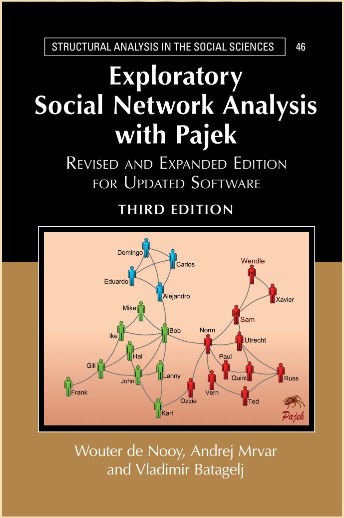
Analysis of Large Networks



Andrej Mrvar

Pajek Project File

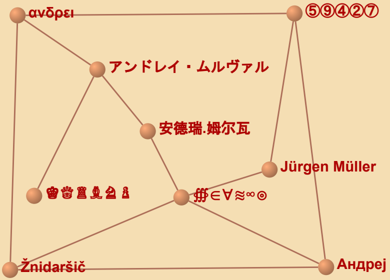
All types of Pajek objects can be combined into a single file – **Pajek Project File** (*.paj). Pajek project file can be constructed in the following way:

- read all data files in Pajek,
- compute some additional data,
- delete (dispose) some data that is not needed,
- save all as a project file with:
  File / Pajek Project File / Save
- next time we can restore everything using
  File / Pajek Project File / Read

Unicode...

Using Unicode for labels

For storing non-latin labels (e.g. labels including `č`, `š`, `ž`, Chinese, Greek laters. . .) Pajek supports **Unicode**. **UTF-8** encoding is used. Pajek requires **BOM** (**Byte Order Mark**) to recognize UTF-8 files.
Pajek stores files in UTF-8 files with BOM if:
Save Files as Unicode UTF-8 and **with BOM**
are checked in **Options** / **Read - Write**.
If not, files are stored as ASCII files where non latin characters are stored using Unicode codes (e.g. `Ž` for `Ž`).
When `č`, `š`, `ž` are entered in cp1250, Windows10 automatically tranformed them to Unicode.



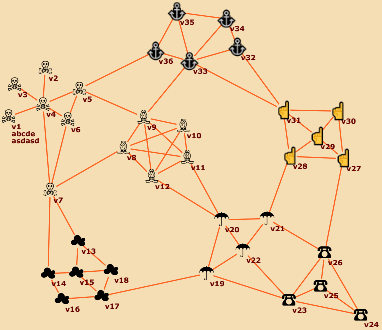
...Unicode

Using Unicode symbols for visualizing partitions

Networks

Pajek Project File
Unicode
Excel2Pajek
Multiple relations
Temporal network
Text→Networks
Acyclic networks
Citation networks
Genealogies
Clust. Coefficient
Islands
Communities
E-I Index
Comparing part.
Short Cycles
Example
Network models
Simulations
Other networks

We can display Unicode symbols in the middle of vertices to visualize some property defined by a partition (e.g. gender).
First select which Unicode symbol should present each cluster:
Draw / Options / Symbols for Partition Clusters
then select some partition as the *Second Partition*, and finally check:
Options / Mark Vertices Using / Cluster Symbols of Second Partition
More on using symbols: `http://mrvar.fdv.uni-lj.si/pajek/Symbols/symbols-examples.htm`



Excel2Pajek, Txt2Pajek

Links among vertices can be provided with pairs of vertices, which are not sequential numbers 1..n, but any numbers or even labels which describe vertices.

We enter labels defining initial and terminal vertices in two columns of Excel file and use program **Excel2Pajek** to transform Excel file to Pajek file.

If we use text file instead of Excel file, we must use tabulator as delimiter among labels and then use program **Txt2Pajek**.

Multiple relations networks...

Networks containing multiple relations on the same set of vertices (*inviting to a party, borrowing teaching material, discussion about personal matters...*), which can all be written to a single file. This can be done in two ways:

- add to a keyword for description of links:
  `*arcs, *edges, *arcslist, *edgeslist, *matrix`
  the number of relation followed by its name: e.g.
  `*Arcslist :3 "inviting to a party"`
  All links controlled by this keyword belong to the specified relation (example: `sampsonmul.net` 'Testdata').
- Any link controlled by `*arcs` or `*edges` , can be assigned to selected relation by starting its description by the number of this relation, e.g.
  `3:  47 14 5`
  Explanation: Link with endvertices 47 and 14 and weight 5 belongs to relation 3.

...Multiple relations networks

Some Pajek commands:

Network / Multiple Relations Network / Info – number of arcs/edges in each relation.
Network / Multiple Relations Network / Extract Relation(s) into Separate Networks – extracting selected relations from multiple relations network.

Visualization of multiple relations networks:

Options / Colors / Edges / Relation Number and
Options / Colors / Arcs / Relation Number - colors of edges/arcs are determined by relation number
Options / Colors / Relation Colors - color table for relation numbers
Options / Lines / Draw Lines / Relations - showing only lines belonging to selected relations e.g. 1-3,6,10-15

Temporal networks...

Temporal network is network that is changing over time.
In temporal network vertices and lines are not necessarily present or active in all time points.
Additional parameters are used to define presence of vertices and lines. They are given between signs `[` and `]`:
`-` is used to divide lower and upper limit of interval,
`,` is used to separate intervals,
`*` means infinity. Example:

```
*Vertices 3
1 "a" [5-10,12-14]
2 "b" [1-3,7]
3 "c" [4-*]
*Edges
1 2 1 [7]
1 3 1 [6-8]
```

Vertex 'a' is active from times 5 to 10, and 12 to 14, vertex 'b' in times 1 to 3 and in time 7, vertex 'c' from time 4 on. Line from 1 to 2 is active only in time 7, line from 1 to 3 in times 6 to 8.

Lines and vertices in a temporal network should satisfy the *consistency* condition: a line can be active in time t only if its end-vertices are active too. When generating time slices of a given temporal network only 'consistent' lines are generated. Note that time records should always be written at the od of the row where vertices / lines are defined.

Example 1: **Sampson monastery data** is a temporal network.

Sampson studied relations among 18 monks in the New England monastery. He measured several relations, e.g. `friendship (affect)`, `esteem`, `influence`, `sanction` in different time points.

Reference: Sampson, S (1969): Crysis in a cloister. Unpublished doctoral dissertation. Cornell University

...Temporal networks...

Example 2: **Lindenstrasse**: temporal network of actors and relations among them in the long-running German soap opera.

For each actor her/his name, gender, birthdate, and several other records are available. Additionally for each actor episode numbers in which the actor played actively are given. For each line in the network its meaning is given: family relation, business relation, unfriendly relationships, . . .

Properties of vertices are represented by different shapes, colors, sizes of vertices: e.g. triangles correspond to men, circles to women; and properties of lines are represented by colors: green line stands for family relation, blue for business relations, ... LindenStrasse is **temporal** but also **multirelational** network:

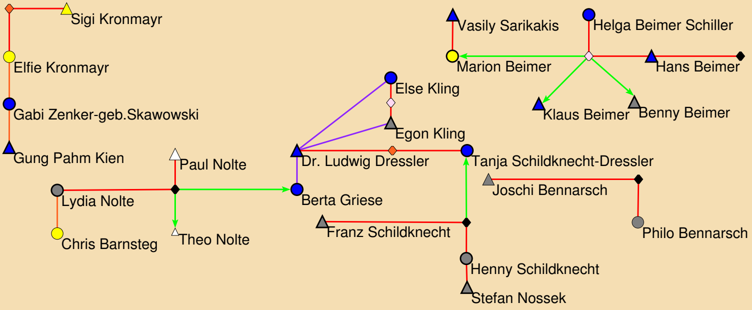
1. ”family relation” (arcs and eges)
2. ”unfriendly relationships” (edges)
3. ”business relations” (edges)
4. ”friendships” (edges)
5. ”partner relations”

...Temporal networks...

Network / Temporal Network / Generate in Time
Generate network in specified time points or interval. Input first time, last time and step (integers).

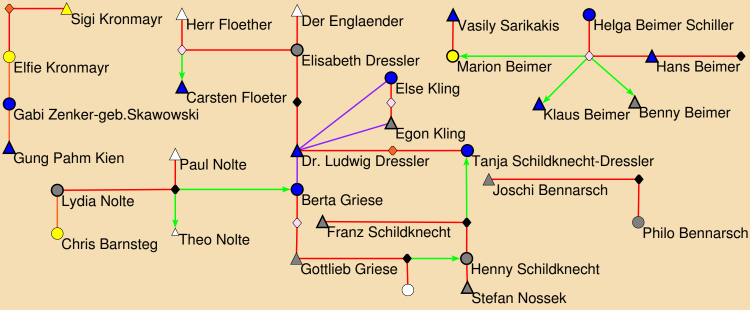
- **All** – Generate all networks in specified time points.
- **Only Different** – Generate network in specified time point only if the new network will differ in at least one vertex or line from the last network which was generated.
- **Interval** – Generate network with vertices and lines present in selected time interval.

Episode 5

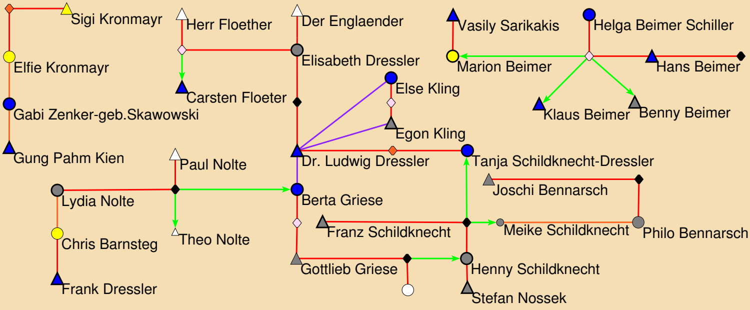

Sigi Kronmayr
Elfie Kronmayr
Gabi Zenker-geb.Skawowski
Gung Pahm Kien
Paul Nolte
Lydia Nolte
Chris Barnsteg
Theo Nolte
Vasily Sarikakis
Helga Beimer Schiller
Marion Beimer
Hans Beimer
Else Kling
Klaus Beimer
Benny Beimer
Egon Kling
Dr. Ludwig Dressler
Tanja Schildknecht-Dressler
Joschi Bennarsch
Berta Griese
Franz Schildknecht
Philo Bennarsch
Henny Schildknecht
Stefan Nossek

...Temporal networks...

Episode 6



Episode 7


Sigi Kronmayr
Herr Floether
Der Englaender
Vasily Sarikakis
Helga Beimer Schiller
Elfie Kronmayr
Elisabeth Dressler
Marion Beimer
Hans Beimer
Carsten Floeter
Else Kling
Gabi Zenker-geb.Skawowski
Klaus Beimer
Benny Beimer
Egon Kling
Gung Pahm Kien
Paul Nolte
Dr. Ludwig Dressler
Tanja Schildknecht-Dressler
Joschi Bennarsch
Lydia Nolte
Berta Griese
Franz Schildknecht
Meike Schildknecht
Philo Bennarsch
Chris Barnsteg
Theo Nolte
Gottlieb Griese
Henny Schildknecht
Frank Dressler
Stefan Nossek

Using program **PajekToSVGAnim** we can prepare animations of temporal networks. Some examples of such animations:

- Sampson monastery: `http://mrvar.fdv.uni-lj.si/sola/info4/andrej/Anim/Sampson/sampson-anim.htm`
- Sampson monastery as multirelation network: `http://mrvar.fdv.uni-lj.si/sola/info4/andrej/Anim/Sampson/sampson-multi-anim.htm`
- Lindenstrasse: `http://mrvar.fdv.uni-lj.si/sola/info4/andrej/Anim/Linden/linden-anim.htm`

First we have to prepare an input file. We can use program *Pajek* for this task:

- In *Pajek* load temporal network (e.g `Sampson.net`, `Sampson-Multi.net`, or `linden.net`).
- Generate networks in selected time points: **Network** / **Temporal Network** / **Generate in Time** / **Only Different**.
- In *Draw window* select **Export** / **Append to Pajek Project File** / **Select File** and select name of generated project file.
- First select the original network and add it to project file: **Export** / **Append to Pajek Project File** / **Append - F3**
- Proceed to network in the first time point, make some manual adjustments of vertices (if needed) and add it to project file (**F3**).
- Proceed to network in next time point, make some manual adjustments of vertices (if needed) and add it to project file (**F3**).
- When you add the network in last time point data preparation in Pajek is finished.

...Temporal networks

...Animations

Next, use program **PajekToSVGAnim** to create the animation:

- Select project file created by Pajek (e.g. `Sampson-Anim.paj`, `Sampson-Multi-Anim.paj`, or `Linden-Anim.paj`) using **Browse** in the **Source** field.
- Change some parameters, if needed.
- Run: **SVG** / **Generate - Firefox** / **Chrome**.

All files (temporal network, input project file generated by Pajek and animations) are available here:

- Sampson: **`http://mrvar.fdv.uni-lj.si/sola/info4/andrej/Anim/Sampson/Sampson.zip`**
- Lindenstrasse: **`http://mrvar.fdv.uni-lj.si/sola/info4/andrej/Anim/Linden/Linden.zip`**

Note that some options work only when run from server and not when run locally.

Networks obtained from texts...

Dictionary networks...

In a dictionary network vertices are terms described in a dictionary; from term *u* there is an arc to term *v* when term *v* appears in the description of term *u*.

ODLIS – Online Dictionary of Library and Information Science

library

From the Latin *liber*, meaning "book." In Greek and the Romance languages, the corresponding term is *bibliotheca*. A collection or group of collections of books and/or other print or nonprint materials organized and maintained for use (reading, consultation, study, research, etc.). Institutional libraries, organized to facilitate access by a specific clientele, are staffed by librarians and other personnel trained to provide services to meet user needs. By extension, the room, building, or facility that houses such a collection, usually but not necessarily built for that purpose. Directory information on libraries is available alphabetically by country in *World Guide to Libraries*, a serial published by K.G. Saur. Two comprehensive worldwide online directories of library homepages are *Libdex* and *Libweb*. See also the UNESCO Libraries Portal. Abbreviated *lib*. ***See also***: academic library, government library, monastic library, new library, proto-library, public library, special library, and subscription library.

Also, a collective noun used by publishers, particularly during the Victorian period, for certain books published in series (*example*: ***Everyman's Library***).

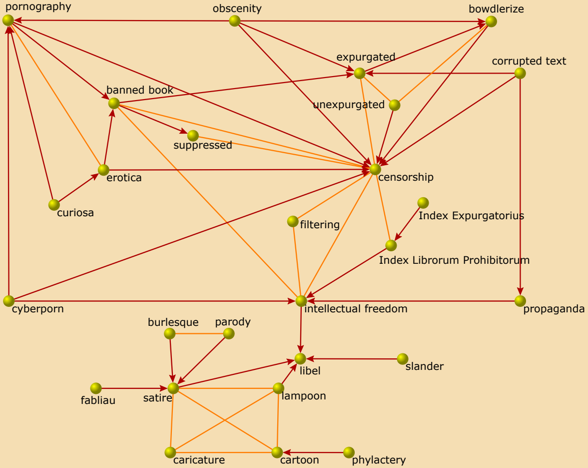
...Networks obtained from texts...

...Dictionary networks...

Networks

Pajek Project File
Unicode
Excel2Pajek
Multiple relations
Temporal network
Text → Networks
Acyclic networks
Citation networks
Genealogies
Clust. Coefficient
Islands
Communities
E-I Index
Comparing part.
Short Cycles
Example
Network models
Simulations
Other networks

ODLIS



FOLDOC – Free On-line Dictionary Of Computing

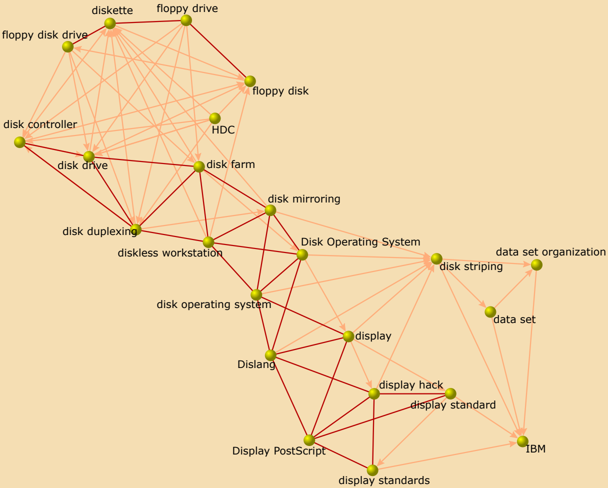
computer

<*computer*>

A machine that can be programmed to manipulate symbols. Computers can perform complex and repetitive procedures quickly, precisely and reliably and can store and retrieve large amounts of data. Most computers in use today are electronic digital computers (as opposed to analogue computers).

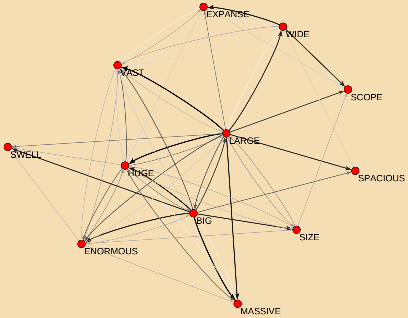
The physical components from which a computer is constructed are known as hardware, which can be of four types: CPU, memory, input devices and output devices.

FOLDOC


diskette
floppy drive
floppy disk drive
floppy disk
HDC
disk controller
disk drive
disk farm
disk mirroring
disk duplexing
diskless workstation
Disk Operating System
data set organization
disk striping
disk operating system
data set
display
Dislang
display hack
display standard
Display PostScript
IBM
display standards

Edinburgh Associative Thesaurus - EAT

In this network associations among words collected on student population are provided. Vertices are words. Links $u \rightarrow v$ are determined using the following question: Which word v comes into your mind when you hear word u? The weight tells us how many times this association was selected.



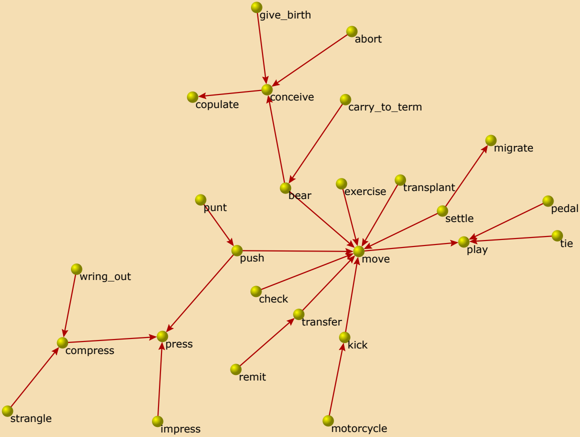
Semantic network

WordNet – A Lexical Database for English

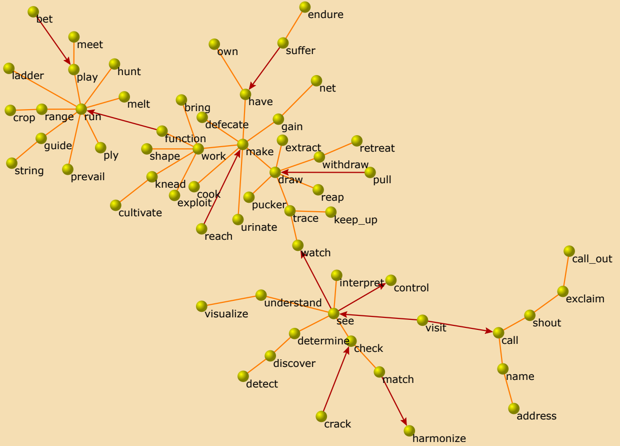
Multirelational network with the following relations:

1. hypernym pointers (maple.tree, tree.plant, fire.attack)
2. entailment pointers (drive.ride push.press)
3. similar pointers (boiling.hot)
4. member meronym pointers (Luxemburg.Benelux), (the Netherlands.Benelux)
5. substance meronym pointers (pavement.street)
6. part meronym pointers (medical-diagnosis.medical-care)
7. cause pointers (anesthetize.sleep, pasteurize.condense)
8. grouped verb pointers (exit.leave)
9. attribute pointers (quality.bad)

WordNet - Entailment pointers



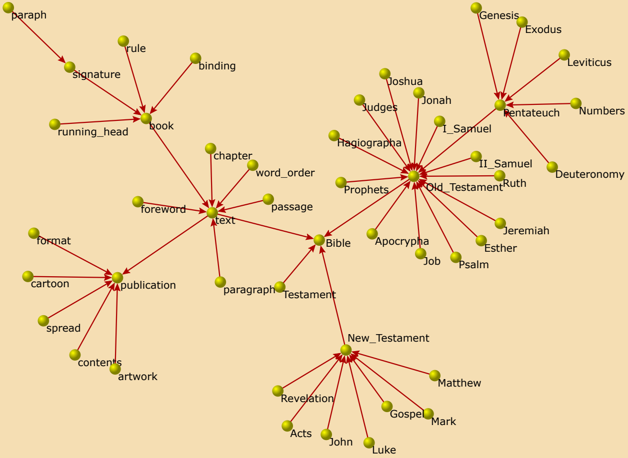
WordNet - Grouped verb pointers



...Networks obtained from texts...

...Semantic networks...

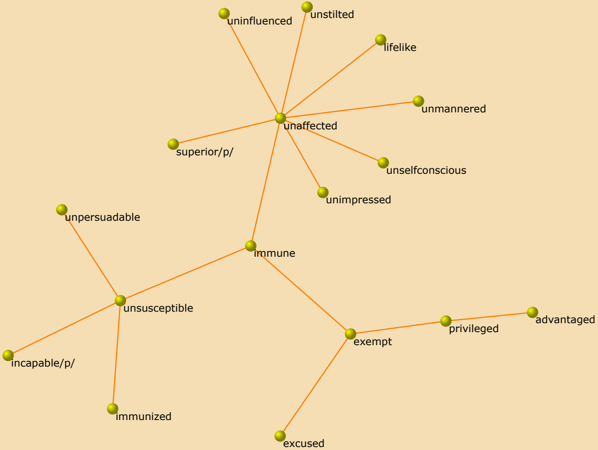
WordNet - Part meronym pointers



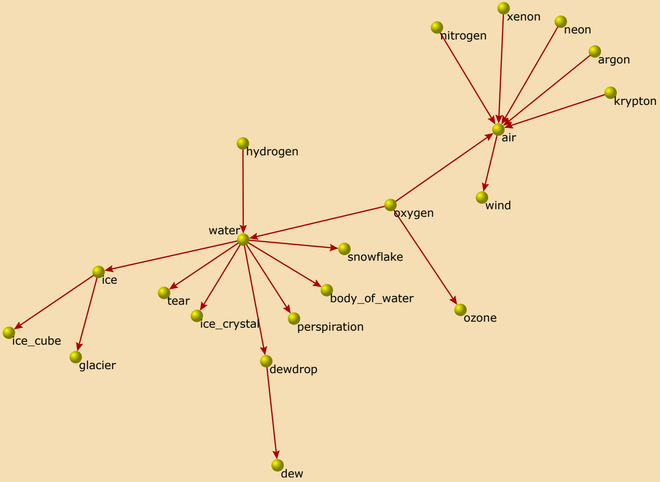
Networks

Pajek Project File
Unicode
Excel2Pajek
Multiple relations
Temporal network
Text→Networks
Acyclic networks
Citation networks
Genealogies
Clust. Coefficient
Islands
Communities
E-I Index
Comparing part.
Short Cycles
Example
Network models
Simulations
Other networks

WordNet - Similar pointers



WordNet - Substance meronym pointers



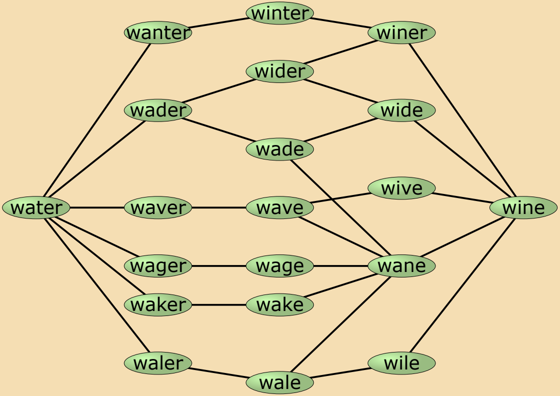
...Networks obtained from texts

Changing, deleting, adding letters

Two words are connected by an edge if we can reach one from the other by

- changing a single character (e. g., `work` — `word`)
- adding / removing a single character (e. g., `ever` — `fever`).

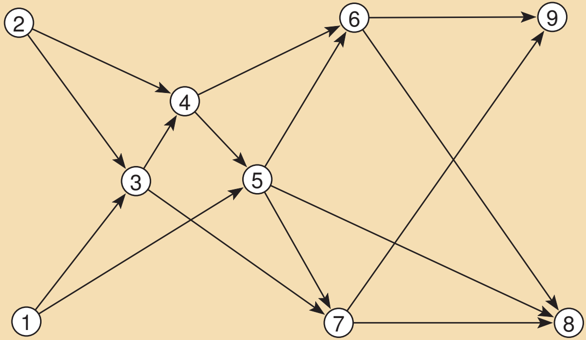
Examples: **Dic28, SI5**



Acyclic networks...

Citation networks...

Input: Acyclic network which represents citations among papers.
Result: Relative importance of papers and relative importance of citations. Example:



In citation networks vertices represent papers, directed link from paper x to paper y means, that y cites x (x is cited in y).

...Acyclic networks...

...Citation networks...

We want to find out which paper (vertex) is the most important and which citation (link) is the most important in such network.

The simplest such measure which could give as answer to this question is output degree (number of arcs going out of the vertex).

Network output degree can be computed easily:
Network / **Create Partition** / **Degree** / **Output**.

In the example network paper 5 is the most important with 3 citations.

A much better measure is composed by taking number of paths passing through each vertex into account.

Traversal weights algorithm

- Unique first (source) and last (sink) vertices are added to acyclic network.
- We make two passes through this network:

  1. In first pass we start in unique source vertex and travel in direction of arcs until we reach the sink vertex. By doing this we remember number of paths coming to each vertex we reach (this can be done efficiently by summing counts for vertices from which we reach the selected vertex).
  2. In second pass we start in unique sink vertex and travel in the opposite direction of acs until we reach the source vertex. In this second pass we remember number of paths leading from this vertex to sink vertex.

From the two counts obtained for each vertex we can compute number of paths from the source to the sink vertex that go through selected vertex.

We use the fundamental theorem of combinatorics:

If there exist M paths from the source vertex to selected vertex and if there exist N paths from selected vertex to the sink vertex then number of paths from source vertex to sink vertex that pass through selected vertex is equal to MN.

In a similar way we find out how many paths use selected link. Lets selected link goes from vertex i to vertex j.

If there exist M paths from the source vertex to vertex i and if there exist N paths from vertex j to the sink vertex then number of paths from source vertex to sink vertex that pass through selected link $i \rightarrow j$ is equal to MN.

If we want to compare these counts across different networks we can normalize them to obtain a measure in interval $[0, 1]$.

Three variants of this algorithm exist:

- **Paths Count (PC) method** – this is exactly the method we described (counting all paths from source to sink vertex);
- **Search Path Link Count (SPLC) method** – we consider each vertex in acyclic network as a source vertex: we add links from source vertex to each other vertex in the network. As result vertices and arcs at the end will receive higher traversal weights.
- **Search Path Node Pair (SPNP) method** – each vertex is considered as a source and as a sink vertex: we add links from source vertex to each other vertex in the network and from each vertex to the sink. As a result, vertices and arcs in the middle will receive higher traversal weights.

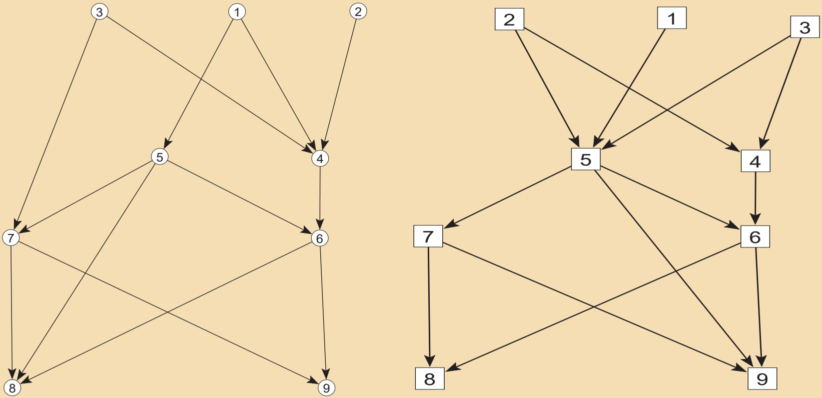
Absolute and relativne importances of papers using Paths Count (PC) and Search Path Link Count (SPLC)

paper	PC	$PC/30$	$SPLC$	$SPLC/58$
1	14	0.4667	14	0.2414
2	16	0.5333	16	0.2759
3	18	0.6000	27	0.4655
4	21	0.7000	35	0.6034
5	20	0.6667	35	0.6034
6	14	0.4667	26	0.4483
7	12	0.4000	22	0.3793
8	17	0.5667	32	0.5517
9	13	0.4333	25	0.4310

Absolute and relative importances of citations using Paths Count (PC) and Search Path Link Count (SPLC)

from	to	PC	$PC/30$	$SPLC$	$SPLC/58$
1	3	9	0.3000	9	0.1551
2	3	9	0.3000	9	0.1551
1	5	5	0.1667	5	0.0862
2	4	7	0.2333	7	0.1207
3	7	4	0.1333	6	0.1034
3	4	14	0.4667	21	0.3621
4	5	15	0.5000	25	0.4310
4	6	6	0.2000	10	0.1724
5	7	8	0.2667	14	0.2414
5	6	8	0.2667	14	0.2414
5	8	4	0.1333	7	0.1207
6	8	7	0.2333	13	0.2241
6	9	7	0.2333	13	0.2241
7	8	6	0.2000	11	0.1897
7	9	6	0.2000	11	0.1897

Two more small examples: citem.net and Citem2a.net



Traversal weights (importances of papers and citations):

Network / **Acyclic Network** / **Create Weighted Network**
+ Vector / **Traversal Weights** / **SPC** or **SPLC**

Pajek returns the following results:

1. Vector with number of paths starting in source and finishing in selected vertex (*Number of Different Incoming Paths*).
2. Vector with number of paths from selected vertex to sink vertex (*Number of Different Outgoing Paths*).
3. Vector containing importances of vertices/papers (*Citation weights SPC/SPLC...*).
4. New network where values of lines represent importances of citations (*Citation weights*).
   If we want to see the counts in **Draw** window, we must check **Options** / **Lines** / **Mark Lines** / **with Values**.
   To see top citations select **Network** / **Info** / **General** and type e.g. 10 if we want to list top 10 important citations.

Determining the Main Path from Source to Sink

Network / Acyclic Network / Create (Sub)Network / Main Paths

Local Search – in each step of the search select the arc(s) with the highest weight that are incident to the current arc. Tolerance (number between 0 and 1) can be used. Tolerance 0 means that in each step only arcs with the highest weights are taken into account, while larger tolerance means that also a bit smaller weights are considered.

1. **Local / Forward**
   This approach consists of choosing the source vertex (or vertices) incident with the arc(s) with the highest weight, selecting the arc(s) and the second vertex(s) of the arc(s), and repeating this step until a sink vertex is reached.
2. **Local / Backward**
   Instead of starting with one or more source vertices, we start with one or more sink vertices incident with the arc(s) with the highest weight and travel against the direction of the arcs.

Global Search – search for main path(s) with the highest overall sum of weights.

1. **Global / Standard**
   Standard global main path is the path from source to sink vertices with the overall highest sum of traversal weights on the path. This method is widely used across scientific disciplines. In project planning, for example, it is called Critical Path Method (CPM).

If we check **Through Vertices in Cluster** we search for local or global main path(s) passing through vertices selected in Cluster.

Sources of genealogies

People collect genealogical data for several different purposes:

- Research of different cultures in sociology, anthropology and history – kinship as fundamental social relation
- Genealogies of families and/or territorial units, e.g.,
  - genealogy of Ragusan (Dubrovnik) nobel families
  - Mormons genealogy
    `http://www.familytreemaker.com/`
  - genealogy of ”Skofja Loka district
  - genealogy of American presidents
- Special genealogies
  - Students and their PhD thesis advisors:
    - Theoretical Computer Science Genealogy:
      `http://sigact.acm.org/genealogy/`
    - Mathematics

Networks

Pajek Project File
Unicode
Excel2Pajek
Multiple relations
Temporal network
Text→Networks
Acyclic networks
Citation networks
Genealogies
Clust. Coefficient
Islands
Communities
E-I Index
Comparing part.
Short Cycles
Example
Network models
Simulations
Other networks

GEDCOM is standard for storing genealogical data, which is used to interchange and combine data from different programs. The following lines are extracted from the GEDCOM file of European Royal families.

```
0 HEAD
1 FILE ROYALS.GED
...
0 @I58@ INDI
1 NAME Charles Philip Arthur/Windsor/
1 TITL Prince
1 SEX M
1 BIRT
2 DATE 14 NOV 1948
2 PLAC Buckingham Palace, London
1 CHR
2 DATE 15 DEC 1948
2 PLAC Buckingham Palace, Music Room
1 FAMS @F16@
1 FAMC @F14@
...
...
0 @I65@ INDI
1 NAME Diana Frances /Spencer/
1 TITL Lady
1 SEX F
1 BIRT
2 DATE 1 JUL 1961
2 PLAC Park House, Sandringham
1 CHR
2 PLAC Sandringham, Church
1 FAMS @F16@
1 FAMC @F78@
...
...
0 @I115@ INDI
1 NAME William Arthur Philip/Windsor/
1 TITL Prince
1 SEX M
1 BIRT
2 DATE 21 JUN 1982
2 PLAC St.Mary's Hospital, Paddington
1 CHR
2 DATE 4 AUG 1982
2 PLAC Music Room, Buckingham Palace
1 FAMC @F16@
...
0 @I116@ INDI
1 NAME Henry Charles Albert/Windsor/
1 TITL Prince
1 SEX M
1 BIRT
2 DATE 15 SEP 1984
2 PLAC St.Mary's Hosp., Paddington
1 FAMC @F16@
...
0 @F16@ FAM
1 HUSB @I58@
1 WIFE @I65@
1 CHIL @I115@
1 CHIL @I116@
1 DIV N
1 MARR
2 DATE 29 JUL 1981
2 PLAC St.Paul's Cathedral, London
```

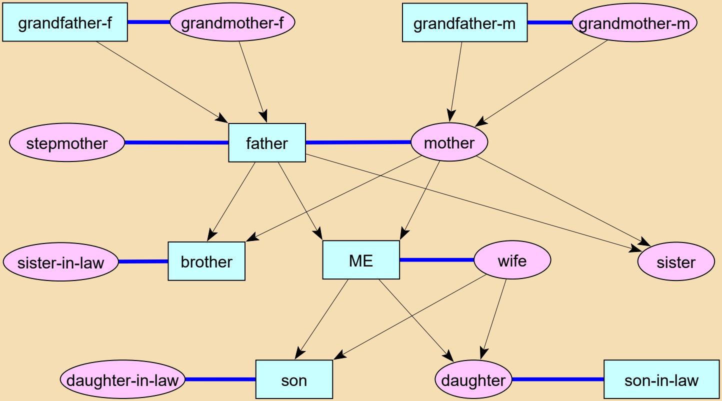
Representation of genealogies using networks

Genealogies can be represented as networks in different ways:

- as Ore-graph,
- as p-graph,
- as bipartite p-graph.

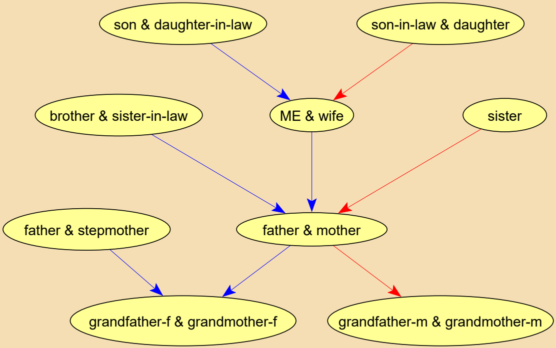
Ore-graph:

In Ore-graph every person is represented by a vertex, marriages are represented with edges and relation *is a parent of* as arcs pointing from each of the parents to their children.

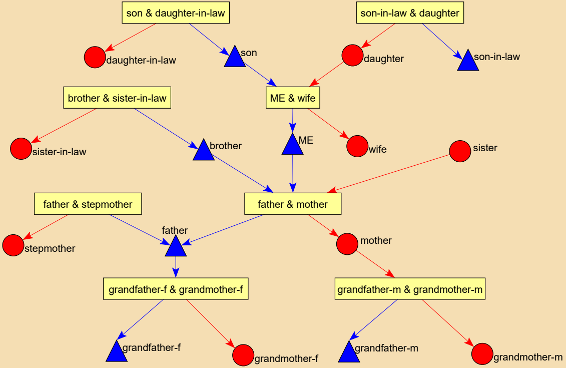


p-graph:

In p-graph vertices represent individuals or couples. In the case that person is not married yet (s)he is represented by a vertex, otherwise person is represented with the partner in a common vertex. There are only arcs in p-graphs – they point from children to their parents.



Bipartite p-graph:
has two types of vertices – vertices representing couples (rectangles) and vertices representing individuals (circles for women and triangles for men). Arcs again point from children to their parents.



Genealogies are sparse networks:

	Ore-graph				p-graph			
data	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert A\rvert$	$\frac{\lvert L\rvert}{\lvert V\rvert}$	$\lvert V_{ip}\rvert$	$\lvert V_{cp}\rvert$	$\lvert A_p\rvert$	$\frac{\lvert A_p\rvert}{\lvert V_p\rvert}$
Bruno	15512	4841	18664	1.52	6000	5289	10053	0.89
Combo	20350	7248	26199	1.64	6931	7945	14845	1.00
Dodderer	16761	5650	22425	1.68	6029	5652	11765	1.01
Drame	29606	8256	41814	1.69	13254	8939	21862	0.99
Little	25968	8778	34640	1.67	9212	8850	18233	1.01
President	2145	978	2223	1.49	218	1042	1222	0.97
Tillotsn	42559	12796	54043	1.57	15177	15959	31234	1.00
Loka	47956	14154	68052	1.71	19189	16039	36192	1.03
Silba	6427	2217	9627	1.84	2001	2479	5281	1.18
Ragusa	5999	2002	9315	1.88	2066	2310	5336	1.22
Tur	1269	407	1987	1.89	0	956	1114	1.17
Royal	3010	1138	3724	1.62	719	1422	2259	1.06

...Acyclic networks...

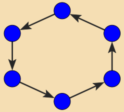
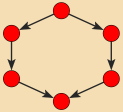
...Genealogies...

Networks

Pajek Project File
Unicode
Excel2Pajek
Multiple relations
Temporal network
Text → Networks
Acyclic networks
Citation networks
Genealogies
Clust. Coefficient
Islands
Communities
E-I Index
Comparing part.
Short Cycles
Example
Network models
Simulations
Other networks

Advantages of p-graphs

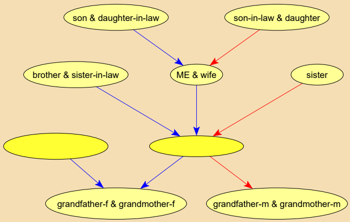
- there are less vertices and lines in p-graphs;
- p-graphs are directed, acyclic networks;

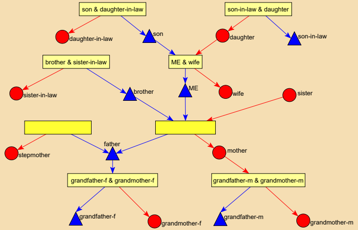
Cycle:  Semicycle: 

- in p-graphs every semi-cycle corresponds to a *relinking marriage*. There exist two types of relinking marriages:
  - blood marriage: e.g., marriage among brother and sister;
  - non-blood marriage: e.g., two brothers marry two sisters from another family;
- p-graphs are more suitable for analyses.

Advantages of bipartite p-graphs

Bipartite p-graphs have additional advantage: we can distinguish between *a married uncle and a remarriage of a father* or between *stepsisters and cousins*. This property enables us, for example, to find marriages between half-brothers and half-sisters.





Relinking index is a measure of relinking by marriages among persons belonging to the same families. Special case of relinking is a blood-marriage.

Let n denotes number of vertices in p-graph, m number of arcs, and M number of maximal vertices (vertices having output degree 0, $M \geq 1$).

If we take a connected genealogy we get

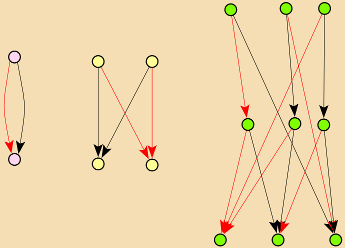
$$RI = \frac{m - n + 1}{n - 2M + 1}$$

For a trivial graph (having only one vertex) we define $RI = 0$.

RI has some interesting properties:

- $0 \leq RI \leq 1$
- If network is a forest/tree, then $RI = 0$ (no relinking).
- There exist genealogies having $RI = 1$ (the highest relinking).
- Relinking is usually computed for the largest biconnected component.

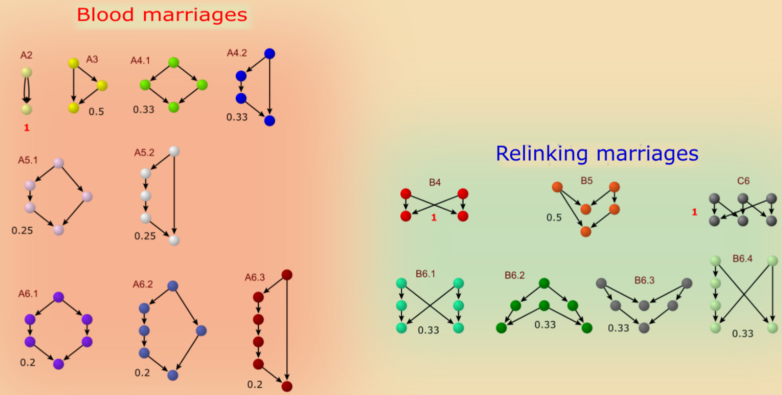
Patterns with Relinking Index = 1



Networks

Pajek Project File
Unicode
Excel2Pajek
Multiple relations
Temporal network
Text$\longrightarrow$Networks
Acyclic networks
Citation networks
Genealogies
Clust. Coefficient
Islands
Communities
E-I Index
Comparing part.
Short Cycles
Example
Network models
Simulations
Other networks

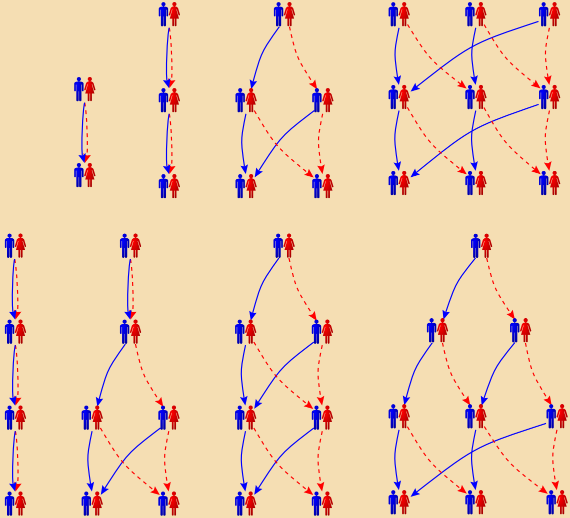
Relinking merriages (p-graphs with 2 up to 6 vertices)



...Acyclic networks...

...Genealogies...

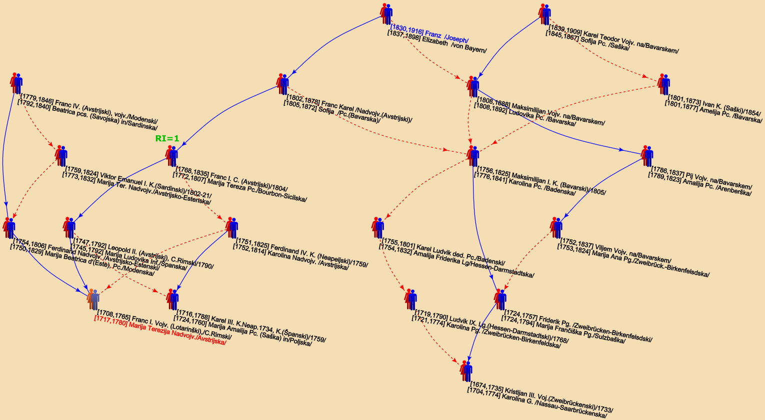
More blood marriages with RI=1



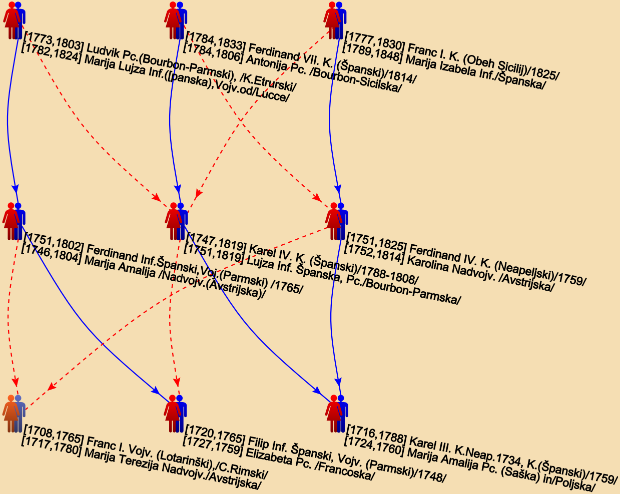
...Acyclic networks...

...Genealogies...

Noble familes are much more relinked than usual families. Example: In 1854 **Franz Jozef (1830-1916)** married his sixteen year old cousin 'Sisi'.

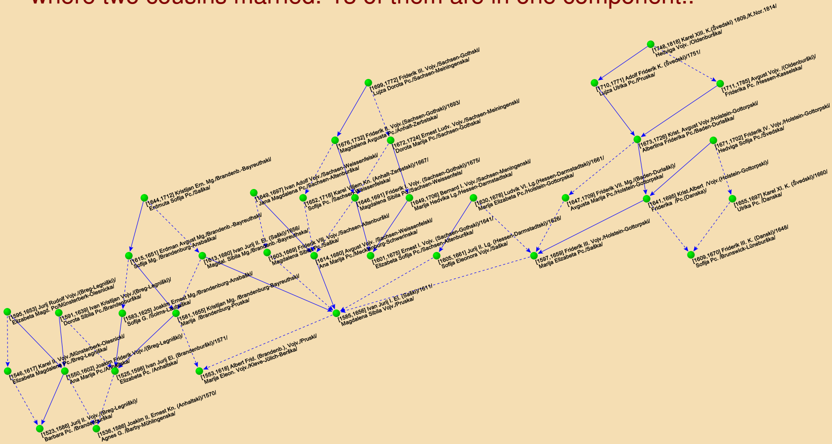


Several brother-sister exchanges in only two generations



European nobility

Genealogy contains records about around 60,000 persons of noble origin (collected by Nenad Novaković). We can find there 333 blood marriages where two cousins married. 13 of them are in one component..

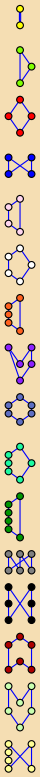


Comparing genealogies

For comparison, we took five genealogies:

- **`Loka.ged`** – genealogy of Škofja Loka district, Slovenia (P. Hawlina).
- **`Silba.ged`** – genealogy of the island Silba, Croatia (P. Hawlina).
  Special geographical position.
- **`Ragusa.ged`** – marriages among Ragusan (Dubrovnik) noble families between 12 and 16 century. Data collected by I. Mahnken (1960); entered to electronic form by P. Dremelj (1999).
  Very restricted marriage rules.
- **`Tur.ged`** – genealogy of Turkish nomads, Yörük. Data collected by Ulla C. Johansen and D.R. White (2001)
  A relinking marriage is a signal of commitment to stay within the nomad group.
- **`Royal.ged`** – genealogy of European royal families.

Frequency distribution of fragments



fragment	Loka	Silba	Ragusa	Tur	Royal	$\sum$
A2	1	0	0	0	0	1
A3	1	0	0	0	3	4
A4.1	12	5	3	65	21	106
B4	54	25	21	40	7	147
A4.2	0	0	0	0	0	0
A5.1	9	7	4	15	13	48
A5.2	0	0	0	0	0	0
B5	19	11	47	19	8	104
A6.1	28	28	2	65	13	140
A6.2	0	2	0	0	1	3
A6.3	0	0	0	0	0	0
C6	10	12	19	15	5	61
B6.1	0	1	2	0	0	3
B6.2	27	39	63	54	12	194
B6.3	47	30	82	46	13	218
B6.4	0	0	5	3	0	8
No. Indi	47956	6427	5999	1269	3010	
Largest bic.	4095	1340	1446	250	435	
RI	0.55	0.78	0.74	0.75	0.37	

Observations

- Generation jumps for more than one generation are very unlikely.
- There are many marriages B6.3 (two grandchildren married into the same family) and B6.2 (two familes were relinked by a marriage between children and again in the next generation by a marriage between grandchildren)
- In Tur there are many marriages of types A4.1 and A6.1.
- For all genealogies number of relinking 'non-blood' marriages is much higher than number of blood marriages (this is especially true for Ragusa, exception is Royal). There were economic reasons for non-blood relinking marriages: to keep the wealth and power within selected families.

type of marriage	Loka	Silba	Ragusa	Tur	Royal
blood-marriages	51	42	9	149	51
relinking-marriages	157	118	239	176	45

Number of individuals in genealogy Tur is much lower than in others, Silba and Ragusa are approximately of the same size, while Loka is much larger genealogy, what we must also take into account.

Networks

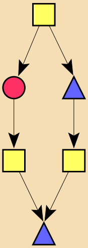
Pajek Project File
Unicode
Excel2Pajek
Multiple relations
Temporal network
Text → Networks
Acyclic networks
Citation networks
Genealogies
Clust. Coefficient
Islands
Communities
E-I Index
Comparing part.
Short Cycles
Example
Network models
Simulations
Other networks

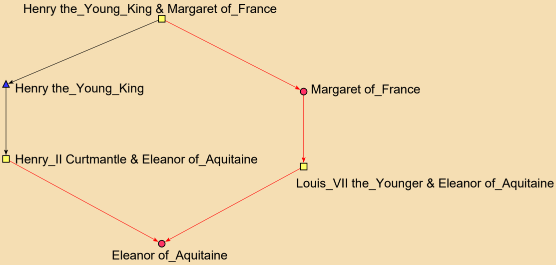
Frequencies normalized with number of couples in p-graph × 1000

pattern	Loka	Silba	Ragusa	Tur	Royal
A2	0.07	0.00	0.00	0.00	0.00
A3	0.07	0.00	0.00	0.00	2.64
A4.1	0.85	2.26	1.50	159.71	18.45
B4	3.82	11.28	10.49	98.28	6.15
A4.2	0.00	0.00	0.00	0.00	0.00
A5.1	0.64	3.16	2.00	36.86	11.42
A5.2	0.00	0.00	0.00	0.00	0.00
B5	1.34	4.96	23.48	46.68	7.03
A6.1	1.98	12.63	1.00	169.53	11.42
A6.2	0.00	0.90	0.00	0.00	0.88
A6.3	0.00	0.00	0.00	0.00	0.00
C6	0.71	5.41	9.49	36.86	4.39
B6.1	0.00	0.45	1.00	0.00	0.00
B6.2	1.91	17.59	31.47	130.22	10.54
B6.3	3.32	13.53	40.96	113.02	11.42
B6.4	0.00	0.00	2.50	7.37	0.00
$\sum$	14.70	72.17	123.88	798.53	84.36

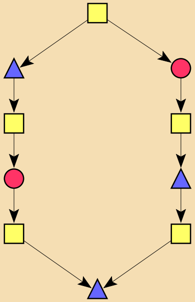
Bipartite p-graphs: Marriage between half-brother and half-sister

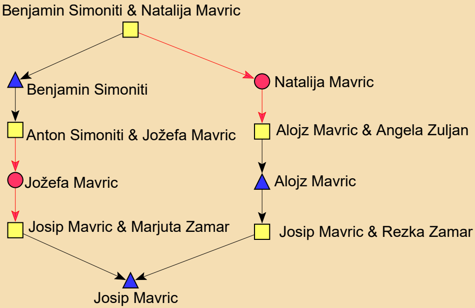
Using p-graphs we cannot distinguish persons married several times. In this case we must use bipartite p-graphs. Using bipartite p-graphs we can find marriages between half-brothers and half-sisters.





Bipartite p-graphs: Marriage among half-cousins





Other analyses

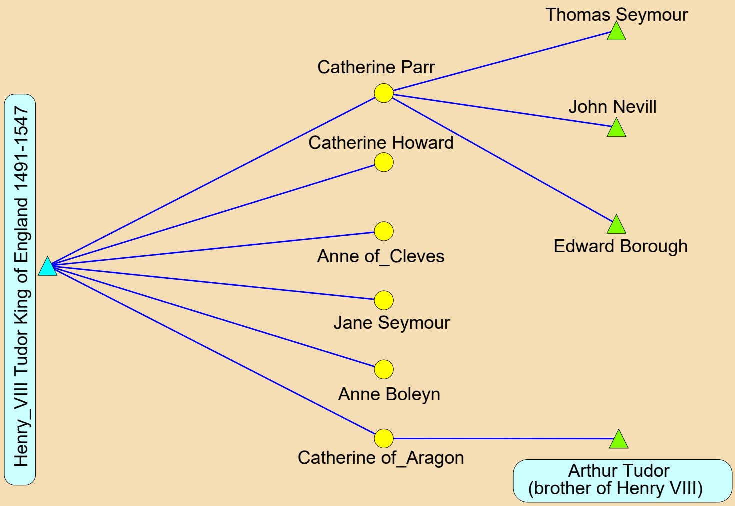
People collecting data about their families are interested in several other 'standard' analyses:

- changes in relinking patterns over time;
- special situations: persons married several times, persons having the highest number of children;
- checking whether the two persons are relatives and searching for the shortest genealogical path between them;
- searching for all predecessors/successors of selected person and searching for person with the largest number of known predecessors or successors;
- the largest difference in age between husband and wife,
  the oldest/youngest person at the time of marriage,
  the oldest/youngest person at the time of child's birth;
- searching for the longest patrilineage and matrilineage;
- special situations → errors made in data entry (network consistency check).

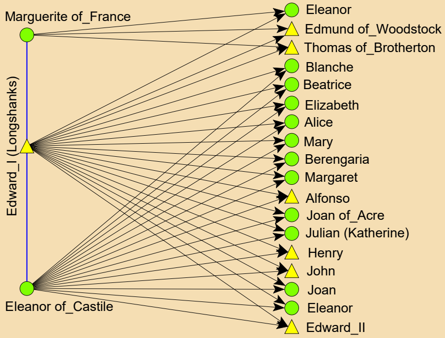
Networks

Pajek Project File
Unicode
Excel2Pajek
Multiple relations
Temporal network
Text → Networks
Acyclic networks
Citation networks
Genealogies
Clust. Coefficient
Islands
Communities
E-I Index
Comparing part.
Short Cycles
Example
Network models
Simulations
Other networks

The highest number of marriages



The highest number of children



English king Edward I (1237-1307) and his wife Eleanor (1241-1290) had 16 children who were born between 1255-1284 (in the picture a daughter without given name is missing). The youngest son (Edward) was the first among sons who survived a childhood. Eleanora had to try sixteen times to fulfill her most important duty as a queen: to give a birth to a men successor who later became a king. 10 out of 16 children died before age 10, only 3 of them lived longer than 40 years.

...Acyclic networks...

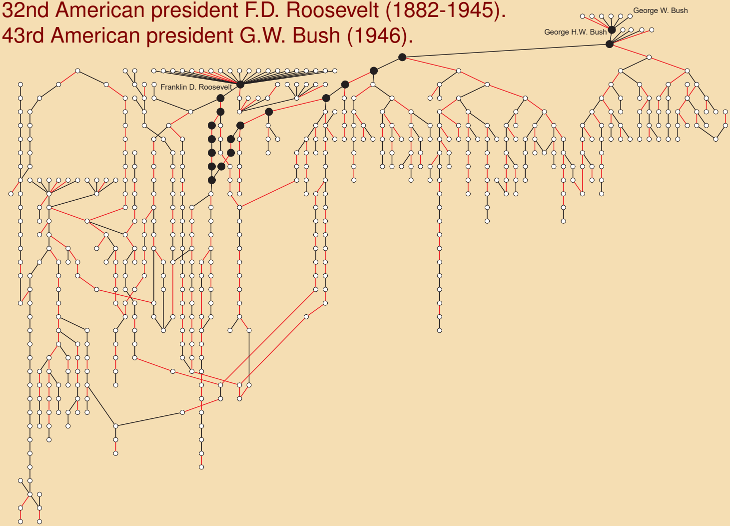
...Genealogies...

Networks

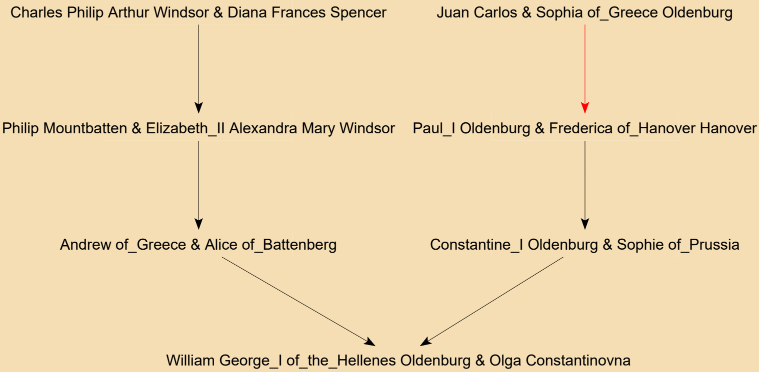
Pajek Project File
Unicode
Excel2Pajek
Multiple relations
Temporal network
Text → Networks
Acyclic networks
Citation networks
Genealogies
Clust. Coefficient
Islands
Communities
E-I Index
Comparing part.
Short Cycles
Example
Network models
Simulations
Other networks

Searching for kinshio relations

32nd American president F.D. Roosevelt (1882-1945).
43rd American president G.W. Bush (1946).



The shortest genealogical path between *Charles P. A. Windsor* (Prince of UK) and *Juan Carlos* (ex King of Spain) in Royal.ged



Clustering Coefficient...

Clustering coefficient is a ratio between number of links among neighbours of selected vertex and maximum number of links among neighbours (which we obtain if all neighbours are linked among themselves).

Let $\deg(v)$ denotes degree of vertex v, $|E(G_1(v))|$ number of lines among vertices in 1-neighborhood of vertex v, MaxDeg maximum degree of vertex in a network, and $|E(G_2(v))|$, number of lines among vertices in 1 and 2-neighborhood of vertex v.

- CC_1 – coefficients considering only 1-neighborhood:

$$CC_1(v) = \frac{2|E(G_1(v))|}{\deg(v)\cdot(\deg(v)-1)} \qquad CC_1'(v) = \frac{\deg(v)}{\mathrm{MaxDeg}} CC_1(v)$$

...Clustering Coefficient

- CC_2 – coefficients considering 2-neighborhood

$$CC_2(v) = \frac{|E(G_1(v))|}{|E(G_2(v))|} \qquad CC_2'(v) = \frac{\deg(v)}{\mathrm{MaxDeg}} CC_2(v)$$

If $\deg(v) \leq 1$ all coefficients for vertex v get missing value (999999998).

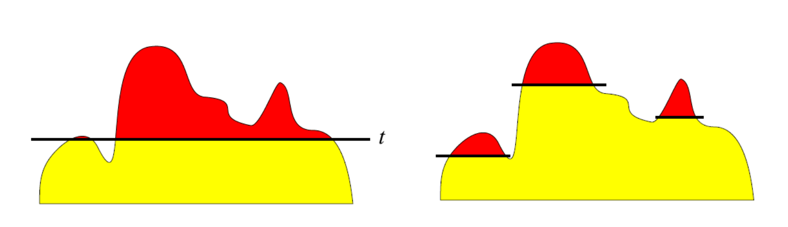
Watts-Strogatz Clustering Coefficient (Transitivity) and Network Clustering Coefficient are also reported.
The clustering coefficient or transitivity of a network is the proportion of all two-paths in the network that are closed.

Pajek command:

Network / Create Vector / Clustering Coefficients

Islands...

If we represent a given or computed value of vertices / links as a height of vertices / links and we immerse the network into a water up to selected level we get islands. Varying the level we get different islands.



...Islands

Line Islands:
Network / Create Partition / Islands / Line Weights

Vertex Islands (we need a Vector describing properties of vertices as additional input):
Operations / Network + Vector / Islands / Vertex Weights

Community Detection...

Communities - dense clusters for which there are more lines inside than among clusters (values of lines are taken into account too).
In Pajek two community detection methods are available:
Louvain method and **VOS Clustering**
When applying **Louvain method** we search for partition into clusters with the highest value of **modularity**. Modularity is defined in the following way:

$$Q = \frac{1}{2m} \sum_s (e_s - r * \frac{K_s^2}{2m})$$

- m – total number of lines in network,
- s – cluster (community),
- $e_s = \sum_{ij \in s} A_{ij}$ – 2 times the number of lines in community s
- $K_s = \sum_{i \in s} k_i$ – sum of degrees in community s
- r – **resolution parameter**, default value 1 means modularity as originally defined

...Community Detection

Similar method is **VOS Clustering**, where **VOS quality function** is taken into account instead of modularity.
Both methods are available in:
Network / Create Partition / Communities
Several parameters can be changed, but they are important only when we are analysing larger networks.
By changing *resolution parameter - r* we can get larger or smaller communities. By default resolution parameter is set to 1. Setting *r* larger than 1 means searching for larger number of smaller communities. Setting *r* smaller than 1 means searching for smaller number of larger communities.

E-I Index

Simple measure of how well clusters divide a network into cohesive subgroups is the **E-I Index: External-Internal Index**. The E-I Index subtracts the number of lines within clusters from the number of lines between the clusters. The difference is divided by the total number of lines. Line values can be taken into account.

Values of the E-I Index range from -1 to 1. If the E-I Index is -1, all lines are inside clusters whereas the value 1 means that all lines are between clusters. The value 0 indicates that the number of lines (or the sum of line values) between clusters equals the number of lines (or sum of line values) inside clusters. If a partition classifies vertices into cohesive groups well, lines should be within clusters rather than between clusters, so the E-I Index should be negative and preferably close to -1.

Operations / Network + Partition / Info / E-I Index

Comparing partitions...

Cramer coefficient

Communities obtained by both methods are usually very similar.
We can check this by
Partitions/Info
which computes **Cramer**, **Adjusted Rand** and **Rajski coefficients** for comparison of two partitions.

Several measures can be computed to compare two partitions which represent nominal properties (usually represented as a contingency table):

Partitions / Info / Cramer's V, Rajski, Adjusted Rand Index

Two nominal variables can be compared using Cramer coefficient (*Cramer's V*).

$$V = \sqrt{\frac{\chi^2}{n(k-1)}}$$

...Comparing partitions...

Rajski coefficients...

Rajski coefficient (1964) is constructed using enthropy: Lets take two nominal variables X and Y. Variable X has n different values, variable Y has m different values.

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i)$$

$$H(Y) = -\sum_{i=1}^{m} p(y_i) \log_2 p(y_i)$$

and

$$H(XY) = -\sum_{i=1}^{n} \sum_{j=1}^{m} p(x_i, y_j) \log_2 p(x_i, y_j)$$

Information between variables X and Y is defined in the following way:

$$I(X, Y) = H(X) + H(Y) - H(XY)$$

Information $I(X, Y)$ gets value 0, when it holds for each pair x_i and y_j: $p(x_i, y_j) = p(x_i)p(y_j)$, what means, that the two variables are undependant.
Information $I(X, Y)$ gets maximum value, when functional relationship exists between the two variables – in each row and each column of corresponding contingency table there is at most one non zero element. Then it holds:

$$H(X) = H(Y) = H(XY) = I(X, Y)$$

Information I(X,Y) is a measure of functional dependancy between X and Y.

Rajski coefficients:

$$R(X \leftrightarrow Y) = \frac{I(X,Y)}{H(XY)}$$

$$R(X \rightarrow Y) = \frac{I(X,Y)}{H(Y)}$$

$$R(X \leftarrow Y) = \frac{I(X,Y)}{H(X)}$$

All three coefficients get values in range 0 to 1. Value 0 means that variables are undependant.
$R(X \rightarrow Y) = 1$, when Y is a function of X,
$R(X \leftarrow Y) = 1$, when X is a function of Y in
$R(X \leftrightarrow Y) = 1$, when variables determine each other (in both directions).

Example 1:

	y_1	y_2	y_3	Sum
x_1	2	2	1	5
x_2	2	1	2	5
Sum	4	3	3	10

$p(x_i, y_j)$	y_1	y_2	y_3	$p(x_i)$
x_1	0.2	0.2	0.1	0.5
x_2	0.2	0.1	0.2	0.5
$p(y_j)$	0.4	0.3	0.3	1

$$R(X \leftrightarrow Y) = 0.0194$$

$$R(X \rightarrow Y) = 0.0312$$

$$R(X \leftarrow Y) = 0.0490$$

All three coefficients are low, we cannot predict the value of one variable if we know the value of the other.

Example 2:

	y_1	y_2	y_3	Sum
x_1	0	3	0	3
x_2	4	0	3	7
Sum	4	3	3	10

$p(x_i, y_j)$	y_1	y_2	y_3	$p(x_i)$
x_1	0	0.3	0	0.3
x_2	0.4	0	0.3	0.7
$p(y_j)$	0.4	0.3	0.3	1

$$R(X \leftrightarrow Y) = 0.5610$$

$$R(X \rightarrow Y) = 0.5610$$

$$R(X \leftarrow Y) = 1$$

Variable X is a function of Y: if we know value of Y we can predict value of X.
However variable Y is not a function of X.

Example 3:

	y_1	y_2	y_3	Sum
x_1	4	0	0	4
x_2	0	0	3	3
x_3	0	3	0	3
Sum	4	3	3	10

$$R(X \leftrightarrow Y) = 1$$

$$R(X \rightarrow Y) = 1$$

$$R(X \leftarrow Y) = 1$$

Variable X is a function of Y. Variable Y is a function of X.

Example 4:

	y_1	y_2	Sum
x_1	2	2	4
x_2	2	2	4
Sum	4	4	8

$$R(X \leftrightarrow Y) = 0$$

$$R(X \rightarrow Y) = 0$$

$$R(X \leftarrow Y) = 0$$

Variables X and Y are independant.

...Comparing partitions

Adjusted Rand Index

Adjusted Rand index [edit]

The adjusted Rand index is the corrected-for-chance version of the Rand index.[1][2][3] Such a correction for chance establishes a baseline by using the expected similarity of all pair-wise comparisons between clusterings specified by a random model. Traditionally, the Rand Index was corrected using the Permutation Model for clusterings (the number and size of clusters within a clustering are fixed, and all random clusterings are generated by shuffling the elements between the fixed clusters). However, the premises of the permutation model are frequently violated; in many clustering scenarios, either the number of clusters or the size distribution of those clusters vary drastically. For example, consider that in K-means the number of clusters is fixed by the practitioner, but the sizes of those clusters are inferred from the data. Variations of the adjusted Rand Index account for different models of random clusterings.[4]

Though the Rand Index may only yield a value between 0 and +1, the adjusted Rand index can yield negative values if the index is less than the expected index.[5]

The contingency table [edit]

Given a set S of n elements, and two groupings or partitions (*e.g.* clusterings) of these elements, namely $X = \{X_1, X_2, \ldots, X_r\}$ and $Y = \{Y_1, Y_2, \ldots, Y_s\}$, the overlap between X and Y can be summarized in a contingency table $[n_{ij}]$ where each entry n_{ij} denotes the number of objects in common between X_i and Y_j : $n_{ij} = |X_i \cap Y_j|$.

$X \setminus Y$	Y_1	Y_2	$\cdots$	Y_s	sums
X_1	n_{11}	n_{12}	$\cdots$	n_{1s}	a_1
X_2	n_{21}	n_{22}	$\cdots$	n_{2s}	a_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
X_r	n_{r1}	n_{r2}	$\cdots$	n_{rs}	a_r
sums	b_1	b_2	$\cdots$	b_s	

Definition [edit]

The original Adjusted Rand Index using the Permutation Model is

$$ARI = \frac{\sum_{ij} \binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{n}{2}}$$

where n_{ij}, a_i, b_j are values from the contingency table.

https://en.wikipedia.org/wiki/Rand_index

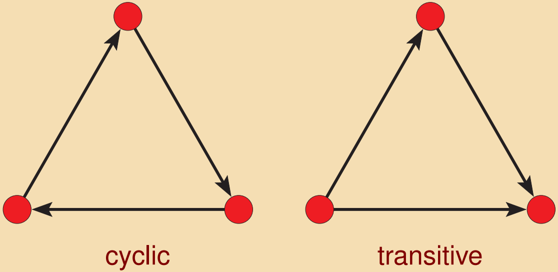
Short Cycles...

Three Rings

How many times each line belongs to predefined three ring. Ring counts are stored as line values.

Network / Create New Network / with Ring Counts stored as Line Values / 3-Rings

- **Undirected** – for undirected networks – count undirected 3-rings.
- **Directed** – for directed networks – count **cyclic**, **transitive**, or all 3-rings, or count how many times each line is a transitive shortcut.

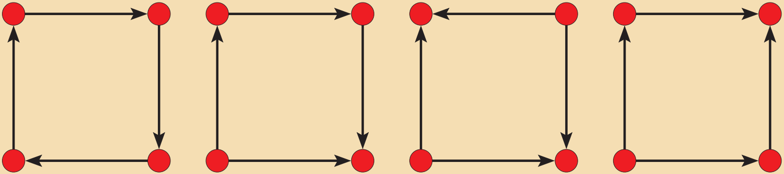


...Short Cycles

Four Rings

How many times each line belongs to predefined four ring. Ring counts are stored as line values.
Network / Create New Network / with Ring Counts stored as Line Values / 4-Rings

- **Undirected** – for undirected networks – count undirected 4-rings.
- **Directed** – for directed networks – count **cyc**lic, **dia**monds, **gen**ealogical, **tra**nsitive, or all 4-rings, or count how many times each line is a transitive shortcut.

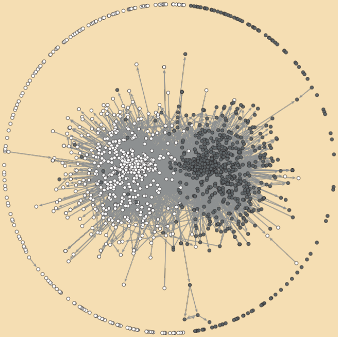


cyclic   transitive   genealogical   diamond

Example Network...

Political blogosphere, United States, February 8, 2005

Network of hyperlinks between 1,490 political blogs in February 2005 discussing the 2004 election in the United States, compiled by L. A. Adamic and N. Glance. An arc between two blogs represents a reference in a blogroll or in a post on the blog's front page.



We will use a simplified - undirected version of this network and a partition into liberal and conservative blogs (**Political-blogs.paj**).

...Example Network...

Political blogosphere, United States, February 8, 2005

Some properties of the blogosphere network:

- number of weakly connected components (WCC) = 268
- size of the largest WCC = 1222 vertices (82%)
- average degree = 22.436
- diameter = 8
- average distance among reachable pairs = 2.738
- clustering coefficient (transitivity) = 0.226
- degree centralization = 0.221
- betweenness centralization = 0.065
- E-I Index = -0.81

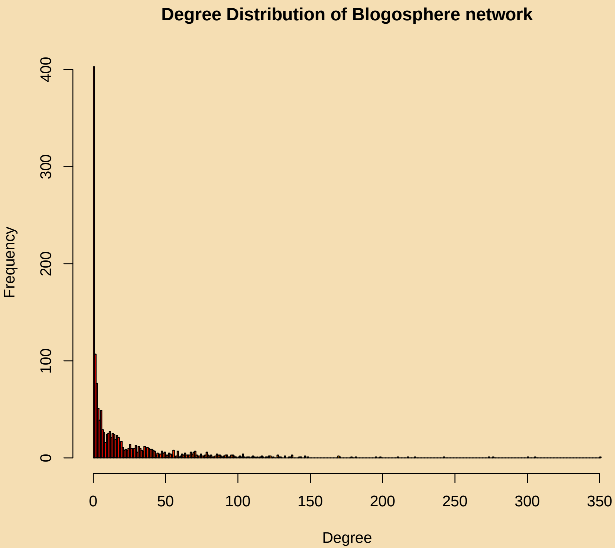
Average distance is computed using the following command:

Network / Create Vector / Distribution of Distances*

It computes distribution of lengths of the shortest paths (distances) and average path length (average distance) among all reachable pairs of vertices.

...Example Network

Political blogosphere, United States, February 8, 2005



Bernoulli model

The most common implementation of random model assigns a line to each pair of vertices independently, with a fixed probability. Independence means that the probability that a pair of vertices is connected by a line is independent from the presence or absence of lines among other pairs.

Network / Create Random Network / Bernoulli/Poisson

The original Erdös–Rényi model fixes the exact number of arcs (instead of probability of an arc), so this characteristic does not vary among random graphs; it is a condition just like the number of vertices. Result is always a directed network.

Network / Create Random Network / Total No. of Arcs

The two models are very unlikely to appear in social networks. In the two models individual does not care 'whom (s)he will select as a friend'.

...Standard network models...

...Bernoulli and Erdös–Rényi...

Some properties of Bernoulli networks:
An average degree over 1 produces with high probability a graph containing one large weakly connected component (WCC) while all other components are about equally small.
The size of the large component grows with both the size of the graph and the average degree; with average degree 1.5, it is already expected to contain over 50% of all vertices. We call it a giant component.
In contrast, average degree below unity is expected to create random graphs containing only small components.

Diameter of such networks is relativelly small, a rough estimate is:

$$Diameter_{expected} = \frac{\ln(n)}{\ln(c)}$$

n is number of vertices, and c average degree.

...Standard network models...

...Bernoulli and Erdös–Rényi...

In a Bernoulli random graph, the expected proportion of transitive triples over connected triples (Clustering coefficient - transitivity) is

$$CC_{expected} = \frac{c}{n-1}$$

which tends toward zero in larger sparse networks.

An obvious shortcoming of the Bernoulli random graph model is its low and network-size-dependent clustering. We need models yielding random graphs of any size with clustering coefficients in the range usually found in social networks, roughly put between 0.05 and 0.50.

The next model which we will describe will solve this problem.

...Standard network models...

...Bernoulli and Erdös–Rényi...

For undirected Bernoulli network with the same number of vertices (1490) and average degree (22.436) as we have in blogosphere network, we get the following expected values:

$$Diameter_{expected} = \frac{\ln(n)}{\ln(c)} = \frac{\ln(1490n)}{\ln(22.436)} = 2.4$$

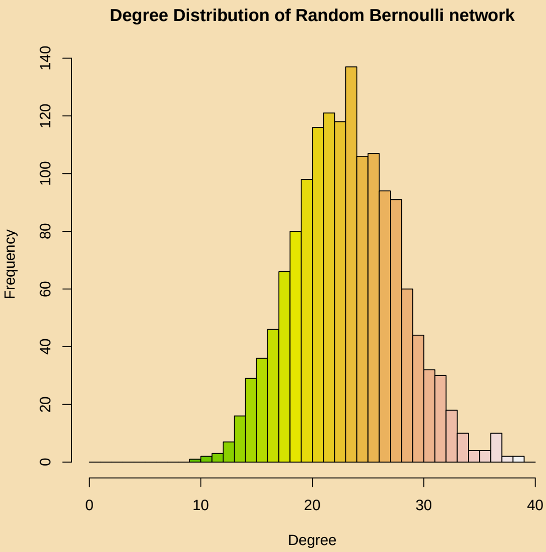
$$CC_{expected} = \frac{c}{n-1} = \frac{22.436}{1490-1} = 0.015$$

The observed diameter (8) is approx. three times larger than the one expected in Bernoulli network (2.4).
The observed transitivity (0.226) is around ten times larger than the one expected in Bernoulli network (0.015).

Generate some random Bernoulli networks with the the same number of vertices (1490) and average degree (22.436) as we have in the blogosphere network and check diameter and clustering coefficients which you obtain. Are the obtained diameter and clustering coefficient close to expected values? What is the number of WCC and the size of the largest WCC?

...Standard network models...

...Bernoulli and Erdös–Rényi



...Standard network models...

Degree conditional Bernoulli model...

We can store degrees in a partition and request to generate a random Bernoulli network where vertices will have the same degrees as defined in a partition.
The partition also determines the number of vertices in the random graph, so the user need not specify this property.

Partition / Make Network / Random Network

After executing this command the obtained network might containe loops and multiple lines, so we usually remove them:

Network / Create New Network / Transform / Remove / Loops

Network / Create New Network / Transform / Remove / Multiple Lines / Single Line

...Standard network models...

...Degree conditional Bernoulli model...

Degrees can be provided manually for each vertex, or we can compute degrees for a given network:

Network / Create Partition / Degree

and generate random network with the same degree distribution:

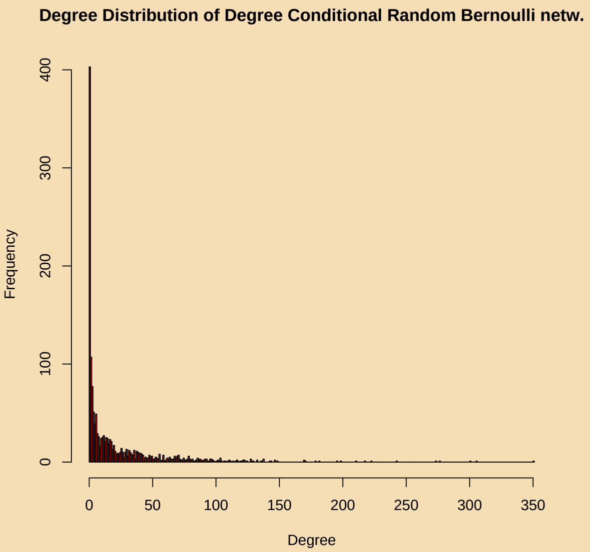
Partition / Make Network / Random Network

In the obtained network all vertices will have the same degrees as vertices in the original network, but lines will be distributed differently.

Generate degree conditional Bernoulli model with the same degree distribution as Blogoshpere network. Do not forget to remove loops and multiple lines. Compare structural features of the obtained random network with the Blogosphere network. Which structural features of both networks are equal, similar, which are different?

...Standard network models...

...Degree conditional Bernoulli model



...Standard network models...

Small world experiment

A psychologist Stanley Milgram made his experiment with letters in 1967. The letter should reach a target person. The persons involved in experiment were asked to send the letter with these instructions to the target person (if they personally know him/her) or (if they do not know him/her personally) to their friend who was more likely to know the target. The letter was sent from Boston (Massachusetts) to target person Omaha (Nebraska).
The average length of the successful paths was 6: *six degrees of separation*.

https://en.wikipedia.org/wiki/Small-world_experiment

The average path length on the internet is 19 clicks.

The networks in which the average shortest path length is small are called **Small Worlds**.

Small world (Watts-Strogatz model): Indivuduals are linked to those who live in the neighbourhood and some others further away.
The original small-world model puts all vertices on a circle and connects each vertex to a fixed number of its neighbors in spatial sense: the vertices nearest in the plane. If the number of connected neighbors exceeds 2, triangles appear because each vertex is linked to its neighbor and its neighbor's neighbor.
The expected clustering coefficient (network transitivity) is determined only by the number of neighbors each vertex is linked to at each side (r) and it is easy to calculate:

$$CC_{expected} = \frac{3r-3}{4r-2}$$

If the number of neighbors on either side is set to 1, the clustering coefficient reaches its minimum 0, whereas the coefficient tends toward .75 for a very large number of neighbors.

...Standard network models...

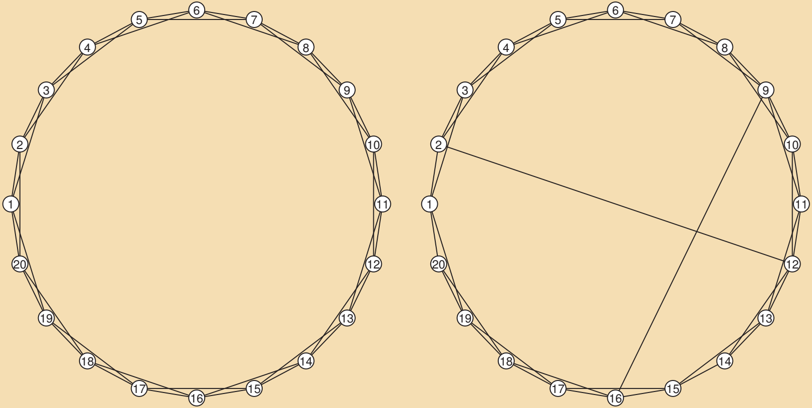
...Small world...

This approach, however, raises a new problem: A large graph containing only local lines has *an average path distance* much higher than the ones encountered in social networks, whereas the small-world phenomenon argues that even in the network containing the entire world population, people are acquainted in a maximum of six steps.

This problem is solved by replacing one endpoint of a small proportion of the local lines by a random vertex (*rewiring*). Rewiring as little as 1 to 10 percent of the local lines suffices to obtain the small-world phenomenon of *low average path distance*. A low proportion of rewired lines does not change the *density* and *average degree* of the graph. It hardly changes its clustering, so the high *clustering characteristic* of social networks is preserved.

...Standard network models...

...Small world...



Small world model is built using:

Network / Create Random Network / Small World

Standard network models

...Small world...

The average degree in the blogs network is 22.4, so we can compare this network with a small-world random graph with vertices linked to the **eleven** nearest neighbors ($r = 22.436/2$), which has about the same average degree. The expected value of the clustering coefficient for this random graph is

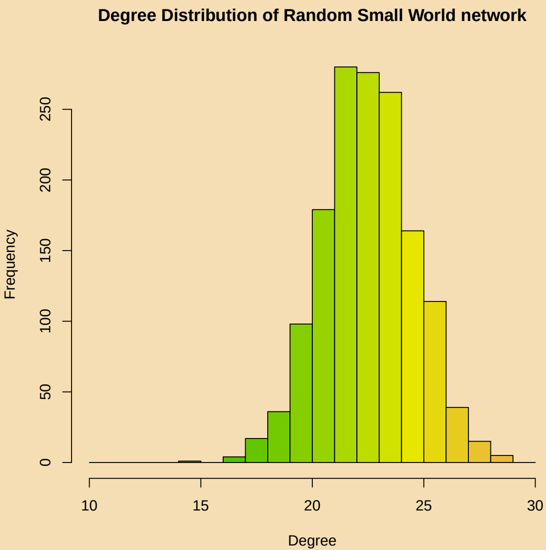
$$CC_{expected} = \frac{3r-3}{4r-2} = \frac{3*11-3}{4*11-2} = 0.71$$

This is in the limit of no rewiring (0.75). The observed value (0.226) is quite a bit lower, the reason probably being that vertex degree is not more or less equal for all vertices as assumed by the small-world random model; instead, it is highly skewed.

Generate some random Small world networks with the same number of vertices (1490) and **eleven** nearest neighbors as we have in blogosphere network. Set probability of rewiring somewhere in range 0.01 to 0.1. Check clustering coefficients which you obtain. Is the obtained clustering coefficient close to expected values?

...Standard network models...

...Small world



...Standard network models...

Preferential attachment - Scale free...

All models mentioned till now still do not have characteristics of real social networks. Such networks usually have some vertices with very high degree.
Preferential attachment models solve this problem by simply assuming that vertices prefer to link to vertices with higher degree. This is a network variant of popular sayings such as “the rich get richer”.
The degree distribution is highly skewed to the right.

Network / Create Random Network / Scale Free

Probability that vertex v is selected as the second endpoint of the new line is determined by:

$$\Pr(v) = \alpha \frac{\mathrm{indeg}(v)}{|E|} + \beta \frac{\mathrm{outdeg}(v)}{|E|} + \gamma \frac{1}{|V|}$$

where $\alpha + \beta + \gamma = 1$.

...Standard network models...

...Preferential attachment - Scale free...

In each step of building scale free network we add a new vertex and some (average degree) lines.

If *Adding=Free* is checked (free adding of lines) then both endpoints of the new line will be selected at random and vertices may remain isolated in the resulting random network.
If *Adding=Free* is not checked then in each step the new added vertex will be connected to one of already exixting vertices.

For undirected network we select only α and set $\beta = \alpha$.
If $\alpha = 0.5$, then $\alpha + \beta = 1$: we always draw vertex proportional to to its degree: the higher the degree of vertex, the larger probability that this vertex will be selected as the second endpoint of the line.
If $\alpha = 0.25$ then $\alpha + \beta = 0.5$: a vertex is drawn with 0.5 probability to its degree and 0.5 uniformly from all vertices (like in Bernoulli model).

...Standard network models...

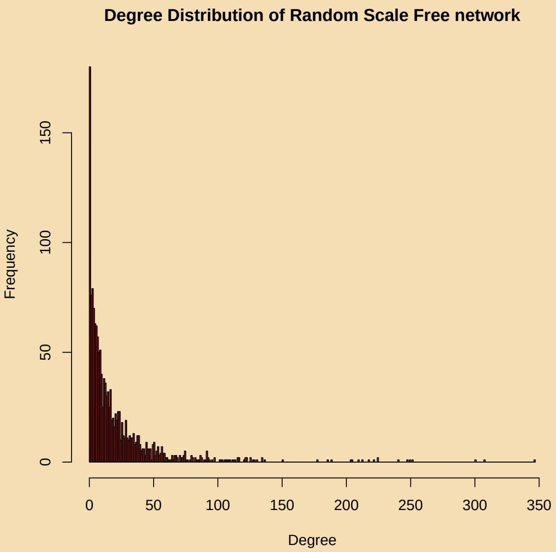
...Preferential attachment - Scale free...

Random graphs generated by the Scale Free commands may contain multiple lines. If these are not desired – for example, because the clustering coefficient cannot handle multiple lines – they should be removed with: **Network** / **Create New Network** / **Transform** / **Remove** / **Multiple Lines** / **Single Line**.

Generate some random Scale free networks (and remove multiple lines) with the same number of vertices (1490) and average degree (22.436) as we have in blogosphere network. Forα (probability that vertex will be drawn proportionallly to its degree) try different values in interval 0..0.5 (note that this number is multiplied by 2 for undirected networks).
Try random generator with option: **Network** / **Create Random Network** / **Scale Free** / **Adding** / **Free**
once checked and once unchecked. What is difference (e.g. number of weakly connected components)?

...Standard network models...

...Preferential attachment - Scale free



For the following networks:

- Blogosphere network
- Bernoulli network ($n = 1490, \overline{deg} = 2.436$)
- Small world ($n = 1490, r = 11, p_{rew} = 0.1$)
- Small world ($n = 1490, r = 11, p_{rew} = 0.2$)
- Scale free ($n = 1490, \overline{deg} = 2.436, \alpha = 0.5$)
- Scale free ($n = 1490, \overline{deg} = 2.436, \alpha = 0.5, Adding = Free$)

compute the following properties:

- number of weakly connected components
- size of the largest weakly connected component
- diameter
- average distance
- clustering coefficient - transitivity
- degree centralization
- betweenness centralization

Write results in a table and compare them. For each property find out which of the random networks is the most similar to Blogosphere network.

For each of the above networks compute degrees:
Network / Create Vector / Centrality / Degree

and draw degree distrubutions in R:

Tools / R / Send to R / Current Vector
R: hist(v??, breaks= 0:???)

Details: Picture of degree distrubution of Small World network was obtained using some additional parameters for drawing histograms in R:

```
par(bg="wheat")
hist(v1,breaks=10:30,col=terrain.colors(30),
main="Degree Distribution of Small World network",
xlab='Degree', right=FALSE)
```

Monte Carlo simulations...

Networks

Pajek Project File
Unicode
Excel2Pajek
Multiple relations
Temporal network
Text→Networks
Acyclic networks
Citation networks
Genealogies
Clust. Coefficient
Islands
Communities
E-I Index
Comparing part.
Short Cycles
Example
Network models
Simulations
Other networks

To be sure about obtained parameters we have to generate more (e.g. 100) such random networks and take the average value.

Network / Create New Network / Random Network / ...

Lets generate 99 more such networks (to get 100 networks):
Macro / Repeat Last Command (F10) 99

Examine info for the first obtained network:
Network / Info / General

Examine info for the rest 99:
Macro / Repeat Last Command (F10) 99

Result is also a new vector, which presents aggregate results for all 100 networks, e.g. vector of dimension 100, where each cell presents average degree for each individual network.
Using **Vector / Info** we can examine descriptive statistics (*arithmetic mean, median, standard deviation, quantiles*) for selected property (e.g. average degree). Similarly we can make an overview for other network properties (e.g. average distance, diameter...).

...Monte Carlo simulations

Repeat the overview assigment but instead of generating only one random network of selected type, generate 100 such networks and provide the aggregate results (results obtained by **Macro / Repeat Last Command**).

Other large networks

Signed networks - relaxed balance
Coauthorship networks (e.g. Erdos).
Two mode networks - direct analysis:
- Blockmodeling
- Important Vertices