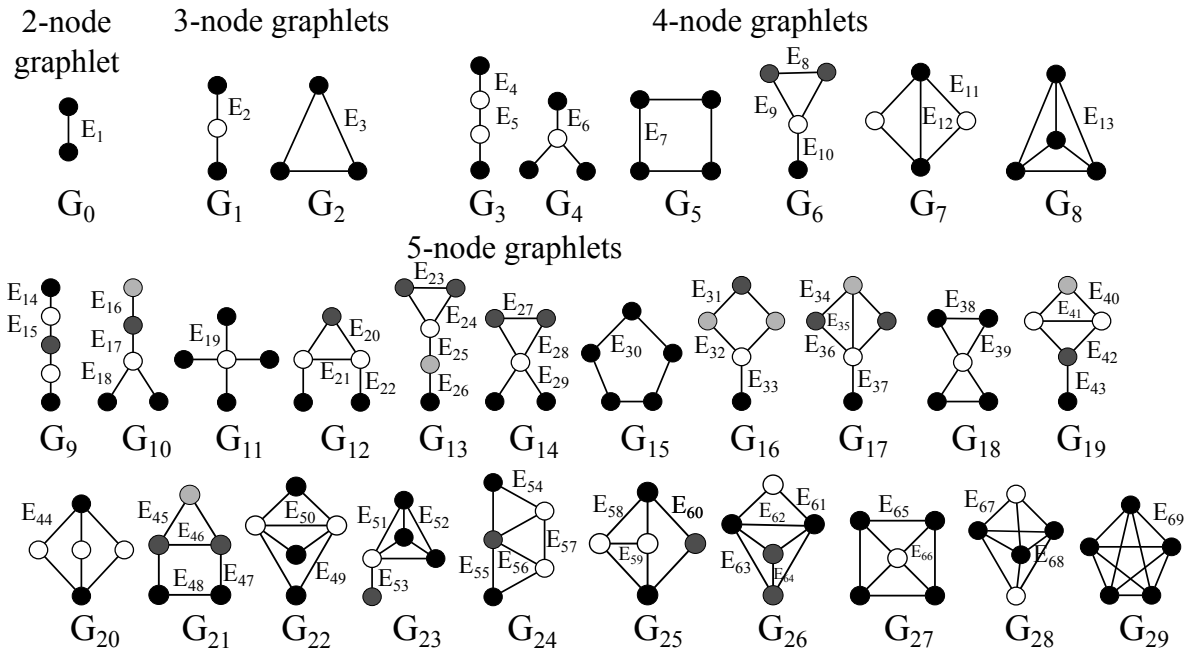


Figure 8: The edge automorphism orbits in 2-, 3-, 4- and 5-node graphlets. Adapted from Solava *et al.* (2012).

	Edge automorphism			Edge automorphism	
Graphlet	Positive	Negative	Graphlet	Positive	Negative
$\mathbf{G_0}$	E_1	–	$\mathbf{G_{15}}$	E_{30}	E_{46}
$\mathbf{G_1}$	E_2	E_3	$\mathbf{G_{16}}$	E_{31}, E_{32}, E_{33}	E_{35}, E_{41}, E_{44}, E_{45}
$\mathbf{G_2}$	E_3	–	$\mathbf{G_{17}}$	E_{34}, E_{35}, E_{36}, E_{37}	E_{49}, E_{52}, E_{54}
$\mathbf{G_3}$	E_4, E_5	E_7, E_9	$\mathbf{G_{18}}$	E_{38}, E_{39}	E_{57}
$\mathbf{G_4}$	E_6	E_8	$\mathbf{G_{19}}$	E_{40}, E_{41}, E_{42}, E_{43}	E_{51}, E_{55}, E_{60}
$\mathbf{G_5}$	E_7	E_{12}	$\mathbf{G_{20}}$	E_{44}	E_{50}, E_{59}
$\mathbf{G_6}$	E_8, E_9, E_{10}	E_{11}	$\mathbf{G_{21}}$	E_{45}, E_{46}, E_{47}, E_{48}	E_{56}, E_{58}
$\mathbf{G_7}$	E_{11}, E_{12}	E_{13}	$\mathbf{G_{22}}$	E_{49}, E_{50}	E_{64}
$\mathbf{G_8}$	E_{13}	–	$\mathbf{G_{23}}$	E_{51}, E_{52}, E_{53}	E_{61}
$\mathbf{G_9}$	E_{14}, E_{15}	E_{21}, E_{24}, E_{30}, E_{32}	$\mathbf{G_{24}}$	E_{54}, E_{55}, E_{56}, E_{57}	E_{63}, E_{65}
$\mathbf{G_{10}}$	E_{16}, E_{17}, E_{18}	E_{20}, E_{23}, E_{28}, E_{31}	$\mathbf{G_{25}}$	E_{58}, E_{59}, E_{60}	E_{62}, E_{66}
$\mathbf{G_{11}}$	E_{19}	E_{27}	$\mathbf{G_{26}}$	E_{61}, E_{62}, E_{63}, E_{64}	E_{67}
$\mathbf{G_{12}}$	E_{20}, E_{21}, E_{22}	E_{36}, E_{40}, E_{48}	$\mathbf{G_{27}}$	E_{65}, E_{66}	E_{68}
$\mathbf{G_{13}}$	E_{23}, E_{24}, E_{25}, E_{26}	E_{39}, E_{42}, E_{47}	$\mathbf{G_{28}}$	E_{67}, E_{68}	E_{69}
$\mathbf{G_{14}}$	E_{27}, E_{28}, E_{29}	E_{34}, E_{38}	$\mathbf{G_{29}}$	E_{69}	–

Table 1: The relations between graphlet types and edge automorphism orbits. The "Positive" columns list the edge automorphism orbits that increase the graphlet count, and the "Negative" columns list the edge automorphism orbits that decrease the graphlet count when an edge is added.

1. `graphletCount(g)`: The counting procedure is based on identification of graphlets. Therefore, each identified graphlet directly increments (or decrements) the change score for the related graphlet by 1. The change score for this term is computed by counting all edge orbits that are associated with the graphlets provided in argument g. When all required edge orbits are counted, these counts are summed to get the overall change in

the number of graphlets. For example, the change score for graphlet G_{12} is equal to the summation of $(E_{20} + E_{21} + E_{22} - E_{36} - E_{40} - E_{48})$ where E_x represents the number of graphlets that touch the flipped edge on edge orbit x.

2. `grorbitCov(attrname, grorbit)`: This term relates a numeric node attribute with the graphlet degrees of the nodes according to Equation 2 as explained in Section 3. The change score of this term depends on the graphlet degrees. Therefore, for each identified graphlet, the nodes of this graphlet are associated with the node orbits that they correspond to. Let us say that a graphlet of type G_4 is identified for the subgraph of nodes a, b, c, d, when the edge (a, b) is added into network. The identified subgraph is in Figure 9. Then the change score for node orbit 6 is incremented by $X_b + X_c + X_d$, and the change score for node orbit 7 is incremented by X_a, where X is the attribute vector keeping the attribute values for all nodes. The same logic applies when an edge is removed from the network. The final change score is obtained by summing these values for all edge orbits that are related with the graphlet that the query node orbit belongs to.

3. `grorbitFactor(attrname, grorbit, base)`: This term relates a categorical attribute with the graphlet degrees of the nodes according to Equation 3 as explained in Section 3. The change score of this term depends on the graphlet degrees. When the flip of an edge affects a node orbit, the change score that relates the category of the affected node with the node orbit is incremented (or decremented) by 1. Let us say a graphlet of type G_4 is identified for the subgraph of nodes a, b, c, d, when an edge (a, b) is added into the network. The identified subgraph is in Figure 9. Nodes a and b belong to "Category 1", c and d belong to "Category 2". In this scenario, the change score for "Node Orbit 7, Category 1" and "Node Orbit 6, Category 1" will increase by 1 with the contribution of nodes a and b. The change score for "Node Orbit 6, Category 2" will increase by 2 because of the nodes c and d. The same logic applies when an edge is removed from the network. The final change score is obtained by summing these values for all edge orbits that are related with the graphlet that the query node orbit belongs to.

4. `grorbitDist(grorbit, d)`: This term identifies the change in the graphlet degree distribution of a node orbit when an edge is flipped during the MCMC process, as explained in Section 3. The change score computation for this term is slightly different from the other terms, as graphlet degrees for all nodes in the network are required for the computation. In order to reduce the computational complexity of the problem, we compute the graphlet signatures of all nodes at the beginning of the MCMC procedure. During the execution of the MCMC procedure, we update these signatures using the change scores. The computation of the changes in graphlet degrees of the nodes is performed similar to the algorithm applied for the other terms. However, as graphlets can convert to each other with the addition or removal of edges, the counting procedure should be applied for all edge orbits. Therefore, it is not possible to restrict the counting procedure to edge orbits that are related with the query node orbits. For these reasons, the computational complexity of this term is higher than the other terms in the **ergm.graphlets** package. We implement the `grorbitDist` term only for graphlets with 2, 3, and 4 nodes, because of the high computational complexity of the computation of change score for graphlets with 5 nodes.

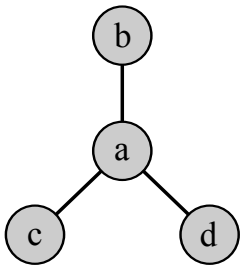


Figure 9: The subgraph that is used for illustrating the change statistics computation for the `grorbitCov` and `grorbitFac` terms.

We first test the correctness of the terms in the **ergm.graphlets** by using the `summary` function. The `summary` function computes the statistics for the provided terms. If the change score function is implemented correctly, the `summary` function produces the actual value of the term statistic for the provided network. In this respect, we validated the correctness of the `graphletCount` term by running:

```
R> summary(ntwk ~ graphletCount)
```

The output of the `summary` function was exactly the same as the graphlet counts produced by the graphlet counting implementation of Pržulj (2007). Evaluating the correctness of the `grorbitCov` term is slightly different from `graphletCount` as it is related with a node attribute value. To test the correctness of this term, we first created a dummy node attribute that is named "dummy". This node attribute has value 1 for all nodes in the network. We validated the correctness of `gorbitCov` by running:

```
R> summary(ntwk ~ grorbitCov("dummy"))
```

The output of the `summary` function was exactly the same with the sum of the graphlet degrees of all nodes for all node orbits. We repeated this test with weighted attribute values, e.g., when all attribute values are set to 2. The correctness of the results is also validated for this case. The validation for the `grorbitFactor` term is similar to the `grorbitCov` term. We assigned the same value for the category attribute, named "dummy", of all nodes and called the `summary` function as:

```
R> summary(ntwk ~ grorbitFactor("dummy", 0:72, 0))
```

This call correctly returns the sum of graphlet degrees of all terms. When the category value is changed to another value, the output of the call does not change. Finally, for validating the `grorbitDist` term, we called the `summary` function as:

```
R> summary(ntwk ~ grorbitDist(0:14, 0:10))
```

This call produces the correct graphlet degree distributions for all node orbits as validated in comparison with the output of the implementation of Pržulj (2007).

We also validated the correctness of our implementation by performing simulations on ERGMs that contain graphlet terms. In these tests, we defined ERGMs containing an `edge` term and a

graphlet term. We manually set the model coefficient for the graphlet related term to various positive and negative values. We simulated 30 networks from each of these ERGMs. With these simulations, we validated that positive coefficients promote the count of the related graphlet in the simulated networks. The count of related graphlet increases up to a certain coefficient value. After this threshold, the simulated networks contain the maximum possible number of related graphlets in the simulated networks. Similarly, negative coefficients have an effect of suppressing the appearance of the graphlet in the simulated networks. As the coefficient value gets closer to 0, the suppressing effect disappears. The range where the graphlet counts increase with the coefficient depends on the coefficients of the other terms in the ERGM.

Affiliation:

Nataša Pržulj
Department of Computing
Imperial College London
180 Queen's Gate
LONDON, SW7 2AZ, United Kingdom
E-mail: natasha@doc.ic.ac.uk
URL: http://www.doc.ic.ac.uk/~natasha/
Telephone: +44/207/594-8287
Fax: +44/207/594-8932

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 65, Issue 12 *Submitted:* 2013-09-18
June 2015 *Accepted:* 2014-10-17