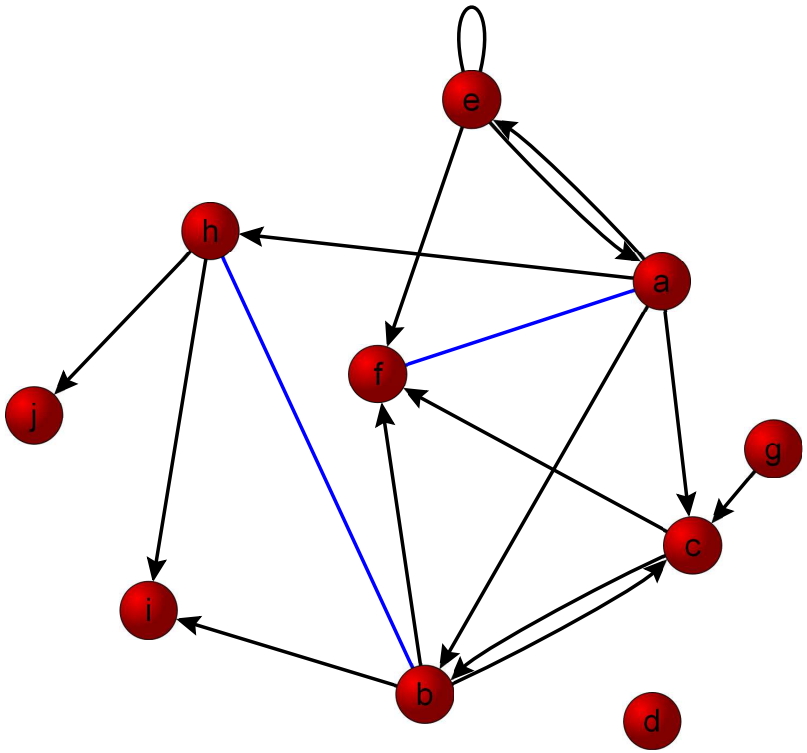
Introduction to Social Network Analysis with Pajek
Day 1

Anja Žnidaršič

University of Maribor, Faculty of Organizational Sciences

Outline

A network



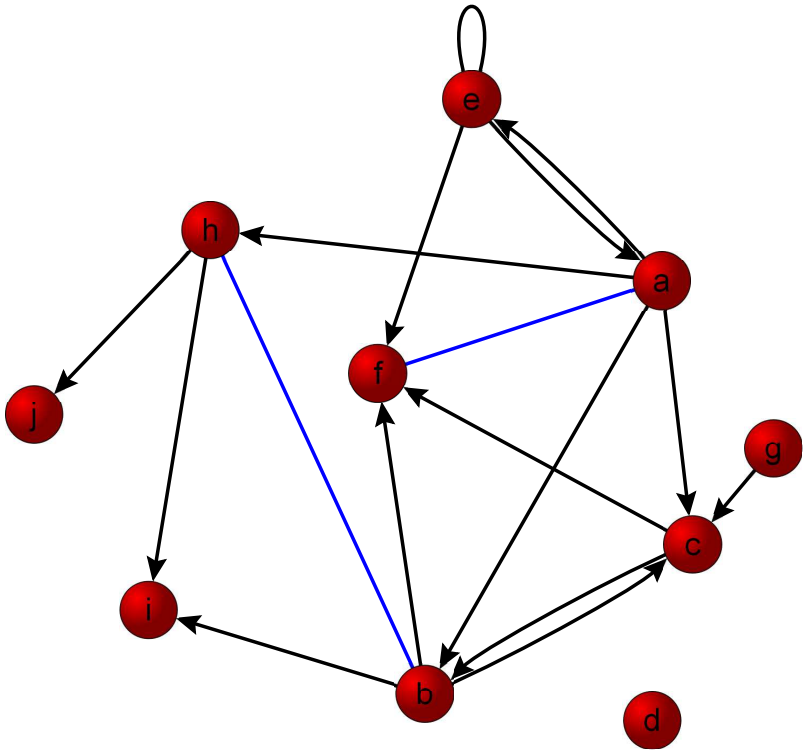
A network consist of two sets:

- set of vertices (or nodes) that represent the selected units and
- set of lines (or links) that represents ties between units.

Set of vertices and set of links determine a graph.

A graph with additional data is **a network**.

A network



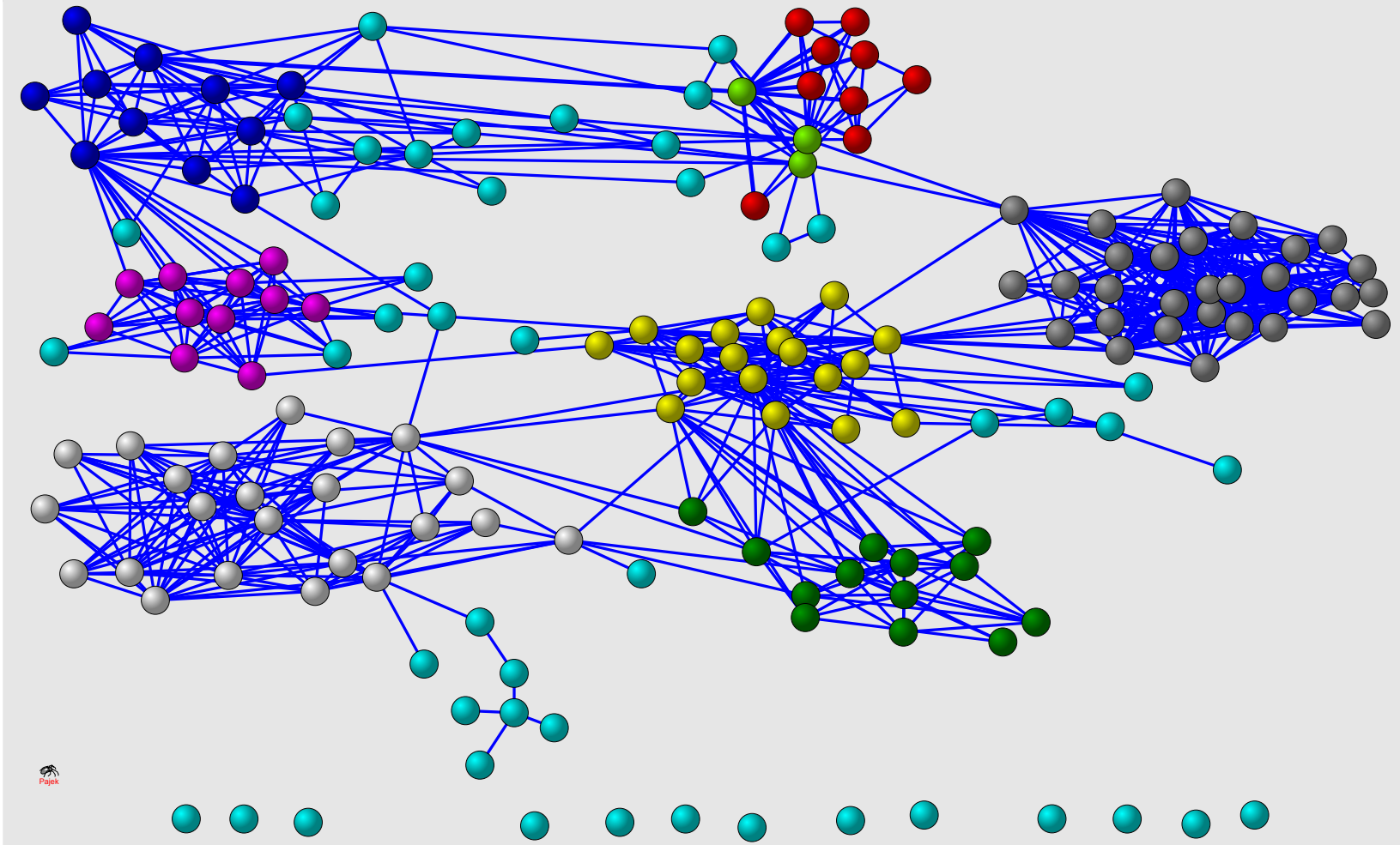
Additional data about vertices or links can be known: name, value, type, ...

Links in a network could be:

- directed - an edge
- undirected - an arc.

Network

Example of an undirected network



Types of network

Beside distinguishing between (binary) directed and undirected networks also other types of networks are known:

- signed networks (three types of ties: positive, negative, and null),
  - like-dislike,
  - respect-disrespect,
  - win-loose,
  - a friend - not be a friend,
- valued networks,

  Weigths (values on lines) in valued networks could represents:
  - trade between countries,
  - distance between cities, airports,...
  - number of passengers, weight or number of pieces of cargo,...
  - number of cars, buses, trains,... between two cities,...
- 2-mode networks,
- multi-relational networks

Pajek - Program for Large Network Analysis

Pajek

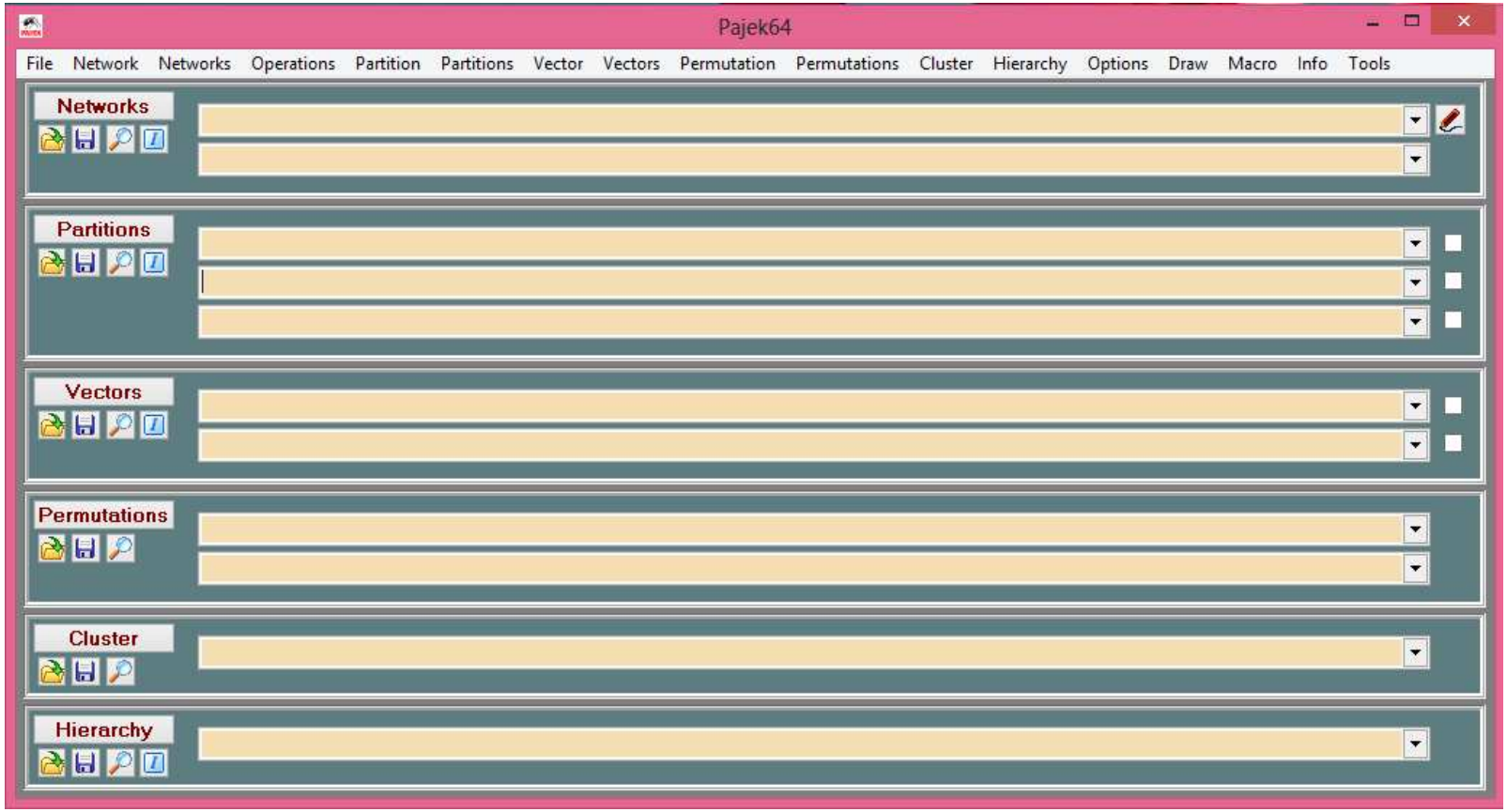
Pajek webpage

http://mrvar.fdv.uni-lj.si/pajek/

Pajek manual

http://mrvar.fdv.uni-lj.si/pajek/pajekman.pdf

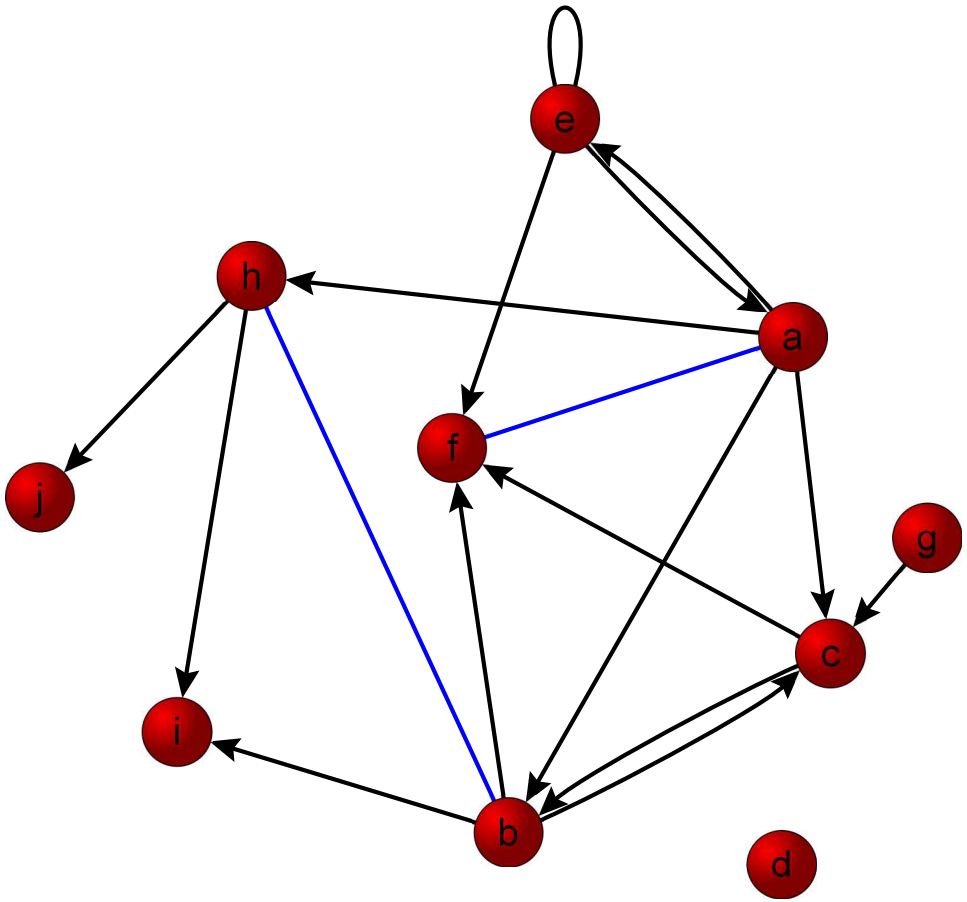
Pajek's Main Window



Six types of objects in Pajek:

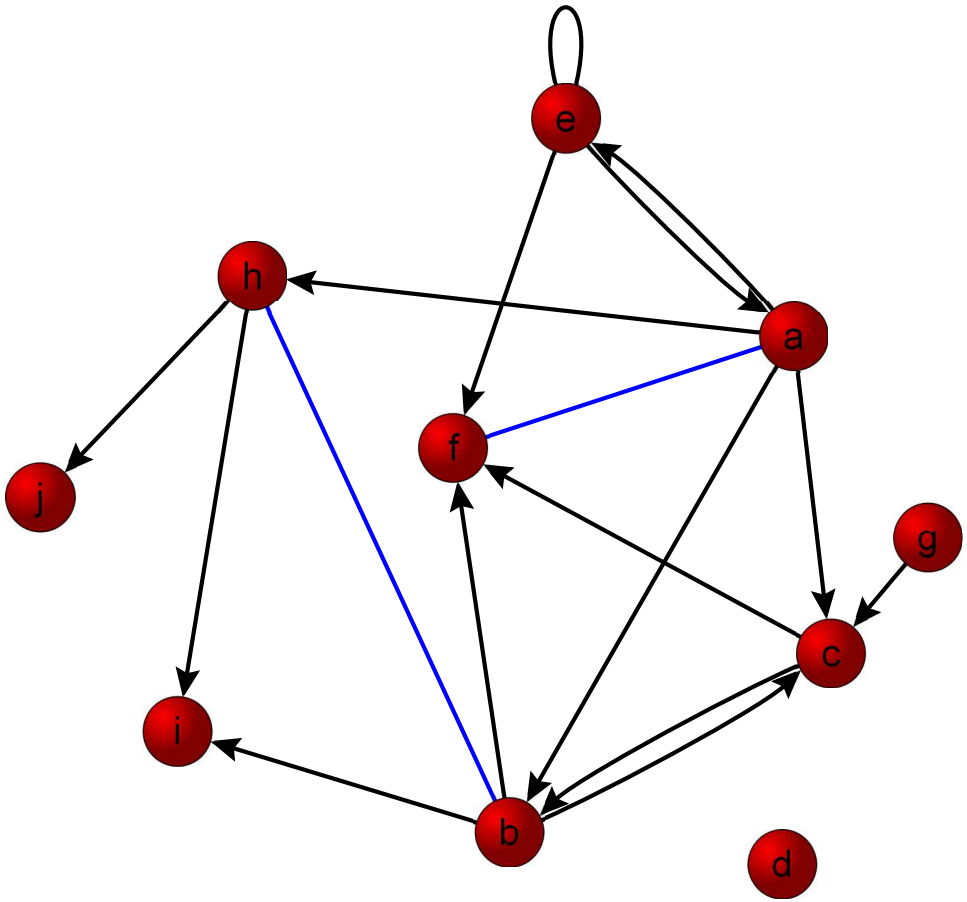
- Networks
- Partitions
- Vectors
- Permutations
- Clusters
- Hierarchies

Graph and set of links (and Pajek's *.net file)



$$\mathcal{V} = \{a, b, c, d, e, f, g, h, i, j\}$$
$$\mathcal{A} = \{(a, b), (a, c), (a, e), (a, h), (b, c), (b, f), (b, i), (c, b), (c, f), (e, a), (e, e), (e, f), (g, c), (h, i), (h, j)\}$$
$$\mathcal{E} = \{(a : f), (b : h)\}$$
$$\mathcal{G} = \{\mathcal{V}, \mathcal{A}, \mathcal{E}\}$$

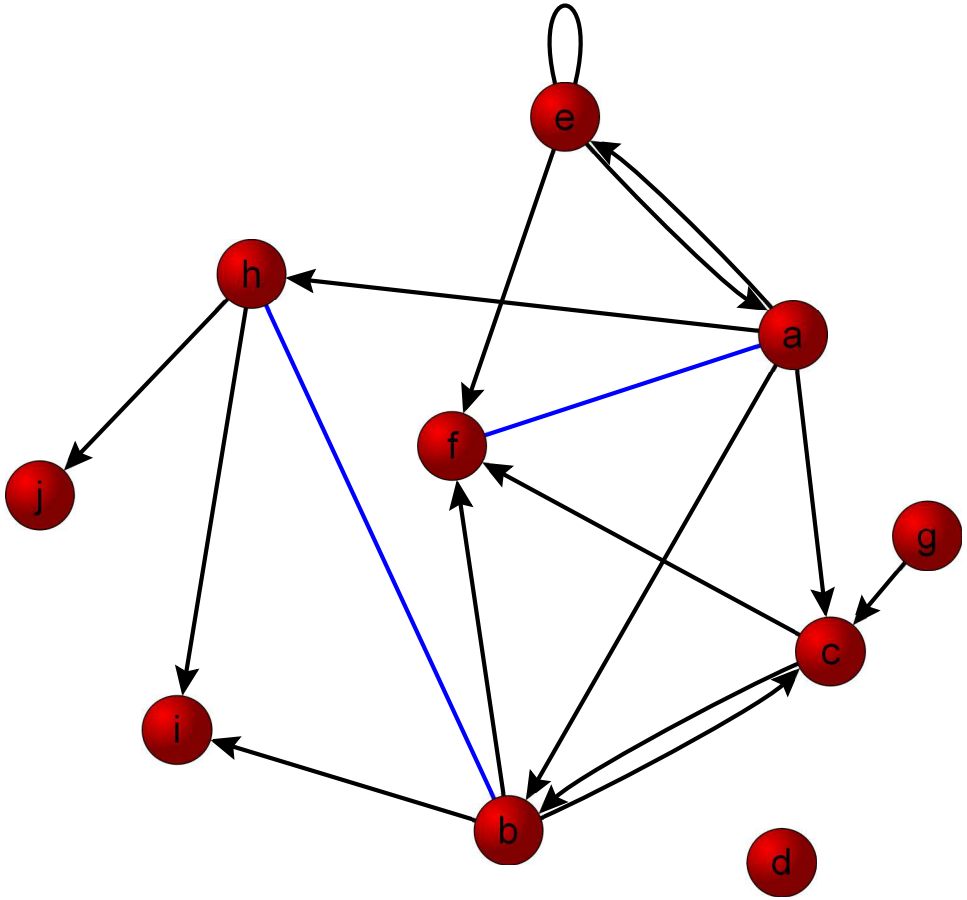
Graph and set of links (and Pajek's *.net file)



Description of data in Pajek - *.net file

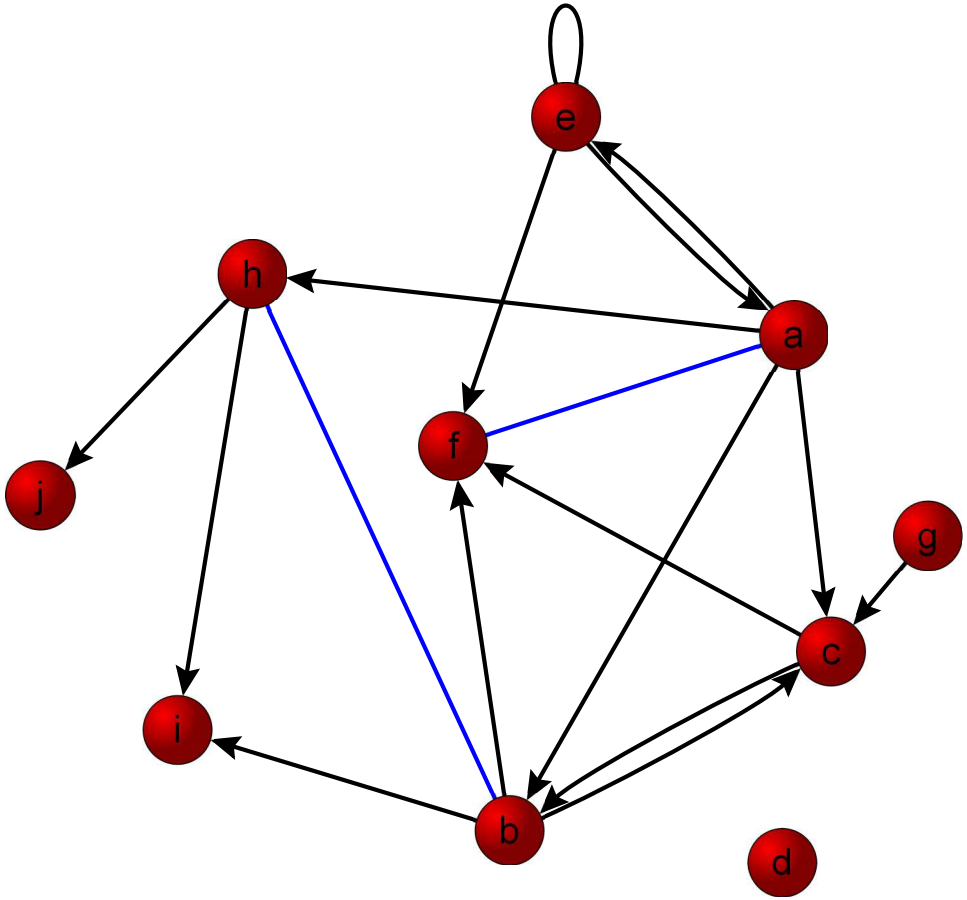
```
*Vertices 10
  1 "a"          0.6840     0.3426     0.5000
  2 "b"          0.5150     0.6169     0.5000
  3 "c"          0.7062     0.5178     0.5000
  4 "d"          0.6811     0.6696     0.5000
  5 "e"          0.5485     0.2219     0.5000
  6 "f"          0.4813     0.4042     0.5000
  7 "g"          0.7832     0.4414     0.5000
  8 "h"          0.3619     0.3091     0.5000
  9 "i"          0.3176     0.5611     0.5000
 10 "j"          0.2360     0.4314     0.5000
*Arcs
  1   2 1
  1   3 1
  1   5 1
  1   8 1
  2   3 1
  2   6 1
  2   9 1
  3   2 1
  3   6 1
  5   1 1
  5   5 1
  8  10 1
  7   3 1
  5   6 1
  8   9 1
*Edges
  1   6 1
  2   8 1
```

Graph and neighbours (and Pajek's *.net file)



$$N_A(a) = \{b, c, e, h\}$$
$$N_A(b) = \{c, f\}$$
$$N_A(c) = \{b, f\}$$
$$N_A(e) = \{a, e, f\}$$
$$N_A(g) = \{c\}$$
$$N_A(h) = \{i, j\}$$
$$N_E(a) = \{f\}$$
$$N_E(b) = \{h\}$$

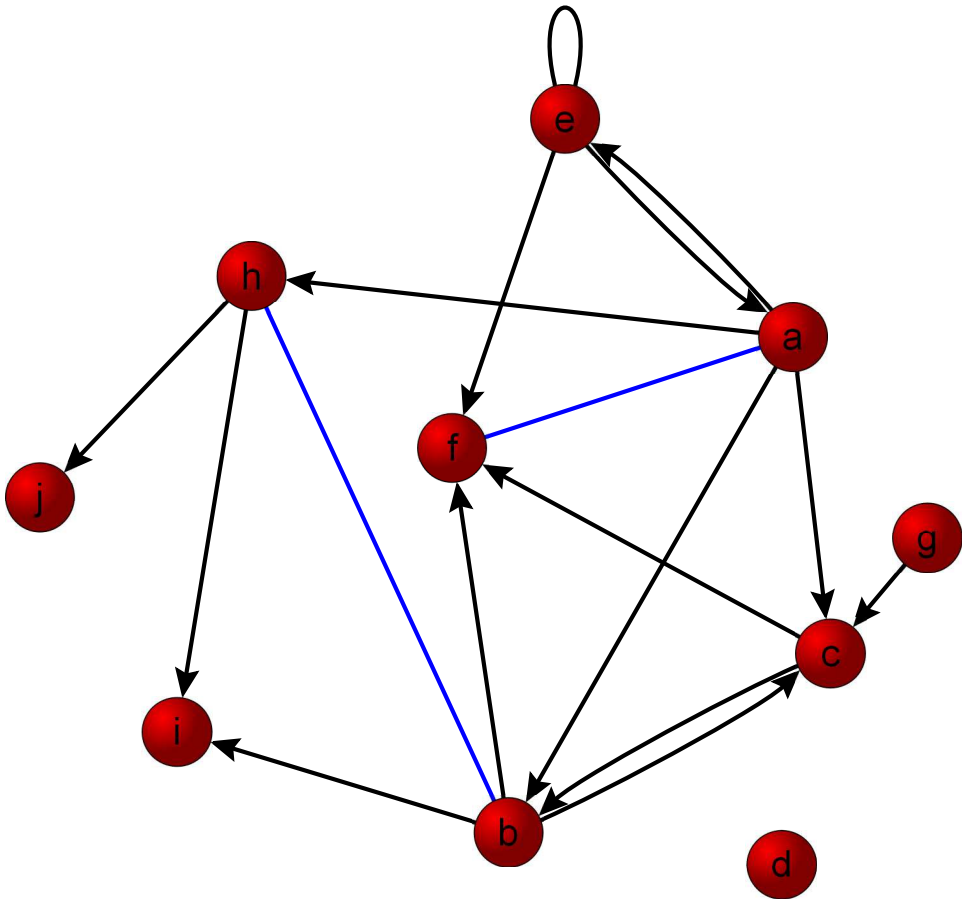
Graph and neighbours (and Pajek's *.net file)



Description of data in Pajek - *.net file

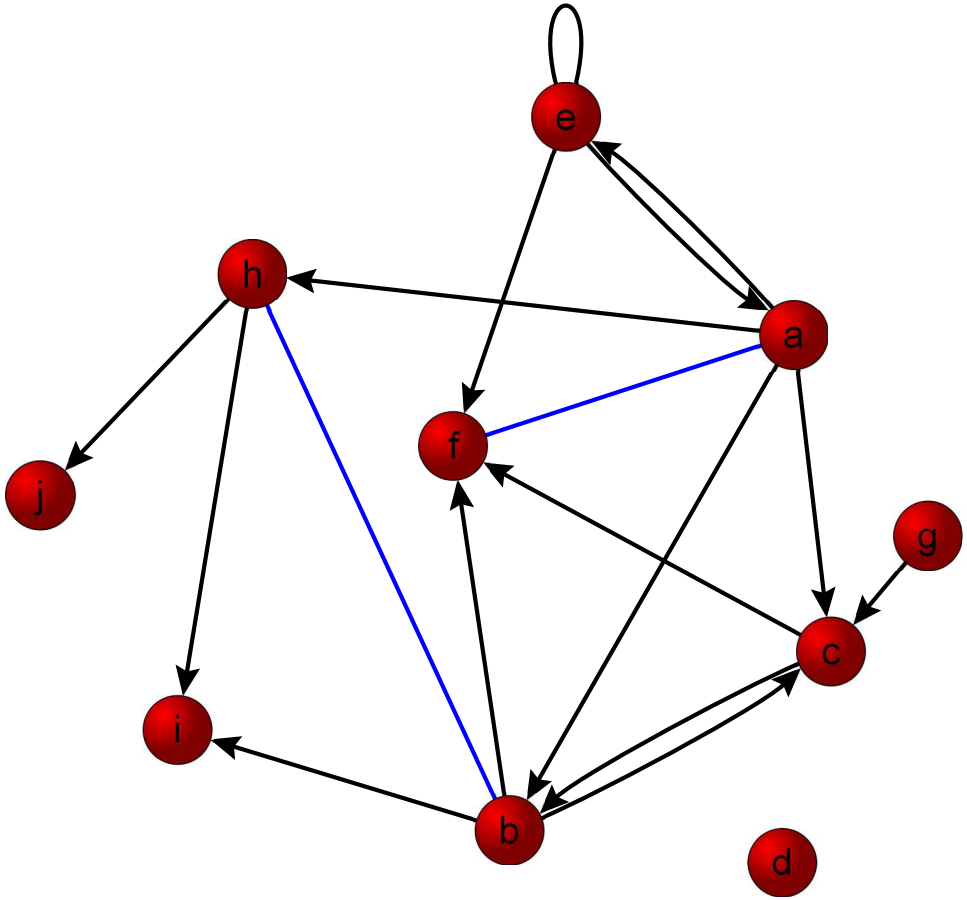
```
*Vertices 10
  1 "a"          0.6840     0.3426     0.5000
  2 "b"          0.5150     0.6169     0.5000
  3 "c"          0.7062     0.5178     0.5000
  4 "d"          0.6811     0.6696     0.5000
  5 "e"          0.5485     0.2219     0.5000
  6 "f"          0.4813     0.4042     0.5000
  7 "g"          0.7832     0.4414     0.5000
  8 "h"          0.3619     0.3091     0.5000
  9 "i"          0.3176     0.5611     0.5000
 10 "j"          0.2360     0.4314     0.5000
*Arcslist
 1  2  3  5  8
 2  3  6  9
 3  2  6
 5  1  5  6
 7  3
 8 10  9
*Edgeslist
 1  6
 2  8
```

Graph and matrix (and Pajek's *.mat file)



	a	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*	*j*
a	0	1	1	0	1	1	0	1	0	0
b	0	0	1	0	0	1	0	1	1	0
c	0	1	0	0	0	1	0	0	0	0
d	0	0	0	0	0	0	0	0	0	0
e	1	0	0	0	1	1	0	0	0	0
f	1	0	0	0	0	0	0	0	0	0
g	0	0	1	0	0	0	0	0	0	0
h	0	1	0	0	0	0	0	0	1	1
i	0	0	0	0	0	0	0	0	0	0
j	0	0	0	0	0	0	0	0	0	0

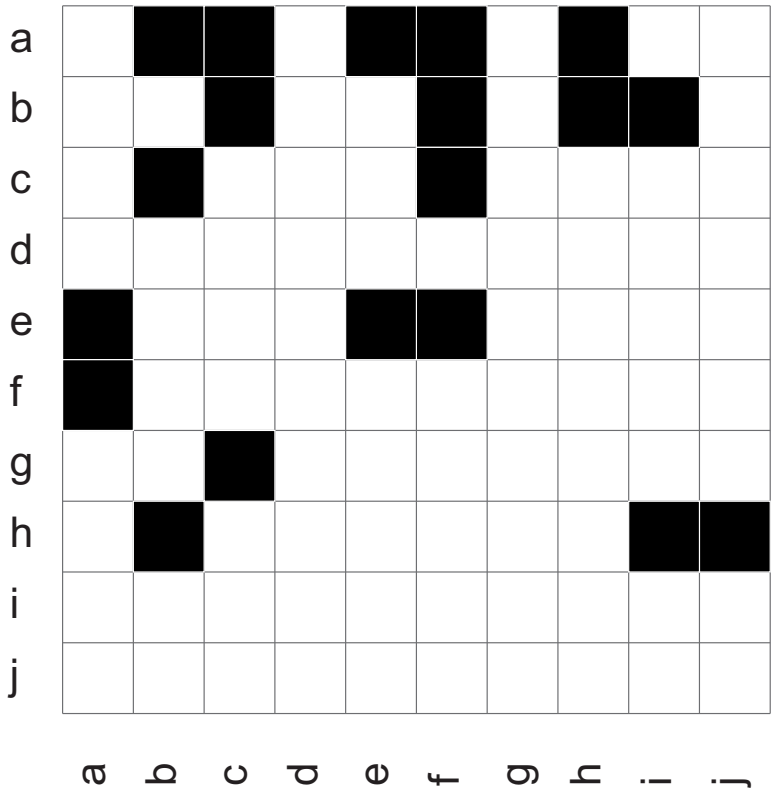
Graph and matrix (and Pajek's *.mat file)



Description of data in Pajek - *.mat file

```
*Vertices 10
  1 "a"          0.6840     0.3426     0.5000
  2 "b"          0.5150     0.6169     0.5000
  3 "c"          0.7062     0.5178     0.5000
  4 "d"          0.6811     0.6696     0.5000
  5 "e"          0.5485     0.2219     0.5000
  6 "f"          0.4813     0.4042     0.5000
  7 "g"          0.7832     0.4414     0.5000
  8 "h"          0.3619     0.3091     0.5000
  9 "i"          0.3176     0.5611     0.5000
 10 "j"          0.2360     0.4314     0.5000
*Matrix
0 1 1 0 1 1 0 1 0 0
0 0 1 0 0 1 0 1 1 0
0 1 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0
1 0 0 0 1 1 0 0 0 0
1 0 0 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0 1 1
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
```

Graph and matrix (and Pajek's *.mat file)



Description of data in Pajek - *.mat file

```
*Vertices 10
  1 "a"          0.6840     0.3426     0.5000
  2 "b"          0.5150     0.6169     0.5000
  3 "c"          0.7062     0.5178     0.5000
  4 "d"          0.6811     0.6696     0.5000
  5 "e"          0.5485     0.2219     0.5000
  6 "f"          0.4813     0.4042     0.5000
  7 "g"          0.7832     0.4414     0.5000
  8 "h"          0.3619     0.3091     0.5000
  9 "i"          0.3176     0.5611     0.5000
 10 "j"          0.2360     0.4314     0.5000
*Matrix
0 1 1 0 1 1 0 1 0 0
0 0 1 0 0 1 0 1 1 0
0 1 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0
1 0 0 0 1 1 0 0 0 0
1 0 0 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0 1 1
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
```

Vertex properties (*.clu, *.vec, *.per files in Pajek)

CLUstering or partition of vertices contains **nominal or ordinal data** about vertices.

VECtor or partition of vertices contains **numeric** data about vertices.

PERmutation or ordering of vertices.

Structure of *.clu, *.vec, and *.per files in Pajek

```
*Vertices n
   v1
   v2
   v3
   .
   .
   .
   vn
```

n is the number of vertices in the network

vertex 3 has value v3

Vertex propertiies (*.clu, *.ver, *.per files in Pajek)

example10.net

```
*Vertices 10
  1 "a"
  2 "b"
  3 "c"
  4 "d"
  5 "e"
  6 "f"
  7 "g"
  8 "h"
  9 "i"
 10 "j"
*Arcs
  1   2  2
  1   3  1
  1   5  3
```

example10.net

```
  1   8  1
  2   3  3
  2   6  4
  2   9  1
  3   2  3
  3   6  1
  5   1  2
  5   5  1
  8  10  1
  7   3  5
  5   6  1
  8   9  1
*Edges
  1   6  1
  2   8  3
```

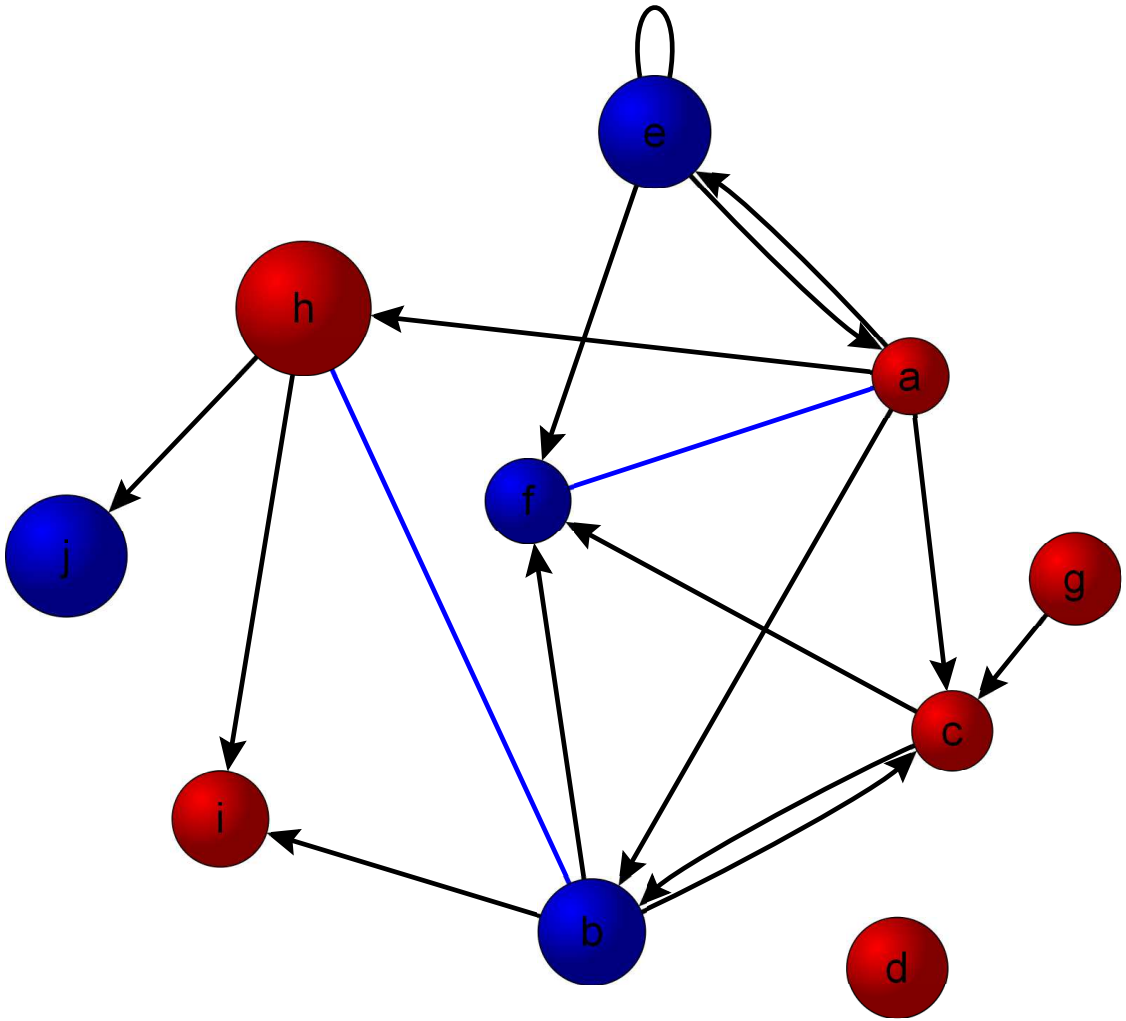
gender.clu

```
*Vertices 10
1
2
1
1
2
2
1
1
1
2
```

age.vec

```
*Vertices 10
1
2
1
1
2
2
1
1
1
2
```

Example of network on 10 vertex properties together with cluster and vector data



Types of networks

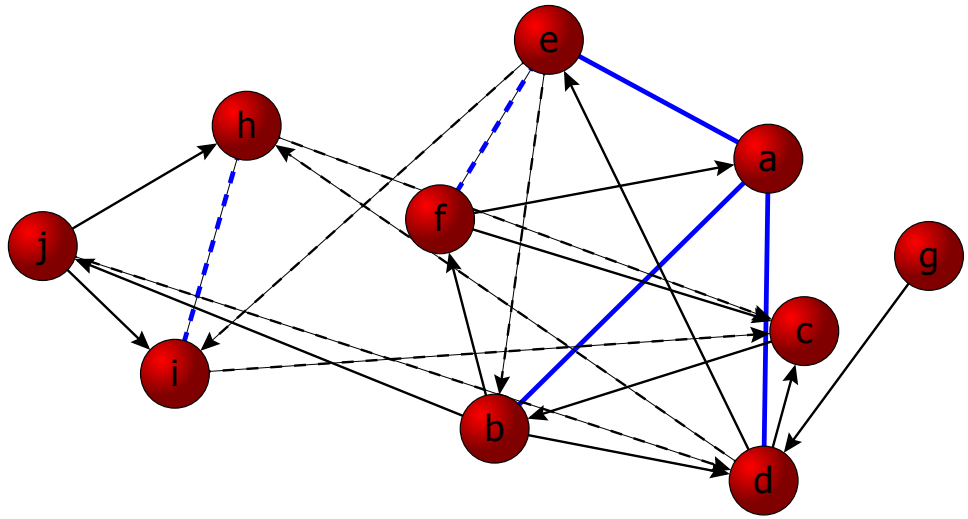
Beside distinguishing between (binary) directed and undirected networks also other types of networks are known:

- signed networks,
- valued networks,
- 2-mode networks,
- multi-relational networks,
- temporal networks,...

Description of signed network in Pajek

Description of signed network in Pajek

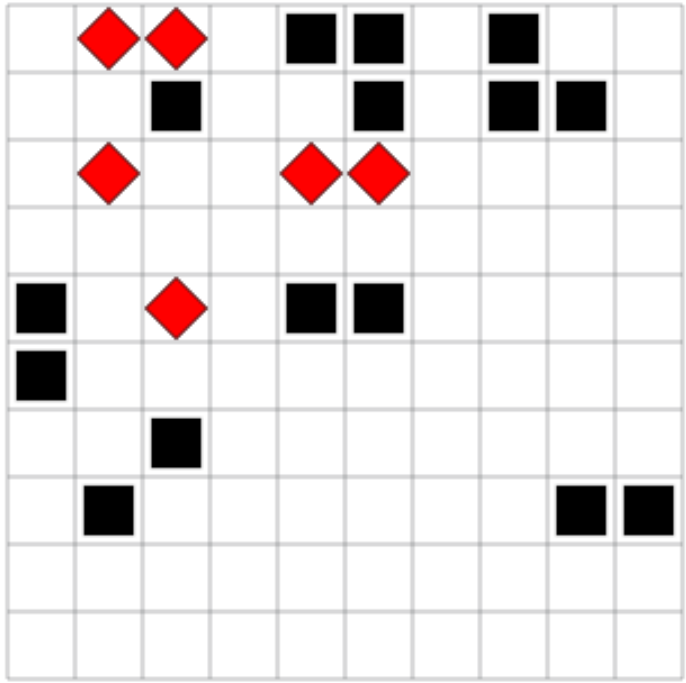
```
*Vertices 10
  1 "a"              0.6426    0.5000
  2 "b"              0.3169    0.5000
  3 "c"              0.1178    0.5000
  4 "d"              0.8696    0.5000
  5 "e"              0.4219    0.5000
  6 "f"              0.2042    0.5000
  7 "g"              0.7414    0.5000
  8 "h"              0.4091    0.5000
  9 "i"              0.8611    0.5000
 10 "j"              0.3314    0.5000
*Arcs
  1   2 1
  1   4 1
  1   5 1
  2   1 1
  2   4 1
  2   6 1
  2  10 1
  3   2 1
  4   1 1
  4   3 1
  4   5 1
  4   8 -1
  5   1 1
  5   2 -1
  5   6 -1
  5   9 -1
  6   1 1
  6   3 1
  6   5 1
  7   4 1
  8   3 -1
  8   9 1
  9   3 -1
  9   8 -1
 10   4 -1
 10   8 1
 10   9 1
```



In signed network we have positive ties (1s; drawn with solid lines) and negative ties (-1s; drawn with dashed lines).

Description of signed network in Pajek

An example of exported EPS matrix of signed network



Signed networks can be exported also as EPS matrix. Positive ties are represented with black boxes, negative ties with red diamonds and null blocks as empty squares.

If we have also weights on ties, different levels of gray are used for positive ties and different levels of red for negative ties.

Commands in Pajek: Export Matrix As EPS

Set the desired options first:

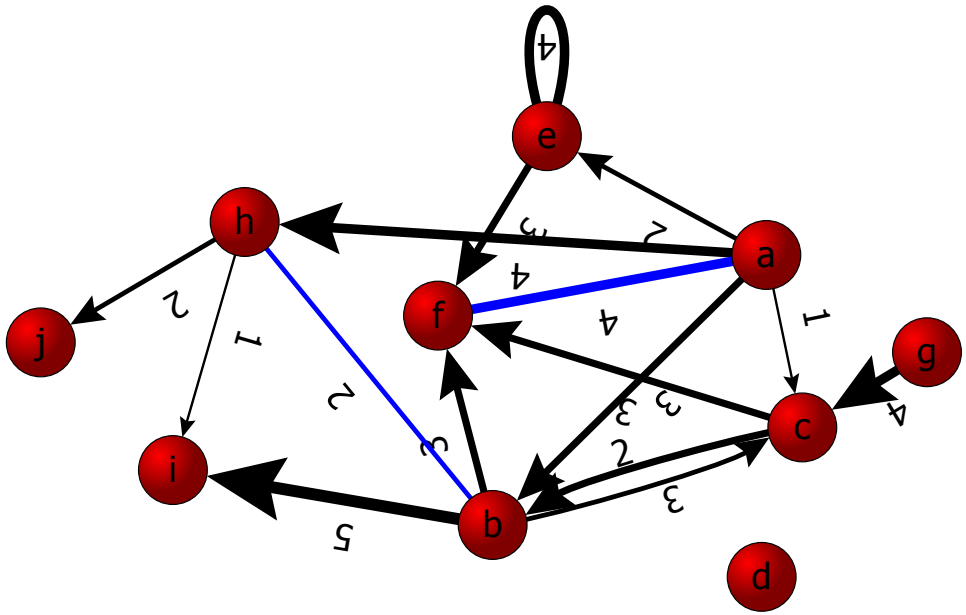
`File/Network/Export as Matix to EPS/Options`

`File/Network/Export as Matix to EPS/Using Permutation + Partition`

Description of weighted (or valued) network in Pajek

Description of weighted network in Pajek

```
*Vertices 10
  1 "a"          0.6840     0.3426     0.5000
  2 "b"          0.5150     0.6169     0.5000
  3 "c"          0.7062     0.5178     0.5000
  4 "d"          0.6811     0.6696     0.5000
  5 "e"          0.5485     0.2219     0.5000
  6 "f"          0.4813     0.4042     0.5000
  7 "g"          0.7832     0.4414     0.5000
  8 "h"          0.3619     0.3091     0.5000
  9 "i"          0.3176     0.5611     0.5000
 10 "j"          0.2360     0.4314     0.5000
*Arcs
  1   2 3
  1   3 1
  1   5 2
  1   8 4
  2   3 2
  2   6 3
  2   9 5
  3   2 3
  3   6 3
  5   5 4
  8  10 2
  7   3 4
  5   6 3
  8   9 1
*Edges
  1   6 4
  2   8 2
```

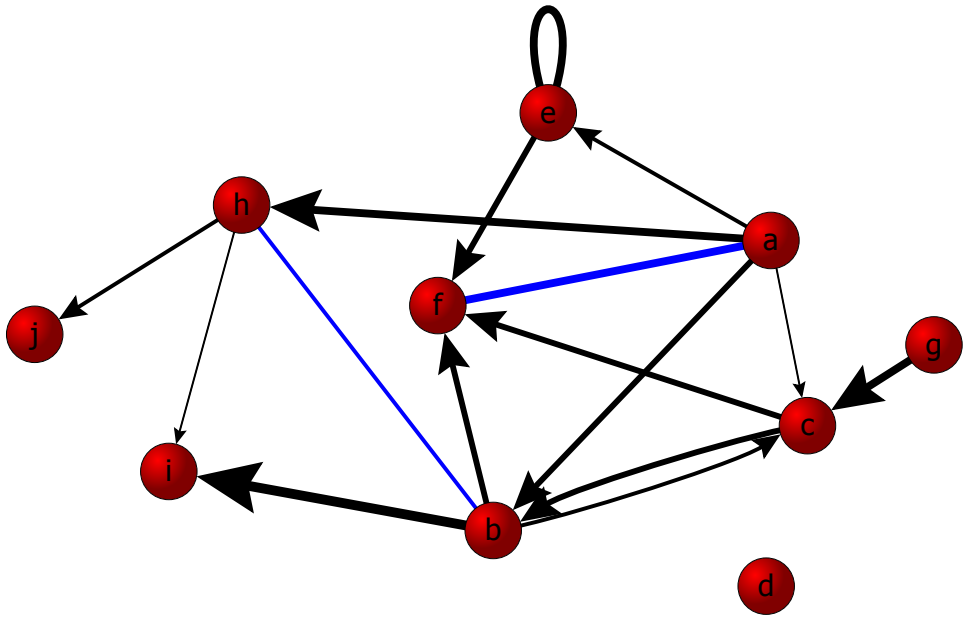


E.g. Arc from vertex *a* and *b* has weight 3.

Arc from vertex *a* to vertex *c* has weight 1.

Description of valued (or weighted) network in Pajek

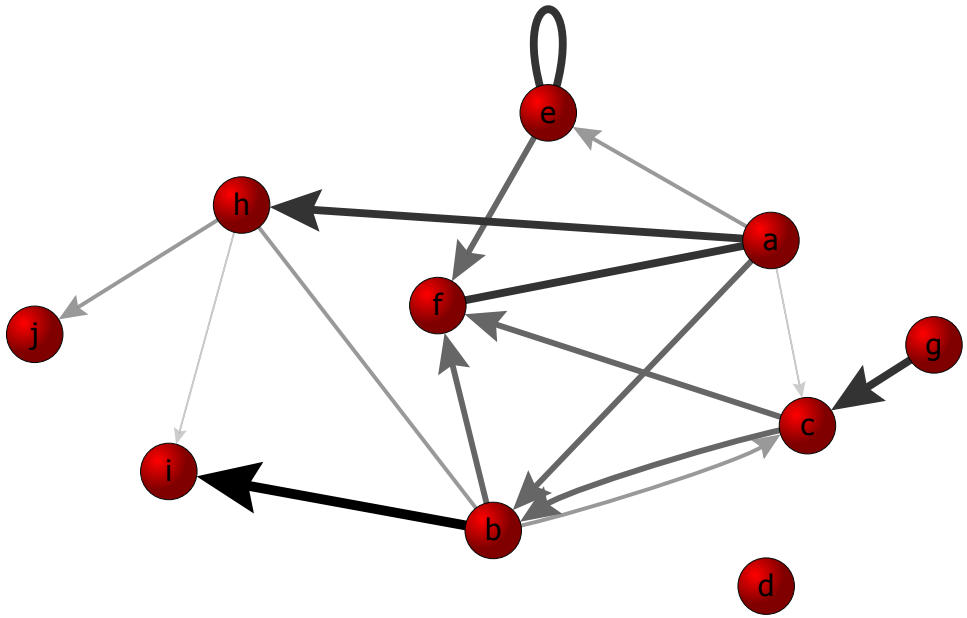
Example of a valued network in Pajek



Lines could be in Pajek represented with different widths.

Description of valued (or weighted) network in Pajek

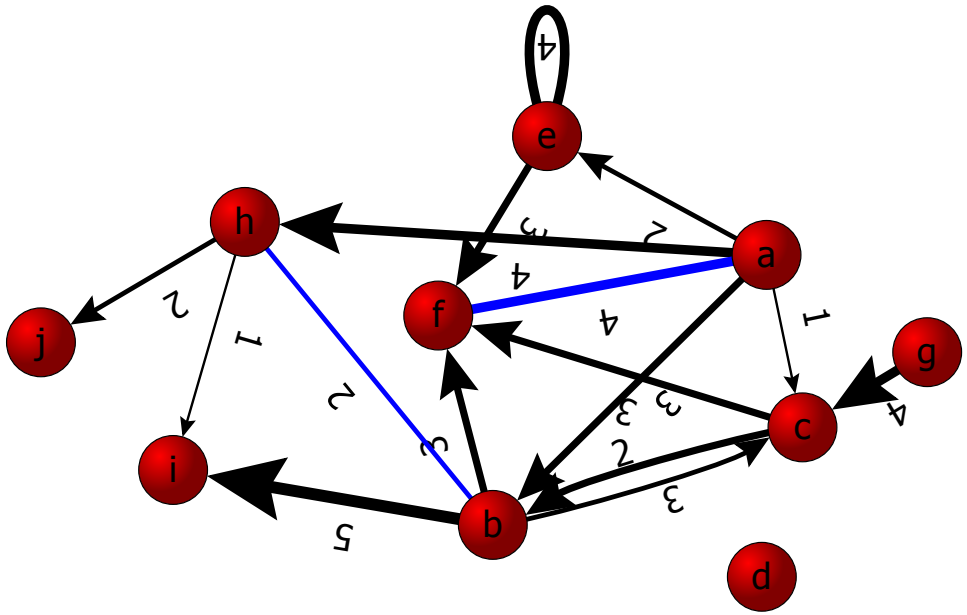
Example of a valued network in Pajek



Lines could be in Pajek represented with different levels of gray (and widths).

Description of valued (or weighted) network in Pajek

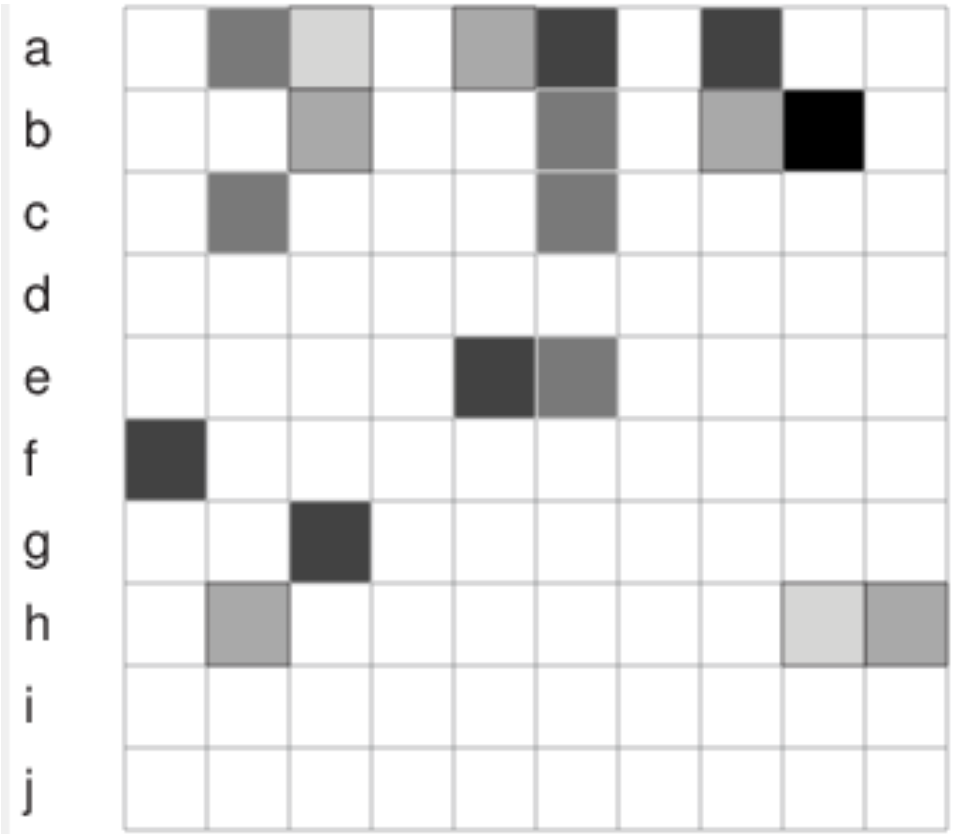
Example of a valued network in Pajek



Lines could be in Pajek represented with numbers (and different widths).

Description of valued (or weighted) network in Pajek

Example of a valued network in Pajek

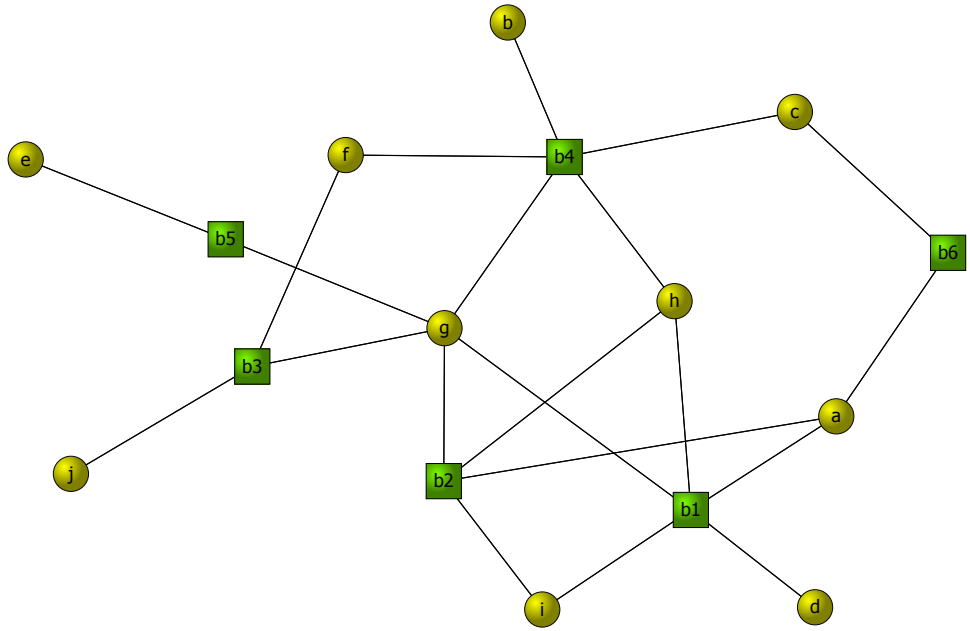


Exported matrix of a network where different levels of gray squares represent different line weights.

Description of 2-mode network in Pajek

Description of 2-mode network in Pajek

```
*Vertices 16 10
  1 "a"             0.8409    0.6522    0.5000 ellipse
  2 "b"             0.5204    0.0548    0.5000 ellipse
  3 "c"             0.8005    0.1905    0.5000 ellipse
  4 "d"             0.8202    0.9418    0.5000 ellipse
  5 "e"             0.0500    0.2625    0.5000 ellipse
  6 "f"             0.3622    0.2559    0.5000 ellipse
  7 "g"             0.4586    0.5185    0.5000 ellipse
  8 "h"             0.6831    0.4775    0.5000 ellipse
  9 "i"             0.5540    0.9452    0.5000 ellipse
 10 "j"             0.0941    0.7394    0.5000 ellipse
 11 "b1"            0.6990    0.7942    0.5000 box
 12 "b2"            0.4579    0.7505    0.5000 box
 13 "b3"            0.2711    0.5766    0.5000 box
 14 "b4"            0.5758    0.2588    0.5000 box
 15 "b5"            0.2451    0.3837    0.5000 box
 16 "b6"            0.9500    0.4041    0.5000 box
*Edgeslist
 1 11 12 16
 2 14
 3 14 16
 4 11
 5 15
 6 13 14
 7 11 12 13 14 15
 8 11 12 14
 9 11 12
10 13
```

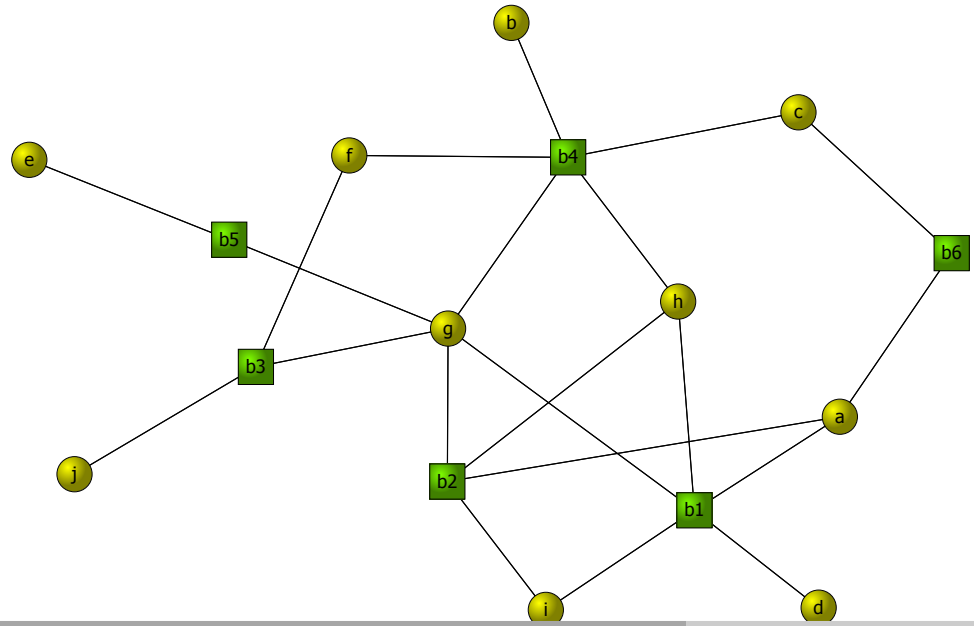


In 2-mode networks we have two distinguished set of vertices and links between this two sets of vertices.

2-mode networks

Commands in Pajek: Create a 2-mode partition

Create Partition into 2-mode:
`Network/2-Mode Network/Partition into 2 Modes`
Draw network: `Draw/Network+First Partition`
In Figure Window
`Layers/In y direction + random in x`



2-mode networks

Commands in Pajek: Create a 2-mode partition

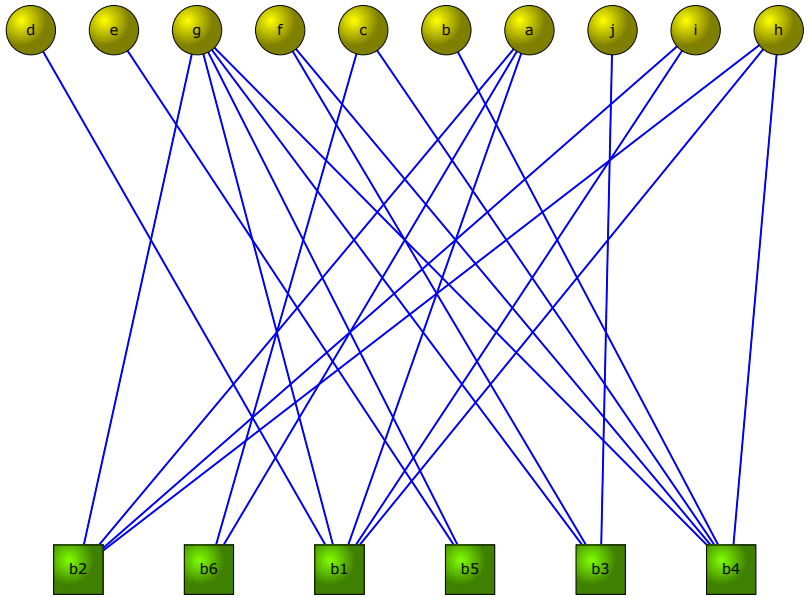
Create Partition into 2-mode:

`Network/2-Mode Network/Partition into 2 Modes`

Draw network: `Draw/Network+First Partition`

In Figure Window

`Layers/In y direction + random in x`



2-mode networks

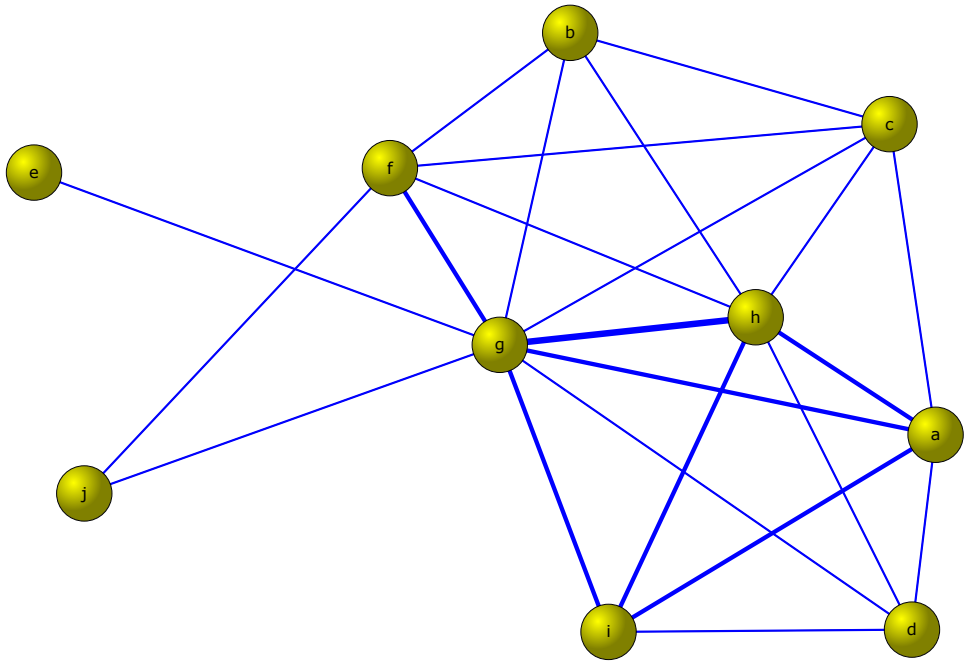
Commands in Pajek: Transform 2-mode network to 1-mode

Create Partition into 2-mode:

`Network/2-Mode Network/2-mode to 1-mode/Rows`

or

`Network/2-Mode Network/2-mode to 1-mode/Columns`



2-mode networks

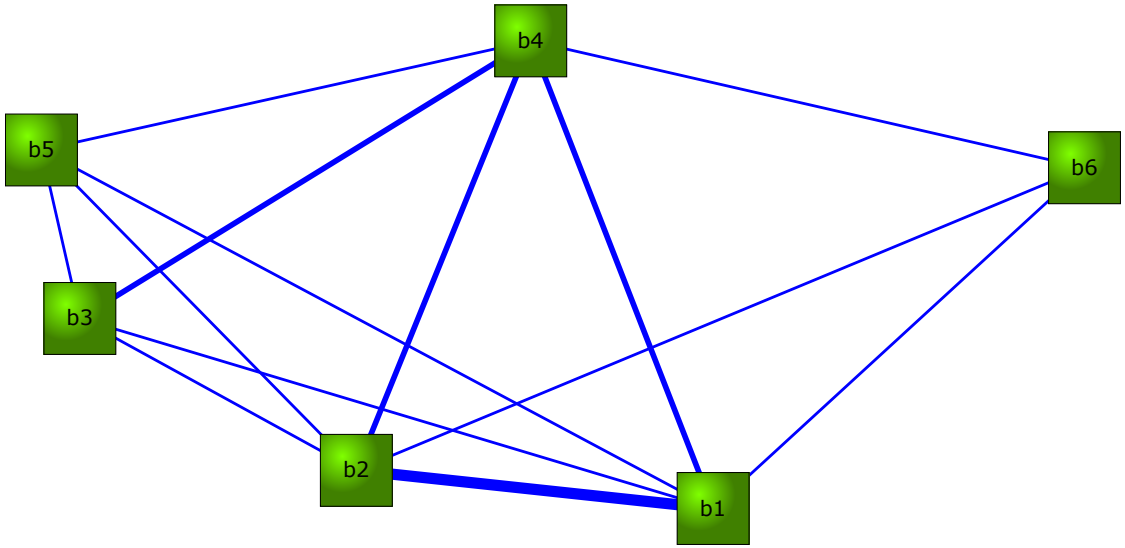
Commands in Pajek: Transform 2-mode network to 1-mode

Create Partition into 2-mode:

`Network/2-Mode Network/2-mode to 1-mode/Rows`

or

`Network/2-Mode Network/2-mode to 1-mode/Columns`



Temporal networks

In temporal networks we have to specify time points or intervals when each vertex and line is present (or active) in the network.

Parameters have one of the following forms:

- $[t_i]$ - in time point t_i,
- $[t_i - t_j]$ - from time point t_i to t_j,
- $[t_i - *]$ - from time point t_i on.

Intervals have to be separated by commas.

The lines and vertices in a temporal network should satisfy the consistency condition: if a line is active in time t then also its end vertices have to be active in time t.

Time records should always be written as last in the row where vertices or lines are defined.

Description of temporal network in Pajek

Description of temporal network in Pajek

```
*Vertices 10
  1 "a"    0.6840    0.3426    0.5000    [1-5]
  2 "b"    0.5150    0.6169    0.5000    [2-4]
  3 "c"    0.7062    0.5178    0.5000    [3-*]
  4 "d"    0.6811    0.6696    0.5000    [1-3]
  5 "e"    0.5485    0.2219    0.5000    [1-5]
  6 "f"    0.4813    0.4042    0.5000    [2-4]
  7 "g"    0.7832    0.4414    0.5000    [2-5]
  8 "h"    0.3619    0.3091    0.5000    [1,3,5]
  9 "i"    0.3176    0.5611    0.5000    [1-5]
 10 "j"    0.2360    0.4314    0.5000    [2,5]
*Arcs
  1   2 1 [1-4]
  1   3 1 [3-4]
  1   5 1 [1-5]
  1   8 1 [3-5]
  2   3 1 [3-4]
  2   6 1 [1,2,4]
  2   9 1 [4-5]
  3   2 1 [3-5]
  3   6 1 [3,4]
  5   1 1 [1-2,3-4]
  5   5 1 [3-*]
  8  10 1 [5]
  7   3 1 [3-5]
  5   6 1 [2-4]
  8   9 1 [1,3,5]
*Edges
  1   6 1 [2-4]
  2   8 1 [3,5]
```

E.g:
Vertex *a* is present in time points 1, 2, 3, 4, and 5.

Vertex *c* is present in time points 3, 4, and 5 (* means infinity or "to the last point").

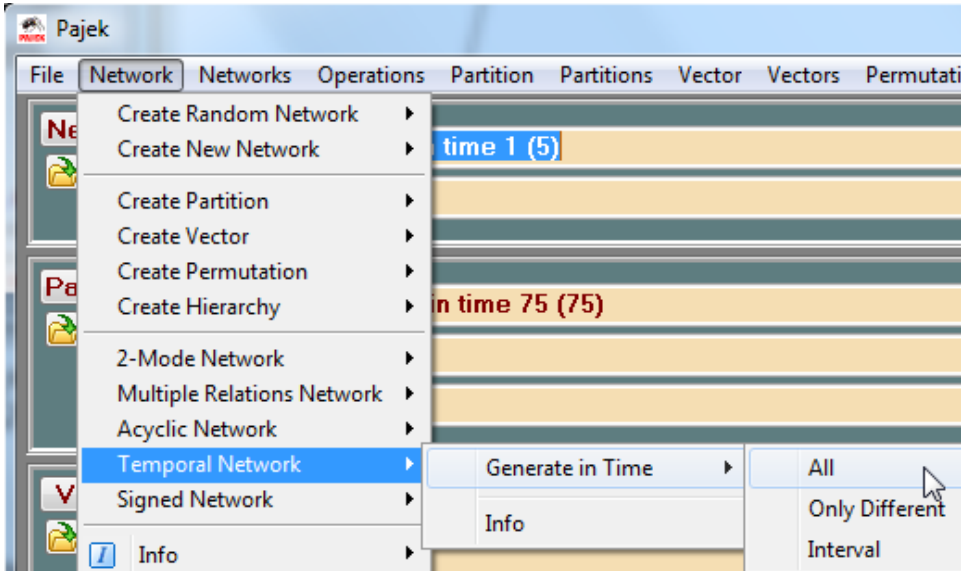
Edge from vertex *a* to vertex *f*, $[1:6]$, is present at time points from 2 to 4.

Example of a temporal network

Network in time t is called time slice.

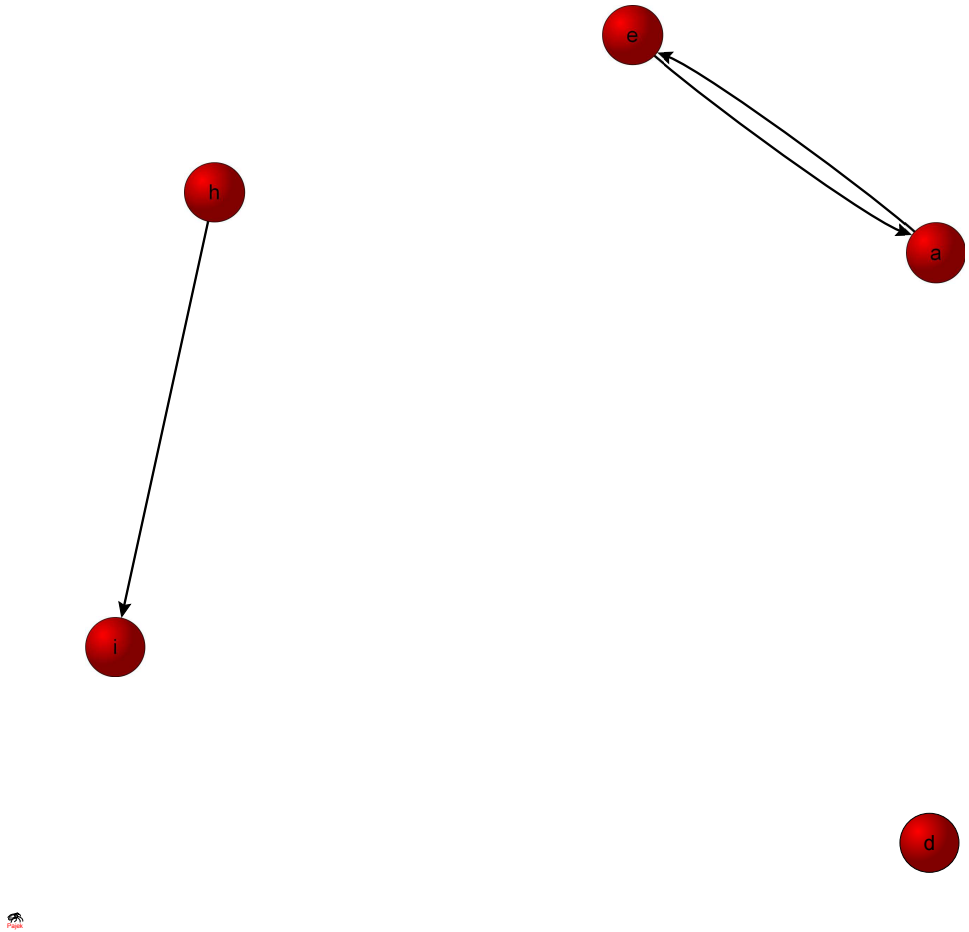
Commands in Pajek: Time slices in Pajek

`Network/Temporal Network/Generate in time`



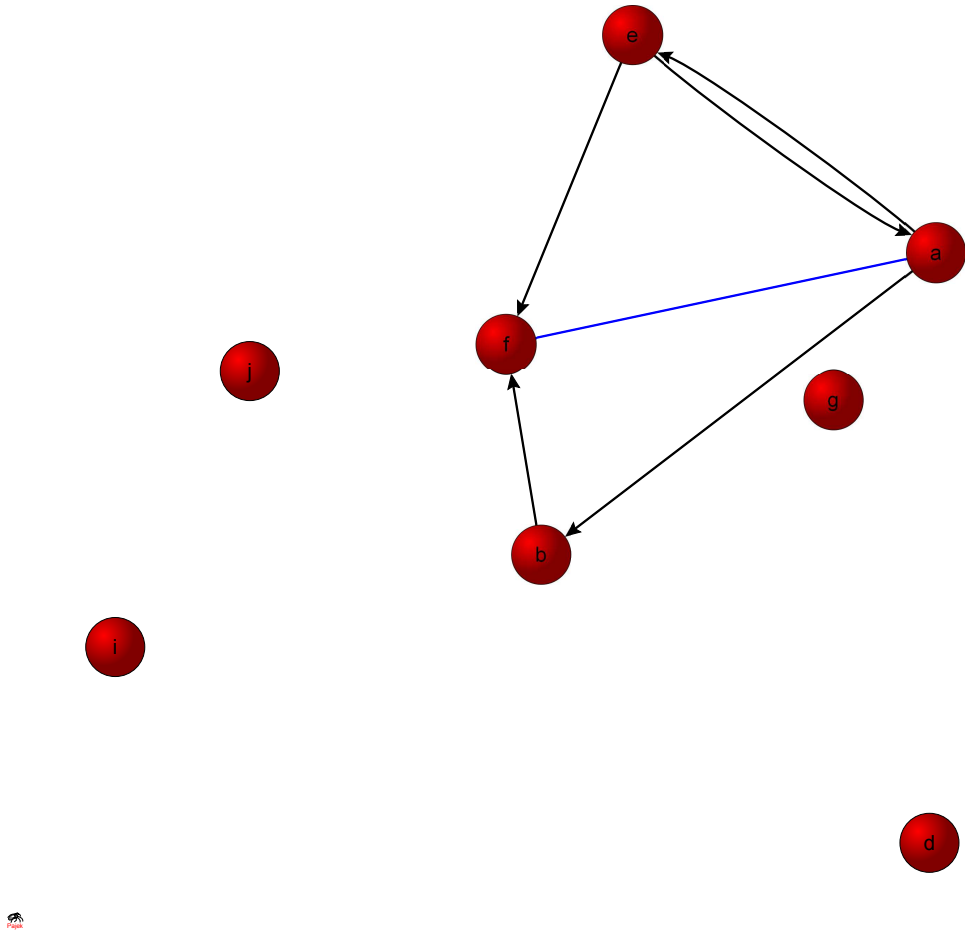
Example of a temporal network

Time slice T1



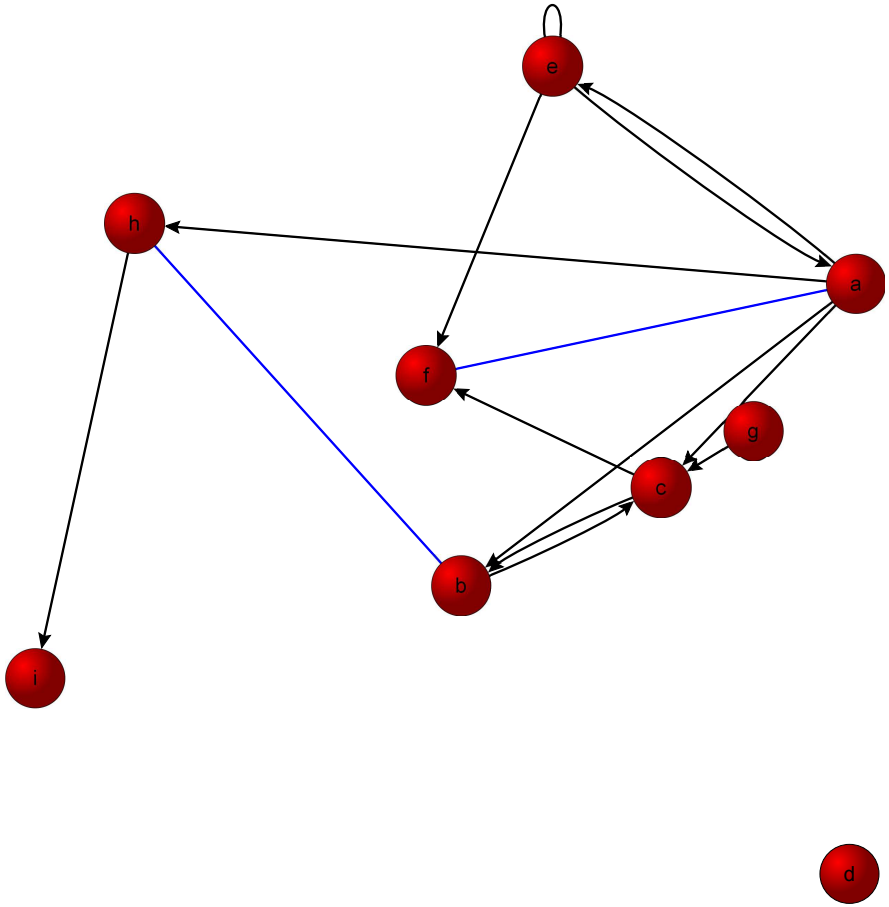
Example of a temporal network

Time slice T2



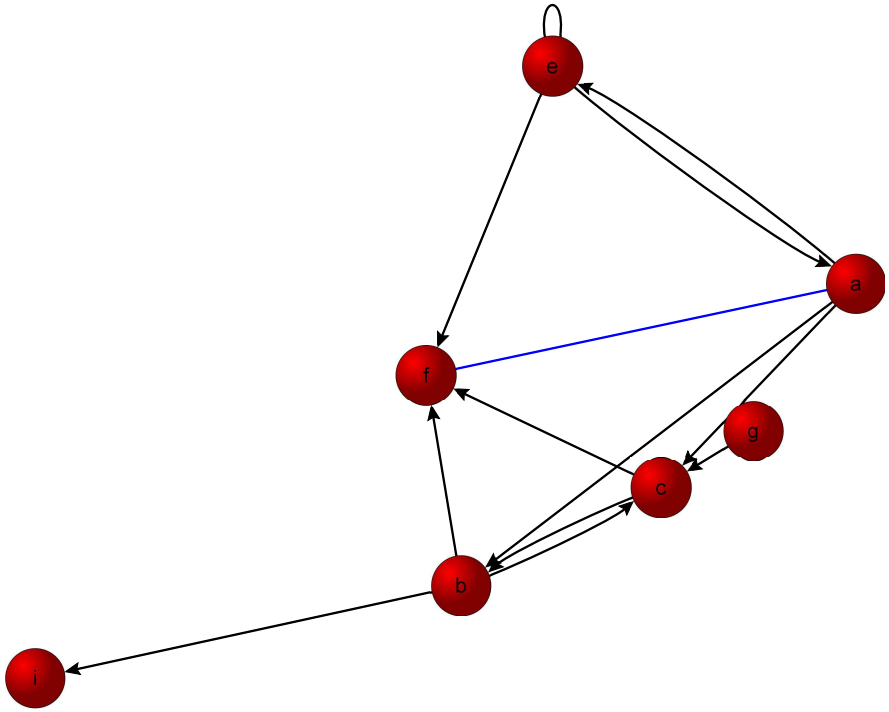
Example of a temporal network

Time slice T3



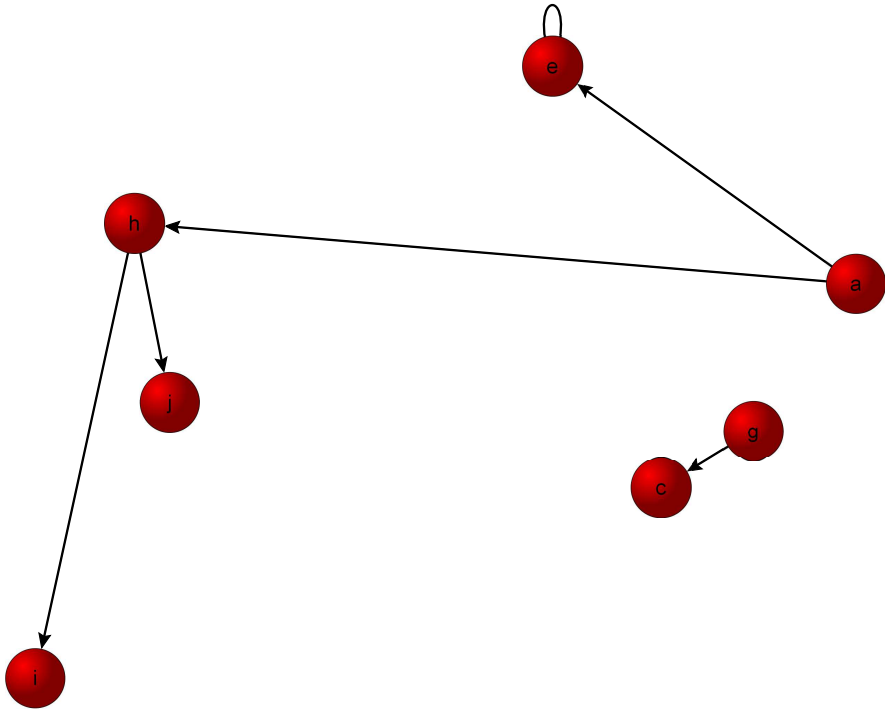
Example of a temporal network

Time slice T4



Example of a temporal network

Time slice T5



Multi-relational networks

In multi-relational networks we have several different relations.

Relations can be specified as follows:

- To a keyword for description of lines (`*arcs`, `*edges`, `*arcslist`, `*edgeslist`, `*matrix`) the number of relation and its name has to be added.
  `*arcs :2 ''be a friend''`
  All lines written below that command belong to the specified relation.
- A number of the corresponding relation can be written at the beginning of the line description (after the `*arcs` or `*edges`).
  `2: 3 8 1`
  means that line from vertex 3 to vertex 8 with weight 1 belongs to relation number 2.

Description of multi-relational network in Pajek

Description of multi-relational network in Pajek

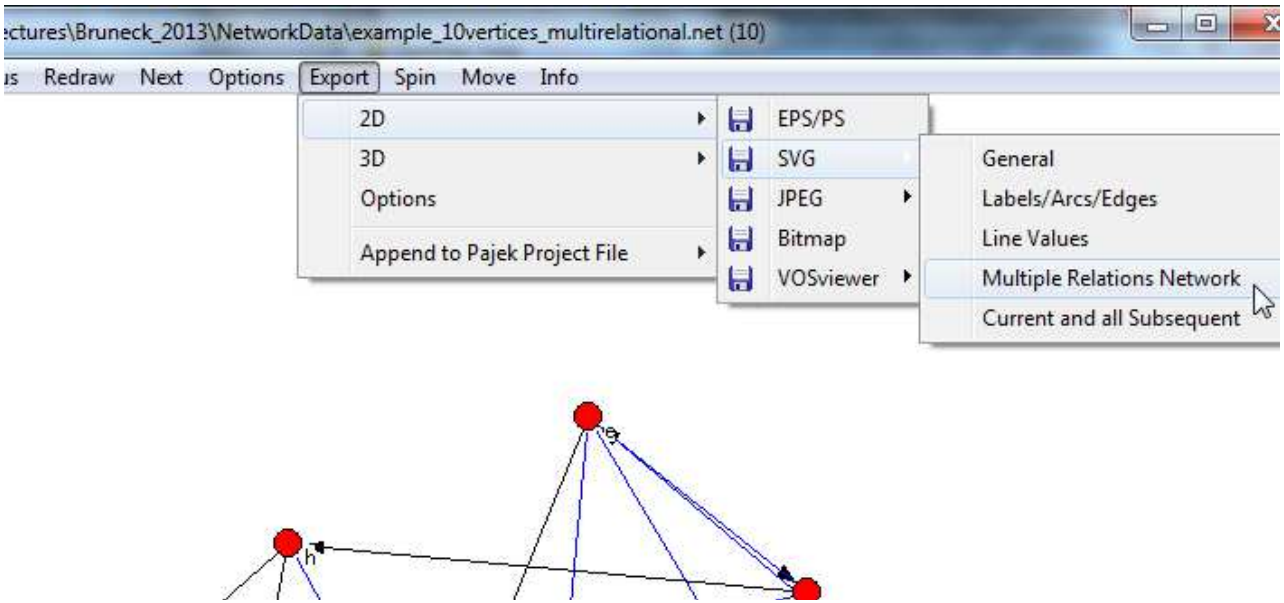
```
*Arcs :1 "be a friend"
  1    2   1
  1    3   1
  1    5   1
  1    8   1
  2    3   1
  2    6   1
  2    9   1
  3    2   1
  3    6   1
  5    1   1
  8   10   1
  7    3   1
  5    6   1
  8    9   1
*Edges  :1 "be a friend"
  1    6   1
  2    8   1
*Edgeslist :2 "be a classmate"
  1    2   3   5   6
  2    3   5   6
  3    5   6
  5    6
```

E.g:
Relation 'be a friend' is defined with the set of arcs and edges, while relation 'be a friend' is defined with the list of edges.

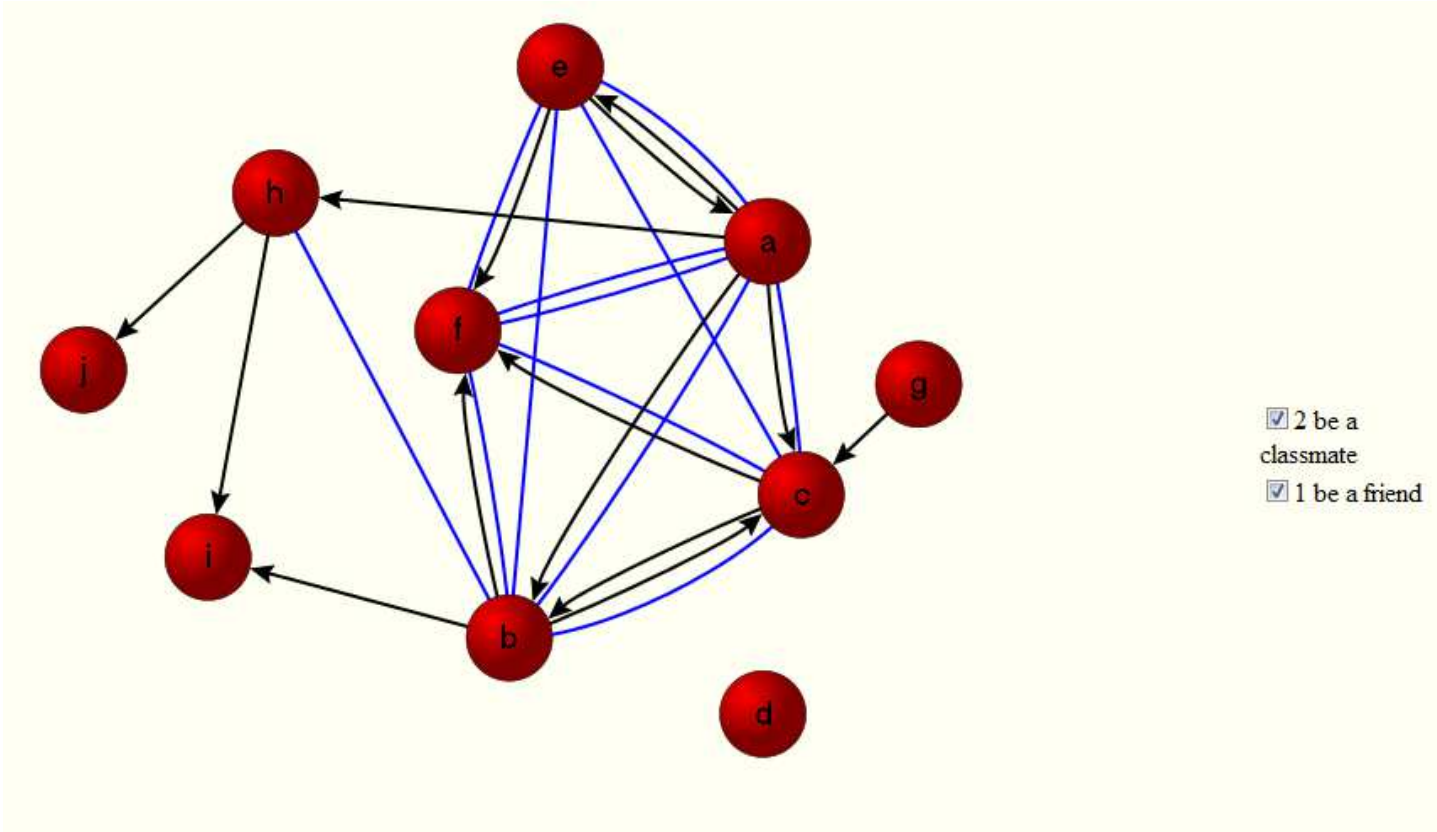
Example of a multi-relational network

Command: Export figure of multi-relational network

`Export/SVG/Multiple Relations Network`



Example of a multi-relational network

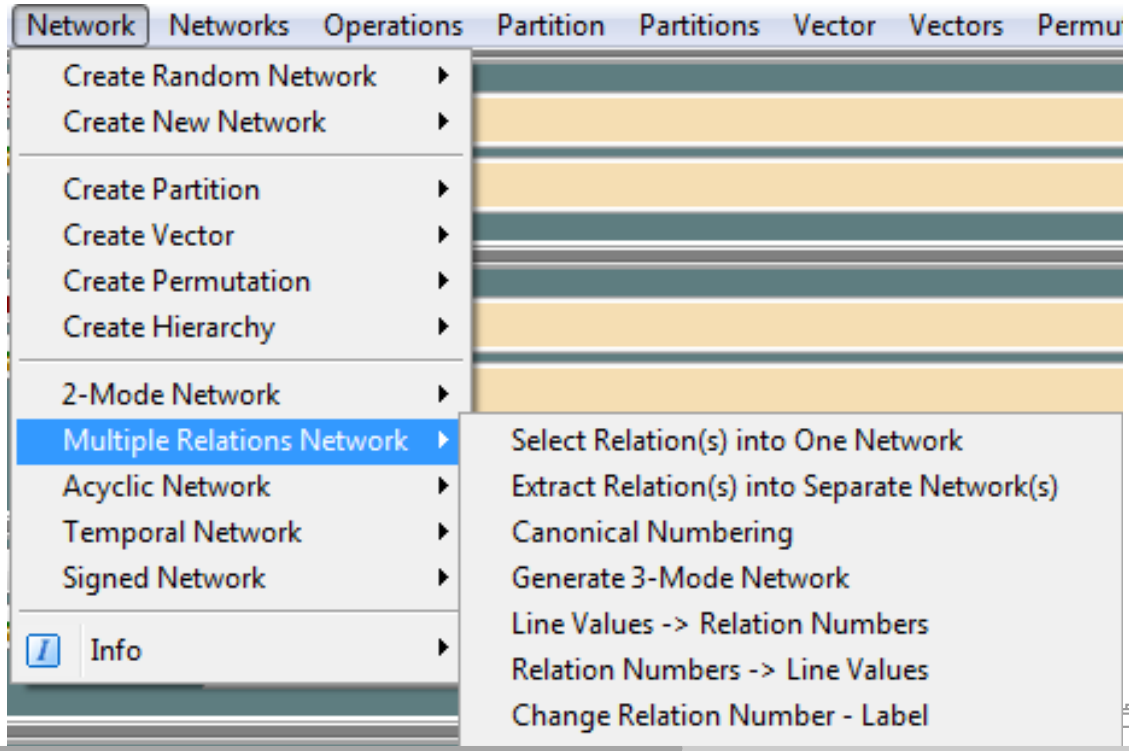


Open the exported figure in the internet browser: fig_example_10vertices_multirelational.htm

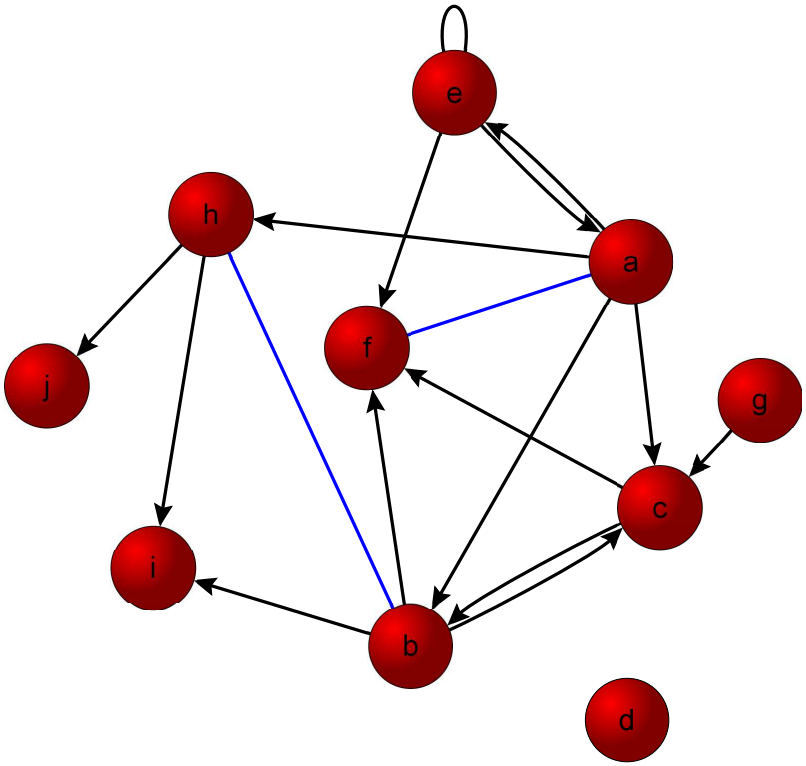
Multi-relational networks

Commands in Pajek: Manipulations with multi-relational networks

Network/Multiple Relations Network



Density of a network



Network statistics:

$n = 10,\ |\mathcal{A}| = 15,\ |\mathcal{E}| = 2$

$m = |\mathcal{A}| + 2 \cdot |\mathcal{E}| = 19$

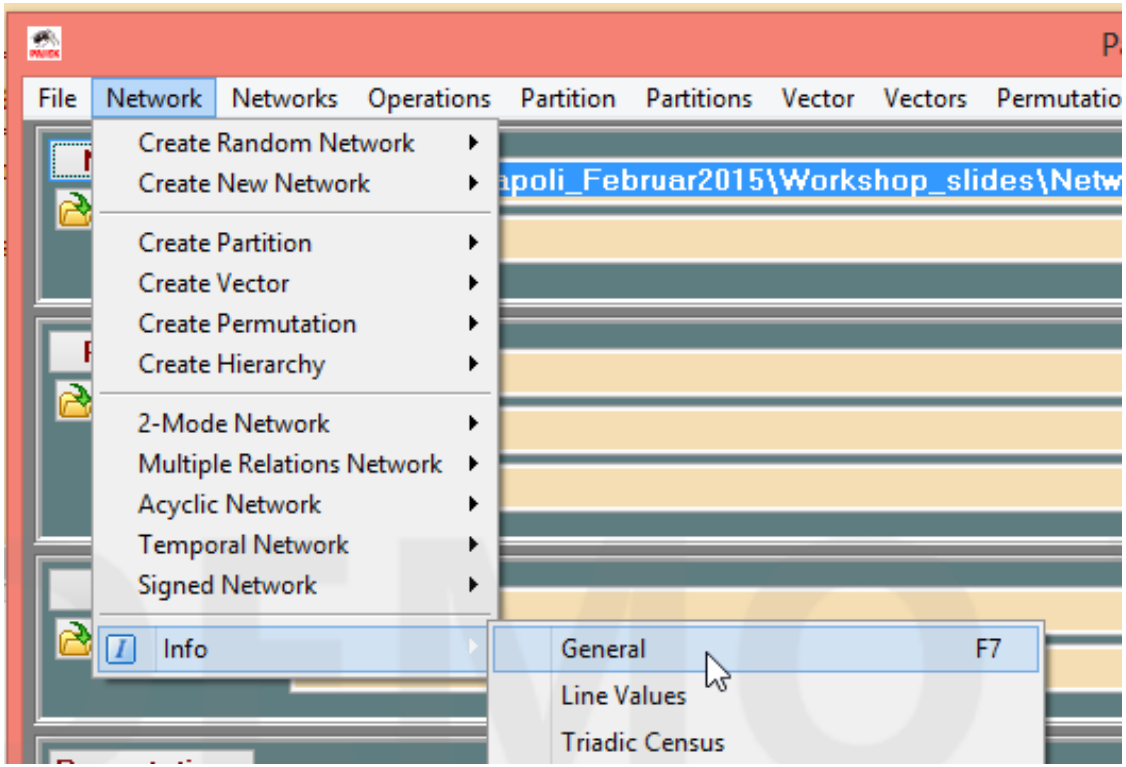
Density of a network:

$$\Delta = \frac{m}{n^2} = \frac{19}{10^2} = 0.19$$

(because we have loops in the network)

Commands in Pajek: Density

Screenshots:

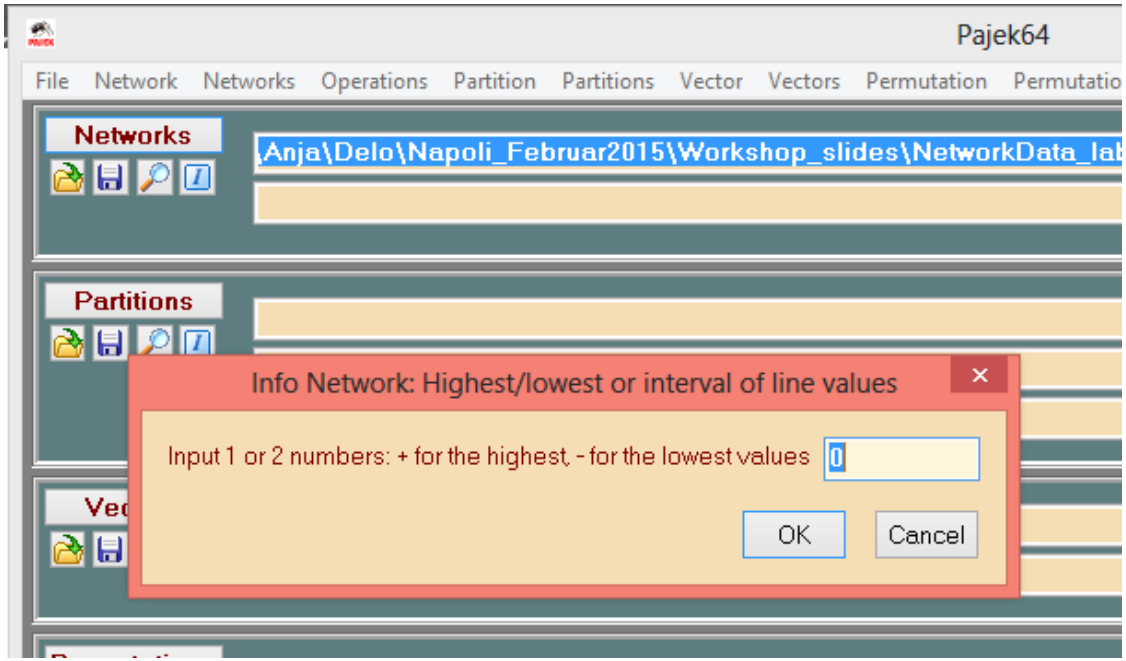


Commands in Pajek: Density

`Network/Info/General [0]`

Commands in Pajek: Density

Screenshots:

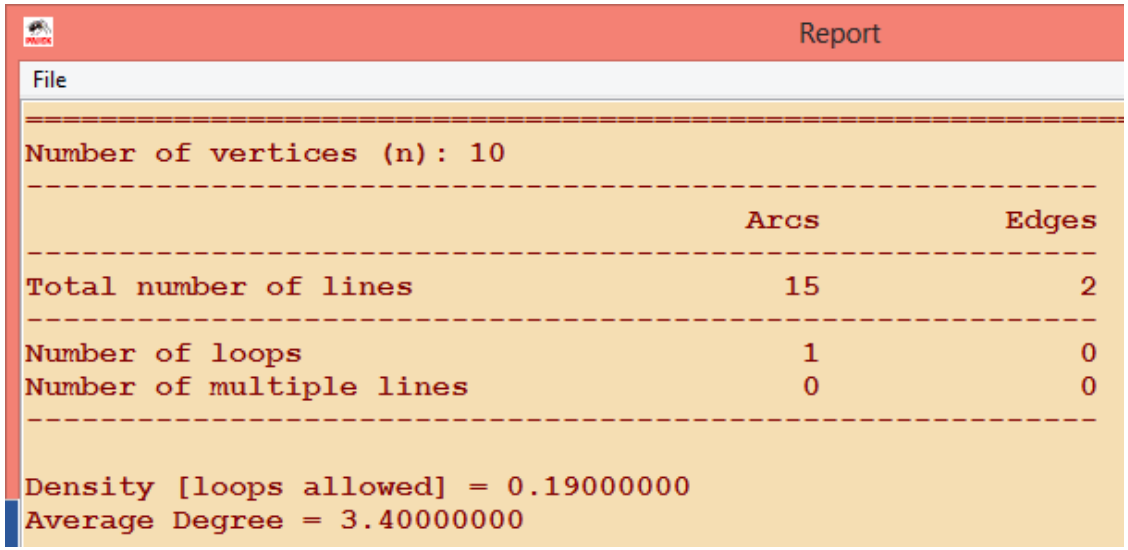


Commands in Pajek: Density

`Network/Info/General [0]`

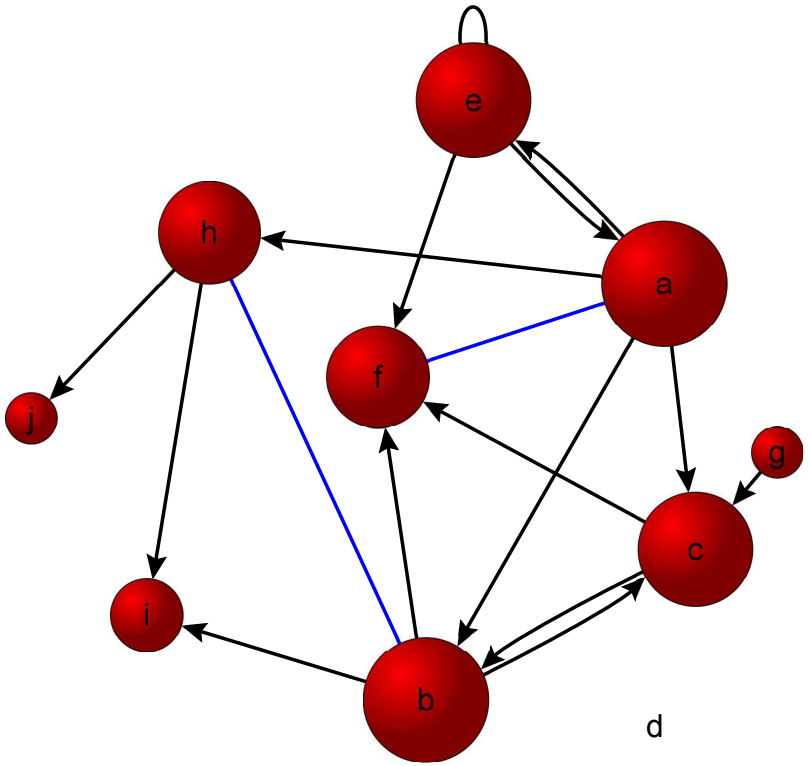
Commands in Pajek: Density

Pajek's Report Window



Degree of a vertex

Degree (also all-degree) of a vertex v, $deg(v)$, is the number of lines incident with a vertex (where loops are counted twice).

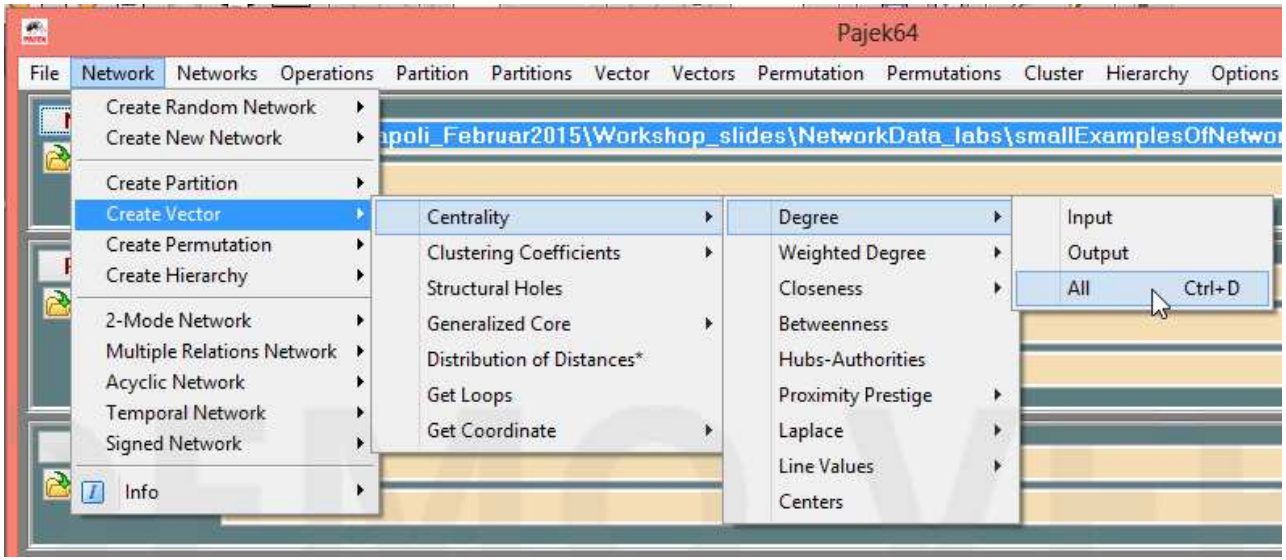


Degrees:

$deg(a) = 6$, $deg(b) = 6$,
$deg(c) = 5$, $deg(d) = 0$,
$deg(e) = 5$, $deg(f) = 4$,
$deg(g) = 1$, $deg(h) = 4$,
$deg(i) = 2$, $deg(j) = 1$

Commands in Pajek: All-degree

Screenshot: All-degree as vector

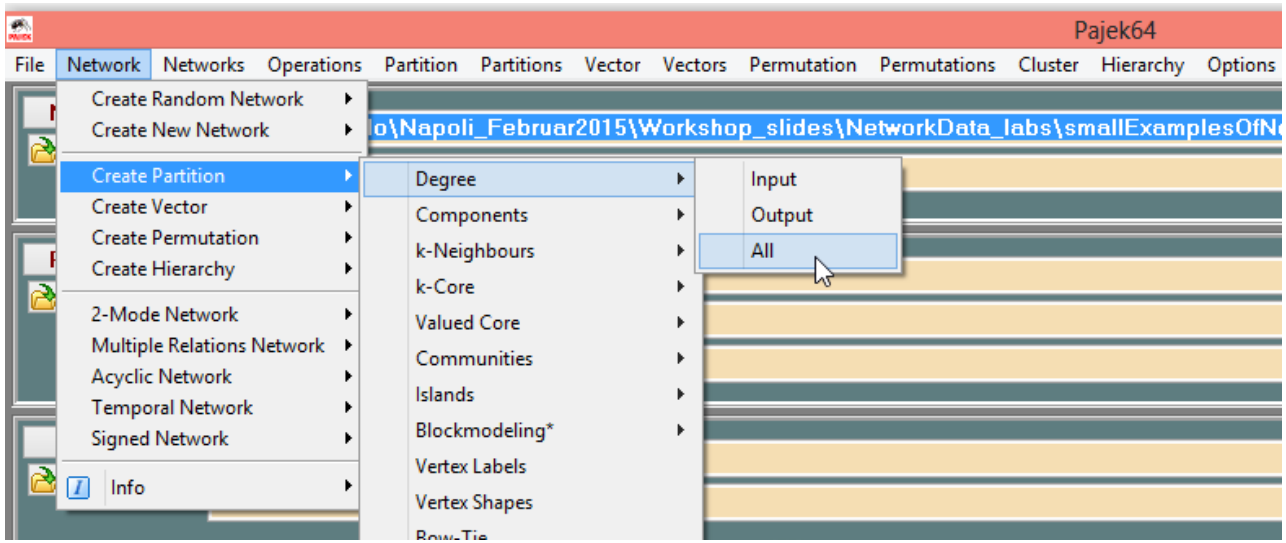


Commands in Pajek: All-degree (produces vector)

`Network/Create Vector/Centrality/Degree/All`

Commands in Pajek: All-degree

Screenshot: All-degree as partition



Commands in Pajek: All-degree (produces partition)

`Network/Create Partition/Degree/All`

Commands in Pajek: Inspect the obtained vector

Screenshot:

Report

File

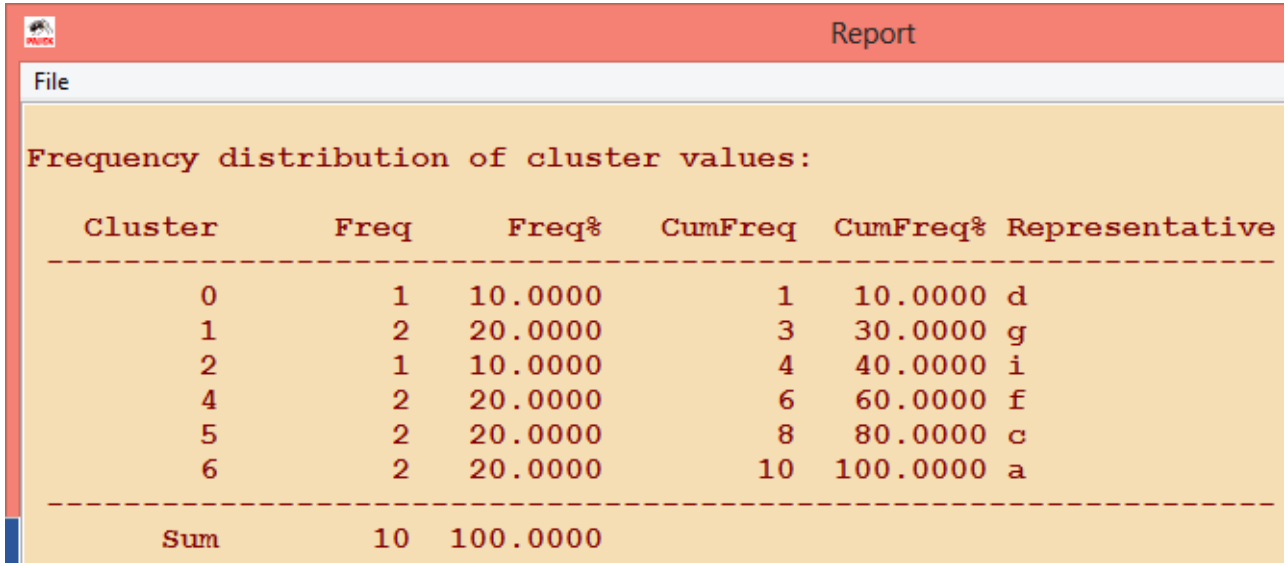
```
================================================================================
1. All Degree of N1 (10)
================================================================================
Dimension: 10
The lowest value:                          0.0000
The highest value:                         6.0000
------------------------------------------------------------
Sum (all values):                         34.0000

Arithmetic mean:                           3.4000
Median:                                    4.0000
Standard deviation:                        2.1071
 2.5% Quantile:                            0.2250
 5.0% Quantile:                            0.4500
95.0% Quantile:                            6.0000
97.5% Quantile:                            6.0000

      Vector Values                                  Frequency       Freq%      CumFreq  CumFreq%
-------------------------------------------------------------------------------------------------
 (                    ...          0.0000]          1      10.0000          1   10.0000
 (               0.0000 ...        1.0000]          2      20.0000          3   30.0000
 (               1.0000 ...        2.0000]          1      10.0000          4   40.0000
 (               2.0000 ...        3.0000]          0       0.0000          4   40.0000
 (               3.0000 ...        4.0000]          2      20.0000          6   60.0000
 (               4.0000 ...        5.0000]          2      20.0000          8   80.0000
 (               5.0000 ...        6.0000]          2      20.0000         10  100.0000
-------------------------------------------------------------------------------------------------
 Total                                             10     100.0000
```

Commands in Pajek: Inspect the obtained vector of all-degree

`Vector/Info [0] [7]`

Commands in Pajek: Inspect the obtained partition

Screenshot:

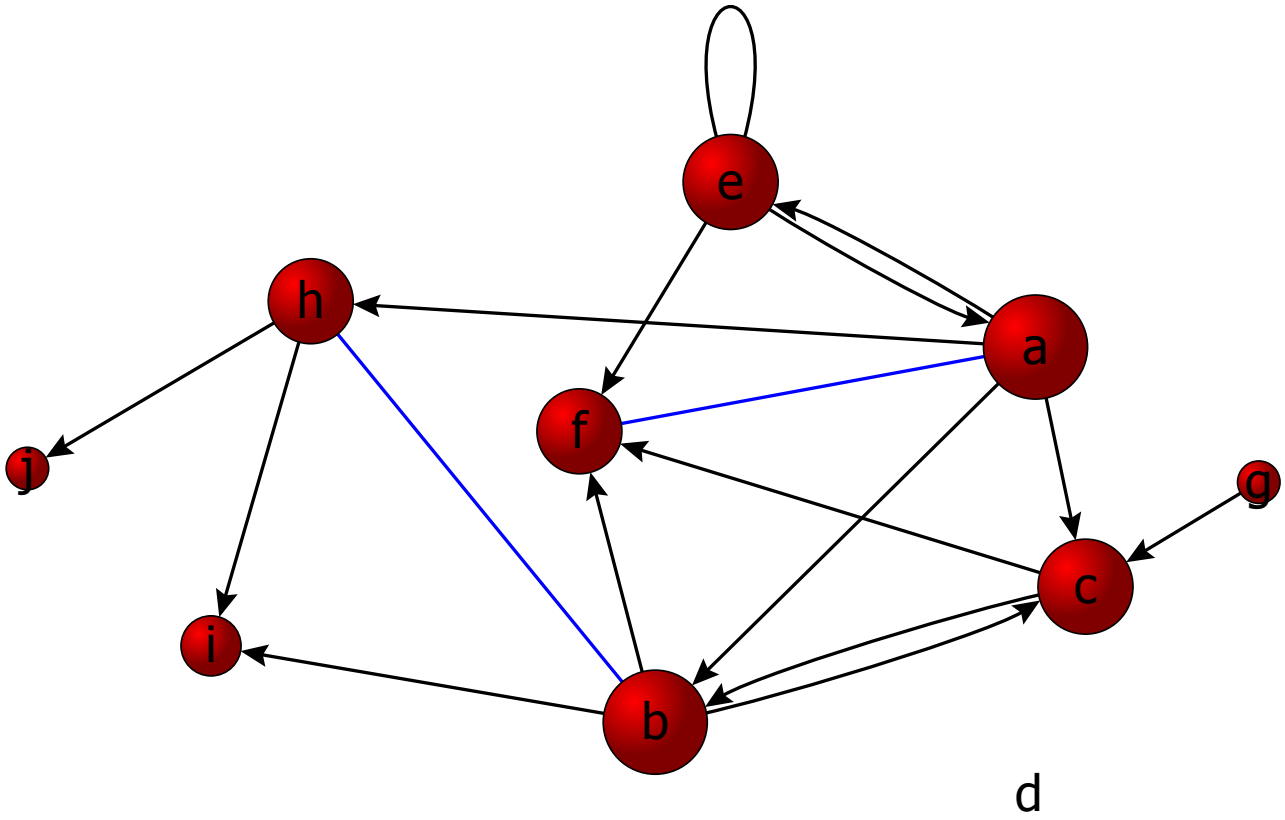


Commands in Pajek: Inspect the obtained partition of all-degree

```
Partition/Info [1] [0]
```

Commands in Pajek: Draw a network according to its all-degree

Figure:

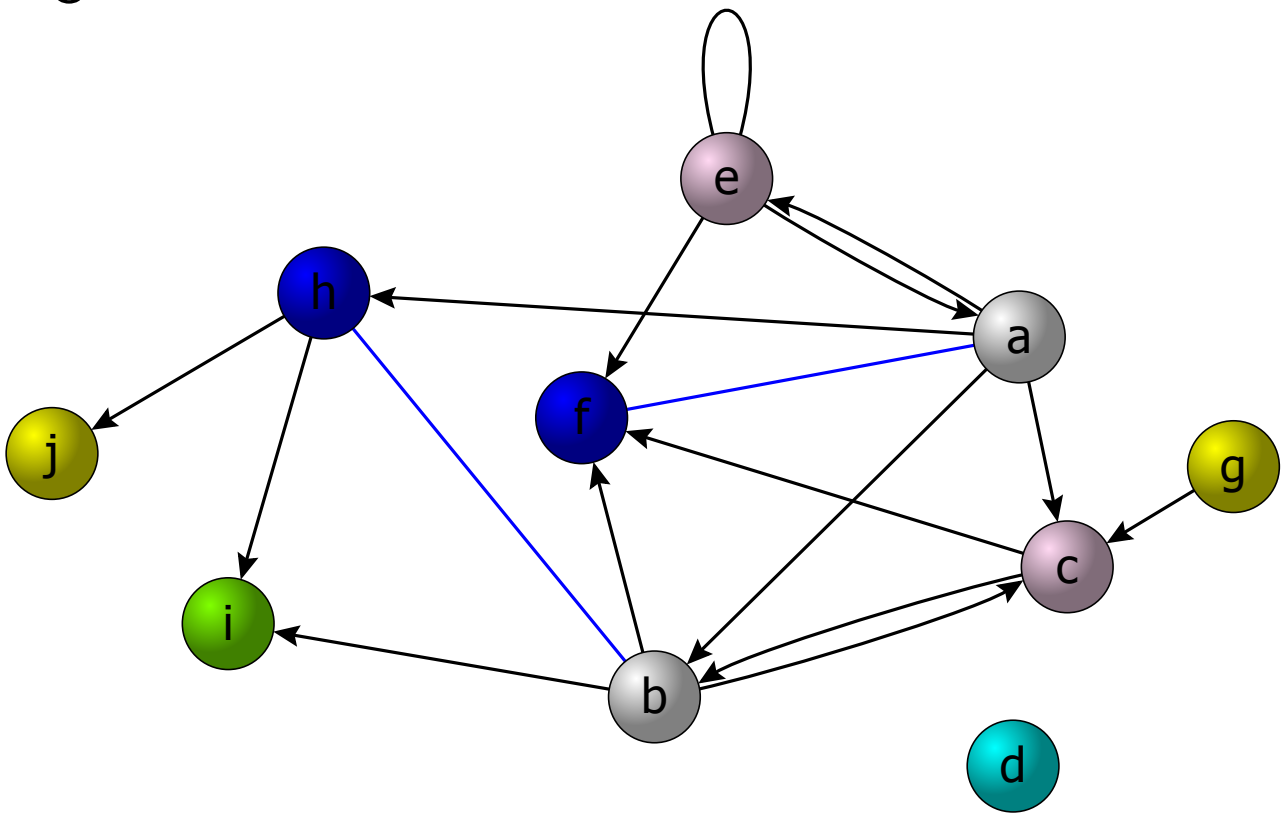


Commands in Pajek: Draw a network with its vector all-degree

`Draw/Network+First Vector`

Commands in Pajek: Draw a network according to its all-degree

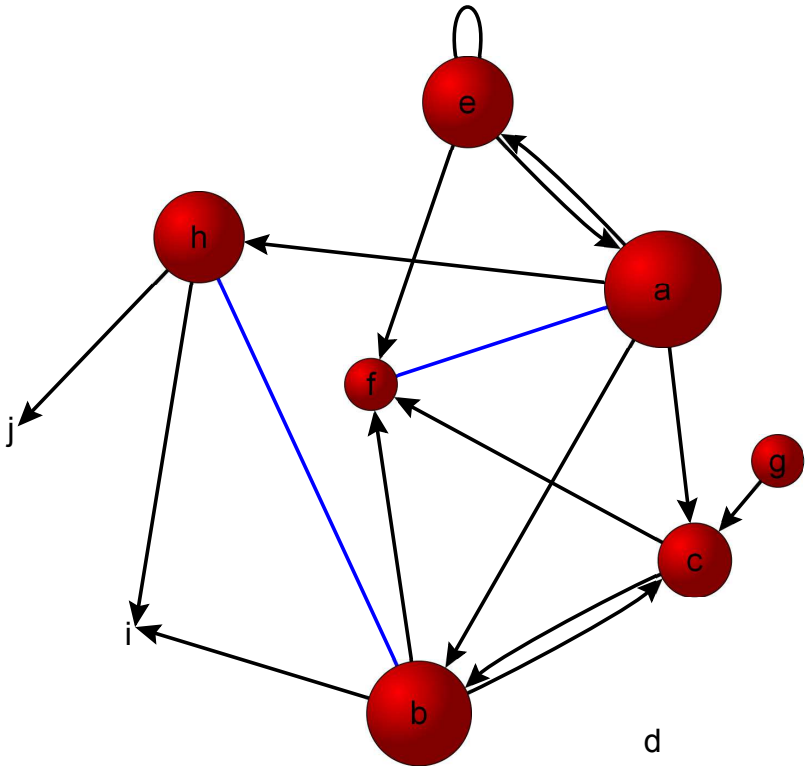
Figure:



Commands in Pajek: Draw a network with its partition all-degree

`Draw/Network+First Vector`

Out-degree of a vertex



Indegrees:

$outdeg(a) = 5$, $outdeg(b) = 4$,
$outdeg(c) = 2$, $outdeg(d) = 0$,
$outdeg(e) = 3$, $outdeg(f) = 1$,
$outdeg(g) = 1$, $outdeg(h) = 3$,
$outdeg(i) = 0$, $outdeg(j) = 0$

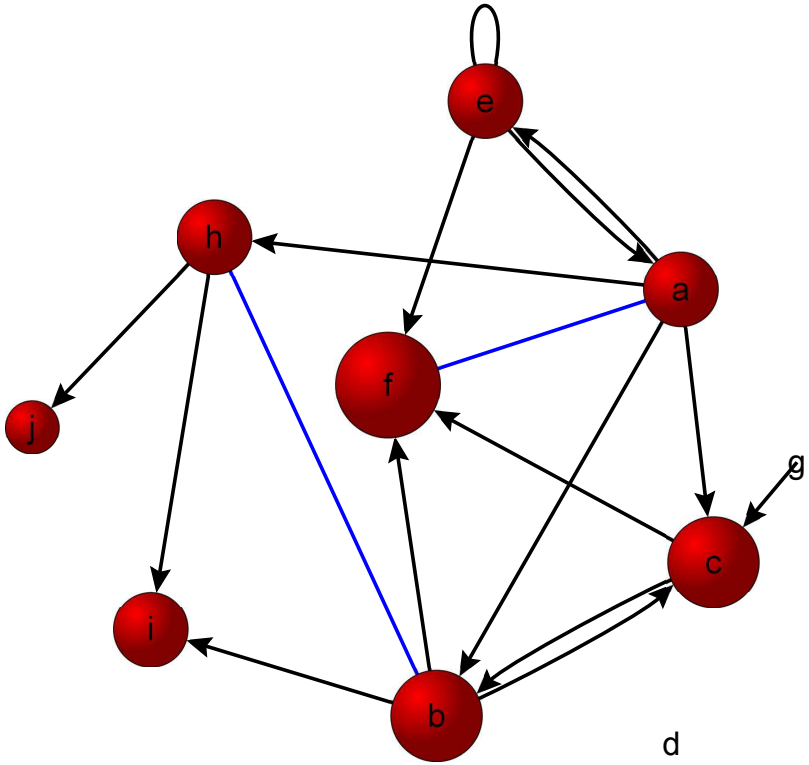
Commands in Pajek: Out-degree

`Network/Create Partition/Degree/Out`

or

`Network/Create Vector/Centrality/Degree/Out`

In-degree of a vertex



Indegrees:

$indeg(a) = 2$, $indeg(b) = 3$,
$indeg(c) = 3$, $indeg(d) = 0$,
$indeg(e) = 2$, $indeg(f) = 4$,
$indeg(g) = 0$, $indeg(h) = 2$,
$indeg(i) = 2$, $indeg(j) = 1$

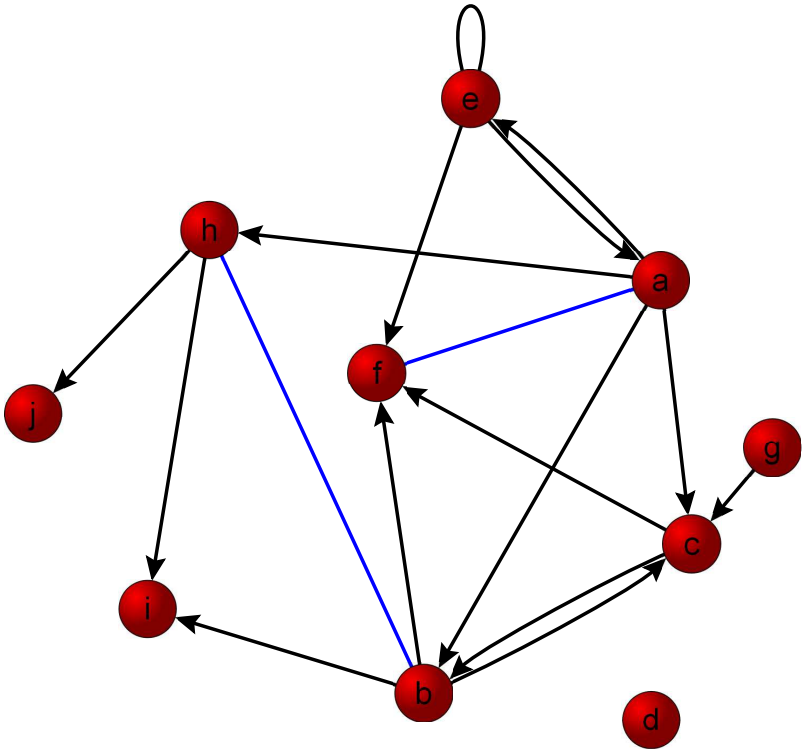
Commands in Pajek: In-degree

`Network/Create Partition/Degree/In`

or

`Network/Create Vector/Centrality/Degree/In`

Degrees in a network

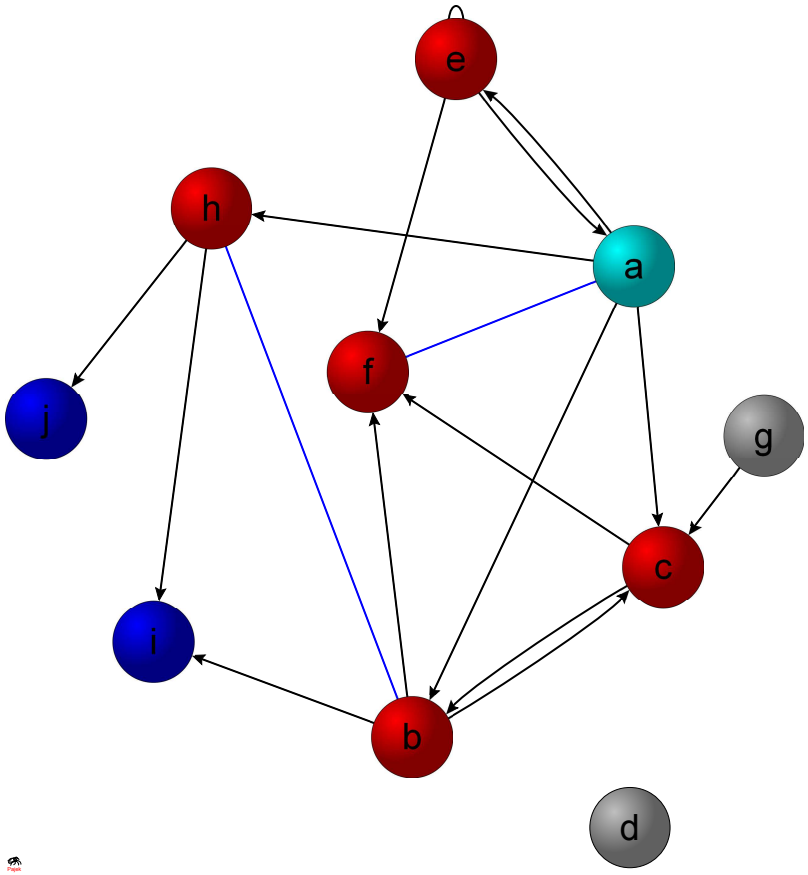


$$\sum_{v \in \mathcal{V}} indeg(v) = \sum_{v \in \mathcal{V}} outeg(v) = |\mathcal{A}| + 2|\mathcal{E}|$$

v is an initial vertex $\Leftrightarrow indeg(v) = 0$
v is a terminal vertex $\Leftrightarrow outdeg(v) = 0$
v is an isolated vertex $\Leftrightarrow deg(v) = 0$

Initial vertices: d, g
Terminal vertices: d, i, j
Isolated vertex: d

Neighborhood of vertices



k-neighbors of vertex a (based on out-degree):

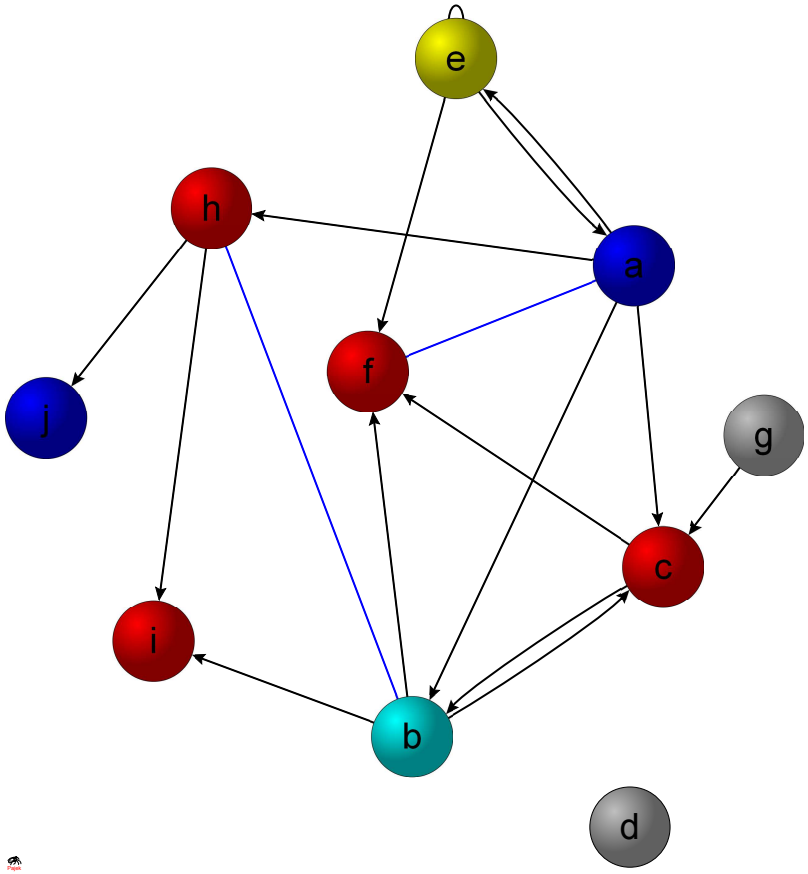
$N^1(a) = \{b, c, e, f, h\}$
$N^2(a) = \{i, j\}$
$N^3(a) = \{\}$
$N^4(a) = \{\}$

d and g are not reachable from vertex b.

Commands in Pajek: Neighbors

`Network/Create Partition/k-Neighbours/`

Neighborhood of vertices



k-neighbors of vertex a (based on out-degree):

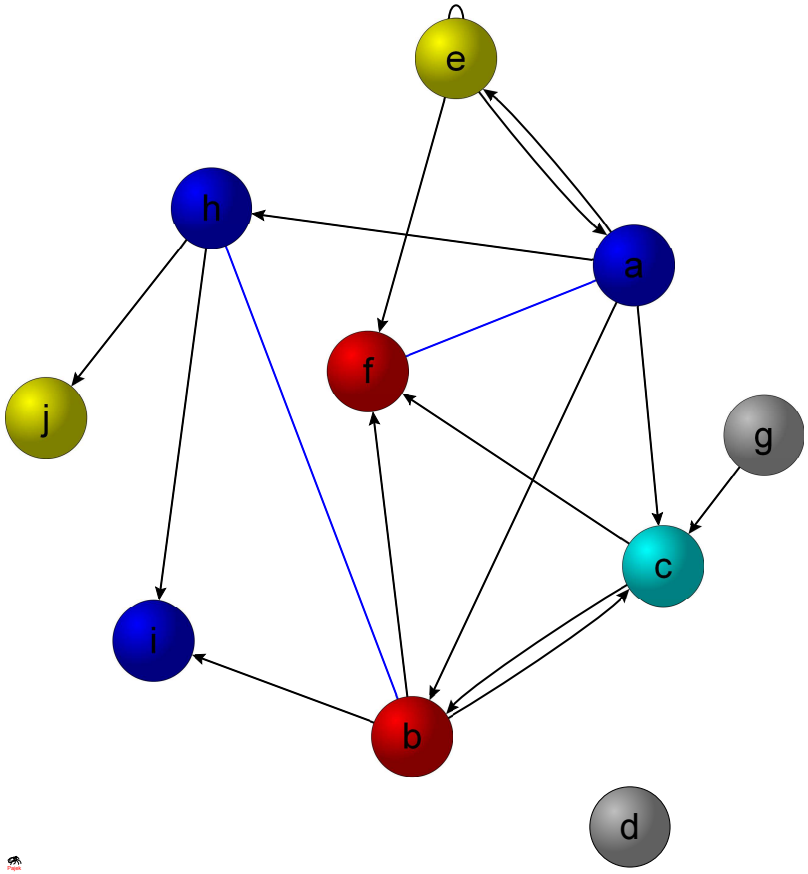
$N^1(a) = \{c, f, h, i\}$
$N^2(a) = \{a, j\}$
$N^3(a) = \{e\}$

d and g are not reachable from vertex b.

Commands in Pajek: Neighbors

`Network/Create Partition/k-Neighbours/`

Neighborhood of vertices



k-neighbors of vertex c (based on out-degree):

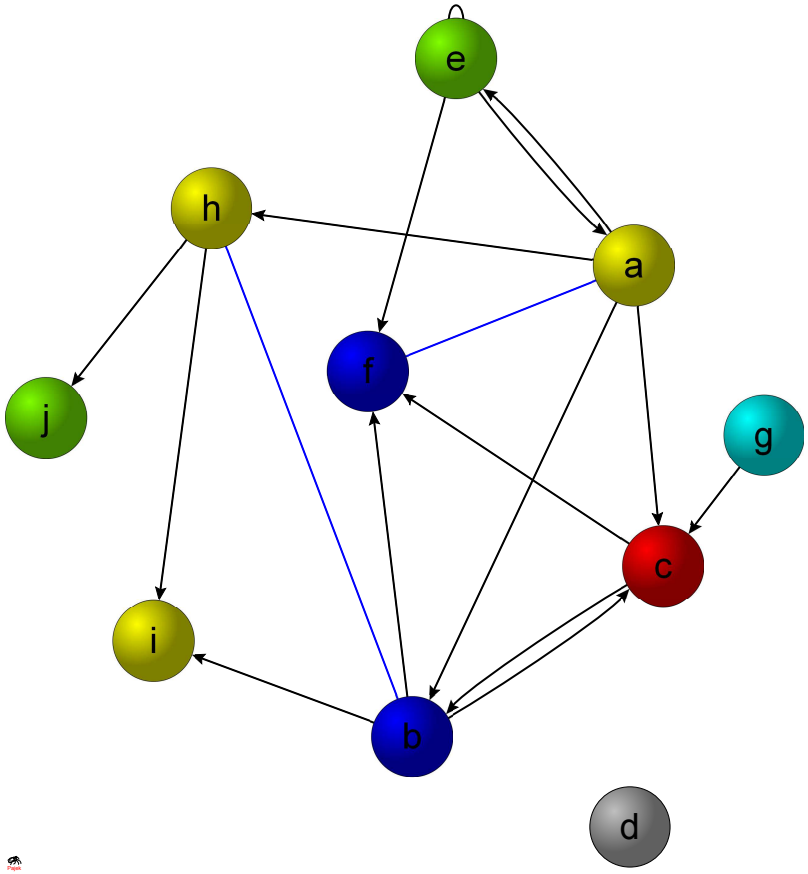
$N^1(a) = \{b, f\}$
$N^2(a) = \{a, h, i\}$
$N^3(a) = \{e, j\}$
$N^4(a) = \{\}$

d and g are not reachable from vertex c.

Commands in Pajek: Neighbors

`Network/Create Partition/k-Neighbours/`

Neighborhood of vertices



k-neighbors of vertex g (based on out-degree):

$N^1(a) = \{c\}$
$N^2(a) = \{b, f\}$
$N^3(a) = \{a, h, i\}$
$N^4(a) = \{e, j\}$

d is not reachable from vertex g.

Commands in Pajek: Neighbors

`Network/Create Partition/k-Neighbours/`

Important units in a network

The most important distinction between different important unit measures is based on whether the considered network is directed or undirected. This gives us two main types of measures of importance:

- undirected case: measures of centrality. Example: A city is central if a lot of roads are passing through it.
- directed case with two subgroups of prestige:
  - measures of support, based on incoming arcs; and
  - measures of inuence, based on out-going arcs. Examples: A unit has high inuence if (s)he gives commands to several others. A unit has high support if a lot of people vote for her/him.

If we change the direction of all arcs (replace the relation with its inverse relation) the measure of in uence becomes a measure of support, and vice versa.

Units centrality

A unit is central if

- it has high degree,
- it is easy accessible or close to all other units,
- it lies on several geodesics (shortest paths) between units.

Betweenness

The units that can control the information fow in the network are important.

If we assume that the information fow uses only the shortest paths (geodesics) we get a measure of betweeness (Freeman 1977):

$$c_B(v) = \sum \frac{\#\text{ of shortest paths between units } y \text{ and z through unit } x}{\#\text{ of shortest path between units } y \text{ and } z}$$

Betweenness

Relative betweenness centrality is defned as:

- $C_B(v) = \frac{c_B(v)}{(n-1)(n-2)/2}$ for undirected networks, and,
- $C_B(v) = \frac{c_B(v)}{(n-1)(n-2)}$ for directed networks.

Commands in Pajek: Betweenness

Commands in Pajek: Betweenness

`Network/Create Vector/Centrality/Betweenness`

Explore 5 vertices with the highest betweenness scores:

`Vector/Info [5] [#4]`

Draw a network: `Draw/Network+First Vector`

Pajek's Report Window:

```
================================================================
4. Betweenness centrality in N17 (10)
================================================================
Dimension: 10
The lowest value:                          0.0000
The highest value:                         0.2083

Highest values:

      Rank    Vertex                        Value    Id
----------------------------------------------------------------
         1         1                      0.2083    a
         2         2                      0.1597    b
         3         6                      0.1111    f
         4         8                      0.1042    h
         5         3                      0.0972    c
----------------------------------------------------------------
```

Commands in Pajek: Betweenness

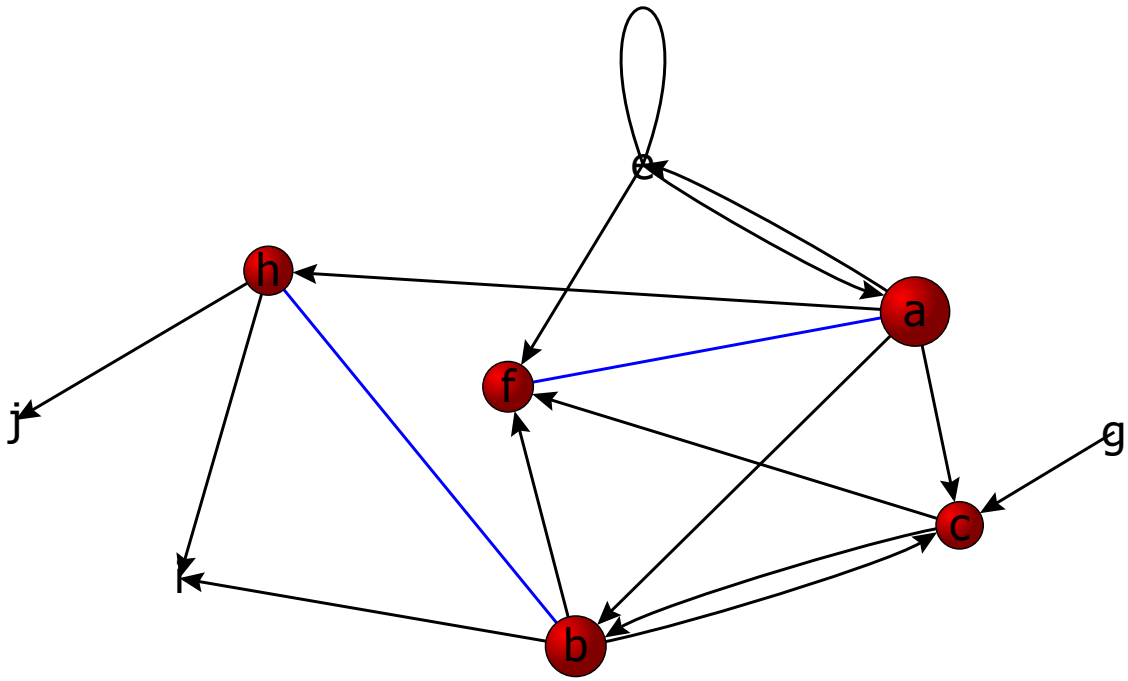
Commands in Pajek: Betweenness

`Network/Create Vector/Centrality/Betweenness`

Explore 5 vertices with the highest betweenness scores:

`Vector/Info` [5] [#4]

Draw a network: `Draw/Network+First Vector`



Closenness

Sabidussi (1966) introduced a measure of centrality according to the closeness of unit x to all others:
$c_C(x) = \sum_{u \in U} \frac{1}{d(x;y)}$, where $d(x;y)$ is the distance of shortest path between unit x and y, and U is the set of all units.

The most central units, according to the closeness centrality measure, can quickly interact with all others because they are close to all others.

Relative closenness

Sabidussi (1966) introduced a relative closeness measure: $C_C(x) = \sum_{u \in U} \frac{n-1}{d(x;y)}$.

Closeness measure can be computed for undirected and directed networks. There are two possibilities for directed networks:

- incoming ties: in how many steps the selected unit can be reached from all others (closeness support measure);
- outgoing ties: in how many steps all other units can be reached from the selected one (closeness inuence measure).

Commands in Pajek: Closeness based on indegree

Commands in Pajek: Closeness based on indegree

`Network/Create Vector/Centrality/Closeness/Indegree`

Explore 5 vertices with the highest in-closeness scores:

`Vector/Info [5] [#4]`

Draw a network: `Draw/Network+First Vector`

Pajek's Report Window:

```
================================================================
5. Input closeness centrality in N17 (10)
================================================================
Dimension: 10
The lowest value:                          0.0000
The highest value:                         0.5250

Highest values:

      Rank    Vertex                        Value    Id
----------------------------------------------------------------
         1         6                        0.5250   f
         2         3                        0.4667   c
         3         2                        0.4667   b
         4         8                        0.3818   h
         5         9                        0.3733   i
----------------------------------------------------------------
```

Commands in Pajek: Closeness based on indegree

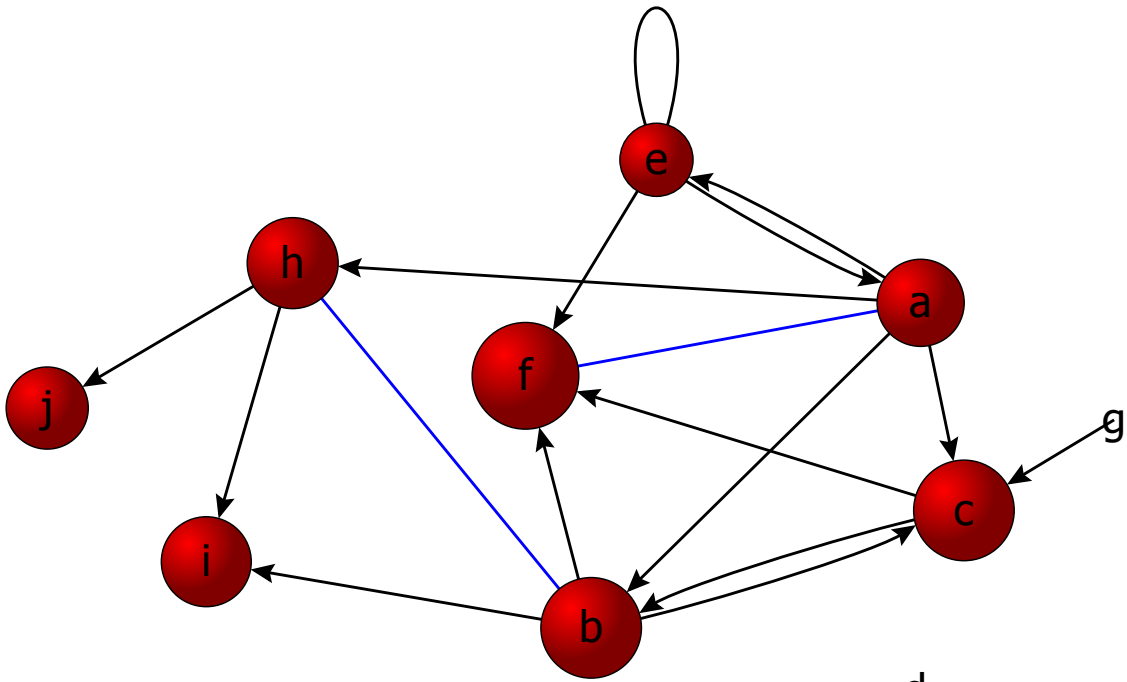
Commands in Pajek: Closeness based on indegree

`Network/Create Vector/Centrality/Closeness/Indegree`

Explore 5 vertices with the highest in-closeness scores:

`Vector/Info [5] [#4]`

Draw a network: `Draw/Network+First Vector`



Commands in Pajek: Closeness based on indegree

Commands in Pajek: Closeness based on outdegree

`Network/Create Vector/Centrality/Closeness/Outdegree`

Explore 5 vertices with the highest out-closeness scores:

`Vector/Info [5] [#4]`

Draw a network: `Draw/Network+First Vector`

Pajek's Report Window:

```
============================================================
6. Output closeness centrality in N17 (10)
============================================================
Dimension: 10
The lowest value:                          0.0000
The highest value:                         0.6222

Highest values:

      Rank    Vertex                        Value    Id
------------------------------------------------------------
         1         1                        0.6222   a
         2         2                        0.5091   b
         3         3                        0.4000   c
         4         5                        0.4000   e
         5         8                        0.4000   h
------------------------------------------------------------
```

Commands in Pajek: Closeness based on indegree

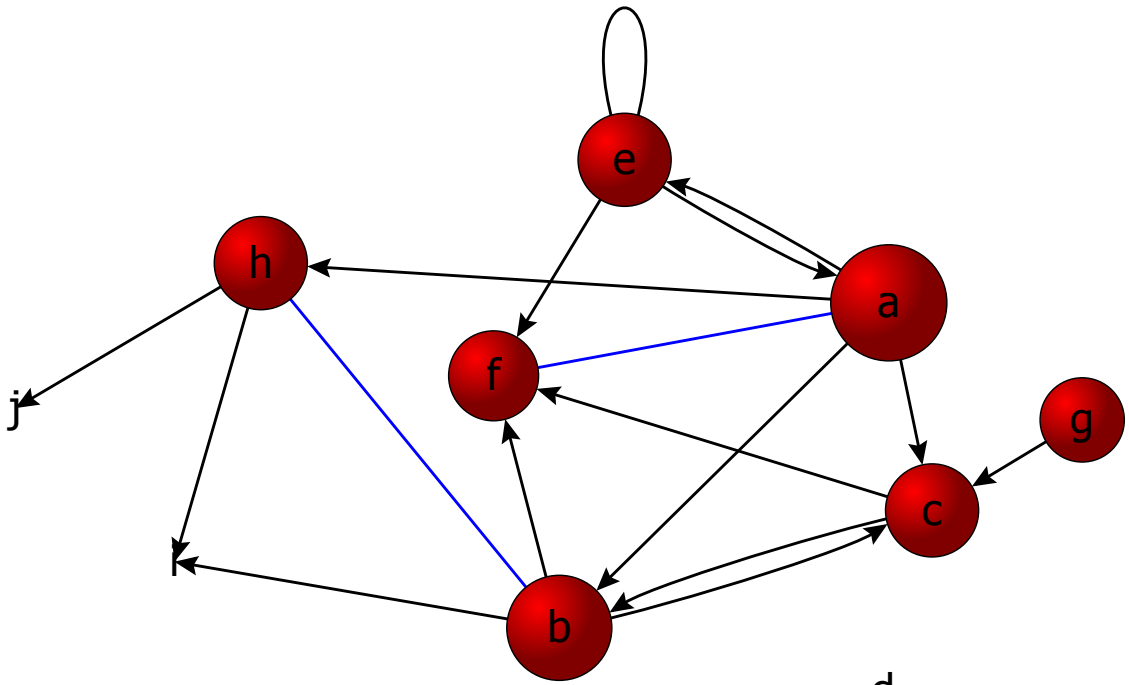
Commands in Pajek: Closeness based on outdegree

`Network/Create Vector/Centrality/Closeness/Outdegree`

Explore 5 vertices with the highest out-closeness scores:

`Vector/Info [5] [#4]`

Draw a network: `Draw/Network+First Vector`



Commands in Pajek: Closeness based on alldegree

Commands in Pajek: Closeness based on alldegree

`Network/Create Vector/Centrality/Closeness/All`

Explore 5 vertices with the highest all-closeness scores:

`Vector/Info [5] [#4]`

Draw a network: `Draw/Network+First Vector`

Pajek's Report Window:

```
============================================================
7. All closeness centrality in N17 (10)
============================================================
Dimension: 10
The lowest value:                          0.0000
The highest value:                         0.6545

Highest values:

      Rank    Vertex                        Value    Id
------------------------------------------------------------
         1         1                      0.6545    a
         2         2                      0.6545    b
         3         3                      0.5538    c
         4         6                      0.5538    f
         5         8                      0.5538    h
------------------------------------------------------------
```

Commands in Pajek: Closeness based on alldegree

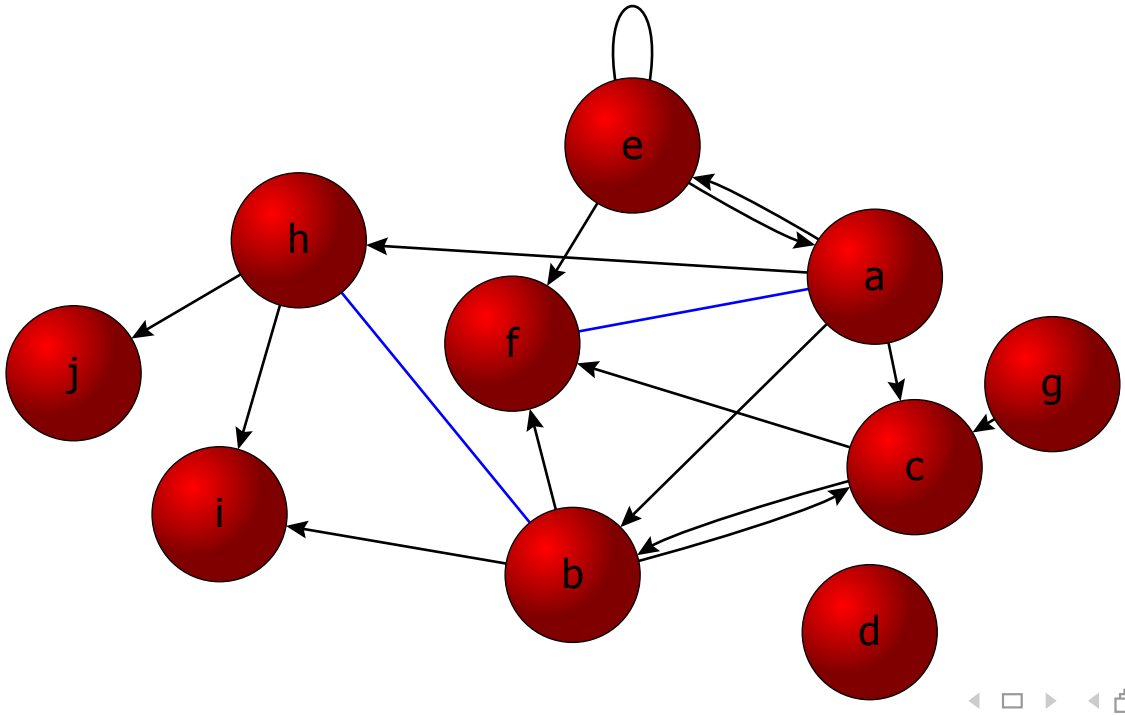
Commands in Pajek: Closeness based on alldegree

`Network/Create Vector/Centrality/Closeness/All`

Explore 5 vertices with the highest all-closeness scores:

`Vector/Info` [5] [#4]

Draw a network: `Draw/Network+First Vector`



Hubs and authorities

In the directed networks another two types of important units can be identiffed (see Kleinberg, 1998): hubs and authorities.

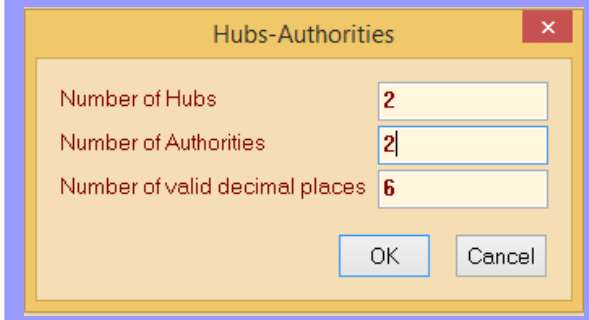
A good authority is selected by good hubs; and good hub points to good authorities.

Commands in Pajek: Hubs and authorities

Commands in Pajek: Hubs and authorities

`Network/Create Vector/Centrality/Hubs and authorities`

Select the number of hubs and authorities you want to find:



Draw a network:

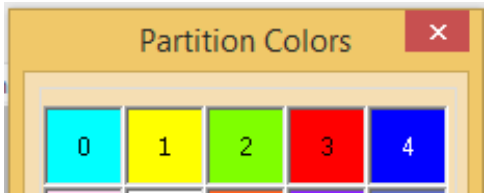
`Draw/Network+First Partition`

Commands in Pajek: Hubs and authorities

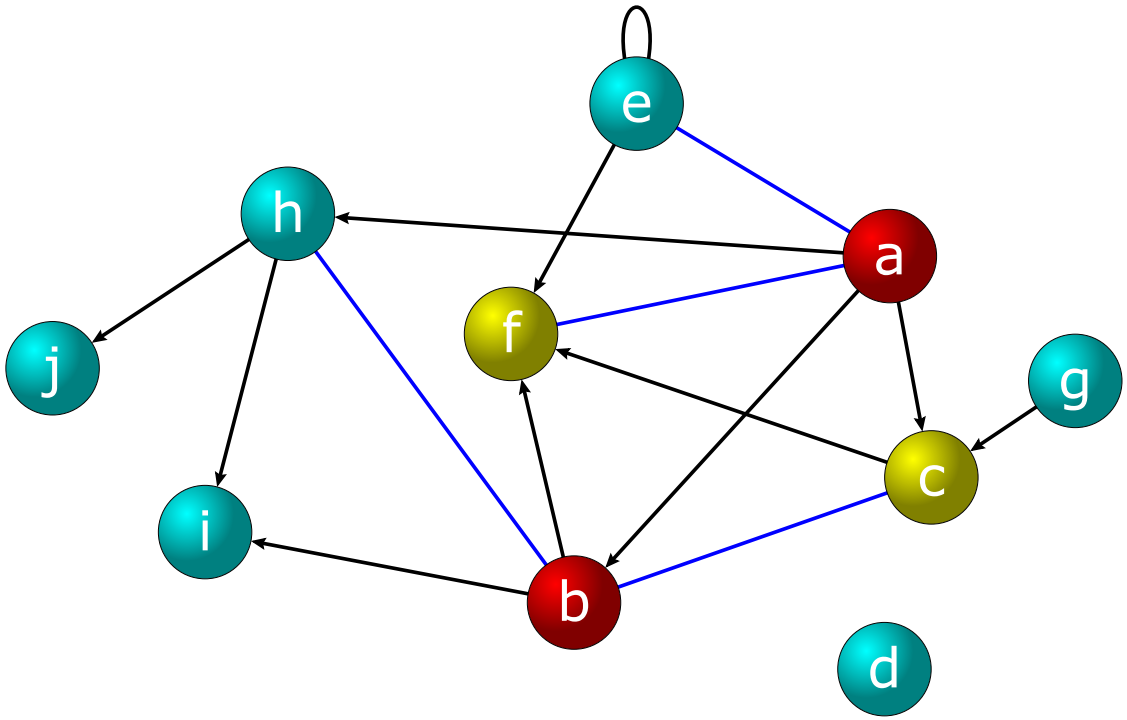
Pajek's Report Window:

```
Partition Clusters:
1 - Authority
2 - Authority and Hub
3 - Hub
```

Selected partition colors:



Commands in Pajek: Hubs and authorities



What else can be calculated in Pajek

- line Islands,
- vertex islands,
- weak components,
- cores
- strong components,
- clusters based on blockmodeling,
- ...