

RESEARCH ARTICLE

Generating global network structures by triad types

Marjan Cugmas*, Anuška Ferligoj, Aleš Žiberna

Faculty of Social Sciences, University of Ljubljana, Ljubljana, Slovenia

* marjan.cugmas@fdv.uni-lj.si



Abstract

This paper addresses the question of whether one can generate networks with a given global structure (defined by selected blockmodels, i.e., cohesive, core-periphery, hierarchical, and transitivity), considering only different types of triads. Two methods are used to generate networks: (i) the newly proposed method of relocating links; and (ii) the Monte Carlo Multi Chain algorithm implemented in the `ergm` package in R. Most of the selected blockmodel types can be generated by considering all types of triads. The selection of only a subset of triads can improve the generated networks' blockmodel structure. Yet, in the case of a hierarchical blockmodel without complete blocks on the diagonal, additional local structures are needed to achieve the desired global structure of generated networks. This shows that blockmodels can emerge based only on local processes that do not take attributes into account.

OPEN ACCESS

Citation: Cugmas M, Ferligoj A, Žiberna A (2018) Generating global network structures by triad types. PLoS ONE 13(5): e0197514. <https://doi.org/10.1371/journal.pone.0197514>

Editor: Floriana Gargiulo, Centre National de la Recherche Scientifique, FRANCE

Received: October 25, 2017

Accepted: May 3, 2018

Published: May 30, 2018

Copyright: © 2018 Cugmas et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: All relevant data are within the paper and its Supporting Information files.

Funding: The project was financially supported by the Slovenian Research Agency (www.ars.gov.si) within the "Young researchers" program, the research program P5-0168 and under the research project J7-8279 (Blockmodeling multilevel and temporal networks).

Competing interests: The authors have declared that no competing interests exist.

Introduction

In both social network analysis and other scientific fields, considerable attention is paid to global network structures, which can be described using a blockmodel [1]. Development of the subject area of blockmodeling was initially based on the work of social theories in the last century [2–4]. Later, blockmodeling has been used in different scientific fields [5–11].

A blockmodel consists of clusters (also called positions) of units and the relationships between those clusters. The units are assigned to the same cluster if they are equivalent according to the pattern of links to the other units. Often, structural equivalence is assumed. Two units are structurally equivalent if they are linked to the same units and by the same others [12], implying they share the same social role [13, 14]. There are several well-known and studied types of blockmodels, e.g., cohesive, core-periphery, transitivity and hierarchical [1] which are described in more detail in the next section.

In order to describe the processes that produce a given global structure, much effort was made to study micro structures in the context of various global structures. For this propose, the triadic census (the collection of all possible networks of size three which are visualized in Fig 1), proposed by Davis [15], is often considered [15–19]. While the triadic census is well studied in the context of different global network structures, no attention was given to the dependencies between the triadic census and different global network structures,

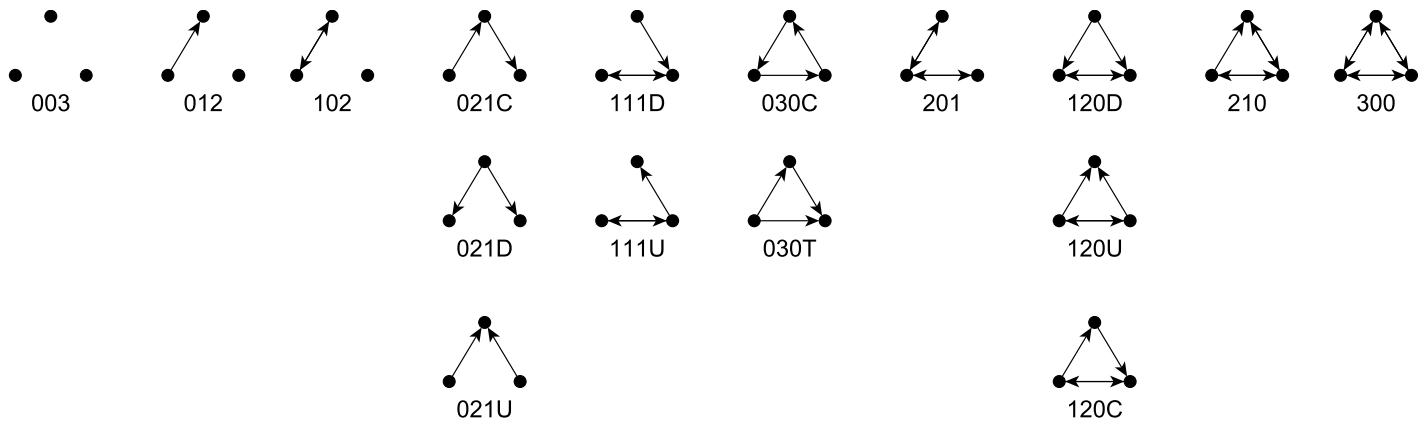


Fig 1. The collection of all triad types (triad census). The labels consist of three digits: the first digit denotes the number of mutual links (\leftrightarrow), the second stands for the number of arcs (\rightarrow) while the third denotes the number of missing links between two units. Some types of triads with the same distribution of links are further differentiated (see columns) and labeled with a letter (C stands for cycle, T for transitivity, U for up and D for down).

<https://doi.org/10.1371/journal.pone.0197514.g001>

operationalized by the types of blockmodels. This is especially important to consider when thinking about the factors that drive a network to a certain global structure in terms of social mechanisms [20].

Therefore, the main objective of this study is to test whether it is possible to generate networks with a given blockmodel structure (seven selected blockmodel types are considered), taking only different types of triads into account. The main objective is further elaborated: is it possible to generate networks with a given blockmodel type while considering only allowed or only forbidden types of triads? The classification of allowed and forbidden types of triads is determined for each type of blockmodel separately. Allowed types of triads are those whose frequency is higher than zero in an ideal blockmodel structure (see section Global network structure for the definition of an ideal blockmodel structure). On the other hand, forbidden triad types are those whose frequency equals zero in an ideal blockmodel. The sets of allowed and forbidden triad types are then reduced based on comparisons of the different types of blockmodels for different levels of errors in the network according to the ideal blockmodel being considered.

The sets of all triad types, the sets of allowed and forbidden triad types and the sets of reduced (called selected) allowed and forbidden triad types are then used to generate networks with a given blockmodel structure. For a blockmodel type which cannot be generated successfully based only on the types of triads, some other local network structures are considered.

Beside the different types of triads, other subgraph types of a size higher than three can be used to generate networks with a given blockmodel. Milo et al. [21] confirmed that different motifs are common in different empirical networks, where motifs are defined as “patterns of interconnections occurring in complex networks at numbers that are significantly higher than those in randomized networks”. Different types of triads, rather than motifs, are considered chiefly because they are the smallest sociological unit from which the dynamic of a multi-person relationship can be observed [17].

Various types of algorithms can be applied to generate networks with a given blockmodel, considering only different triad types. In this study, the Relocating Links algorithm (RL algorithm) and the Monte Carlo Multi Chain algorithm (MCMC algorithm) are used. If the generated structures with the selected set of triads, obtained using both algorithms, are very similar and close to the assumed ideal structure, one may conclude it is possible to generate networks with the assumed blockmodel structure by only considering the selected types of triads. On the

other hand, if the generated networks are not similar and in line with the assumed blockmodel structure, one may speculate whether this is a consequence of the specifics of the algorithms or that the set of selected local structures is insufficient to generate this specific blockmodel.

In this study, it is assumed that the assignment of a unit to a cluster is unknown. Considering the information on the cluster assignment would require a different methodological approach. It is also assumed that the units' characteristics are not known. Kogut [22] reported that a certain structure's emergence in a network is often the consequence of rules that generate self-organization dynamics. These rules do not need to be technological in origin, but can also reflect institutional or cultural norms and are also deeply embedded in the social identity of the units, meaning they are often invisible or unknown when examining an empirical social network.

Moreover, the study does not address the question of how the specific selected social mechanisms affect the emergence of given blockmodel types but instead examines the field in terms of the presence of different social mechanisms for when can recognize a given global network structure (blockmodel) in an empirical network. For example, when a given global network structure is highly characterized by a very high number of transitive triads (or, when a given blockmodel type can emerge due to units' tending towards the creation of transitive links), one can discuss several social mechanisms which are related to transitive triads. In this regard, one must note that different social mechanisms can lead to a specific social output and a specific social mechanism can lead to different social outputs [23]. Moreover, the social mechanisms are very context-specific, which is why the study does not focus on specific social mechanisms in light of the selected blockmodel types.

This paper is organized in the following way: first, the global network structures are described in terms of blockmodels and then the local network structures (namely different triad types) are presented. In this context, the sets of allowed and forbidden triad types and the sets of selected allowed and selected forbidden triad types are proposed for each blockmodel type. Considering these, random networks with the chosen blockmodels are generated using the proposed RL algorithm and the MCMC algorithm. Generating a hierarchical blockmodel without complete blocks on the diagonal is further discussed in a separate section (*Improvement of the hierarchical blockmodel without complete blocks on the diagonal*) while the section *Concluding remarks about generating networks with triads* briefly summarizes the ability to generate networks with a given blockmodel considering only different triad types. Limitations of the study and further research ideas are presented in the Conclusion.

Global network structures

Here, global network structures are defined by blockmodels. A blockmodel is a reduced network in which the units are clusters (positions) of units from the network. The term block refers to the link between two clusters (positions) and represents the block in the adjacency matrix when units are grouped by clusters. The links in the blockmodel represent relationships between the positions [1]. Using the blockmodeling procedure, a blockmodel can be derived from a given empirical network. Blockmodeling is an approach for reducing a large, potentially incoherent network to a smaller, comprehensible and interpretable structure [1]. It can entail either a direct or indirect approach.

Several blockmodeling approaches have been developed to establish the best blockmodel structure according to the given network and equivalence [1]. In this study, pre-specified direct blockmodeling (generalized blockmodeling) was used. Here, the blockmodeling procedure is a local optimization procedure. The solution is optimized with a relocation algorithm which minimizes the value of the criterion function [24–26] by relocating units from one

cluster into another. Compared to indirect blockmodeling, direct blockmodeling produces a solution with a lower criterion function value. Further, in the case of direct blockmodeling, the risk of a local optimum exists and therefore the algorithm must be repeated several times in the hope of obtaining the global optimum, and its computational complexity is high when a larger number of units is analyzed. Community detection methods are similar to blockmodeling approaches when a cohesive blockmodel is pre-specified (see [27]).

There are several well-known and studied blockmodel types, e.g., cohesive, core-periphery, hierarchical and transitivity [1, 28]. Even though all these structures have often been studied, different definitions of them exist [29]. In this study, the definitions of blockmodel structures are taken from Doreian et al. [1].

Structural equivalence [12, 30] is considered in all blockmodels. It was shown [1] that in the case of structural equivalence only complete and null blocks exist. In ideal complete blocks, all possible links are present while in ideal null blocks no link exists. The left matrix in Fig 2 represents a blockmodel with three clusters (positions). There are only complete and null blocks where structural equivalence is completely satisfied. Such a blockmodel is a pre-specified blockmodel and called an “ideal blockmodel”. In the blockmodel in Fig 2A, the diagonal blocks are complete and all the others are null blocks. This type of blockmodel is called a cohesive blockmodel. Usually, the empirical blockmodels being considered are not completely consistent with the selected equivalency and errors exist when comparing such a blockmodel with an ideal one. An error exists when a link is present in a null block or there is a non-link in a complete block. Fig 2B–2D illustrates a cohesive blockmodel with different levels of errors. A level of errors *LE* is defined on a scale between 0 and 1, where 0 corresponds to an ideal blockmodel and 1 corresponds to a totally randomized network with the same density as in the ideal blockmodel. The level of errors increases linearly as links are moved from complete blocks to null blocks, until the densities of both block types are the same (the level of errors then equals to 1). In such a network, it is not possible to distinguish between blocks. The number of links *k* needed to be randomly relocated in order to achieve a given level of errors in the network can be calculated as

$$k = m - \left(\frac{m^2}{n^2 - n} \right) * LE \tag{1}$$

where *m* is the number of links and *n* is the number of units in a selected type of blockmodel (see S1 Appendix for more details).

The following types of ideal blockmodels are defined and considered:

- **Cohesive blockmodel** is visualized in Fig 3A. With this blockmodel, several internally highly connected clusters of units (positions) are present. The units from different clusters are not

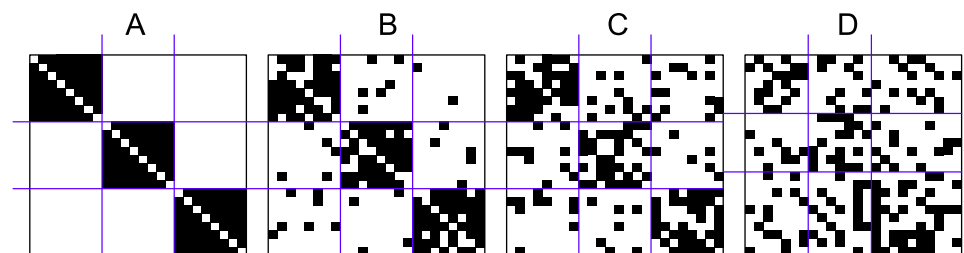


Fig 2. Cohesive blockmodel with different level of errors. (A) level of errors is 0 (ideal blockmodel), (B) level of errors is 0.25, (C) level of errors is 0.50, (D) level of errors is 1 (random network).

<https://doi.org/10.1371/journal.pone.0197514.g002>

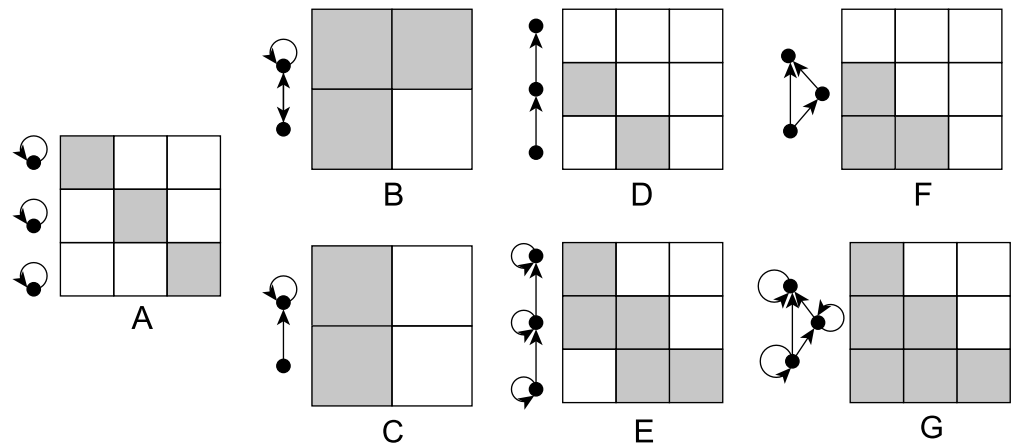


Fig 3. Different types of blockmodel structures: Graph (left) and matrix (right) representation. (A) cohesive, (B) symmetric core-periphery, (C) asymmetric core-periphery, (D) hierarchical without complete blocks on the diagonal, (E) hierarchical with complete blocks on the diagonal, (F) transitive without complete blocks on the diagonal, (G) transitive with complete blocks on the diagonal; gray colored cells denote complete blocks while white colored cells denote null blocks.

<https://doi.org/10.1371/journal.pone.0197514.g003>

linked to each other. This is a very basic network structure type and was also studied, e.g. in the context of the structural organization of the brain [31].

This is also (approximately in some cases) the global structure found by community detection methods (see [27]).

- The most common **core-periphery blockmodel** consists of one cluster of units which are highly internally linked to each other. Peripheral units which are not linked to each other are also assumed in this type of network. The core-periphery blockmodel is called symmetric when the links between the peripheral and core units are mutual (Fig 3B) and asymmetric when only the peripheral units are linked to the core ones (Fig 3C) or when only the core units are linked to the peripheral ones. The core-periphery blockmodel structure lies in the middle of several extreme network properties, e.g. clique vs. star configurations, network assortativity vs. network disassortativity, hierarchy vs. non-hierarchy, etc. [32]. Yet, for the degree dispersion, the core-periphery model is the extreme model [33].

The core-periphery model is often associated with the existence of elites. An elite cluster is a small cluster of units that are all linked to each other (core). Compared to peripheral units, core units have greater prestige, usually defined by a higher number of incoming links (higher in-degree).

A clear core-periphery blockmodel was found among high school students where a link between students exists if the first student asked the second one to lend their study notes [34]. It was also found when studying individual creative performances in the Hollywood film industry [35], in the analysis of metabolic networks [36], and in many studies of scientific co-authorships [37–39].

- A **hierarchical blockmodel** consists of several clusters of units which can be ordered into a hierarchy based on the direction of the links between the clusters. The units inside the clusters can be either linked to each other (Fig 3D) or not (Fig 3E). A hierarchical structure is often associated with companies' organizational structure [40, 41].
- A **transitivity blockmodel** (Fig 3F and 3G) is similar to a hierarchical model. The only difference is that units from the clusters on the lowest level are linked to all clusters on the

upper levels. This results in many transitivity relations (a relation R on a set A is called transitive if, for any $a, b, c \in A$ the conditions aRb and bRc imply aRc) which are very frequent when networks are formed among humans [42, 43] and animals [44–46]. In the literature, both hierarchical and transitive global network structures are often called hierarchical.

Algorithms for generating networks

Different statistical models have been developed [47] to explain the impact of local mechanisms on global network structures or to characterize the global network structures in terms of local network structures. Many of these models capture different global network characteristics such as a specific distribution of in-degree or out-degree, the clustering coefficient or the small-world effect and less the specific global configuration of links in the network (e.g. in terms of the blockmodel structure) [48]. Blockmodels provide a very detailed description of the global network structure, which is especially important in the context of the social roles which follow from the units' position in the blockmodel.

To generate networks with a given blockmodel by considering different types of triads, two similar algorithms are used: the RL algorithm and the MCMC algorithm implemented in the `ergm` package implemented in R [49]. They both share the assumption that the units tend to create such a constellation of links that would result in a desirable distribution of subgraphs of size three or other characteristics in the network. Following the distinction between network evolution models (NEM), network attribute models (NAM) and ERGM [47], the RL algorithm can be classified to the NEM category.

The latter are primarily used to study how a specific rule (or set of rules) about creating and dissolving links affects the global network structure, which is especially important in the theory of social mechanisms [23, 50]. On the other hand, the ERGM is used to check to what extent the global network structure can be explained when considering the structure of links and/or characteristics of the units. It can also be used to generate networks based on estimated or fixed parameter values. Both approaches are described and compared in more detail in the following sections.

Generating networks with the Relocating Links algorithm (RL algorithm)

The RL algorithm (see Algorithm 1) assumes that all considered local network statistics for the case of an ideal network are represented by the vector \mathfrak{T} . The number of elements g of this vector equals the number of local network statistics considered. The numbers of different types of triads are considered here, but some other local network statistics could also be chosen. The distribution of all or only a subset of all triad types can be given (for forbidden triad types, corresponding values of \mathfrak{T} equal zero). Beside \mathfrak{T} , the initial random network Y_r has to be given. Before the iterative procedure starts, the Y_r is saved as a new network Y_{new} .

The iterative procedure is repeated many times. Upon each iteration, a pair of linked units i and j and a pair of unlinked units k and l are randomly chosen. Then, the link between i and j is dissolved and the link between k and l is established. The modified network is saved as the proposed network Y_p . From Y_p , the number of each triad type considered \mathfrak{T}_p is calculated. The proposed network is saved as Y_{new} (the new network) if the CR ratio is greater than 1. The CR ratio is defined as

$$CR = \frac{\sum_{i=1}^g ((\mathfrak{T}_p - \mathfrak{T})_i)^2}{\sum_{i=1}^g ((\mathfrak{T}_{new} - \mathfrak{T})_i)^2} \tag{2}$$

Then, the new iteration is performed and, after many iterations, the last Y_{new} is the final solution. Besides the Y_{new} , the values of CR can be saved and further analyzed.

Algorithm 1 The Relocating Links algorithm

Require: $\mathfrak{I} \triangleright \mathfrak{I}$ denotes the distribution of local network statistics in an ideal network
Require: $Y_r \triangleright Y_r$ denotes a random network
Require: $M \triangleright M$ denotes the number of iterations

```

1:  $Y_{new} \leftarrow Y_r$ 
2:  $Y_p \leftarrow Y_r$ 
3: for  $m$  in 1:  $M$  do
4:   randomly select a tie  $y_{i,j}$  in  $Y_{new}$ 
5:   randomly select a non-tie  $y_{k,l}$  in  $Y_{new}$ 
6:   transform a tie  $y_{i,j}$  to a non-tie in  $Y_p$ 
7:   transform a non-tie  $y_{k,l}$  to a tie in  $Y_p$ 
8:   if  $CR > 1$  then  $\triangleright CR$  is defined in Eq 2
9:      $Y_{new} \leftarrow Y_p$ 
10:  else
11:     $Y_p \leftarrow Y_{new}$ 
12:  end if
13: end for
14: return  $Y_{new}$ 

```

Compared to the MCMC algorithm introduced in the next section, the RL algorithm is deterministic since a link is only allocated if the distribution of the triads of the proposed network is closer to the distribution of the triads in the case of an ideal blockmodel. This may result in lower variability of the global network structure of generated networks when the RL is used since, compared to the MCMC algorithm, RL strive to generate networks with the exact number of the selected types of triads. However, the risk of a local optimum exists which could be avoided by further improving the algorithm. Moreover, RL is computationally very intensive: as will be illustrated later, a higher number of iterations is required, especially in the case of denser networks.

Generating networks with the MCMC algorithm

To describe how the networks were generated using the MCMC algorithm, Exponential Random Graph Modelling (ERGM) has to be defined. Let us consider a random network Y (y is a given empirical network) consisting of N units. Here, the link between the i -th and the j -th unit can be represented by a random variable Y_{ij} , while the set of all possible random networks of this size is denoted by \mathcal{Y} . The distribution of Y can be written as

$$P_{\theta,\mathcal{Y}}(Y = y) = \frac{\exp\{\theta^T g(y)\}}{\kappa(\theta, \mathcal{Y})} \tag{3}$$

where $y \in \mathcal{Y}$. Here, θ is a vector of coefficients while $g(y)$ is the vector of statistics obtained for y . The normalization constant $\kappa(\theta, \mathcal{Y})$ in the denominator is needed to ensure the sum of probabilities equals 1.

Different methods can be used [51–54] to estimate the parameters θ . After that, one can generate random networks based on the model obtained. To do this, several types of MCMC algorithms have been proposed.

The simplest version of the algorithm that is based on Gibbs sampling is described below. Generally, the start is represented by a network in \mathcal{Y} . Then, based on a uniform distribution, one of the links or non-links is chosen. According to the model, the probability of establishing or dissolving a link is calculated and then, based on this probability, the chosen link or non-link is established or dissolved. The probability (y_{ij}^c denotes the complement of y_{ij}), y_{ij}^+ is the

network with a link between i and j ($y_{ij} = 1$) and y_{ij}^- is a network without a link between i and j ($y_{ij} = 0$)

$$\text{logit}[P_{\theta, \mathcal{Y}}(Y_{ij} = 1 | Y_{ij}^c = y_{ij}^c)] = \theta^T (g(y_{ij}^+) - g(y_{ij}^-)) \tag{4}$$

is calculated by the change in values of the estimated statistics before and after the change in the link between i and j . The iterative process stops when approximate convergence to $P_{\theta, \mathcal{Y}}(Y = y)$ is reached [49].

In this study a more general Metropolis-Hastings algorithm, implemented in the `ergm` package was used. The benefit of this more general approach is that by selecting suitable proposal distribution one can place suitable restrictions on the network, e.g. fixed density. The algorithm takes a proposed network $y_{proposed}$ from an auxiliary distribution depending on a current network $y_{current}$ with probability

$$\min \left\{ 1, \frac{P_{\theta, \mathcal{Y}}(Y = y_{proposed}) q(y_{current}, y_{proposed})}{P_{\theta, \mathcal{Y}}(Y = y_{current}) q(y_{proposed}, y_{current})} \right\} \tag{5}$$

Normalizing constant disappears from the ratio of ERGM probabilities (Eq 5) and the ratio is simplified

$$\frac{P_{\theta, \mathcal{Y}}(Y = y_{proposed})}{P_{\theta, \mathcal{Y}}(Y = y_{current})} = \exp\{\theta_0 [g(y_{proposed}) - g(y_{current})]\} \tag{6}$$

The definition of the probability in Eq 5 is similar to the definition of the CR in Eq 2: both compare the proposed network with the current one through the values of the proposed statistics. The elements of \mathfrak{T} are the exact values from the network with the ideal global network structure (where the number of units plays a significant role) while the values of θ are regression coefficients and are, therefore, less directly related to the global network structure. In the case of the RL algorithm, the link is relocated (i.e., one link is dissolved and one is established) always when CR is greater than 1 and never when it is below 1.

As described, the method most often used to estimate the parameters θ is MCMC-MLE which can be computationally hard to estimate. In our study, the parameters can be estimated based on networks with a given blockmodel without or with only very low levels of errors. Using this approach, the estimation algorithm does not converge in many cases, probably due to the high level of multicollinearity of the triads. In addition, from the end-user point of view, estimating the values of all parameters for each blockmodel type would be very difficult.

Instead, the values of the ERGM parameters θ are arbitrarily set to 2 (allowed) or -2 (forbidden). It has been clearly shown that some triad types are much more likely to appear in an ideal network (compared to a random network). By setting all the parameters' values to 2 or -2, we essentially assume that all types of allowed triads have the same importance (and similar for all forbidden triad types). Such a setting is critical when all types of triads are included in the model and result in a relatively unstable model, particularly when the density is not fixed.

All types of triads are generated considering the number of links fixed (to the same value as in ideal networks) on one hand, and free (with the density being the variable) on the other. With the latter, the value of parameter `edge` is set to such a value that the mean density of 30 generated networks lies within the ideal-density interval ± 0.05 .

Choosing triads for different types of ideal blockmodels

When generating networks with a specific type of a blockmodel (according to different triad types), all triad types or only a subset of all of them can be considered. Considering only a

subset of all possible triad types is particularly important when generating networks with the RL algorithm. This is because the distribution of triads must be known in advance for each type of ideal blockmodel separately. Here, it should be pointed out that the distributions of triads can vary among the same type of blockmodel with a different number of positions.

Since the number of different triads is also affected by the network density [55], the value of the A-measure can be used to select a smaller number of different triad types (see [S1 Appendix](#)) needed to generate the networks with a selected blockmodel. The A-measure is defined as the ratio between the absolute number of a certain type of triad in an ideal blockmodel and the mean number of such triads in a totally randomized network of the same density—see [S2 Appendix](#) for more information about generating totally randomized networks and networks with a given level of errors.

The classifications of allowed and forbidden triad types for different blockmodel types are presented in the next section followed by the classifications of selected allowed and selected forbidden triad types, based on values of the A-measure.

Allowed and forbidden triad types

The triad types can be classified into the set of allowed or into the set of forbidden triad types for each blockmodel type, based on the counts of triad types in an ideal blockmodel. Triad types with the count equal to zero are said to be forbidden in a given blockmodel and are thus classified into the set of forbidden triad types (for a given blockmodel). All the other triad types are classified into the set of allowed triad types. This classification is essential for the MCMC algorithm as it determines the values of the appropriate parameters in the ERGM model (see the previous section).

A more detailed insight into how common a certain triad type is for a certain blockmodel can be obtained by interpreting the A-measure values. The A-measure allows the relative number of triads to be compared within a certain type of blockmodel and also the relative number of triads between different types of blockmodels. In order to obtain values of the A-measure, 10,000 totally randomized ($LE = 1$) networks for each ideal blockmodel type were generated (see [S2 Appendix](#)). The A-measure values are presented in [Table 1](#). Values greater than 1 indicate triad types that are more likely to occur in an ideal network structure than would be expected in randomized networks. Such are complete subgraphs of size three (a triad of type 300) in a cohesive blockmodel.

When the A-measure value is close to 1, the number of triads in the case of an ideal network structure is close to the number of triads in totally randomized networks. This could indicate their occurrence is mainly a consequence of the density rather than the type of blockmodel. The A-measure values in the cells without any number in [Table 1](#) equal zero and therefore denote forbidden triads. It can be seen that the values of the A-measure of certain triad types exceed zero in some, but not all, blockmodel types.

Reducing the number of triad types used to generate networks with a given blockmodel can be beneficial in several ways. For example, it can help to identify the main (e.g. social) mechanisms that cause a given blockmodel structure to be formed.

In addition, there are practical reasons which differ with respect to the algorithm used. For the RL algorithm, the reduction to only forbidden triad types (or a subset of forbidden triad types) is especially appealing as it does not require knowledge of the exact distribution of triad types in the ideal network (as this algorithm otherwise requires) because the frequency of all forbidden triad types is 0.

The frequencies of different forbidden triad types are also not affected by the sizes and number of clusters. This means that, when generating networks by considering only the

Table 1. A-measure values and the classification of allowed and forbidden triad types, and selected allowed and forbidden triad types for different types of blockmodels. Values greater than zero denote allowed types of triads while the values which equals zero (cells without numbers) denotes forbidden types of triads; grey color denotes selected triad types.

	COHESIVE	ASYMMETRIC CORE-PERIPHERY	SYMMETRIC CORE-PERIPHERY	HIERARCHICAL WITHOUT COMPLETE BLOCKS ON THE DIAGONAL	HIERARCHICAL WITH COMPLETE BLOCKS ON THE DIAGONAL	TRANSITIVITY WITHOUT COMPLETE BLOCKS ON THE DIAGONAL	TRANSITIVITY WITH COMPLETE BLOCKS ON THE DIAGONAL
003	2.3	7.1	7.2	1.5		1.1	
300	96.3	7.5	2.7		3.7		1.2
120D		8.2			4.1		5.1
120U					4.1		5.1
102	10.2				5.8		
021C				2.2	3.1		
021U		8.2		4.0		5.1	
021D				4.0		5.1	
030T						3.5	3.5
201			6.6				
120C							
111D							
111U							
030C							
210							
012							

<https://doi.org/10.1371/journal.pone.0197514.t001>

forbidden triad types, this information is not taken into account, which may be either desired or not. On the other hand, the frequencies of all allowed triad types contain all the information that is included in all (allowed and forbidden) triad types.

Networks generated by the RL algorithm can still differ as the CR (see Eq 2) is computed slightly differently. For the MCMC algorithm, these issues are not relevant since the exact distribution of triad types is never taken into account when setting the parameter values. However, the MCMC algorithm is affected by multicollinearity, which can be reduced by selecting only a subset of all triad types. Given the point of this algorithm, it is best to select only a small number of relatively different triad types.

Selecting subsets of triad types

The sets of allowed and forbidden triad types can be further reduced to the selected allowed and selected forbidden triad types. There are several ways to select the subset of triad types. In this study, choice of triad types is based on their sensitivity to different levels of errors (from 0.2 to 1 with step 0.2), where the sensitivity is evaluated through the value of the A-measure (10,000 random networks were generated for each level of errors). A more detailed description of the process is given in S1 Appendix. The study does not focus on how to select the smallest sufficient subset of triad types to generate networks with a given blockmodel, and therefore we do not imply our procedure is the best possible one. Rather, the aim is to test whether it is possible to generate networks with a given blockmodel by considering a smaller set of triad types.

The selected triad types are shown in grey in Table 1. It may be seen that only a few triad types are allowed for each type of blockmodel. Almost all of these types of triads are chosen for all types of blockmodels. The exceptions are triad type 021C in the case of a hierarchical blockmodel without complete blocks on the diagonal and triad type 300 in the case of a transitivity

blockmodel with complete blocks on the diagonal. On the other hand, for some blockmodel types only a small number of all forbidden triad types is selected, e.g. in the case of an asymmetric core-periphery only one, and in the case of a cohesive blockmodel only two.

Simulation design

To address the objective of this study, it is assessed how well networks generated with the previously described algorithms (the number of iterations is set to 6,000 in the RL algorithm and to 10,000 in the MCMC algorithm) match the assumed blockmodel type. With each algorithm, 50 networks (of size 24 units) with a given blockmodel structure are generated for each selected set of triads. Each generated network is randomized. The R programming language [56] is used.

Pre-specified blockmodeling (also called generalized blockmodeling) is applied to model networks and randomized networks where the number of clusters is set as in the ideal networks (to two or three clusters, see Fig 3) (the `blockmodeling` [57] package implemented in R). The partition is determined by 100 restarts of the blockmodeling algorithm. For each generated network, the minimal value of the blockmodeling criterion function is preserved.

Here, it should be highlighted that there may be bias in the values of the criterion function, where the networks are generated by the RL algorithm and all allowed triad types are considered. This is because the information on the number and sizes of the clusters is embedded in the frequencies of the different allowed triad types when using the RL algorithm. Yet this is not the case when the MCMC algorithm is used and/or other subsets of triads are considered.

As the criterion function is not generally comparable for different blockmodels, we propose the Mean Improvement Value (MIV) which is calculated as

$$MIV = 1 - \frac{1}{k} \sum_{i=1}^k \left(\frac{P_i^m}{P_i^r} \right) \quad (7)$$

where P_i^r is the value of the criterion function of the i -th randomized network and P_i^m is the value of the criterion function of i -th network generated based on the model and k is the number of generated networks. Its expected value in the case of a random partition is 0 and in the case of an ideal blockmodel the value of the MIV equals to 1. The MIV obtained on a network with a different number of units is generally not comparable.

For each type of generated network (the RL algorithm, the MCMC algorithm with fixed density, and the MCMC algorithm with non-fixed density), visualizations of the P^r and the P^m are presented in S19–S21 Figs for each type of blockmodel and each combination of triad types considered. The corresponding values of the MIV are discussed in the next section.

Results

This section is organized in several subsections. First, to evaluate whether it is possible to generate networks with a given blockmodel by considering different triad types, the global network structures of the networks generated with the RL algorithm and MCMC algorithm (fixed and non-fixed density) are evaluated. For each algorithm, the networks generated by considering different sets of triad types are compared. Then, considerations are presented of certain other local network structures in the event triads do not generate the networks with the expected blockmodel. Finally, a general statement on generating networks using triads is given.

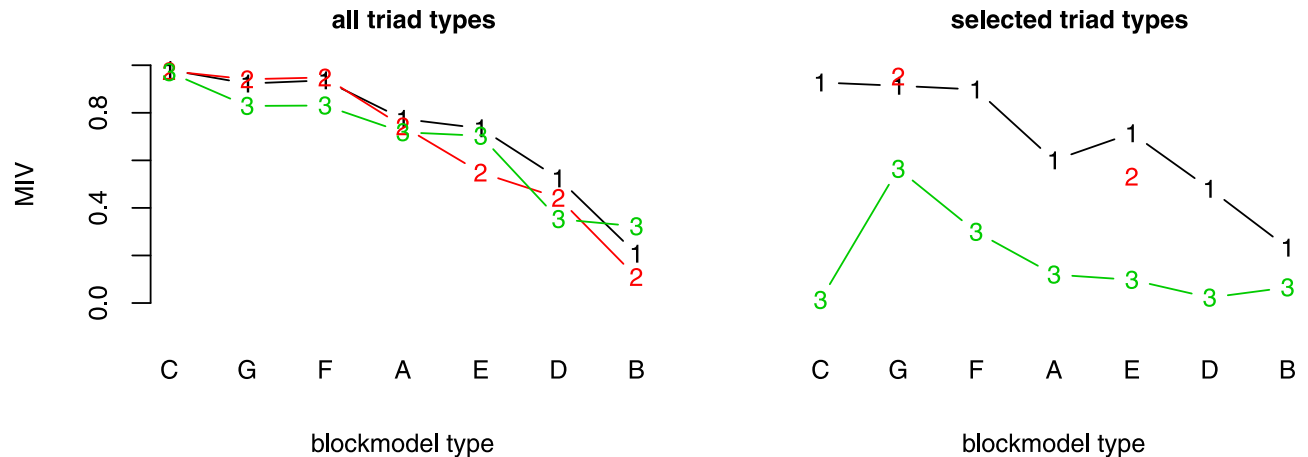


Fig 4. The MIV for each blockmodel type (generated by the RL algorithm) and selected set of triad types. (A) cohesive; (B) symmetric core-periphery; (C) asymmetric core-periphery; (D) hierarchical without complete blocks on the diagonal (E) hierarchical with complete blocks on the diagonal; (F) transitivity without complete blocks on the diagonal; (G) transitivity with complete blocks on the diagonal, (1) allowed and forbidden triad types, (2) allowed triad types, (3) forbidden triad types. Note: only networks of transitivity with complete blocks on the diagonal blockmodel type and hierarchical with complete blocks on the diagonal blockmodel type were generated by considering the selected allowed triad types.

<https://doi.org/10.1371/journal.pone.0197514.g004>

Networks generated with the RL algorithm

When the RL algorithm is used to generate networks and all triad types are considered, the overall MIV is around 72%, which is more than for all other sets of triads considered (Fig 4). On the other hand, the MIV corresponding to networks generated based only on all forbidden triads or all allowed triads is slightly lower or the same. What is outstanding is the symmetric core-periphery with the lowest MIV varying between 11 and 32% among the different models (all, all allowed, or all forbidden triad types). As has been emphasized, when the network is very dense the RL algorithm is less effective in finding the right link to relocate. This is expressed in a very small peripheral part in the case of a symmetric core-periphery blockmodel.

The MIVs are usually lower when all forbidden triad types are considered. The MIVs corresponding to the cohesive blockmodel are very similar, yet the structure of the blockmodels so generated is different when only the set of forbidden triad types is considered (the cluster sizes are more variable).

When comparing the different blockmodel types, the highest MIV is observed in the case of an asymmetric core-periphery blockmodel (98% when all allowed or all forbidden types of triads are considered), in the case of a transitivity blockmodel without complete blocks on the diagonal (95% when all allowed types of triads are considered and 94% when all types of triads are considered) and in the case of a transitivity blockmodel with complete blocks on the diagonal (94% when all allowed triad types are considered and 92% when all triad types are considered). In the latter case, there is quite considerable variability in the cluster sizes when all forbidden types of triads are considered. To be more precise, the tendency to form one cluster with a relatively high number of units and two clusters with a lower number of units is present. This happens because different types of triads can be present in different parts of the network.

When generating networks with a hierarchical blockmodel without complete blocks on the diagonal, a blockmodel structure which is not assumed emerges. Instead, there are links in the blocks below the diagonal of the matrix and in the blocks above the diagonal. This means there are links from the top to the lowest clusters and from the low clusters to the higher clusters. On the level of units, only asymmetric links are possible. However, the density is still higher in

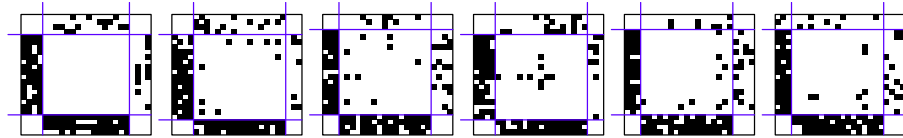


Fig 5. Some generated networks with the expected hierarchical blockmodel without complete blocks on the diagonal. The RL algorithm is used and all triad types are considered.

<https://doi.org/10.1371/journal.pone.0197514.g005>

complete than in null blocks (Fig 5), which may be a consequence of the optimization algorithm for pre-specified blockmodeling.

When all allowed triads are included in the process of generating networks, one would expect a similar MIV as when all triads are included in the model. This is because all the information for generating the networks embedded in all triad types is also embedded in only allowed triad types (as all the rest have a count of 0).

The set of all allowed triad types and the set of triads with selected allowed types of triads vary only in the case of a hierarchical blockmodel with complete blocks on the diagonal and a transitivity blockmodel with complete blocks on the diagonal. The selection of triad types slightly improves the MIV in the case of both blockmodel types. In the former case, the blockmodel structure can be visually recognized in most, but not all, generated networks. On the other hand, there are very low levels of errors in all generated networks with a transitive blockmodel with complete blocks on the diagonal.

Comparing the networks generated with all forbidden triad types and the networks generated with only the selected forbidden triad type, the MIV is generally lower in the latter case for all types of blockmodels. By visually observing some generated networks, it is hard to recognize the assumed blockmodel structure, except for some transitive blockmodels with complete blocks on the diagonal.

Networks generated with the MCMC algorithm: Fixed density

Since the RL algorithm is more deterministic, it generally performs better than the MCMC algorithm. But when networks are denser the MCMC algorithm might perform better as in the case when considering the set of all allowed types of triads when generating a symmetric core-periphery blockmodel. This is another reason one has to consider different algorithms when studying micro structures in the context of various global network structures using simulations.

When all possible triad types are considered, the overall MIV among all blockmodel types is higher when the networks are generated using the RL algorithm and lower when the networks are generated using the MCMC algorithm with a fixed density (Fig 6). Yet generated networks have an assumed blockmodel structure (Fig 7) with a relatively low level of errors, except the hierarchical one without complete blocks on the diagonal where the global network structure obtained is similar to that produced with the RL algorithm (considering selected allowed triad types) (see Fig 5). Further, the hierarchical structures with complete blocks on the diagonal and the cohesive one are less clear than the others.

Considering only all allowed or only all forbidden triad types does not produce networks with a significantly higher level of errors. The MIVs are lower when selected forbidden triad types are considered compared to the case when all forbidden triad types are considered. In this instance, the generated networks do not have the expected blockmodel.

Further selection of the different types of triads that are allowed does not improve a hierarchical blockmodel with complete blocks on the diagonal, even though some MIVs indicate the

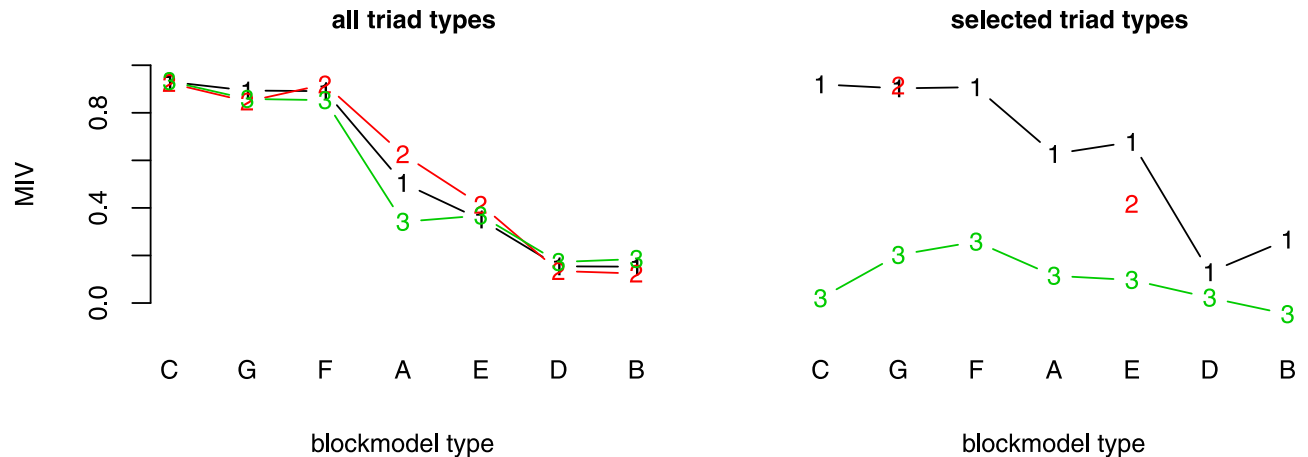


Fig 6. The MIV for each blockmodel type (generated by the MCMC algorithm with fixed density) and selected set of triad types. (A) cohesive; (B) symmetric core-periphery; (C) asymmetric core-periphery; (D) hierarchical without complete blocks on the diagonal (E) hierarchical with complete blocks on the diagonal; (F) transitivity without complete blocks on the diagonal; (G) transitivity with complete blocks on the diagonal, (1) allowed and forbidden triad types, (2) allowed triad types, (3) forbidden triad types. Note: only the networks of transitivity with complete blocks on the diagonal blockmodel type and hierarchical with complete blocks on the diagonal blockmodel type were generated by considering the selected allowed triad types.

<https://doi.org/10.1371/journal.pone.0197514.g006>

opposite. Conversely, a further selection of allowed triad types improves the global structure of networks with an expected transitivity blockmodel with complete blocks on the diagonal.

The further selection of all possible triad types (allowed and forbidden) improves all the MIV values, especially those corresponding to the hierarchical blockmodel with complete blocks on the diagonal and the cohesive blockmodel.

The generated networks with the expected hierarchical blockmodel structure without complete blocks on the diagonal are not in line with the expected global network structure. This is true for any set of triad types considered.

Networks generated with the MCMC algorithm: Non-fixed density

In the event the initial networks are totally randomized ideal networks, networks generated using the MCMC algorithm with a non-fixed density are close to the networks with a fixed density (Fig 8). The further selection of different triad types is not seen as so important in the case of an asymmetric core-periphery blockmodel and a transitivity blockmodel (with or without complete blocks on the diagonal), while it improves the structure of generated networks with an expected symmetric core-periphery blockmodel and a hierarchical blockmodel with complete blocks on the diagonal.

Here, it is noted that the way the initial networks are chosen has a great impact on the networks generated. When considering the random networks (with the expected number of links is equal to the number of links in an ideal network) as initial networks and the MCMC algorithm with a non-fixed density is used, many generated networks are totally empty or totally full. This is especially when all triad types are included in the model. In this study, the randomized ideal networks are used as initial networks, meaning the density of the initial networks is not variable and is the same as in the ideal networks.

Improvement of the hierarchical blockmodel without complete blocks on the diagonal

The proposed models for generating networks with a hierarchical blockmodel structure without complete blocks on the diagonal perform poorly. This is seen by the mean improvement

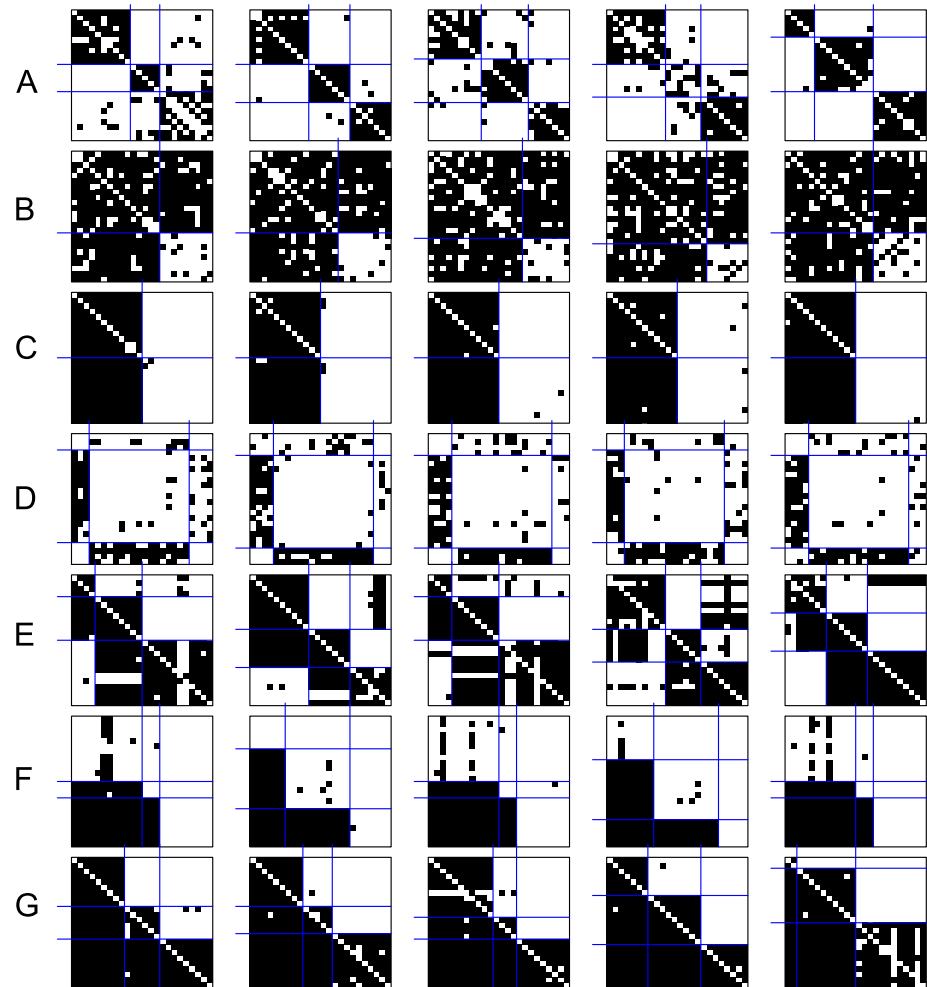


Fig 7. Some empirical generated networks using the RL algorithm by considering all triad types. By rows: (A) cohesive; (B) symmetric core-periphery; (C) asymmetric core-periphery; (D) hierarchical without complete blocks on the diagonal; (E) hierarchical with complete blocks on the diagonal; (F) transitivity without complete blocks on the diagonal; (G) transitivity with complete blocks on the diagonal.

<https://doi.org/10.1371/journal.pone.0197514.g007>

values and the distribution of the values of the criterion function (see S1 and S21 Figs, Figs 6, 7D and 9A for some empirical examples).

The obtained blockmodel structure is often hierarchical but has additional links from the upper to the lower clusters and with all asymmetric links. This is especially typical of networks generated using the MCMC algorithm. Therefore, the main focus is put on the networks generated using the MCMC algorithm with a non-fixed density. The resulting global structure probably emerges since all considered triad types appear in all parts of the network. Their combination produces a network that is highly determined by paths of length three (e.g., $1 \rightarrow 2 \rightarrow 3 \rightarrow 2$, where digits denote clusters). The MIV is 0.17.

Therefore, by considering the paths of length three, the links from the upper to the lower positions are omitted. Here, it should be pointed out that the number of triads is unit-based while the number of paths of length three is an edge-based count. However, an additional parameter paths of length three (in the case of networks with a different number of positions, paths of different lengths should be considered) is added to the model with the value of -2 (as forbidden). Networks generated using this model have the expected hierarchical structure but

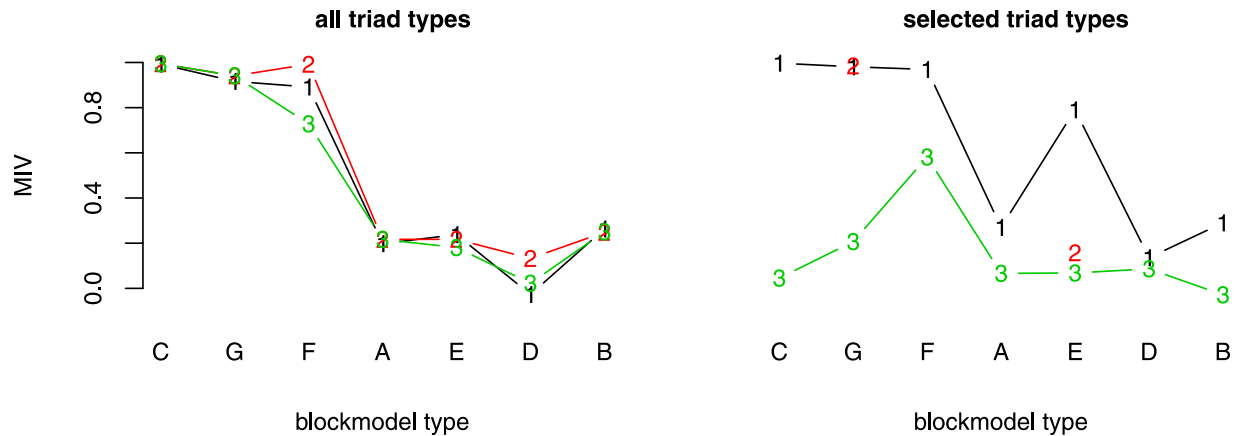


Fig 8. The MIV for each blockmodel type (generated by the MCMC algorithm with variable density) and selected set of triad types. (A) cohesive; (B) symmetric core-periphery; (C) asymmetric core-periphery; (D) hierarchical without complete blocks on the diagonal (E) hierarchical with complete blocks on the diagonal; (F) transitivity without complete blocks on the diagonal; (G) transitivity with complete blocks on the diagonal, (1) allowed and forbidden triad types, (2) allowed triad types, (3) forbidden triad types. Note: only the networks of transitivity with complete blocks on the diagonal blockmodel type and hierarchical with complete blocks on the diagonal blockmodel type were generated by considering the selected allowed triad types.

<https://doi.org/10.1371/journal.pone.0197514.g008>

with only two clusters (Fig 9B). From time to time, networks with a transitivity blockmodel without complete blocks on the diagonal are also produced (MIV = 0.74).

To obtain three positions (instead of two), the parameter's value of triad type 021C has to be increased, e.g. to the value to 4. Such a model produces networks with a very clear hierarchical structure without complete blocks on the diagonal (Fig 9C). There are no errors in all generated networks in null blocks while some appear in complete blocks (MIV = 0.93).

All of the described networks are generated using the MCMC algorithm. When the RL algorithm is used, all allowed types of triads and paths of length three can be considered. In that

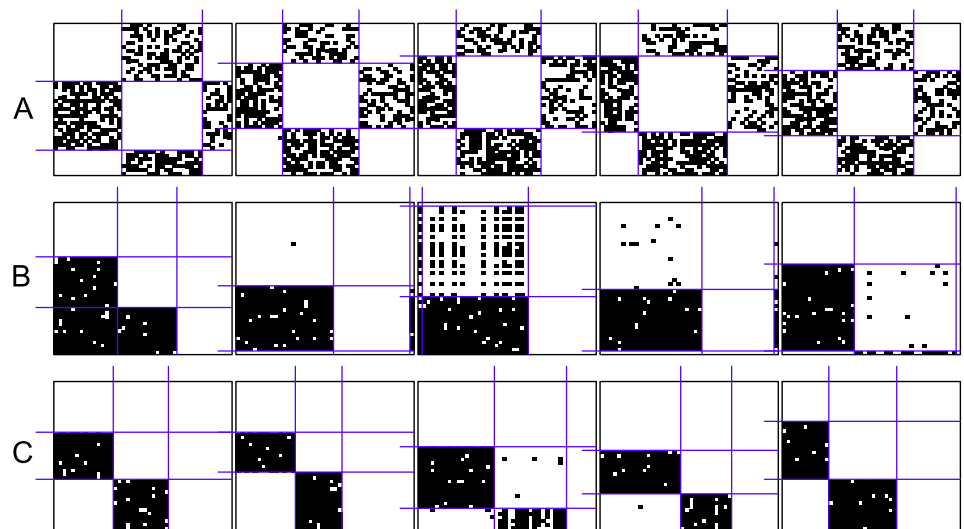


Fig 9. Some generated networks with a hierarchical blockmodel structure without complete blocks on the diagonal using different models. The networks are generated by the MCMC algorithm with a non-fixed density. (a) All selected types of triads; (b) All selected types of triads and paths of length three; (c) All selected types of triads (with a higher parameter value for the triad of type 021D) and paths of length three.

<https://doi.org/10.1371/journal.pone.0197514.g009>

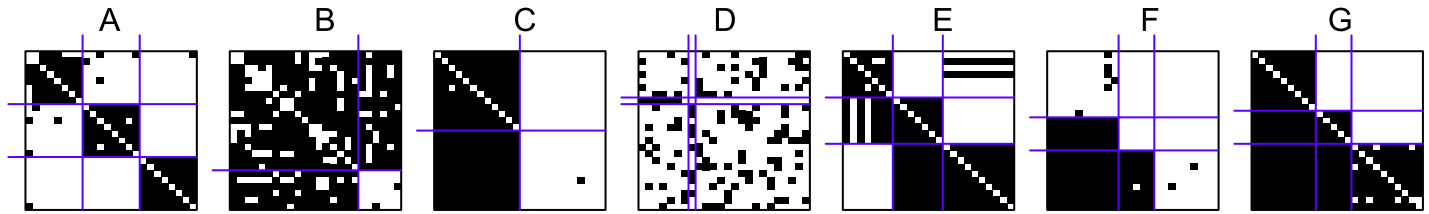


Fig 10. Some examples of generated networks for each type of a blockmodel. The networks are generated using the RL algorithm by selected allowed types of triads. (A) cohesive; (B) symmetric core-periphery; (C) asymmetric core-periphery; (D) hierarchical without complete blocks on the diagonal; (E) hierarchical with complete blocks on the diagonal; (F) transitivity without complete blocks on the diagonal; (G) transitivity with complete blocks on the diagonal.

<https://doi.org/10.1371/journal.pone.0197514.g010>

case, some errors appear in both null and complete blocks, which is a consequence of the fixed density. However, with a higher number of iterations, the number of errors could also be lower.

Concluding remarks about generating networks with triads

Three approaches are proposed for generating networks with a given blockmodel structure. The networks so generated are compared with totally random networks of the same density. As expected, there are fewer errors in these networks when generated using the RL algorithm compared to the MCMC algorithm. With both approaches, the selection of triad types does not necessarily result in generated networks with higher or lower levels of errors.

Both algorithms perform well in the case of asymmetric core-periphery blockmodels. However, in the case of the symmetric core-periphery blockmodel, the MIVs are usually small, which is reflected by the insufficiently small periphery in the generated networks. It is also hard to generate a hierarchical blockmodel without complete blocks on the diagonal when considering only different triad types, regardless of the algorithm used to generate the networks. By adding paths of length three, the empirical networks produced have the expected blockmodel type with a very low level of errors (see Fig 9).

One of the most important findings is that the number of different types of triads reflects the assumed global network structure (all generated networks have the expected blockmodel, see Fig 10), where it is often sufficient to consider only some of all possible types of triads.

Conclusion

This paper examined whether often used global structures, operationalized by blockmodels, can emerge as a consequence of local processes operationalized by different triad types. It was shown that for most of the studied blockmodels this indeed happened. The only exception is a hierarchical blockmodel without complete blocks on the diagonal, where additional local structures are needed to obtain a good fit with the assumed blockmodel structure.

The main conclusion of the study is that the global network structures, considered in this paper, can emerge due to the local mechanisms, regardless of the characteristics of the units.

This was shown by generating networks that considered different triad types using two algorithms: the proposed deterministic Relocating Links (RL) algorithm and the Monte Carlo Markov Chain (MCMC) algorithm (specifically Metropolis-Hastings algorithm) [49]. The RL algorithm randomly selects a link and exchanges it with a randomly selected non-link. The change is accepted if the new network's local structure count is closer to the target count than in the previous network. With the MCMC algorithm, the same local structures were used as parameters in the ergm model.

To determine the target count for the RL algorithm and the parameter values for the MCMC algorithm, the count of different triad types in ideal networks (namely, those that perfectly comply with a certain blockmodel) had to be determined. This was achieved by considering the specific blockmodel type and corresponding cluster sizes. All types of triads were classified in the set of forbidden and set of allowed triad types for each blockmodel. Allowed triad types are those that are present in ideal networks and forbidden triad types are those that are not. The RL algorithm uses counts of a selected local structure in an ideal blockmodel while for the MCMC algorithm the parameter values were determined based on the classification into allowed and forbidden triad types. Both algorithms performed very well; the exception is the hierarchical model without complete blocks on the diagonal (an additional parameter must be added) and to a smaller extent the symmetrical core-periphery model. On average, the RL algorithm performed slightly better.

This paper also explored whether one can reduce the required local structure information by using only allowed or only forbidden triad types. Using only forbidden types of triads is especially desirable for the RL algorithm as the count for this triad type is zero. In addition, the reduction of all these sets (all, allowed, forbidden) of triad types was studied based on their sensitivity to errors according to the blockmodel structure. Most of these reductions of sets of different triad types overall resulted in only a slightly worse fit and in some cases even in an improved performance. The only exception is when using only selected forbidden triad types, which did not generate the assumed blockmodel structure.

Some considered blockmodel types are defined for only two clusters (symmetric and asymmetric core-periphery blockmodels). The other blockmodel types can consist of more than three clusters. The initial tests suggest that for cohesive blockmodel, transitive blockmodel (both types: with complete blocks on the diagonal and with null blocks on the diagonal), the results presented in this paper can be generalized to the blockmodels with higher number of clusters, while for others such speculations based on our tests are not possible.

There are several ways this study could be extended. One would be to move from considering the local structures (e.g. types of triads) to explicitly define other types of rules for creating and dissolving links in the network. Another extension of this study could be to consider the selected social mechanisms that lead to a change from one particular blockmodel to another particular blockmodel, which was not addressed in this study due to the very high context-dependency of the social mechanisms and blockmodels. Something similar was done by Berman et al. [58] who showed that the sexual and romantic network structure of students is close to a hierarchical structure due to a social norm that prohibits cycles of length four.

Besides developing sociological theory, understanding such mechanisms might also help generate the evolution of networks towards given blockmodels. This can also be useful when testing dynamic blockmodeling algorithms [59–61].

Different triad types were studied in the context of different blockmodel types using simulations. Based on the results given in this paper, a more mathematical relationship between the different triad types and different blockmodel types can be established.

Supporting information

S1 Appendix. Generating totally randomised networks and networks with a given level of errors.

(PDF)

S2 Appendix. Selected allowed and forbidden triad types.

(PDF)

S1 Fig. Some empirical generated networks using the RL algorithm by considering all triad types. By rows: (A) cohesive; (B) symmetric core-periphery; (C) asymmetric core-periphery; (D) hierarchical without complete blocks on the diagonal; (E) hierarchical with complete blocks on the diagonal; (F) transitivity without complete blocks on the diagonal; (G) transitivity with complete blocks on the diagonal.

(PDF)

S2 Fig. Some empirical generated networks using the RL algorithm by considering allowed triad types. By rows: (A) cohesive; (B) symmetric core-periphery; (C) asymmetric core-periphery; (D) hierarchical without complete blocks on the diagonal; (E) hierarchical with complete blocks on the diagonal; (F) transitivity without complete blocks on the diagonal; (G) transitivity with complete blocks on the diagonal.

(PDF)

S3 Fig. Some empirical generated networks using the RL algorithm by considering forbidden triad types. By rows: (A) cohesive; (B) symmetric core-periphery; (C) asymmetric core-periphery; (D) hierarchical without complete blocks on the diagonal; (E) hierarchical with complete blocks on the diagonal; (F) transitivity without complete blocks on the diagonal; (G) transitivity with complete blocks on the diagonal.

(PDF)

S4 Fig. Some empirical generated networks using the RL algorithm by considering all selected triad types. By rows: (A) cohesive; (B) symmetric core-periphery; (C) asymmetric core-periphery; (D) hierarchical without complete blocks on the diagonal; (E) hierarchical with complete blocks on the diagonal; (F) transitivity without complete blocks on the diagonal; (G) transitivity with complete blocks on the diagonal.

(PDF)

S5 Fig. Some empirical generated networks using the RL algorithm by considering selected allowed triad types. By rows: (A) cohesive; (B) symmetric core-periphery; (C) asymmetric core-periphery; (D) hierarchical without complete blocks on the diagonal; (E) hierarchical with complete blocks on the diagonal; (F) transitivity without complete blocks on the diagonal; (G) transitivity with complete blocks on the diagonal.

(PDF)

S6 Fig. Some empirical generated networks using the RL algorithm by considering selected forbidden triad types. By rows: (A) cohesive; (B) symmetric core-periphery; (C) asymmetric core-periphery; (D) hierarchical without complete blocks on the diagonal; (E) hierarchical with complete blocks on the diagonal; (F) transitivity without complete blocks on the diagonal; (G) transitivity with complete blocks on the diagonal.

(PDF)

S7 Fig. Some empirical generated networks using the MCMC algorithm (fixed density) by considering all triad types. By rows: (A) cohesive; (B) symmetric core-periphery; (C) asymmetric core-periphery; (D) hierarchical without complete blocks on the diagonal; (E) hierarchical with complete blocks on the diagonal; (F) transitivity without complete blocks on the diagonal; (G) transitivity with complete blocks on the diagonal.

(PDF)

S8 Fig. Some empirical generated networks using the MCMC algorithm (fixed density) by considering allowed triad types. By rows: (A) cohesive; (B) symmetric core-periphery; (C) asymmetric core-periphery; (D) hierarchical without complete blocks on the diagonal; (E)

hierarchical with complete blocks on the diagonal; (F) transitivity without complete blocks on the diagonal; (G) transitivity with complete blocks on the diagonal.

(PDF)

S9 Fig. Some empirical generated networks using the MCMC algorithm (fixed density) by considering forbidden triad types. By rows: (A) cohesive; (B) symmetric core-periphery; (C) asymmetric core-periphery; (D) hierarchical without complete blocks on the diagonal; (E) hierarchical with complete blocks on the diagonal; (F) transitivity without complete blocks on the diagonal; (G) transitivity with complete blocks on the diagonal.

(PDF)

S10 Fig. Some empirical generated networks using the MCMC algorithm (fixed density) by considering all selected triad types. By rows: (A) cohesive; (B) symmetric core-periphery; (C) asymmetric core-periphery; (D) hierarchical without complete blocks on the diagonal; (E) hierarchical with complete blocks on the diagonal; (F) transitivity without complete blocks on the diagonal; (G) transitivity with complete blocks on the diagonal.

(PDF)

S11 Fig. Some empirical generated networks using the MCMC algorithm (fixed density) by considering selected allowed triad types. By rows: (A) cohesive; (B) symmetric core-periphery; (C) asymmetric core-periphery; (D) hierarchical without complete blocks on the diagonal; (E) hierarchical with complete blocks on the diagonal; (F) transitivity without complete blocks on the diagonal; (G) transitivity with complete blocks on the diagonal.

(PDF)

S12 Fig. Some empirical generated networks using the MCMC algorithm (fixed density) by considering selected forbidden triad types. By rows: (A) cohesive; (B) symmetric core-periphery; (C) asymmetric core-periphery; (D) hierarchical without complete blocks on the diagonal; (E) hierarchical with complete blocks on the diagonal; (F) transitivity without complete blocks on the diagonal; (G) transitivity with complete blocks on the diagonal.

(PDF)

S13 Fig. Some empirical generated networks using the MCMC algorithm (variable density) by considering all triad types. By rows: (A) cohesive; (B) symmetric core-periphery; (C) asymmetric core-periphery; (D) hierarchical without complete blocks on the diagonal; (E) hierarchical with complete blocks on the diagonal; (F) transitivity without complete blocks on the diagonal; (G) transitivity with complete blocks on the diagonal.

(PDF)

S14 Fig. Some empirical generated networks using the MCMC algorithm (variable density) by considering allowed triad types. By rows: (A) cohesive; (B) symmetric core-periphery; (C) asymmetric core-periphery; (D) hierarchical without complete blocks on the diagonal; (E) hierarchical with complete blocks on the diagonal; (F) transitivity without complete blocks on the diagonal; (G) transitivity with complete blocks on the diagonal.

(PDF)

S15 Fig. Some empirical generated networks using the MCMC algorithm (variable density) by considering forbidden triad types. By rows: (A) cohesive; (B) symmetric core-periphery; (C) asymmetric core-periphery; (D) hierarchical without complete blocks on the diagonal; (E) hierarchical with complete blocks on the diagonal; (F) transitivity without complete blocks on the diagonal; (G) transitivity with complete blocks on the diagonal.

(PDF)

S16 Fig. Some empirical generated networks using the MCMC algorithm (variable density) by considering all selected triad types. By rows: (A) cohesive; (B) symmetric core-periphery; (C) asymmetric core-periphery; (D) hierarchical without complete blocks on the diagonal; (E) hierarchical with complete blocks on the diagonal; (F) transitivity without complete blocks on the diagonal; (G) transitivity with complete blocks on the diagonal.
(PDF)

S17 Fig. Some empirical generated networks using the MCMC algorithm (variable density) by considering selected allowed triad types. By rows: (A) cohesive; (B) symmetric core-periphery; (C) asymmetric core-periphery; (D) hierarchical without complete blocks on the diagonal; (E) hierarchical with complete blocks on the diagonal; (F) transitivity without complete blocks on the diagonal; (G) transitivity with complete blocks on the diagonal.
(PDF)

S18 Fig. Some empirical generated networks using the MCMC algorithm (variable density) by considering selected forbidden triad types. By rows: (A) cohesive; (B) symmetric core-periphery; (C) asymmetric core-periphery; (D) hierarchical without complete blocks on the diagonal; (E) hierarchical with complete blocks on the diagonal; (F) transitivity without complete blocks on the diagonal; (G) transitivity with complete blocks on the diagonal.
(PDF)

S19 Fig. The distribution of the criterion function for the generated networks and for randomised networks with the mean improvement value obtained by the RL algorithm. R denotes the distribution of P_R and M denotes the distribution of P_m . By rows: (A) cohesive; (B) symmetric core-periphery; (C) asymmetric core-periphery; (D) hierarchical without complete blocks on the diagonal; (E) hierarchical with complete blocks on the diagonal; (F) transitivity without complete blocks on the diagonal; (G) transitivity with complete blocks on the diagonal. By columns: (H) all; (I) all allowed; (J) all forbidden; (K) selected; (L) selected allowed; (M) selected forbidden types of triads.
(PDF)

S20 Fig. The distribution of the criterion function for the generated networks and for randomised networks with the mean improvement value obtained by the MCMC algorithm with fixed density. R denotes the distribution of P_R and M denotes the distribution of P_m . R denotes the distribution of P_R and M denotes the distribution of P_m . By rows: (A) cohesive; (B) symmetric core-periphery; (C) asymmetric core-periphery; (D) hierarchical without complete blocks on the diagonal; (E) hierarchical with complete blocks on the diagonal; (F) transitivity without complete blocks on the diagonal; (G) transitivity with complete blocks on the diagonal. By columns: (H) all; (I) all allowed; (J) all forbidden; (K) selected; (L) selected allowed; (M) selected forbidden types of triads.
(PDF)

S21 Fig. The distribution of the criterion function for the generated networks and for randomised networks with the mean improvement value obtained by the MCMC algorithm with non-fixed density. R denotes the distribution of P_R and M denotes the distribution of P_m . R denotes the distribution of P_R and M denotes the distribution of P_m . By rows: (A) cohesive; (B) symmetric core-periphery; (C) asymmetric core-periphery; (D) hierarchical without complete blocks on the diagonal; (E) hierarchical with complete blocks on the diagonal; (F) transitivity without complete blocks on the diagonal; (G) transitivity with complete blocks on the diagonal. By columns: (H) all; (I) all allowed; (J) all forbidden; (K) selected; (L) selected

allowed; (M) selected forbidden types of triads.
(PDF)

Author Contributions

Conceptualization: Anuška Ferligoj.

Formal analysis: Marjan Cugmas.

Funding acquisition: Anuška Ferligoj, Aleš Žiberna.

Investigation: Marjan Cugmas, Anuška Ferligoj, Aleš Žiberna.

Methodology: Marjan Cugmas, Anuška Ferligoj, Aleš Žiberna.

Project administration: Marjan Cugmas.

Software: Marjan Cugmas, Aleš Žiberna.

Supervision: Anuška Ferligoj, Aleš Žiberna.

Validation: Marjan Cugmas, Anuška Ferligoj, Aleš Žiberna.

Visualization: Marjan Cugmas.

Writing – original draft: Marjan Cugmas.

Writing – review & editing: Anuška Ferligoj, Aleš Žiberna.

References

1. Doreian P, Batagelj V, Ferligoj A. Generalized blockmodeling. Cambridge university press; 2005.
2. Nadel SF. The Theory of Social Structure. Routledge; 1957.
3. Homans GC. The human group. Harcourt, Brace & World; 1950.
4. Lewin K. Principles of topological psychology. McGraw-Hill Book Company; 1936.
5. Barnett GA, Danowski JA. The Structure of Communication: A Network Analysis of the International Communication Association. Human Communication Research. 1992; 19(2):264–285. <https://doi.org/10.1111/j.1468-2958.1992.tb00302.x>
6. Alderson AS, Beckfield J. Power and Position in the World City System. American Journal of Sociology. 2004; 109(4):811–851. <https://doi.org/10.1086/378930>
7. Glückler J, Panitz R. Unpacking Social Divisions of Labor in Markets: Generalized Blockmodeling and the Network Boom in Stock Photography. Social Networks. 2016; 47(1):156–166.
8. Žiberna A, Lazega E. Role Sets and Division of Work at Two Levels of Collective Agency: The Case of Blockmodeling a Multilevel (Inter-Individual and Inter-Organizational) Network. In: Lazega E, Snijders TAB, editors. Multilevel Network Analysis for the Social Sciences. No. 12 in Methodos Series. Springer International Publishing; 2016. p. 173–209.
9. Kronegger L, Ferligoj A, Doreian P. On the Dynamics of National Scientific Systems. Quality & Quantity. 2011; 45(5):989–1015. <https://doi.org/10.1007/s11135-011-9484-3>
10. Prota L, D'Esposito MR, De Stefano D, Giordano G, Vitale MP. Modeling Cooperative Behaviors in Innovation Networks: An Empirical Analysis. Spohrer JC, Freund LE, editors. Crc Press-Taylor & Francis Group; 2013.
11. Prota L, Doreian P. Finding Roles in Sparse Economic Hierarchies: Going beyond Regular Equivalence. Social Networks. 2016; 45(1):1–17. <https://doi.org/10.1016/j.socnet.2015.10.005>
12. Lorrain F, White HC. Structural equivalence of individuals in social networks. The Journal of mathematical sociology. 1971; 1(1):49–80. <https://doi.org/10.1080/0022250X.1971.9989788>
13. Luczkovich JJ, Borgatti SP, Johnson JC, Everett MG. Defining and measuring trophic role similarity in food webs using regular equivalence. Journal of Theoretical Biology. 2003; 220(3):303–321. <https://doi.org/10.1006/jtbi.2003.3147> PMID: 12468282
14. White HC, Boorman SA, Breiger RL. Social Structure from Multiple Networks. I. Blockmodels of Roles and Positions. American Journal of Sociology. 1976; 4(81):730. <https://doi.org/10.1086/226141>

15. Davis JA, Leinhardt S. The structure of positive interpersonal relations in small groups. ERIC; 1967.
16. Cartwright D, Harary F. Structural balance: a generalization of Heider's theory. *Psychological review*. 1956; 63(5):277. <https://doi.org/10.1037/h0046049> PMID: 13359597
17. Davis JA. Clustering and structural balance in graphs. *Human relations*. 1967; 20(2):181–187. <https://doi.org/10.1177/001872676702000206>
18. Holland PW, Leinhardt S. Transitivity in structural models of small groups. *Comparative group studies*. 1971; 2(2):107–124. <https://doi.org/10.1177/104649647100200201>
19. Johnsen EC. Network macrostructure models for the Davis-Leinhardt set of empirical sociomatrixes. *Social networks*. 1985; 7(3):203–224. [https://doi.org/10.1016/0378-8733\(85\)90015-2](https://doi.org/10.1016/0378-8733(85)90015-2)
20. Hedström P, Swedberg R. Social mechanisms. *Acta Sociologica*. 1996; 39(3):281–308. <https://doi.org/10.1177/000169939603900302>
21. Milo R, Shen-Orr S, Itzkovitz S, Kashtan N, Chklovskii D, Alon U. Network motifs: simple building blocks of complex networks. *Science*. 2002; 298(5594):824–827. <https://doi.org/10.1126/science.298.5594.824> PMID: 12399590
22. Kogut B. The network as knowledge: Generative rules and the emergence of structure. *Strategic management journal*. 2000; 21(3):405–425. [https://doi.org/10.1002/\(SICI\)1097-0266\(200003\)21:3%3C405::AID-SMJ103%3E3.0.CO;2-5](https://doi.org/10.1002/(SICI)1097-0266(200003)21:3%3C405::AID-SMJ103%3E3.0.CO;2-5)
23. Hedström P, Ylikoski P. Causal Mechanisms in the Social Sciences. *Annual Review of Sociology*. 2010; 36(1):49–67. <https://doi.org/10.1146/annurev.soc.012809.102632>
24. Batagelj V. Notes on blockmodeling. *Social Networks*. 1997; 19(2):143–155. [https://doi.org/10.1016/S0378-8733\(96\)00297-3](https://doi.org/10.1016/S0378-8733(96)00297-3)
25. Batagelj V, Ferligoj A, Doreian P. Fitting pre-specified blockmodels. In: *Data science, classification, and related methods*. Springer; 1998. p. 199–206.
26. Doreian P, Batagelj V, Ferligoj A. Partitioning networks based on generalized concepts of equivalence. *Journal of Mathematical Sociology*. 1994; 19(1):1–27. <https://doi.org/10.1080/0022250X.1994.9990133>
27. Lancichinetti A, Fortunato S. Community detection algorithms: a comparative analysis. *Physical review E*. 2009; 80(5):056117. <https://doi.org/10.1103/PhysRevE.80.056117>
28. Wasserman S, Faust K. *Social network analysis: Methods and applications*. Cambridge university press; 1994.
29. Borgatti SP, Everett MG. Models of core/periphery structures. *Social networks*. 2000; 21(4):375–395. [https://doi.org/10.1016/S0378-8733\(99\)00019-2](https://doi.org/10.1016/S0378-8733(99)00019-2)
30. Batagelj V, Ferligoj A, Doreian P. Direct and indirect methods for structural equivalence. *Social networks*. 1992; 14(1-2):63–90. [https://doi.org/10.1016/0378-8733\(92\)90014-X](https://doi.org/10.1016/0378-8733(92)90014-X)
31. Shen K, Hutchison RM, Bezgin G, Everling S, McIntosh AR. Network structure shapes spontaneous functional connectivity dynamics. *The Journal of Neuroscience*. 2015; 35(14):5579–5588. <https://doi.org/10.1523/JNEUROSCI.4903-14.2015> PMID: 25855174
32. Csermely P, London A, Wu LY, Uzzi B. Structure and dynamics of core/periphery networks. *Journal of Complex Networks*. 2013; 1(2):93–123. <https://doi.org/10.1093/comnet/cnt016>
33. Snijders TAB. The degree variance: an index of graph heterogeneity. *Social networks*. 1981; 3(3):163–174. [https://doi.org/10.1016/0378-8733\(81\)90014-9](https://doi.org/10.1016/0378-8733(81)90014-9)
34. Batagelj V, Mrvar A, Ferligoj A, Doreian P. Generalized blockmodeling with Pajek. *Metodoloski zvezki*. 2004; 1(2):455–467.
35. Cattani G, Ferriani S. A core/periphery perspective on individual creative performance: Social networks and cinematic achievements in the Hollywood film industry. *Organization science*. 2008; 19(6):824–844. <https://doi.org/10.1287/orsc.1070.0350>
36. Da Silva MR, Ma H, Zeng AP. Centrality, network capacity, and modularity as parameters to analyze the core-periphery structure in metabolic networks. *Proceedings of the IEEE*. 2008; 96(8):1411–1420. <https://doi.org/10.1109/JPROC.2008.925418>
37. Hu C, Racherla P. Visual representation of knowledge networks: A social network analysis of hospitality research domain. *International journal of hospitality management*. 2008; 27(2):302–312. <https://doi.org/10.1016/j.ijhm.2007.01.002>
38. Cugmas M, Ferligoj A, Kronegger L. The stability of co-authorship structures. *Scientometrics*. 2016; 106(1):163–186. <https://doi.org/10.1007/s11192-015-1790-4> PMID: 27840460
39. Chinchilla-Rodríguez Z, Ferligoj A, Miguel S, Kronegger L, de Moya-Anegón F. Blockmodeling of co-authorship networks in library and information science in Argentina: a case study. *Scientometrics*. 2012; 93(3):699–717. <https://doi.org/10.1007/s11192-012-0794-6>

40. Oberg A, Walgenbach P. Hierarchical structures of communication in a network organization. *Scandinavian Journal of management*. 2008; 24(3):183–198. <https://doi.org/10.1016/j.scaman.2008.03.011>
41. Ahuja MK, Carley KM. Network structure in virtual organizations. *Journal of computer-mediated communication*. 1998; 3(4):0–0. <https://doi.org/10.1111/j.1083-6101.1998.tb00079.x>
42. Leinhardt S. The development of transitive structure in children's interpersonal relations. *Behavioral science*. 1973; 18(4):260–271. <https://doi.org/10.1002/bs.3830180405>
43. Schaefer DR, Light JM, Fabes RA, Hanish LD, Martin CL. Fundamental principles of network formation among preschool children. *Social networks*. 2010; 32(1):61–71. <https://doi.org/10.1016/j.socnet.2009.04.003> PMID: 20161606
44. Chase ID. Dynamics of hierarchy formation: the sequential development of dominance relationships. *Behaviour*. 1982; 80(3):218–239. <https://doi.org/10.1163/156853982X00364>
45. Fararo TJ, Skvoretz J. E-state structuralism: A theoretical method. *American sociological review*. 1986; p. 591–602. <https://doi.org/10.2307/2095486>
46. Skvoretz J, Faust K, Fararo TJ. Social structure, networks, and E-state structuralism models. *Journal of mathematical sociology*. 1996; 21(1-2):57–76. <https://doi.org/10.1080/0022250X.1996.9990174>
47. Toivonen R, Kovanen L, Kivelä M, Onnela JP, Saramäki J, Kaski K. A comparative study of social network models: Network evolution models and nodal attribute models. *Social networks*. 2009; 31(4):240–254. <https://doi.org/10.1016/j.socnet.2009.06.004>
48. Kejžar N, Batagelj V. *Evolution of networks*. Faculty of Mathematics and Physics, University of Ljubljana; 2007.
49. Hunter DR, Handcock MS, Butts CT, Goodreau SM, Morris M. *ergm*: A package to fit, simulate and diagnose exponential-family models for networks. *Journal of statistical software*. 2008; 24(3): nihpa54860. <https://doi.org/10.18637/jss.v024.i03> PMID: 19756229
50. Hedström P, Manzo G. Recent trends in agent-based computational research: A brief introduction. *Sociological Methods & Research*. 2015; 44(2):179–185. <https://doi.org/10.1177/0049124115581211>
51. Robins G, Pattison P, Kalish Y, Lusher D. An introduction to exponential random graph (p^*) models for social networks *Social networks*. 2007; 29(2):173–191.
52. Corander J, Dahmström K, Dahmström P. *Maximum likelihood estimation for Markov graphs*. University of Stockholm, Department of Statistics; 1998.
53. Strauss D, Ikeda M. Pseudolikelihood estimation for social networks. *Journal of the American statistical association*. 1990; 85(409):204–212. <https://doi.org/10.1080/01621459.1990.10475327>
54. Hyvarinen A. Connections between score matching, contrastive divergence, and pseudolikelihood for continuous-valued variables. *IEEE Transactions on neural networks*. 2007; 18(5):1529–1531. <https://doi.org/10.1109/TNN.2007.895819> PMID: 18220201
55. Faust K. Comparing social networks: size, density, and local structure. *Metodoloski zvezki*. 2006; 3(2):185.
56. Team RC. *R language definition*. Vienna, Austria: R foundation for statistical computing. 2000.
57. Žiberna A. *Blockmodeling 0.1. 7: An R package for Generalized and classical blockmodeling of valued networks*. Available at <http://www2.arnes.si/~aziber4/blockmodeling>, accessed: 2018-01-29.
58. Bearman PS, Moody J, Stovel K. Chains of affection: The structure of adolescent romantic and sexual networks. *American journal of sociology*. 2004; 110(1):44–91. <https://doi.org/10.1086/386272>
59. Xing EP, Fu W, Song L. A State-Space Mixed Membership Blockmodel for Dynamic Network Tomography. *The Annals of Applied Statistics*. 2010; 4(2):535–566. <https://doi.org/10.1214/09-AOAS311>
60. Matias C, Miele V. Statistical Clustering of Temporal Networks through a Dynamic Stochastic Block Model. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*. 2017; 79(4).
61. Xu KS, Hero AO. *Dynamic Stochastic Blockmodels: Statistical Models for Time-Evolving Networks*. In: Hutchison D, Kanade T, Kittler J, Kleinberg JM, Mattern F, Mitchell JC, et al., editors. *Social Computing, Behavioral-Cultural Modeling and Prediction*. vol. 7812. Berlin, Heidelberg: Springer Berlin Heidelberg; 2013. p. 201–210.