How to
Analyze Large Networks with Pajek

Vladimir Batagelj
Andrej Mrvar

University of Ljubljana
Slovenia

SUNBELT XXIII, International Social Network Conference

Westin Regina Resort Hotel, Cancún, Quintana Roo, México
February 12-16, 2003

Outline

Large Networks

Large network – several thousands or millions of vertices.

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
Actors	3.9M	392400	1342595	Notre Dame Networks

Two main approaches: **statistics** and **decomposition**.

Statistics

Input data

- numeric → **`vector`**
- ordinal → **`permutation`**
- nominal → **`clustering`**

Computed properties

global: number of vertices, edges/arcs, components; maximum core number, . . .

local: degrees, cores, indices (betweeness, hubs, authorities, . . .)

inspections: partition, vector, values of lines, . . .

Associations between computed (structural) data and input (measured) data.

. . . Statistics

The global computed properties are reported by Pajek’s commands or can be seen using the **`Info`** option.

The local properties are computed by Pajek’s commands and stored in vectors or partitions. To get information about their distribution use the **`Info`** option.

As an example, let us look at WWW network

```
File/Network/Read  WWW.net
Info/Network/General
```

It has 325729 vertices and 1497135 arcs (27455 loops).

… Statistics

```
Net/Partitions/Degree/Input
Info/Partition  +10 | 100
```

The largest input degrees have the vertices: $indeg(12129) = 10721$, $indeg(325729) = 7619$, $indeg(124802) = 7026$, $indeg(31331) = 4300$, …

There are 195777 vertices of indegree 1, 45331 vertices of indegree 2, …

Statistics / Pajek and R

Pajek 0.89 (and higher) supports interaction with statistical program R and the use of other programs as tools.

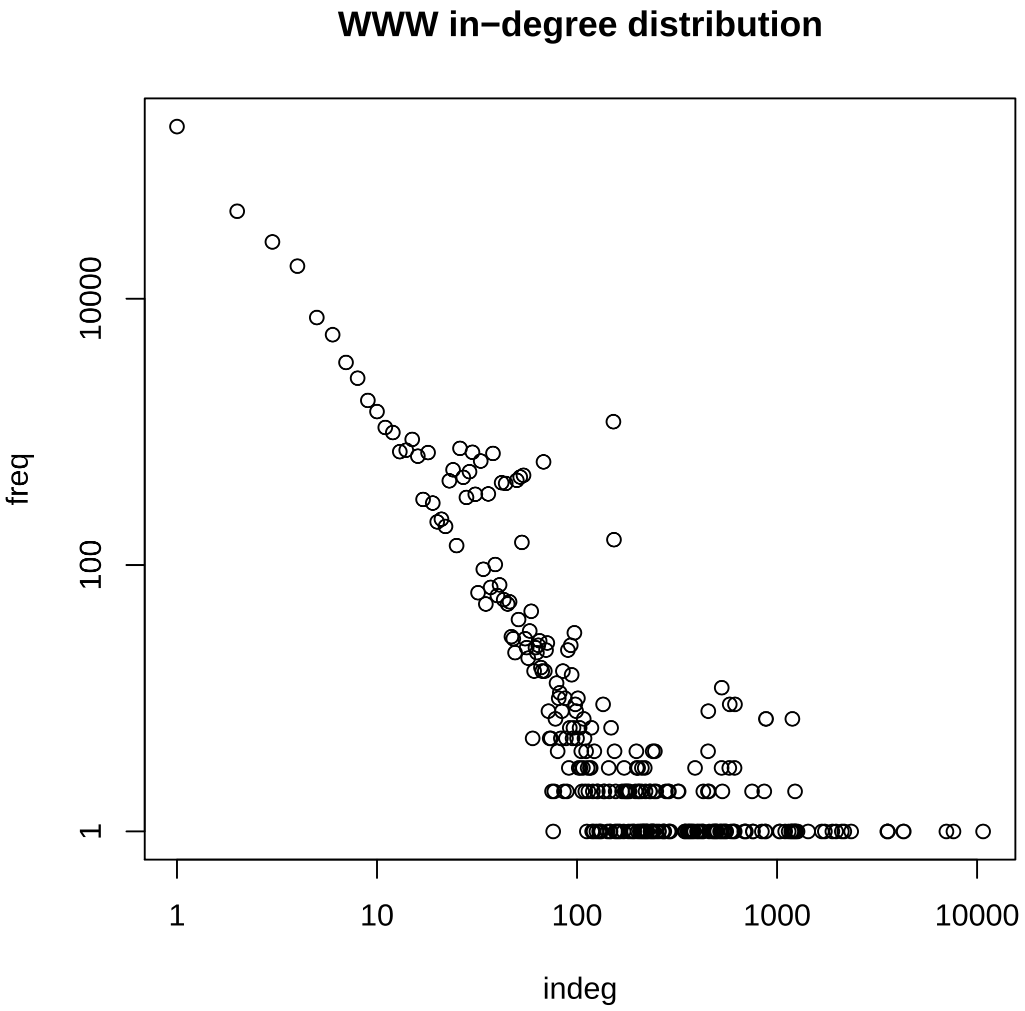
```
Partition/Make Vector
Tools/Program R/Send to R/Current Vector
```

We send the vector of in-degrees to R and draw their distribution

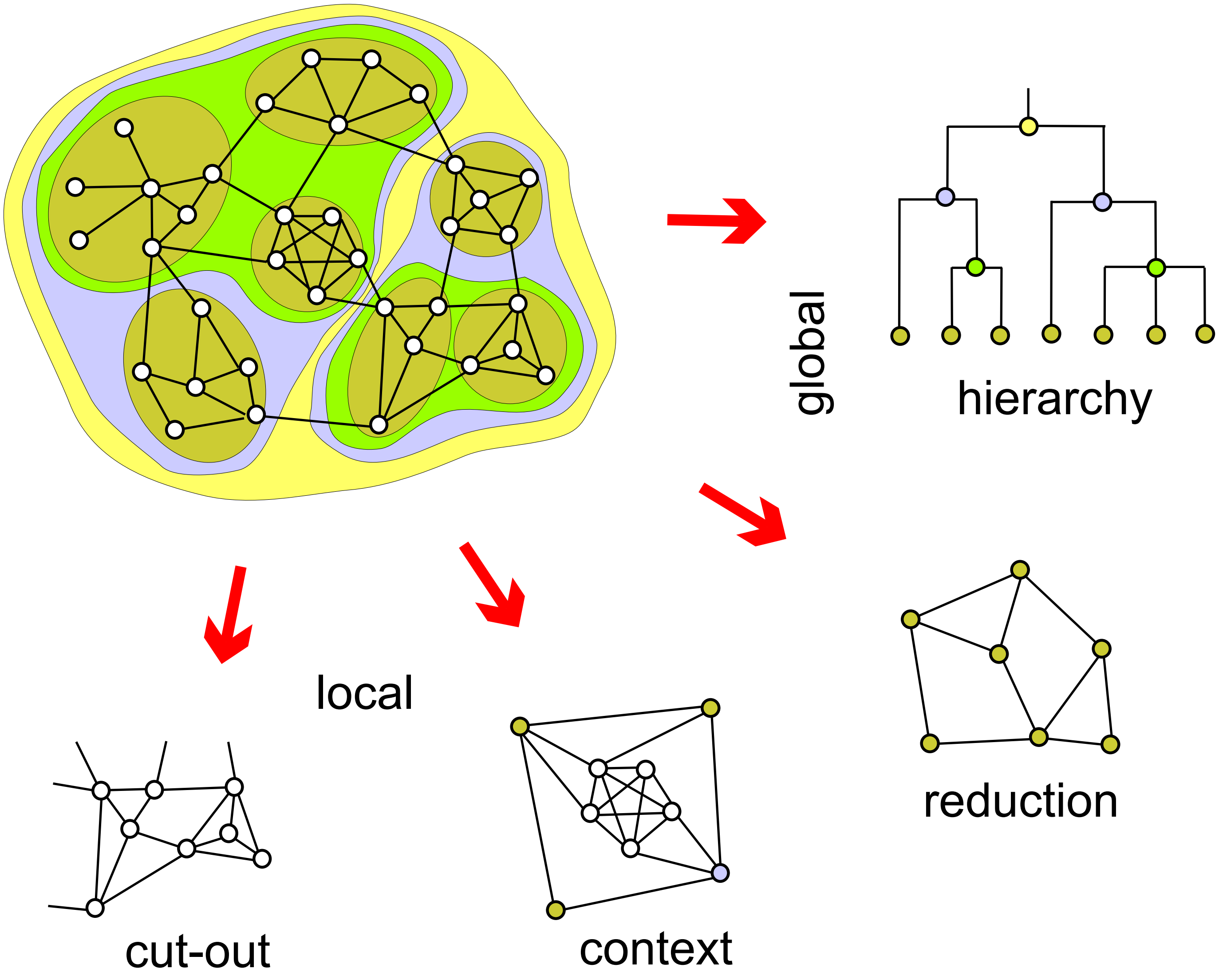
```
summary(v2)
t <- tabulate(v2)
c <- t[t>0]
i <- (1:length(t))[t>0]
pdf(file='indeg.pdf')
plot(i,c,log='xy',main='WWW in-degree distribution',
   xlab='indeg',ylab='freq')
dev.off()
```

To obtain the picture in Windows metafile format (for inclusion in Word) replace the `pdf` command with `win.metafile(file='indeg.wmf')`.

WWW in-degree distribution



Decomposition



global

hierarchy

reduction

local

cut-out

context

Line-Cuts

If we *line-cut* a network $\mathbf{N} = (V, L, w)$ at selected level t

$$L' = \{e \in L : w(e) \geq t\}$$

we get a subnetwork $\mathbf{N}(t) = (V(L'), L', w)$, $V(L')$ is the set of all endpoints of the lines from L'.

We then look at the components of $\mathbf{N}(t)$. Their number and sizes depend on t. Usually there are many small components. Often we consider only components of size at least k.

The values of thresholds t and k are determined by inspecting the distribution of weights and the distribution of component sizes.

Vertex-Cuts

In some networks we can have also a function $p : V \rightarrow \mathbb{R}$ that describes some property of vertices. Its values can be obtained by measuring, or they are computed (for example, centrality indices).

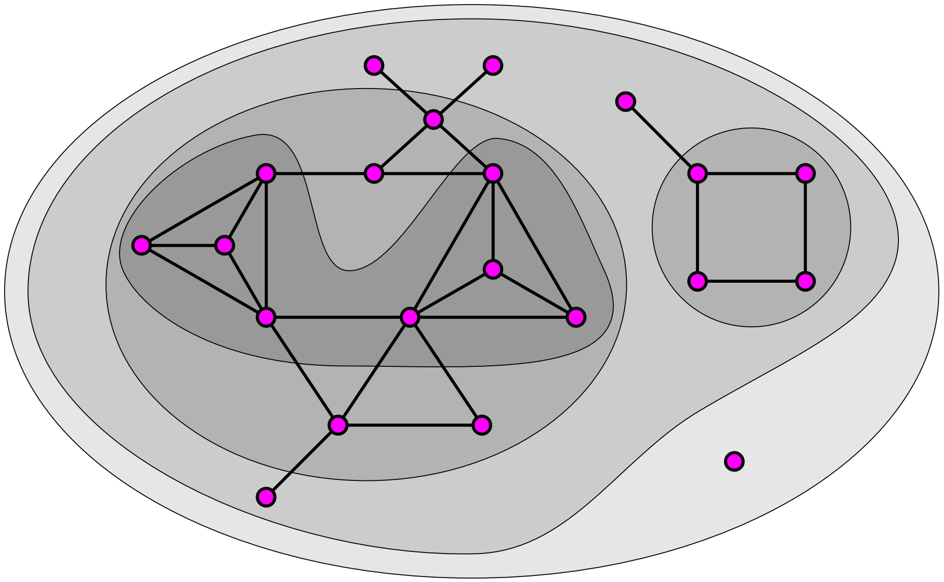
The *vertex-cut* of a network $\mathbf{N} = (V, L, p)$ at selected level t is a network $\mathbf{N}(t) = (V', L(V'), p)$, determined by the set

$$V' = \{v \in V : p(v) \geq t\}$$

and $L(V')$ is the set of lines from L that have both endpoints in V'.

Cores

The notion of core was introduced by Seidman in 1983.



Let $\mathbf{G} = (V, L)$ be a graph. V is the set of *vertices* and L is the set of *lines* (*edges* or *arcs*). A subgraph $\mathbf{H} = (W, L|W)$ induced by the set W is a k-*core* or a *core of order* k iff $\forall v \in W : \deg_H(v) \geq k$, and $\mathbf{H}$ is a maximum subgraph with this property.

…Cores

The core of maximum order is also called the *main* core. The *core number* of vertex v is the highest order of a core that contains this vertex. The degree $\deg(v)$ can be: in-degree, out-degree, in-degree + out-degree, …, determining different types of cores.

- The cores are nested: $i < j \implies \mathbf{H}_j \subseteq \mathbf{H}_i$
- Cores are not necessarily connected subgraphs.

The notion of cores can be generalized to *valued cores*.

Example

Let us determine the core of the WWW network

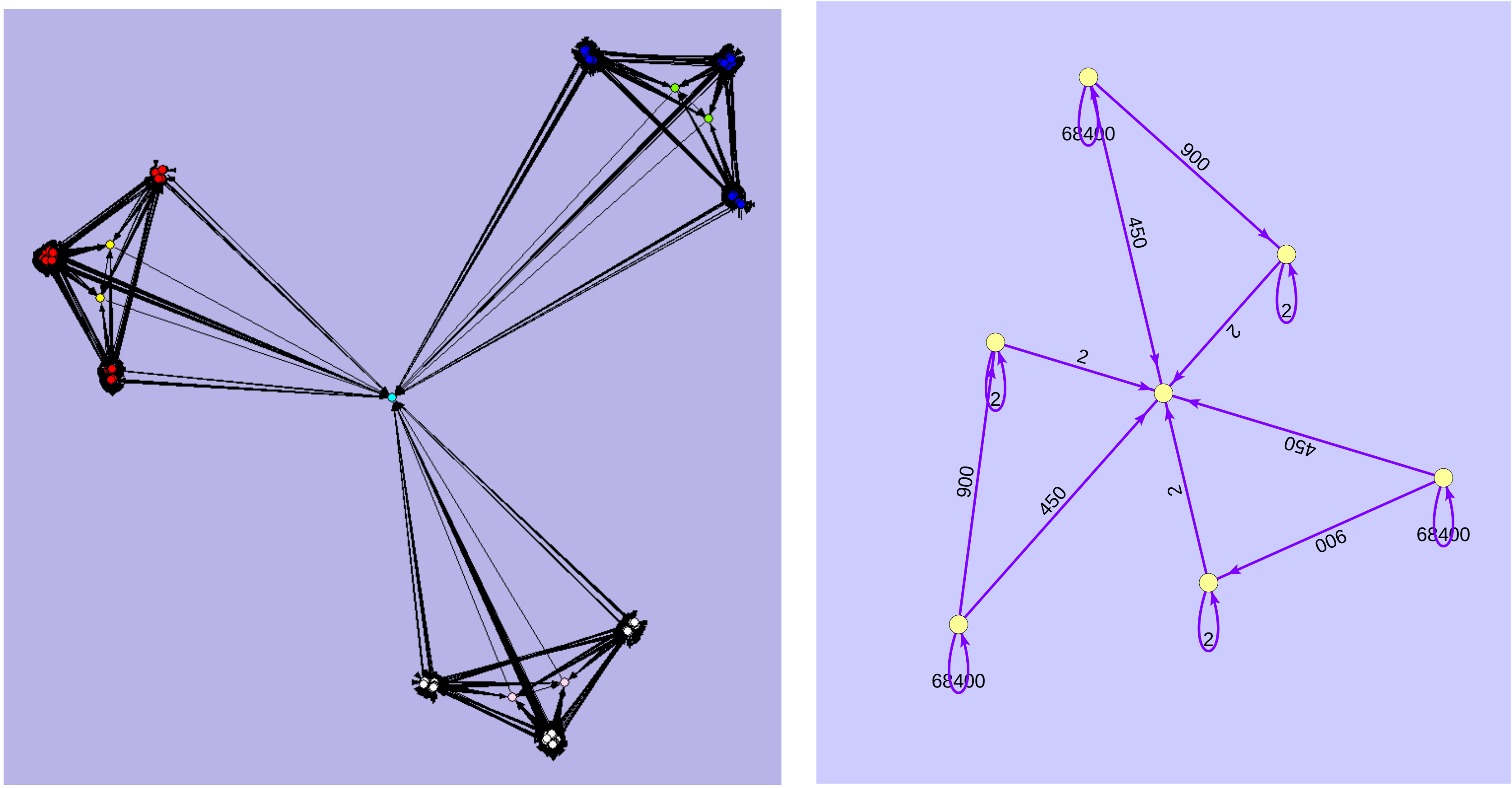
```
Net/Partitions/Core/All
Info/Partition
Operations/Extract from Network/Partition 307 | 307
Net/Components/Strong 2
Draw
Layout/Energy/Fruchterman Rheinhold/2D
```

The main core is of order 307 and has 1357 vertices. We draw it. The obtained picture is pretty regular. This is confirmed by the shrunk network.

```
Operations/Shrink Network/Partition 1 | 0
Draw
Move/Circles 6 | 12
Options/Lines/Mark lines/with values
```

Additional decomposition shows that each ’triangle’ has the form $K_{150} \times K_3$.

Example/Pictures



Main core and shrunk main core.

Citation networks

In a given set of units $\mathbf{U}$ (articles, books, works, $\ldots$) we introduce a *citing* relation $R \subseteq \mathbf{U} \times \mathbf{U}$

$$uRv \equiv v \text{ cites } u$$

which determines a *citation network* $\mathbf{N} = (\mathbf{U}, R)$.

A citing relation is usually *irreflexive* and (almost) *acyclic*.

Citation weights

An approach to the analysis of citation network is to determine for each unit / arc its *importance* or *weight*. These values are used afterward to determine the essential substructures in the network. Some methods of assigning weights $w : R \to \mathbb{R}_0^+$ to arcs were proposed by Hummon and Doreian (1989):

- *node pair projection count* (NPPC) method: $w_1(u,v) = |R^{\text{inv}^\star}(u)| \cdot |R^\star(v)|$
- *search path link count* (SPLC) method: $w_2(u,v)$ equals the number of ”*all possible search paths through the network emanating from an origin node*” through the arc $(u,v) \in R$.
- *search path node pair* (SPNP) method: $w_3(u,v)$ ”*accounts for all connected vertex pairs along the paths through the arc* $(u,v) \in R$”.

Citation weights algorithm

To compute the SPLC and SPNP weights we introduce a related *search path count* (SPC) method for which the weights $N(u, v)$, uRv count the number of different paths from $\mathrm{Min}\, R$ to $\mathrm{Max}\, R$ through the arc (u, v).

There exists a very efficient (linear in number of arcs) algorithm to determine the citation weights.

We get the SPLC weights by applying the SPC method on the network obtained from a given standardized (added source s and sink t) network by linking the source s by an arc to each nonminimal vertex from **U**; and the SPNP weights by applying the SPC method on the network obtained from the SPLC network by additionally linking by an arc each nonmaximal vertex from **U** to the sink t.

Citation example

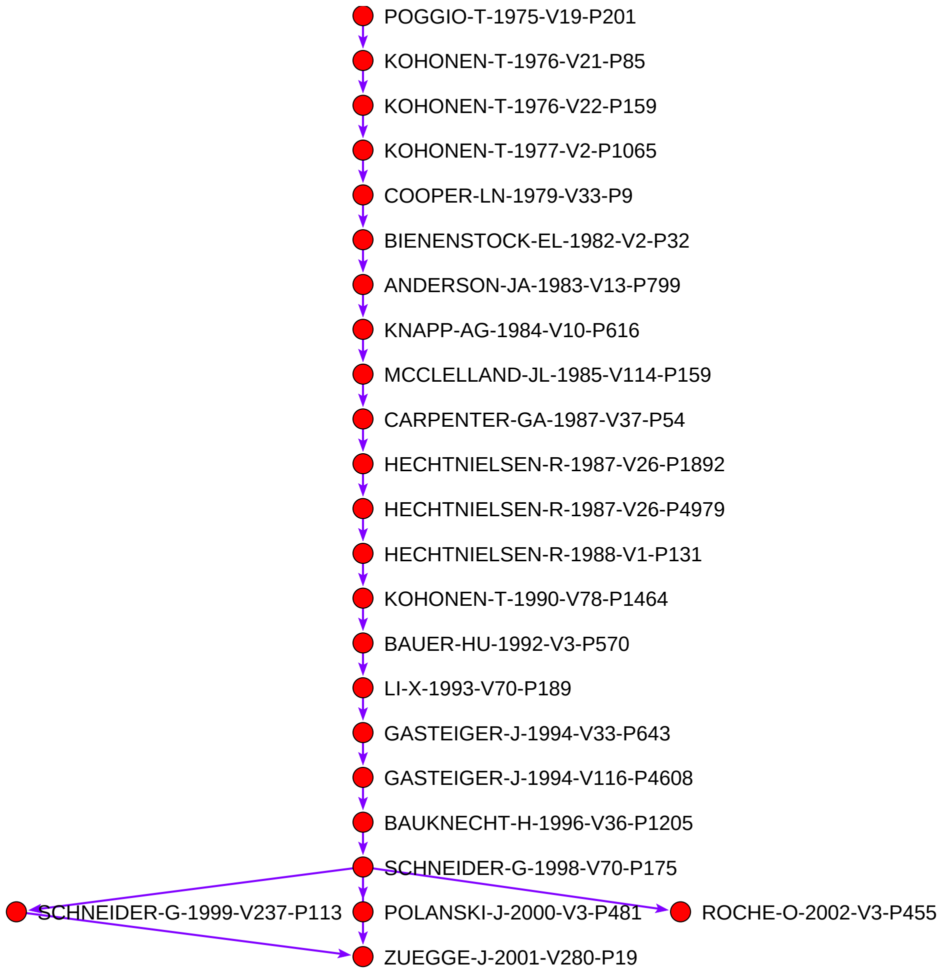
We read the citation network (with additional information) `Kohonen.paj`. First we test the network for acyclicity. Since there are 11 nontrivial strong components we eliminate them by shrinking each component into a single vertex. This operation produces some loops that should be removed.

```
File/Pajek Project File/Read [Kohonen.paj]
Net/Components/Strong [2]
Operations/Shrink Network/Partition [1][0]
Net/Transform/Remove/loops [yes]
Net/Citation Weights/Source-Sink
Macro/Play [ /macro/genea/layers.mcr ]
Draw/Draw-Partition { Main path }
Export/EPS/PS [path.eps]
```

To compute the citation weights we selected the SPC (search path count) method. It returns the following results: the network with citation weights on arcs, the main path network and the vector with vertex weights.

First we draw the main path network using macro `Layers`.

. . . Citation main path


POGGIO-T-1975-V19-P201
KOHONEN-T-1976-V21-P85
KOHONEN-T-1976-V22-P159
KOHONEN-T-1977-V2-P1065
COOPER-LN-1979-V33-P9
BIENENSTOCK-EL-1982-V2-P32
ANDERSON-JA-1983-V13-P799
KNAPP-AG-1984-V10-P616
MCCLELLAND-JL-1985-V114-P159
CARPENTER-GA-1987-V37-P54
HECHTNIELSEN-R-1987-V26-P1892
HECHTNIELSEN-R-1987-V26-P4979
HECHTNIELSEN-R-1988-V1-P131
KOHONEN-T-1990-V78-P1464
BAUER-HU-1992-V3-P570
LI-X-1993-V70-P189
GASTEIGER-J-1994-V33-P643
GASTEIGER-J-1994-V116-P4608
BAUKNECHT-H-1996-V36-P1205
SCHNEIDER-G-1998-V70-P175
SCHNEIDER-G-1999-V237-P113
POLANSKI-J-2000-V3-P481
ROCHE-O-2002-V3-P455
ZUEGGE-J-2001-V280-P19

. . . Citation example

Inspecting the distribution of values of weights on arcs (lines) we select a threshold 0.001 and delete all arcs with weights lower than selected threshold. We delete also all isolated vertices (degree $= 0$) and extract from the vertex weights vector the weights of vertices that remained in the reduced network. We draw the reduced network.

```
Main: [select network: Citation weights]
Info/Network/Line Values [#100][No]
Net/Transform/Remove/lines with value/lower than [0.001][Yes]
Net/Partitions/Degree/All
Operations/Extract from Network/Partition [1][9999]
Vector/Extract Subvector [1][9999]
Macro/Play [ /macro/genea/layers.mcr ]
Draw/Draw-Partition-Vector
Layers/Resolution/High
Layers/Optimize Layers in x direction/Complete [yes][OK][OK]
Options/Mark Vertices Using/No Labels
Options/Lines/Different Widths
Options/Size/of Lines [25]
Options/Size/of Vertices [10]
```

. . . Citation example

We notice some small unimportant components. We preserve only the large main component and the corresponding part of vector of vertex weights. We also restore the layers partition on the reduced network and draw it. We improve the obtained layout manually. We allow only the movement of vertices in x direction.

```
Main: Net/Components/Weak [1]
Info/Partition [1][0]  {1 is the large component}
Operations/Extract from Network/Partition [1][1]
Vector/Extract Subvector [1][1]
Net/Partitions/Depth/Genealogical
Draw/Draw-Partition-Vector
Move/Fix/y
```

Finally we label the 'most important vertices' with their labels. A vertex is considered important if it is an endpoint of an arc with the weight above the selected threshold (in our case 0.05). The arc weights are represented by the thickness of arcs, and the vertex weights by the size of the circles representing vertices.

. . . Citation example

The picture was exported as nested partitions in SVG format. It allows, using Javascript support, interactive display of different *slices* – subnetworks induced by arcs with weights larger than a threshold value.

```
Info/Network  {threshold 0.05 selected}
Net/Transform/Remove/lines with value/lower than [0.05]
Net/Partitions/Degree/All
Partition/Binarize [1][999]
[select the reduced network]
Draw/Draw-Partition-Vector
Options/Mark Vertices Using/Mark Cluster Only
Options/Mark Vertices Using/Labels
Export/SVG/Line Values/Options/GreyScale
Export/SVG/Line Values/Options/Different Widths
Export/SVG/Line Values/Nested Classes  [main.htm][#20]
```

The SVG viewer supports also zooming and moving of the picture – right button click.

Pattern search

If a selected pattern does not occur frequently in a sparse network the standard backtracking algorithm applied for *pattern search*ing finds all appearences of the pattern very fast even in very large networks.

To speed up the search or to consider some additional properties of the pattern, a user can set some additional options:

- vertices in network should match with vertices in pattern in some nominal, ordinal or numerical property (for example, type of atom in molecula)
- values of lines must match (for example, lines representing male/female links in the case of p-graphs)
- the first vertex in the pattern can be selected only from a given subset of vertices in the network.

Applications of pattern search

Pattern searching was successfully applied to searching for patterns of atoms in molecula (carbon rings) and searching for relinking marriages in genealogies.

For counting triads a special procedure is available.

To be extended !!!

Sources

Vladimir Batagelj, Andrej Mrvar: Pajek.
`http://vlado.fmf.uni-lj.si/pub/networks/pajek/`

Vladimir Batagelj: Papers on network analysis.
`http://vlado.fmf.uni-lj.si/pub/networks/doc/`

Stefan Ernst: photo ***Gartenkreuzspinne*** / ***Araneus diadematus*** from
`http://www.naturfoto-online.de`