Photo: Vladimir Batagelj: ***Dragon***

Course 10
Network Analysis
Introduction

Vladimir Batagelj

University of Ljubljana

ECPR Summer School, July 30 – August 16, 2008

Faculty of Social Sciences, University of Ljubljana

Outline

Teachers and Teaching Assistants

prof. Vladimir Batagelj

vladimir.batagelj@fmf.uni-lj.si

prof. Andrej Mrvar

andrej.mrvar@fdv.uni-lj.si

Anja Žnidaršič

anja.znidarsic@siol.net

Ana Slavec

ana.slavec@gmail.com

Course 10: Network Analysis

The course aims to provide an introduction into the main topics and concepts of social network analysis. It focuses on the analysis and visualization of complete networks.

Participants will get understanding of basic network analysis concepts like centrality, cohesion, blockmodeling, … A special attention will be given to the analysis of large networks.

After the course participants should be able to examine data in 'social networks way' – they should be able to identify and formulate their own network analysis problems, solve them using network analysis software and interpret the obtained results.

http://vlado.fmf.uni-lj.si/pub/networks/doc/ecpr08.htm

Setup

Each meeting consists of a 90 minutes lecture, and a 80 minutes lab session with 10 minutes break in between. The lectures are in room 13 and labs in room 24.

The concepts explained are applied in **`Pajek`** during the lab sessions. Students will receive test datasets to practice.

An assignment will be handed out each day after the lectures. The students are expected to return an individual report next day to TA.

The exam will consist of a set of questions to be answered by short answers.

The final grade $= 0.6 \times$ assignments $+ 0.4 \times$ exam.

Program

date	hours	topic
Mon 04	15.30 — 17.00	Introduction to the course. Networks. Pajek and network analysis software. Example.
Tue 05	09.00 — 12.00	(1) Basic network concepts. Partitions and vectors. Visualization. Types of networks. Pajek and network analysis software.
Wed 06		(2) Local and global views. Subnetworks. Cuts. Paths in networks.
Thu 07		(3) Connectivity. Acyclic networks. Short cycles.
Fri 08		(4) Centrality and prestige. Hubs and authorities. Triads. Pattern search.
Mon 11		(5) Cohesion, cliques, cores, generalized cores, islands.
Tue 12		(6) 2-mode networks. 4-rings. 2-mode cores.
Wed 13		(7) Clustering and blockmodeling.
Thu 14		(8) Multiplication of networks. Networks from the tables. Temporal, multirelational and sequences of networks.
Fri 15		(9) Large networks. Scale-free networks.
Sat 16	09.00 — 12.00	Exams.

Development of SNA

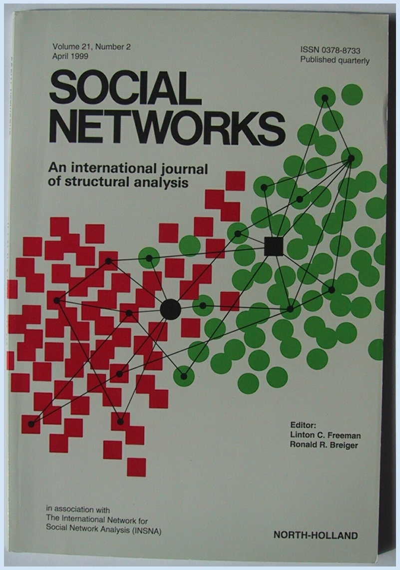
Graph theory: Euler, Hamilton, Kirchoff, Kekule, Ford and Fulkerson, Harary, Berge, ...

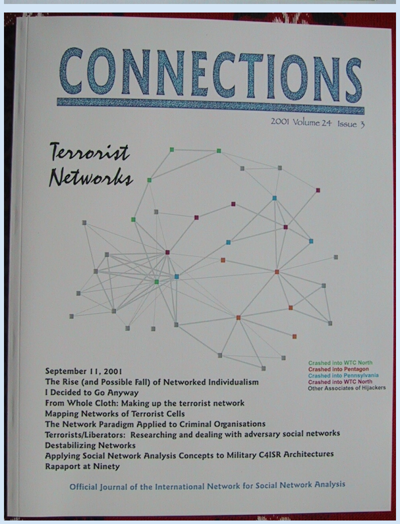
Moreno

- Moreno (1934) – sociometry
- Lewin (1936)
- Warner and Lunt (1941)
- Heider (1946)
- Bavelas (1948) – centrality
- Homans (1950)
- Cartwright and Harary (1956)
- Nadel (1957) – social structure, social positions, roles
- Mitchell (1969)

Freeman L.C. (2004) The Development of Social Network Analysis

Some important events





- International Association of Social Network Analysis – INSNA, 1978
- Journal: Social Networks, 1978
- Newsletter: Connections, 1978
- SUNBELT conferences, 1981
- e-Journal: Journal of Social Structure, 2000

Selected Books on SNA

- J. P Scott: *Social Network Analysis: A Handbook*. SAGE Publications, 2000. Amazon.
- A. Degenne, M. Forsé: *Introducing Social Networks*. SAGE Publications, 1999. Amazon.
- S. Wasserman, K. Faust: *Social Network Analysis: Methods and Applications*. CUP, 1994. Amazon.
- W. de Nooy, A. Mrvar, V. Batagelj: *Exploratory Social Network Analysis with Pajek*, CUP, 2005. Amazon. ESNA page.
- P. Doreian, V. Batagelj, A. Ferligoj: *Generalized Blockmodeling*, CUP, 2004. Amazon.
- P.J. Carrington, J. Scott, S. Wasserman (Eds.): *Models and Methods in Social Network Analysis*. CUP, 2005. Amazon.
- U. Brandes, T. Erlebach (Eds.): *Network Analysis: Methodological Foundations*. LNCS, Springer, Berlin 2005. Amazon.

Courses on NA

- Steve Borgatti, UCINET
- Barry Wellman, University of Toronto
- Douglas White, University of California Irvine
- Lada Adamic, University of Michigan
- James Moody, Duke University
- Mark Newman, University of Michigan
- Jon Kleinberg, Cornell University
- Robert A. Hanneman, University of California, Riverside; workshop
- Noah Friedkin, University of California, Santa Barbara
- John Levi Martin, University of Wisconsin, Madison
- Vladimir Batagelj, University of Ljubljana
- Andrej Mrvar, University of Ljubljana

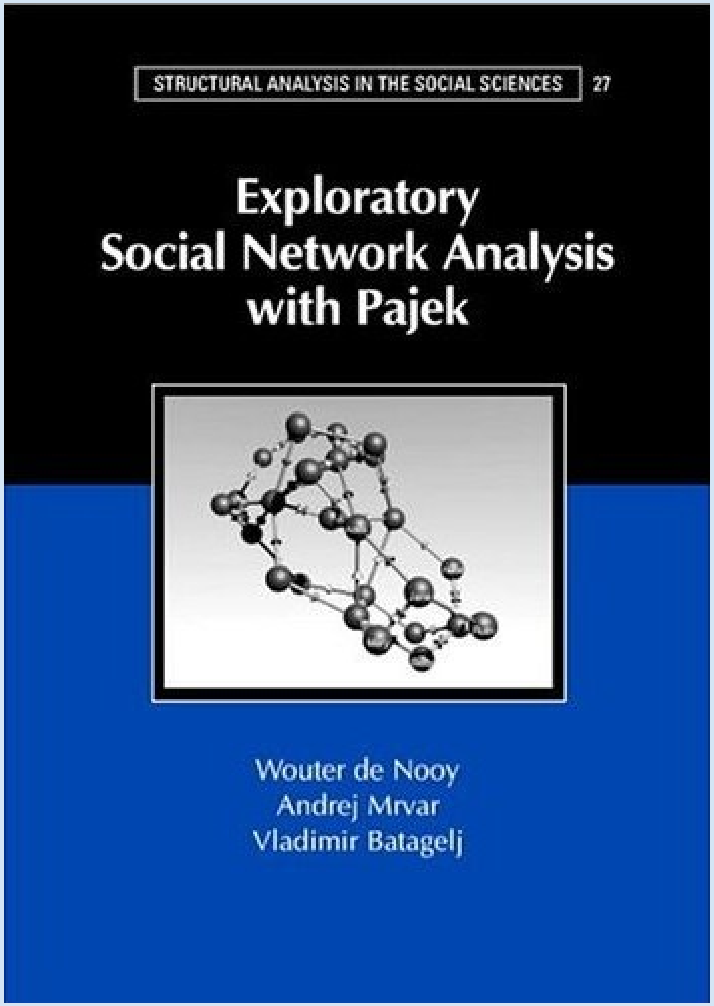
Software for SNA

UCINET, NetDraw	**Pajek**	Netminer
Visone	SNA/R	StOCNET
Negopy	InFlow	GUESS
NetworkX	prefuse	JUNG
BGL/Python		

See also the INSNA list and recent overview by M. Huisman and M.A.J. van Duijn.

Visual Complexity

ESNA **Pajek**



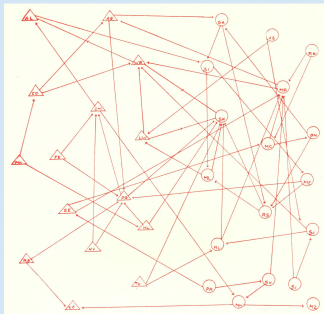
An introduction to social network analysis with **Pajek** is available in the book ESNA (de Nooy, Mrvar, Batagelj 2005).

Pajek – program for analysis and visualization of large networks is freely available, for noncommercial use, at its web site.

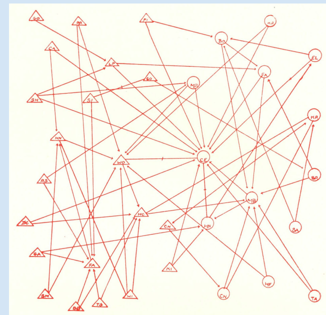
http://pajek.imfm.si/

Moreno: Who shall survive?

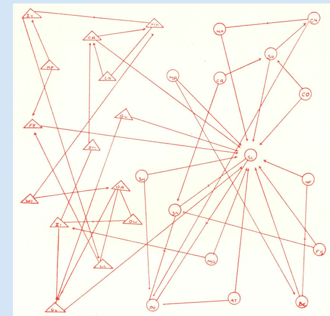
K:



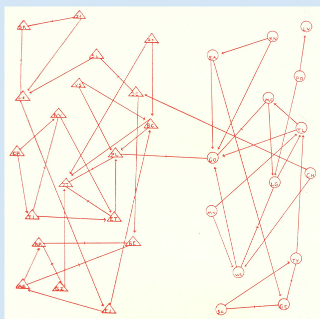
1:



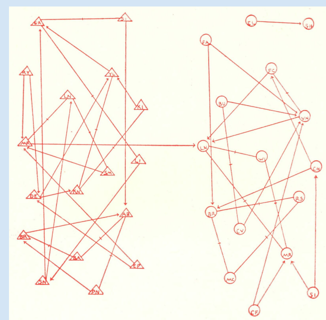
2:



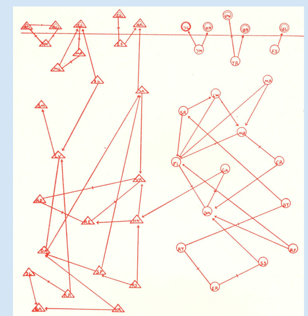
3:



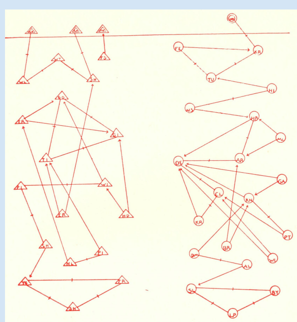
4:



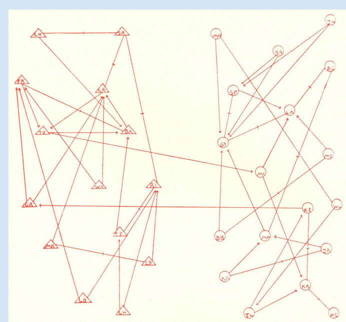
5:



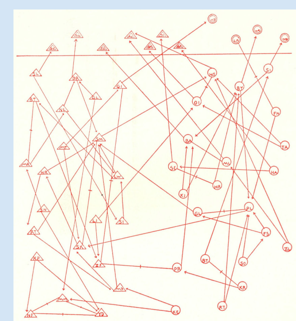
6:



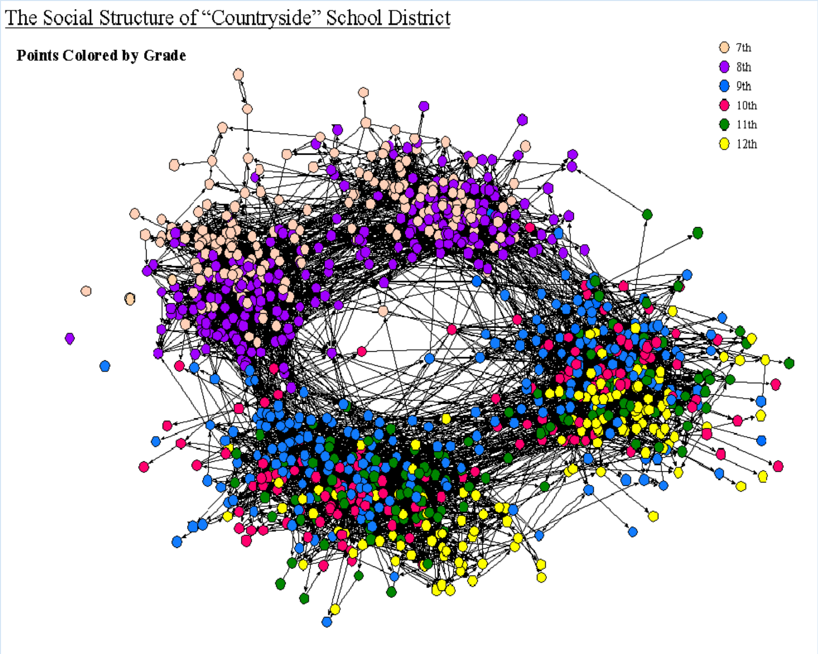
7:

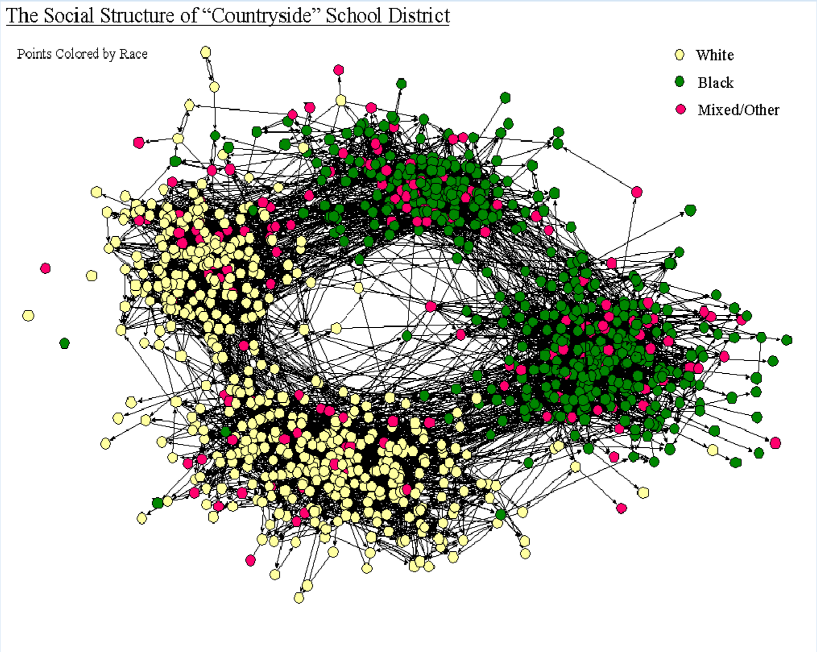


8:

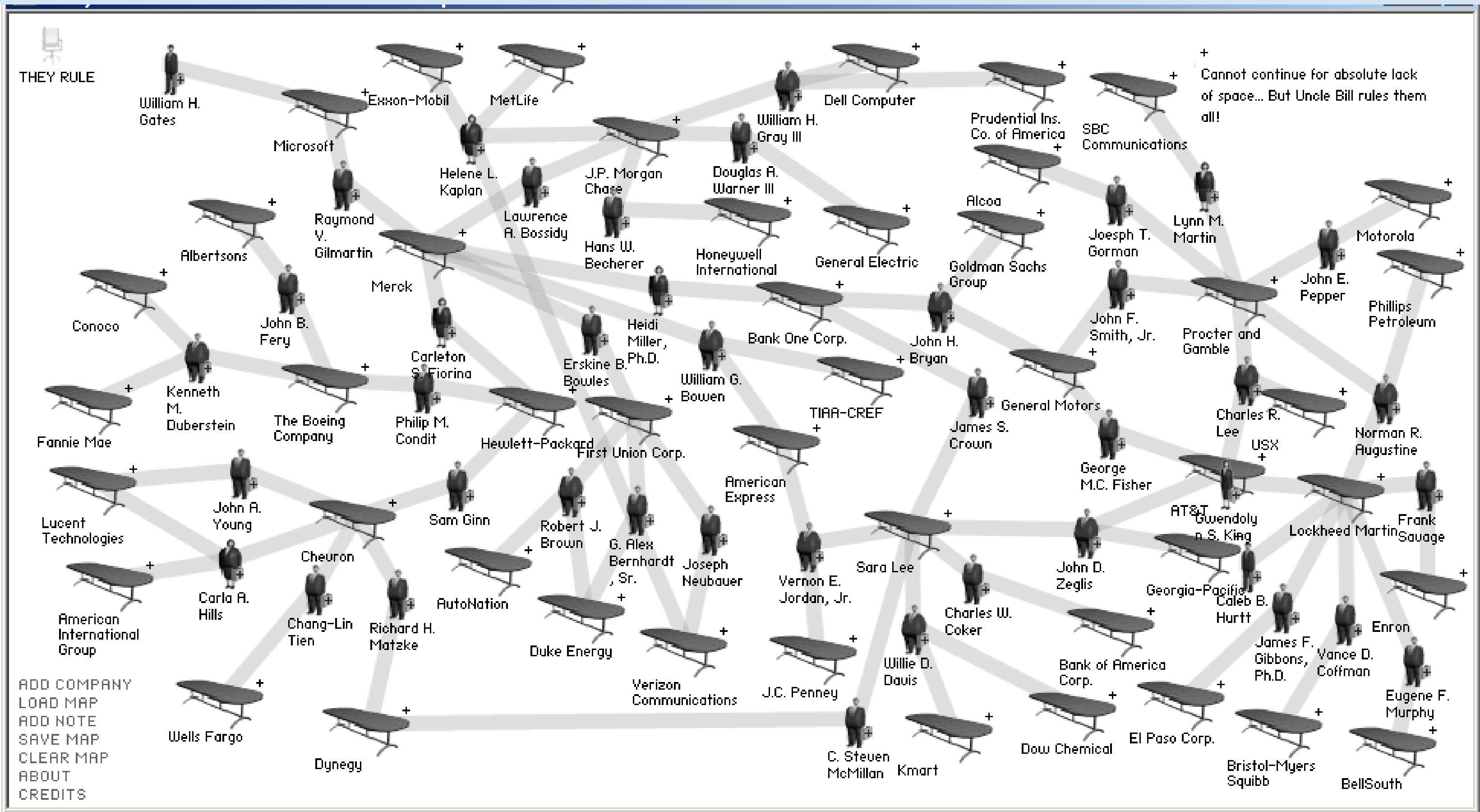


James Moody: Display of properties – school



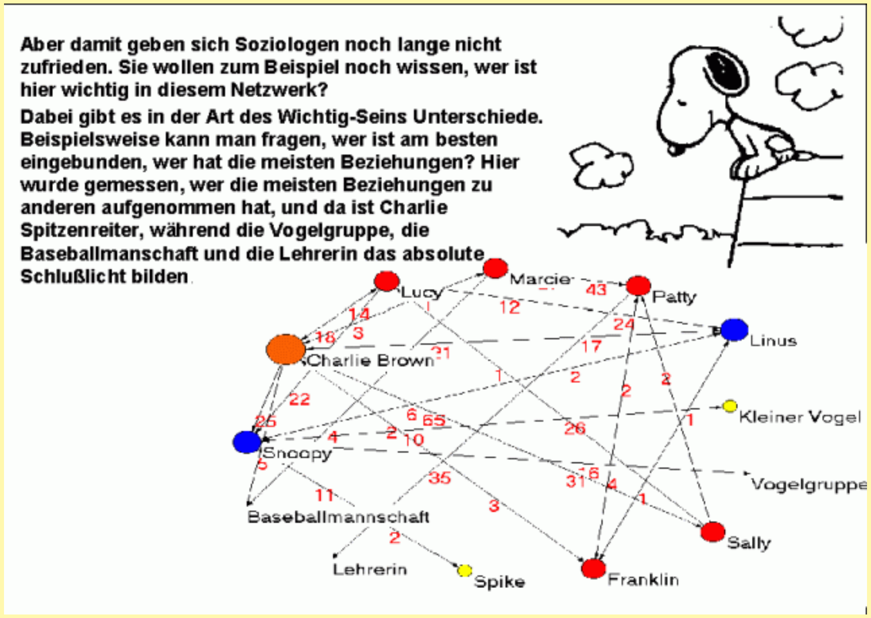


They Rule


THEY RULE
William H. Gates
Microsoft
Exxon-Mobil
MetLife
Helene L. Kaplan
J.P. Morgan Chase
William H. Gray III
Dell Computer
Prudential Ins. Co. of America
SBC Communications
Cannot continue for absolute lack of space... But Uncle Bill rules them all!
Douglas A. Warner III
Raymond V. Gilmartin
Lawrence A. Bossidy
Albertsons
Hans W. Becherer
Honeywell International
General Electric
Alcoa
Goldman Sachs Group
Joesph T. Gorman
Lynn M. Martin
Motorola
Conoco
Merck
John B. Fery
Carleton S. Fiorina
Erskine B. Bowles
Heidi Miller, Ph.D.
William G. Bowen
Bank One Corp.
John H. Bryan
John F. Smith, Jr.
Procter and Gamble
John E. Pepper
Phillips Petroleum
Fannie Mae
Kenneth M. Duberstein
The Boeing Company
Philip M. Condit
Hewlett-Packard
First Union Corp.
TIAA-CREF
James S. Crown
General Motors
Charles R. Lee
USX
Norman R. Augustine
Lucent Technologies
John A. Young
Sam Ginn
Robert J. Brown
G. Alex Bernhardt, Sr.
Joseph Neubauer
American Express
George M.C. Fisher
AT&T
Gwendolyn S. King
Lockheed Martin
Frank Savage
Chevron
American International Group
Carla A. Hills
Chang-Lin Tien
Richard H. Matzke
AutoNation
Duke Energy
Vernon E. Jordan, Jr.
Sara Lee
Charles W. Coker
John D. Zeglis
Georgia-Pacific
Caleb B. Hurtt
James F. Gibbons, Ph.D.
Vance D. Coffman
Enron
Willie D. Davis
Verizon Communications
J.C. Penney
Bank of America Corp.
Eugene F. Murphy
ADD COMPANY
LOAD MAP
ADD NOTE
SAVE MAP
CLEAR MAP
ABOUT
CREDITS
Wells Fargo
Dynegy
C. Steuen McMillan
Kmart
Dow Chemical
El Paso Corp.
Bristol-Myers Squibb
BellSouth

Lothar Krempel

Networks



Alexandra Schuler/ Marion Laging-Glaser:
Analyse von Snoopy Comics

A *network* is based on two sets – set of *vertices* (nodes), that represent the selected *units*, and set of *lines* (links), that represent *ties* between units. They determine a *graph*. A line can be *directed* – an *arc*, or *undirected* – an *edge*.

Additional data about vertices or lines can be known – their *properties* (attributes). For example: name/label, type, value, . . .

Network = Graph + Data

The data can be measured or computed.

Networks / Formally

A *network* $\mathcal{N} = (\mathcal{V}, \mathcal{L}, \mathcal{P}, \mathcal{W})$ consists of:

- a *graph* $\mathcal{G} = (\mathcal{V}, \mathcal{L})$, where $\mathcal{V}$ is the set of vertices and $\mathcal{L} = \mathcal{E} \cup \mathcal{A}$ is the set of lines; $\mathcal{A}$ is the set of arcs and $\mathcal{E}$ is the set of edges.
  $n = |\mathcal{V}|$, $m = |\mathcal{L}|$
- $\mathcal{P}$ *vertex value functions* / properties: $p : \mathcal{V} \to A$
- $\mathcal{W}$ *line value functions* / weights: $w : \mathcal{L} \to B$

Size of network

The size of a network/graph is expressed by two numbers: number of vertices $n = |\mathcal{V}|$ and number of lines $m = |\mathcal{L}|$.

In a *simple undirected* graph (no parallel edges, no loops) $m \leq \frac{1}{2}n(n-1)$; and in a *simple directed* graph (no parallel arcs) $m \leq n^2$.

The quotient $\gamma = \frac{m}{m_{max}}$ is a *density* of graph.

Small networks (some tens of vertices) – can be represented by a picture and analyzed by many algorithms (***UCINET***, ***NetMiner***).

Also *middle size* networks (some hundreds of vertices) can still be represented by a picture (!?), but some analytical procedures can't be used.

Till 1990 most networks were small – they were collected by researchers using surveys, observations, archival records, ... The advances in IT allowed to create networks from the data already available in the computer(s). *Large* networks became reality. Large networks are too big to be displayed in details; special algorithms are needed for their analysis (**Pajek**).

Large Networks

Large network – several thousands or millions of vertices. Can be stored in computer's memory – otherwise *huge* network.

Usually sparse $m \ll n^2$; typical: $m = O(n)$ or $m = O(n \log n)$.

Examples:

network	size	$n = \|V\|$	$m = \|L\|$	source
ODLIS dictionary	61K	2909	18419	ODLIS online
Citations SOM	168K	4470	12731	Garfield's collection
Molecula 1ATN	74K	5020	5128	Brookhaven PDB
Comput. geometry	140K	7343	11898	BiBTEX bibliographies
English words 2-8	520K	52652	89038	Knuth's English words
Internet traceroutes	1.7M	124651	207214	Internet Mapping Project
Franklin genealogy	12M	203909	195650	Roperld.com gedcoms
World-Wide-Web	3.6M	325729	1497135	Notre Dame Networks
Internet Movie DB	113.6M	1324748	3792390	IMDB
Wikipedia	53.8M	659388	16582425	Wikimedia
US patents	82M	3774768	16522438	Nber
SI internet	38M	5547916	62259968	Najdi Si

Pajek datasets.

Complexity of algorithms

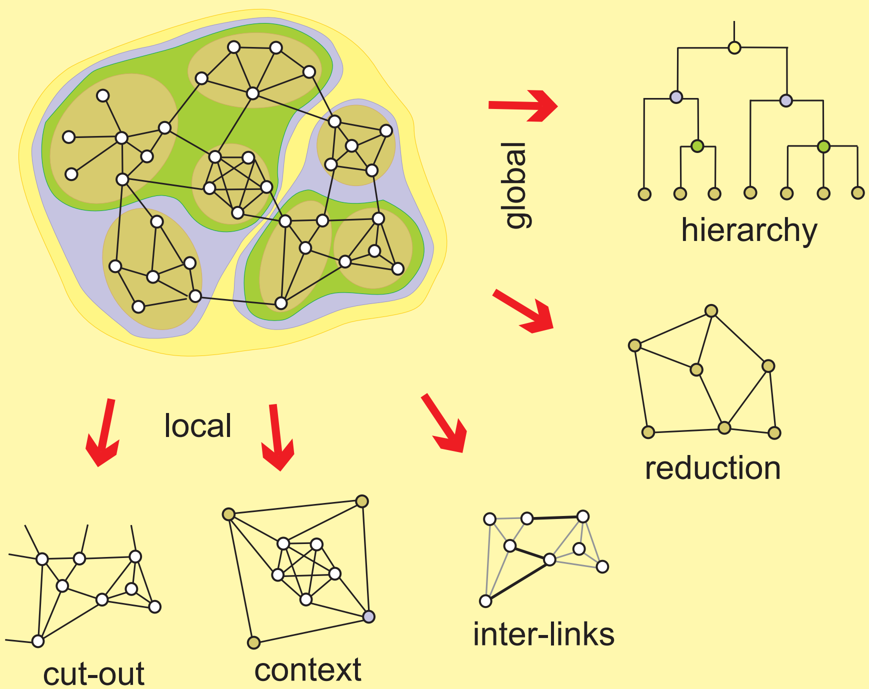
From some thousands to some (tens) millions of units (vertices).

Let us look to time complexities of some typical algorithms:

	T(n)	1.000	10.000	100.000	1.000.000	10.000.000
LinAlg	O(n)	0.00 s	0.015 s	0.17 s	2.22 s	22.2 s
LogAlg	O($n \log n$)	0.00 s	0.06 s	0.98 s	14.4 s	2.8 m
SqrtAlg	O($n\sqrt{n}$)	0.01 s	0.32 s	10.0 s	5.27 m	2.78 h
SqrAlg	O(n^2)	0.07 s	7.50 s	12.5 m	20.8 h	86.8 d
CubAlg	O(n^3)	0.10 s	1.67 m	1.16 d	3.17 y	3.17 ky

For the interactive use on large graphs already quadratic algorithms, O(n^2), are too slow.

Main goals in design of **Pajek**



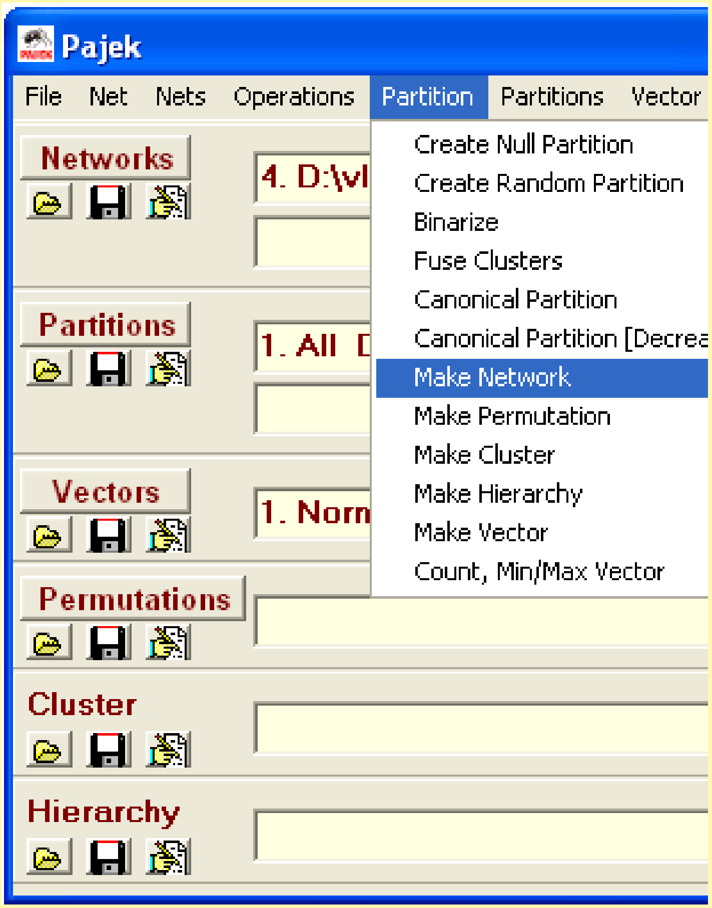
The main goals in the design of **Pajek** are:

- to support abstraction by (recursive) *decomposition* of a large network into several smaller networks that can be treated further using more sophisticated methods;
- to provide the user with some powerful *visualization* tools;
- to implement a selection of efficient *subquadratic* algorithms for analysis of large networks.

With **Pajek** we can: *find* clusters (components, neighbourhoods of 'important' vertices, cores, etc.) in a network, *extract* vertices that belong to the same clusters and *show* them separately, possibly with the parts of the context (detailed local view), *shrink* vertices in clusters and show relations among clusters (global view).

Pajek's data types

In **Pajek** analysis and visualization are performed using 6 data types:



- *network* (graph),
- *partition* (nominal or ordinal properties of vertices),
- *vector* (numerical properties of vertices),
- *cluster* (subset of vertices),
- *permutation* (reordering of vertices, ordinal properties), and
- *hierarchy* (general tree structure on vertices).

Pajek supports also *multi-relational*, *temporal* and *two-mode* networks.

...`Pajek`'s data types

The power of **Pajek** is based on several transformations that support different transitions among these data structures. Also the menu structure of the main **Pajek**'s window is based on them. **Pajek**'s main window uses a 'calculator' paradigm with list-accumulator for each data type. The operations are performed on the currently active (selected) data and are also returning the results through accumulators.

The procedures are available through the main window menus. Frequently used sequences of operations can be defined as *macro*s. This allows also the adaptations of **Pajek** to groups of users from different areas (social networks, chemistry, genealogy, computer science, mathematics...) for specific tasks. **Pajek** supports also *repetitive operations* on series of networks.

Approaches to large networks

In analysis of a *large* network (several thousands or millions of vertices, the network can be stored in computer memory) we can't display it in its totality; also there are only few algorithms available.

To analyze a large network we can use statistical approach or we can use the described decomposition approach – identify smaller (sub) networks that can be analyzed further using more sophisticated methods.

Clusters, clusterings, partitions, hierarchies

A nonempty subset $C \subseteq \mathcal{V}$ is called a *cluster* (group). A nonempty set of clusters $\mathbf{C} = \{C_i\}$ forms a *clustering*.

Clustering $\mathbf{C} = \{C_i\}$ is a *partition* iff

$$\cup\mathbf{C} = \bigcup_i C_i = \mathcal{V} \quad \text{in} \quad i \neq j \Rightarrow C_i \cap C_j = \emptyset$$

Clustering $\mathbf{C} = \{C_i\}$ is a *hierarchy* iff

$$C_i \cap C_j \in \{\emptyset, C_i, C_j\}$$

Hierarchy $\mathbf{C} = \{C_i\}$ is *complete*, iff $\cup\mathbf{C} = \mathcal{V}$; and is *basic* if for all $v \in \cup\mathbf{C}$ also $\{v\} \in \mathbf{C}$.

Example: Snyder and Kick World Trade

The data are available as a **`Pajek`**'s project file

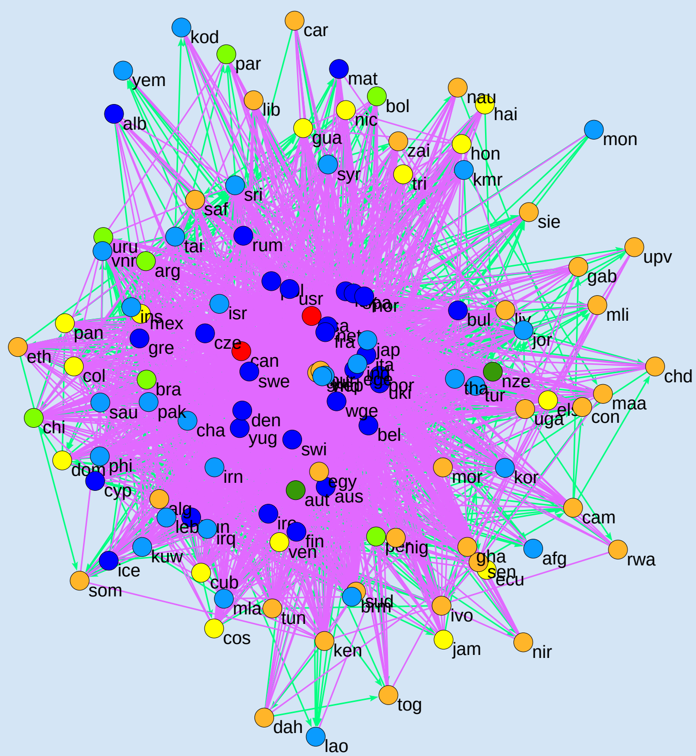
`SaKtrade.paj`

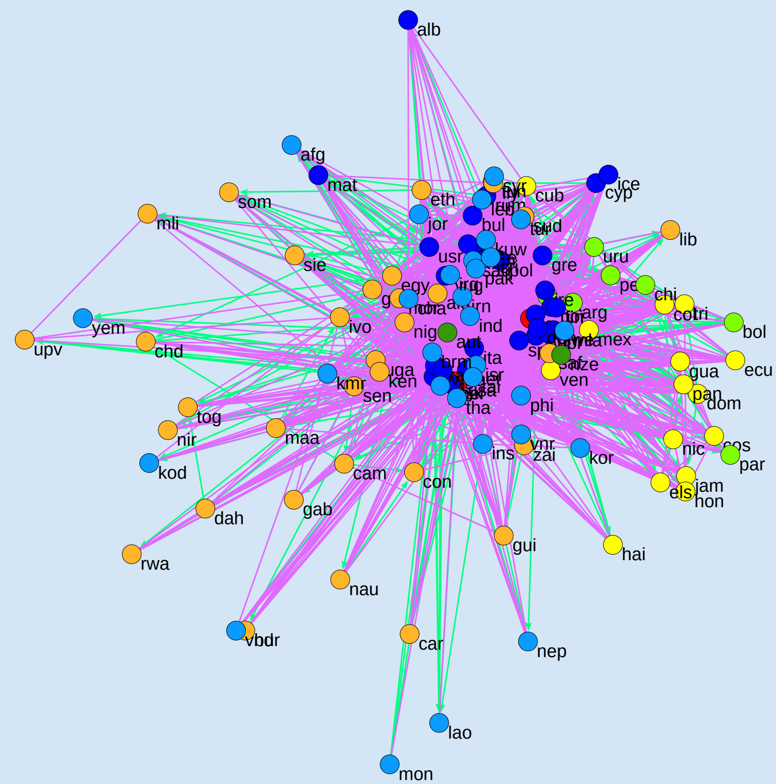
The network consists of trade relations (118 vertices, 515 arcs, 2116 edges). The source of the data is the paper: Snyder, David and Edward Kick (1979). *The World System and World Trade: An Empirical Exploration of Conceptual Conflicts*, Sociological Quaterly, 20,1, 23-36.

The project file contains also the (sub)continents partition:

1 - Europe, 2 - North America, 3 - Latin America, 4 - South America, 5 - Asia, 6 - Africa, 7 - Oceania.

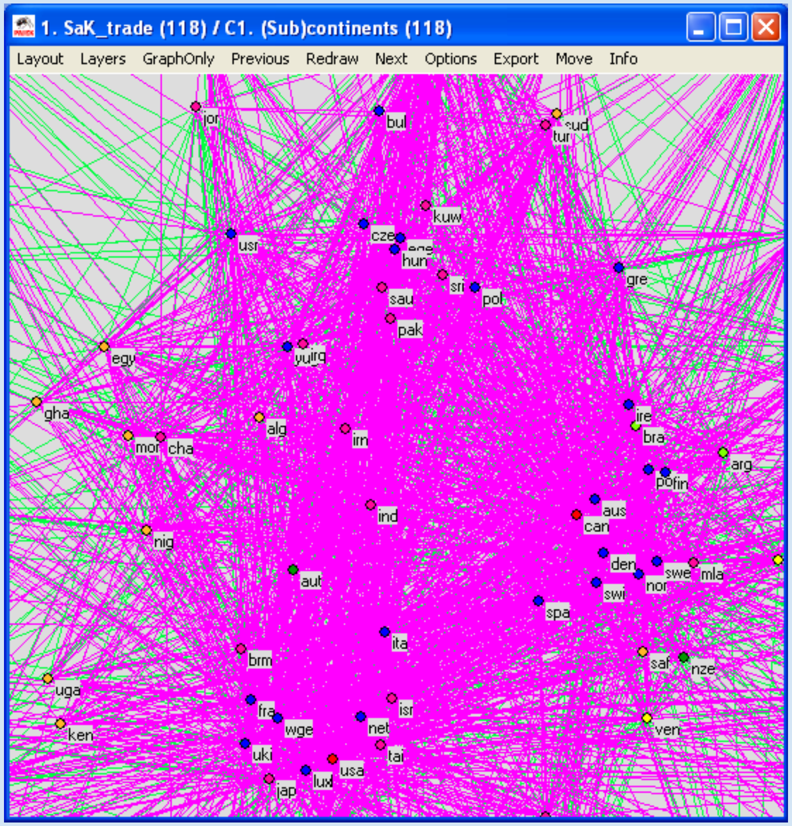
Draw / Partition





```
Draw/Draw Partition
Layout/Energy/Kamada-Kawai/Free
Layout/Energy/Fruchterman Reingold/2D
```

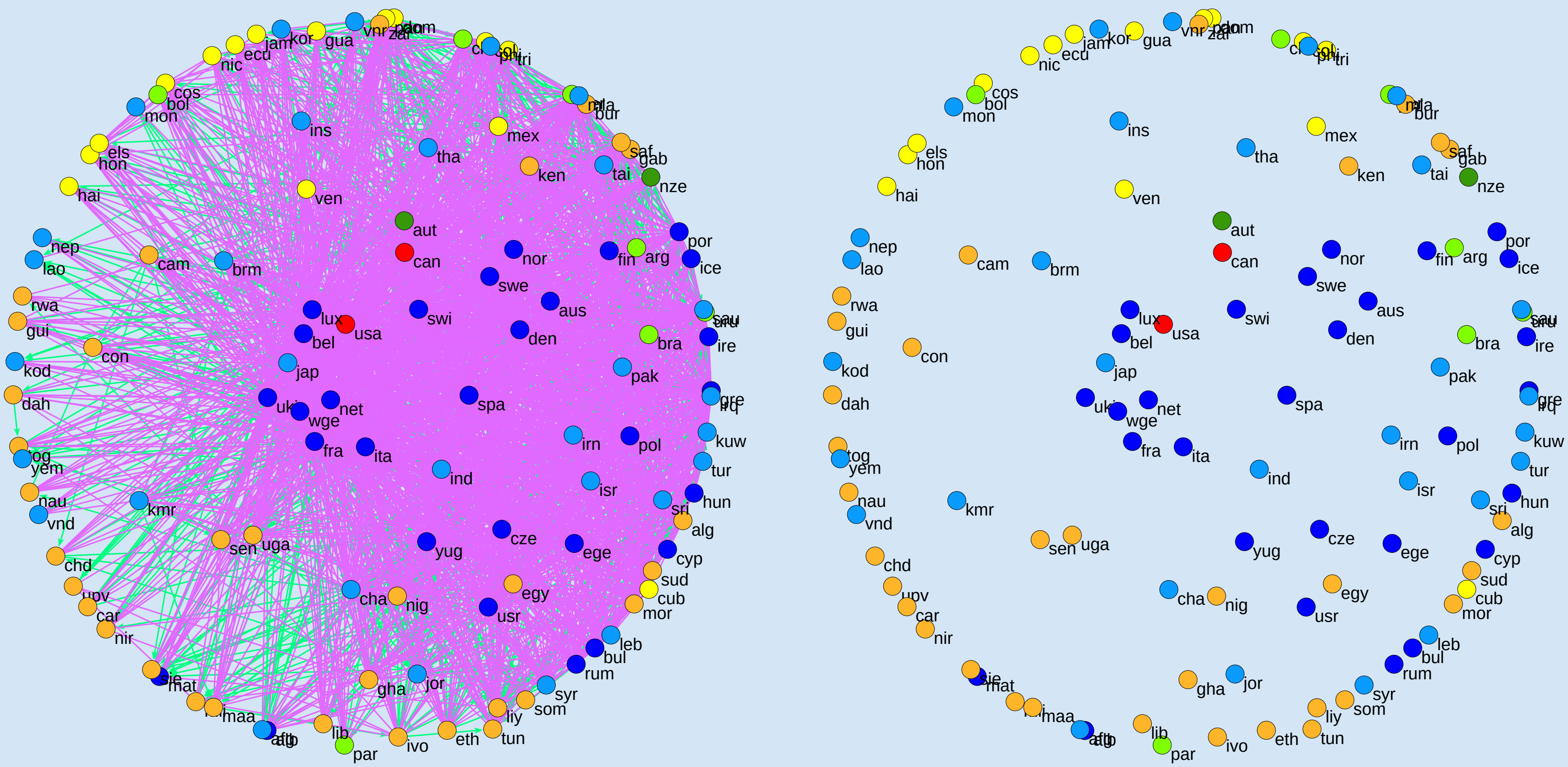
Zoom in



Using right button on the mouse select the zoom area.

To restore the standard view select `Redraw`.

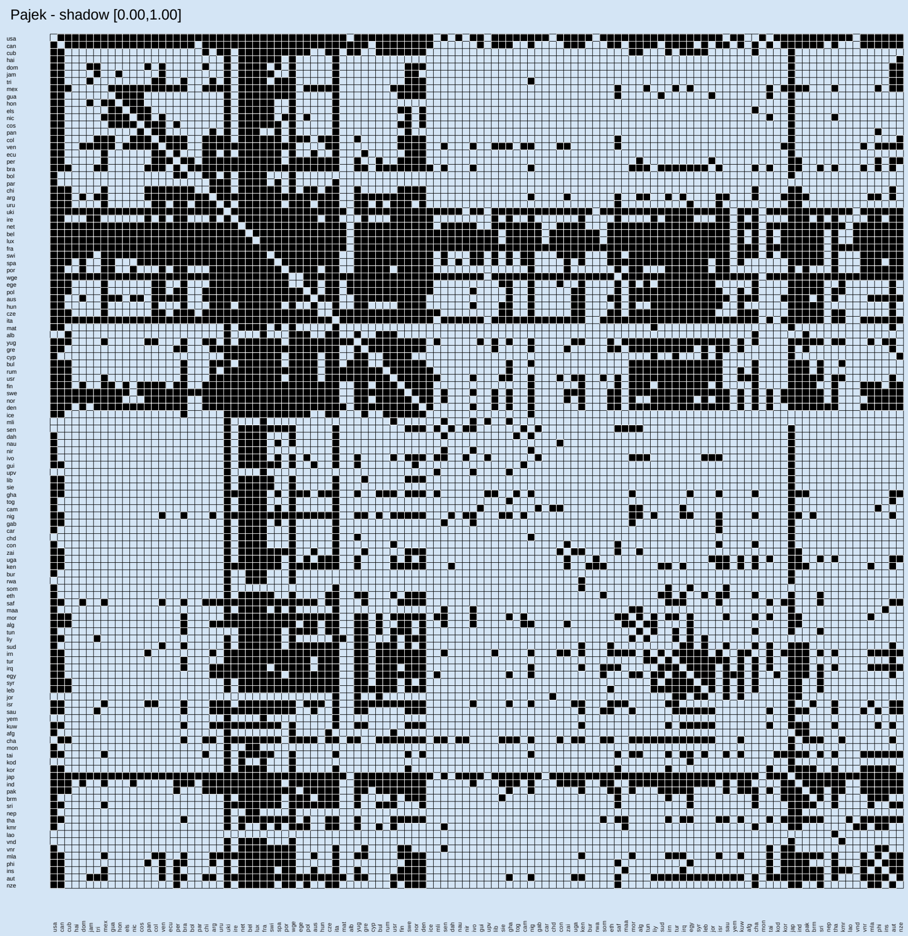
Fruchterman Reingold / factor $= 9$



`Layout/Energy/Fruchterman Reingold/3D`

3D picture / King

Matrix representation




```
Partition/Make Hierarchy [Yes][No]
Hierarchy/Make Permutation
File/Network/Export Matrix to EPS/Using Permutation [SaKmatrix.EPS][No]
GsView: Media/User Defined ... [1300 pt][1300 pt]
```

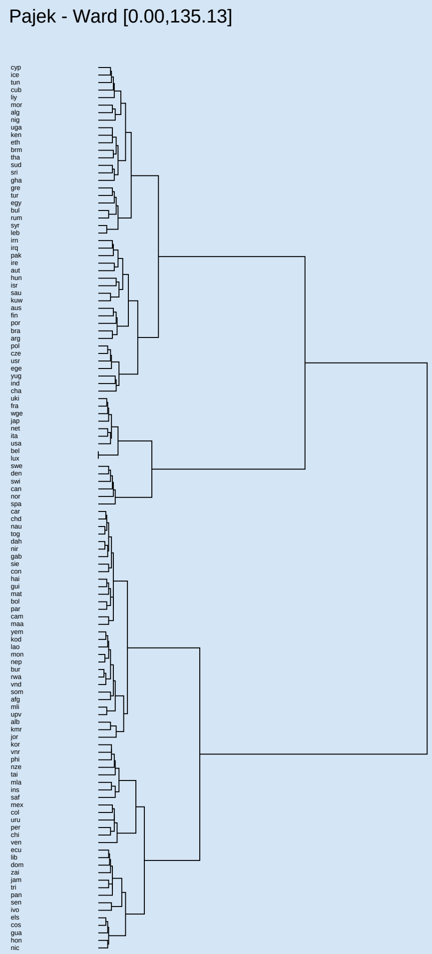
Clustering

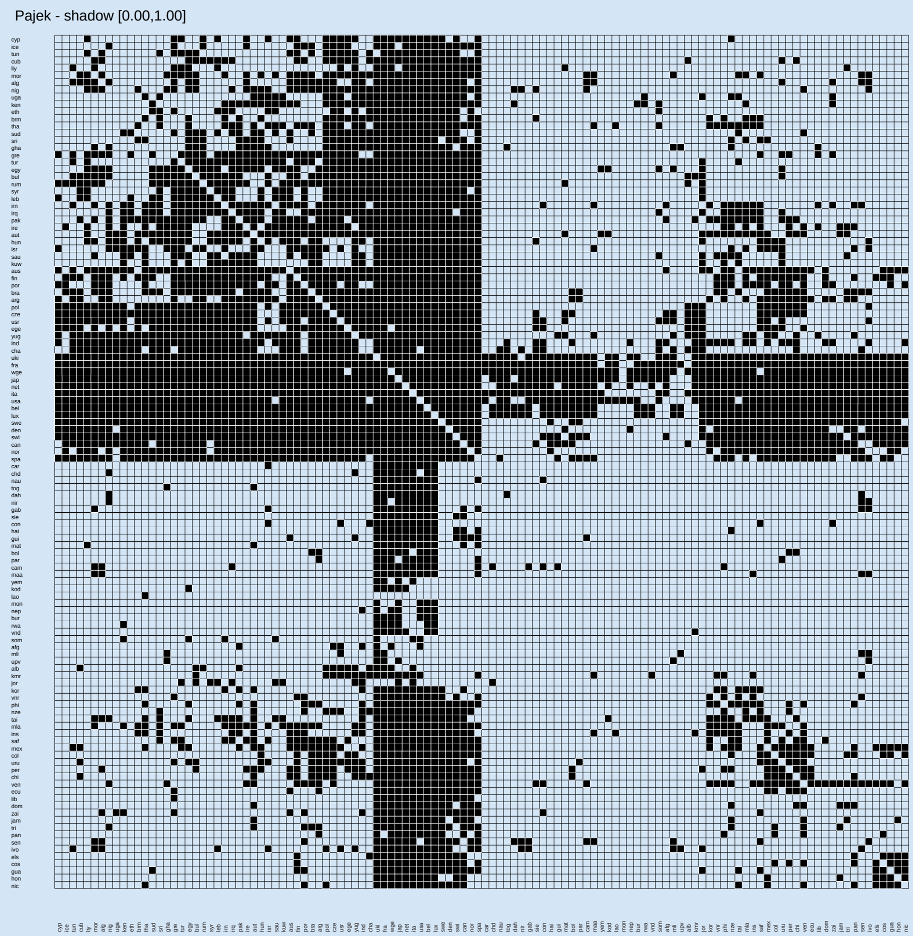
Better network matrix reorderings can be obtained using clustering:

```
Cluster/Create Complete Cluster [118]
Operations/Dissimilarity*/d5 [1][SaKdendro.EPS]
Hierarchy/Make Permutation
[select network SaK.net]
File/Network/Export Matrix to EPS/Using Permutation [SaKmatrix.EPS][No]
```

or blockmodeling.

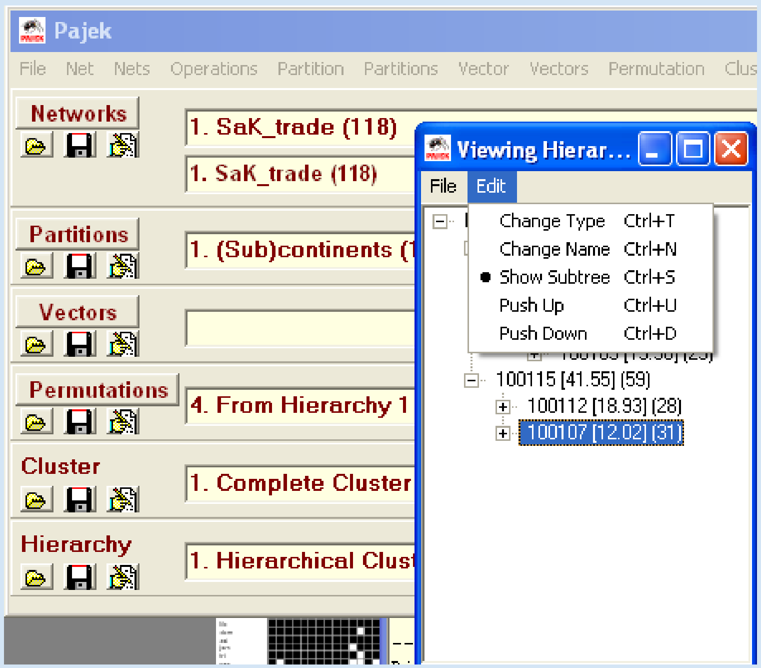
Clustering

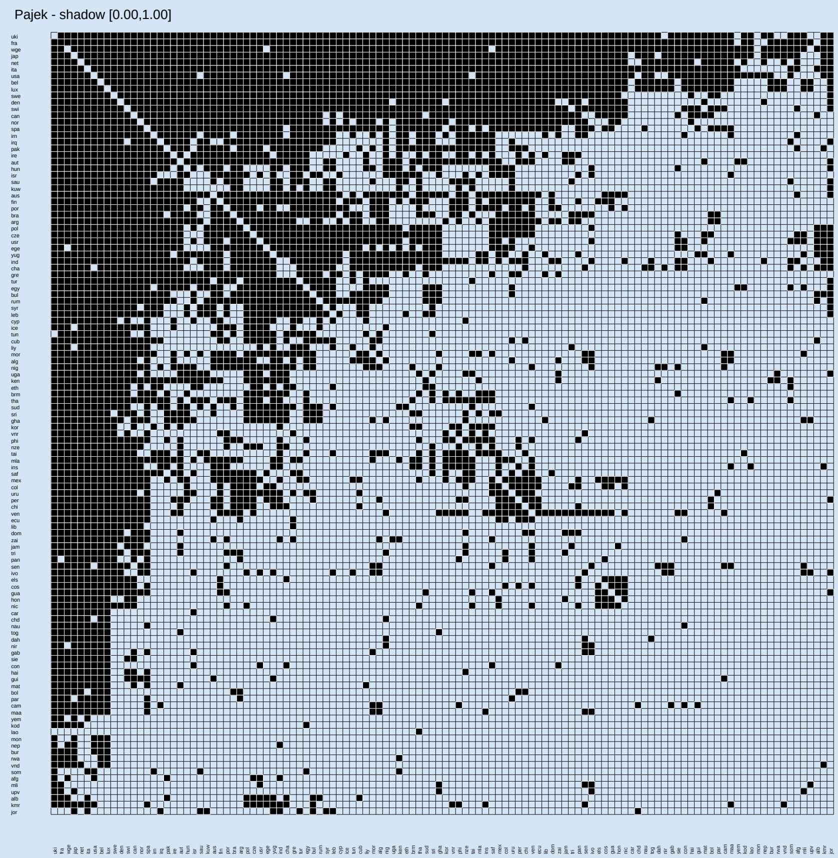




Reordering clustering

The order of clusters in a hierarchy is not fixed and can be changed.





We see the typical center–periphery structure.

Contraction of cluster

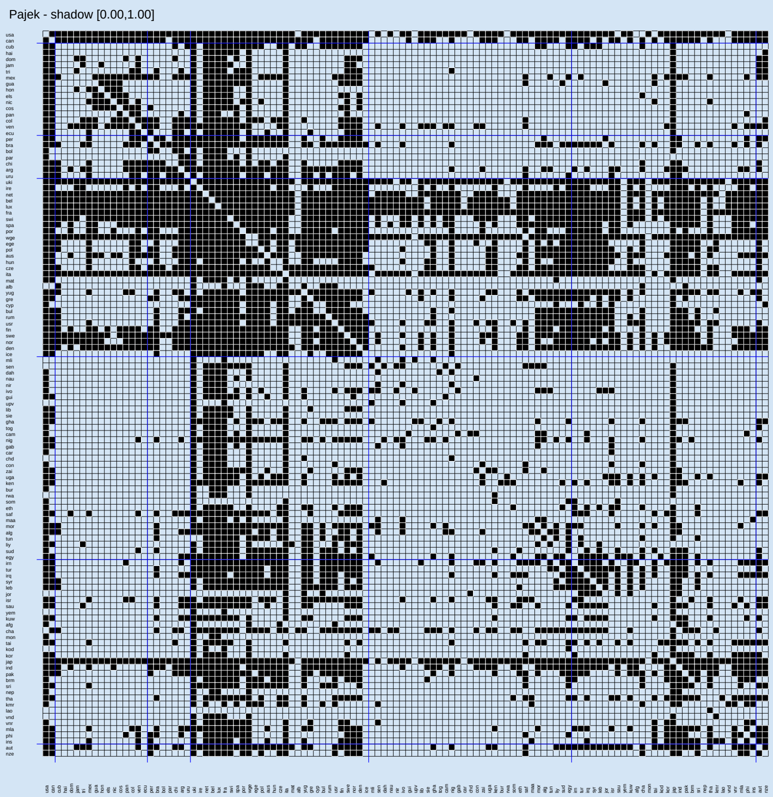
Contraction of cluster C is called a graph $\mathcal{G}/C$, in which all vertices of the cluster C are replaced by a single vertex, say c. More precisely:

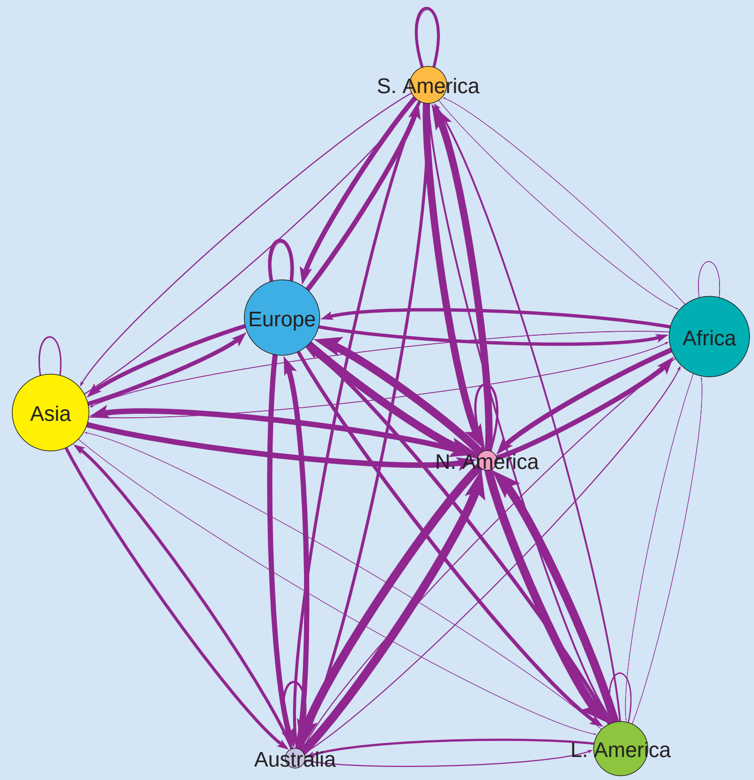
$\mathcal{G}/C = (\mathcal{V}', \mathcal{L}')$, where $\mathcal{V}' = (\mathcal{V} \setminus C) \cup \{c\}$ and $\mathcal{L}'$ consists of lines from $\mathcal{L}$ that have both end-vertices in $\mathcal{V} \setminus C$. Beside these it contains also a 'star' with the center c and: arc (v, c), if $\exists p \in \mathcal{L}, u \in C : p(v, u)$; or arc (c, v), if $\exists p \in \mathcal{L}, u \in C : p(u, v)$. There is a loop (c, c) in c if $\exists p \in \mathcal{L}, u, v \in C : p(u, v)$.

In a network over graph $\mathcal{G}$ we have also to specify how are determined the values/weights in the shrunk part of the network. Usually as the sum or maksimum/minimum of the original values.

`Operations/Shrink Network/Partition`

Contracted clusters – international trade





Snyder and Kick's international trade. Matrix display of dense networks.

$$w(C_i, C_j) = \frac{n(C_i, C_j)}{n(C_i) \cdot n(C_j)}$$

Computing the weights

```
File / Pajek Project File / Read [SaKtrade.paj]
Net / Transform / Remove / Loops [No]
Net / Transform / Edges -> Arcs [No]
Operations / Shrink Network / Partition [1][0]
```

		1	2	3	4	5	6	7
#usa	1.	2	30	13	56	42	45	4
#cub	2.	30	74	25	196	20	37	12
#per	3.	12	28	33	124	16	36	5
#uki	4.	55	217	130	695	427	483	41
#mli	5.	42	8	14	406	122	117	11
#irn	6.	43	37	43	444	142	307	30
#aut	7.	4	4	5	39	9	30	2

```
Partition / Make Permutation
[select partition (Sub)continents]
Operations / Functional Composition / Partition*Permutation
Partition / Count
Partition / Make Vector
Operations / Vector / Put loops
```

...Computing the weights

		1	2	3	4	5	6	7
#usa	1.	4	30	13	56	42	45	4
#cub	2.	30	89	25	196	20	37	12
#per	3.	12	28	40	124	16	36	5
#uki	4.	55	217	130	723	427	483	41
#mli	5.	42	8	14	406	155	117	11
#irn	6.	43	37	43	444	142	337	30
#aut	7.	4	4	5	39	9	30	4
count		2	15	7	29	33	30	2

```
Vector / Create Identity Vector [7]
[select as second vector From partition ...]
Vectors / Divide First by Second
Operations / Vector / Vector # Network / input
Operations / Vector / Vector # Network / output
[edit partition - rename vertices]
```

		1	2	3	4	5	6	7
N.Am	1.	1.00	1.00	0.93	0.97	0.64	0.75	1.00
L.Am	2.	1.00	0.40	0.24	0.45	0.04	0.08	0.40
S.Am	3.	0.86	0.27	0.82	0.61	0.07	0.17	0.36
Euro	4.	0.95	0.50	0.64	0.86	0.45	0.56	0.71
Afri	5.	0.64	0.02	0.06	0.42	0.14	0.12	0.17
Asia	6.	0.72	0.08	0.20	0.51	0.14	0.37	0.50
Ocea	7.	1.00	0.13	0.36	0.67	0.14	0.50	1.00

In **`Pajek`** sequences of commands can be combined into a macro command using

`Macro / Record`

and

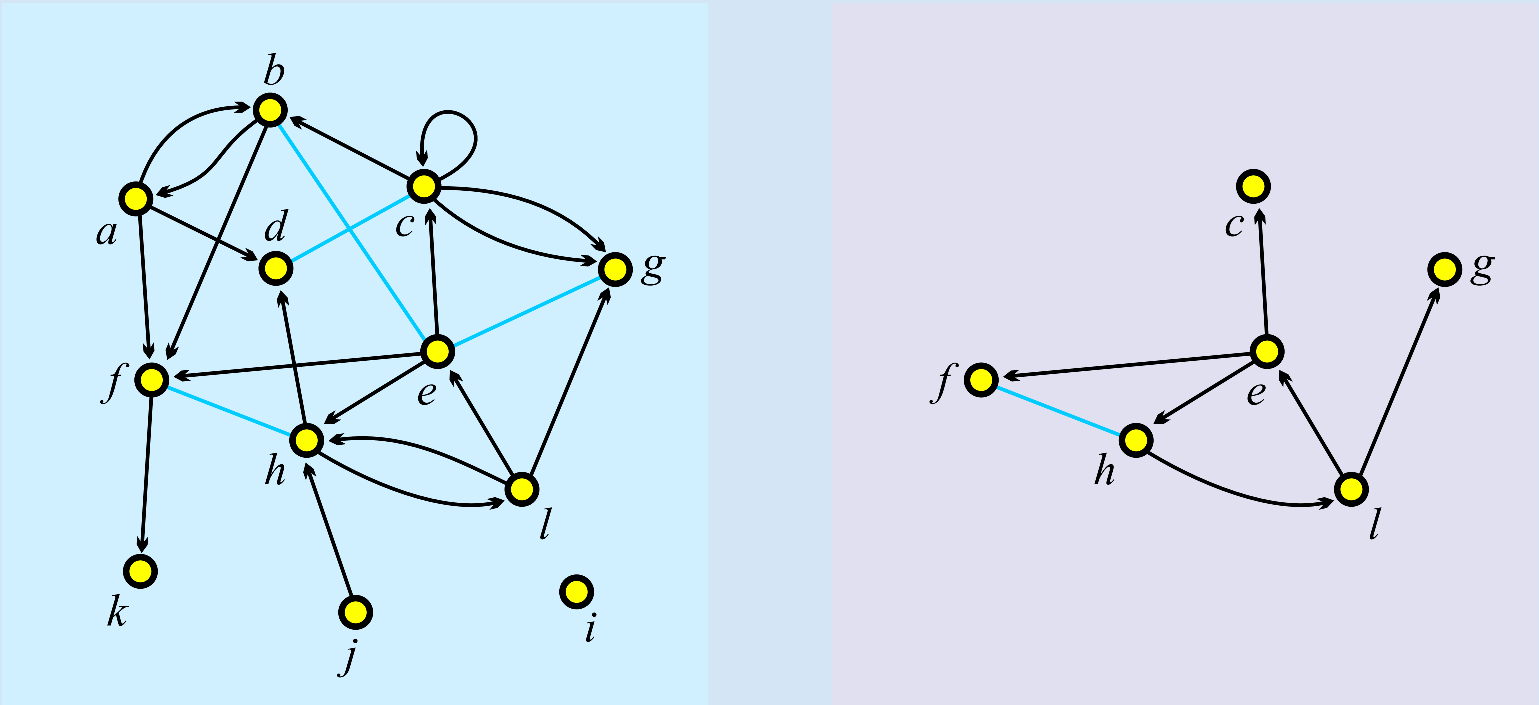
`Macro / Recording...`

The macro can be activated by

`Macro / Play`

The sequence for computing the weights $w(C_i, C_j)$ is saved in the macro `weights`.

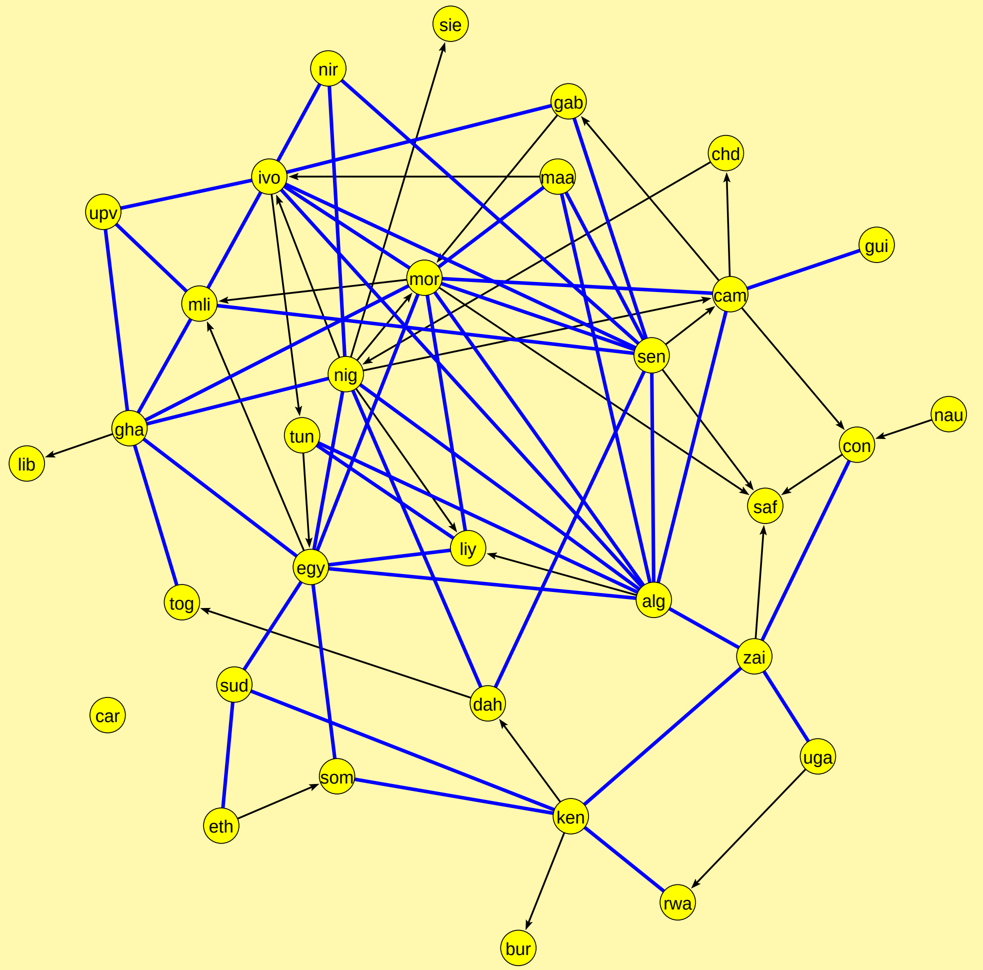
Subgraph



A *subgraph* $\mathcal{H} = (\mathcal{V}', \mathcal{L}')$ of a given graph $\mathcal{G} = (\mathcal{V}, \mathcal{L})$ is a graph which set of lines is a subset of set of lines of $\mathcal{G}$, $\mathcal{L}' \subseteq \mathcal{L}$, its vertex set is a subset of set of vertices of $\mathcal{G}$, $\mathcal{V}' \subseteq \mathcal{V}$, and it contains all end-vertices of $\mathcal{L}'$.

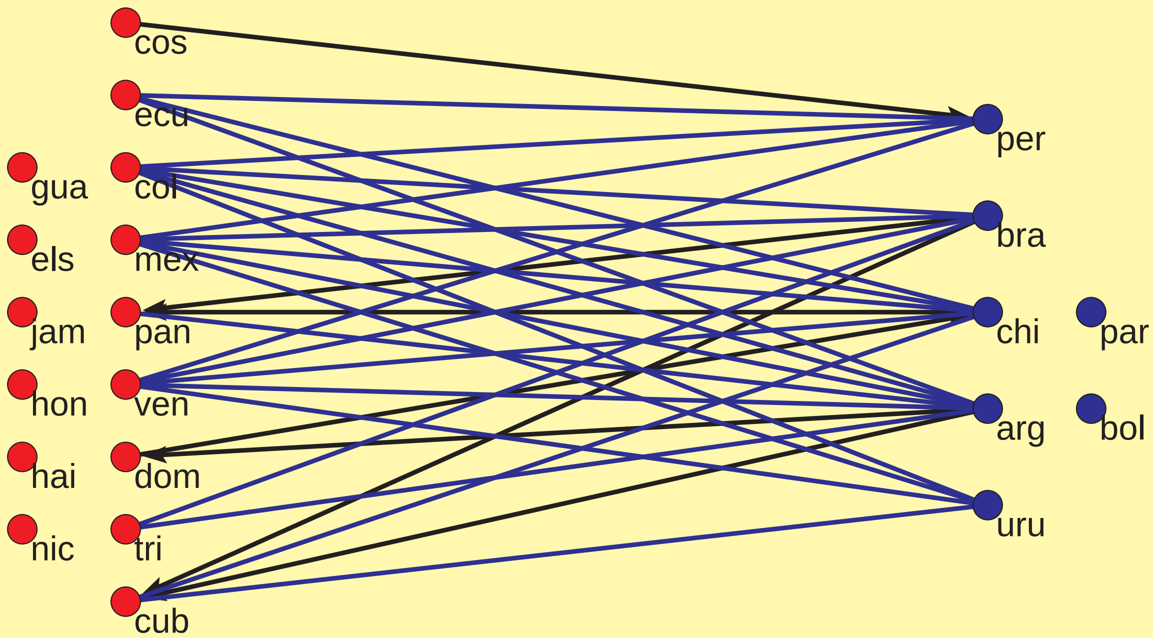
A subgraph can be *induced* by a given subset of vertices or lines. It is a *spanning* subgraph iff $\mathcal{V}' = \mathcal{V}$.

Cut-out – induced subgraph: Snyder and Kick – Africa



`Operations/Extract from Network/Partition [6]`

Cut-out: Snyder and Kick
Latin America : South America




```
Operations/Extract from Network/Partition  [3,4]
Operations/Transform/Remove lines/Inside clusters   [3,4]
```

The vertices can be manually put on a rectangular grid produced by

```
[Draw] Move/Grid
```