

# blockmodeling: An R package for generalized blockmodeling

Miha Matjašič, Marjan Cugmas, Aleš Žiberna\*

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

---

## Abstract

This paper presents the R package `blockmodeling` which is primarily meant as an implementation of generalized blockmodeling (more broadly blockmodeling) for valued networks where the values of the ties are assumed to be measured on at least interval scale. Blockmodeling is one of the most commonly used approaches in the analysis of (social) networks, which deals with the analysis of relationships or connections, between the units studied (e.g., peoples, organizations, journals etc.). The R package `blockmodeling` implements several approaches for the generalized blockmodeling of binary and valued networks. Generalized blockmodeling is commonly used to cluster nodes in a network with regard to the structure of their links. The theoretical foundations of generalized blockmodeling for binary and valued networks are summarized in the paper while the use of the R package `blockmodeling` is illustrated by applying it to an empirical dataset.

---

## 1. Introduction

The aim of this paper is to present the R package `blockmodeling` which is primarily meant as an implementation of generalized blockmodeling for valued networks.

A network is defined by the set of nodes (also called vertices, units or actors) and the set of links among the nodes. These two sets determine a graph which describes the network's structure. In, e.g. social sciences, the nodes often represent individuals and the links among them represent a selected (social) relationship among individuals. Additional data can be assigned to the nodes (e.g., gender or age) and links (e.g., the number of contacts) to describe their properties (also called attributes) (Batagelj et al., 2004).

Since real-world networks may be large and complex, researchers try to simplify them to smaller and more understandable structures that are easier to interpret. A common way of accomplishing this goal is a blockmodeling approach which partitions the nodes of a network and determines the ties among the (obtained) clusters of nodes (Batagelj et al., 2004). In the social sciences, blockmodeling is also a very important explanatory tool for studying social roles because it is assumed that the way a cluster of nodes is embedded in the network structure is closely associated with the nodes' social role(s) (Borgatti and Everett, 1992).

---

\*Corresponding author

*Email addresses:* `miha.matjasic@fdv.uni-lj.si` (Miha Matjašič),  
`marjan.cugmas@fdv.uni-lj.si` (Marjan Cugmas), `ales.ziberna@fdv.uni-lj.si` (Aleš Žiberna)

While blockmodeling may entail several different methods, the focus of this paper is on generalized blockmodeling of binary and valued networks using the blockmodeling package for the R programming language (R Core Team, 2018).

The structure of this paper is as follows: In Section 2, we describe blockmodeling. In Section 3, we describe the R package blockmodeling and in the Section 4, we provide examples of the package use, while in Section 5, we summarise the main functionalities of the package.

## 2. Blockmodeling

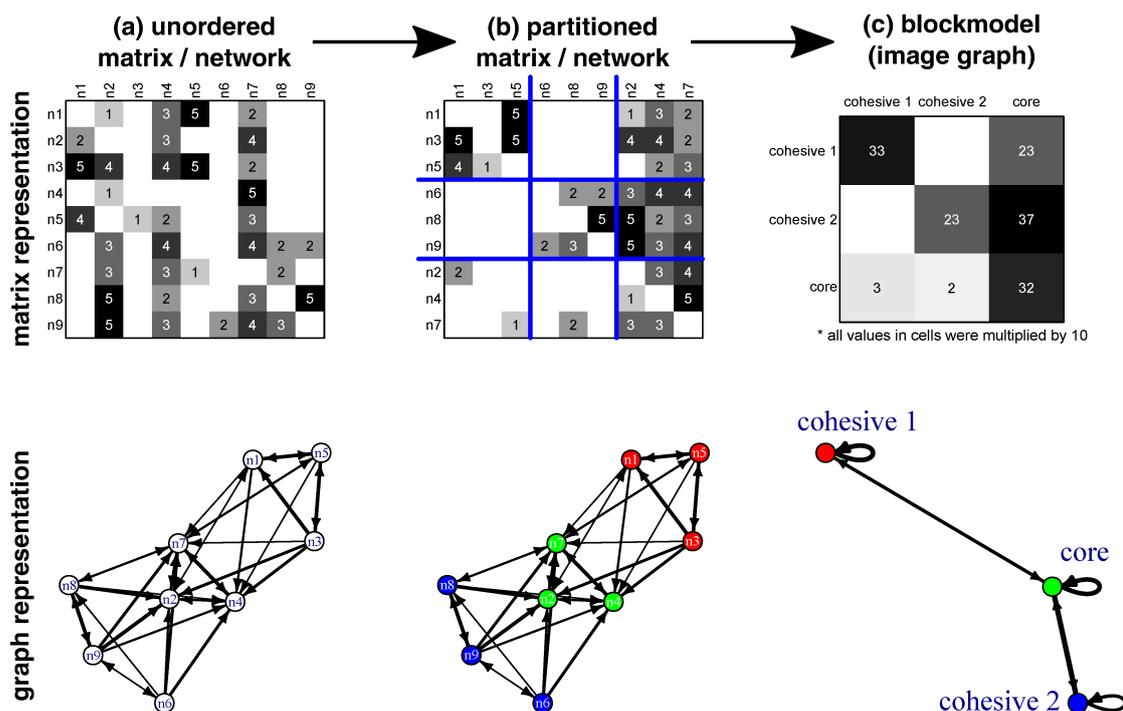
Blockmodeling is a set of approaches for partitioning nodes into clusters (also called positions) and simultaneously partitioning the links into blocks which are defined by the obtained clusters (Lorrain and White, 1971; Batagelj et al., 2004; Žiberna, 2007). A block is a submatrix showing the links between nodes from the same or different clusters.

The concept of blockmodeling is presented in Figure 1 where the illustrative valued network is shown in both matrix form and as a graph (Figure 1a). The units are ordered in rows and columns according to their names ( $n_1, n_2, n_3, \dots$ ). The units are then partitioned by considering the weights such that those with similar patterns of links are partitioned into the same clusters. The network is consistently represented in matrix form in accordance with the clusters obtained in matrix form so that units from the same clusters are placed next to each other and different clusters are separated by blue lines (Figure 1b). Nodes of the same cluster are coloured using the same colour in the corresponding graphic visualization. Then nodes from the same clusters are shrieked and represented as nodes of a blockmodel, which is shown as both a matrix and a graph (Figure 1c). The block densities are provided in the matrix. These summarize the strength of the relationship within and between the clusters. It can be seen that one core cluster and two cohesive clusters were identified. The core cluster is linked with both cohesive clusters (and vice versa) whereas the cohesive clusters are not linked to each other.

The nodes are clustered according to some notion of equivalence (Wasserman and Faust, 1994). The most commonly used are structural equivalence (Lorrain and White, 1971) and regular equivalence (White and Reitz, 1983), both originally defined for binary networks (Žiberna, 2007). Two nodes are structurally equivalent if they are identically linked to the rest of the network (and to themselves), while the nodes are regularly equivalent if they are connected in the same way to equivalent others. Regular equivalence is a generalization of structural equivalence. While analysing valued networks, regular equivalence should be replaced by  $f$ -regular equivalence, where  $f$  refers to any function, such as sum, max or mean (Žiberna, 2007).

In practice, structural equivalence is probably the most commonly used type of equivalence (Žnidaršič, Ferligoj and Doreian, 2012). At the same time, regular equivalence has never achieved widespread use (Žiberna, 2013), especially because it is rarely present in empirical data (Boyd and Jonas, 2001) and very sensitive to small changes in the network (Žnidaršič, Ferligoj and Doreian, 2012). Concerns have also been voiced about regular equivalence's applicability to social theory (Boyd, 2002).

In terms of generalized blockmodeling, a chosen type of equivalence defines the possible block types (and vice versa, i.e., the allowed block types in generalized blockmodeling imply the type of equivalence). For example, when binary networks are analysed, and structural equivalence is used, only null (ideally there are no links) and complete (ideally there are all possible links) blocks are possible, while with regular equivalence null, complete and regular



**Figure 1:** An example of a network and its corresponding blockmodel

(where at least one tie exists in each row and column) blocks are possible. The block types which are most often used are listed (with their descriptions for different blockmodeling approaches) in Table 1.

While the type of equivalence selected defines the possible block types, in some blockmodeling approaches a researcher can specify the block types allowed instead of a single type of equivalence. In this case, so-called generalized equivalence is considered in the blockmodeling procedure. Generalized equivalence is namely defined by block types and possibly their position in the blockmodel. Since different block types allow different patterns of links within the blocks, generalized equivalence may be seen as a way of specifying custom equivalences rather than as a single equivalence.

It often happens in empirical networks that some links are inconsistent with the ideal block types (e.g. some links appear in null blocks). The number of such links (and the amount of inconsistency) is reflected by the value of a criterion function (often called error or inconsistency), which measures the deviation of the actual blocks from the corresponding ideal blocks according to the model, while also being sensitive to the type of equivalence chosen (Doreian, Batagelj and Ferligoj, 2005; Žiberna, 2007).

### 2.1. Approaches to blockmodeling

Several approaches to blockmodeling have been proposed over the last century. They may be arranged in two classes: (i) deterministic blockmodeling approaches; and (ii) stochastic blockmodeling approaches (Holland, Laskey and Leinhardt, 1983; Snijders and Nowicki, 1997; Peixoto, 2020; Funke and Becker, 2019). The latter are not discussed in this paper because they are implemented in another software (e.g. the blockmodels package (INRA and Leger, 2015) or the dynsbm package (Matias and Miele, 2020)). Deterministic blockmodeling approaches can be further split into direct and indirect blockmodeling approaches (Doreian et al., 2005). (Indirect) conventional blockmodeling approaches are described in this paper along with generalized blockmodeling approaches, which are a special instance

**Table 1:** Characterisations of ideal blocks (Žiberna, 2007)

Ideal block name	Description for binary blockmodeling	Description for valued blockmodeling	Description for homogeneity blockmodeling
null	all 0 <sup>a</sup>	all 0 <sup>b</sup>	all 0 <sup>c</sup>
complete	all 1 <sup>d</sup>	all values at least m <sup>d</sup>	all equal <sup>e</sup>
row-dominant	an all 1 row exists <sup>d</sup>	a row where all values are at least m exists <sup>d</sup>	a row exists where values are all equal <sup>c</sup>
col-dominant	an all 1 column exists <sup>d</sup>	a column where all values are at least m exists <sup>d</sup>	a column exists where all values are equal <sup>c</sup>
row(-f)-regular	at least one 1 in each row exists <sup>d</sup>	the f over each row is at least m <sup>d</sup>	f over all rows equal
column(-f)-regular	at least one 1 in each column exists	the f over each column is at least m	f over all columns equal
(f)-regular	at least one 1 in each row and each column exists	the f over each row and each column is at least m	f over all rows and all columns separately equal
row-functional	exactly one 1 in each row exists	exactly one tie with value at least m in each row exists, all other 0	max over all rows equal, all other values 0
column-functional	exactly one 1 in each column exists	exactly one tie with value at least m in each column exists, all other 0	max over all rows equal, all other values 0

<sup>a</sup> An exception may be cells on the diagonal, their values should all be equal to 1.

<sup>b</sup> An exception may be cells on the diagonal, their values should all be least m.

<sup>c</sup> An exception may be cells on the diagonal, their values should be equal.

<sup>d</sup> Diagonal, their values should all be equal to 0.

<sup>e</sup> Cells on the diagonal may be treated separately - their values should all be equal, however they can be different from the values of the off-diagonal cells.

of the direct blockmodeling approach. Use of the blockmodeling package is demonstrated for (direct) generalized blockmodeling only.

Some non-generalized direct blockmodeling approaches (for non-signed and signed networks) are implemented in the dBlockmodeling package (Brusco, 2020), while generalized blockmodeling for binary networks and some direct approaches for signed networks are implemented in Pajek (Batagelj et al. 2004).

### 2.1.1. Conventional blockmodeling

Conventional blockmodeling (Doreian et al., 2005) is an indirect approach involving two steps: (i) obtaining a dissimilarity matrix on the nodes using a dissimilarity measure which is consistent with the type of equivalence selected (e.g., corrected Euclidean distance (Batagelj, Ferligoj and Doreian, 1992) for structural equivalence); and (ii) clustering the nodes with a hierarchical clustering method (e.g. Ward's agglomerative clustering method (Ward, 1963)), based on the dissimilarity matrix obtained. Since the second step is well supported by other R packages, the blockmodeling package only provides functions for computing (dis)similarity matrices according to structural equivalence (`sedist` function) and regular equivalence (REGE function and other functions).

### 2.1.2. Generalized blockmodeling

With generalized blockmodeling, a blockmodel is directly obtained from the network data by optimizing a criterion function, typically with a relocation algorithm (Batagelj et al., 1992). Different types of equivalences and/or block types can be specified.

Generalized blockmodeling holds several advantages over conventional blockmodeling (Doreian, 2006; Doreian et al., 2005; Batagelj et al., 2006): (i) since the direct approach already includes the criterion function in the process of optimizing partitions, at least a locally optimal solution will be obtained with the generalized approach; (ii) the partitions obtained by generalized blockmodeling frequently outperform those obtained with the conventional approach (at least in the case of structural and regular equivalence); (iii) conventional blockmodeling has mainly been used in an inductive way, meaning that researchers have accepted what was delineated through the clustering procedure. Yet, researchers often possess some prior knowledge about the global network structure that can be included in the blockmodel's specification.

Examples of generalized blockmodeling use are found in Doreian et al. (2005), Mrvar and Doreian (2009), Cugmas, Ferligoj, and Kronegger (2016) and Cugmas et al. (2020). These examples include social relations in working settings, classroom networks, political unit networks, scientific collaboration and citation networks, sport networks and other types of networks.

In this paper, the following types of generalized blockmodeling are considered:

- (i) Generalized binary blockmodeling, which is intended for analysing binary networks. The binary blockmodeling concept is presented thoroughly in Doreian et al. (2005).
- (ii) Generalized valued blockmodeling, which was developed because earlier researchers were converting valued networks into binary networks and analysing them as binary networks. The binarization was accomplished by recoding values above (or equal to) a certain threshold (often 1) into 1s and the other into 0s (see Doreian et al., 2005), which however caused a loss of considerable amount of information. The valued blockmodeling approach reduces the amount of information lost, although some loss may still occur.

Valued blockmodeling may be seen as an extension of binary blockmodeling. It extends the equivalence relations and thereby the definitions of possible block types by replacing the stipulations for 1 with analogous stipulations for the value  $m$  (the minimal value that characterizes the tie between a unit and either a cluster or another unit such that this tie satisfies the condition of the block). Therefore, the criterion function used in the valued blockmodeling measures block inconsistencies as the deviation of appropriate values from either 0 or  $m$  (Žiberna, 2007).

- (iii) Generalized homogeneity blockmodeling, which is based on the idea that blocks should be as homogeneous as possible with respect to some property. Accordingly, the inconsistencies of an empirical block with respect to its ideal block are measured by the within-block variability of appropriate values.

One of the two variability criteria can be used: the sum of the squared deviations from the mean or the sum of absolute deviations from the median (Žiberna, 2007).

## 2.2. Prespecified blockmodeling

A researcher can consider a prior knowledge concerning the ties among the clusters while conducting blockmodeling (Doreian *et al.*, 2005). This may be done by specifying not only the number of clusters and allowed block types (the same for all blocks), but also by specifying the allowed block types for each block separately. Typically, only one block type is specified as allowed for at least some blocks.

## 3. Package description

A stable version of the R package `blockmodeling`<sup>1</sup> is available from the Comprehensive R Archive Network (CRAN) at <https://cran.r-project.org/web/packages/blockmodeling> while test versions are available from the R-Forge at [https://r-forge.r-project.org/R/?group\\_id=203](https://r-forge.r-project.org/R/?group_id=203). The package has been around since 2007 and is currently written in the programming languages R, C and Fortran. In this paper, version 1.0.0 is used.

The package supports generalized and indirect blockmodeling. For generalized blockmodeling, one-mode, two-mode and multilevel networks (also linked networks (Žiberna, 2020)) with one or more relations are supported. However, for purpose of clarity and simplicity, this paper is limited to generalized blockmodeling of one-mode single relational networks.

To obtain a generalized blockmodeling solution, a researcher might want to use the function `optRandomParC`, which optimizes a specified number of randomly generated partitions based on the criterion function selected (to optimize a single partition, a researcher can use the function `optParC` that optimizes only the supplied (one) partition, although so as to avoid a local minimum this is not recommended). The main arguments of the function are:

- `M`: an adjacency matrix representing the (usually valued) network.
- `k`: the number of clusters.

---

<sup>1</sup>The blockmodeling package leverages functions from a variety of other packages. Key computations use `stats` (R Core Team, 2019a), `methods` (R Core Team, 2019b), `Matrix` (Bates and Maechler, 2019), `parallel` (R Core Team, 2019c) and others.

- `approach`: the chosen generalized blockmodeling approach; 'bin' for generalized binary blockmodeling, 'val' for generalized valued blockmodeling and 'hom' for homogeneity blockmodeling.
- `regFun`: the function  $f$  specifies regular block types (e.g., max-regular block when `regFun = 'max'`). The function is only relevant when the  $f$ -regular blocks are specified by the argument blocks.
- `blocks`: a vector with the names of allowed block types. At least two must be specified for binary and valued blockmodeling. Possible types are: null ('nul'), complete ('com'), regular ('reg'), column-(function) regular ('cre') and row-(function) regular block ('rre'). In the case of binary and valued blockmodeling, a researcher can also specify column-dominant block ('cfn') and row-dominant block ('rfn') and with valued blockmodeling a researcher can also specify average block ('avg'). The option "do not care" ('dnc') is also available. When pre-specification is used, the argument is stated in the form of an array, as shown in the example section.
- `rep`: the number of different starting partitions.
- `nCores`: the number of physical CPU cores to be used. All available but one physical CPU cores are used when `nCores = 0`.
- `preSpecM`: the value  $m$  must be specified only in the case of generalized valued blockmodeling.

To calculate only the value of a criterion function, a researcher can use the function `critFunC`. The same arguments apply as for the case of the function `oprRandParC`, except that `k` is replaced by a partition (a vector) `clu` and the arguments `rep` and `nCores` are omitted.

Once a blockmodel and partition have been obtained by either `optParC` or `optRandomParC`, a researcher can use the function `IM` to extract a blockmodel, the function `clu` to extract an obtained partition or the function `err` to extract the value of a criterion function.

The package contains some other handy functions such as `funByBlocks` (which computes the value of a function (mean by default) over blocks of a matrix defined by a partition), `plotMat` (which plots a network in matrix form by considering the corresponding partition) and functions for computing the adjusted and original Rand Index (e.g. `crand2`). A plot method that internally calls `plotMat` is available for S3 classes returned by `optParC` and `optRandomParC`.

#### 4. Demonstration of the package use

The use of various generalized blockmodeling approaches is illustrated using the Baker citation network data (Baker, 1992). Here, the nodes represent journals from the field of social work (the 20 journals listed in Table 2). There is an arc from journal  $i$  to journal  $j$  if journal  $i$  cited journal  $j$ . The values on the arcs correspond to the number of citations in 1985.

The data can be loaded from the package `blockmodeling` using `data('baker')`. The diagonal values, representing the number of citations by papers from the same journals, are replaced with 0s (`diag(baker) <- 0`). The network can be visualized with the function `plotMat`. Since the partition is not yet obtained, the `clu` argument is not set.

**Table 2:** Journals in Social Work Citation Network

Label	Journal
AMH	Administration in Mental Health
ASW	Administration in Social Work
BJSW	British Journal of Social Work
CAN	Child Abuse and Neglect
CCQ	Chile Care Quarterly
CW	Child Welfare
CYSR	Children and Youth Services Review
CSWJ	Clinical Social Work Journal
FR	Family Relations
IJ6W	Indian Journal of Social Work
JGSW	Journal of Gerontological Social Work
JSP	Journal of Social Policy
JSWE	Journal of Social Work Education
PW	Public Welfare
SCW	Social Casework
SSR	Social Services Review
SW	Social Work
SWG	Social Work with Groups
SWHC	Social Work in Health Care
SWRA	Social Work Research and Abstracts

```
plotMat(baker, main = 'Baker Network Data',
        mar = c(1, 1, 3, 1), title.line = 2)
```

Figure 2 is obtained with the function `plotMat`. To make the plot easier to read, the cell values are automatically multiplied by the factor (in this case 0.1) which (by default) places their absolute values in the range [0, 100). The factor by which the values are multiplied is automatically selected and reported, as noted below the plot.

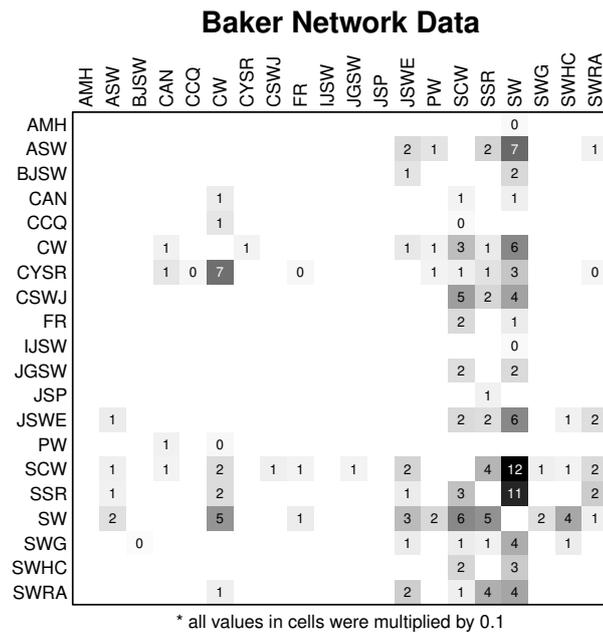
It is immediately apparent from Figure 2 that the network is relatively sparse, meaning the journals did not tend to cite each other. However, the highest number of citations extends from SCW to SW and from SSR to SW. The latter journal is also the one which cited the highest number of other journals.

#### 4.1. Binary blockmodeling

To analyse valued networks by using generalized blockmodeling for binary networks, a re-researcher must binarize the valued network and for this adopt one of several ways, such as keeping all of the arcs with values greater than 0.

```
bakerBinar <- baker
bakerBinar[bakerBinar > 0] <- 1
```

In all of the following examples, the function `optRandomParC` is used. The number of clusters is set to 2 or 3. The number of clusters is chosen arbitrarily by examining multiple partitions with different numbers of clusters (2 or 3 seem to be the most appropriate) (also see Doreian *et al.*, 2005). The 1000 randomly generated partitions are optimized and multiple cores are used. For binary blockmodeling, the `approach` argument must be set to 'bin'.



**Figure 2:** Baker network data in matrix form

#### 4.1.1. Structural equivalence

If structural equivalence is used, only null and complete block types are possible. Therefore, a vector `c('nul', 'com')` is provided to blocks (the structural equivalence is set by the vector of the allowed block types).

```
resBinStr <- optRandomParC(M = bakerBinar, k = 3, rep = 1000,
  nCores = 0, blocks = c('nul', 'com'), approach = 'bin')
```

The number of errors (inconsistencies) of the blockmodel then obtained is 47 (accessed via the function `err`). The obtained partition can be accessed with the function `clu` while the blockmodel can be seen in the form of an image matrix using the function `IM`. The image matrix specifies the block types by blocks. The function `IM` shows the image matrix obtained (not the specified one) with blockmodeling.

```
IM(resBinStr)

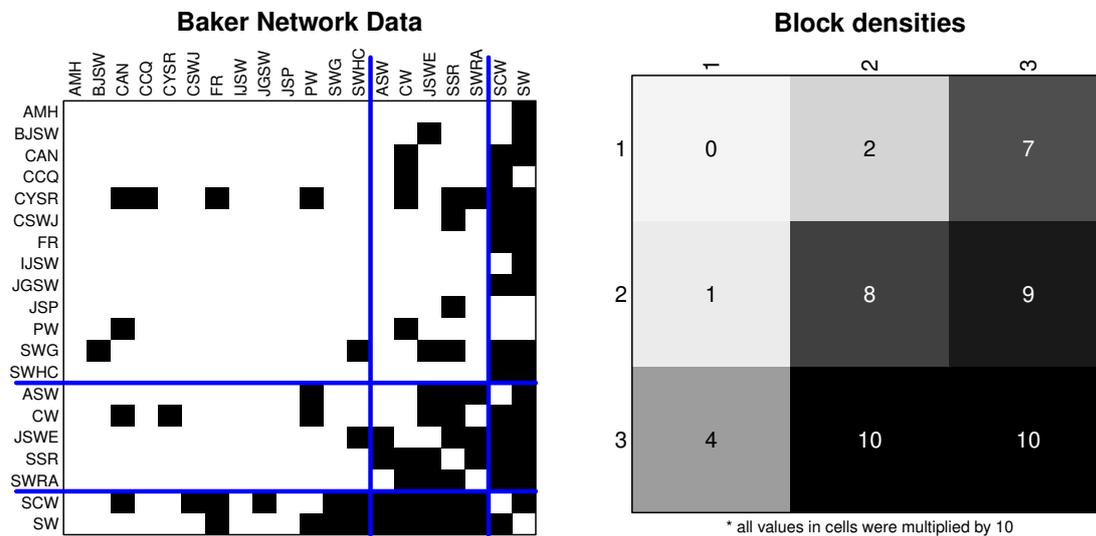
      [,1] [,2] [,3]
[1,] "nul" "nul" "com"
[2,] "nul" "com" "com"
[3,] "nul" "com" "com"
```

The image matrix shows the journals in cluster 2 and cluster 3 cite each other both within and between the clusters. Journals in cluster 1 do not cite each other in general, but they cite journals in cluster 3. Cluster 3 can be identified as the most central cluster while cluster 1 as a peripheral cluster because in this cluster the journals are generally not cited much by other journals.

The block densities can be calculated with the function `funByBlocks` and visualized with the function `plotMat`. Finally, the empirical network can be visualized in matrix form and in line with the blockmodeling solution that is obtained. When using the func-

tion `plotMat`, the obtained partition has to be provided to the function by the argument `clu`. The latter is not necessary when using the function `plot` (the `S3` method exists for the `optMorePar` class that is returned by the `optRandomParC` function), as shown below. The clusters of journals obtained are separated by lines in Figure 3.

```
plot(resBinStr, main = 'A Baker Network Data',
     mar = c(1, 2, 3, 1), title.line = 2)
```



**Figure 3:** Matrix representation of the network of journals partitioned into 3 clusters using binary blockmodeling with structural equivalence and the corresponding block densities

It can be seen in Figure 3 (left) that the block densities are lowest in the null blocks, as expected. Among the null blocks, however, the density is highest in the block belonging to the link (citing) from cluster 3 to cluster 1, which reflects a tendency for reciprocity.

The most central cluster (cluster 3) only consists of two journals, SCW and SW, while cluster 2 contains the following journals: ASW, CW, JSWE, SSR and SWRA. All other journals are located in the peripheral cluster.

#### 4.1.2. Regular equivalence

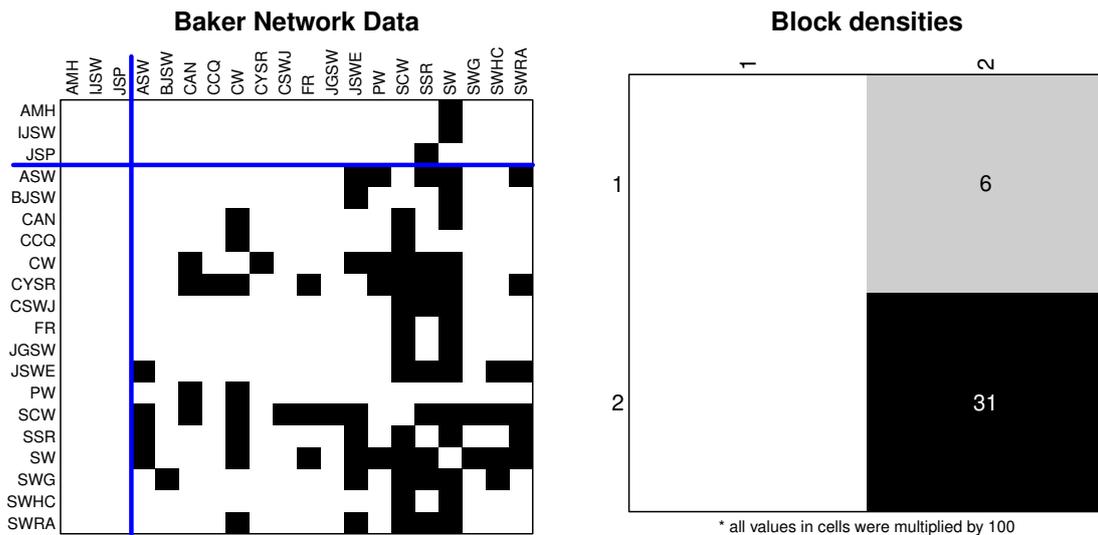
Here regular equivalence is used and the number of clusters is set to 2. The regular equivalence is specified in the function `optRandomParC` by adding a regular block type among the possible block types.

```
resBinReg <- optRandomParC(M = bakerBinar, k = 2, rep = 1000,
                          nCores = 0, blocks = c('nul', 'com', 'reg'),
                          approach = 'bin')
```

The partitioned matrix in Figure 4 shows that a small cluster (cluster 1) of journals exists that are not cited by any journal. The three journals in this cluster are AMH, IJSW and JSP. The first two journals cited SW while JSP cited SSR (all of the cited journals are in cluster 2). These citations represent inconsistent links (`err(resBinReg)`).

The similarity of the obtained partitions can be measured with the Adjusted Rand Index (Rand, 1971; Hubert and Arabie, 1985), where the expected value is 0 in the case of two random partitions and the maximum value of the measure is 1 (in the event of two identical partitions).

```
crand2(clu1 = clu(resBinStr), clu2 = clu(resBinReg))
```

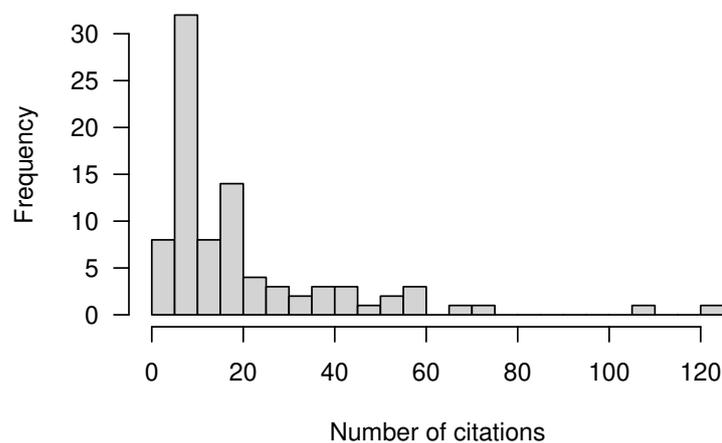


**Figure 4:** The network of journals partitioned into 2 clusters using binary blockmodeling with regular equivalence and the corresponding block densities

The value  $-0.12$  confirms what is seen when comparing Figure 3 and Figure 4, i.e. that the partitions obtained (by using structural equivalence vs. regular equivalence) are very different.

#### 4.2. Valued blockmodeling

The main dilemma in valued blockmodeling is how to determine the most appropriate value of  $m$ . The best approach is to choose a value based on prior knowledge about how high the value of a tie should be for it to be considered as strong or relevant. In the absence of such prior knowledge, a researcher may refer to one of the guidelines provided by Žiberna (2007) or select the most appropriate  $m$  based on the distribution of all tie values (Figure 5).



**Figure 5:** Distribution of the number of citations among the journals

Here,  $m$  is set to a median value (only values greater than 0 are taken into account), which is 13.

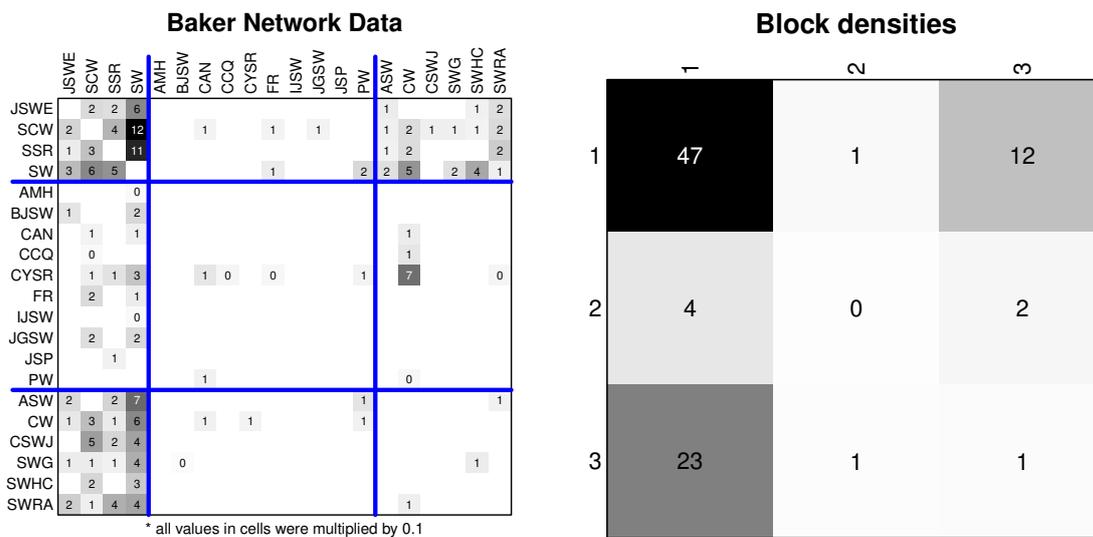
##### 4.2.1. Structural equivalence

To apply blockmodeling of valued networks, a researcher must set `approach = 'val'` and specify the value  $m$  by setting the argument `preSpecM`. In addition, the allowed block

types (indirectly the type of equivalence) and number of clusters must be specified. The number of clusters is set to three.

```
resValStr <- optRandomParC(M = baker, k = 3, rep = 1000,
  preSpecM = 13, approach = 'val', blocks = c('nul', 'com'),
  nCores = 0)
```

It can be seen in Figure 6 that cluster 1 and cluster 3 form a very clear symmetric core-periphery global network structure since the journals in cluster 1 (core; JSWE, SCW, SSR and SW) mutually cited each other and also cited those in cluster 3 (periphery; ASW, CW, CSWJ, SWG, SWHC and SWRA). Another internally non-linked cluster (cluster 2) of journals exists. Some journals in this cluster cited the journals in the core cluster.



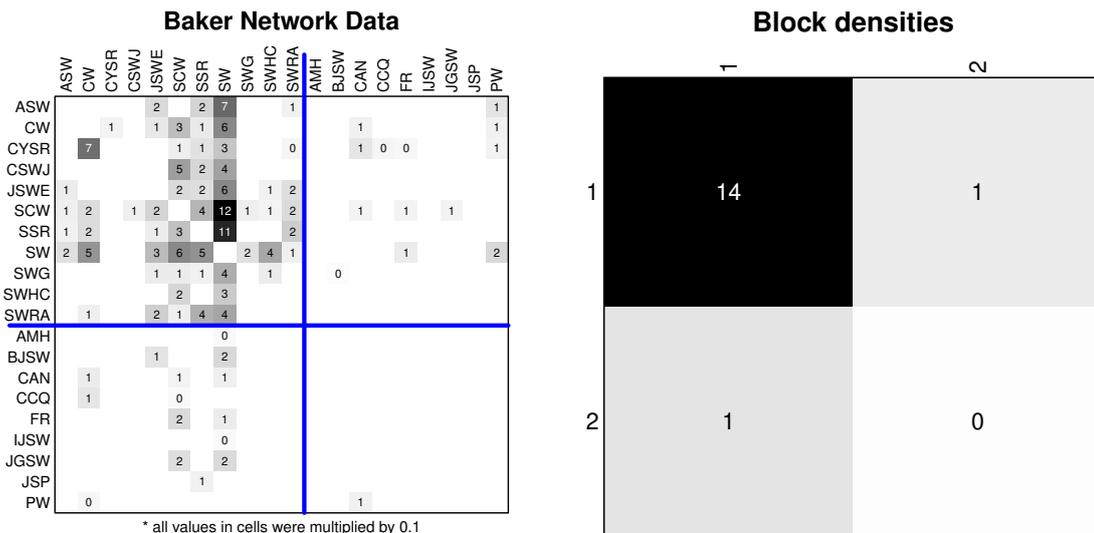
**Figure 6:** The network of journals partitioned into 3 clusters using valued blockmodeling ( $m = 13$ ) with structural equivalence and the corresponding block densities

#### 4.2.2. Regular equivalence

In the case of valued blockmodeling with regular equivalence, a researcher must select the function  $f$  to specify the type of  $f$ -regular blocks. This is set to max by default, although it can also be set to sum, mean or other functions. Here, the max-regular block type is to be allowed and therefore the argument `regFun` within the function `optRandomPar` is set to 'max'. A regular block type is added to the vector of allowed block types. The number of clusters is arbitrarily set to two clusters.

```
resValReg <- optRandomParC(M = baker, k = 2, rep = 1000,
  preSpecM = 13, approach = 'val',
  blocks = c('nul', 'com', 'reg'), nCores = 0, regFun = 'max')
```

The blockmodel (image matrix) that is obtained is the same as that obtained by binary blockmodeling with regular equivalence, but the sizes and obtained partitions are different with a value of the Adjusted Rand Index of 0.1. There are more links in null blocks (compared to binary blockmodeling with structural equivalence), but the corresponding link values are relatively low. Consequently, the size of the cluster with the journals that are less cited (and cite less) is bigger.



**Figure 7:** The network of journals partitioned into 2 clusters using valued blockmodeling ( $m = 13$ ) with max regular equivalence and the corresponding block densities

### 4.3. Homogeneity blockmodeling

The homogeneity blockmodeling approach’s advantage over the valued blockmodeling approach is that no parameters (such as the binarization threshold or parameter  $m$ ) need to be set. Therefore, it is very well suited as a preliminary or the main approach to valued networks when no prior knowledge about these values is available. Homogeneity blockmodeling emphasizes the similarity of tie strengths within blocks over the pattern of ties.

#### 4.3.1. Structural equivalence

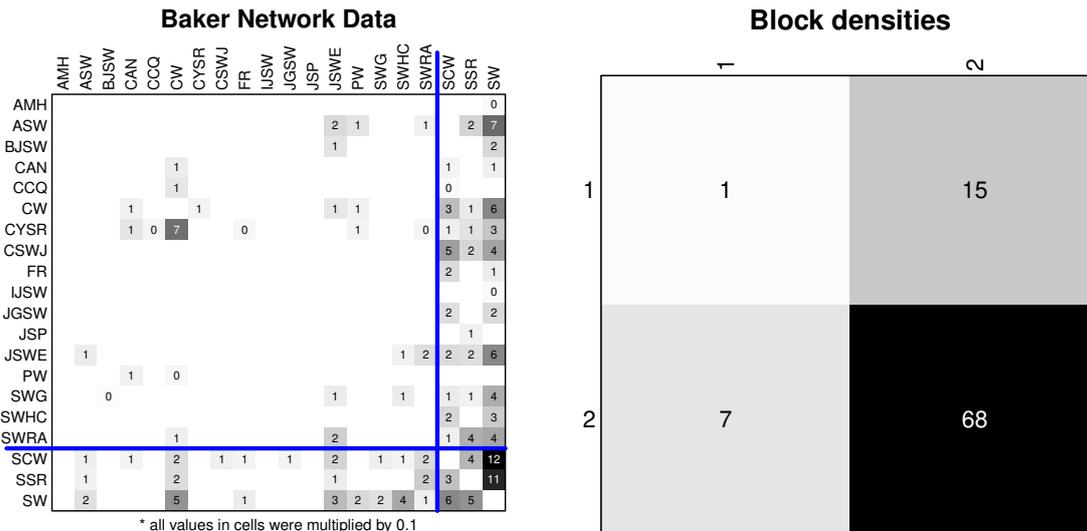
To use homogeneity blockmodeling, the approach argument must be set to 'hom'. To apply sum of squares homogeneity blockmodeling, the homFun argument must be set to 'ss' while, to apply absolute deviation blockmodeling, the argument must be set to 'ad' in the optRandomParC function.

Because the computation of inconsistencies is very similar for sum of squares and absolute deviations blockmodeling, application of the first approach is only shown here.

```
resHomSSStr <- optRandomParC(M = baker, k = 2, rep = 1000,
  approach = 'hom', homFun = 'ss', blocks = c('nul', 'com'),
  nCores = 0)
```

Usually, the image matrix is not of interest in the case of homogeneity blockmodeling because the null blocks are a special case of complete blocks and thus only classified as null when the mean of the block is exactly 0, which rarely happens in practice. Instead, blocks with low block means are interpreted as null blocks (see Žiberna (2013) for another way of identifying null blocks).

The results shown in Figure 8 suggests the global network structure of the journal citation network can be characterized as a symmetric core-periphery structure. Here, the core cluster is cluster 2 because the corresponding journals (SCW, SSR and SW) not only cited each other, but also cited and were cited by other journals (according to the block densities, the peripheral journals cited the core journal more often than the other way around). All other journals are located in the peripheral cluster with a very few citations found within the cluster.



**Figure 8:** The network of journals partitioned into 3 clusters using homogeneity blockmodeling (sum of squares) structural equivalence and the corresponding block densities

#### 4.3.2. Regular equivalence

To apply blockmodeling with homogeneity regular equivalence, the regular block type must be added to the vector of possible block types in the function `optRandomParC` and the `f` function must be defined, e.g. 'max', as an argument of `regFun`.

```
resHomSSReg <- optRandomParC(M = baker, k = 2, rep = 1000,
  approach = 'hom', blocks = c('nul', 'com', 'reg'),
  regFun = 'max', nCores = 0)
```

Given that the partition and blockmodel which are obtained are the same as those in the case of structural equivalence, they are not interpreted.

#### 4.4. Pre-specified blockmodeling

A blockmodel can be fully or partially specified (see the subsection Prespecified blockmodeling). The following gives an example of the use of pre-specified blockmodels.

In the case of a journal citation network, a researcher might possess prior knowledge that the global network structure is symmetric core-periphery, i.e. there are some journals (the core) which are cited by most journals, while other journals (the periphery) cite journals in the core and not those in their own cluster. Therefore, the pre-specified blockmodel may be represented by the following image matrix:

```
preImageReg <- rbind(c('com', 'reg'), c('reg', 'nul'))
```

Here, the blocks connecting the core and the periphery are of the regular type. Alternatively, a researcher can assume these blocks can be of the regular or complete type. When this is the case, the image matrix must be specified as an array.

```
preImageRegCom <- array(NA, dim = c(2, 2, 2))
preImageRegCom[1,,] <- rbind(c('com', 'reg'), c('reg', 'nul'))
preImageRegCom[2,,] <- rbind(c('com', 'com'), c('com', 'nul'))
```

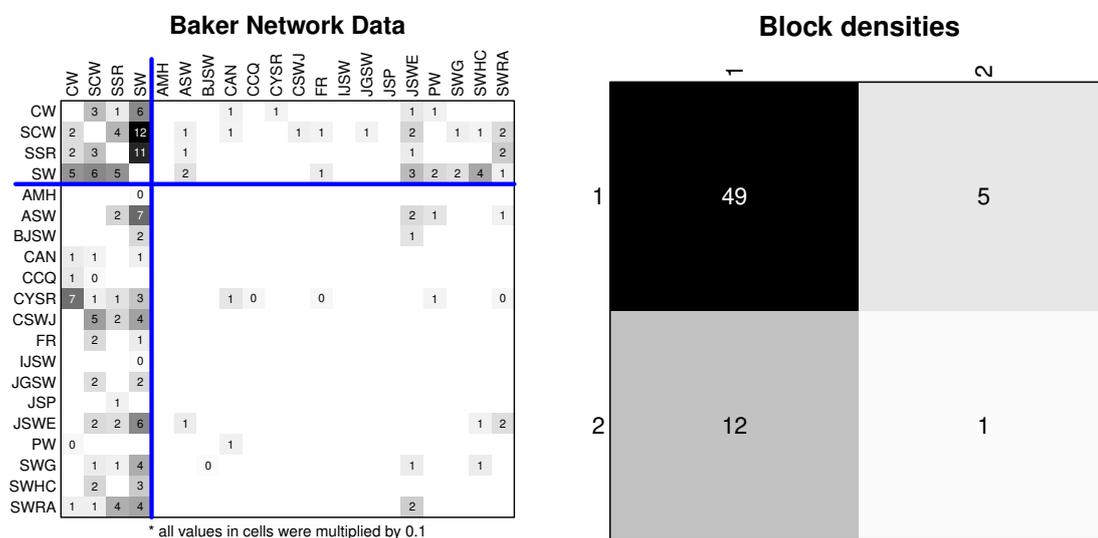
To apply pre-specified blockmodeling, the above matrix or array must be provided as the argument to `blocks` within the function `optRandomParC`. To apply valued blockmodeling with  $m = 13$ , the `approach` and `preSpecM` arguments must be set to 'val' and 13, respectively.

```
resValPre <- optRandomParC(M = baker, k = 2, rep = 1000,
  preSpecM = 13, approach = 'val', blocks = preImageRegCom,
  nCores = 0)
```

The obtained image matrix (blockmodel) is the following one,

```
IM(resValPre)
      [,1] [,2]
[1,] "com" "reg"
[2,] "reg" "nul"
```

indicating that the journals CW, SCW, SSR and SW are all part of a closely connected core (cluster 1) while other journals are classified in the periphery (cluster 2). The core and the periphery are connected with max-regular links and the density (Figure 9) is higher within the block that links periphery to the core than within the block that links the core to the periphery.



**Figure 9:** The network of journals partitioned into 2 clusters using homogeneity blockmodeling (sum of squares) max-regular equivalence and the corresponding block densities

## 5. Conclusion

Generalized blockmodeling is an approach for finding clusters of equivalent units in a network and for determining the ties among these units. As such, it is used to study global network structures and the (social) positions of the units. While generalized blockmodeling is also implemented in the Pajek software (Batagelj et al., 2004), the implementation of generalized blockmodeling in the blockmodeling package for the R programming language, which is presented in this paper, is the only one that supports also the blockmodeling of valued networks and the generalized blockmodeling of more complex networks (e.g., multilevel, multi-relational). In addition, it also supports some other blockmodeling approaches (indirect approach) besides generalized blockmodeling.

This paper demonstrates the use of the blockmodeling package for generalized blockmodeling of binary and valued one-mode networks on a real network data set, namely Baker’s data (Baker, 1992) set on citing among the journals. Based on the examples given,

it is clear that blockmodeling solutions can vary across different blockmodeling approaches, underlining the fact that a prior knowledge concerning the analysed networks is crucial, not only for the choice of the most appropriate blockmodeling approach, but also when it comes to interpreting the results obtained.

Ultimately, this paper, together with the package documentation, can serve as a basis for analysing more complex networks and further explorations of the package's capabilities.

## Acknowledgment

This research was financially supported by the Slovenian Research Agency (<http://www.arrs.si>) within the research program P5-0168 and the research project J7-8279 (Blockmodeling multilevel and temporal networks).

## References

- [1] Baker, D. (1992): A structural analysis of the social work journal network: 1985–1986. *Journal of Social Service Research*, **15**, 153–167.
- [2] Batagelj, V., Bock, H., Ferligoj, A., and Žiberna, A. (2006): *Data science and classification*. Berlin: Springer.
- [3] Batagelj, V., Doreian, P., Ferligoj A., and Kejžar, N. (2014): *Understanding large temporal networks and spatial networks: Exploration, pattern searching, visualization and network evolution*. New York, NY: John Wiley & Sons.
- [4] Batagelj, V., Ferligoj, A., and Doreian, P. (1992): Direct and indirect methods for structural equivalence. *Social Networks*, **14**, 63–90.
- [5] Batagelj, V., Mrvar, A., Ferligoj, A., and Doreian, P. (2004): Generalized blockmodeling with Pajek. *Metodološki zvezki*, **1**, 455–467.
- [6] Bates, D. and Maechler, M. (2019): Matrix: Sparse and dense matrix classes and methods. R package version 1.2-17. Retrieved from <https://cran.r-project.org/web/packages/Matrix/index.html>.
- [7] Borgatti, S.P. and Everett, M.G. (1992): Notions of position in social network analysis. *Sociological Methodology*, **22**, 1–35.
- [8] Boyd, J.P. (2002): Finding and testing regular equivalence. *Social Networks*, **24**, 315–331.
- [9] Boyd, J.P. and Jonas, K.J. (2001): Are social equivalences ever regular? Permutation and exact tests. *Social Networks*, **23**, 87–123.
- [10] Brusco, M. (2020): dBlockmodeling: Deterministic blockmodeling of signed, one-mode and two-mode networks. R package version 0.2.0. Retrieved from <https://CRAN.R-project.org/package=dBlockmodeling>.
- [11] Cugmas, M., DeLay, D., Žiberna, A., and Ferligoj, A. (2020): Symmetric core-cohesive blockmodel in preschool children's interaction networks. *PLOS ONE*, **15**, e0226801.

- [12] Cugmas, M., Ferligoj, A., and Kronegger, L. (2016): The stability of co-authorship structures. *Scientometrics*, **106**, 163–186.
- [13] Doreian, P. (2006): Some open problem sets for generalized blockmodeling. In V. Batagelj, H.-H. Bock, A. Ferligoj, A. Žiberna (eds.): *Data science and classification*, 119–130. Berlin: Springer.
- [14] Doreian, P., Batagelj, V., and Ferligoj, A. (2005): *Generalized blockmodeling. Structural analysis in the social sciences*. New York, NY: Cambridge University Press.
- [15] Funke, T. and Becker, T. (2019): Stochastic block models: A comparison of variants and inference methods. *PLOS ONE*, **14**, e0215296.
- [16] Holland, P.W., Laskey, K.B., and Leinhardt, S. (1983): Stochastic blockmodels: First steps. *Social Networks*, **5**, 109–137.
- [17] Hubert, L. and Arabie, P. (1985): Comparing partitions. *Journal of Classification*, **2**, 193–218.
- [18] INRA, L. J. B. (2015): blockmodels: Latent and stochastic block model estimation by a ‘V-EM’ algorithm. R package version 1.1.1. Retrieved from <https://CRAN.R-project.org/package=blockmodels>.
- [19] Lorrain, F. and White, C. H. (1971): Structural equivalence of individuals in social networks. *Journal of Mathematical Sociology*, **1**, 49–80.
- [20] Matias, C. and Miele, V. (2020): dynsbm: Dynamic stochastic block models. R package version 0.7. Retrieved from <https://CRAN.R-project.org/package=dynsbm>.
- [21] Mrvar, A. and Doreian, P. (2009): Partitioning signed two-mode networks. *Journal of Mathematical Sociology*, **33**, 196–221.
- [22] Peixoto, T. (2020). Bayesian Stochastic Blockmodeling. In P. Doreian, V. Batagelj, A. Ferligoj (eds.): *Advances in Network Clustering and Blockmodeling*, 289–332. New York, NY: Wiley.
- [23] R Core Team. (2018): R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. Retrieved from <https://www.R-project.org>.
- [24] R Core Team. (2019a): R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. Retrieved from <https://www.R-project.org>.
- [25] R Core Team. (2019b): R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. Retrieved from <https://www.R-project.org>.
- [26] R Core Team. (2019c): R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. Retrieved from <https://www.R-project.org>.

- [27] Rand, W. M. (1971): Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, **66**, 846–850.
- [28] Snijders, T. and Nowicki, K. (1997): Estimation and prediction for stochastic block-models for graphs with latent block structure. *Journal of Classification*, **14**, 75–100.
- [29] Ward, J. H. (1963): Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, **58**, 236–244.
- [30] Wasserman, S. and Faust, K. (1994): *Social network analysis: Methods and applications*. Cambridge: Cambridge University Press.
- [31] White, D. R and Reitz, K. P. (1983): Graph and semigroup homomorphisms on networks of relations. *Social Networks*, **5**, 193–234.
- [32] Žiberna, A. (2007): Generalized blockmodeling of valued networks. *Social Networks*, **29**, 105–126.
- [33] Žiberna, A. (2013): Generalized blockmodeling of sparse networks. *Metodološki zvezki*, **10**, 99–119.
- [34] Žiberna, A. (2020): Blockmodeling linked networks. In P. Doreian, V. Batagelj, A. Ferligoj (eds.): *Advances in network clustering and blockmodeling*, 267–287. New York, NY: John Wiley & Sons.
- [35] Žnidaršič, A., Ferligoj, A., and Doreian, P. (2012): Non-response in social networks: The impact of different non-response treatments on the stability of blockmodels. *Social Networks*, **34**, 438–450.