Blockmodeling
Indirect approach
Direct approach - structural equivalence
Direct approach - regular equivalence
Direct approach - generalized equivalence
Direct approach - regular equivalence

# Social Network Analysis - blockmodeling Lab sessions

Anja Žnidaršič

University of Maribor

Moscow, ANR-Lab, 28 May 2019

# Outline



1 Blockmodeling
- Indirect approach
- Direct approach - structural equivalence
- Direct approach - regular equivalence
- Direct approach - generalized equivalence
- Direct approach - regular equivalence

# Blockmodeling: indirect approach



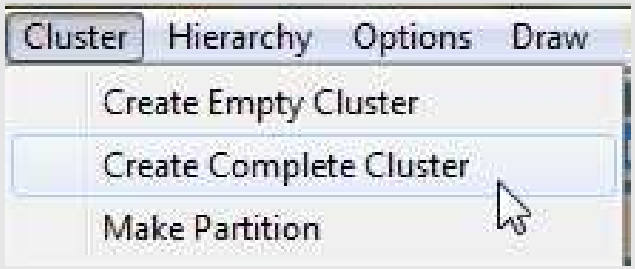
## Commands in Pajek:

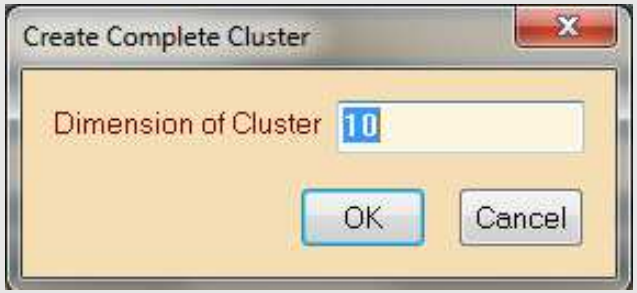
Network used in the example: the Everett network

1. Create complete cluster:

   `Cluster` → `Create Complete Cluster`→

   (*New window: Dimension of Cluster*) [10]→OK



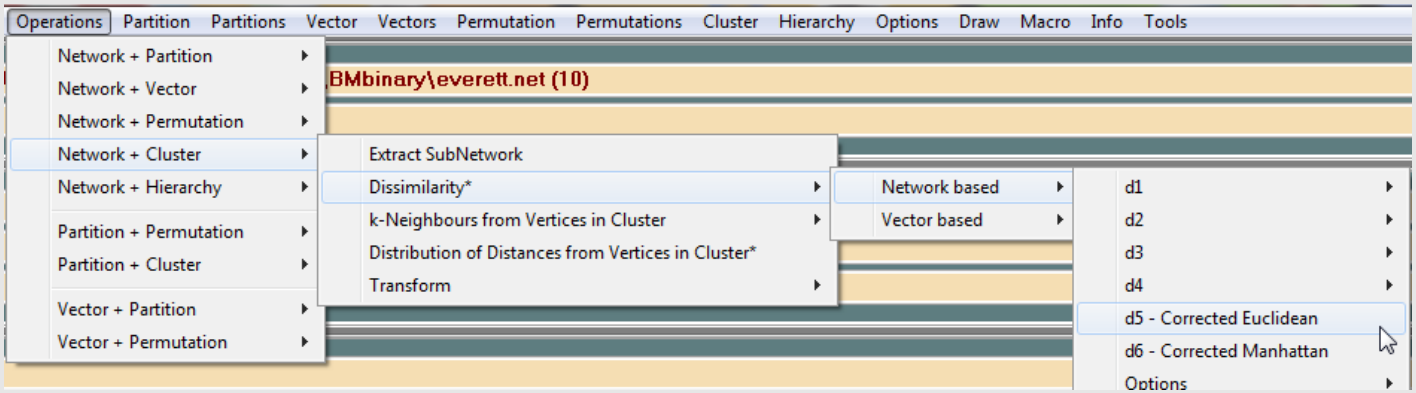
# Blockmodeling: indirect approach



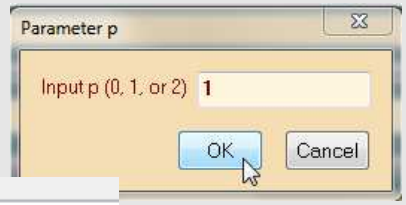
## Commands in Pajek:

2. Compute dissimilarities:

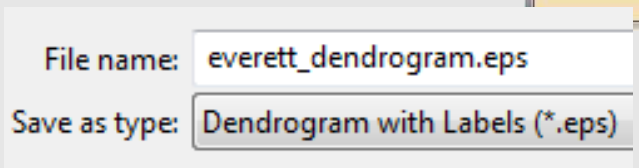
`Operations` → `Network + Cluster` → `Dissimilarity*` → `Network based` → `d5 - Corrected Euclidean` →




→(New window: Parameter p) [1]→OK




→ Save as window:

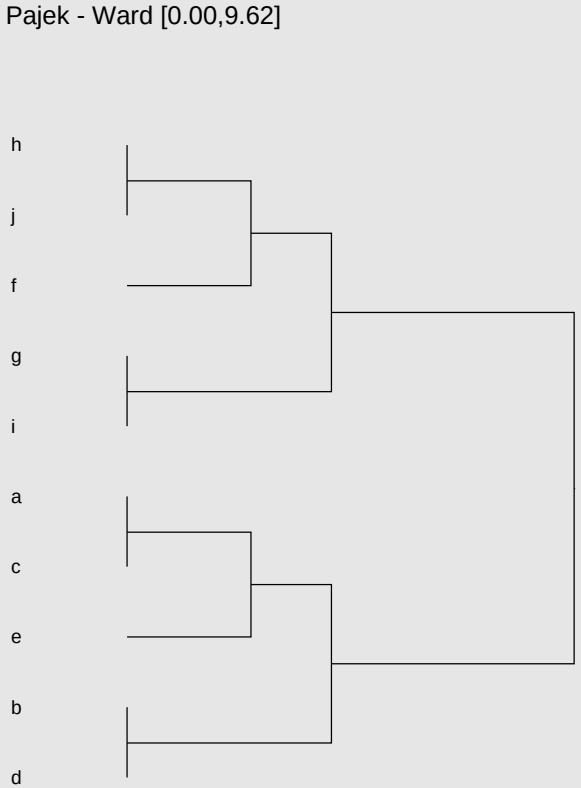


# Blockmodeling: indirect approach



## Commands in Pajek:

3. Check the dendrogram and select the appropriate number of clusters:




Dendrogram clearly shows two clusters.

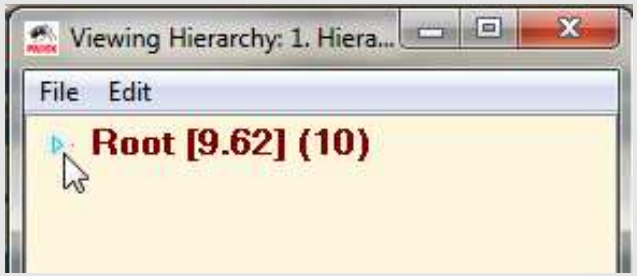
# Blockmodeling: indirect approach



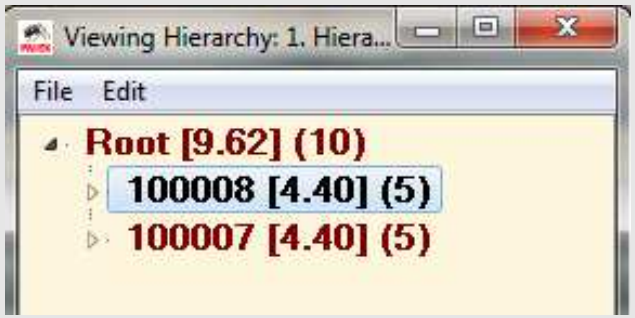
## Commands in Pajek:

4. Select clusters according to hierarchy:
   - click on View/Edit in Hierarchy panel


   - open/close different levels by clicking on an arrow




   - select desired clusters



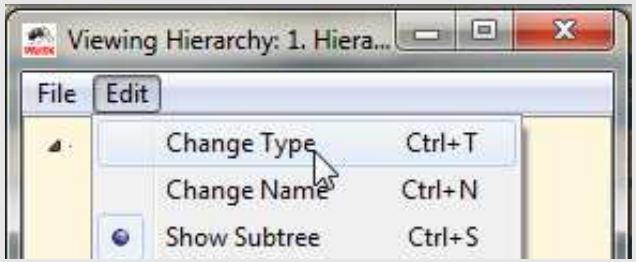
# Blockmodeling: indirect approach

A. Žnidaršič

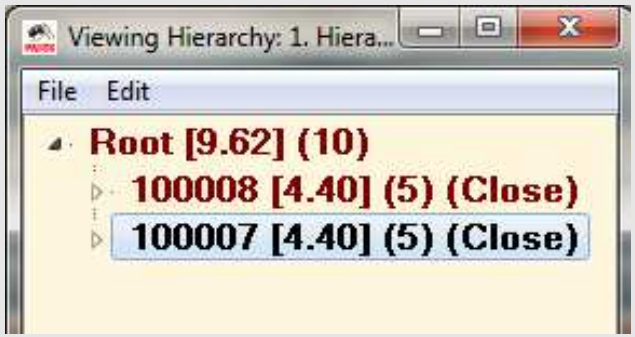
Blockmodeling
Indirect approach
Direct approach - structural equivalence
Direct approach - regular equivalence
Direct approach - generalized equivalence
Direct approach - regular equivalence

## Commands in Pajek:

- change the type of the clusters
  `Edit` → `Change Type`




- desired clusters, close 'Viewing Hierarchy' window

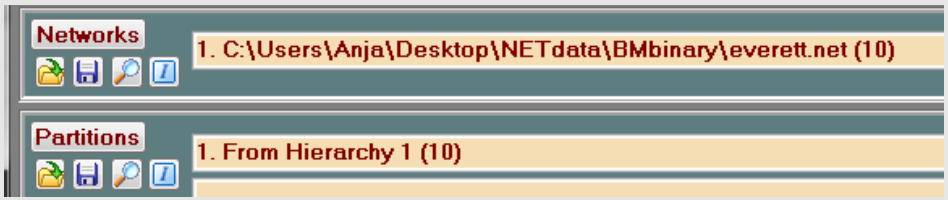


# Blockmodeling: indirect approach

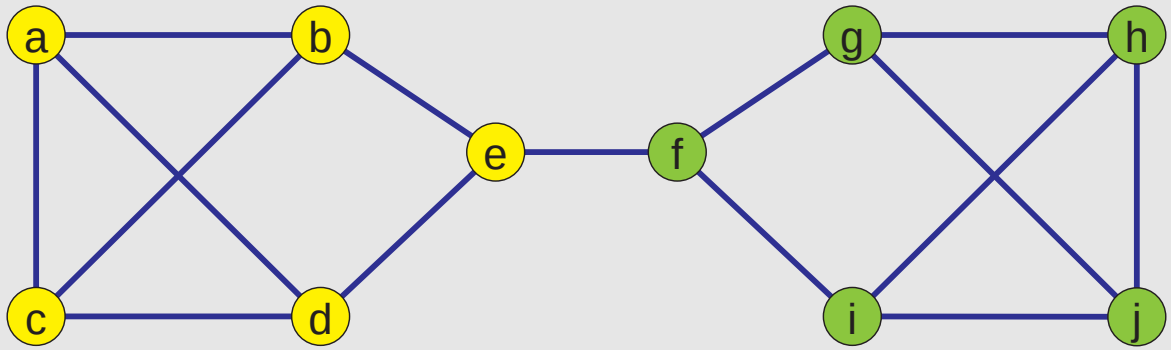


## Commands in Pajek:

5. Draw the network with obtained partition from hierarchy:
   - *Hierarchy* → *Make Partition*
   - Select the original network in the first row of the Network panel.




   - *Draw* → *Network + First Partition*

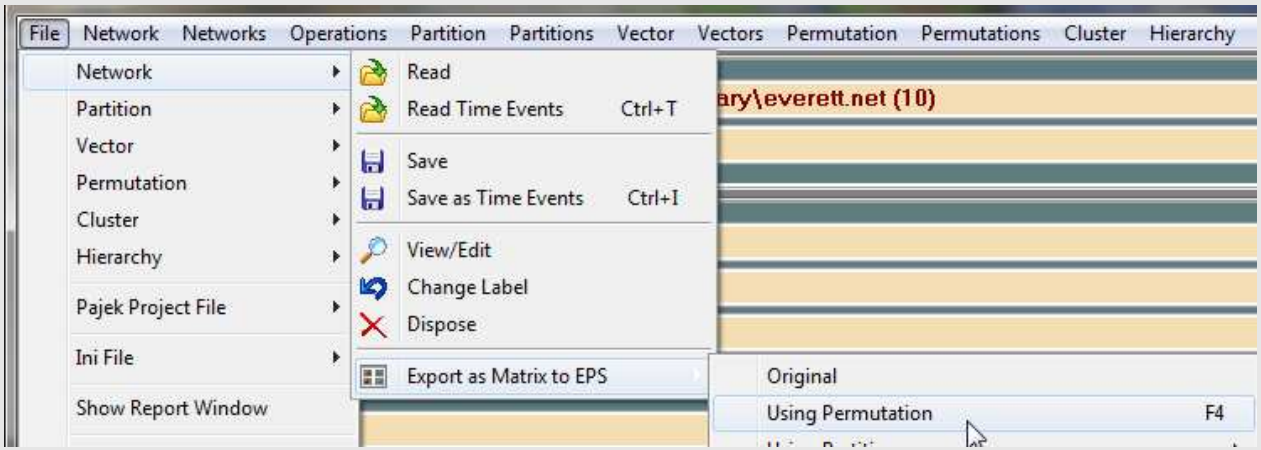


# Blockmodeling: indirect approach



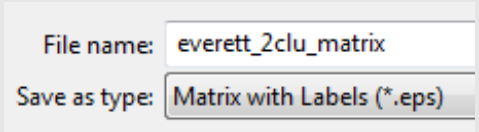
## Commands in Pajek:

6. Export the matrix with:
   - *Hierarchy* → *Make Permutation*
   - *File* → *Network* → *Export as Matrix to EPS* → *Using Permutation*




→ *Draw lines according to partition?* → Yes

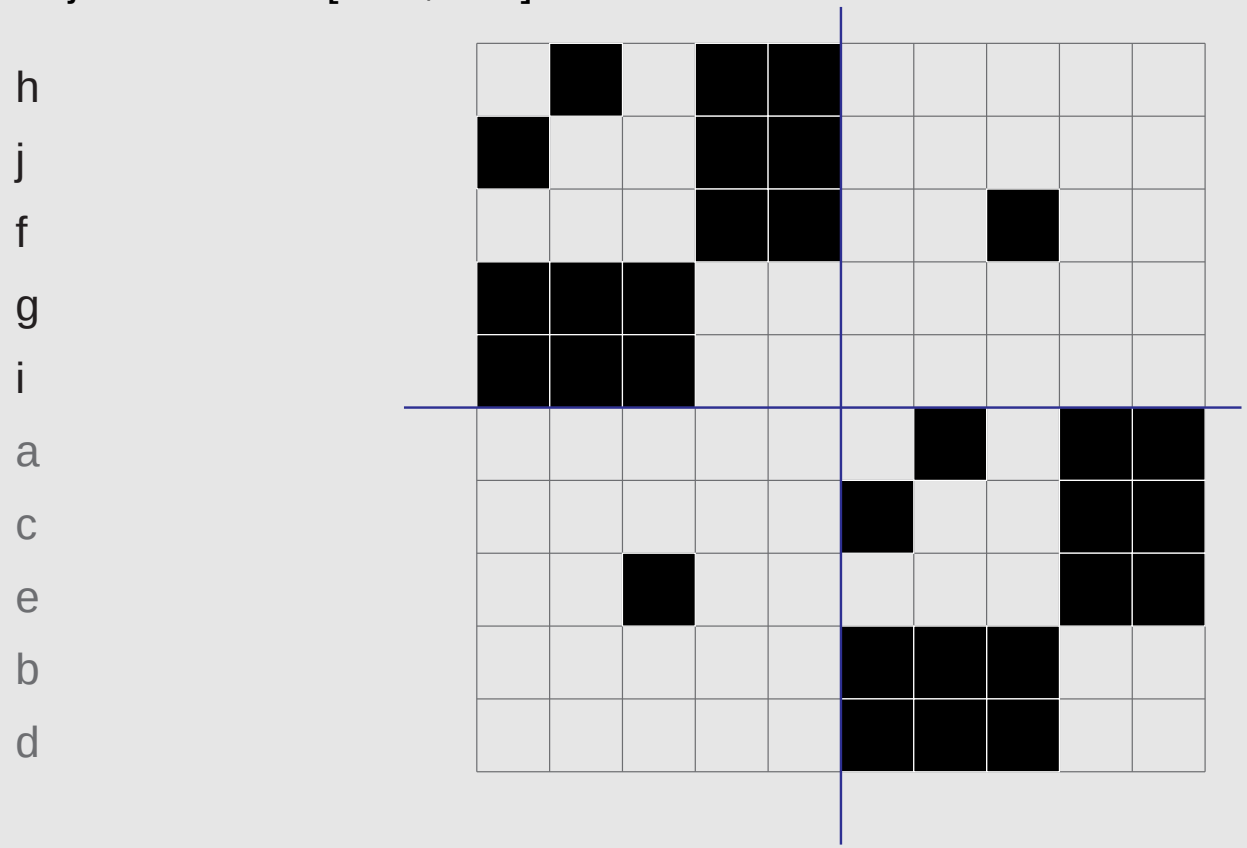
→ Save as window:



# Indirect approach



## The obtained matrix:



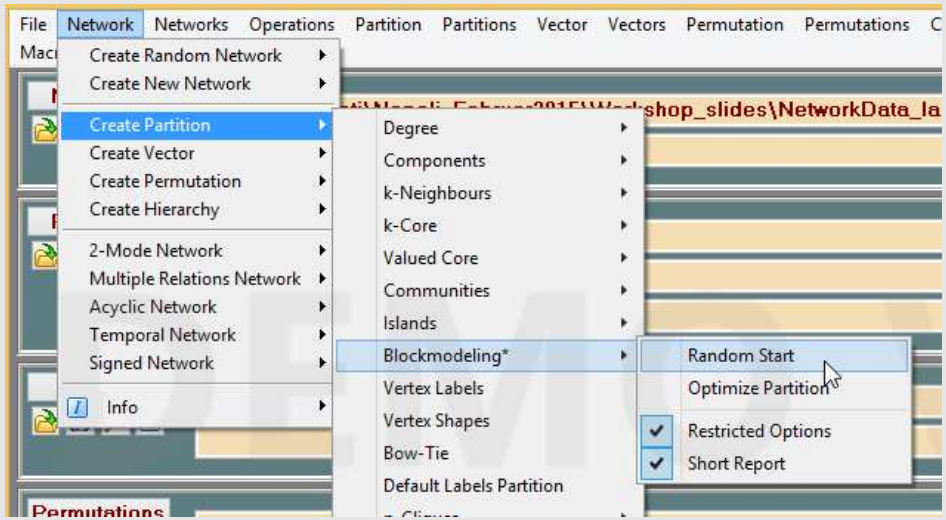
# Direct approach - structural equivalence



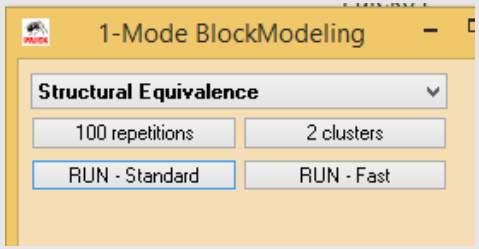
## Commands in Pajek:

1. Run the blockmodeling based on structural equivalence:

   *Network* → *Create Partition* → *Blockmodeling** → *Random Start*




→ New window:

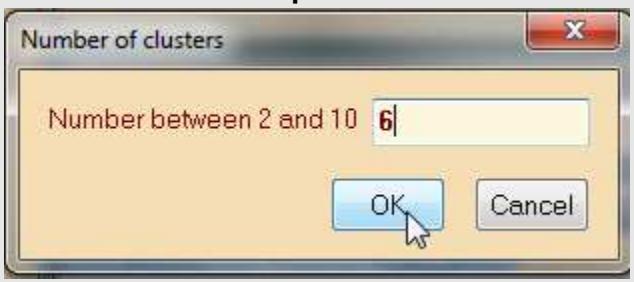


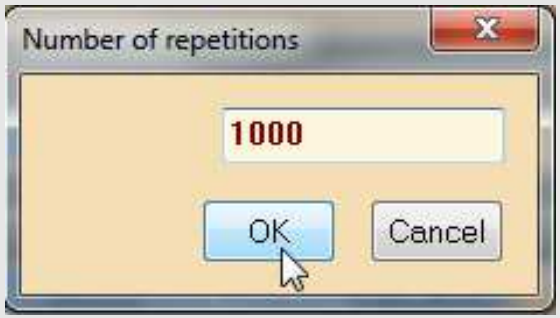
# Direct approach - structural equivalence



## Commands in Pajek:

2. In the '1-Mode Blockmodeling' window select:
   - Type of equivalence
   - Number of repetitions
   - Number of clusters
   - Click the `RUN` button.



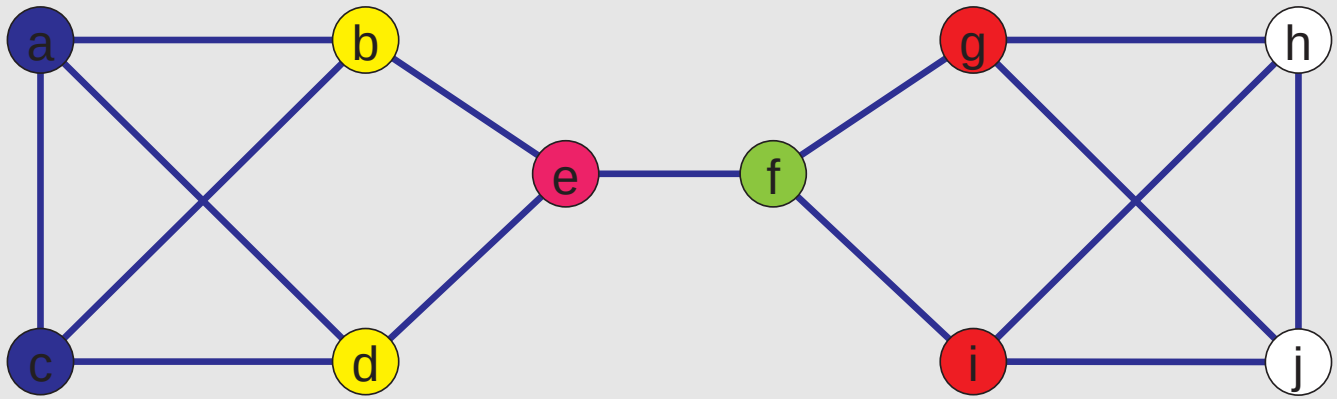

1. Draw the network with obtained partition:

   `Draw -> Network + First Partition.`

# Direct approach - structural equivalence



The obtained network with partition into 6 clusters based on structural equivalence:



# Direct approach - regular equivalence

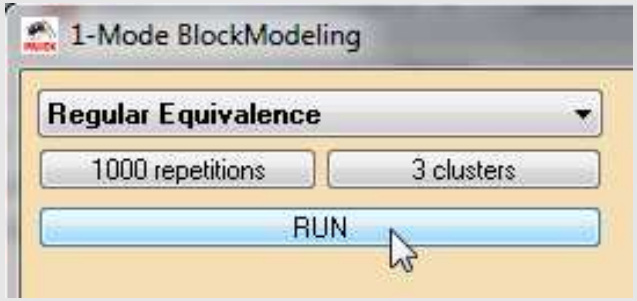


## Commands in Pajek:

1. Run the blockmodeling based on regular equivalence:
   *Network* → *Create Partition* → *Blockmodeling** → *Random Start*

In the '`1-Mode Blockmodeling`' window select:

- type of equivalence
  `Regular equivalence`
- number of repetitions, e.g.
  `[1000]`
- number of clusters
  `[3]`

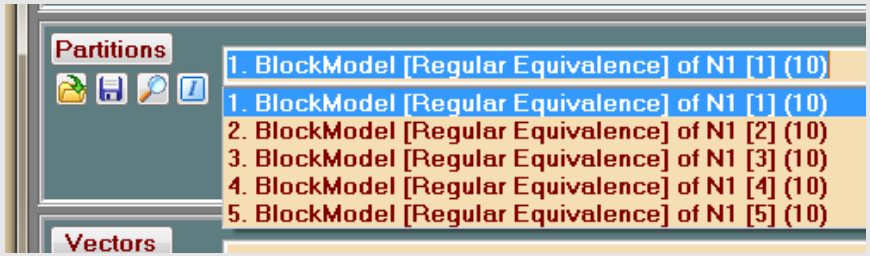


# Direct approach - regular equivalence

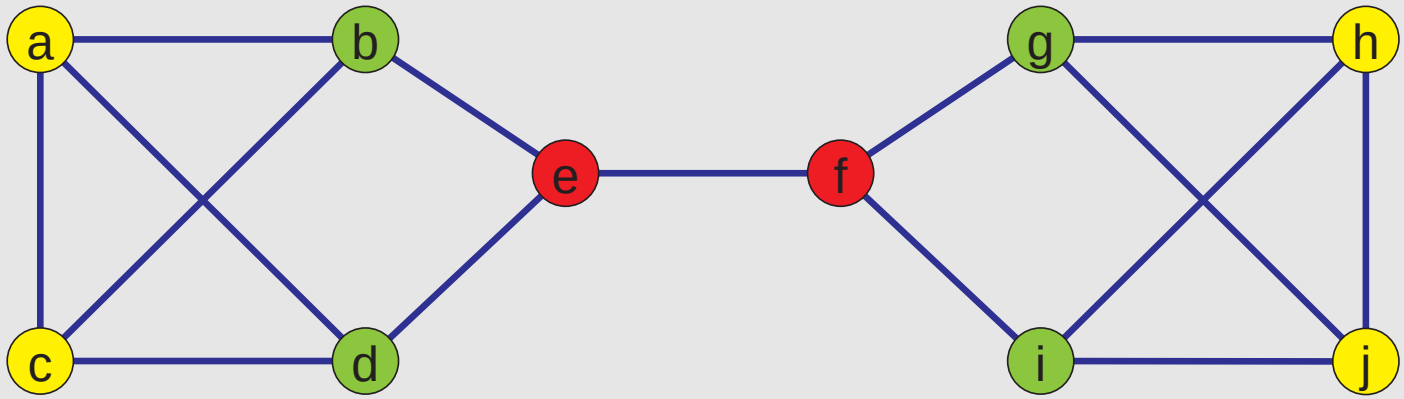


## Results:

We get five equally[1] well fitting partitions which can be observed from the report window or Partitions panel.




One of the obtained equally well fitting partitions:




[1] Blockmodeling is local optimization algorithm. There are 7 equally well fitting partitions for k = 3 and regular equivalence.

# Direct approach - generalized equivalence



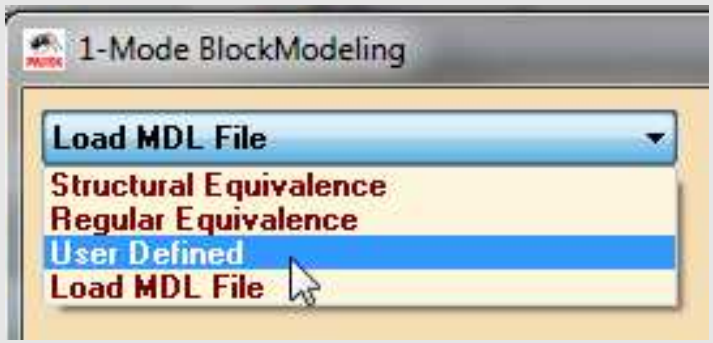
## Commands in Pajek:

Network used in the example: the Transatlantic Industries network (from the Baseball LittleLeague collection of networks)

1. Run the blockmodeling based on generalized equivalence:
   *Network* → *Create Partition* → *Blockmodeling** → *Random Start*

   In the '`1-Mode Blockmodeling`' window select:

   - type of equivalence
     `User Defined`
   - number of repetitions, e.g.
     `[1000]`
   - number of clusters
     `[4]`

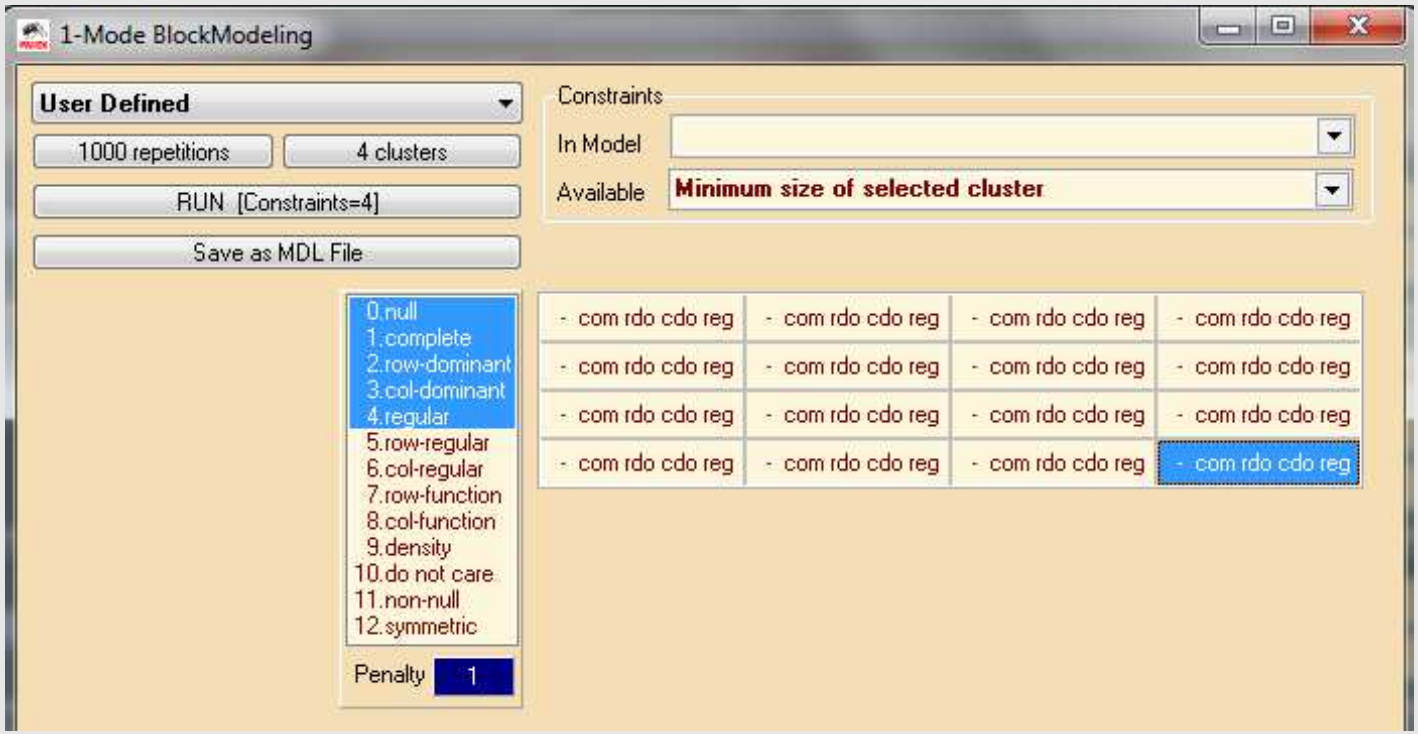


# Direct approach - generalized equivalence



## Commands in Pajek:

2. For each of 16 blocks we select `null`, `complete`, `row-dominant`, `col-dominant`, and `regular block` as possible blocks.



# Direct approach - generalized equivalence



## Commands in Pajek:

We get 10 equally well fitting partitions with 0 onconsistencis, but several clusters have only one vertex.

3. Therefore, we write a constraints in MDL file (see Pajek manual, page 102).

NullComRegRdoCdo_min2vertices_4clu_mdl -

File Edit Format View Help

```
*MODEL UserDefined
 4
  1  1          1 0 1 2 3 4
  1  2          1 0 1 2 3 4
  1  3          1 0 1 2 3 4
  1  4          1 0 1 2 3 4
  2  1          1 0 1 2 3 4
  2  2          1 0 1 2 3 4
  2  3          1 0 1 2 3 4
  2  4          1 0 1 2 3 4
  3  1          1 0 1 2 3 4
  3  2          1 0 1 2 3 4
  3  3          1 0 1 2 3 4
  3  4          1 0 1 2 3 4
  4  1          1 0 1 2 3 4
  4  2          1 0 1 2 3 4
  4  3          1 0 1 2 3 4
  4  4          1 0 1 2 3 4
*CONSTRAINTS
 5 100 2 1
 5 100 2 2
 5 100 2 3
 5 100 2 4
*EOM
```

# Direct approach - generalized equivalence

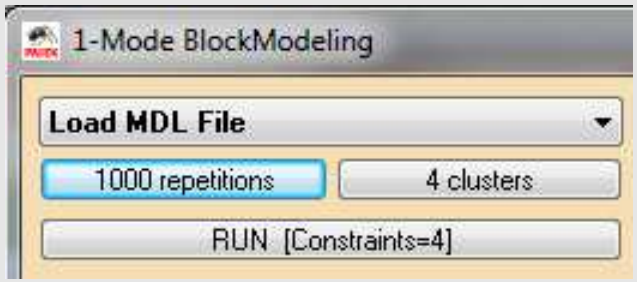


## Commands in Pajek:

4. Load and run the mdl file:

   *Network* → *Create Partition* → *Blockmodeling*$^*$ → *Random Start*

5. In the '`1-Mode Blockmodeling`' window select:
   - type of equivalence
     `Load MDL File` → select the prepared mdl file
   - number of repetitions, e.g. `[1000]`
   - number of clusters `[4]`




6. Draw the network with obtained partition:

   `Draw - > Network + First Partition.`

# Direct approach - generalized equivalence



## Result - report window:

Final Image Matrix:

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | com | - | - | cdo |
| 2 | - | - | - | cdo |
| 3 | - | - | com | cdo |
| 4 | rdo | rdo | - | reg |

Final Error Matrix:

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 |

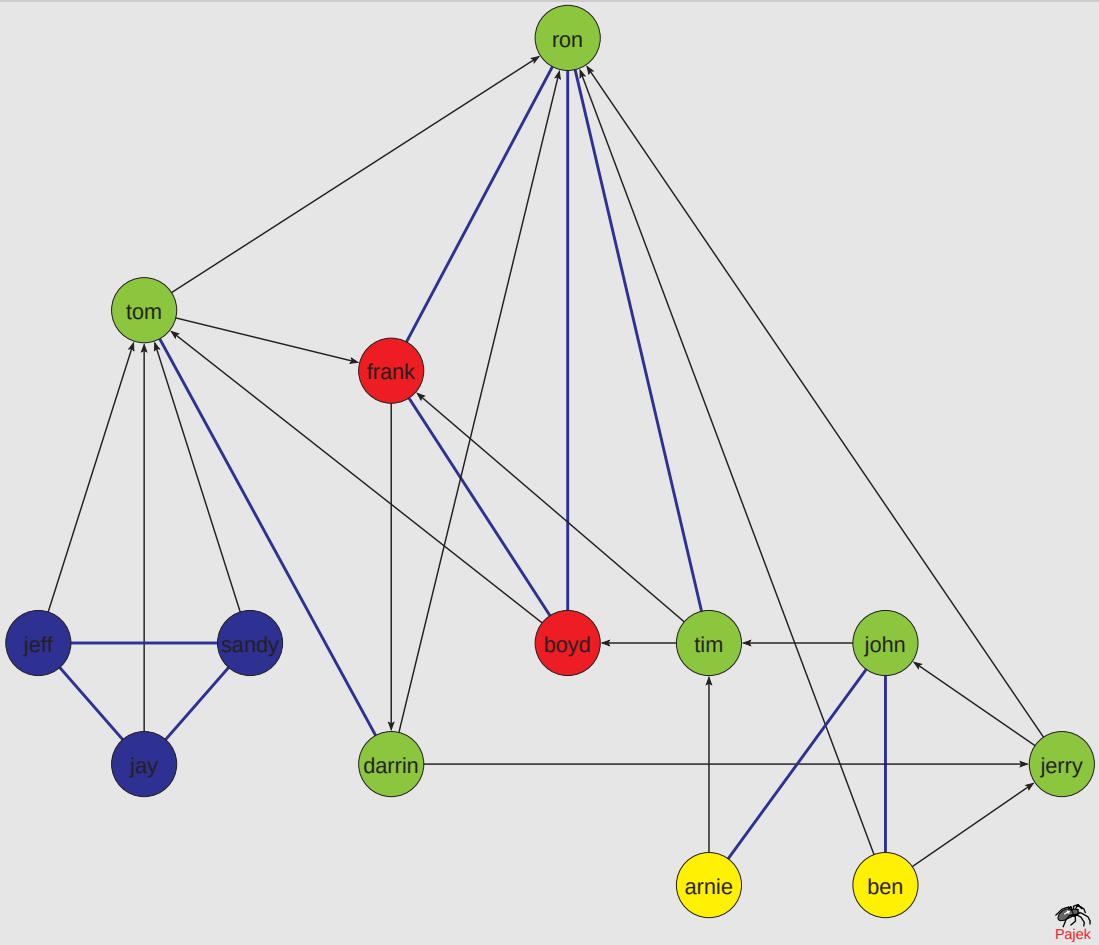
Final error = 0.000

# Direct approach - generalized equivalence



Blockmodeling
Indirect approach
Direct approach - structural equivalence
Direct approach - regular equivalence
Direct approach - generalized equivalence
Direct approach - regular equivalence

## Result:



# Direct approach - pre-specified



## Commands in Pajek:

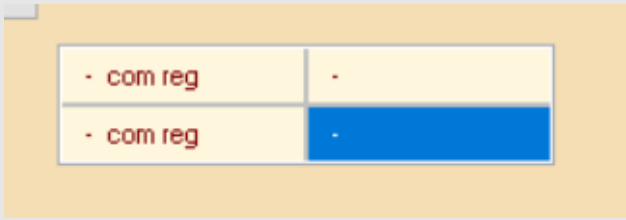
1. Run the blockmodeling based on pre-specified model
   The three basic types of models are: core-periphery, hierarchical, and cohesive subgroups.

Network used in the example: borrowing study materials

*Network* → *Create Partition* → *Blockmodeling** → *Random Start*

In the '`1-Mode Blockmodeling`' window select:

- type of equivalence
  `User defined`
- number of repetitions, e.g.
  `[1000]`
- number of clusters



# Direct approach - pre-specified



Blockmodeling
Indirect approach
Direct approach - structural equivalence
Direct approach - regular equivalence
Direct approach - generalized equivalence
Direct approach - regular equivalence

## Results: Image matrix

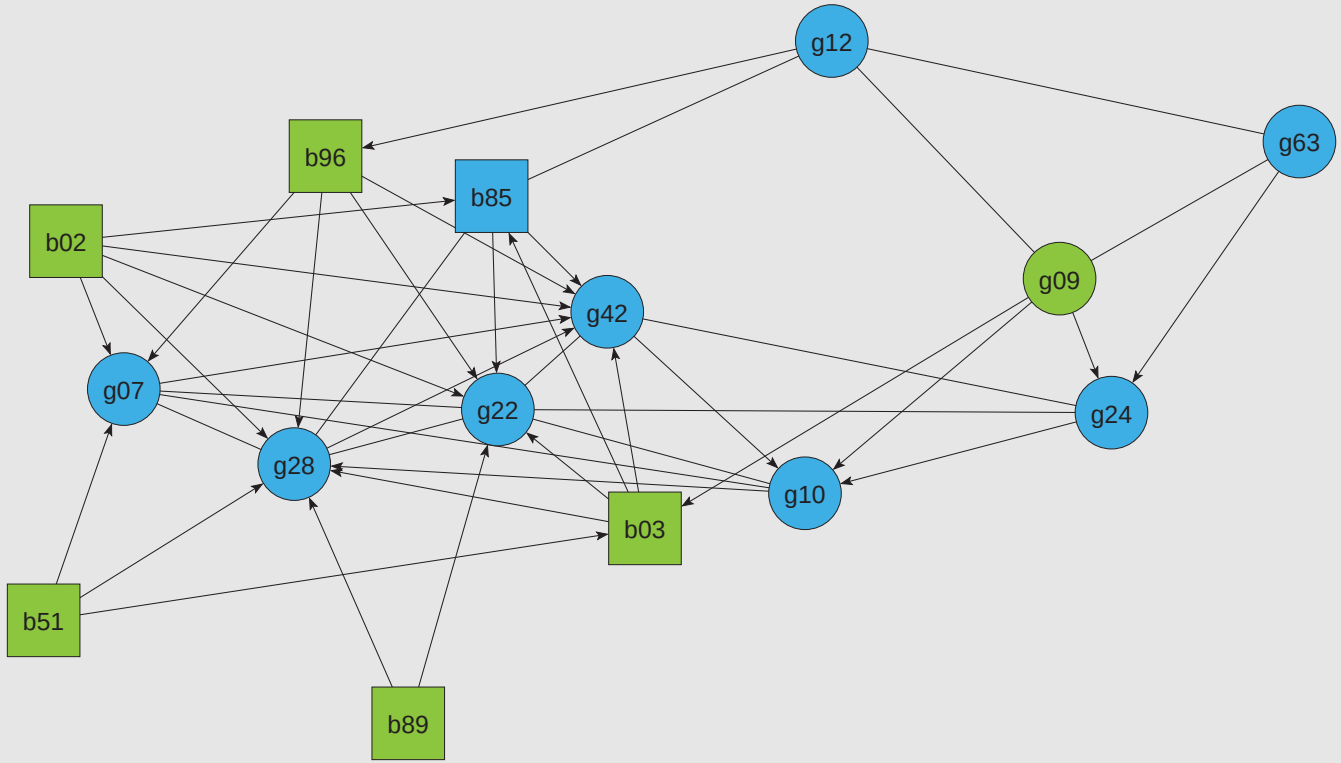
```
Final Image Matrix:

       1   2
  1  reg   -
  2  reg   -


Final Error Matrix:

       1     2
  1    0     3
  2    0     2

Final error = 5.000
```

# Direct approach - pre-specified



Blockmodeling
Indirect approach
Direct approach - structural equivalence
Direct approach - regular equivalence
Direct approach - generalized equivalence
Direct approach - regular equivalence

## Results: Network with obtained partition



# Next lab session:



## Announcement

Blockmodeling in R: package `blockmodeling` (author: Žiberna Aleš)