Generalized blockmodeling in R
using `blockmodeling` package

Aleš Žiberna and Marjan Cugmas

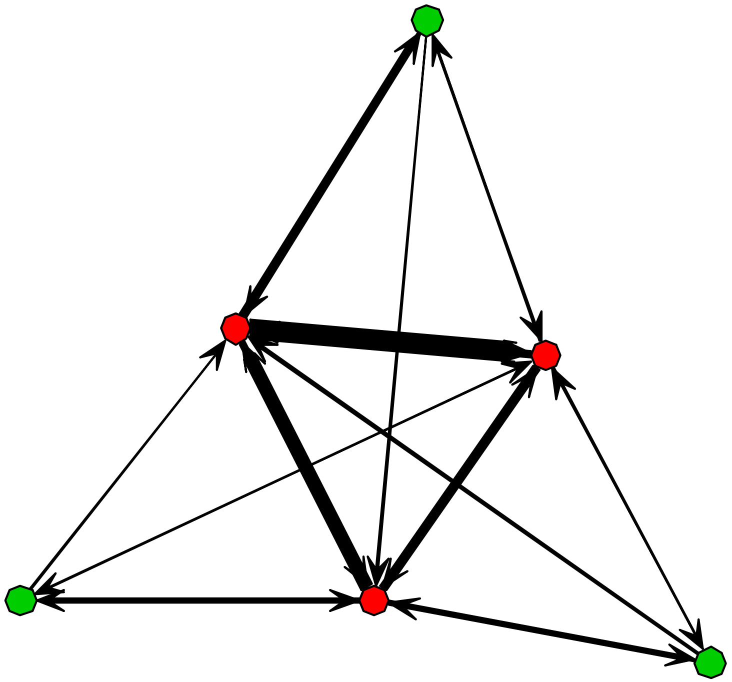
University of Ljubljana

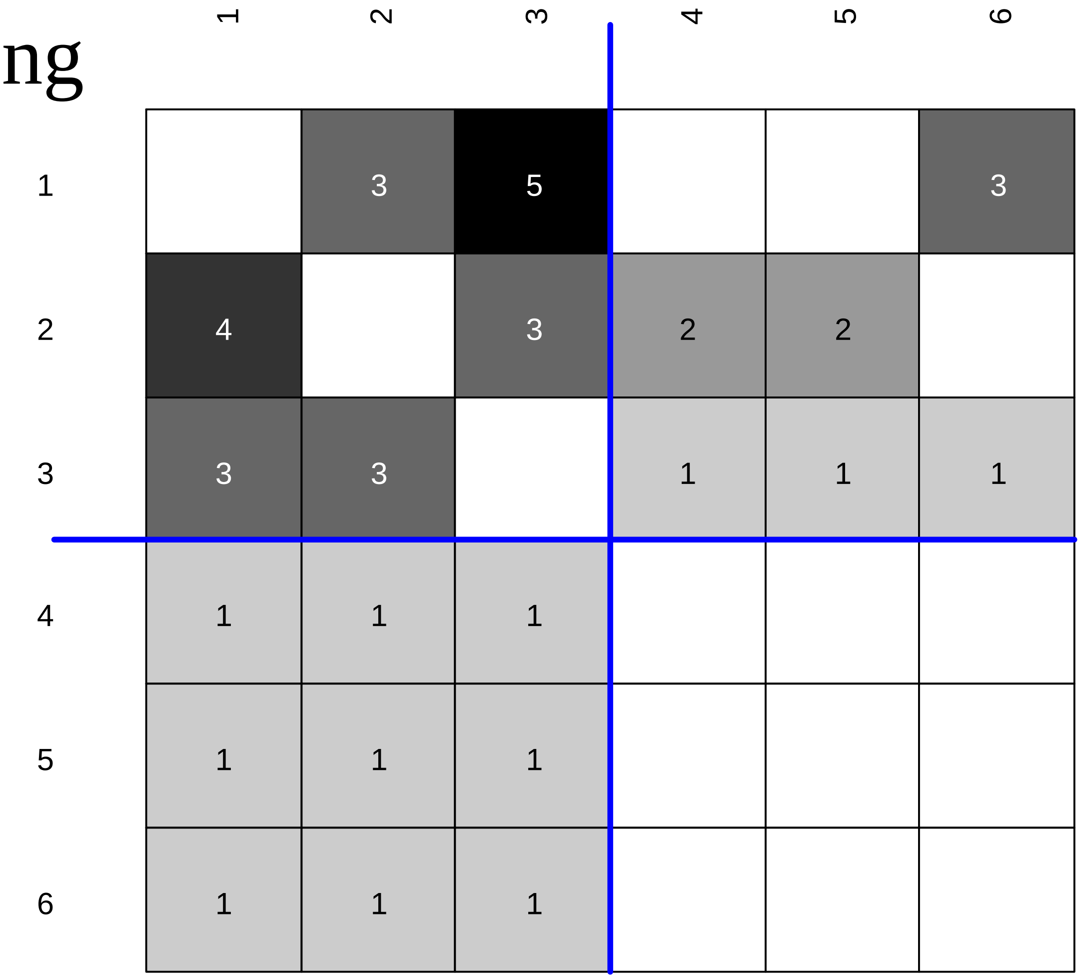
Workshop plan

- (Generalized) Blockmodeling – theory
- `blockmodeling` R package – main functions
- Main workshop part – guided usage of the package on Texas SAR network
- Your questions, working on own data or exercises.
- Extensions (time permitting)

Introduction

Blockmodeling is a group of methods for clustering units (and ties) in a network.





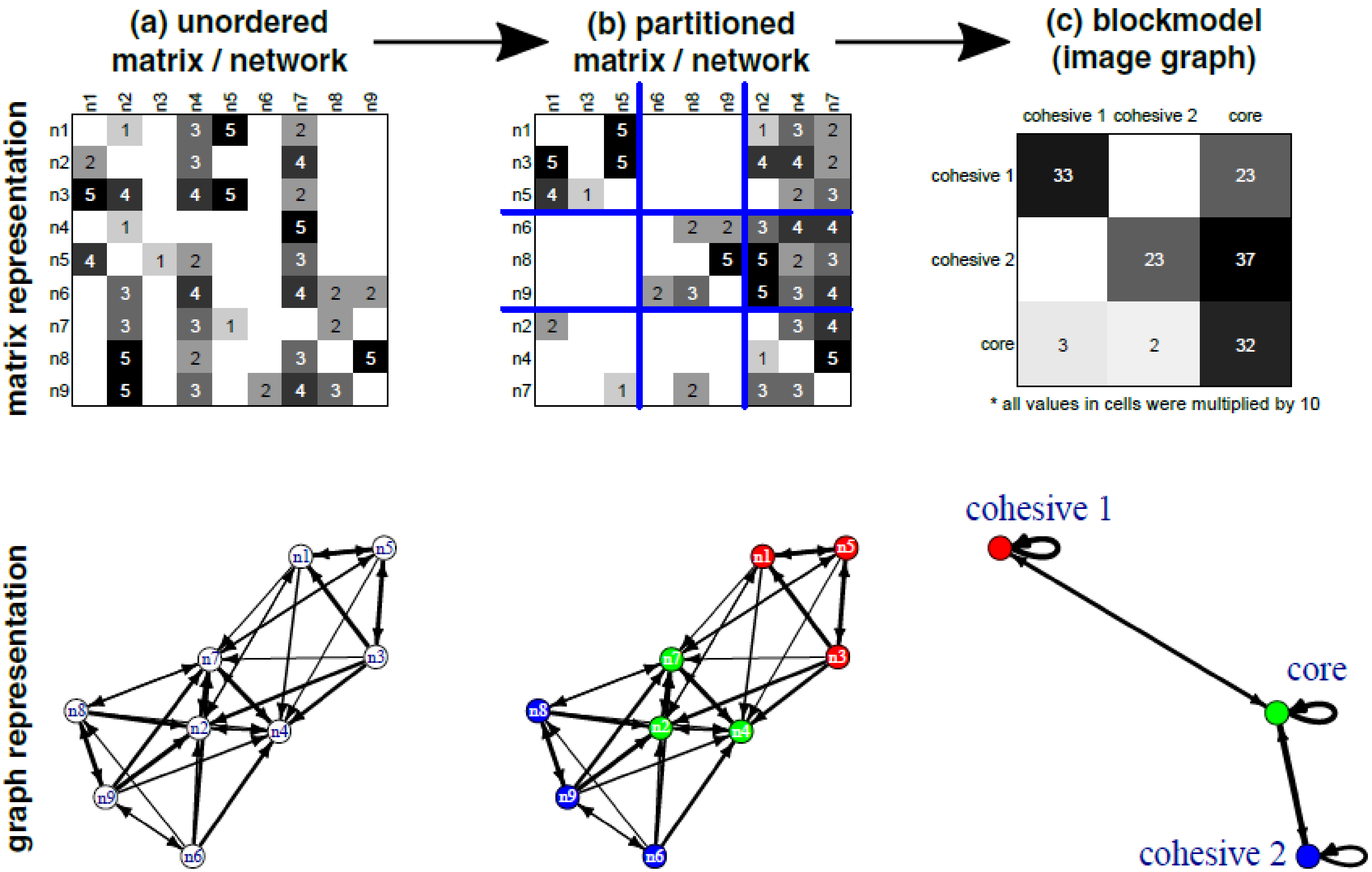
	1	2	3	4	5	6
1		3	5			3
2	4		3	2	2	
3	3	3		1	1	1
4	1	1	1			
5	1	1	1			
6	1	1	1			

Introduction

Blockmodeling is a metod for

- finding clusters of units in the network and, at the same time
- determining ties within and between these clusters.

Blockmodeling process



(a) unordered matrix / network

	n1	n2	n3	n4	n5	n6	n7	n8	n9
n1		1		3	5		2		
n2	2			3			4		
n3	5	4		4	5		2		
n4		1					5		
n5	4		1	2			3		
n6		3		4			4	2	2
n7		3		3	1			2	
n8		5		2			3		5
n9		5		3		2	4	3	

(b) partitioned matrix / network

	n1	n3	n5	n6	n8	n9	n2	n4	n7
n1			5				1	3	2
n3	5		5				4	4	2
n5	4	1						2	3
n6					2	2	3	4	4
n8						5	5	2	3
n9				2	3		5	3	4
n2	2							3	4
n4							1		5
n7			1		2		3	3	

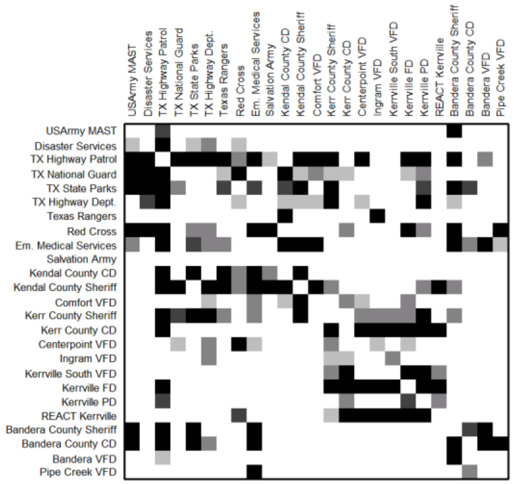
(c) blockmodel (image graph)

	cohesive 1	cohesive 2	core
cohesive 1	33		23
cohesive 2		23	37
core	3	2	32

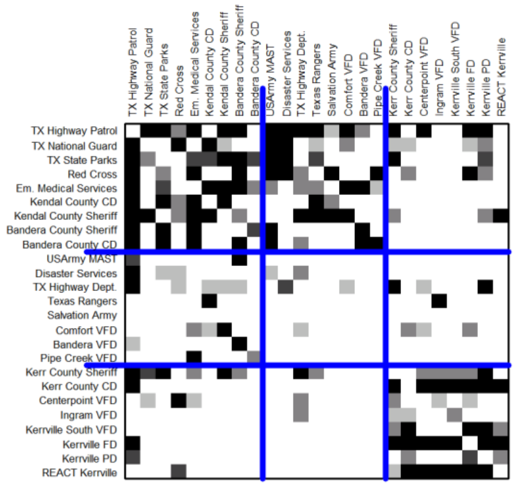
* all values in cells were multiplied by 10

Example: Communication between organizations after flash flooding for three counties: Bandera, Kendall and Kerr.

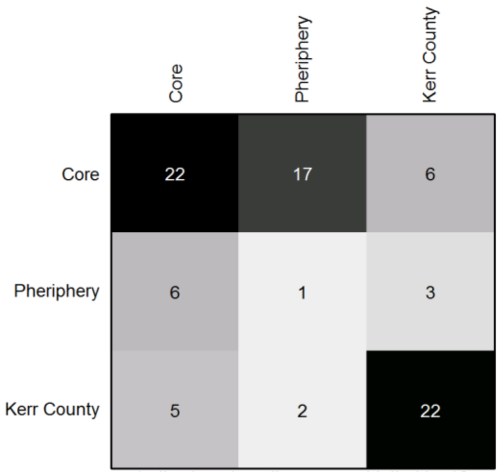
Unpartitioned matrix



Partitioned matrix

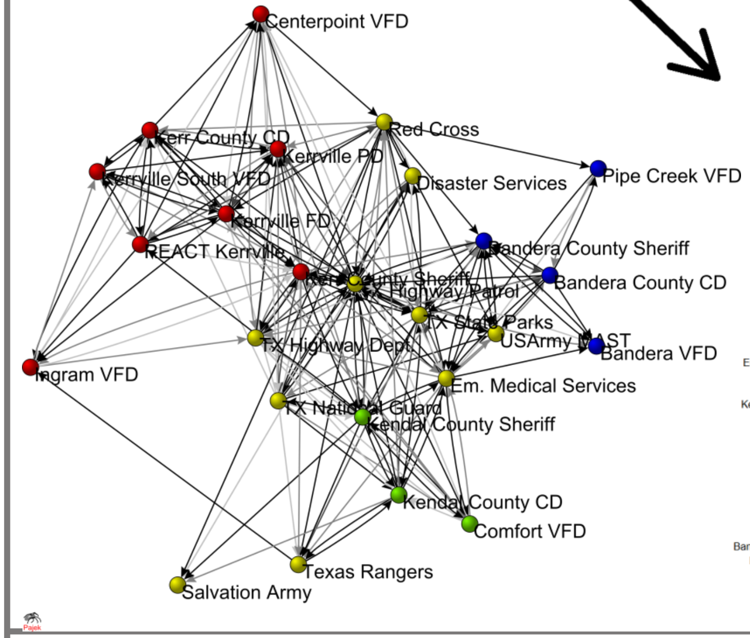


Density (image) matrix



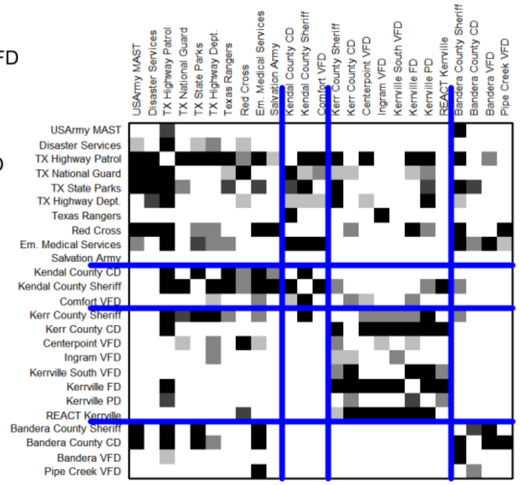
	Core	Pheriphery	Kerr County
Core	22	17	6
Pheriphery	6	1	3
Kerr County	5	2	22

* all values in cells were multiplied by 10

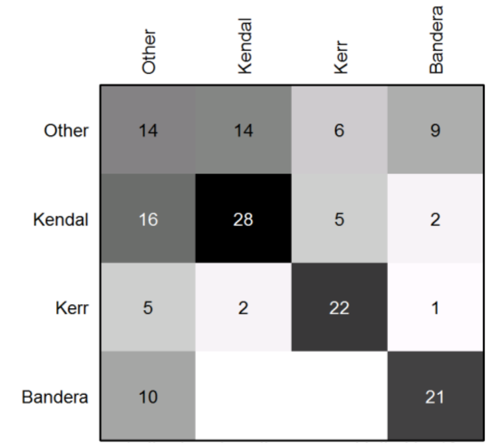


CRand = 0.418
Rand = 0.760
p = 1.21e-05

Matrix by counties



Density (image) matrix counties



	Other	Kendal	Kerr	Bandera
Other	14	14	6	9
Kendal	16	28	5	2
Kerr	5	2	22	1
Bandera	10			21

* all values in cells were multiplied by 10

Introduction

Blockmodeling is used for:

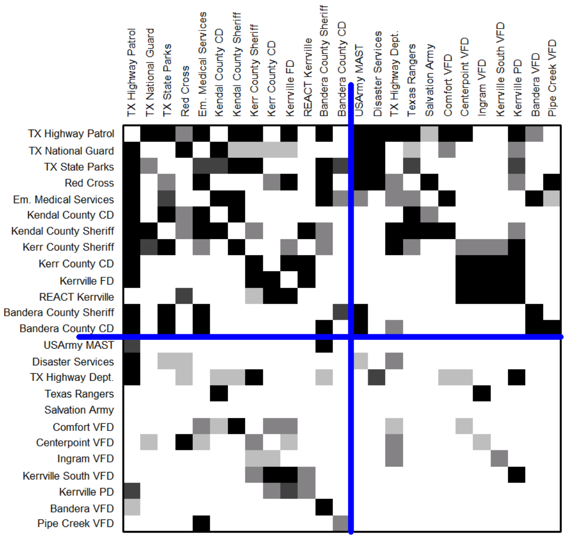
1. Finding clusters of units that have similar (in some sense) ties to other units (previous example)
2. Finding clusters of units according to some pre-specified structure of the network (e.g., core-periphery model, cohesive groups, hierarchy, …)
3. Evaluating the fit of a given partition to a network (and pre-specified structure)

Example: Texas flooding (Bandera, Kendall and Kerr)

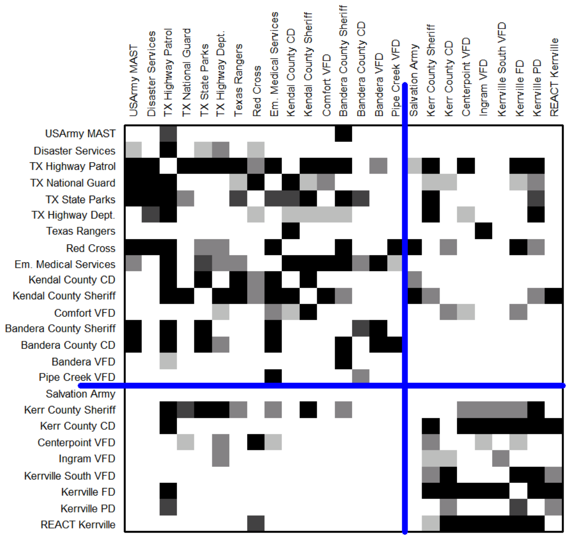
Pre-specified blockmodeling

Generalized blockmodeling in R

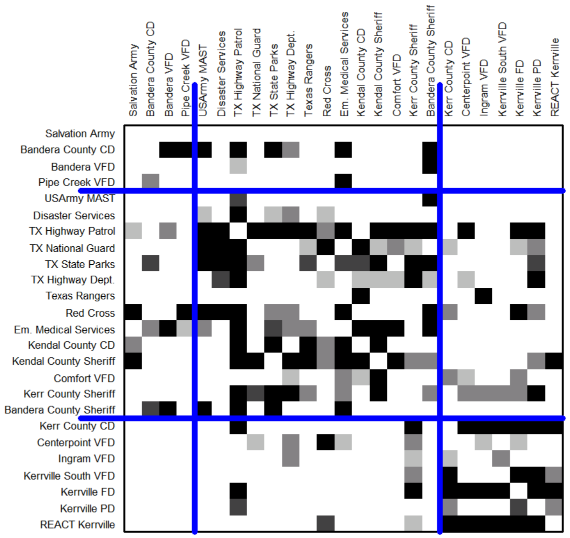
Core-pheriphery model



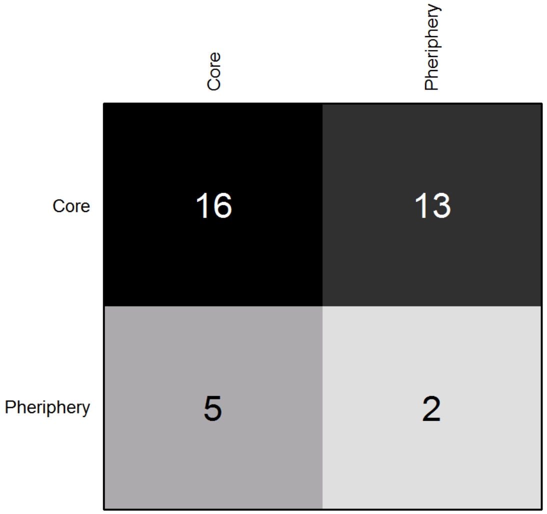
2 cohesive groups



3 cohesive groups



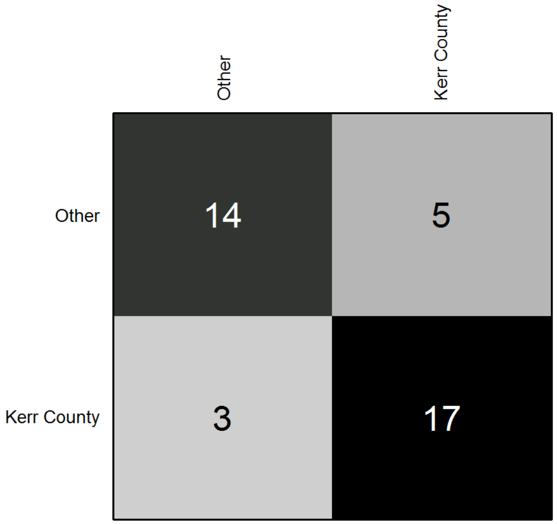
Density (image) matrix



	Core	Pheriphery
Core	16	13
Pheriphery	5	2

* all values in cells were multiplied by 10

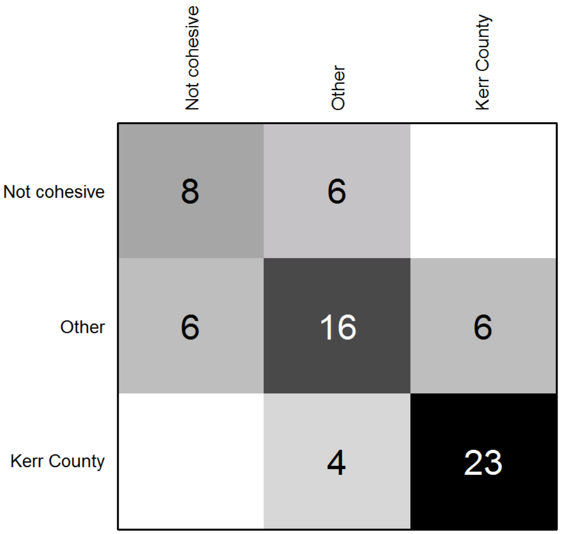
Density (image) matrix



	Other	Kerr County
Other	14	5
Kerr County	3	17

* all values in cells were multiplied by 10

Density (image) matrix



	Not cohesive	Other	Kerr County
Not cohesive	8	6	
Other	6	16	6
Kerr County		4	23

* all values in cells were multiplied by 10

Equivalences

Define when two units are equivalent → should be clustered together (put in the same cluster). Most well-known equivalences:

- ○ Structural equivalence (Lorrain and White, 1971)
  - “Structurally equivalent [units] are related in the same way to each other and all other [units]” (White and Reitz, 1983: 200). → They are indistinguishable if their labels are removed.
  - The “compatible” blocks are either complete (all 1s in binary networks) or completely empty (null) (all 0s)

Equivalences

- Regular equivalence (White and Reitz, 1983)
  - “Regularly equivalent [units] are connected in the same way to matching equivalents” (White and Reitz, 1983: 200)
  - The “compatible” blocks must have at lest one tie in each row and column or are completely empty (null) (all 0s)
- Stochastic equivalence (Holland et al., 1983): Two units are stochastically equivalent if they have the same probabilities of ties to all other units.
- Generalized equivalence (Doreian, Batagelj, and Ferligoj, 1994)

Approaches to blockmodeling

- Deterministic blockmodeling
  - “Conventional” blockmodeling
  - Generalized blockmodeling
  - Special purpose network partitioning

- Stochastic blockmodeling

  Based on some probabilistic model

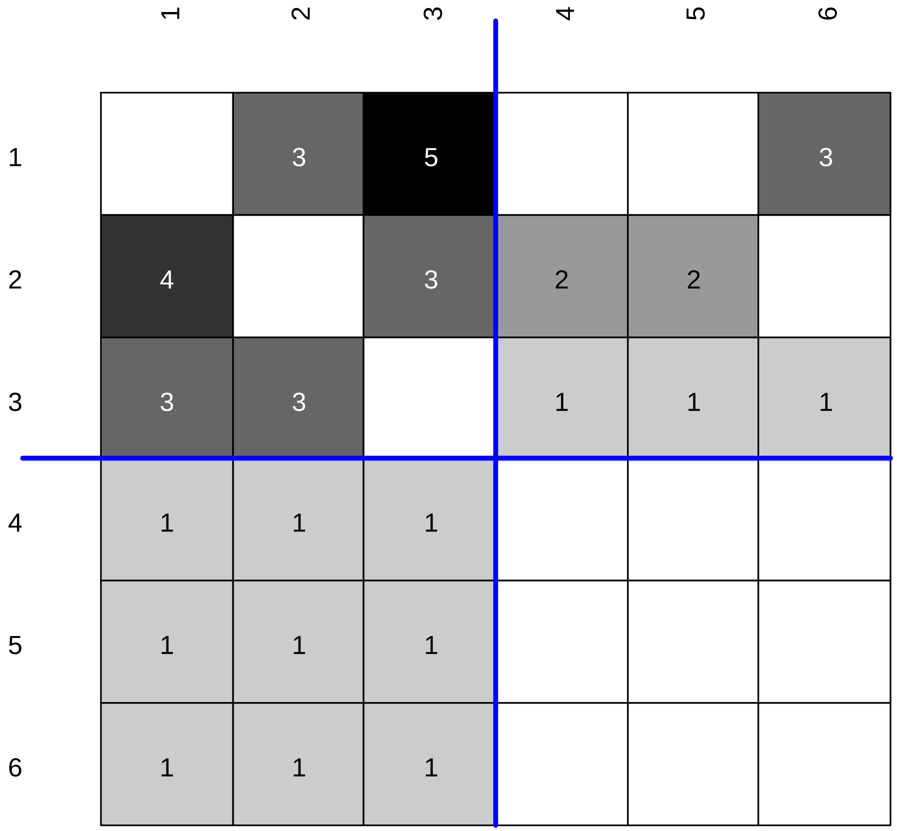
Generalized blockmodeling

- A direct approach: An explicit criteria function is optimized
- Based on generalized equivalence (Doreian, Batagelj and Ferligoj, 1994) → Equivalence is defined by set of allowed block types and possibly their position
- Optimization is usually done through local search (some attempts also use tabu search)

Generalized blockmodeling

Basic procedure for computation of criterion function:

- The matrix (of a network) is split into blocks
- An error is computed for all allowed block types in each block
- For each block, the block type with the minimum error is chosen
- The total error of matrix (network) is the sum of all block errors.



	1	2	3	4	5	6
1		3	5			3
2	4		3	2	2	
3	3	3		1	1	1
4	1	1	1			
5	1	1	1			
6	1	1	1			

Types of Generalized Blockmodeling

- Blockmodeling type is defined by the set of allowed block types
- All block types within one blockmodeling type must be compatible with each other (must be based on the same principles)

Types of Generalized Blockmodeling

These types are:

- Binary Blockmodeling
- Valued Blockmodeling
- Implicit Blockmodeling
- Homogeneity Blockmodeling
  - Sum of Squares (Homogeneity) Blockmodeling
  - Absolute Deviations (Homogeneity) Blockmodeling

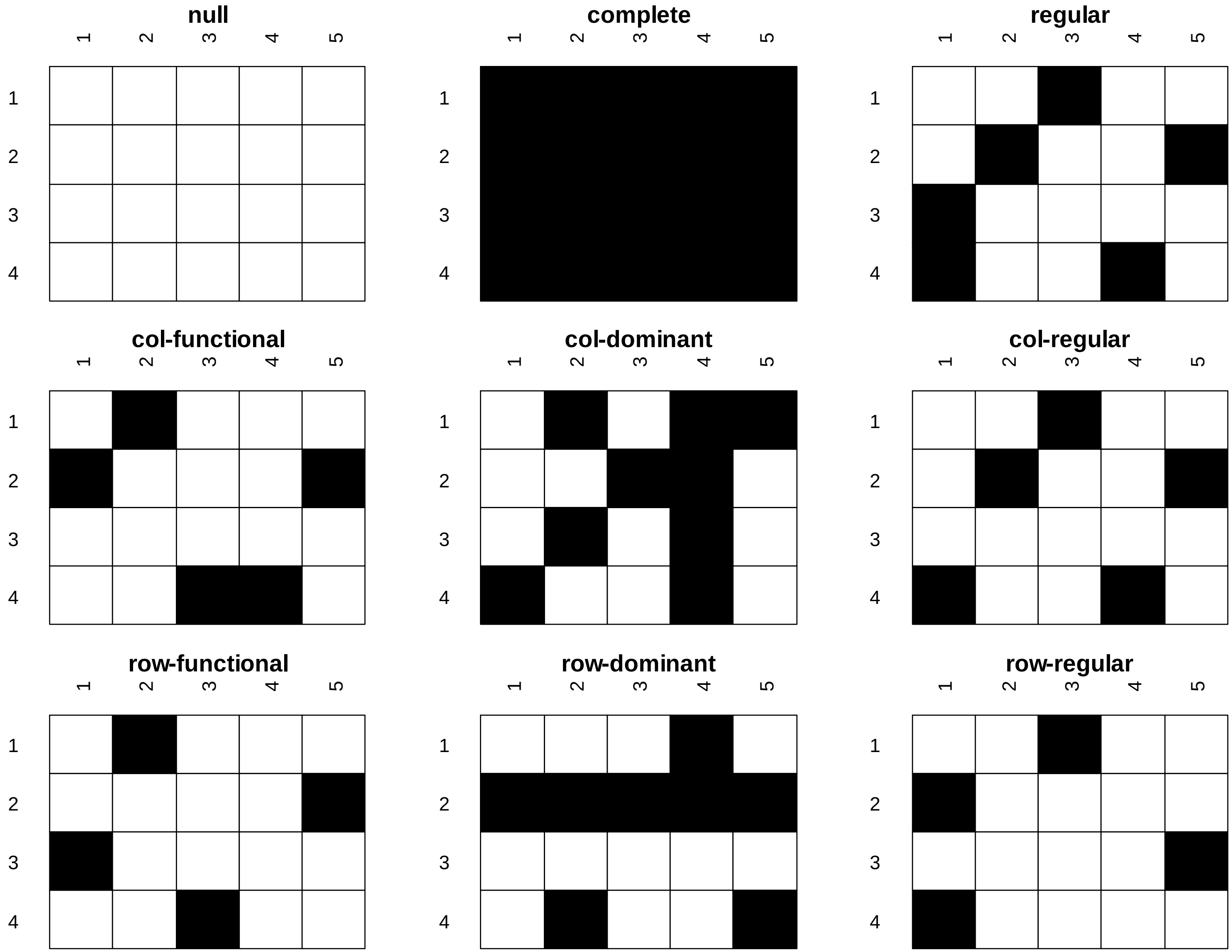
Valued networks

Binary (Generalized) Blockmodeling

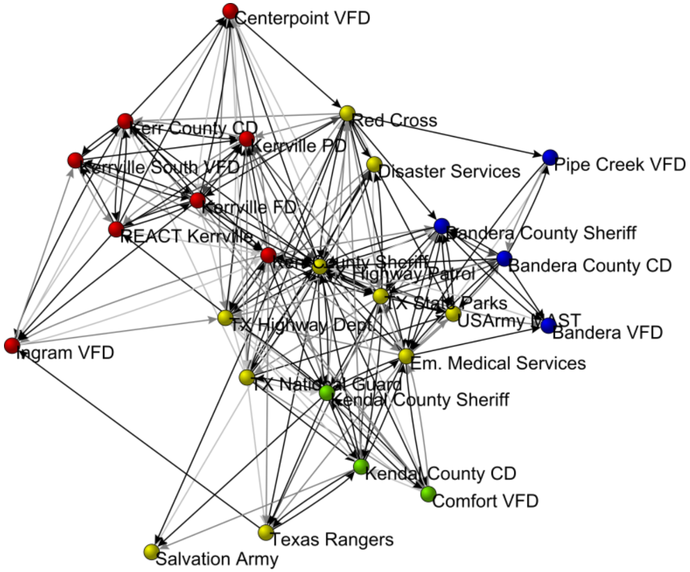
- Values of ties are not taken into account, only the presence/absence of ties is considered.
- Values can be only considered by determining the threshold for binarizing the network.
- The inconsistency (error) is computed as the number of ties that do not match* the ones in ideal blocks.

* How these ties are determined depends on the block type.

Ideal blocks
binary blockmodeling



Example: SAR network: Structural equivalence via binary blockmodeling



Partitioned matrix

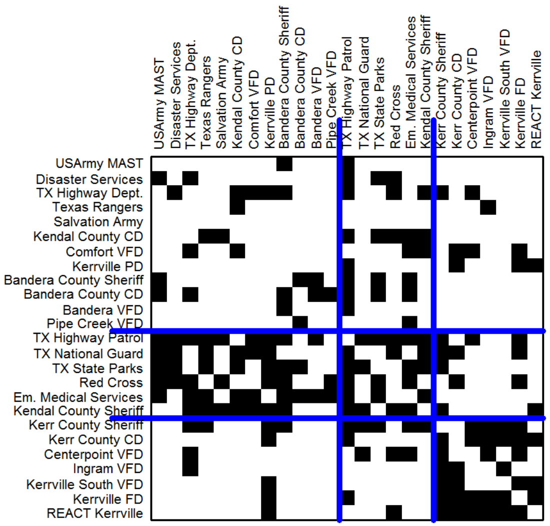


Image graph

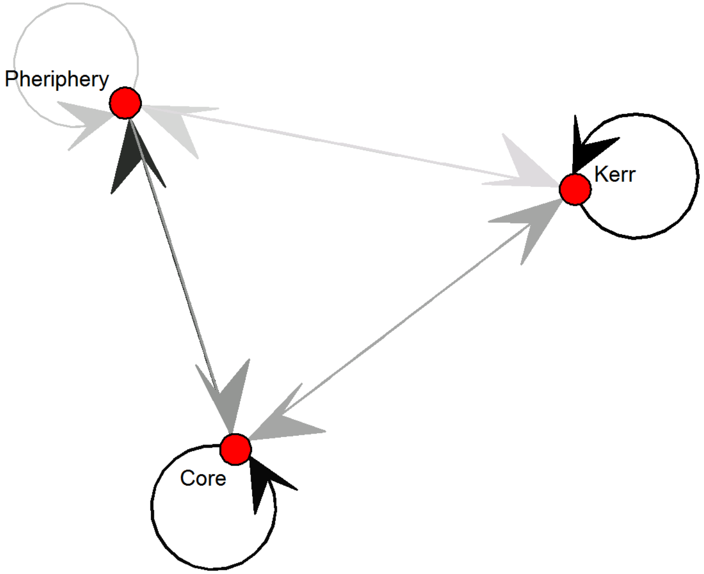
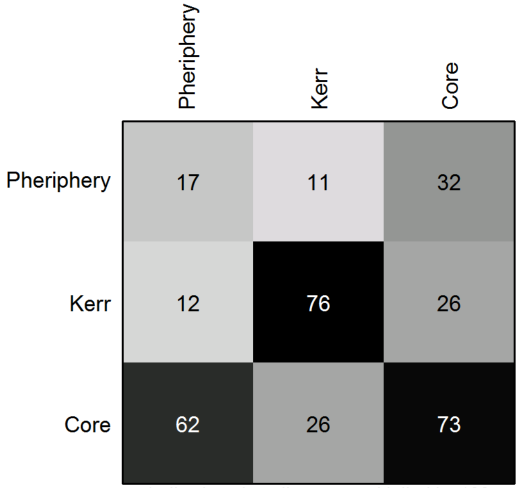


Image matrix



	Pheriphery	Kerr	Core
Pheriphery	17	11	32
Kerr	12	76	26
Core	62	26	73

* all values in cells were multiplied by 100

Valued Blockmodeling

- Values of ties are taken into account; however, we have to specify the value *m* over which ties are considered “relevant”.
- The inconsistencies are then computed as a deviation from either from 0 (tie should not be present) or from *m* (tie should be present)

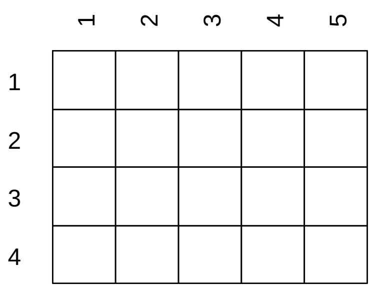
  Note: Deviation from *m* is computed only if tie value is lower than *m*.

Valued Blockmodeling

- The main problem is to determine the most appropriate value of *m* – distributions of either cell values or values of function *f* over rows or columns might be useful.
- Binary blockmodeling can be seen as a special case of valued blockmodeling.

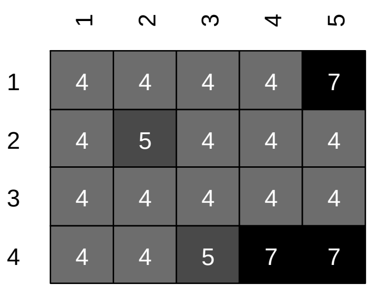
Ideal blocks valued blockmodeling (*m* = 4)

null



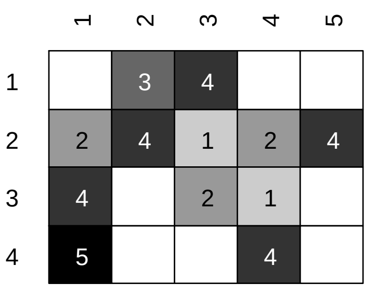
	1	2	3	4	5
1					
2					
3					
4					

complete



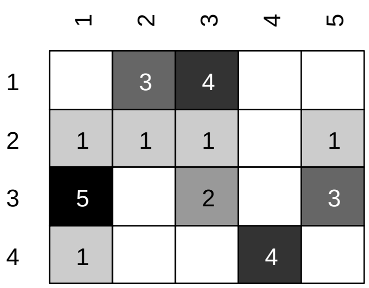
	1	2	3	4	5
1	4	4	4	4	7
2	4	5	4	4	4
3	4	4	4	4	4
4	4	4	5	7	7

***max*-regular**



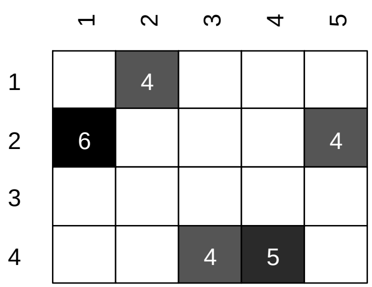
	1	2	3	4	5
1		3	4		
2	2	4	1	2	4
3	4		2	1	
4	5			4	

***sum*-regular**



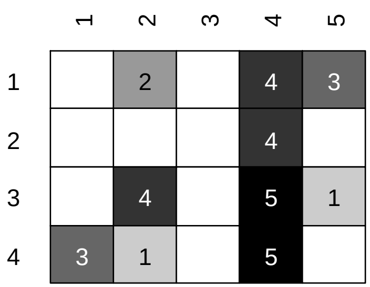
	1	2	3	4	5
1		3	4		
2	1	1	1		1
3	5		2		3
4	1			4	

col-functional



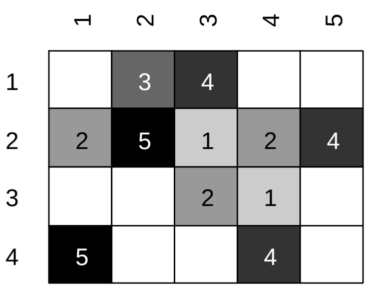
	1	2	3	4	5
1		4			
2	6				4
3					
4			4	5	

col-dominant



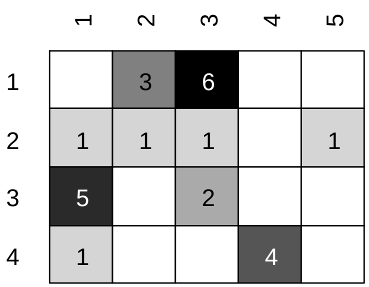
	1	2	3	4	5
1		2		4	3
2				4	
3		4		5	1
4	3	1		5	

col-*max*-regular



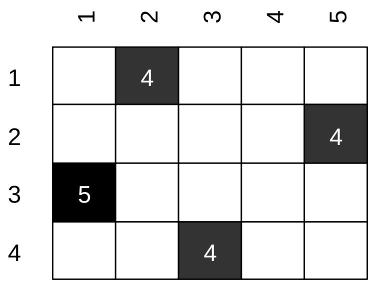
	1	2	3	4	5
1		3	4		
2	2	5	1	2	4
3			2	1	
4	5			4	

col-*sum*-regular



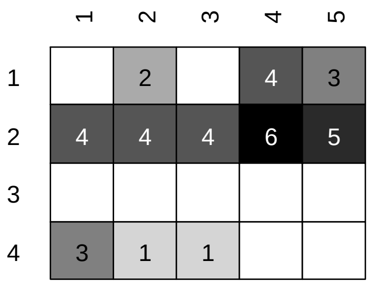
	1	2	3	4	5
1		3	6		
2	1	1	1		1
3	5		2		
4	1			4	

row-functional



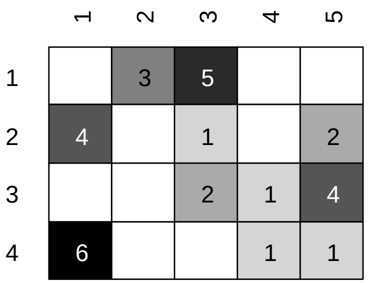
	1	2	3	4	5
1		4			
2					4
3	5				
4			4		

row-dominant



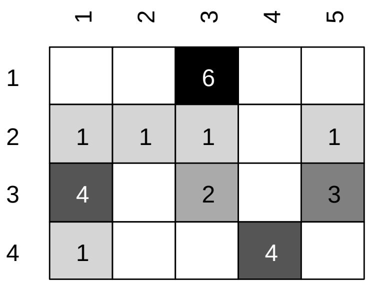
	1	2	3	4	5
1		2		4	3
2	4	4	4	6	5
3					
4	3	1	1		

row-*max*-regular



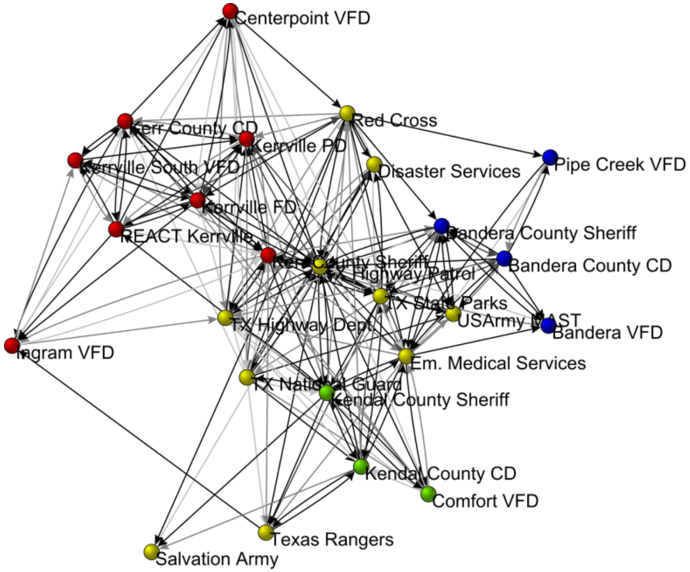
	1	2	3	4	5
1		3	5		
2	4		1		2
3			2	1	4
4	6			1	1

row-*sum*-regular



	1	2	3	4	5
1			6		
2	1	1	1		1
3	4		2		3
4	1			4	

Example: SAR network - Structural equivalence via valued blockmodeling



Partitioned matrix

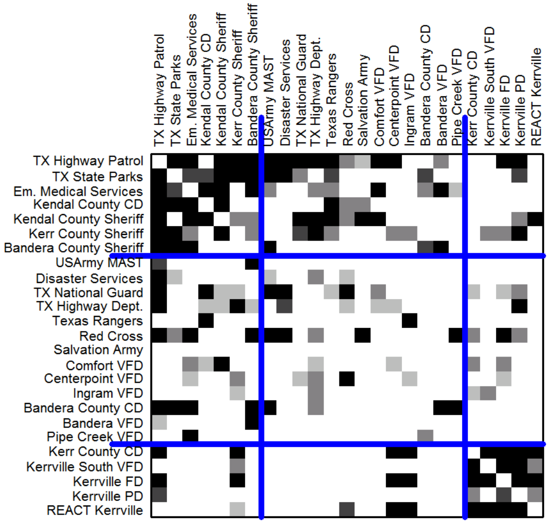


Image graph

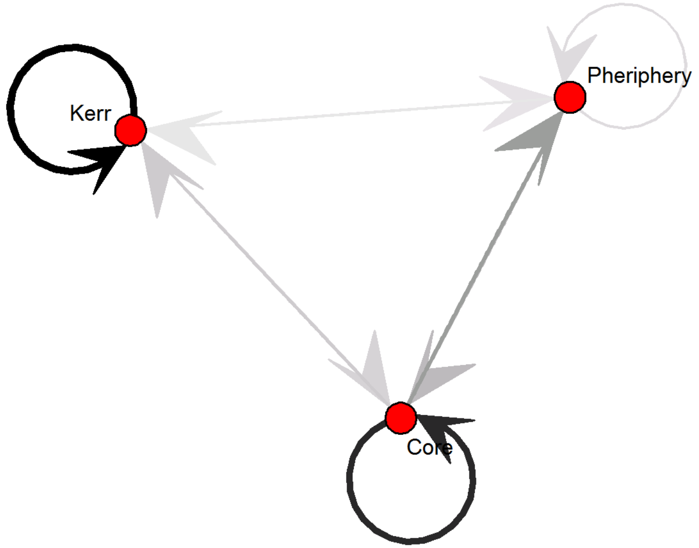
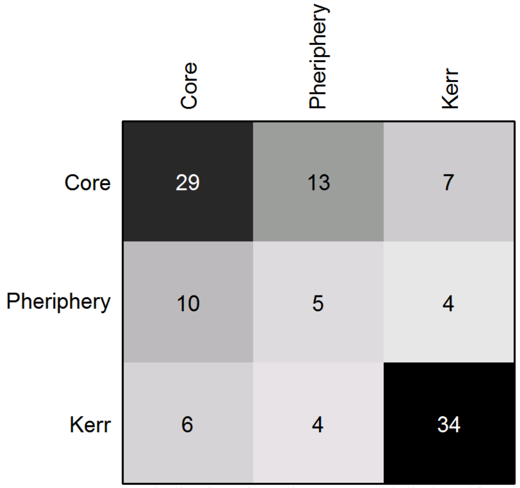


Image matrix



	Core	Pheriphery	Kerr
Core	29	13	7
Pheriphery	10	5	4
Kerr	6	4	34

* all values in cells were multiplied by 10

Homogeneity blockmodeling

The idea is to minimize a measure of variability within blocks for a relevant quantity, i.e.:

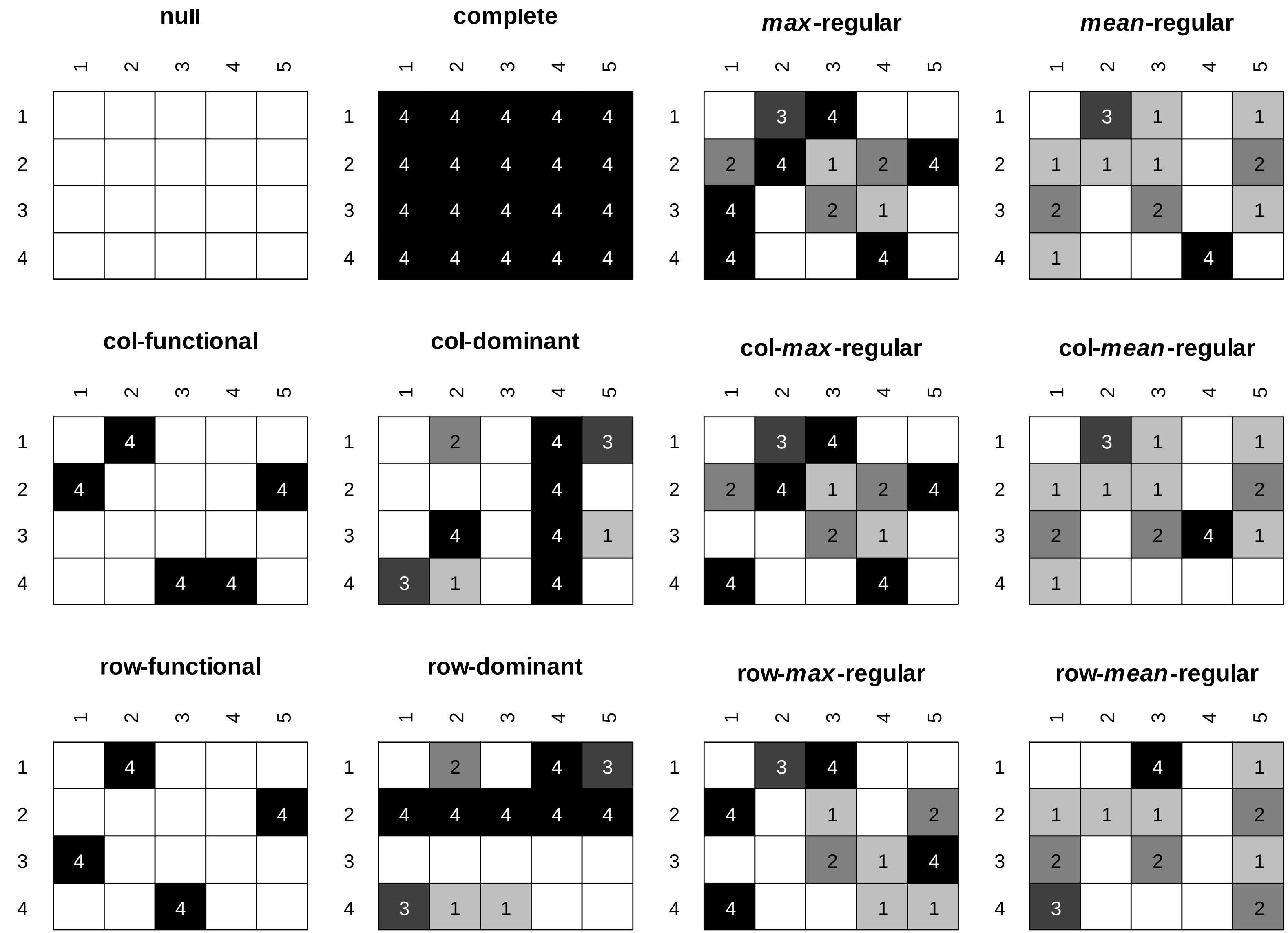
- In the case of structural equivalence: cell values (similar idea was also presented by Borgatti and Everett, 1992)
- In the case of *f*- regular equivalence: values of a function *f* over rows or columns
- In the case of other block types: we can select values over which a measure of variability should be computed (experimental)

Homogeneity Blockmodeling

Two measures of variability are suggested:

- Sum of Squared deviations from the mean (Sum of Squares)
- Sum of Absolute Deviations from the median (Absolute Deviations)

Ideal blocks
implicit and homogeneity blockmodeling



null

	1	2	3	4	5
1					
2					
3					
4					

complete

	1	2	3	4	5
1	4	4	4	4	4
2	4	4	4	4	4
3	4	4	4	4	4
4	4	4	4	4	4

***max*-regular**

	1	2	3	4	5
1		3	4		
2	2	4	1	2	4
3	4		2	1	
4	4			4	

***mean*-regular**

	1	2	3	4	5
1		3	1		1
2	1	1	1		2
3	2		2		1
4	1			4	

col-functional

	1	2	3	4	5
1		4			
2	4				4
3					
4			4	4	

col-dominant

	1	2	3	4	5
1		2		4	3
2				4	
3		4		4	1
4	3	1		4	

col-*max*-regular

	1	2	3	4	5
1		3	4		
2	2	4	1	2	4
3			2	1	
4	4			4	

col-*mean*-regular

	1	2	3	4	5
1		3	1		1
2	1	1	1		2
3	2		2	4	1
4	1				

row-functional

	1	2	3	4	5
1		4			
2					4
3	4				
4			4		

row-dominant

	1	2	3	4	5
1		2		4	3
2	4	4	4	4	4
3					
4	3	1	1		

row-*max*-regular

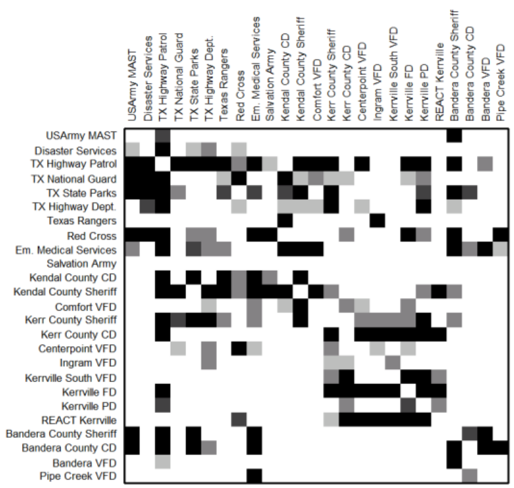
	1	2	3	4	5
1		3	4		
2	4		1		2
3			2	1	4
4	4			1	1

row-*mean*-regular

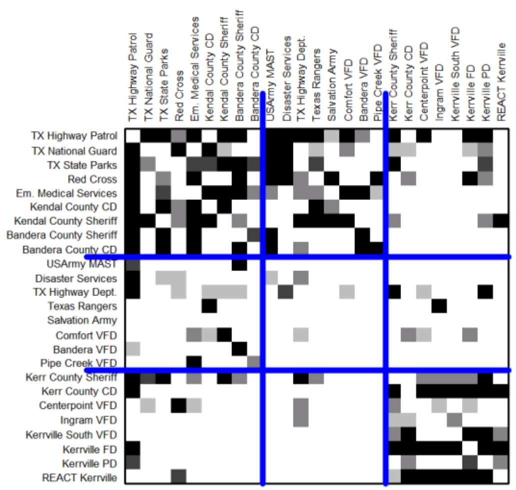
	1	2	3	4	5
1			4		1
2	1	1	1		2
3	2		2		1
4	3				2

Example: Communication between organizations after flash flooding for three counties: Bandera, Kendall and Kerr.

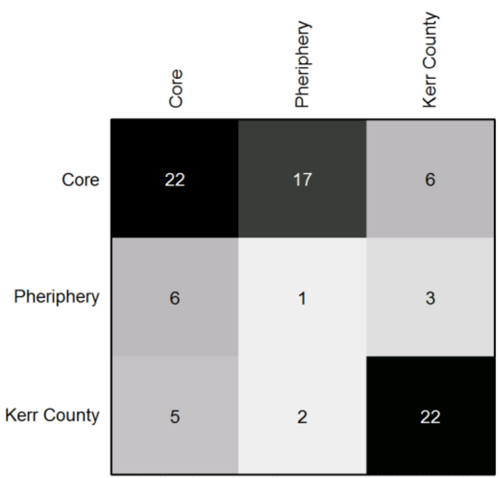
Unpartitioned matrix



Partitioned matrix

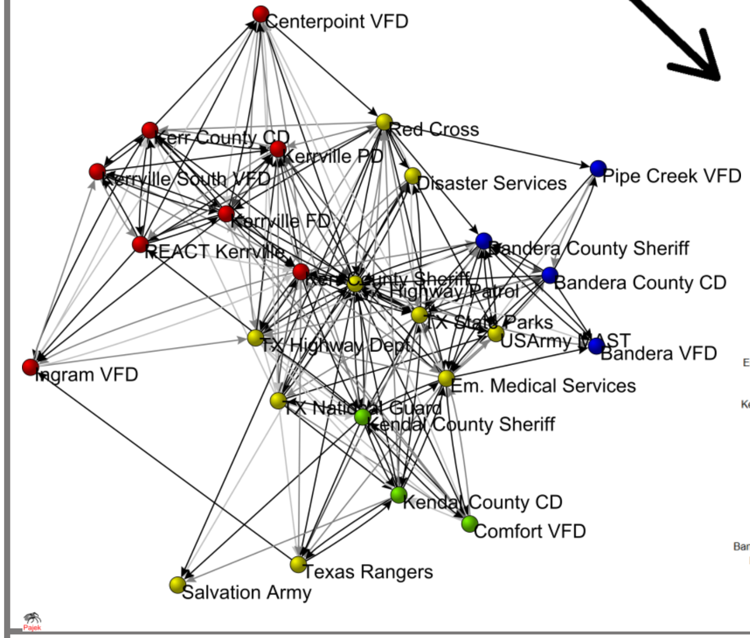


Density (image) matrix



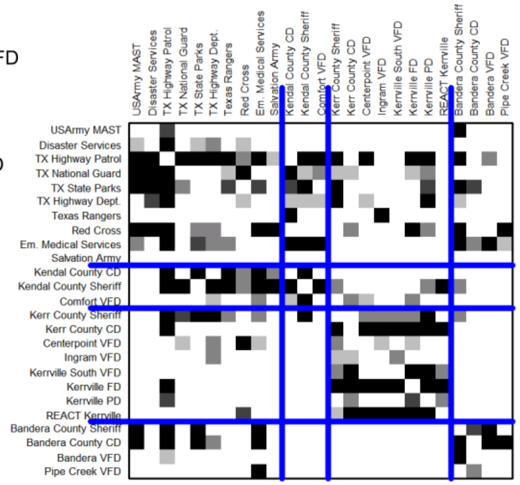
	Core	Pheriphery	Kerr County
Core	22	17	6
Pheriphery	6	1	3
Kerr County	5	2	22

* all values in cells were multiplied by 10

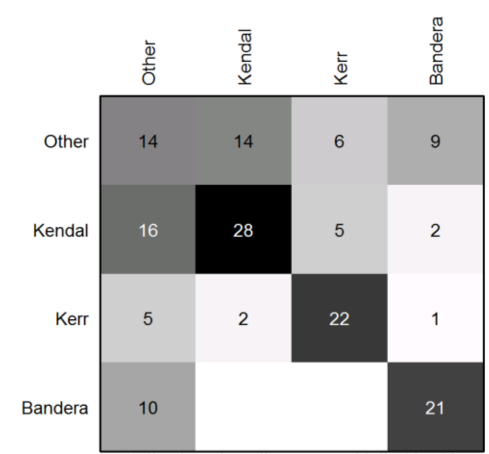


CRand = 0.418
Rand = 0.760
p = 1.21e-05

Matrix by counties



Density (image) matrix counties



	Other	Kendal	Kerr	Bandera
Other	14	14	6	9
Kendal	16	28	5	2
Kerr	5	2	22	1
Bandera	10			21

* all values in cells were multiplied by 10

Types of (generalized) blockmodeling

Blockmodleing type	**Inconsistency**
Binary	Equal (or not) to 0 (absent tie) or 1 (present tie)
Valued	Deviation from 0 (nonexistent or irrelevant tie) or parameter *m* (relevant tie), if the tie is lower than *m*
Implicit	Deviation from 0 (nonexistent or irrelevant tie) or block maximum * (relevant tie)
Homogeneity	Variability

Characterizations of ideal blocks

Ideal block with "label"	Description for binary blockmodeling	Description for valued blockmodeling	Description for homogeneity and implicit blockmodeling
null "nul"	all 0 *	all 0 *	all 0 *
complete "com"	all 1 *	all at least m *	all equal *
(*f*-) regular "reg"	At least 1 tie in each row and each column	the *f* over each row and each column at least m	*f* over all rows and all columns separately equal

* - exception - different rules may apply for the diagonal (of the diagonal blocks)

Evaluation of generalized blockmodeling types

- Binary and valued blockmodeling
  - We have a threshold that tells us when a tie is relevant.
  - Blocks differ by block types, not by tie values.
- Homogeneity blockmodeling
  - Blocks mainly differ by tie values (not block types)
  - Does not distinguish between null and other block types (The null block problem)
- Implicit blockmodeling
  - Very nice theoretical properties – problems in practice
  - Block types are determined relatively to block maximum
  - Blockmodels can be illogical if null blocks are used
  - Without null blocks, better partitions, but is it meaningful

Pre-specified blockmodeling

- Pre-specifying block-types by position (as opposed to for the whole network (all positions in the image matrix)
- We specify allowed block types for each position
- E. g.: Core periphery model (one of possibilities):

	Core	**Periphery**
Core	Complete	Null, Complete, Regular
Periphery	Null, Complete, Regular	Null

Pre-specified blockmodeling

- E. g.: Cohesive groups (2 groups):

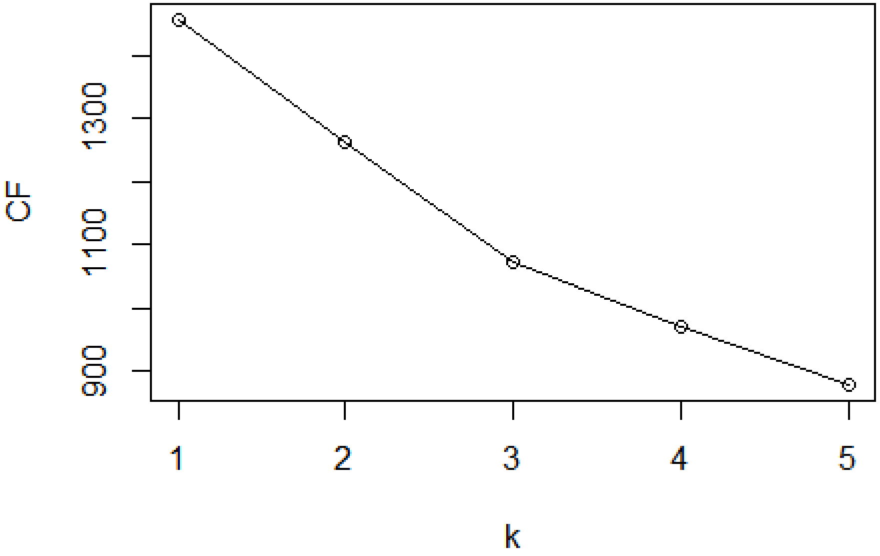
	1	2
1	Complete	Null
2	Null	Complete

- E. g.: Cohesive groups (3 groups):

	1	2	3
1	Complete	Null	Null
2	Null	Complete	Null
3	Null	Null	Complete

Selecting the number of clusters

- Still an open problem
- Simplest possibility – compare criterion function (CF) values across different number of clusters. As at least for "free" models, the CF can only decrees with the number of clusters, usually a "break" in scree plot is sought.
- A relative fit index can be used (Cugmas et al., 2021).



`blockmodeling` R package – Main function - `optRandomParC`

Optimizes selected number of random starting partitions.

Main parameters:

- **`M`** – The network in (dense) matrix format. An array for multirelational network
- **`k`** – the number of clusters. If data contain more sets, this is a vector (advanced).
- **`approaches`** – one of `"bin"` (binary), `"val"` (valued) or `"hom"` (homogeneity) blockmodeling.
- **`blocks`** – allowed block types (see next slide)
- **`rep`** – the number of random starting partitions
- **`nCores`** – For multicore computers. Defaults to 1. 0 means automatic (number of available cores – 1).

`blockmodeling` R package - `blocks` parameter

`blocks` parameter – specifies the equivalences as:

- A character vector of **allowed block types** or a list of such vectors for multirelational networks.
- **A pre-specified blockmodel**: An array with four dimensions. The third and the fourth represent the clusters (for rows and columns). The first is as long as the maximum number of allows block types for a given block. If some block has less possible block types, the empty slots should have values NA. The second dimension is the number of relations (can be omitted for single-relational networks or set to 1). The values in the array should be the ones from the next slide.

`blockmodeling` R package – block types

- `"nul"` - null or empty block
- `"com"` - complete block
- `"rdo"`, `"cdo"` - row and column-dominant blocks (binary and valued approach only)
- `"reg"` - (f-)regular block
- `"rre"`, `"cre"` - row and column-(f-)regular blocks
- `"rfn"`, `"cfn"` - row and column-dominant blocks (binary and valued approach only)
- `"den"` - density block (binary approach only)
- `"avg"` - average block (valued approach only)
- `"dnc"` - do not care block - the error is always zero

The ordering is important, since if several block types have identical error, the first on the list is selected.

`blockmodeling` R package – other functions

- `optParC` – Optimizes one partition
- `critFunC` – Evaluates a fit of one partition
- `plotMat` – Plots a possibly partitioned matrix
- `plot` – Plot methods exists for results or `optRandomParC` and `optParC`
- `err, clu, IM` – functions for extracting relevant parts of the results of `optRandomParC` and `optParC`
- `funByBlocks` – Computes a function (e.g. `mean`) by each block, possibly by ignoring the diagonal.
- `orderClu` - Orders the partition so that mean values of `fun` applied to columns (default), rows or both is decreasing by clusters.

`BlockmodelingGUI` R package

Fabio Ashtar Telarico has created a GUI - `BlockmodelingGUI` R package and standalone program.

The package supports

- The analysis of one-mode single-relational networks
- Generalized blockmodeling of binary and values networks
- Selection of allowed block types or pre-specification of the blockmodel
- Reading of network from several file formats
- Visualization of (partitioned) network as a graph or matrix.

Extensions to generalized blockmodeing and special cases

- The null block problem
- Blockmodeling sparse binary networks
- Blockmodeling valued networks where units have different relational capabilities
- Blockmodeling multi-relational networks and multi-criteria blockmodeling
- Blockmodeling multilevel/linked networks

The null block problem

- The problem occurs when the null block is a special case of most other block type
- This happens in homogeneity blockmodeling and some versions of implicit blockmodeling
- Consequences:
  - Null blocks can not be identified (distinguished from other block types)
  - “null” blocks are “less empty” than null blocks obtained (found) with other methods

The null block problem: solution

- The use of constrained non-null blocks (complete, *f*-regular, …)
- The constrain is that the value from which deviations (when measuring variability) is computed should be higher than some threshold.
- An appropriate value for the threshold in case of binary networks is twice the density of the network (valued approaches can be used on binary networks).

The null block problem: solution

Example: Computing inconsistency of a constrained complete block

$$\delta = \begin{cases} \sum_{i \in C_a} \sum_{j \in C_b} \left(b_{ij} - \bar{B}\right)^2 & \text{if } \bar{B} \geq p \\ \sum_{i \in C_a} \sum_{j \in C_b} \left(b_{ij} - p\right)^2 & \text{otherwise} \end{cases}$$

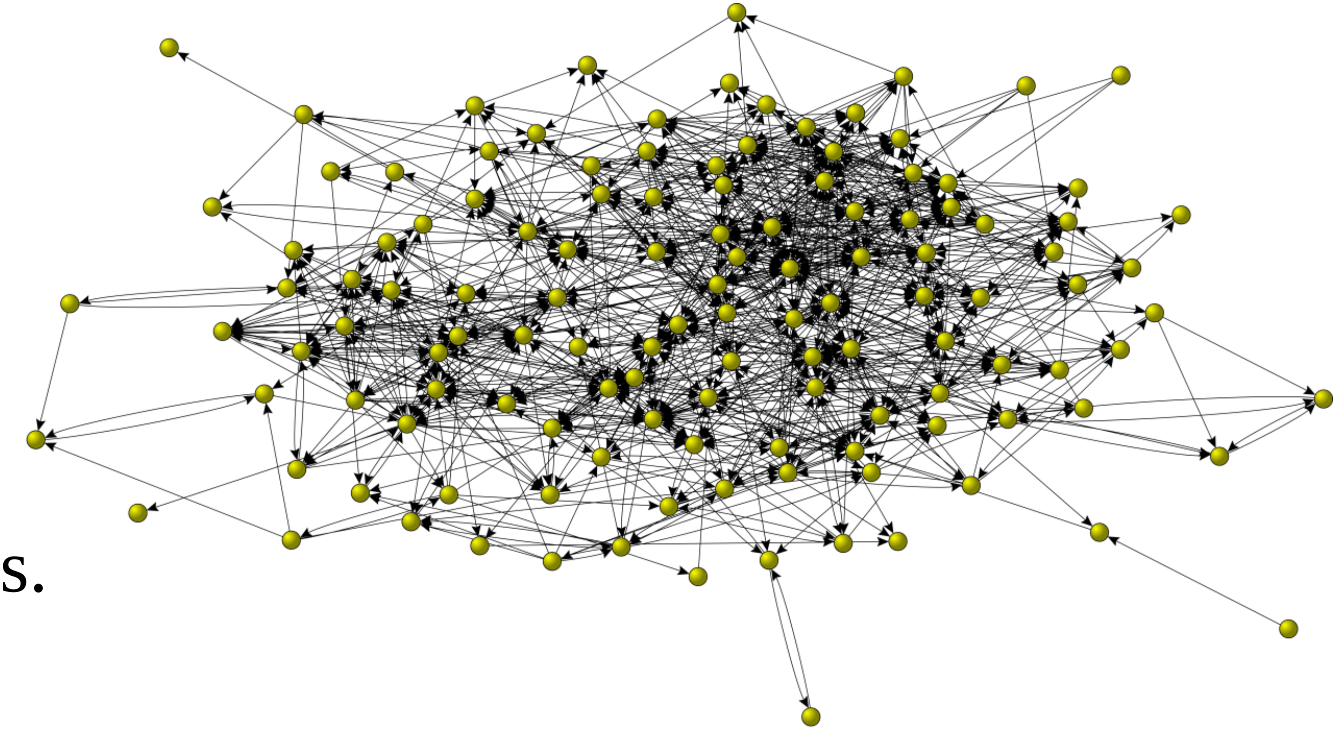
- Where:
- δ is the computed block inconsistency
- $\bar{B}$ is the mean of the block
- C_a is the row cluster
- C_b is the column cluster
- b_{ij} is the value of the tie from unit *i* to unit *j*
- p is the pre-specifed value

Blockmodeling sparse binary networks

- Problematic with binary blockmodeling:
  - Regular equivalence: almost all units are classified in the same equivalence class
  - Structural equivalence (binary version): finds only very small complete blocks.
- General suggestion – use either:
  - Binary blockmodeling according to structural equivalence with different weights for inconsistencies
  - Sum of squares (homogeneity) blockmodeling with null and constrained complete blocks → for similar densities.

Blockmodeling sparse binary networks - example

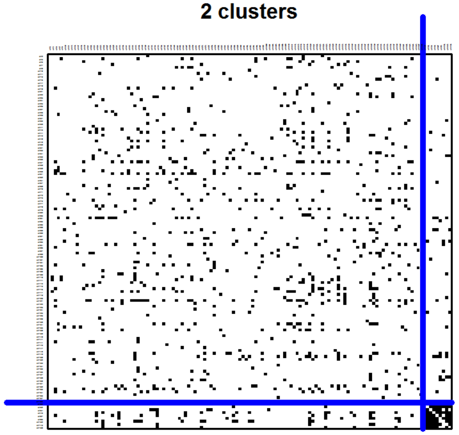
Data: Network of the elite of cancer researchers in France (Lazega, Jourda, Mounier, and Stofer, 2008). The aggregated network of researchers (several relations were merged) is used - measures which researchers each researcher specified as their collaborator/s.



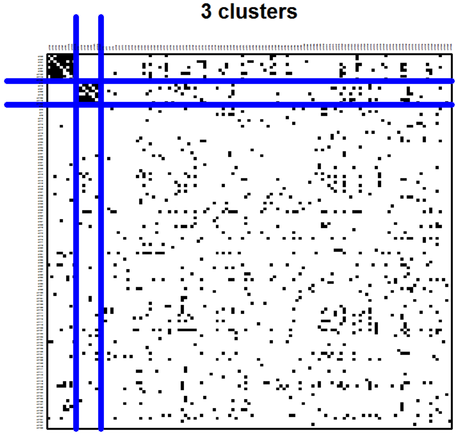
Blockmodeling sparse binary networks - example

- Binary blockmodeling according to structural equivalence → Only very small clusters

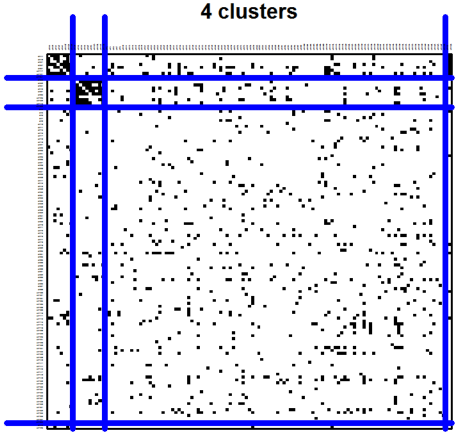
2 clusters



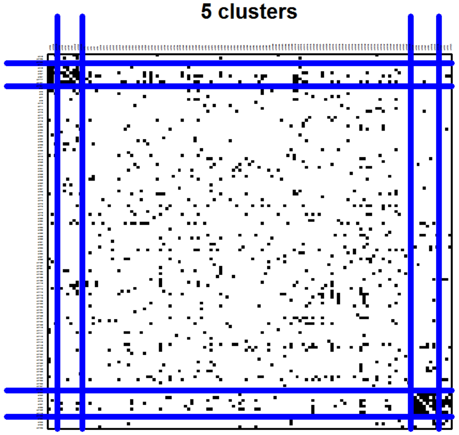
3 clusters



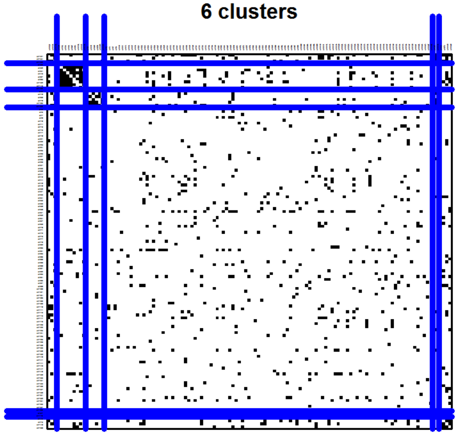
4 clusters



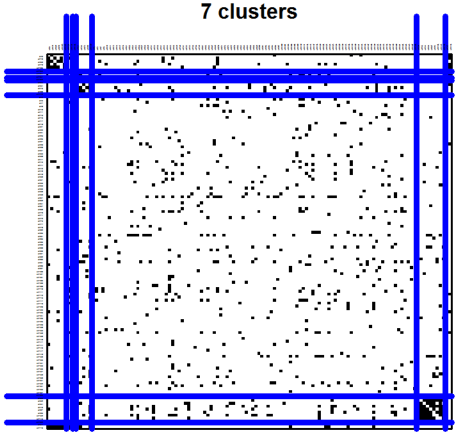
5 clusters



6 clusters



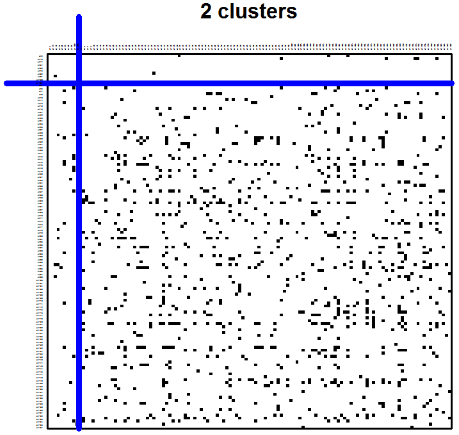
7 clusters

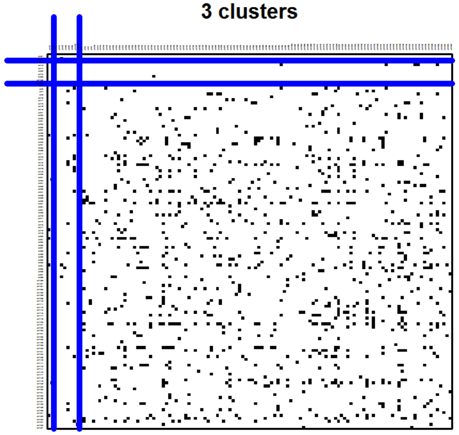


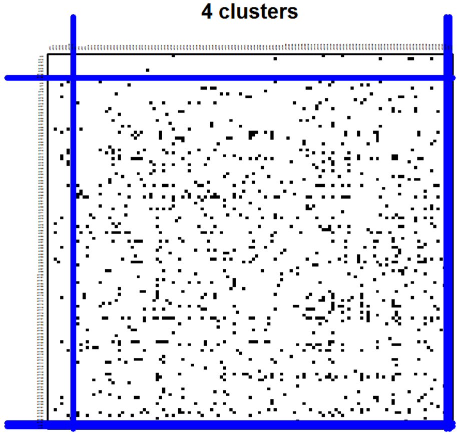
Blockmodeling sparse binary networks - example

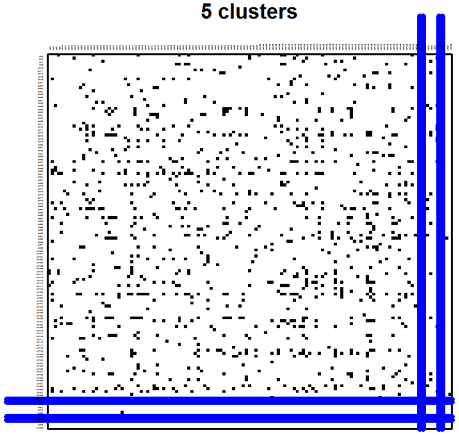
- Binary blockmodeling according to regular equivalence

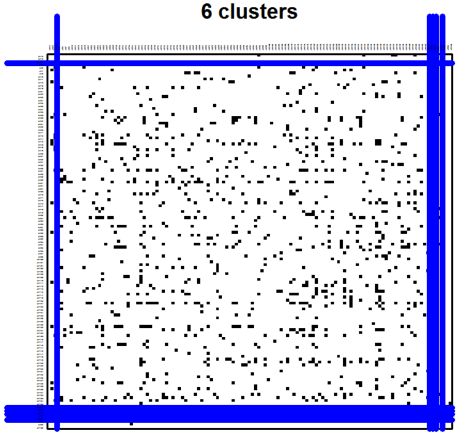
  → Always one giant cluster

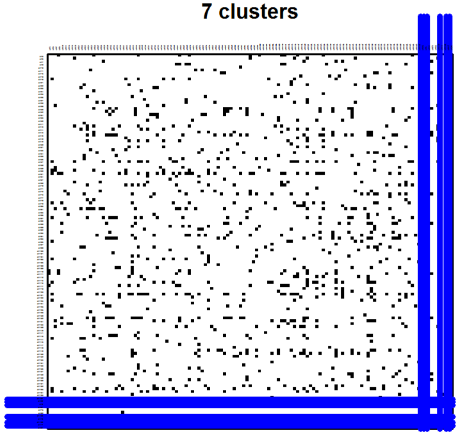

2 clusters


3 clusters


4 clusters


5 clusters


6 clusters

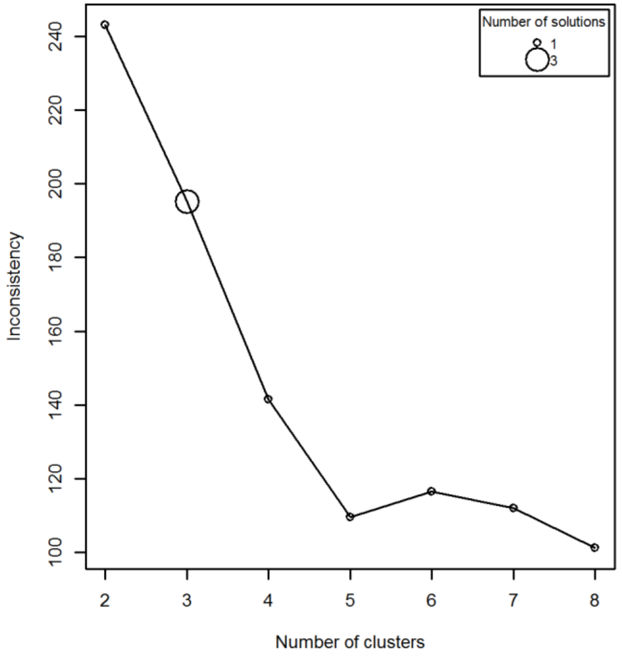

7 clusters

Generalized blockmodeling in R

Blockmodeling sparse binary networks - example

- Binary blockmodeling with null and density blocks – all non-null block have similar densities

Inconsistencies



Partitioned matrix

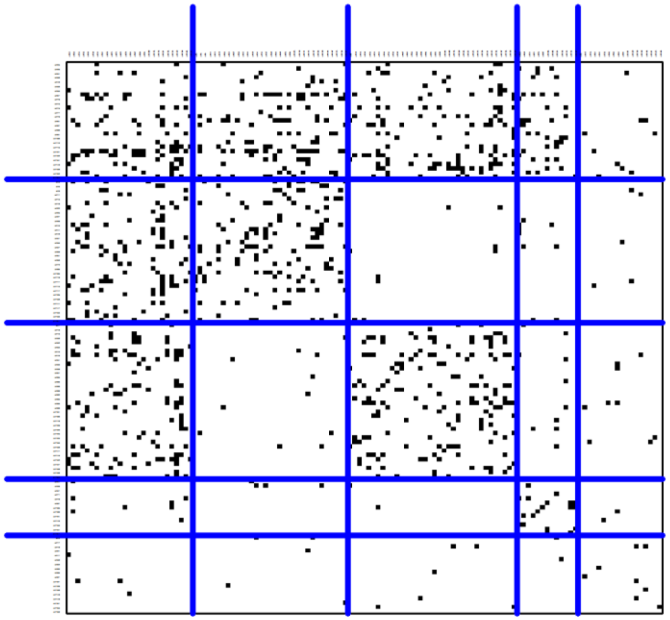
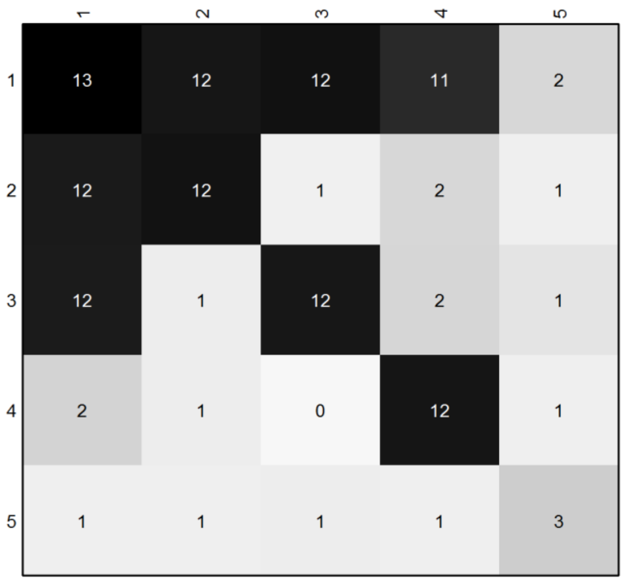


Image (density) matrix

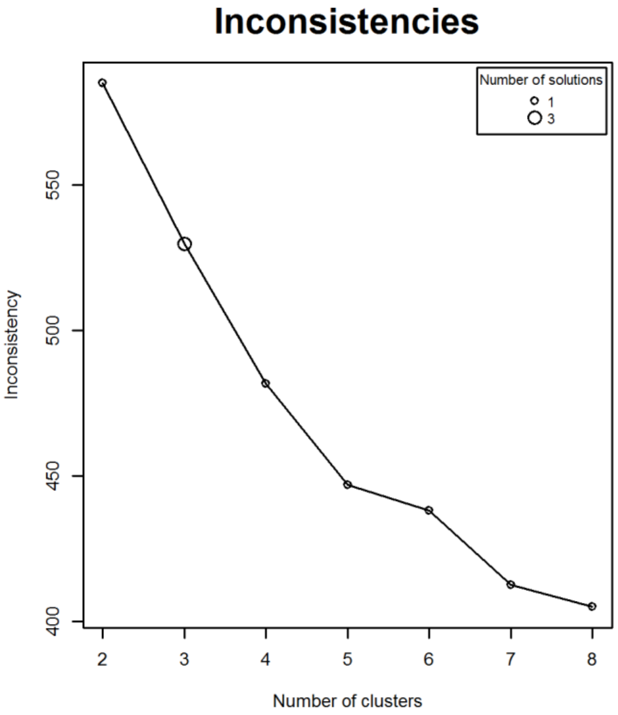


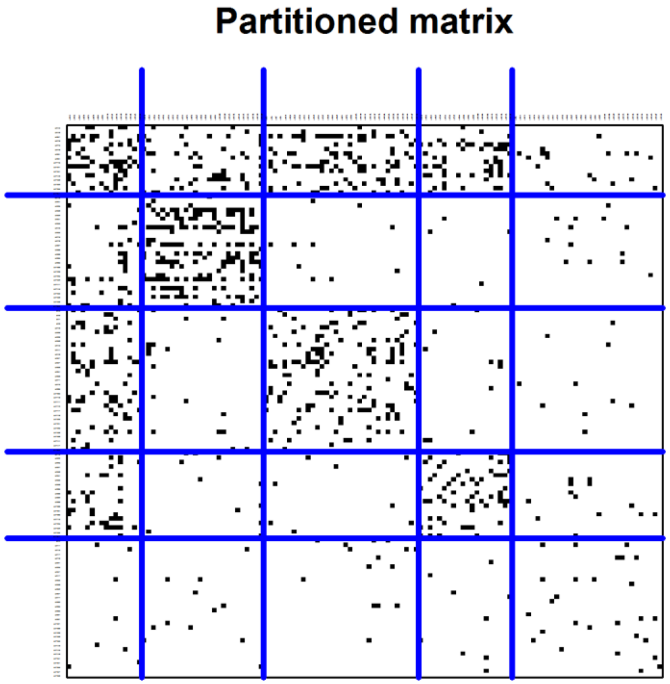
	1	2	3	4	5
1	13	12	12	11	2
2	12	12	1	2	1
3	12	1	12	2	1
4	2	1	0	12	1
5	1	1	1	1	3

* all values in cells were multiplied by 100

Blockmodeling sparse binary networks - example

- Binary blockmodeling according to structural equivalence with unequal weighting – nice partition


Inconsistencies


Partitioned matrix

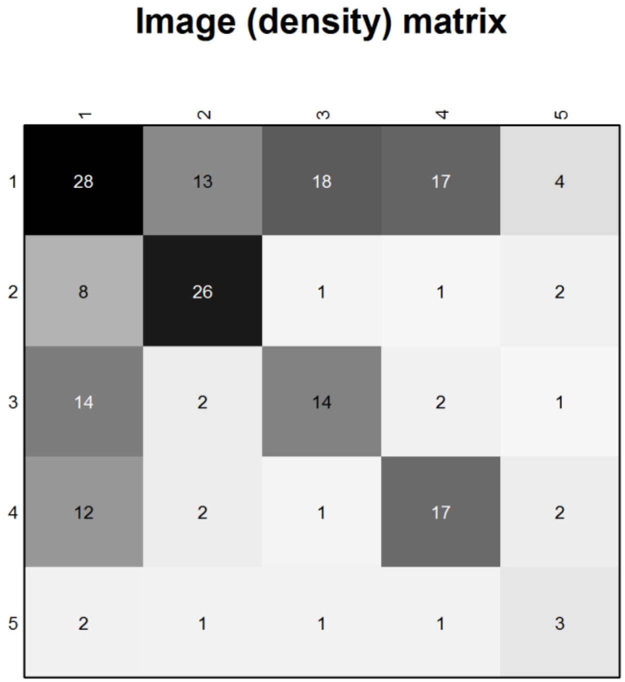


Image (density) matrix

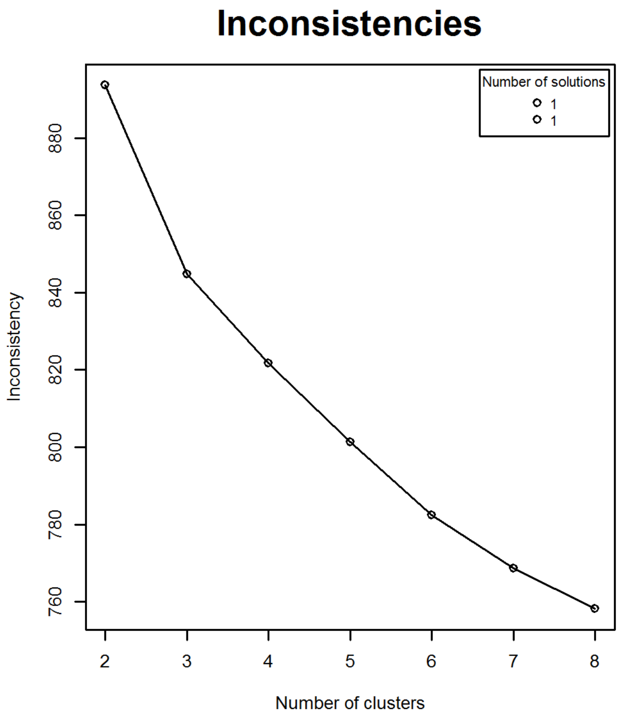
	1	2	3	4	5
1	28	13	18	17	4
2	8	26	1	1	2
3	14	2	14	2	1
4	12	2	1	17	2
5	2	1	1	1	3

* all values in cells were multiplied by 100

Blockmodeling sparse binary networks - example

- Sum of squares blockmodeling with null and constrained complete blocks → differential densities

Inconsistencies



Partitioned matrix

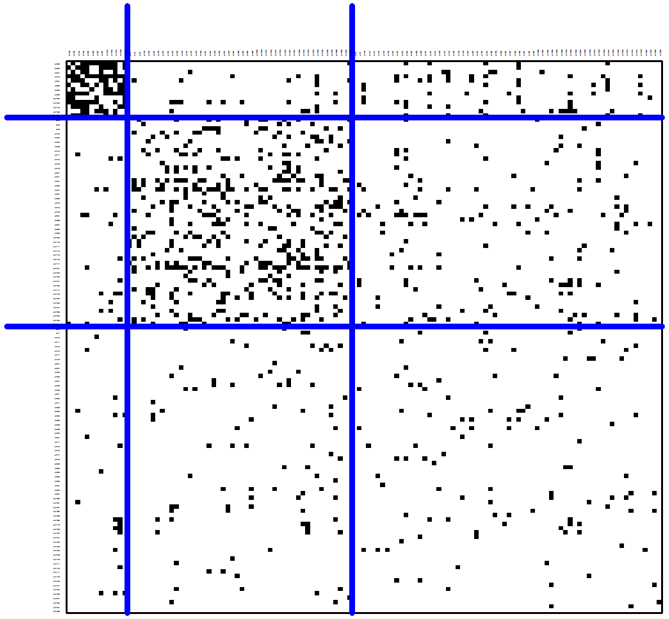
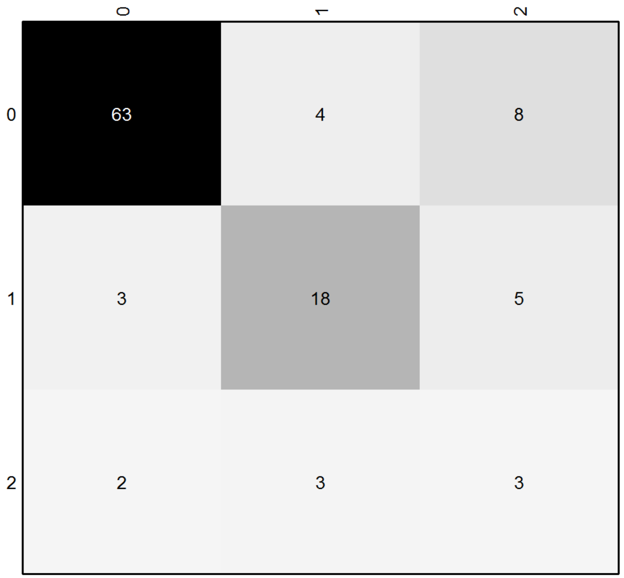


Image (density) matrix



	0	1	2
0	63	4	8
1	3	18	5
2	2	3	3

* all values in cells were multiplied by 100

Blockmodeling sparse binary networks example – pre-specified blockmodeling

Interesting partitions can be also found using pre-specified blockmodeling

- Core-pheriphery model
- Cohesive groups: 3 or 4 clusters

Blockmodeling sparse binary networks - example

Core-periphery model:

- Binary blockmodeling (with and without weighting):

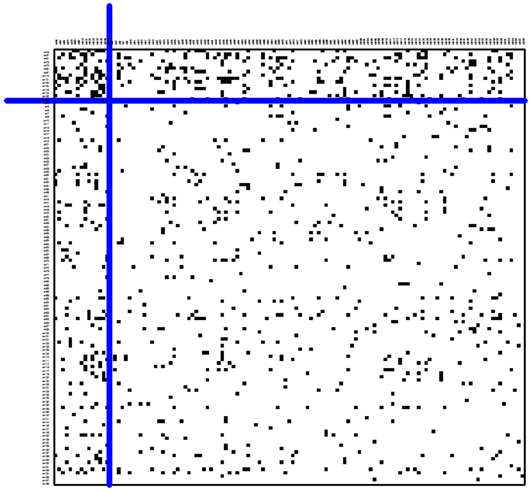
	1	2
1	com	dnc
2	dnc	nul

- Sum of squares blockmodeling with null, constrained and unconstrained complete blocks

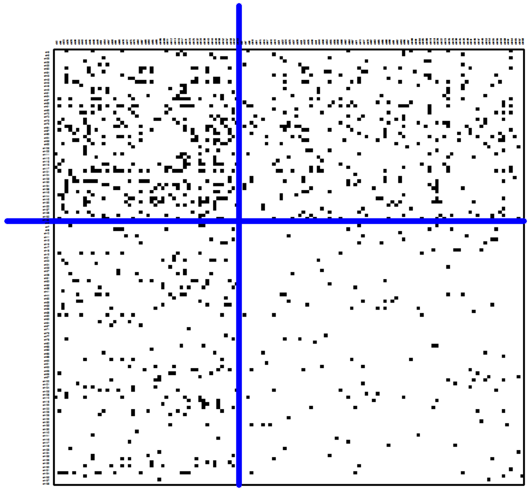
	1	2
1	com - constrained	com
2	com	null

Blockmodeling sparse binary networks - example

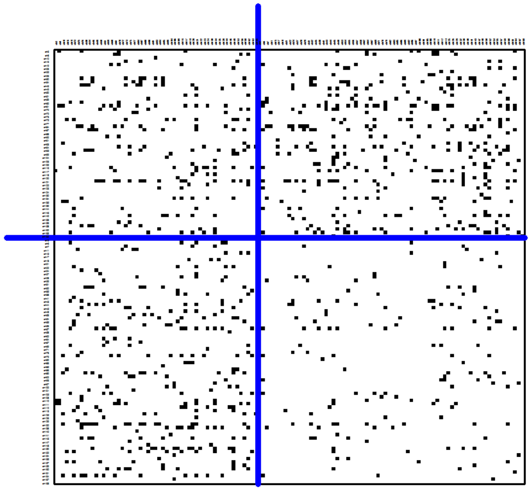
Binary only diagonal



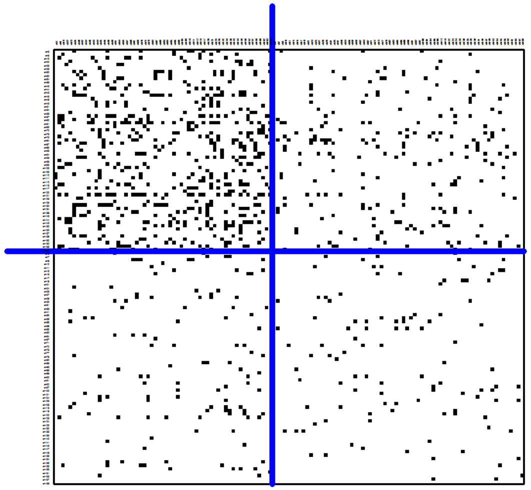
Binary only diag. - weighted



Sum of squares - only diag.



Sum of squares - all



Blockmodeling sparse binary networks - example

Cohesive groups model:

- 3 or 4 clusters/cohesive groups
- Pre-specified blockmodel for 3 clusters

	1	2	3
1	com	nul	nul
2	nul	com	nul
3	nul	nul	com

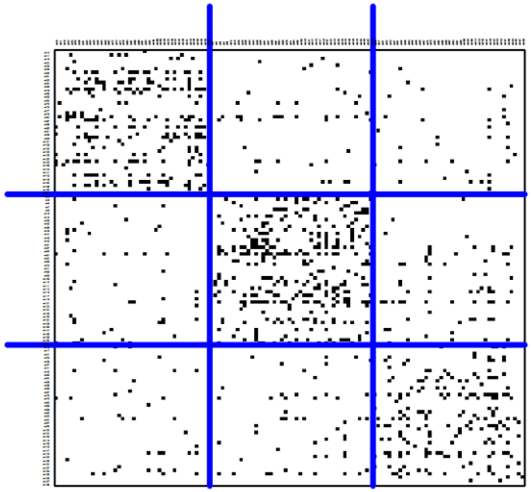
- Since this is very restrictive model binary blockmodeling according to structural equivalence (without weighting) can be used
- This model will minimize

  “the number of ‘null’ ties in the diagonal blocks” + “the number of ties in the off-diagonal blocks”

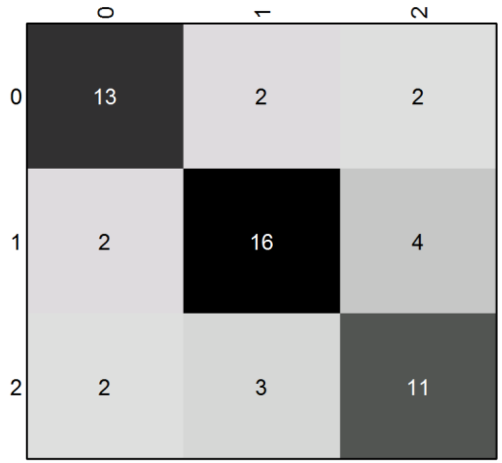
Blockmodeling sparse binary networks - example

- ○ Cohesive groups binary blockmodeling according to structural equivalence

3 clusters



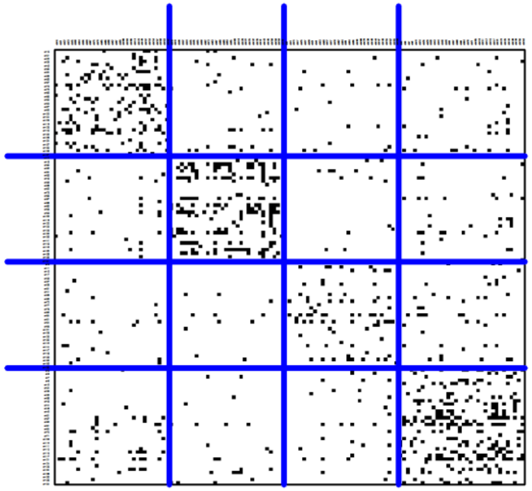
3 clusters - Image



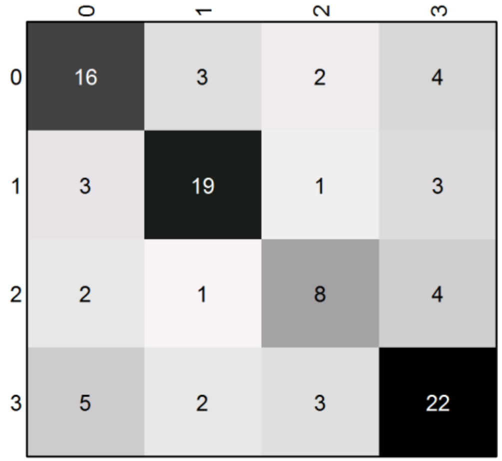
	0	1	2
0	13	2	2
1	2	16	4
2	2	3	11

* all values in cells were multiplied by 100

4 clusters



4 clusters - Image



	0	1	2	3
0	16	3	2	4
1	3	19	1	3
2	2	1	8	4
3	5	2	3	22

* all values in cells were multiplied by 100

Blockmodeling sparse binary networks - example

- "Understanding" partitions – association with other variables – 4 cluster cohesive groups

	solid tumors	hema-tology	surgery	public health	labo-ratory research	basic research
1	0.28	0.24	0.03	0.28	**0.72**	**0.72**
2	0.19	**0.71**	0.03	0.03	0.29	0.42
3	**0.68**	0.13	0.16	0.26	0.32	0.19
4	**0.71**	0.09	0.18	0.06	0.29	0.21

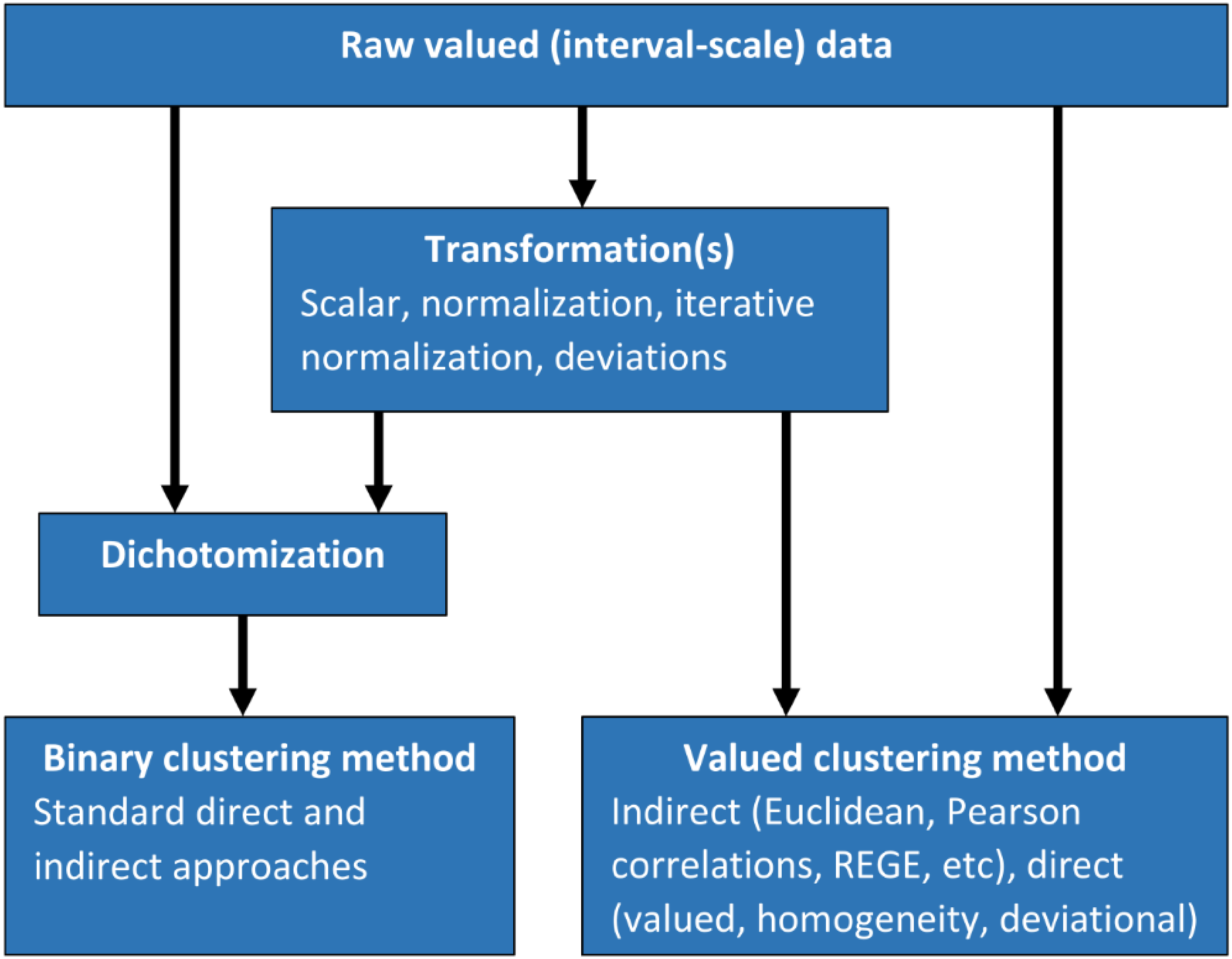
Different types of valued networks

- A subjective judgment using different rating scales (e.g. a 5-point scale). Features:
  - **subjective** (based on the perception of the responding actor).
  - **extremes are defined** and are the same for all actors.
  - **Example: Liking on 5-point likertscale**
- Direct measurements of **flows/frequencies/intensities** of interactions. Features:
  - **lack a pre-specified maximum**, natural limits may exist
  - the perception of **what is a strong tie** in such measurements varies among actors
  - **Example: Trade (in EUR) among EUR countries**

Relational capabilities of untis

	Similar	**Different**
actors have similar capabilities for creating ties	yes	no
have a similar upper ceiling	yes	no
level on which the threshold for “strong” tie is determined	network	actor or dyad
variance in valued in- and out-degrees	small	large

Possible paths of analysis



Raw valued (interval-scale) data

Transformation(s)
Scalar, normalization, iterative normalization, deviations

Dichotomization

Binary clustering method
Standard direct and indirect approaches

Valued clustering method
Indirect (Euclidean, Pearson correlations, REGE, etc), direct (valued, homogeneity, deviational)

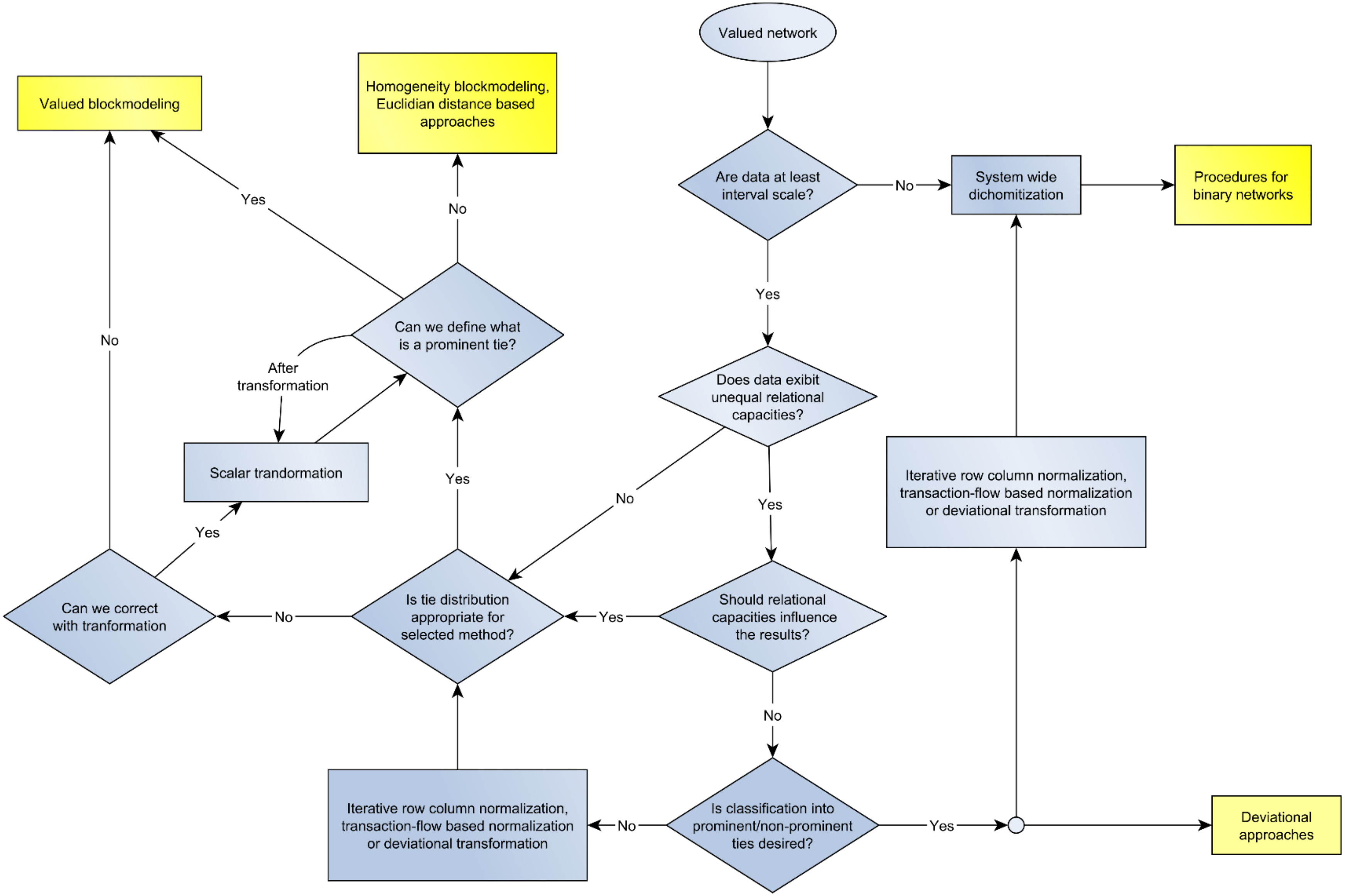
General recommendations

- Equal relational capacities?
  - Yes → approaches focusing on tie strengths
  - No → Should they be controlled for?
    - No → approaches focusing on tie strengths
    - Yes → correlation or deviational based approaches or strength-focused approaches after suitable transformations

General recommendations

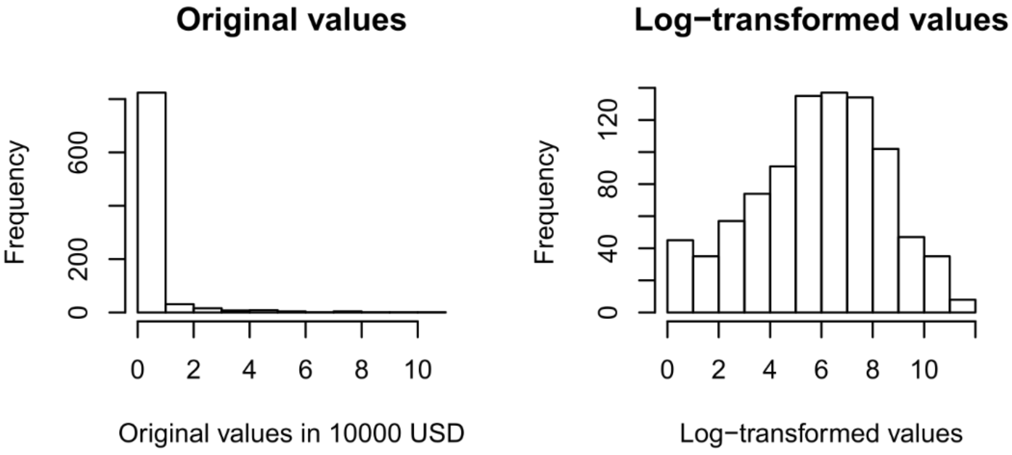
- Avoid the dichotomization, unless ordinal.
- The approach should be selected based on:
  - What ties or tie values mean
  - the properties of the blockmodeling solution we are looking for
- Direct (vs. indirect) approaches are preferred if possible in a reasonable time.
- Use different approaches for alternative “views”

Choice of methods



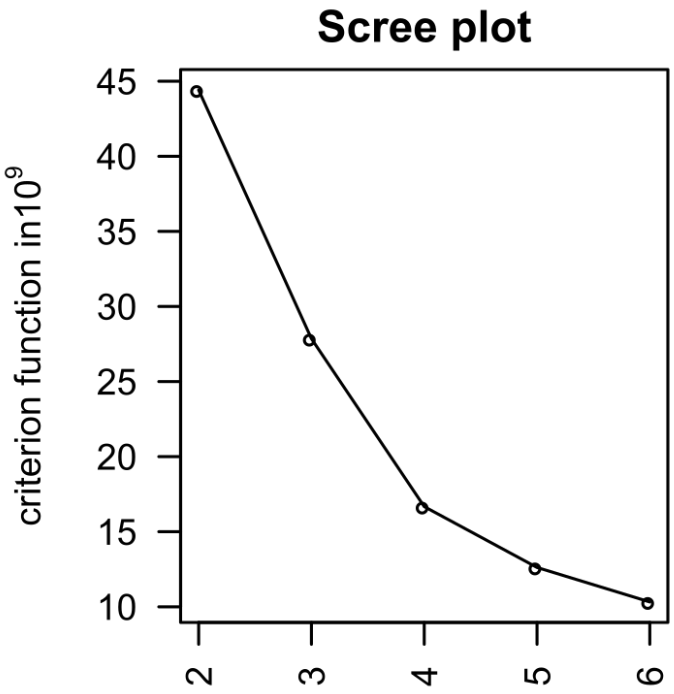
Example: **EU/EFTA trade**

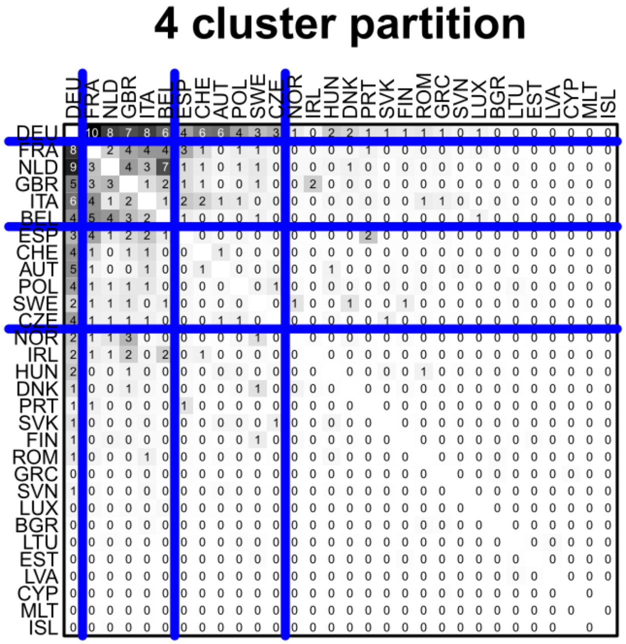
- **Commodity trade** among 30 EU/EFTA countries within for 2010 in **millions of USD**
- tie values: Range: 1 - 103,434; Mean: 3,954; Median: 483, SD:10,749 → very skewed
  - **No theoretical maximum, different capacities, no network wide (theoretical) threshold.**

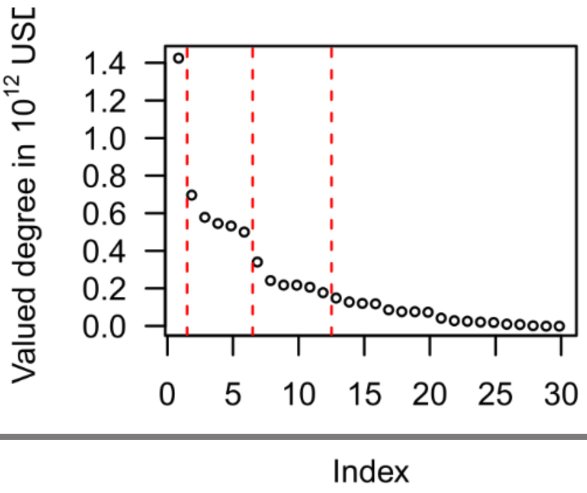


Example 2: EU/EFTA– original data

- SS approach
- Clusters determined by valued degree
- C-P model

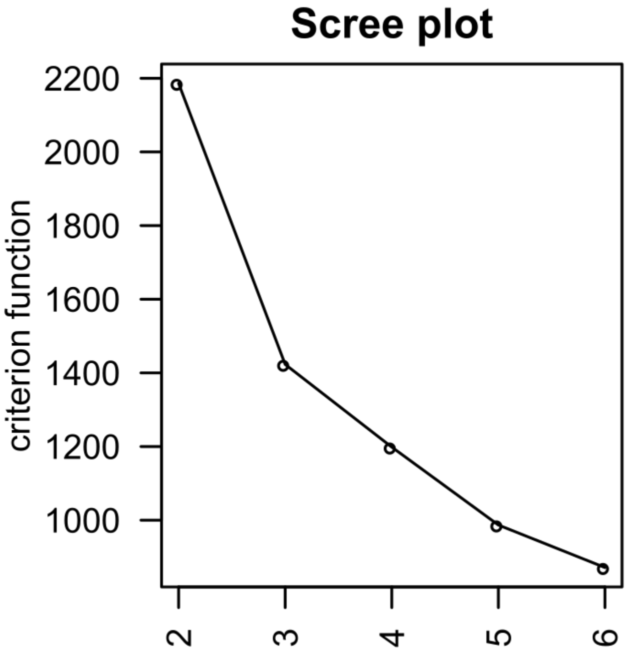


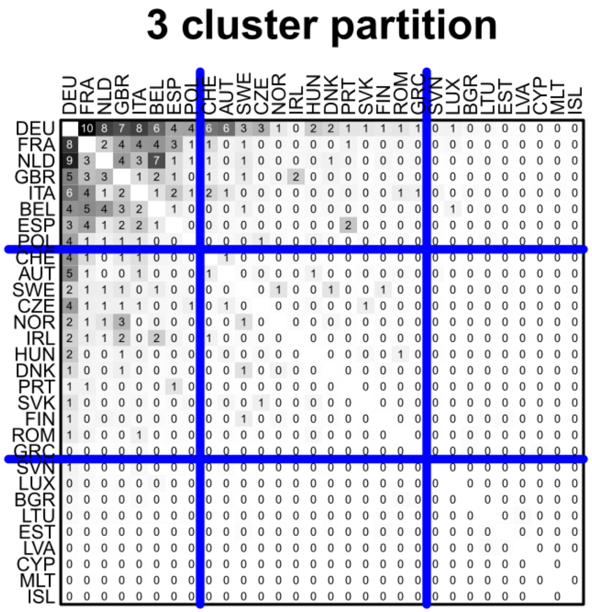




Example 2: EU/EFTA– log transf.

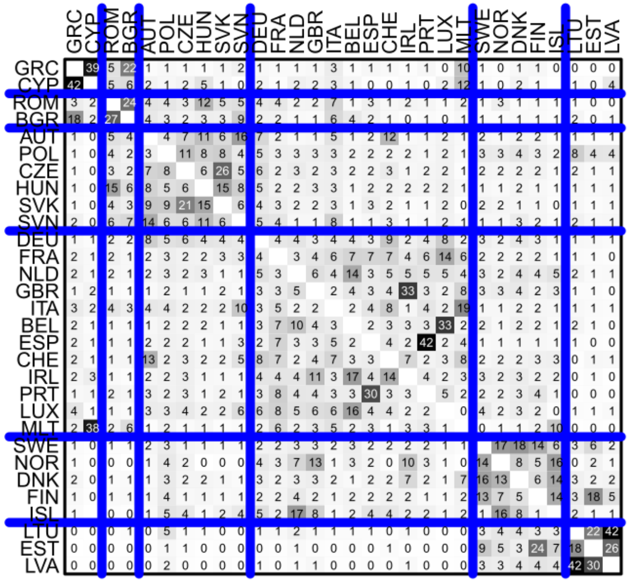
- SS approach
- Similar as before, Germany not so special anymore



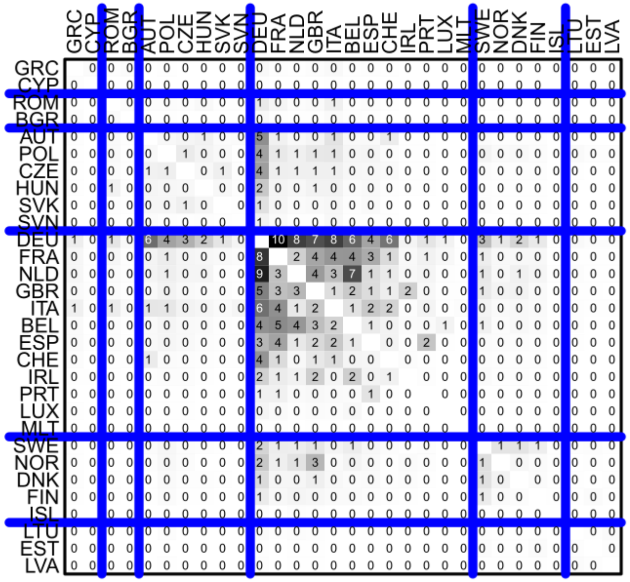


Example 2: EU/EFTA– iterated row/column marginal normalization

- AD approach
- Mainly cohesive groups - trade gravity effects



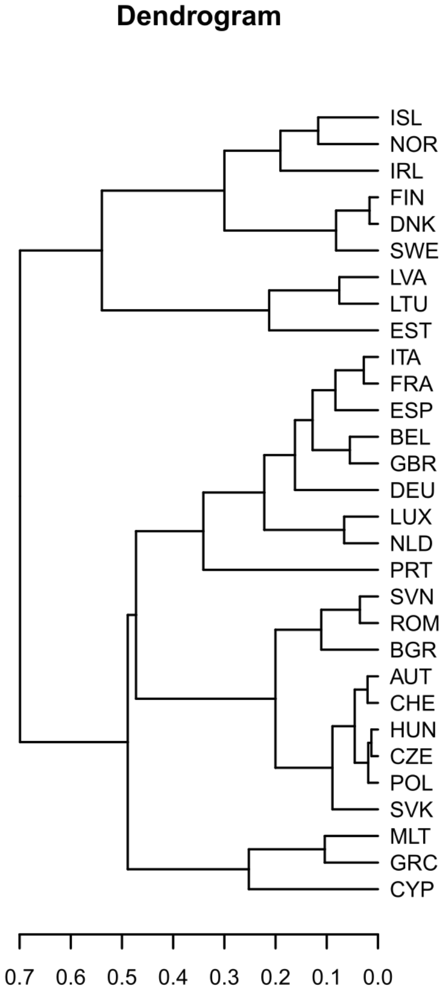
normalized, 6 clusters



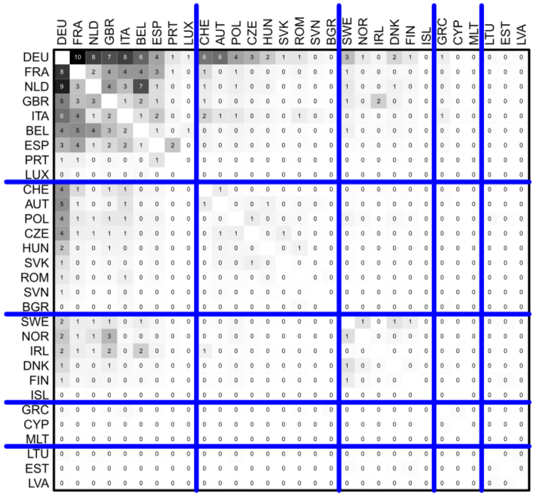
original, 6 clusters

Example 2: EU/EFTA– Ward on correlations

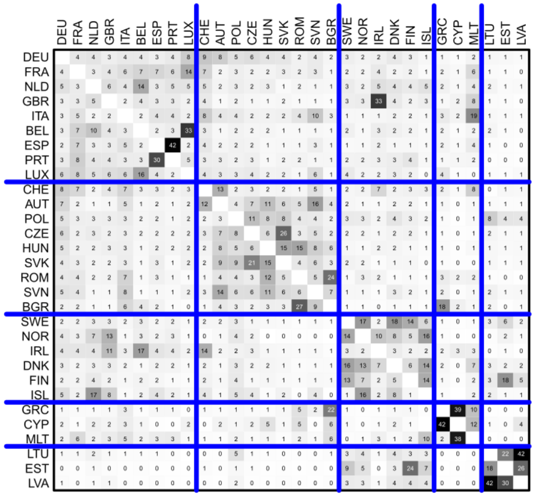
- Cohesive groups - trade gravity effects



Dendrogram



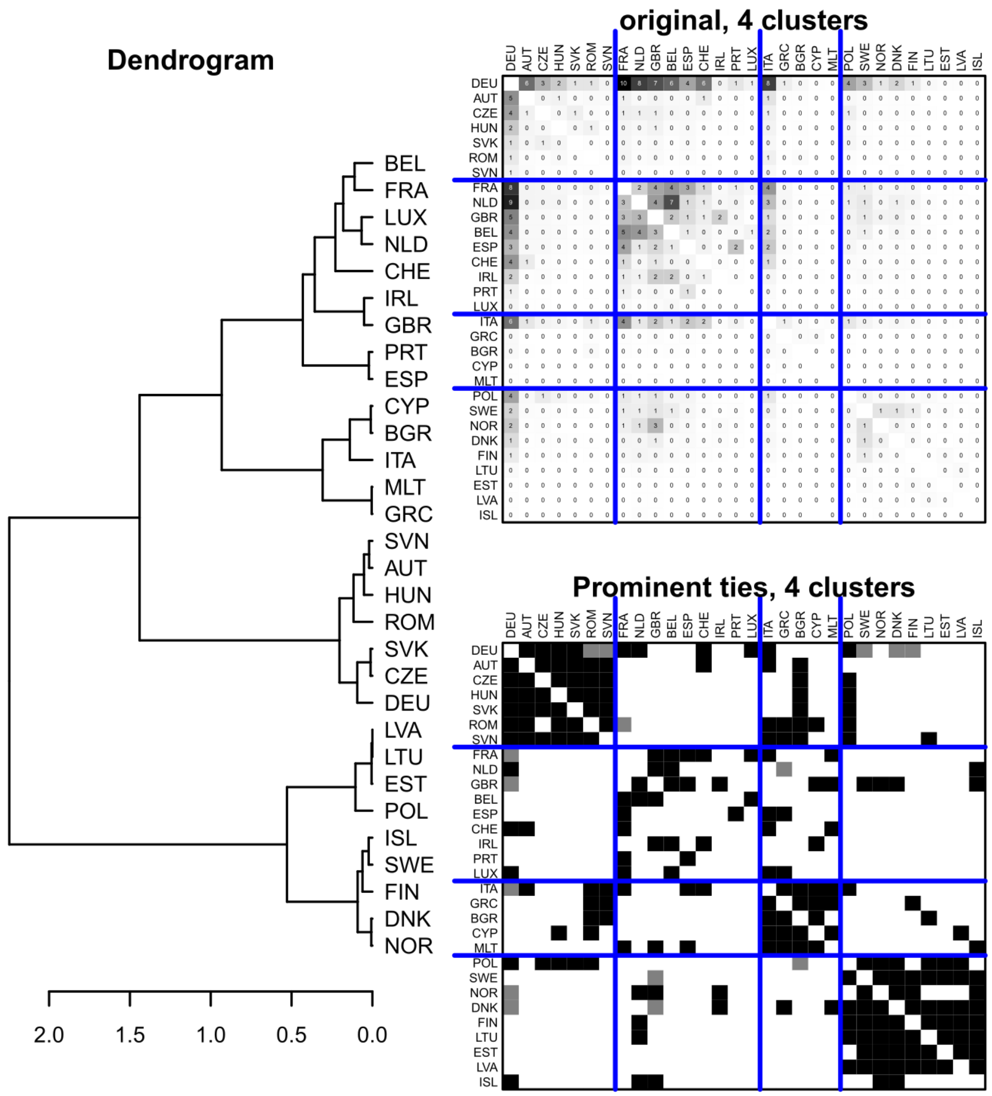
original, 5 clusters



normalized, 5 clusters

Example 2: EU/EFTA– Deviational approach

- Cohesive groups - trade gravity effects



Conclusions

- Many clustering methods, many possible transformation → An explosion of possible combinations
- Crucial questions: do units have equal relational capacities and does it matter?
- Often different analysis tell different stories

Multi-relational networks and Multi-criteria blockmodeling

Multi-relational networks:

- Implementation can be straightforward
- Questionable meaning (Doreain, Batagelj, Ferligoj, 2005)
- Weighting of different relations (differnet scales, characteristics, models, …) → Multi-criteria blockmodeling

Multi-criteria/objective blockmodeling:

- Weighted sum is the simplest approach
- Estimating the set of Pareto efficient blockmodels is probably better (Brusco et al., 2013)

Blockmodeling multilevel/linked networks (Žiberna 2014, 2020)

Type of data:

- Network(s) of individuals (level one units)
- Network(s) of institutions (level two units)
- Two-mode network tying individuals to institutions (ties between level one and level two units)

Aim:

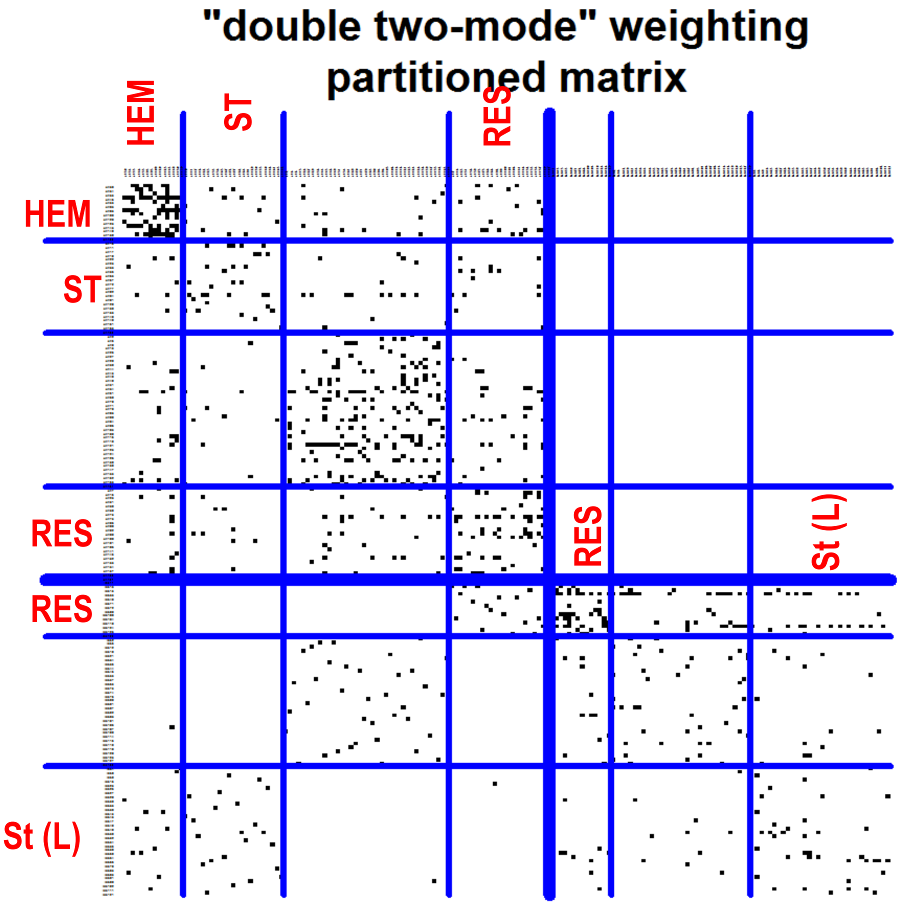
- **Find groups and ties among** them in the multilevel networks by taking account of **all levels** *and*
- taking account that there should be **some association** among groups from both levels

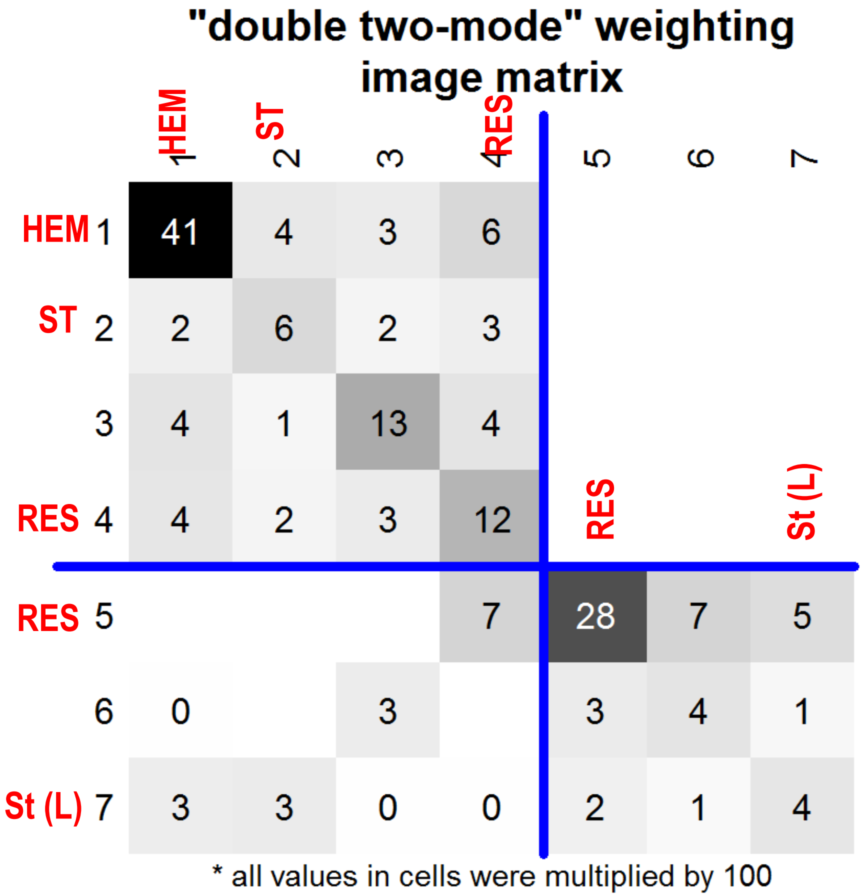
Possible approaches

- Partition both networks individually and compare results
- Partition the units of one level based on all available information
- Partition units of both levels simultaneously with some limitations based on two-mode network (ties among both levels) → True multilevel approach

Example: True multilevel approach

“Double two-mode weighting”: 1 for the network of researchers, 2.346 for the network of labs and 10.956 for two-mode network


"double two-mode" weighting partitioned matrix


"double two-mode" weighting image matrix

	HEM 1	ST 2	3	RES 4	RES 5	6	St (L) 7
HEM 1	41	4	3	6			
ST 2	2	6	2	3			
3	4	1	13	4			
RES 4	4	2	3	12			
RES 5				7	28	7	5
6	0		3		3	4	1
St (L) 7	3	3	0	0	2	1	4

* all values in cells were multiplied by 100

Conclusions – Multilevel blockmodeling

- Several approaches are possible
- Blockmodeling each level separately is a good exploratory technique and should always be the first step
- Approaches based on “combining” levels give us the possibility to obtain **one** partition based on all information → new insight, better partition
- Multilevel approach is the most flexible approach that promises the richest results, but not yet well tested (and most computer intensive)

Generalized blockmodeling
Opened issues

- Selection of number of clusters (relative fit?)
- Compatibility of block types in homogeneity blockmodeling
- Optimization (alternatives to local search), especially for multilevel
- Implementation (speed)
- Combining different types of generalized blockmodeling
- Blockmodeling valued networks were actors have different “relational capacity”
- Ordinal data
- ?Stochastic version?

Conclusions

- Blockmodeling is a technique for finding clusters of units in the network and, at the same time determining ties within and between these clusters.
- Numerous approaches (classical, generalized, stochastic, special cases)
- Approaches exist for binary and valued networks
- Generalized blockmodeling – can be used for numerous purposes (‘free’, pre-specified, …)
- Extensions exist for other types of networks (e.g. multilevel networks)
- Still a lot of open issues

References

- Borgatti S. P., Everett M. G. (1992): Regular blockmodels of multiway multimode matrices. *Social Networks, 14,* 91-120.
- Brusco, M., Doreian, P., Steinley, D., Satornino, C.B. (2013): Multiobjective Blockmodeling for Social Network Analysis. *Psychometrika, 78,* 498–525.
- Cugmas, M., Žiberna, A., & Ferligoj, A. (2021). The Relative Fit measure for evaluating a blockmodel. *Statistical Methods & Applications*. doi: 10.1007/s10260-021-00595-1
- Doreian P., Batagelj V., Ferligoj A. (2005): *Generalized Blockmodeling*. New York: Cambridge University Press.
- Žiberna, A. (2007): Generalized Blockmodeling of Valued Networks. *Social Networks, 29 (1),* 105-126.
- Žiberna, A. (2008): Direct and Indirect Approaches to the Blockmodeling of Valued Networks in Terms of Regular Equivalence. *Journal of Mathematical Sociology, 32,* 57–84.
- Žiberna, A. (2009): Evaluation of Direct and Indirect Blockmodeling of Regular Equivalence in Valued Networks by Simulations. *Metodološki zvezki 6(2),* 99–134.
- Žiberna, A. (2013): Generalized Blockmodeling of Sparse Networks. *Metodološki zvezki 10(2)*, 99–119.
- Žiberna, A. (2014). Blockmodeling of multilevel networks. *Social Networks*, *39*, 46–61. doi: 10.1016/j.socnet.2014.04.002
- Žiberna, A. (2020). Blockmodeling Linked Networks. In *Advances in Network Clustering and Blockmodeling* (pp. 267–287). John Wiley & Sons, Ltd. doi: 10.1002/9781119483298.ch10
- Žiberna, A. (2022): *blockmodeling 1.1.3: An R package for generalized and classical blockmodeling of valued networks*. http://aleszib.cmi-fdv.si/doku.php?id=blockr.

Questions?

Aleš Žiberna
Faculty of Social Sciences
University of Ljubljana

Email: ales.ziberna@fdv.uni-lj.si

My R package blockmodeling:
http://aleszib.cmi-fdv.si/doku.php?id=blockr