Social networks visualization

V. Batagelj

On visualization of (social) networks

Vladimir Batagelj

IMFM Ljubljana and IAM UP Koper

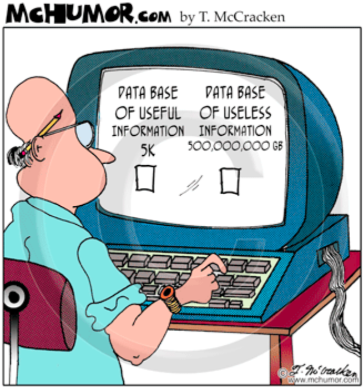
Big Data Visual Analytics
Shonan meeting – November 8-11, 2015

Social networks visualization

V. Batagelj

- Networks
- Approaches
- Statistics
- Drawing of large graphs
- Drawing of dense graphs
- Network decomposition
- Layouts
- Problems
- References

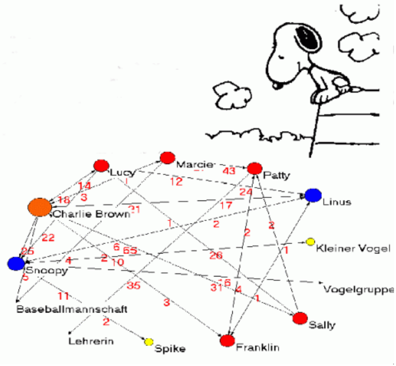
Outline



Vladimir Batagelj:
vladimir.batagelj@fmf.uni-lj.si

Current version of slides (November 8, 2015, 05 : 13):
http://vlado.fmf.uni-lj.si/pub/slides/shonan.pdf

Networks

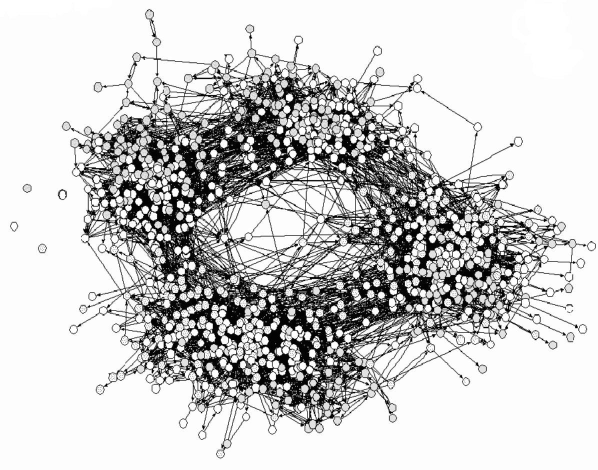


Alexandra Schuler/ Marion Laging-Glaser: Analyse von Snoopy Comics

A *network* $\mathcal{N} = (\mathcal{V}, \mathcal{L}, \mathcal{P}, \mathcal{W})$ is based on two sets – a set of *nodes* (vertices) $\mathcal{V}$, that represent the selected *units*, and a set of *links* (lines) $\mathcal{L}$, that represent *ties* between units. They determine a *graph*. A line can be *directed* – an *arc*, or *undirected* – an *edge*; $\mathcal{L} = \mathcal{E} \cup \mathcal{A}$. Additional data about nodes or links can be known – the *properties* (attributes) $\mathcal{P}$ of nodes and *weights* $\mathcal{W}$ of links.

Network = Graph + Data

"Countryside" school district



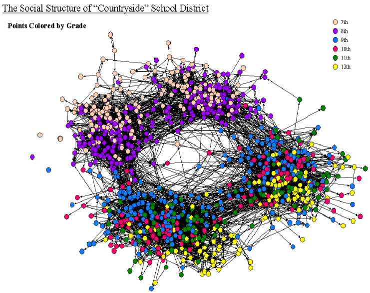
Only small or sparse networks can be displayed readably.

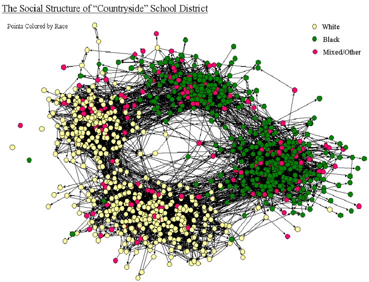
On large networks graph drawing algorithms can reveal their overal structure.

Can we explain the obtained structure?

James Moody (2001) AJS Vol 107, 3,679–716, friendship relation

Display of properties – school (Moody)





To understand networks we need (additional) data!

Approaches to large networks

From algorithmic complexity analysis: for dealing with large data (millions of units) we are limited to *subquadratic* algorithms.

Most of large networks are *sparse* – Dunbar's number.

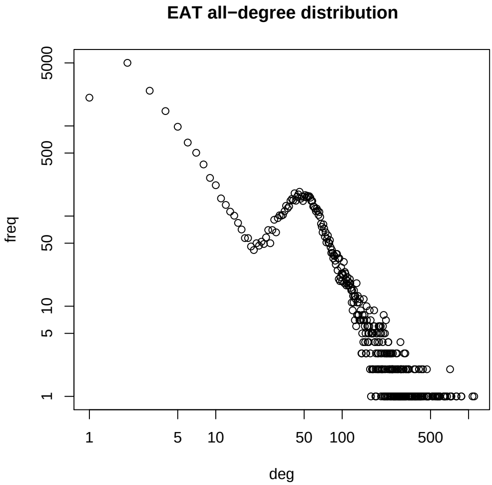
Approaches to large networks analysis and visualization:

- statistics;
- important parts:
  - decomposition;
  - identification of important elements and substructures.

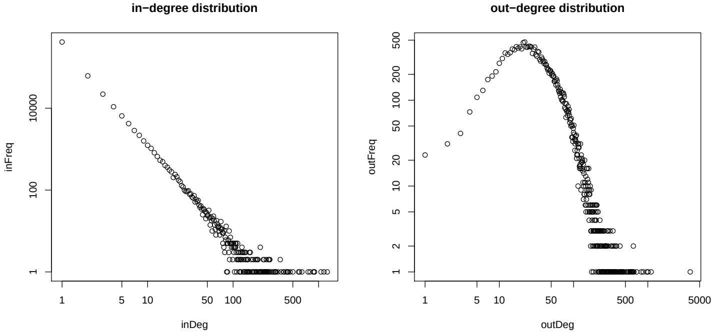
Visualization: initial network exploration, reporting results, story telling.

For additional details see [4, 3, 5].

EAT all-degree distribution



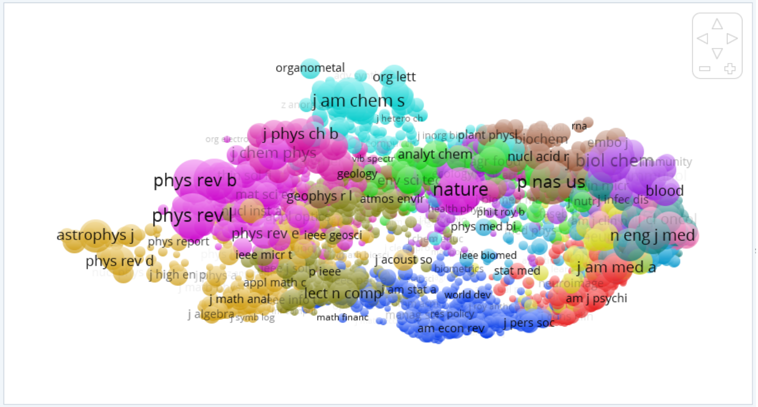
in/out-degree distributions



The in-degree distribution is "scale-free"-like. The parameters can be determined using the package of Clauset, Shalizi and Newman. See also Stumpf, et al.: Critical Truths About Power Laws.

It is important to consider the direction of links!

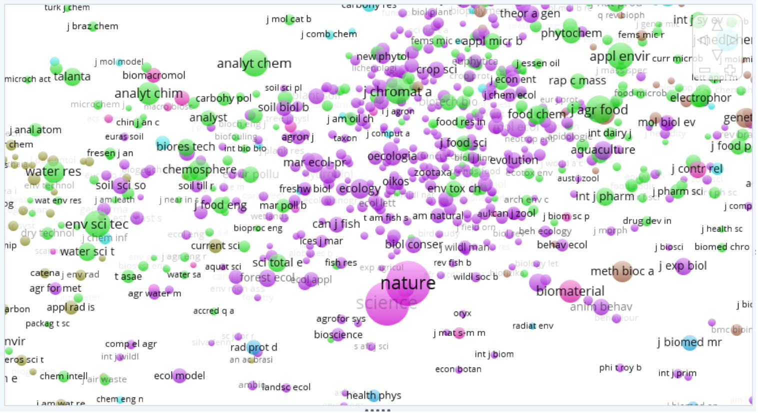
VOSviewer / journals



links → projection of nodes

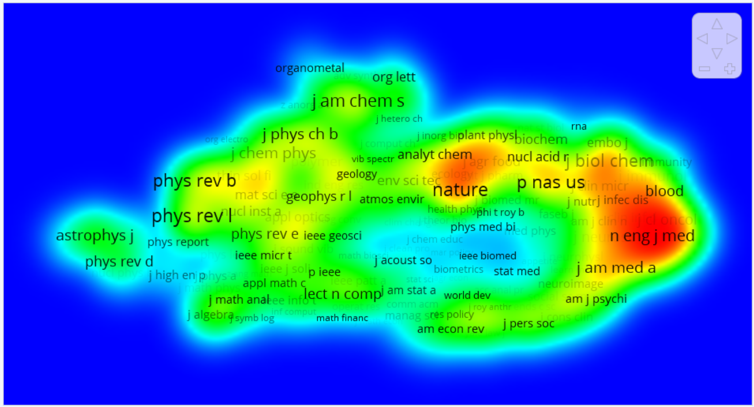
There are many algorithms for drawing large graphs and networks [11] (Brandes and Pich – MDS [8]; VOSviewer - level of detail, zooming, contours [17]).

VOSviewer



zooming

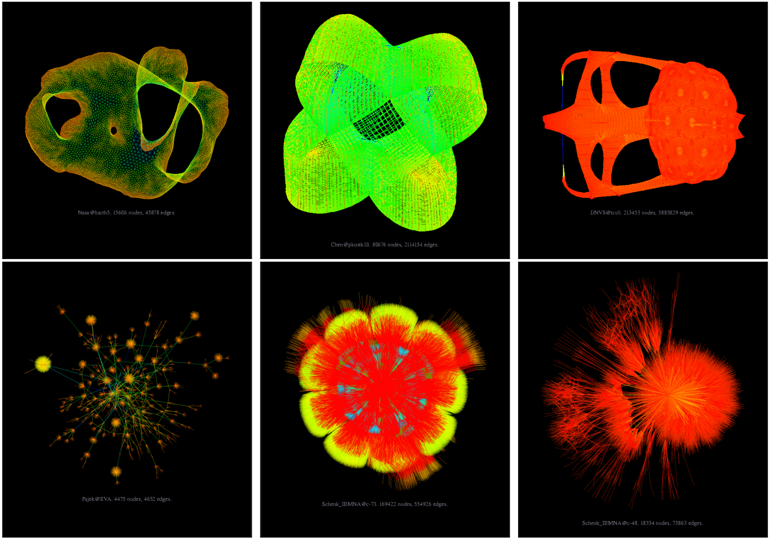
VOSviewer



nodes → density of nodes → contours

Examples from the Gallery of Large Graphs

Yifan Hu



Dense networks

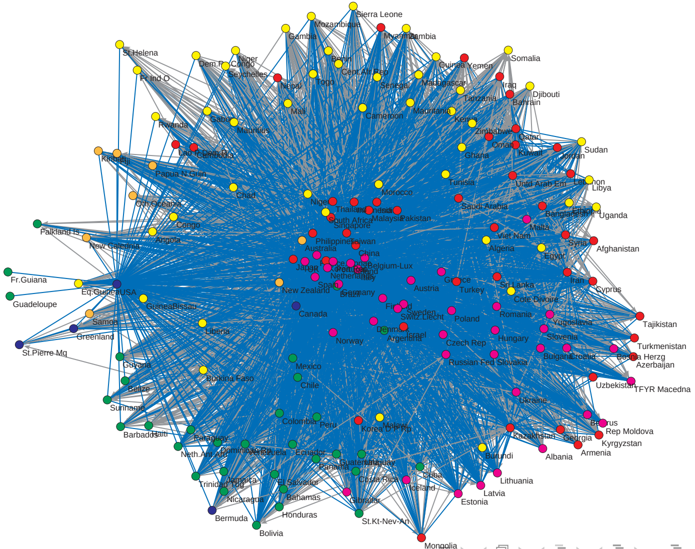
Yifan Hu developed a multilevel graph drawing algorithm for visualization of large graphs. For demonstration he applied it to the *University of Florida Sparse Matrix collection*. The results are available in the *Gallery of Large Graphs* [15].

From these examples we can see that, in some cases, the graph drawing algorithms can detect symmetries in a given graph and also a 'structure' ((sub)trees, clusters, planarity, etc.). The main problem are graphs with dense part(s).
For dense networks a better approach is to display them using matrix representation (*Pajek* [6], *Matrix Zoom* [1, 2], and *MatrixExplorer* [10]). A matrix representation is determined by an ordering of nodes. There exist several algorithms to produce the orderings [13, 12].

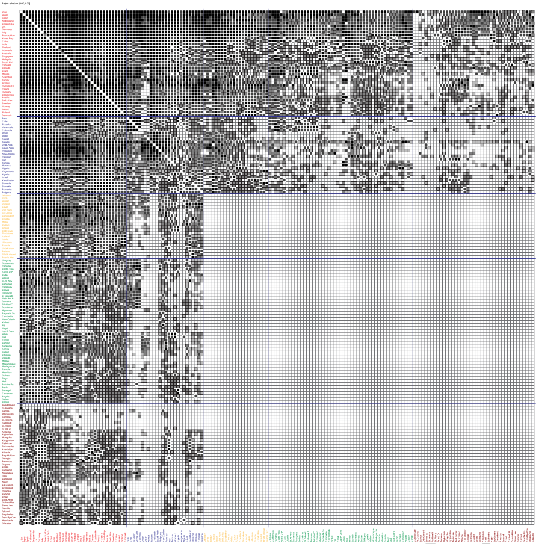
Most ordering algoritms were designed for applications in numerical, and not data, analysis.

The orderings can be also determined using clustering or blockmodeling methods [9].

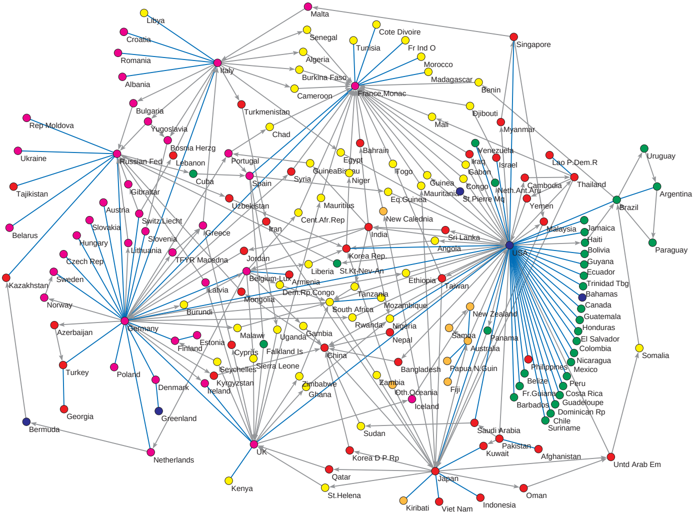
A display of World Trade 1999 network



Matrix display of World Trade 1999 network



Pathfinder skeleton of World Trade 1999 network



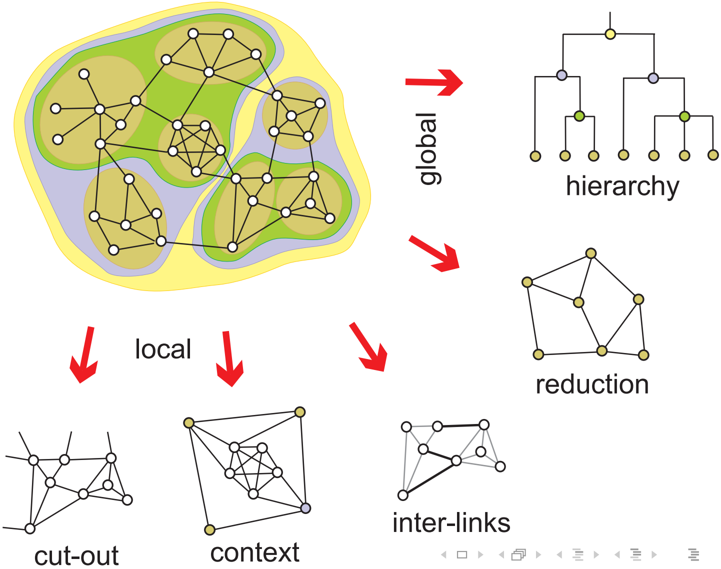
Goals

While the technical problems of graph drawing could ask for a single 'best' picture, the network analysis is also a part of data analysis. Its goal is to get insight not only into the structure and characteristics of a given network, but also into how this structure influences processes going on over the network. We usually need several pictures to present the obtained results.

The main tool to deal with large objects is abstraction. In graphs it is usually realized using a hierarchy of partitions. Shrinking or extracting selected classes of the partition we obtain a smaller reduced graph.

Small graphs can be presented in their totality and in detail within a single view. In a comprehensive view of large graphs, details become lost – conversely a detailed view can encompass only a part of the graph.

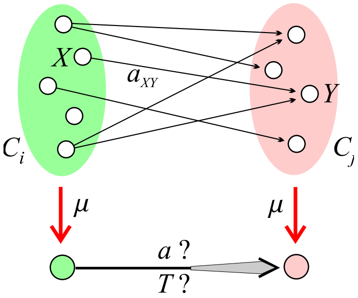
Decomposition and parts of network



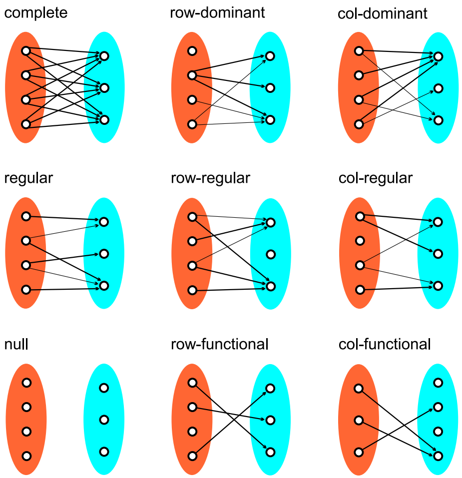
Generalized Blockmodeling

A *blockmodel* consists of structures obtained by identifying all units from the same cluster of the clustering **C**. For an exact definition of a blockmodel we have to be precise also about which blocks produce an arc in the *reduced graph* and which do not, and of what *type* [9].

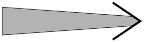
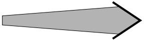
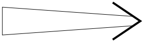
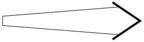
Some types of connections are presented in the figure on the next slide. The reduced graph can be represented by relational matrix, called also *image matrix*.



Block Types


complete
row-dominant
col-dominant
regular
row-regular
col-regular
null
row-functional
col-functional

Characterizations of Types of Blocks

null	nul	all 0 *	
complete	com	all 1 *	
regular	reg	1-covered rows and columns	
row-regular	rre	each row is 1-covered	
col-regular	cre	each column is 1 -covered	
row-dominant	rdo	$\exists$ all 1 row *	
col-dominant	cdo	$\exists$ all 1 column *	
row-functional	rfn	$\exists!$ one 1 in each row	
col-functional	cfn	$\exists!$ one 1 in each column	
non-null	one	$\exists$ at least one 1	

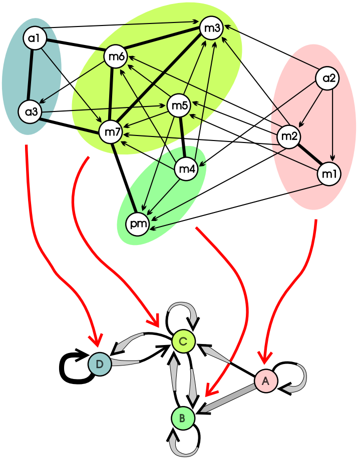
* except this may be diagonal

A block is *symmetric* iff $\forall X, Y \in C_i \times C_j : (XRY \Leftrightarrow YRX)$.

Blockmodeling as a clustering problem

The goal of *blockmodeling* is to reduce a large, potentially incoherent network to a smaller comprehensible structure that can be interpreted more readily. Blockmodeling, as an empirical procedure, is based on the idea that units in a network can be grouped according to the extent to which they are equivalent, according to some *meaningful* definition of equivalence.

In model nodes we can preserve the information about the subgraph structure: connectivity, acyclic, empty, ...



Large networks

In larger/denser networks there is often too much information to be presented at once. A possible answer are interactive layouts on the computer screen where the user controls what (s)he wants to see.

The computer screen is a different medium, which offers many new possibilities: parallel views (global and local); brushing and linking; zooming and panning; temporary elements (additional information about the selected elements, labels, legends, markers, etc.); highlighted selections; and others. These features can and should be maximally leveraged to support data analytic tasks; or repeating the Shneiderman's mantra: overview first, zoom and filter, then details on-demand (extended with: relate, history and extract).

Sheet of paper

The literature on graph drawing, is dominated by the 'sheet of paper' paradigm – the solutions and techniques are mainly based on the assumption that the final result is a static picture on a sheet of paper. In this case to present a large data set we need a large 'sheet of paper' – but this has a limit.

In an interactive dynamic visualization of a graph on the computer screen it needs not to be displayed in its totality. Inspecting a visualization the user can select which parts and elements will be displayed and how.

Layouts

The basic steps in graph/network visualization are:

graph/network $\rightarrow$ $\boxed{\text{analyses}}$ $\rightarrow$ layouts $\rightarrow$ $\boxed{\text{viewer}}$ $\rightarrow$ pictures

The networks to be visualized are usually not large – up to some thousans of nodes. On this scheme the development of different tools can be based depending on the kind of users (simple, advanced) and their tasks (exploration, monitoring, analysis, reporting, learning, story telling) they address.

In some cases already a standard viewer will be sufficient (for example SVG viewer, X3D viewer, or a special graph layout viewer), in others a complete network analysis system is needed.

Layouts are obtained by augmenting the network data with results of analyses and user's decisions to be used to visualize the network. In `Pajek`'s input format there are several layout elements from `Pajek`'s predecessors (see `Pajek`'s manual, pages 69-73).

Styles

As in typesetting

$$\text{text} + \text{formatting} = \text{formated text}$$

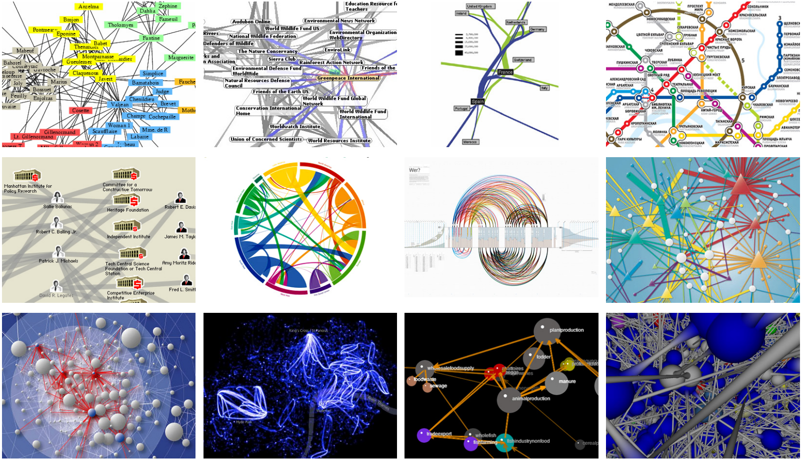
so in network visualization

$$\text{network} + \text{layout} = \text{picture}$$

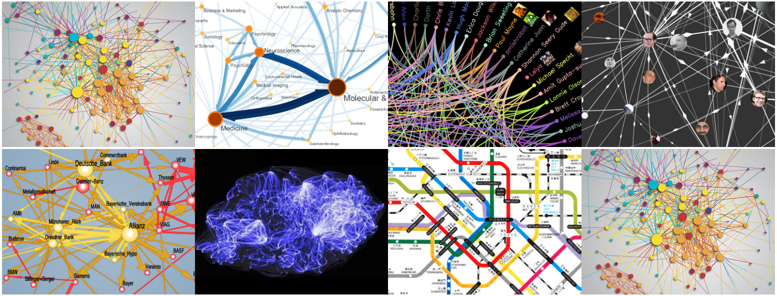
It would be useful to define a common layout format (an extension of GraphML or JSON ?) so that independent viewer modules can be developed and combined with different layout algorithms. To specify layouts we can borrow from the typesetting the notion of *style*.

Some useful ideas can be found in nViZn ("envision") system [14]. Another interesting option are network visualization solutions based on the library d3.js (Javascript and SVG).

Visual complexity

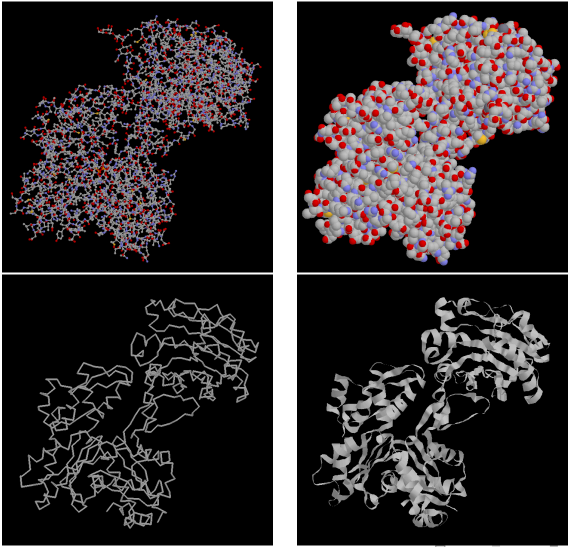


Visual complexity



Glasses: Rasmol displays

BallStick, SpaceFill, Backbone, Ribbons



Glasses, lenses and zooming

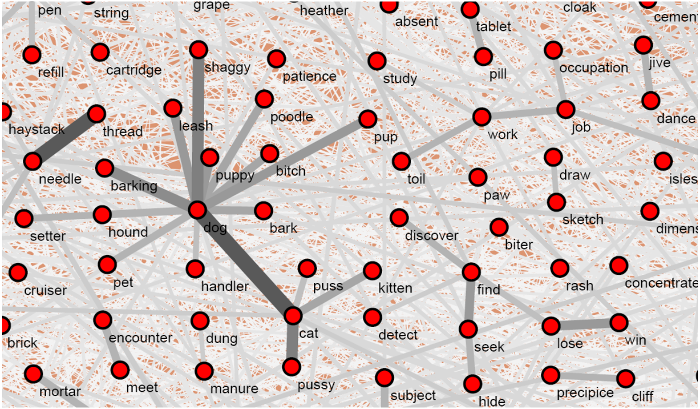
Glasses have effect on the entire window, and *lenses* only on the selected region or elements. Selecting different glasses we obtain different views on the same data – supporting different visualisation aims.

There are many kinds of glasses in representation of graphs. For example fish-eye views, matrix representation, using application field conventions (genealogies, molecules, electric circuits, SBGN), displaying nodes only, selecting the type of labels (long/short name, value), displaying only the important nodes and/or links, size of nodes determined by core number or betweenness.

In the two pictures produced by James Moody the glasses are the coloring of its nodes by different partitions: age partition (left picture) and race partition (right picture).

Part of the big picture (EAT)

The glasses in this case are based on ordering the edges in increasing order of their values and drawing them in this order – stronger edges cover the weaker. The picture emphasizes the strongest substructures; the remaining elements form a background.



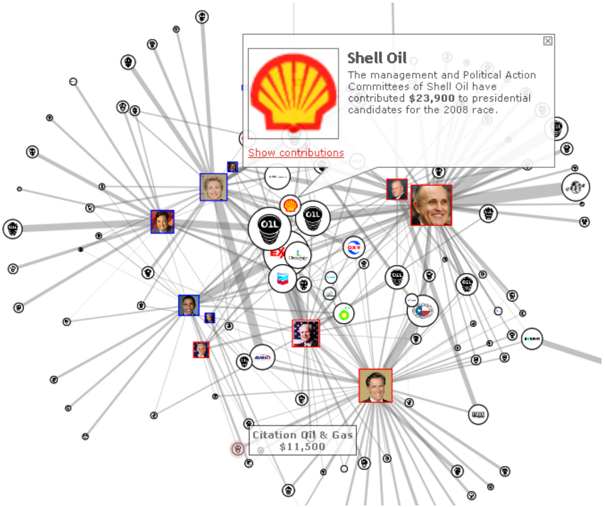
Lenses: temporary info about the selected vertex

An example of lens is presented on the next slide – contributions of companies to different presidential candidates from *Follow the Oil Money*. For a selected node the information about it is displayed. Another example of lens would be to temporarily enhance the display of neighbors of the selected node or to display their labels. The "shaking" option used in `Pajek` to visually identify all vertices from selected cluster is also a kind of lense.

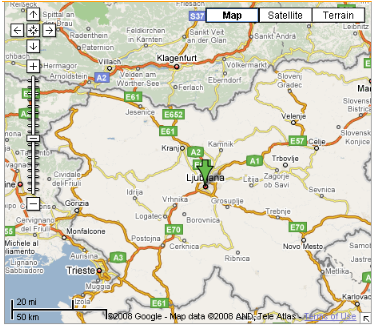
Additional enhancement of a presentation can be achieved by the use of support elements such as labels, grids, legends, and various forms of help facilities.

An important concept connected with zooming is the LOD (Level of Detail) – subobjects are displayed differently depending on the zoom depth.

Lenses: temporary info about the selected vertex

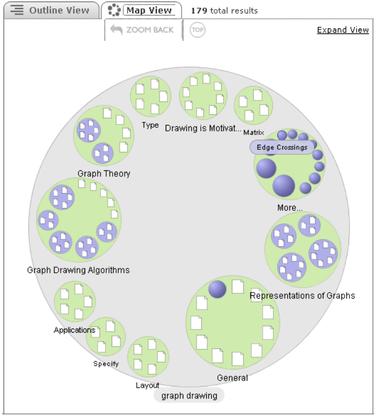


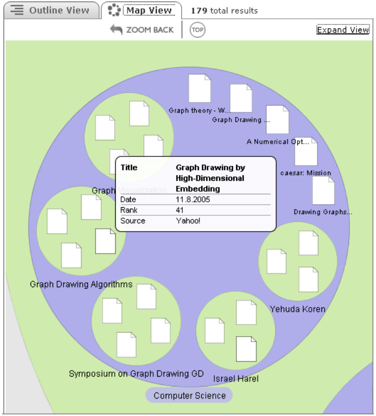
Zoom, glasses, lenses, navigation: Google Maps



A nice example of combination of these techniques is the Google Maps service. It combines zooming, glasses (Map, Satellite, Terrain), navigation (left, right, up, down) and lenses (info about points). The maps at different zoom levels provide information at different level of detail and in different form.

Zoom, glasses, lenses, navigation: Grokker



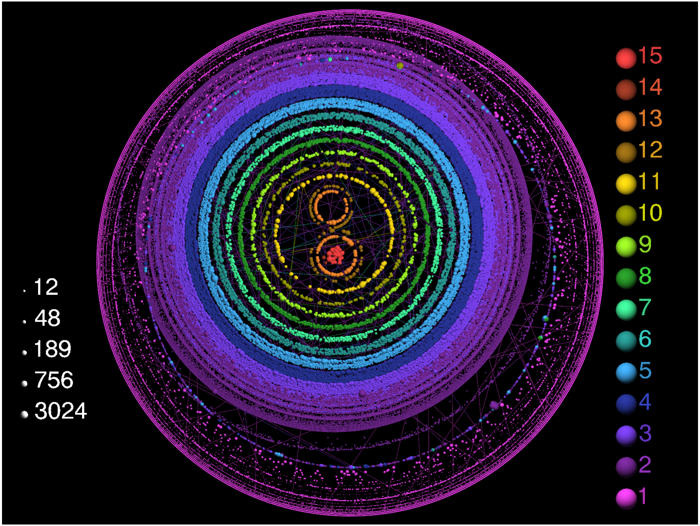


Maps

For a given hierarchical clustering of its nodes a similar approach to Grokker could be used also for inspection of large graphs/networks. To produce higher level 'maps' different methods can be used: k-core representation, density contours, generalized blockmodeling, clustering, skeletons preserving only important nodes and links, etc. In visualizing the 'maps' new graphical elements (many of them still to be invented) can be used preserving/indicating the information about the structure at lower level.

k-core structure of the .fr domain

Alessandro Vespignan et. al.



Problems

1. Network maps:
   area photo $\rightarrow$ map;
   network $\rightarrow$ **?** ;
2. Streams in acyclic networks:
   AcyNet $\rightarrow$ **Map** $\rightarrow$ **layout** of AcyNet based on Map;
3. Description of layouts:
   data $\rightarrow$ $\boxed{\text{NA}}$ $\rightarrow$ **layout** $\rightarrow$ $\boxed{\text{viewer(s)}}$ $\rightarrow$ picture(s).

References I

- Abello, J., van Ham, F.: *Matrix Zoom – A Visual Interface to Semi-external Graphs*. IEEE Symposium on Information Visualization 2004, October 10-12, Austin, Texas, USA, 183-190.
- Abello, J., van Ham, F., Neeraj Krishnan: *ASK-GraphView : A Large Scale Graph Visualization System*. IEEE Transactions on Visualization and Computer Graphics, 12 (2006) 5.
- Batagelj, V.: Complex Networks, Visualization of. In R.A. Meyers, ed., Encyclopedia of Complexity and Systems Science, Springer 2009: 1253-1268.
- Batagelj, V.: Some visualization challenges from SNA. Graph Drawing 2005 workshop slides. Limerick, 2005.
- Batagelj, V., Doreian, P., Ferligoj, A., Kejžar, N.: Understanding Large Temporal Networks and Spatial Networks. Wiley, 2014.

References II

- Batagelj, V., Mrvar, A., Zaveršnik, M.: *Partitioning approach to visualization of large graphs*, In Kratochvíl, J. (ed), *Lecture notes in computer science*, 1731. Springer, Berlin, 1999, 90–97.
- Vladimir Batagelj, Andrej Mrvar: Pajek manual.
- Brandes, U., Pich, C.: Eigensolver Methods for Progressive Multidimensional Scaling of Large Data. In: Proceedings of Graph Drawing '06 (LNCS 4372), 2007, pp. 42-53.
- Doreian, P., Batagelj, V., Ferligoj, A.: *Generalized Blockmodeling*. Cambridge University Press, 2005.
- Henry, N., Fekete, J-D.: *MatrixExplorer: a Dual-Representation System to Explore Social Networks*. IEEE Transactions on Visualization and Computer Graphics, 12 (2006) 5.

References III

- Kobourov, S.G.: Force-Directed Drawing Algorithms. In R. Tamassia: Handbook of Graph Drawing and Visualization. CRC, 2013, pp. 383-408.
- Mueller, C.: Matrix visualizations. `http://www.osl.iu.edu/~chemuell/data/ordering/sparse.html`
- Mueller, C., Martin, B., Lumsdaine, A.: *A Comparison of Vertex Ordering Algorithms for Large Graph Visualization*. APVIS 2007.
- SPSS nViZn: `http://www.spss.com/research/wilkinson/nViZn/nvizn.html`
- Yifan Hu: Gallery of Large Graphs. `http://www.research.att.com/~yifanhu/GALLERY/GRAPHS/index1.ht`
- De Nooy, W., Mrvar, A., Batagelj, V.: Exploratory Social Network Analysis with Pajek; Revised and Expanded Second Edition. Cambridge University Press, September 2011.

References IV

- Waltman, L., van Eck, N.J., Noyons, E.C.M.: A unified approach to mapping and clustering of bibliometric networks. Journal of Informetrics, 4(2019)4, pp. 629–635.
- Wilkinson, L.: The Grammar of Graphics. Springer-Verlag, New York, 2005.