Tree and Network Analysis and Visualization

Dr. Katy Börner
Cyberinfrastructure for Network Science Center, Director
Information Visualization Laboratory, Director
School of Library and Information Science
Indiana University, Bloomington, IN
http://info.slis.indiana.edu/~katy

With special thanks to Kevin W. Boyack, Micah Linnemeier, Russell J. Duhon, Patrick Phillips, Joseph Biberstine, Chintan Tank Nianli Ma, Hanning Guo, Mark A. Price, Angela M. Zoss, and Scott Weingart

Guest Lecture in S604/S764 Information Networks by Staša Milojević
November 14, 2011

12 Tutorials in 12 Days at NIH—Overview

1st Week

1. Science of Science Research
2. Information Visualization
3. CIShell Powered Tools: Network Workbench and Science of Science Tool

2nd Week

4. Temporal Analysis—Burst Detection
5. Geospatial Analysis and Mapping
6. Topical Analysis & Mapping

3rd Week

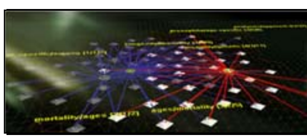
7. **Tree Analysis and Visualization**
8. **Network Analysis**
9. **Large Network Analysis**

4th Week

10. Using the Scholarly Database at IU
11. VIVO National Researcher Networking
12. Future Developments

[#07] Tree Analysis and Visualization

- General Overview
- Designing Effective Tree Visualizations
- Notions and Notations
- Sci2-Reading and Extracting Trees
- Sci2-Visualizing Trees
- Outlook



Sample Trees and Visualization Goals & Objectives

Sample Trees

Hierarchies
- File systems and web sites
- Organization charts
- Categorical classifications
- Similarity and clustering

Branching Processes
- Genealogy and lineages
- Phylogenetic trees

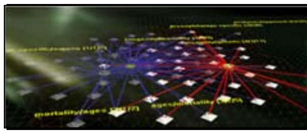
Decision Processes
- Indices or search trees
- Decision trees

Goals & Objectives

Representing hierarchical data
- Structural information
- Content information

Objectives
- Efficient Space Utilization
- Interactivity
- Comprehension
- Esthetics

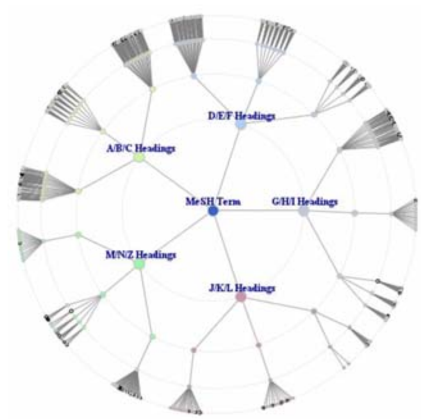
Pat Hanrahan, Stanford U
http://www-graphics.stanford.edu/~hanrahan/talks/todrawatree/



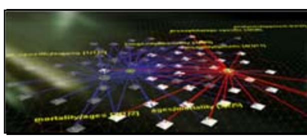
Radial Tree – How does it work?

See also http://iv.slis.indiana.edu/sw/radialtree.html

- All nodes lie in concentric circles that are focused in the center of the screen.
- Nodes are evenly distributed.
- Branches of the tree do not overlap.



Greg Book & Neeta Keshary (2001) Radial Tree Graph Drawing Algorithm for Representing Large Hierarchies. University of Connecticut Class Project.



Radial Tree – Pseudo Algorithm

Circle Placement

Maximum size of the circle corresponds to minimum screen width or height.

Distance between levels d := radius of max circle size / number of levels in the graph.

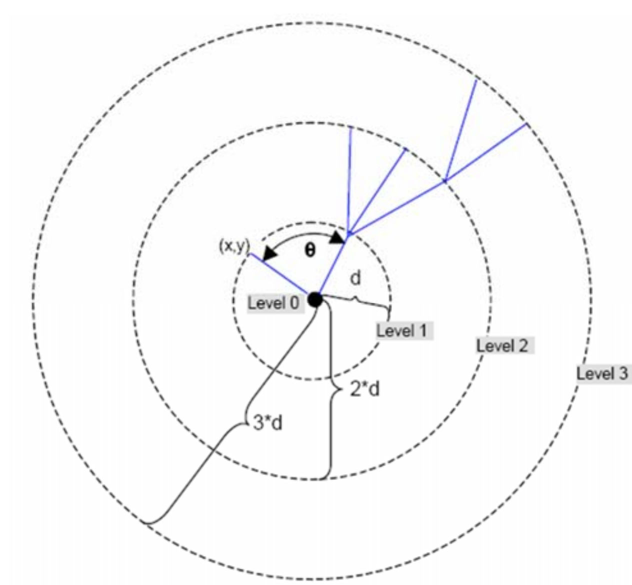
Node Placement

Level 0

The root node is placed at the center.

Level 1

All nodes are children of the root node and can be placed over all the 360° of the circle - divide 2pi by the number of nodes at level 1 to get angle space between the nodes on the circle.



Radial Tree – Pseudo Algorithm cont.

Levels 2 and greater

Use information on number of parents, their location, and their space for children to place all level x nodes.

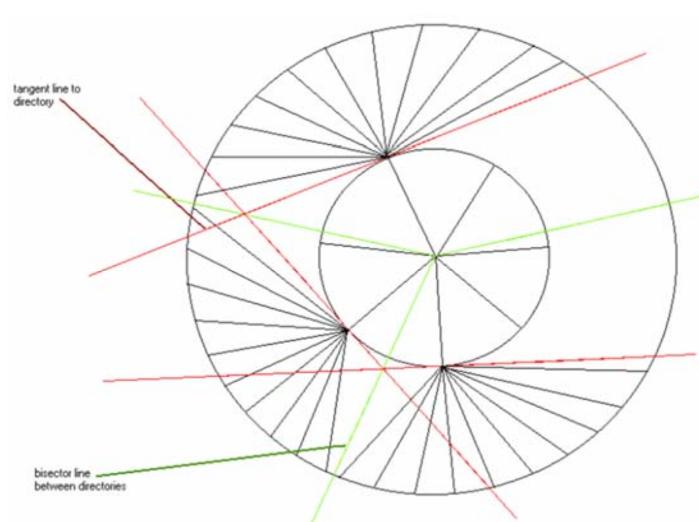
Loop through the list of parents and then loop through all the children for that parent and calculate the child's location relative to the parent's, adding in the offset of the limit angle.

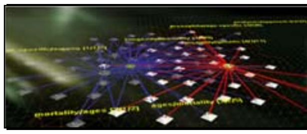
After calculating the location, if there are any directories at the level, we must calculate the bisector and tangent limits for those directories.

Radial Tree – Pseudo Algorithm cont.

We then iterate through all the nodes at level 1 and calculate the position of the node

Bisector Limits

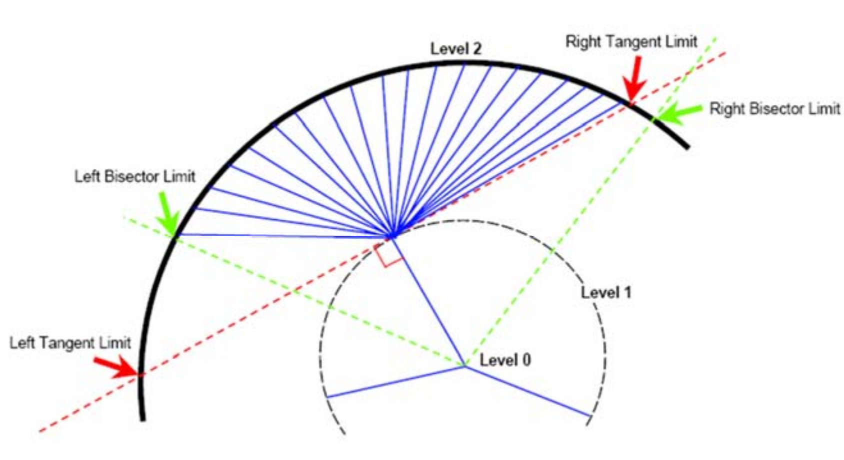


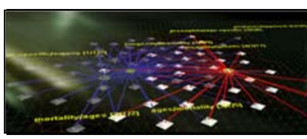


Radial Tree – Pseudo Algorithm cont.

Tangent and bisector limits for directories

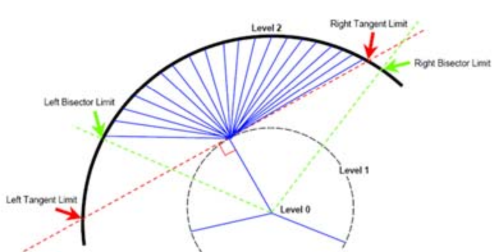
Between any two directories, a bisector limit is calculated to ensure that children do not overlap the children of an adjacent directory.





Radial Tree – Pseudo Code

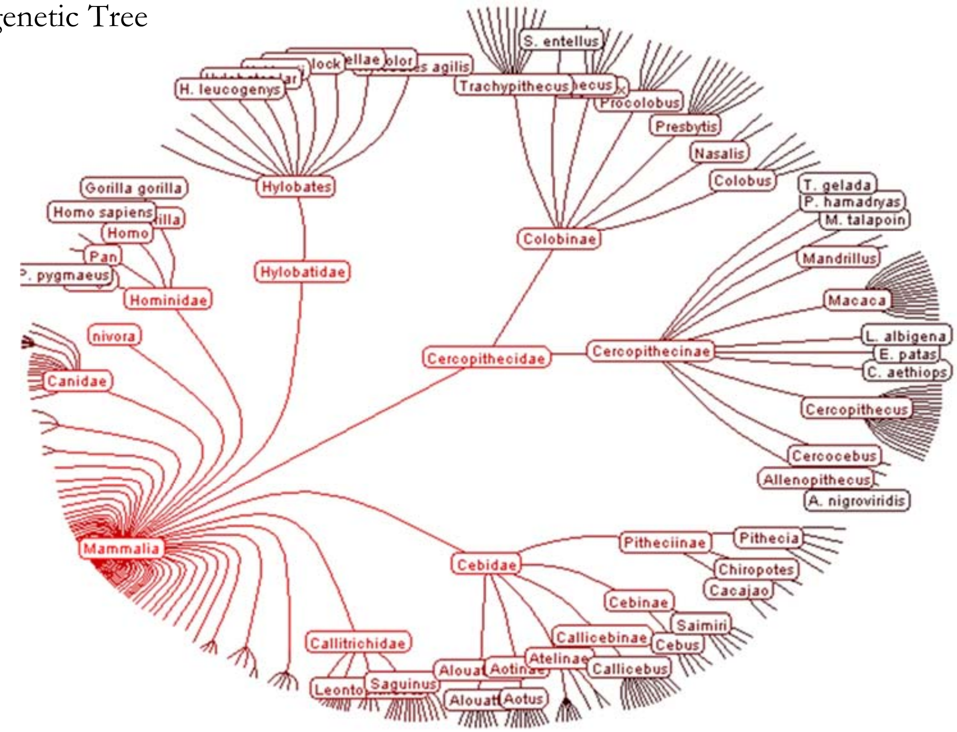
```
Loop through each level in the data structure

Switch level

        case 0:
                Find the center of the drawing area, to center the graph
                Set RootNode = CenterX, CenterY

        case 1:
                AngleSpace = (2Π radians/NumNodesAtThisLevel)
                Loop through all nodes at this level
                        Calculate x,y positions:
                        If (Node.type == Parent)
                                Calculate bisector limits and tangent limits for the node
                End loop
        case 2:
                Nodes in levels two and higher must be grouped according to their parent.
                Loop through all nodes at this level and get a list of the parent nodes
                And get the number of children for each parent
                Calculate the AngleSpace for each parent:
                        AngleSpace = (leftLimit – rightLimit)/NumNodesForThisParent
                Foreach parent
                        Loop through all nodes for that parent
                                Calculate x,y position for the child node
                                If (childnode.type == Directory)
                                        Calculate bisector and tangent limits
                        End loop
                End foreach

End switch
```



Hyperbolic Tree – How does it work?

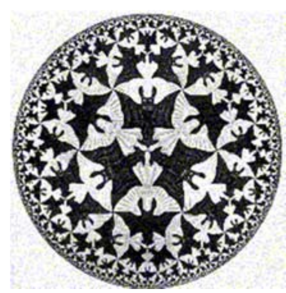
See also http://sw.slis.indiana.edu/sw/hyptree.html

Phylogenetic Tree



Hyperbolic Geometry

Inspired by Escher's Circle Limit IV (Heaven and Hell), 1960.

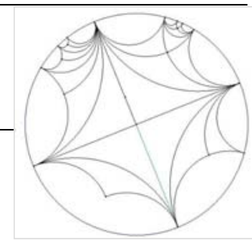


- Focus+context technique for visualizing large hierarchies
- Continuous redirection of the focus possible.

The hyperbolic plane is a non-Euclidean geometry in which parallel lines diverge away from each other. This leads to the convenient property that the circumference of a circle on the hyperbolic plane grows exponentially with its radius, which means that exponentially more space is available with increasing distance.

J. Lamping, R. Rao, and P. Pirolli (1995) A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. Proceedings of the ACM CHI '95 Conference - Human Factors in Computing Systems, 1995, pp. 401-408.

Hyperbolic Tree Layout



2 Steps:

Recursively lay out each node based on local information.

- A node is allocated a wedge of the hyperbolic plane, angling out from itself, to put its descendants in.
- It places all its children along an arc in that wedge, at an equal distance from itself, and far enough out so that the children are some minimum distance apart from each other.
- Each of the children then gets a sub-wedge for its descendants. (Because of the divergence of parallel lines in hyperbolic geometry, each child will typically get a wedge that spans about as big an angle as does its parent's wedge.)

Map hyperbolic plane onto the unit disk

Poincar´e model is a canonical way of mapping the hyperbolic plane to the unit disk. It keeps one vicinity in the hyperbolic plane in focus at the center of the disk while the rest of the hyperbolic plane fades off in a perspective-like fashion toward the edge of the disk.

Poincar´e model preserves the shapes of fan-outs at nodes and does a better job of using the screen real-estate.

Change of Focus – Animated Transitions

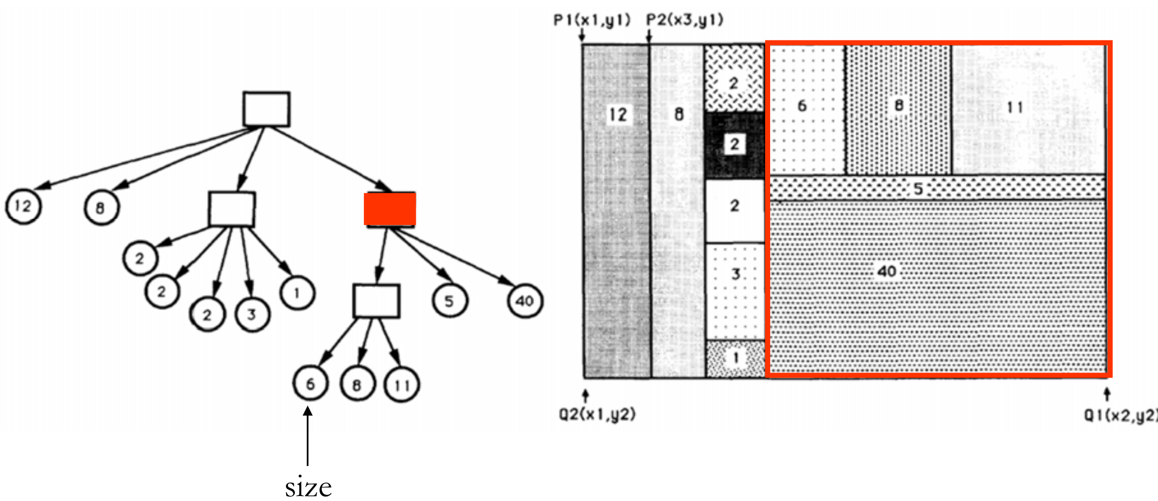
Node & Edge Information

Treemap – How does it work?

See also http://sw.slis.indiana.edu/sw/treemap.html

Shneiderman, B. (1992) [Tree visualization with tree-maps: 2-d space-filling approach](). ACM Transactions on Graphics 11, 1 (Jan. 1992), pp 92 - 99. See also [http://www.cs.umd.edu/hcil/treemaps/]()

Treemaps – Layout



Ben Shneiderman, Tree Visualization with Tree-Maps: 2-d Space-Filling Approach

Treemap – Pseudo Code

```
Input
Tree root & a rectangular area defined by upper left and lower right
coordinates Pl(xl, yl), Q1(x2, y2).

Recursive Algorithm
active_node := root_node;
partitioning_direction := horizontal; // nodes are partitioned vertically at
   even levels and horizontally at odd levels

Tremap(active_node) {
   determine number n of outgoing edges from the active_node;
   if (n<1)
        end;
   if (n>1) {
        divide the region [xl, x2] in partitioning_direction were the size of
        the n partitions correspond to their fraction
        (Size(child[i])/Size(active)) of the total number of bytes
        in the active_node;
        change partitioning_direction;
        for (1<=i<=n) do
                Treemap(child[i]);
   }
```

Treemap – Properties

Strengths

- Utilizes 100% of display space
- Shows nesting of hierarchical levels.
- Represents node attributes (e.g., size and age) by area size and color
- Scalable to data sets of a million items.

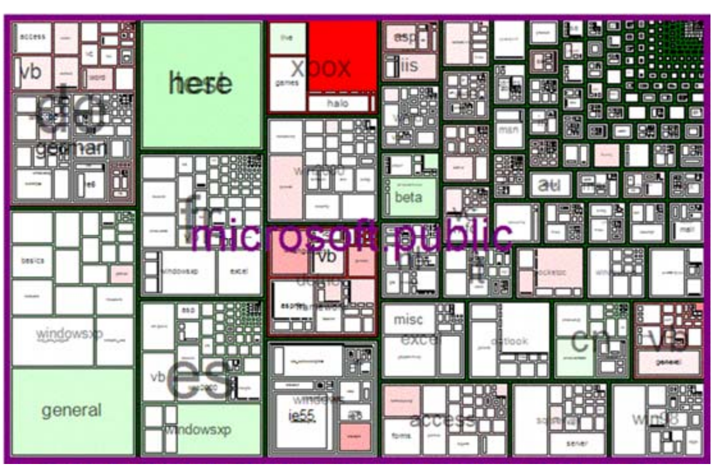
Weaknesses

- Size comparison is difficult
- Labeling is a problem.
- Cluttered display
- Difficult to discern boundaries
- Shows only leaf content information

Treemap – Algorithm Improvements

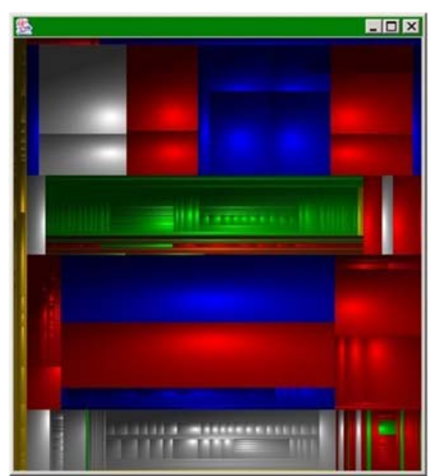
Sorted treemap

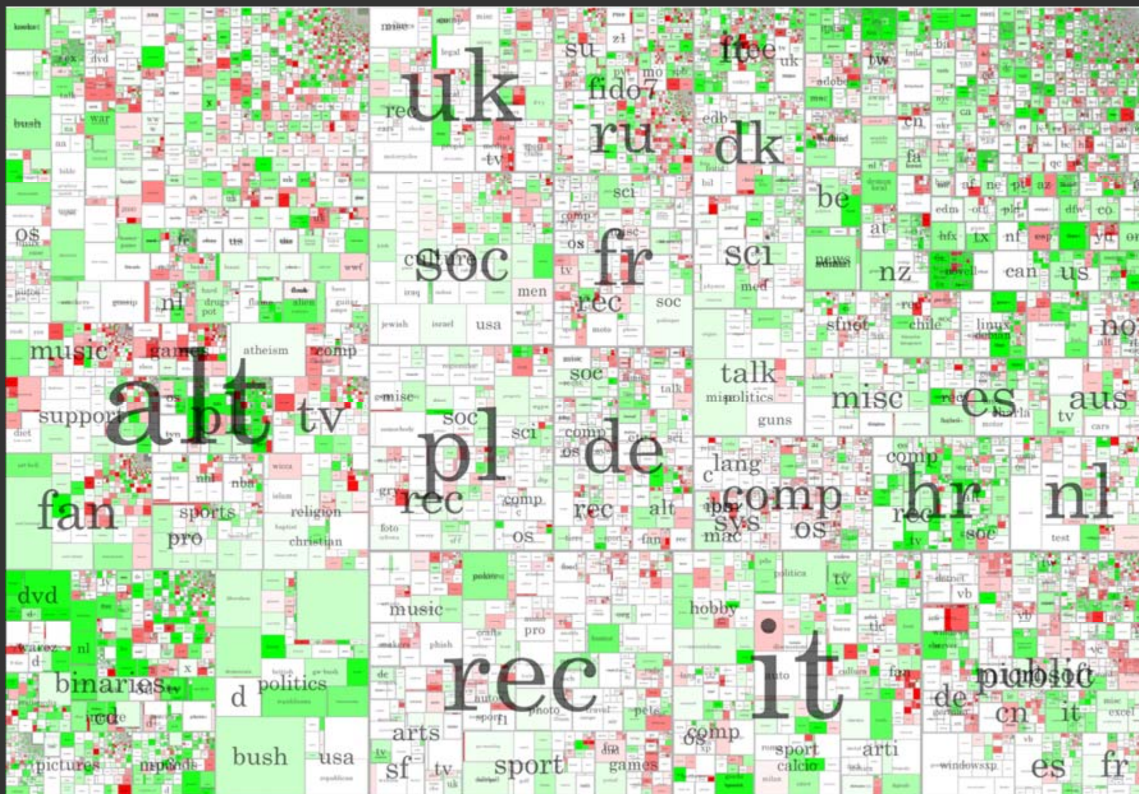
Marc Smith



Cushion treemap

http://treemap.sourceforge.net/

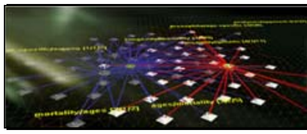




Treemap View of 2004 Usenet Returnees - Marc Smith, Danyel Fisher, Tony Capone - 2005

[#07] Tree Analysis and Visualization

- General Overview
- Designing Effective Tree Visualizations
- Notions and Notations
- Sci2-Reading and Extracting Trees
- Sci2-Visualizing Trees
- Outlook
- Exercise: Identify Promising Tree Analyses of NIH Data



Tree Nodes and Edges

The **root node** of a tree is the node with no parents.

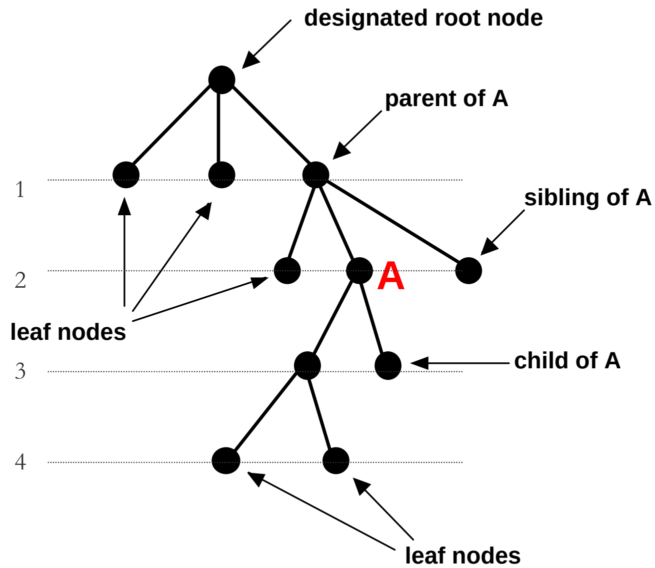
A **leaf node** has no children.

In-degree of a node is the number of edges arriving at that node.

Out-degree of a node is the number of edges leaving that node.

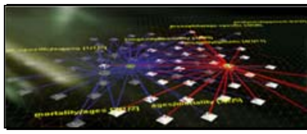
Sample tree of

size 11 (=number of nodes) and **height** 4 (=number of levels).

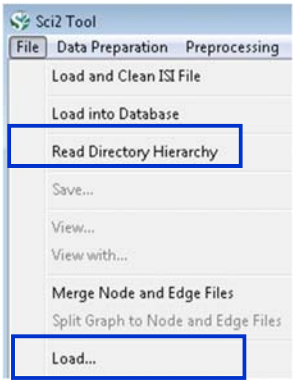


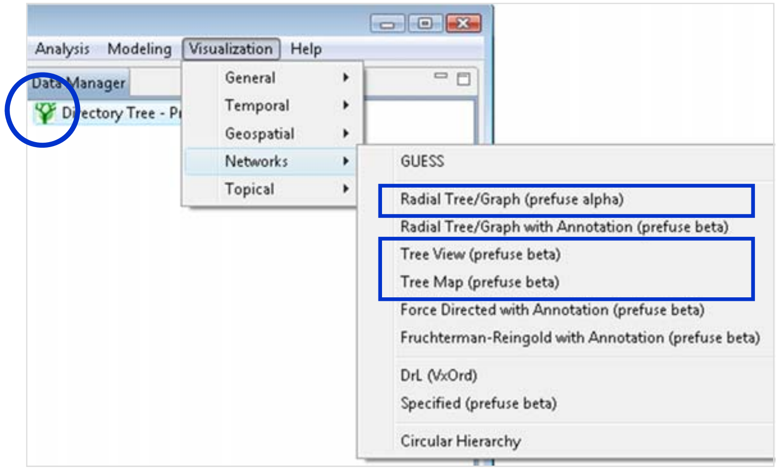
[#07] Tree Analysis and Visualization

- General Overview
- Designing Effective Tree Visualizations
- Notions and Notations
- Sci2-Reading and Extracting Trees
- Sci2-Visualizing Trees
- Outlook
- Exercise: Identify Promising Tree Analyses of NIH Data

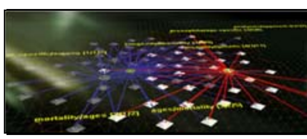


Read and Visualize Trees with Sci2 Tool



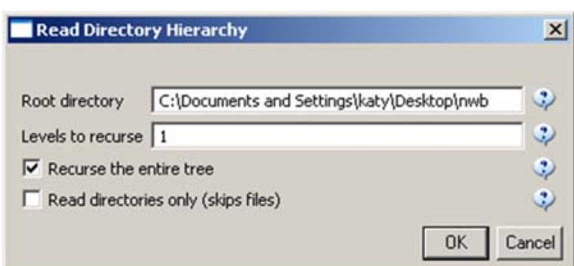


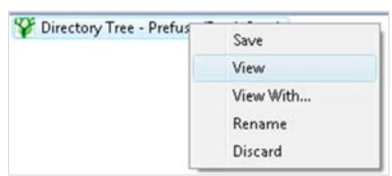
See Science of Science (Sci2) Tool User Manual, Version Alpha 3, Section 3.1 for a listing and brief explanations of all plugins. [*http://sci.slis.indiana.edu/registration/docs/Sci2_Tutorial.pdf*](http://sci.slis.indiana.edu/registration/docs/Sci2_Tutorial.pdf)



Sample Tree: Read Directory Hierarchy

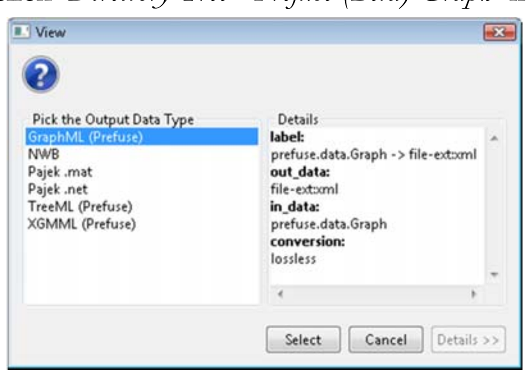
Use *'File > Read Directory Hierarchy'* with parameters





To view file in different formats right click *'Directory Tree - Prefuse (Beta) Graph'* in Data Manager and select *View*.

Select a data format.



Sample Tree: View Directory Hierarchy

File Formats: GraphML (Prefuse)

See documentation at https://nwb.slis.indiana.edu/community/?n=DataFormats.HomePage

```
<?xml version="1.0" encoding="UTF-8" ?>
- <graphml xmlns="http://graphml.graphdrawing.org/xmlns" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://graphml.graphdrawing.org/xmlns http://graphml.graphdrawing.org/xmlns/1.0/graphml.xsd">
    <!-- prefuse GraphML Writer | Sat Jul 17 11:51:03 EDT 2010 -->
  - <key id="label" for="node" attr.name="label" attr.type="string">
      <default />
    </key>
  - <key id="label" for="edge" attr.name="label" attr.type="string">
      <default />
    </key>
  - <graph edgedefault="undirected">
      <!-- nodes -->
    - <node id="n0">
        <data key="label">sci2-with-scimaps</data>
      </node>
    - <node id="n1">
        <data key="label">.eclipseproduct</data>
      </node>
    - <node id="n2">
        <data key="label">sci2.exe</data>
      </node>
    - <node id="n3">
        <data key="label">sci2.ini</data>
      </node>
    - <node id="n4">
        <data key="label">configuration</data>
      </node>
    - <node id="n5">
        <data key="label">config.ini</data>
      </node>
```


Sample Tree: View Directory Hierarchy

File Formats: NWB

See documentation at https://nwb.slis.indiana.edu/community/?n=DataFormats.HomePage

```
*Nodes
id*int label*string
1      "sci2-with-scimaps"
2      ".eclipseproduct"
3      "sci2.exe"
4      "sci2.ini"
5      "configuration"
6      "config.ini"
7      "default_menu.xml"
...

*UndirectedEdges
source*int target*int label*string
1      2      ""
1      3      ""
1      4      ""
1      5      ""
5      6      ""
5      7      ""
5      8      ""
5      9      ""
5      10     ""
```

Sample Tree: View Directory Hierarchy

File Formats: Pajek .net
See documentation at
https://nwb.slis.indiana.edu/community/?n=DataFormats.HomePage

```
*Vertices 568
1 sci2-with-scimaps
2 .eclipseproduct
3 sci2.exe
4 sci2.ini
5 configuration
6 config.ini
7 default_menu.xml
...

*Edges 567
1 2 1 ""
1 3 1 ""
1 4 1 ""
1 5 1 ""
5 6 1 ""
5 7 1 ""
5 8 1 ""
5 9 1 ""
5 10 1 ""
```

Note similarity to .nwb

```
*Nodes
id*int label*string
1      "sci2-with-scimaps"
2      ".eclipseproduct"
3      "sci2.exe"
4      "sci2.ini"
5      "configuration"
6      "config.ini"
7      "default_menu.xml"
...

*UndirectedEdges
source*int target*int label*string
1      2      ""
1      3      ""
1      4      ""
1      5      ""
5      6      ""
5      7      ""
5      8      ""
5      9      ""
5      10     ""
```


Sample Tree: View Directory Hierarchy

File Formats: Pajek .mat
See documentation at https://nwb.slis.indiana.edu/community/?n=DataFormats.HomePage

```
*Vertices 568
1 "sci2-with-scimaps"
2 ".eclipseproduct"
3 "sci2.exe"


*Matrix
0.0 1.0 1.0 1.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0
1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
```

V. Batagelj, A. Mrvar, and W. de Nooy: Pajek

Network – .NET

Network can be defined in different ways on input file. Look at three of them:

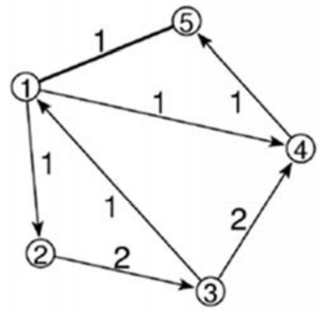
1. List of neighbours (Arcslist / Edgeslist)

```
*Vertices 5
1 "a"
2 "b"
3 "c"
4 "d"
5 "e"
*Arcslist
1 2 4
2 3
3 1 4
4 5
*Edgeslist
1 5
```

V. Batagelj, A. Mrvar, and W. de Nooy: Pajek

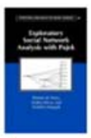
3. Matrix

```
*Vertices 5
1 "a"
2 "b"
3 "c"
4 "d"
5 "e"
*Matrix
0 1 0 1 1
0 0 2 0 0
1 0 0 2 0
0 0 0 0 1
1 0 0 0 0
```

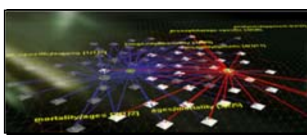


Explanation:

In this format directed lines (arcs) are given in the matrix form (*Matrix).



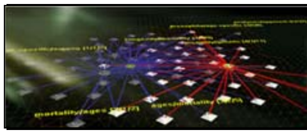
Exploratory Social Network Analysis with Pajek by de Nooy, Wouter
(9)
$35.19



Sample Tree: View Directory Hierarchy

File Formats: TreeML (Prefuse)
See documentation at https://nwb.slis.indiana.edu/community/?n=DataFormats.HomePage

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- prefuse TreeML Writer | Sat Jul 17 12:05:02 EDT 2010 -->
- <tree>
  - <declarations>
      <attributeDecl name="label" type="String" />
    </declarations>
  - <branch>
      <attribute name="label" value="sci2-with-scimaps" />
    - <leaf>
        <attribute name="label" value=".eclipseproduct" />
      </leaf>
    - <leaf>
        <attribute name="label" value="sci2.exe" />
      </leaf>
    - <leaf>
        <attribute name="label" value="sci2.ini" />
      </leaf>
    - <branch>
        <attribute name="label" value="configuration" />
      - <leaf>
          <attribute name="label" value="config.ini" />
        </leaf>
      - <leaf>
          <attribute name="label" value="default_menu.xml" />
        </leaf>
```



Sample Tree: View Directory Hierarchy

File Formats: XGMML (Prefuse)
See documentation at https://nwb.slis.indiana.edu/community/?n=DataFormats.HomePage


```
- <graph directed="0" label="Network" xmlns="http://www.cs.rpi.edu/XGMML">
    <!--   nodes    -->
    <node id="1" label="edu.iu.scipolicy.database.isi.extract.network.cocitation.journal.core_0.0.1.jar" />
    <node id="2" label="org.cishell.templates.jythonrunner_1.0.0" />
    <node id="3" label="feature.xml" />
    <node id="4" label="META-INF" />
    <node id="5" label="isiCoCitation.properties" />
    <node id="6" label="edu.iu.nwb.converter.nwbpajeknet_1.0.0.jar" />
    <node id="7" label="freehep-graphicsio-pdf-2.0.jar" />
    <node id="8" label="Welcome.properties" />
    <node id="9" label="org.cishell.reference.gui.persistence_1.0.0.jar" />

    <!--   edges    -->
    <edge source="2" target="244" label="" />
    <edge source="2" target="337" label="" />
    <edge source="2" target="479" label="" />
    <edge source="4" target="335" label="" />
    <edge source="25" target="360" label="" />
    <edge source="26" target="362" label="" />
    <edge source="34" target="371" label="" />
    <edge source="35" target="177" label="" />
    <edge source="35" target="372" label="" />
    <edge source="36" target="366" label="" />
```


[#07] Tree Analysis and Visualization

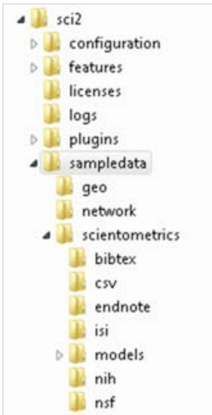
- General Overview
- Designing Effective Tree Visualizations
- Notions and Notations
- Sci2-Reading and Extracting Trees
- Sci2-Visualizing Trees
- Outlook
- Exercise: Identify Promising Tree Analyses of NIH Data

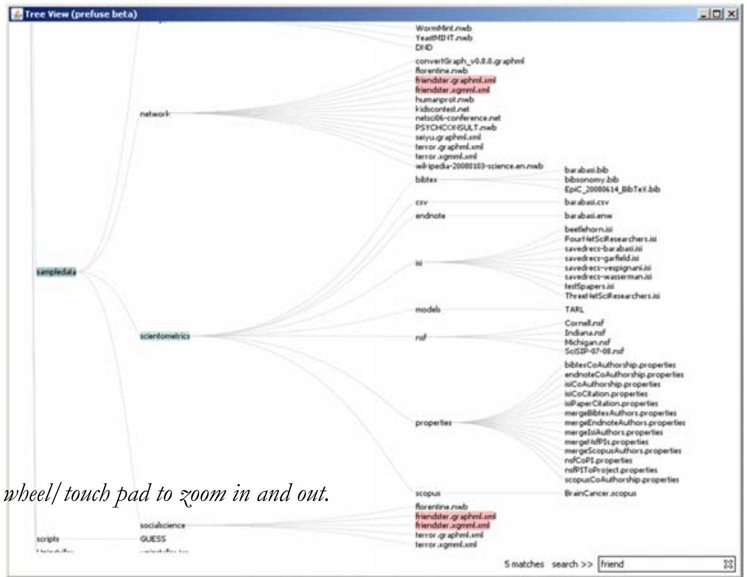
Sample Tree Visualizations

Indented Lists and ***Tree View*** showing nesting of, e.g., directory hierarchies.

Visualize *'Directory Tree - Prefuse (Beta) Graph'* using

- *"Visualization > Networks > Tree View (prefuse beta)'*





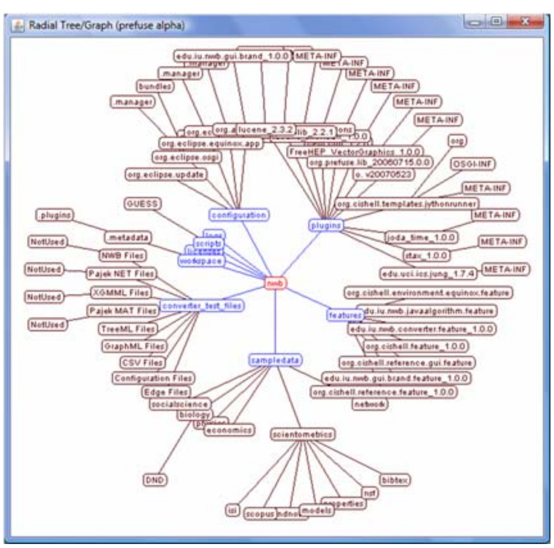
Press right mouse button and use mouse wheel/touch pad to zoom in and out.
Click on directory to expand/collapse.
Use search field to find specific files.

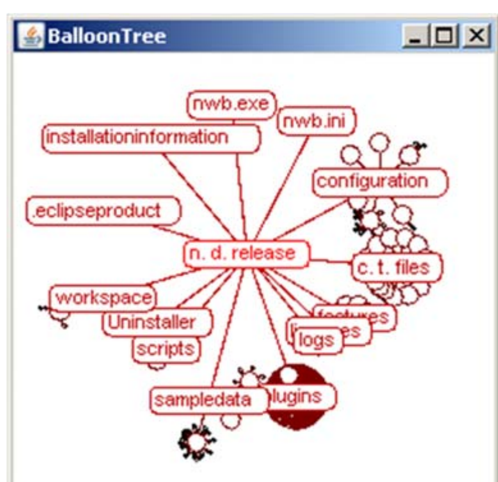
Sample Tree Visualizations

Radial Tree and ***Ballon Tree*** showing the structure of, e.g., directory hierarchies.

Visualize *'Directory Tree - Prefuse (Beta) Graph'* using

- *"Visualization > Networks > Radial Tree/Graph (prefuse alpha)'*
- *"Visualization > Networks > Balloon Graph (prefuse alpha)' (not in Sci2 Tool, Alpha 3)*





Sample Tree Visualization

Tree Map showing the structure of, e.g., directory hierarchies.

Visualize *'Directory Tree - Prefuse (Beta) Graph'* using

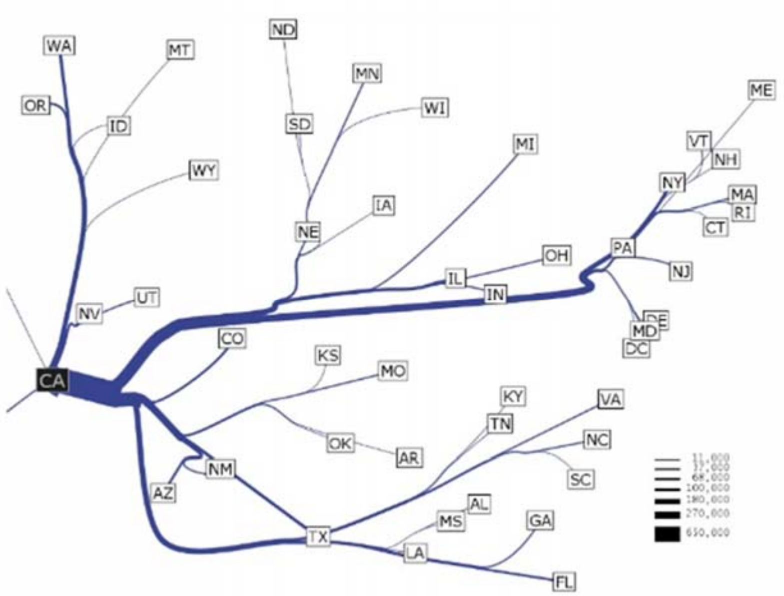
- *'Visualization > Networks > Tree Map (prefuse beta)'*

Sample Tree Visualization

Flow Maps showing migration patterns

http://graphics.stanford.edu/papers/flow_map_layout

Soon available in Sci2 Tool.



[#07] Tree Analysis and Visualization

- General Overview
- Designing Effective Tree Visualizations
- Notions and Notations
- Sci2-Reading and Extracting Trees
- Sci2-Visualizing Trees
- Outlook
- Exercise: Identify Promising Tree Analyses of NIH Data

Outlook

Planned extensions of Sci2 Tool:

- (Flowmap) tree network overlays for geo maps and science maps.
- Bimodal network visualizations.
- Scalable visualizations of large hierarchies.

Research Collaborations by the Chinese Academy of Sciences
By Weixia (Bonnie) Huang, Russell J. Duhon, Elisha F. Hardy, Katy Börner, Indiana University, USA

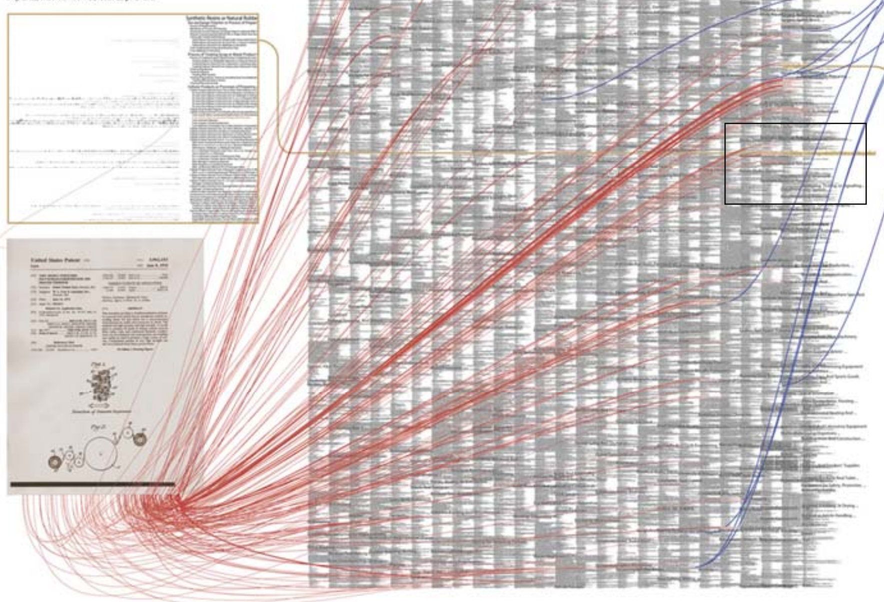
US Patent Hierarchy

Impact

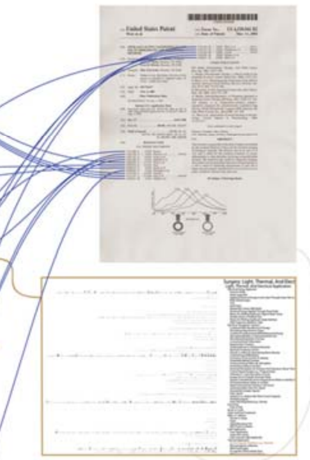
The United States Patent and Trademark Office does scientists and industry a great service by granting patents to protect inventions. Inventions are categorized in a taxonomy that groups patents by industry or use, proximate function, effect or product, and structure. At the time of this writing there are 160,523 categories in a hierarchy that can get as deep as 15 levels. We display the first three levels (13,529 categories) at right in what might be considered a textual map of inventions.

Patent applications are required to be unique and non-obvious, partially by revealing any previous patents that might be similar in nature or provide a foundation for the current invention. In this way we can trace the impact of a single patent, seeing how many patents and categories it affects.

The patent on Goretex—a lightweight, durable synthetic fiber—is an example of one that has had significant impact. The box below enlarges the section of the hierarchy where it is filed, and the red lines (arranged to start along a time line from 1981 to 2006) point to the 130 categories that contain 182 patents, from waterproof clothing to surgical cosmetic implants, that mention Goretex as prior art.



Prior Art

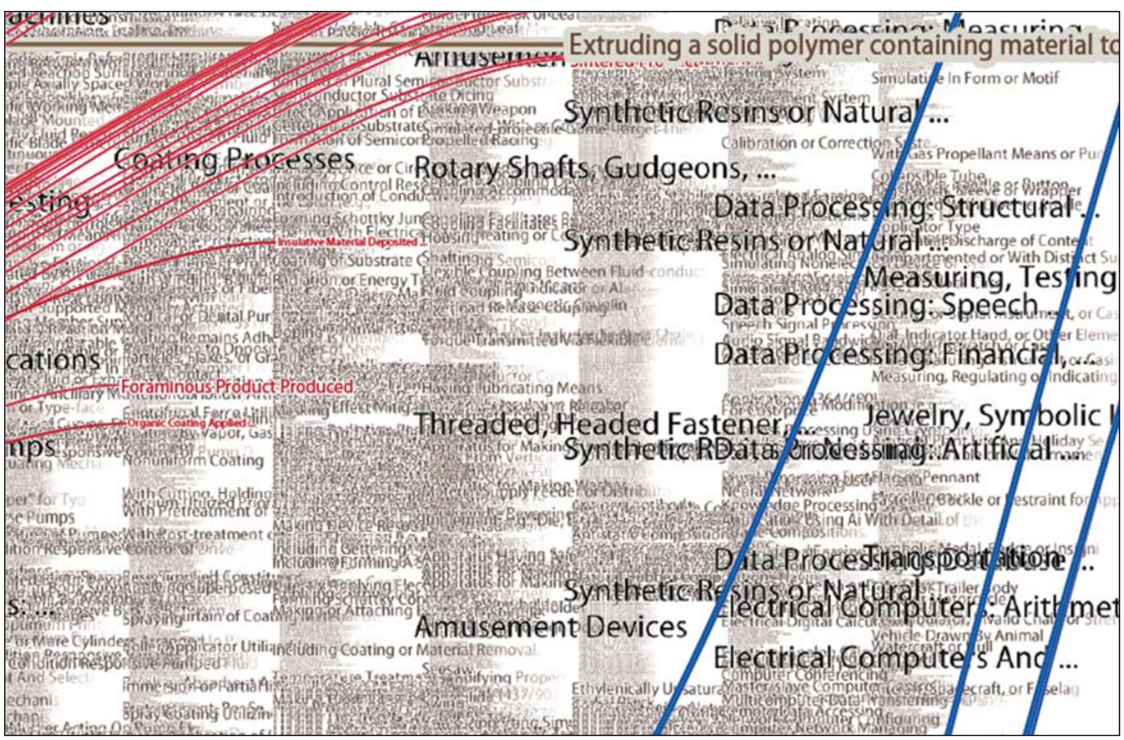


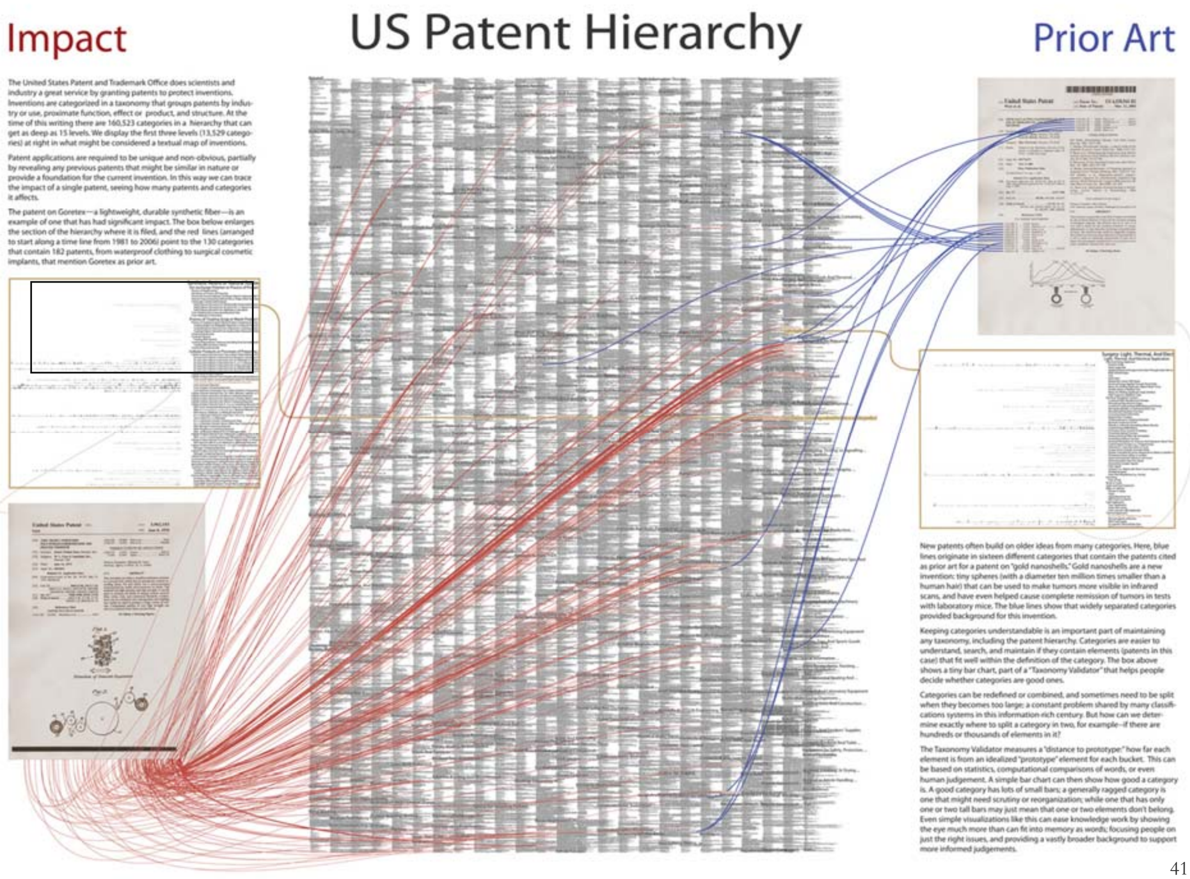
New patents often build on older ideas from many categories. Here, blue lines originate in sixteen different categories that contain the patents cited as prior art for a patent on "gold nanoshells." Gold nanoshells are a new invention: tiny spheres (with a diameter ten million times smaller than a human hair) that can be used to make tumors more visible in infrared scans, and have even helped cause complete remission of tumors in tests with laboratory mice. The blue lines show that widely separated categories provided background for this invention.

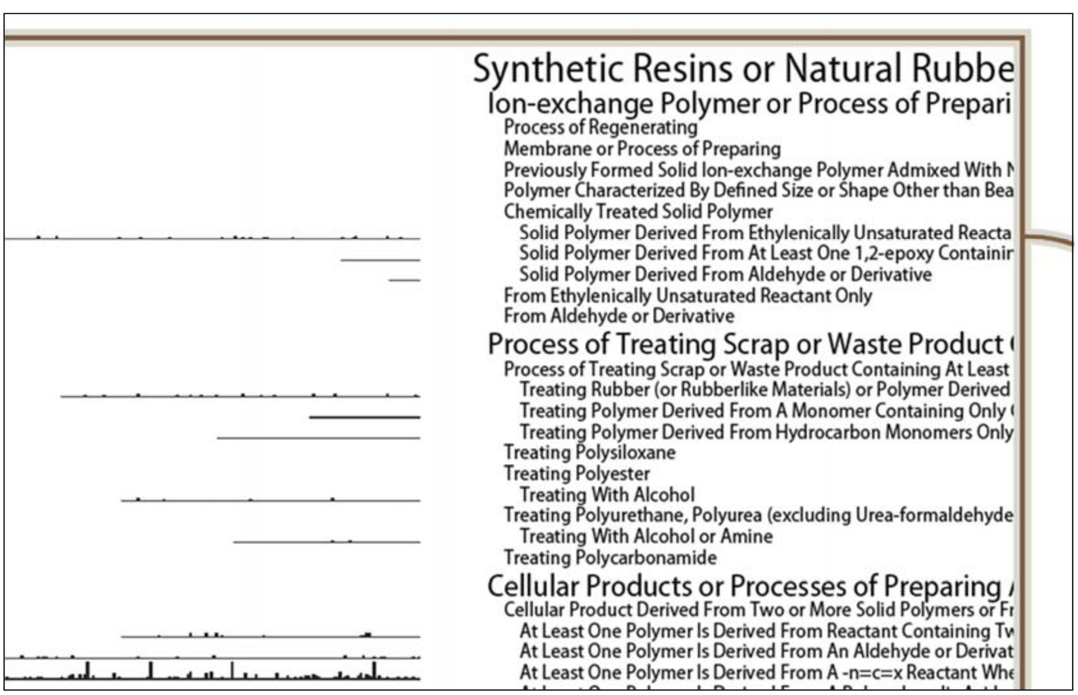
Keeping categories understandable is an important part of maintaining any taxonomy, including the patent hierarchy. Categories are easier to understand, search, and maintain if they contain elements (patents in this case) that fit well within the definition of the category. The box above shows a tiny bar chart, part of a "Taxonomy Validator" that helps people decide whether categories are good ones.

Categories can be redefined or combined, and sometimes need to be split when they becomes too large; a constant problem shared by many classifications systems in this information-rich century. But how can we determine exactly where to split a category in two, for example–if there are hundreds or thousands of elements in it?

The Taxonomy Validator measures a "distance to prototype:" how far each element is from an idealized "prototype" element for each bucket. This can be based on statistics, computational comparisons of words, or even human judgement. A simple bar chart can then show how good a category is. A good category has lots of small bars; a generally ragged category is one that might need scrutiny or reorganization; while one that has only one or two tall bars may just mean that one or two elements don't belong. Even simple visualizations like this can ease knowledge work by showing the eye much more than can fit into memory as words; focusing people on just the right issues, and providing a vastly broader background to support more informed judgements.

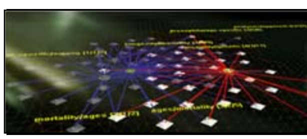



Impact
US Patent Hierarchy
Prior Art
The United States Patent and Trademark Office does scientists and industry a great service by granting patents to protect inventions. Inventions are categorized in a taxonomy that groups patents by industry or use, proximate function, effect or product, and structure. At the time of this writing there are 160,523 categories in a hierarchy that can get as deep as 15 levels. We display the first three levels (13,529 categories) at right in what might be considered a textual map of inventions.
Patent applications are required to be unique and non-obvious, partially by revealing any previous patents that might be similar in nature or provide a foundation for the current invention. In this way we can trace the impact of a single patent, seeing how many patents and categories it affects.
The patent on Goretex—a lightweight, durable synthetic fiber—is an example of one that has had significant impact. The box below enlarges the section of the hierarchy where it is filed, and the red lines (arranged to start along a time line from 1981 to 2006) point to the 130 categories that contain 182 patents, from waterproof clothing to surgical cosmetic implants, that mention Goretex as prior art.
New patents often build on older ideas from many categories. Here, blue lines originate in sixteen different categories that contain the patents cited as prior art for a patent on "gold nanoshells." Gold nanoshells are a new invention: tiny spheres (with a diameter ten million times smaller than a human hair) that can be used to make tumors more visible in infrared scans, and have even helped cause complete remission of tumors in tests with laboratory mice. The blue lines show that widely separated categories provided background for this invention.
Keeping categories understandable is an important part of maintaining any taxonomy, including the patent hierarchy. Categories are easier to understand, search, and maintain if they contain elements (patents in this case) that fit well within the definition of the category. The box above shows a tiny bar chart, part of a "Taxonomy Validator" that helps people decide whether categories are good ones.
Categories can be redefined or combined, and sometimes need to be split when they becomes too large; a constant problem shared by many classifications systems in this information-rich century. But how can we determine exactly where to split a category in two, for example--if there are hundreds or thousands of elements in it?
The Taxonomy Validator measures a "distance to prototype:" how far each element is from an idealized "prototype" element for each bucket. This can be based on statistics, computational comparisons of words, or even human judgement. A simple bar chart can then show how good a category is. A good category has lots of small bars; a generally ragged category is one that might need scrutiny or reorganization; while one that has only one or two tall bars may just mean that one or two elements don't belong. Even simple visualizations like this can ease knowledge work by showing the eye much more than can fit into memory as words; focusing people on just the right issues, and providing a vastly broader background to support more informed judgements.


Synthetic Resins or Natural Rubbe
Ion-exchange Polymer or Process of Prepari
Process of Regenerating
Membrane or Process of Preparing
Previously Formed Solid Ion-exchange Polymer Admixed With N
Polymer Characterized By Defined Size or Shape Other than Bea
Chemically Treated Solid Polymer
Solid Polymer Derived From Ethylenically Unsaturated Reacta
Solid Polymer Derived From At Least One 1,2-epoxy Containir
Solid Polymer Derived From Aldehyde or Derivative
From Ethylenically Unsaturated Reactant Only
From Aldehyde or Derivative
Process of Treating Scrap or Waste Product
Process of Treating Scrap or Waste Product Containing At Least
Treating Rubber (or Rubberlike Materials) or Polymer Derived
Treating Polymer Derived From A Monomer Containing Only
Treating Polymer Derived From Hydrocarbon Monomers Only
Treating Polysiloxane
Treating Polyester
Treating With Alcohol
Treating Polyurethane, Polyurea (excluding Urea-formaldehyde
Treating With Alcohol or Amine
Treating Polycarbonamide
Cellular Products or Processes of Preparing
Cellular Product Derived From Two or More Solid Polymers or Fr
At Least One Polymer Is Derived From Reactant Containing Tw
At Least One Polymer Is Derived From An Aldehyde or Derivat
At Least One Polymer Is Derived From A -n=c=x Reactant Whe

[#08] Network Analysis and Visualization

- General Overview
- Designing Effective Network Visualizations
- Notions and Notations
- Sci2-Reading and Extracting Networks
- Sci2-Analysing Networks
- Sci2-Visualizing Networks
- Outlook



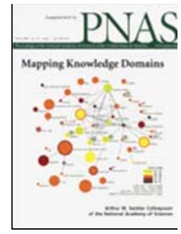
Sample Networks

- Communication networks
  - Internet, telephone network, wireless network.
- Network applications
  - The World Wide Web, Email interactions
- Transportation network/ Road maps
- Relationships between objects in a data base
  - Function/module dependency graphs
  - Knowledge bases

Network Properties

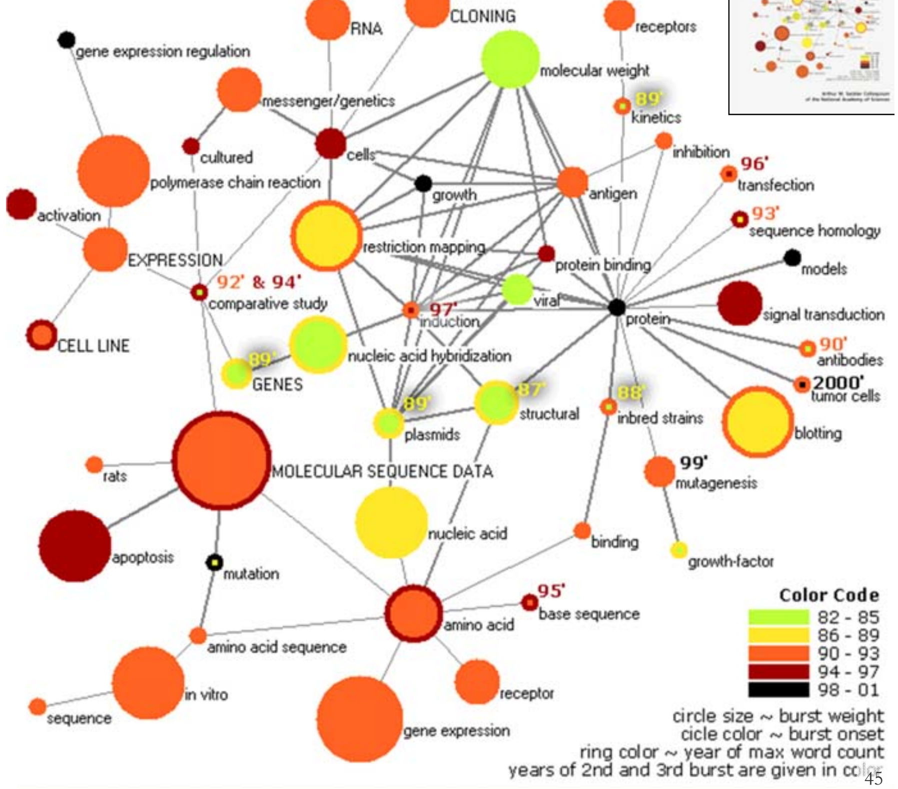
- Directed vs. undirected
- Weighted vs. unweighted
- Additional node and edge attributes
- One vs. multiple node & edge types
- Network type (random, small world, scale free, hierarchical networks)

Reducing the number of edges via pathfinder network scaling.



Co-word space of the top 50 highly frequent and bursty words used in the top 10% most highly cited PNAS publications in 1982-2001.

(Mane & Börner, 2004)



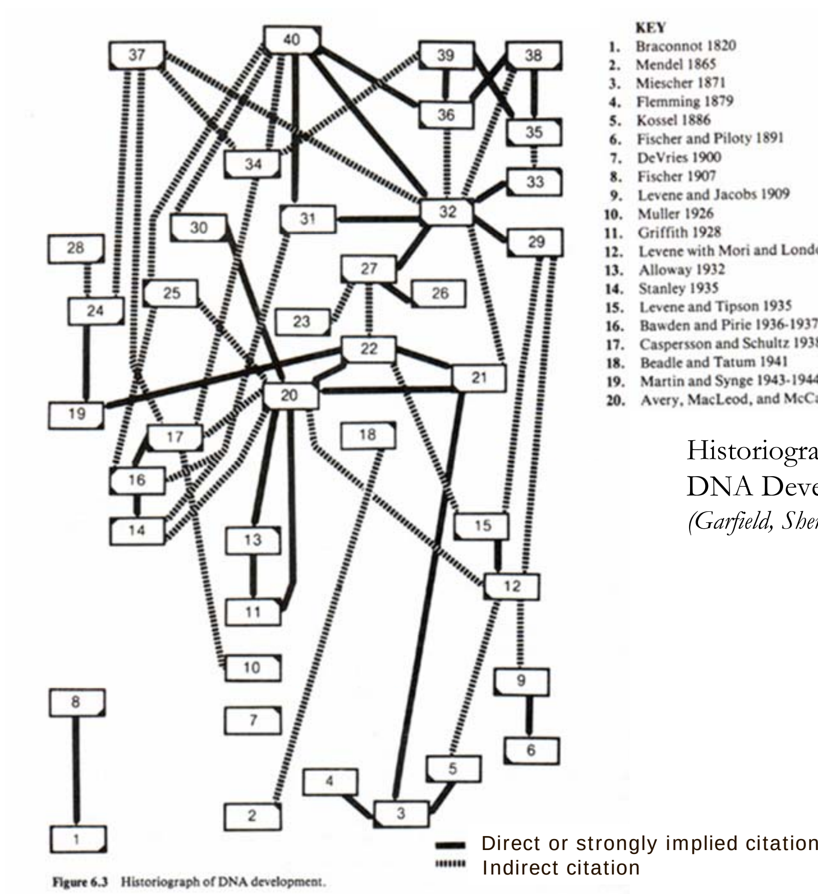
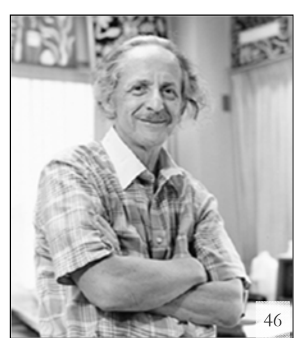


Figure 6.3 Historiograph of DNA development.

KEY
1. Braconnot 1820
2. Mendel 1865
3. Miescher 1871
4. Flemming 1879
5. Kossel 1886
6. Fischer and Piloty 1891
7. DeVries 1900
8. Fischer 1907
9. Levene and Jacobs 1909
10. Muller 1926
11. Griffith 1928
12. Levene with Mori and London 1929
13. Alloway 1932
14. Stanley 1935
15. Levene and Tipson 1935
16. Bawden and Pirie 1936-1937
17. Caspersson and Schultz 1938-1939
18. Beadle and Tatum 1941
19. Martin and Synge 1943-1944
20. Avery, MacLeod, and McCarty 1944
21. Chargaff 1947
22. Chargaff 1950
23. Pauling and Corey 1950-1951
24. Sanger 1951-1953
25. Hershey and Chase 1952
26. Wilkins 1953
27. Watson and Crick 1953
28. DuVigneaud 1953
29. Todd 1955
30. Palade 1954-1956
31. Fraenkel-Conrat 1955-1957
32. Ochoa 1955-1956
33. Kornberg 1956-1957
34. Hoagland 1957-1958
35. Jacob and Monod 1960-1961
36. Hurwitz 1960
37. Dintzis 1961
38. Novelli 1961-1962
39. Allfrey and Mirsky 1962
40. Nirenberg and Matthaei 1961-1962

Historiograph of DNA Development
(Garfield, Sher, & Torpie, 1964)



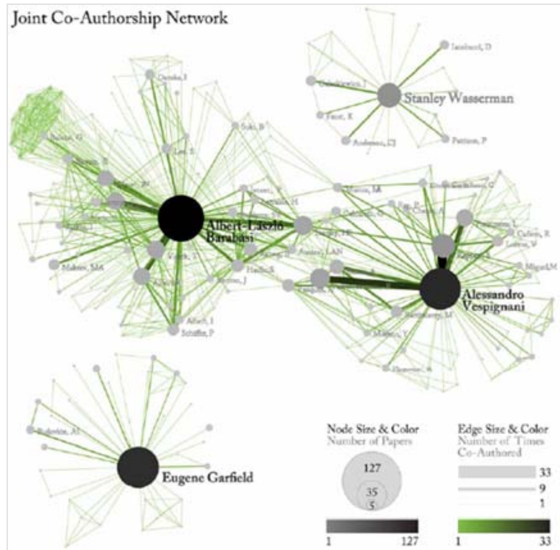
Force Directed Layout – How does it work?

The algorithm simulates a system of forces defined on an input graph and outputs a locally minimum energy configuration. Nodes resemble mass points repelling each other and the edges simulate springs with attracting forces. The algorithm tries to minimize the energy of this physical system of mass particles.

Required are

- A force model
- Technique for finding locally minimum energy configurations.

P. Eades,"A heuristic for graph drawing" Congressus Numerantium, 42,149-160,1984.

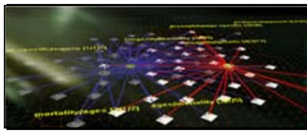


Force Directed Layout cont.

Force Models

Force Model	Formula	Example of usage
Spring Force	***F = k(1-a)*** *k- stiffness of spring* *a- natural length of spring*	Assigning different k and a to different edges to separate nodes by different distances.
Gravity Force	**F = g/r²** $F = g/r^2$ *g- associated with mass of node, usually equals 1.*	Apply gravity force between node pairs to prevent node overlapping.
Electrical and Magnetic Force	**F = eE** **F = qB** *E- electric field strength* *B- magnetic field strength*	Changes nodes distribution along a direction.

A simple algorithm to find the equilibrium configuration is to trace the move of each node according to Newton's 2nd law. This takes time O n^3, which makes it unsuitable for large data sets. Rob Forbes (1987) proposed two methods that were able to accelerate convergence of a FDP problem 3-4 times. One stabilizes the derivative of the repulsion force and the other uses information on node movement and instability characteristics to make a predictive extrapolation.



Force Directed Layout cont.

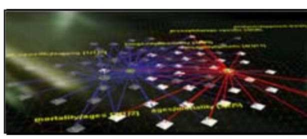
Most existing algorithms extend Eades' algorithm (1984) by providing methods for the intelligent initial placement of nodes, clustering the data to perform an initial coarse layout followed by successively more detailed placement, and grid-based systems for dividing up the dataset.

GEM (Graph EMbedder) attempts to recognize and forestall non-productive rotation and oscillation in the motion of nodes in the graph as it cools, see

Frick, A., A. Ludwig and H. Mehldau (1994). A fast adaptive layout algorithm for undirected graphs. Graph Drawing, Springer-Verlag: 388-403.

Walshaw's (2000) multilevel algorithm provides a "divide and conquer" method for laying out very large graphs by using clustering, see

Walshaw, C. (2000). A multilevel algorithm for force-directed graph drawing. 8th International Symposium Graph Drawing, Springer-Verlag: 171-182.



Force Directed Layout cont.

VxOrd (Davidson, Wylie et al. 2001) uses a density grid in place of pair-wise repulsive forces to speed up execution and achieves computation times order O(N) rather than O(N2). It also employs barrier jumping to avoid trapping of clusters in local minima.

Davidson, G. S., B. N. Wylie and K. W. Boyack (2001). "Cluster stability and the use of noise in interpretation of clustering." Proc. IEEE Information Visualization 2001: 23-30.

An extremely fast layout algorithm for visualizing large-scale networks in three-dimensional space was proposed by (Han and Ju 2003).

*Han, K. and B.-H. Ju (2003). "A fast layout algorithm for protein interaction networks." Bioinformatics **19**(15): 1882-1888.*

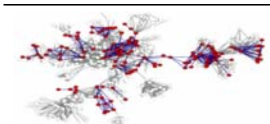
Today, the algorithm developed by Kamada and Kawai (Kamada and Kawai 1989) and Fruchterman and Reingold (Fruchterman and Reingold 1991) are most commonly used, partially because they are available in Pajek.

*Fruchterman, T. M. J. and E. M. Reingold (1991). "Graph Drawing by Force-Directed Placement." Software-Practice & Experience **21**(11): 1129-1164.*

*Kamada, T. and S. Kawai (1989). "An algorithm for drawing general undirected graphs." Information Processing Letters **31**(1): 7-15.*

[#08] Network Analysis and Visualization

- General Overview
- Designing Effective Network Visualizations
- Notions and Notations
- Sci2-Reading and Extracting Networks
- Sci2-Analysing Networks
- Sci2-Visualizing Networks
- Outlook
- Exercise: Identify Promising Network Analyses of NIH Data



Notions and Notations

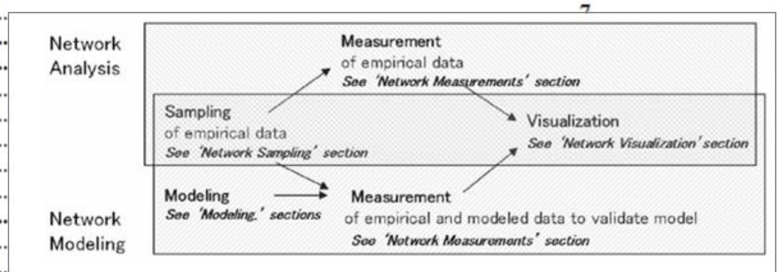


Figure 1. General network science research process.

Börner, Katy, Sanyal, Soma and Vespignani, Alessandro (2007). Network Science. In Blaise Cronin (Ed.), ARIST, Information Today, Inc./American Society for Information Science and Technology, Medford, NJ, Volume 41, Chapter 12, pp. 537-607. http://ivl.slis.indiana.edu/km/pub/2007-borner-arist.pdf

Notions and Notations

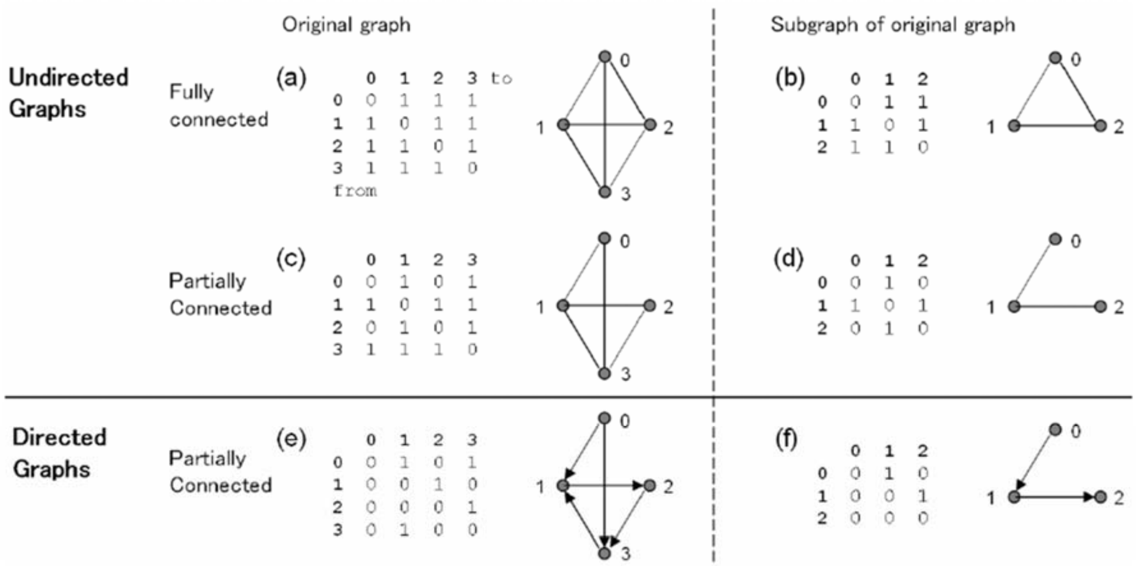


Figure 2: Adjacency matrix and graph presentations of different undirected and directed graphs.

Börner, Katy, Sanyal, Soma and Vespignani, Alessandro (2007). Network Science. In Blaise Cronin (Ed.), ARIST, Information Today, Inc./American Society for Information Science and Technology, Medford, NJ, Volume 41, Chapter 12, pp. 537-607. http://ivl.slis.indiana.edu/km/pub/2007-borner-arist.pdf

Notions and Notations

2.2.1 Node Degree

In undirected graphs, the degree k of a node is termed the number of edges connected to it. In directed graphs, the degree of a node is defined by the sum of its in-degree and its out-degree, $k_i = k_{in,i} + k_{out,i}$, where the *in-degree* $k_{in,i}$ of the node i is defined as the number of edges pointing to i; its *out-degree* $k_{out,i}$ is defined as the number of edges departing from i. In terms of the adjacency matrix, we can write

$$k_{in,i} = \sum_j A_{ji} \ , \ \ k_{out,i} = \sum_j A_{ij} \ . \tag{1}$$

For an undirected graph, with a symmetric adjacency matrix, $k_{in,i} = k_{out,i} \equiv k_i$ holds. For example, node 1 in Figure 2a has a degree of three. Node 1 in Figure 2e has an in-degree of two and an out-degree of one.

2.2.2 Nearest Neighbors

The nearest neighbors of a node i are the nodes to which it is connected directly by an edge, so the number of nearest neighbors of the node is equal to the node degree. For example, node 1 in Figure 2a has nodes 0, 2, and 3 as nearest neighbors.

2.2.3 Path

A *path* P_{i_0,i_n} that connects the nodes i_0 and i_n in a graph $G = (V, E)$ is defined as an ordered collection of n+1 nodes $V_P = \{i_0, i_1, \ldots, i_n\}$ and n edges $E_P = \{(i_0,i_1),(i_1,i_2),\ldots,(i_{n-1},i_n)\}$, such that $i_\alpha \in V$ and $(i_{\alpha-1}, i_\alpha) \in E$, for all α. The *length* of the path P_{i_0,i_n} is n. For example, the path in Figure 2f that interconnects nodes 0, 1, and 2 has a length of two.

Börner, Katy, Sanyal, Soma and Vespignani, Alessandro (2007). Network Science. In Blaise Cronin (Ed.), ARIST, Information Today, Inc./American Society for Information Science and Technology, Medford, NJ, Volume 41, Chapter 12, pp. 537-607. http://ivl.slis.indiana.edu/km/pub/2007-borner-arist.pdf

Notions and Notations

Betweenness centrality is a measure that aims to describe a node's position in a network in terms of the flow it is able to control. As an example, consider two highly connected subgraphs that share one node but no other nodes or edges. Here, the shared node controls the flow of information, for example, rumors in a social network. Any path from any node in one subgraph to any node in the other subgraph leads through the shared node. The shared node has a rather high betweenness centrality. Mathematically, the betweenness centrality is defined as the number of shortest paths between pairs of nodes that pass through a given node (Freeman, 1977). More precisely, let $L_{h,j}$ be the total number of shortest paths from h to j and $L_{h,i,j}$ be the number of those shortest paths that pass through the node i. The betweenness b of node i is then defined as $b_i = \sum L_{h,i,j} / L_{h,j}$, where the sum runs over all h,j pairs with $j \neq h$. An efficient algorithm to compute betweenness centrality was reported by Brandes (2001). The betweenness centrality is often used in transportation networks to provide an estimate of the traffic handled by different nodes, assuming that the frequency of use can be approximated by the number of shortest paths passing through a given node. It is important to stress that while the betweenness centrality is a local attribute of any given node, it is calculated by looking at all paths among all nodes in the network and therefore it is a measure of the node centrality with respect to the global topology of the network.

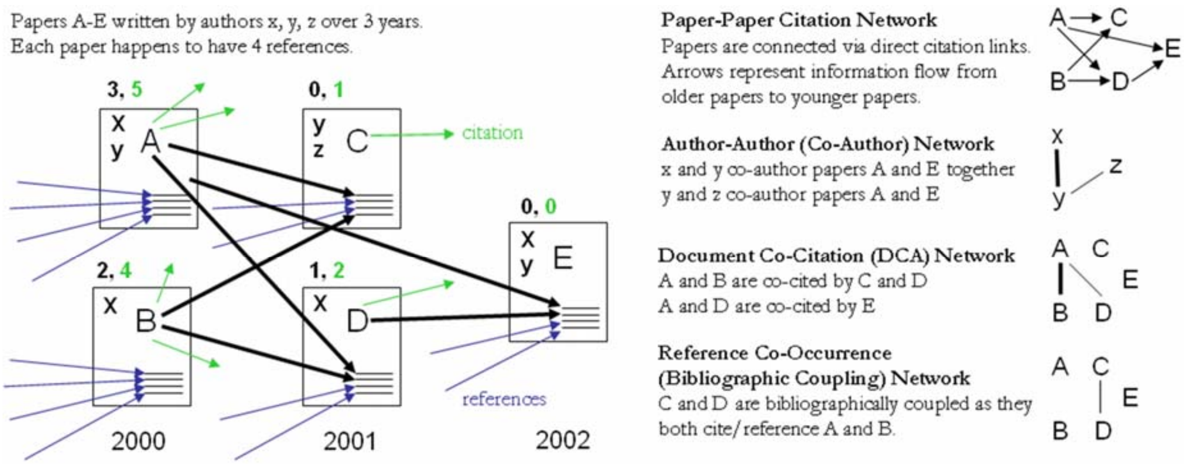
Börner, Katy, Sanyal, Soma and Vespignani, Alessandro (2007). Network Science. In Blaise Cronin (Ed.), ARIST, Information Today, Inc./American Society for Information Science and Technology, Medford, NJ, Volume 41, Chapter 12, pp. 537-607. http://ivl.slis.indiana.edu/km/pub/2007-borner-arist.pdf

[#08] Network Analysis and Visualization

- General Overview
- Designing Effective Network Visualizations
- Notions and Notations
- Sci2-Reading and Extracting Networks
- Sci2-Analysing Networks
- Sci2-Visualizing Networks
- Outlook
- Exercise: Identify Promising Network Analyses of NIH Data

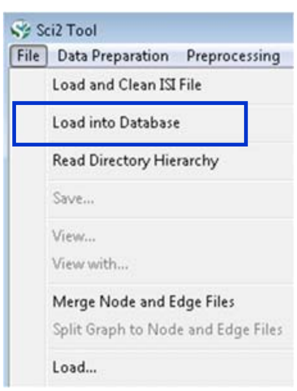
Network Extraction - Examples

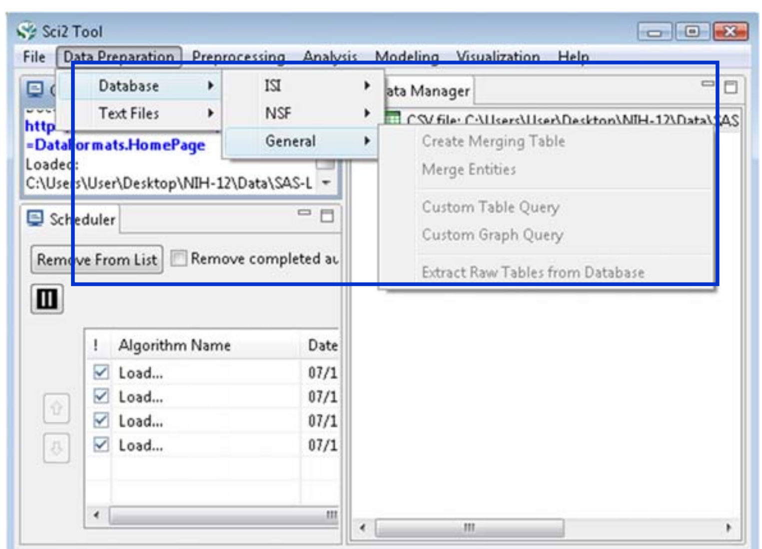
Sample paper network (left) and four different network types derived from it (right).
From ISI files, about 30 different networks can be extracted.



Local citation counts (within this dataset) are given in **black** and global citation counts (ISI times cited) are given in green above each paper.

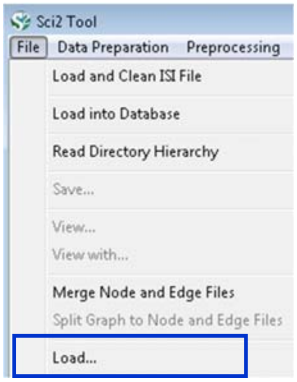
Extract Networks with Sci2 Tool – Database

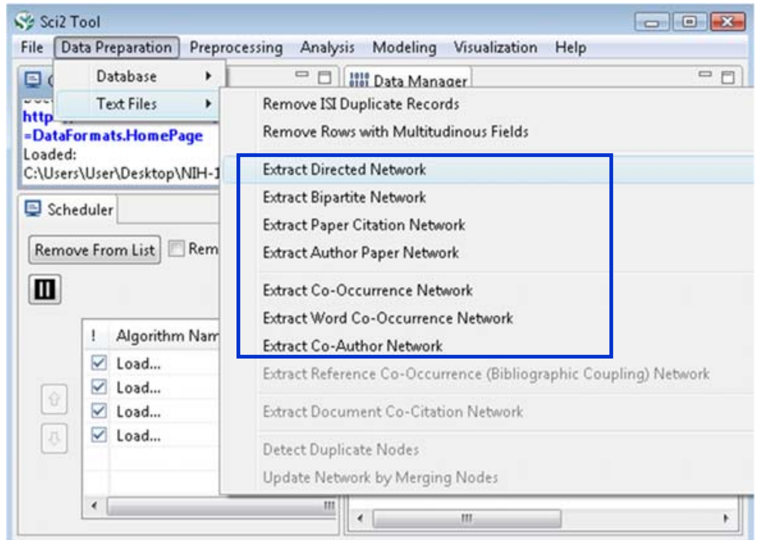




See Science of Science (Sci2) Tool User Manual, Version Alpha 3, Section 3.1 for a listing and brief explanations of all plugins. http://sci.slis.indiana.edu/registration/docs/Sci2_Tutorial.pdf
See also **Tutorial #3**

Extract Networks with Sci2 Tool – Text Files





See Science of Science (Sci2) Tool User Manual, Version Alpha 3, Section 3.1 for a listing and brief explanations of all plugins. [*http://sci.slis.indiana.edu/registration/docs/Sci2_Tutorial.pdf*](http://sci.slis.indiana.edu/registration/docs/Sci2_Tutorial.pdf)
See also **Tutorial #3**

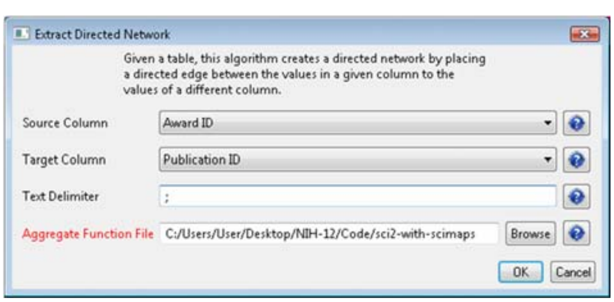
Fake NIH Dataset of Awards and Resulting Publications

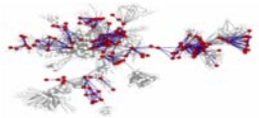
Ten existing awards and a fake set of resulting publications.

Award ID	Publication ID
C06CA058690	9485464;9096302
C06CA059267	20527532;8858722;20427856;20185186;20019401;10587228
C06RR011192	16913728;16362150;19490921
C06RR012176	9714740;19490921
C06RR012488	15345738;11994348;12586855;12865481
C06RR012511	19896513;19487298;19214230
C06RR012512	18991629;17125941;18636192;16621538;18595716;17504144;17350279;17134906;19155177
C06RR012537	18207467;17318410;17961182;19490921
C06RR013551	16136041
C06RR014469	17621683

Load resulting using 'File > Load > Fake-NIH-Awards+Publications.csv' as csv file format.

Extract author bipartite grant to publications network using 'Data Preparation > Text Files > Extract Directed Network' using parameters:





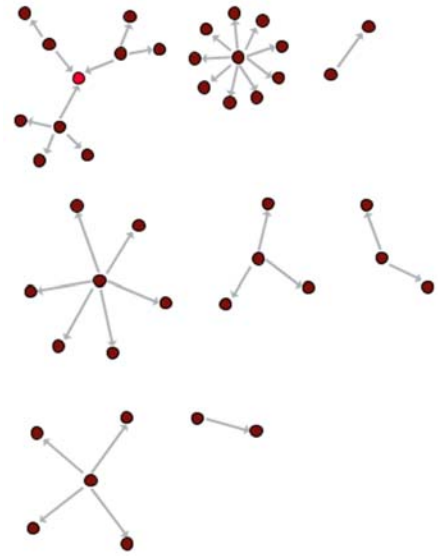
Fake NIH Dataset cont.

Network Analysis Toolkit (NAT)

This graph claims to be directed.
Nodes: 43
Isolated nodes: 0
Edges: 35
No self loops were discovered.
No parallel edges were discovered.
Did not detect any edge attributes
This network does not seem to be a valued network.

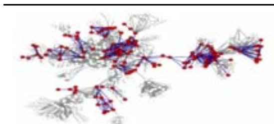
Average total degree: 1.6279
Average in degree: 0.814
Average out degree: 0.814
This graph is not weakly connected.
There are **8 weakly connected components**. (0 isolates)
The largest connected component consists of 10 nodes.

Density (disregarding weights): 0.0194



GUESS

GEM Layout, Bin pack



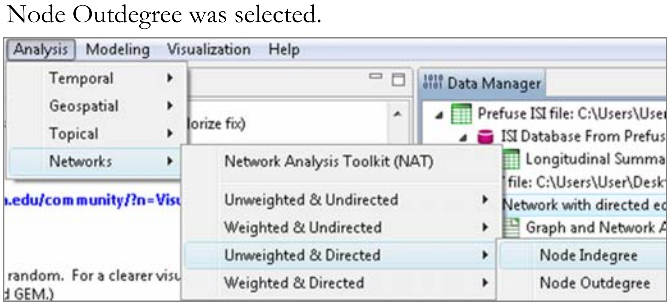
Fake NIH Dataset cont.

In Sci2

Node Indegree was selected.

....

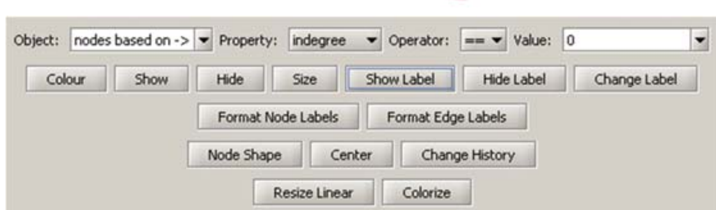
Node Outdegree was selected.

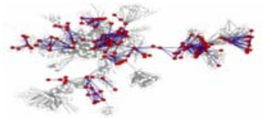


GUESS

GEM Layout, Bin pack

Color using Graph Modifier





Fake NIH Dataset cont.

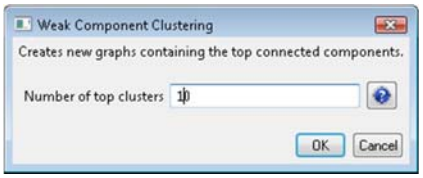
In Sci2

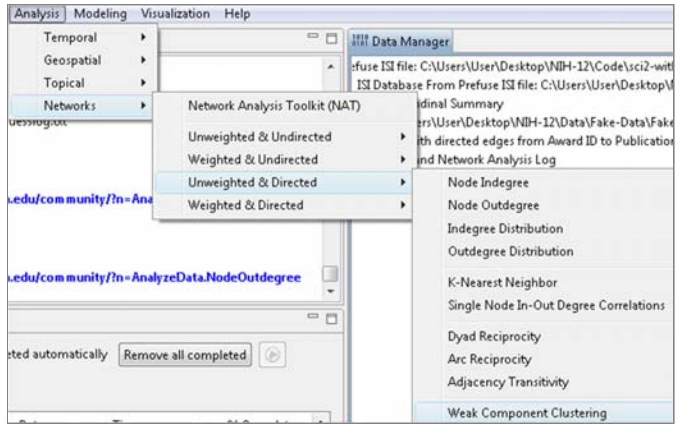
Weak Component Clustering.

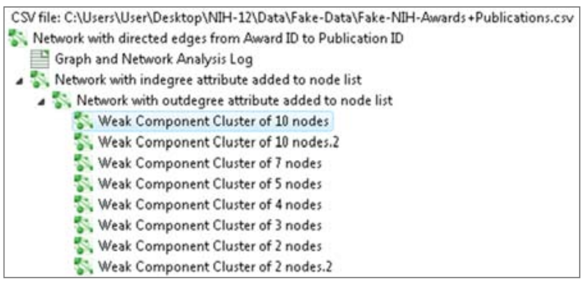
Input Parameters:

Number of top clusters: 10

8 clusters found, generating graphs for the top 8 clusters.



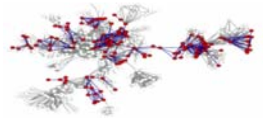




Visualize giant component in GUESS

[#08] Network Analysis and Visualization

- General Overview
- Designing Effective Network Visualizations
- Notions and Notations
- Sci2-Reading and Extracting Networks
- Sci2-Analyzing Networks
- Sci2-Visualizing Networks
- Outlook
- Exercise: Identify Promising Network Analyses of NIH Data



Couple Network Analysis and Visualization to Generate Readable Layouts of Large Graphs

Discover Landmark Nodes based on

- Connectivity (degree or BC values)
- Frequency of access

(Source: Mukherjea & Hara, 1997; Hearst p. 38 formulas)

Identify Major (and Weak) Links

Identify the Backbone

Show Clusters

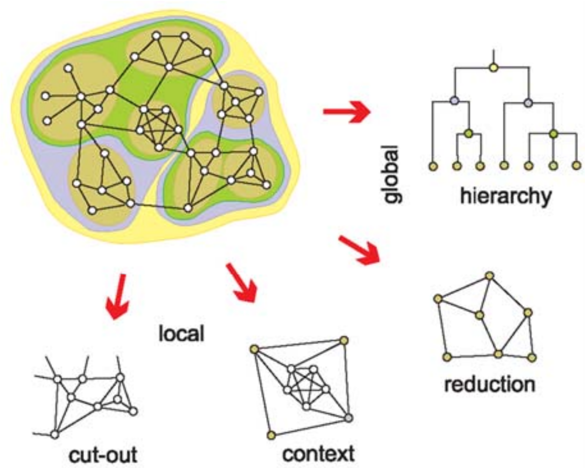


Figure 2: *Approaches to deal with large networks*

Pajek Tutorial

See also Ketan Mane's Qualifying Paper

http://ella.slis.indiana.edu/~kmane/phdprogress/quals/kmane_quals.pdf

http://ella.slis.indiana.edu/~katy/teaching/ketan-quals-slides.ppt

[#08] Network Analysis and Visualization

- General Overview
- Designing Effective Network Visualizations
- Notions and Notations
- Sci2-Reading and Extracting Networks
- Sci2-Analysing Networks
- Sci2-Visualizing Networks
- Outlook
- Exercise: Identify Promising Network Analyses of NIH Data

Network Visualization

General Visualization Objectives

- Representing structural information & content information
- Efficient space utilization
- Easy comprehension
- Aesthetics
- Support of interactive exploration

Challenges in Visualizing Large Networks

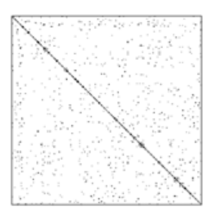
- Positioning nodes without overlap
- De-cluttering links
- Labeling
- Navigation/interaction

General Network Representations

Matrices

1	0	0	6	0
0	10.5	0	0	0
0	0	.015	0	0
0	250.5	0	-280	33.32
0	0	0	0	12

Structure Plots

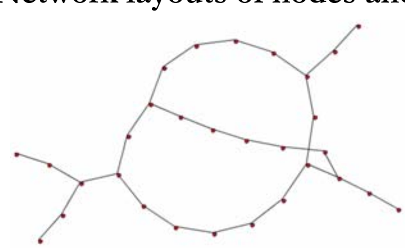


Equivalenced representation of US power network

Lists of nodes & links

```
*Vertices 3
1 "Doc1" 0.0 0.0 0.0 ic Green bc Brown
2 "Doc2" 0.0 0.0 0.0 ic Green bc Brown
3 "Doc3" 0.0 0.0 0.0 ic Green bc Brown
*Arcs
1 2 3 c Green
2 3 5 c Black
*Edges
1 3 4 c Green
```

Network layouts of nodes and links



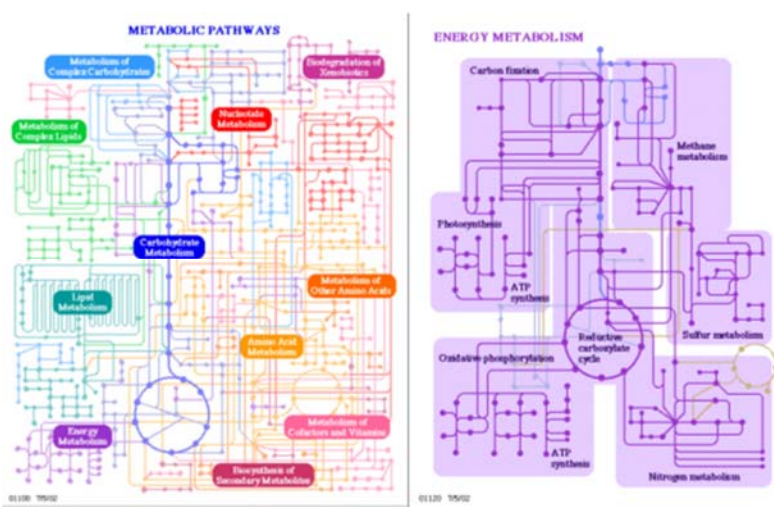
Aesthetic Criteria for Network Visualization

- Symmetric.
- Evenly distributed nodes.
- Uniform edge lengths.
- Minimized edge crossings.
- Orthogonal drawings.
- Minimize area / bends / slopes / angles

Optimization criteria may be relaxed to speed up layout process.

(Source: Fruchterman & R. alg p. 76, see Table & discussion Hearst, p 88)

Aesthetic Network Visualization



http://www.genome.ad.jp/kegg/pathway/map/map01100.html

Small Networks

- Up to 100 nodes
- All nodes and edges and most of their attributes can be shown.

General mappings for

nodes

- # -> (area) size
- Intensity (secondary value) -> color
- Type -> shape 

edges

- # -> thickness
- Intensity, age, etc. -> color
- Type -> style

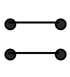
Medium Size Networks

- Up to 10,000 nodes
- Most nodes can be shown but not all their labels.
- Frequently, the number of edges and attributes need to be reduced.

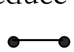
Major design strategies:

Show only important nodes, edges, labels, attributes

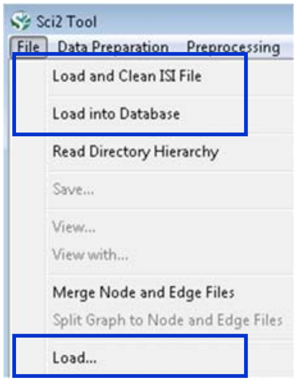
Order nodes spatially

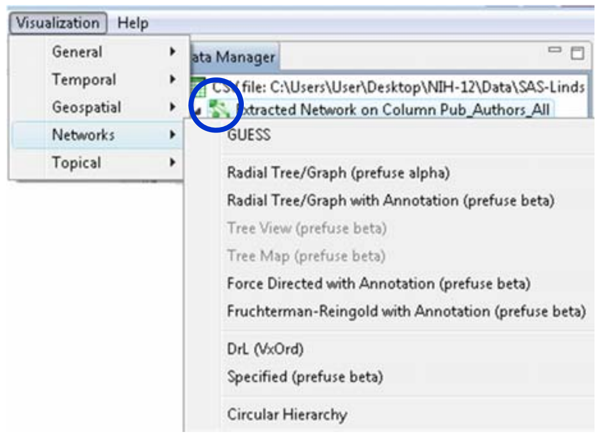


Reduce number of displayed nodes

3 

Visualize Networks with Sci2 Tool



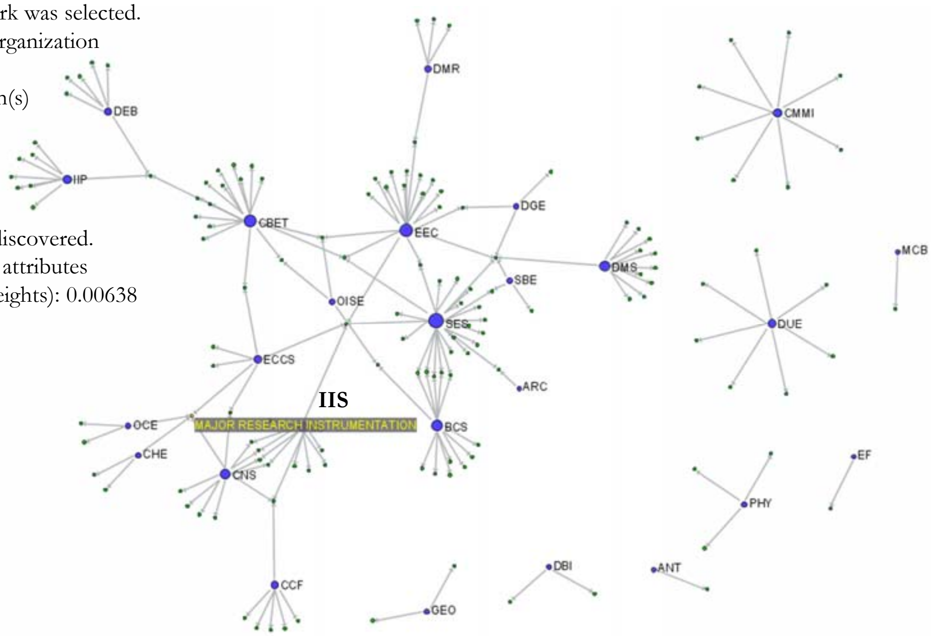


See Science of Science (Sci2) Tool User Manual, Version Alpha 3, Section 3.1 for a listing and brief explanations of all plugins. http://sci.slis.indiana.edu/registration/docs/Sci2_Tutorial.pdf

NSF Medical+Health Funding: Bimodal Network of NSF Organization to Program(s)

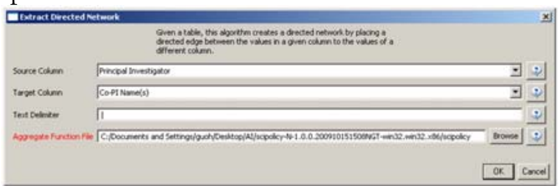
Extract Directed Network was selected.
Source Column: NSF Organization
Text Delimiter: |
Target Column: Program(s)

Nodes: 167
Isolated nodes: 0
Edges: 177
No parallel edges were discovered.
Did not detect any edge attributes
Density (disregarding weights): 0.00638



NSF Medical+Health Funding:
Extract Principal Investigator: Co-PI Networks

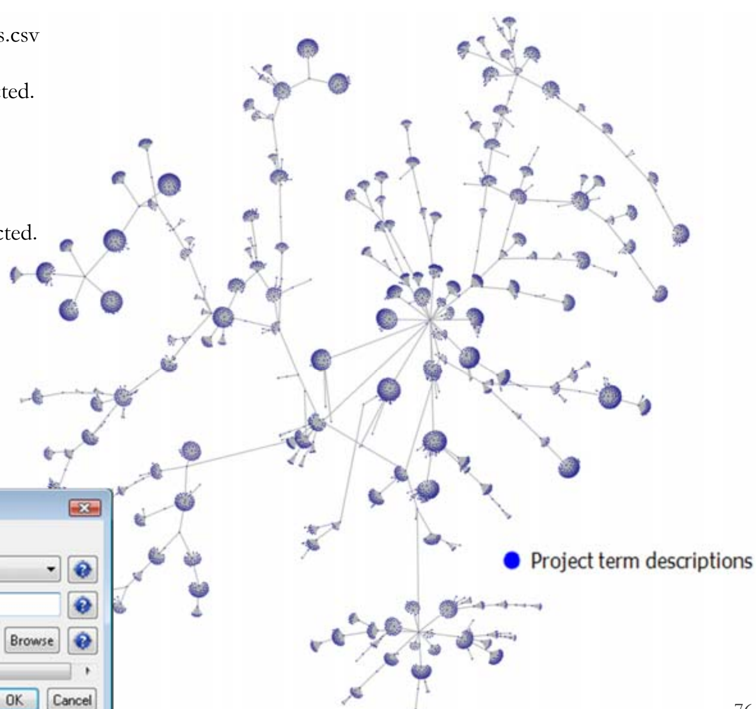
- Load into NWB, open file to count records, compute total award amount.
- Run '*Scientometrics > Extract Directed Network*' using parameters:

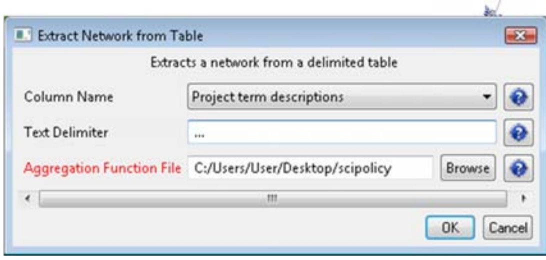


- Select *"Extracted Network .."* and run *'Analysis > Network Analysis Toolkit (NAT)'*
- Remove unconnected nodes via *'Preprocessing > Delete Isolates'*.
- Run '*Analysis > Unweighted & Directed Network > Node Indegree / Node Outdegree*'.
- *'Visualization > GUESS'* , layout with GEM, Bin Pack
- Use Graph Modifier to color/size network.

NIH CTSA Grants:
Co-Project Term Descriptions Occurrence Network

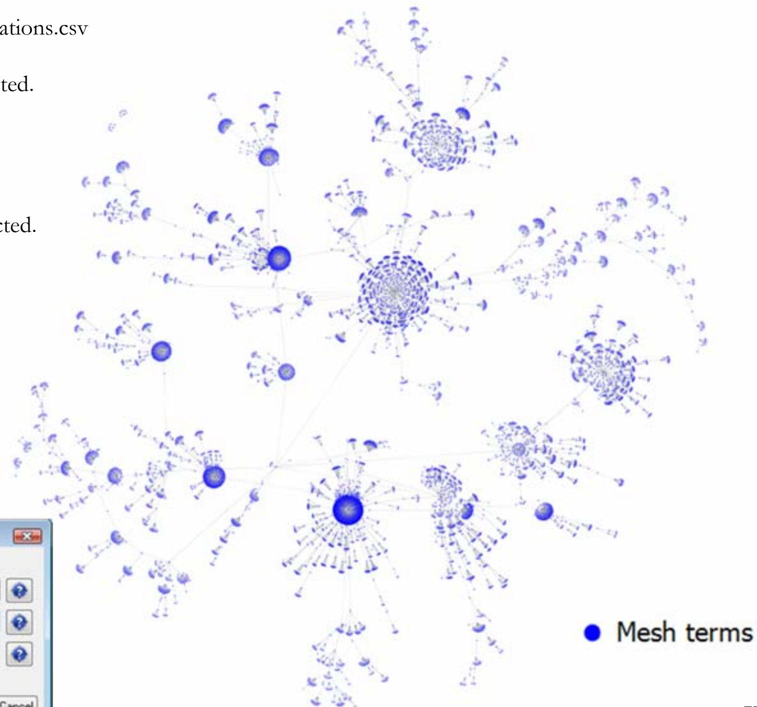
Load... was selected.
Loaded: …\NIH-data\NIH-CTSA-Grants.csv
....
Extract Co-Occurrence Network was selected.
Input Parameters:
Text Delimiter: ...
Column Name: Project term descriptions
....
Network Analysis Toolkit (NAT) was selected.
Nodes: 5723
Isolated nodes: 3
Edges: 353218

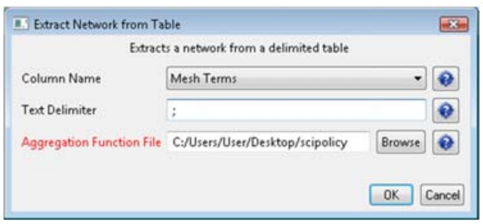




NIH CTSA Publications: Co-Mesh Terms Occurrence Network

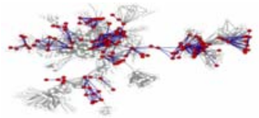
Load... was selected.
Loaded: …\NIH-data\NIH-CTSA-Publications.csv
....
Extract Co-Occurrence Network was selected.
Input Parameters:
Text Delimiter: ;
Column Name: Mesh Terms
....
Network Analysis Toolkit (NAT) was selected.
Nodes: 10218
Edges: 163934





[#09] Large Network Analysis and Visualization

- General Overview
- Designing Effective Network Visualizations
- Sci2-Reading Networks
- Sci2-Analysing Large Networks
- Sci2-Visualizing Large Networks and Distributions
- Outlook



Large Networks

- More than 10,000 nodes.
- Neither all nodes nor all edges can be shown at once. Sometimes, there are more nodes than pixels.

Examples of large networks

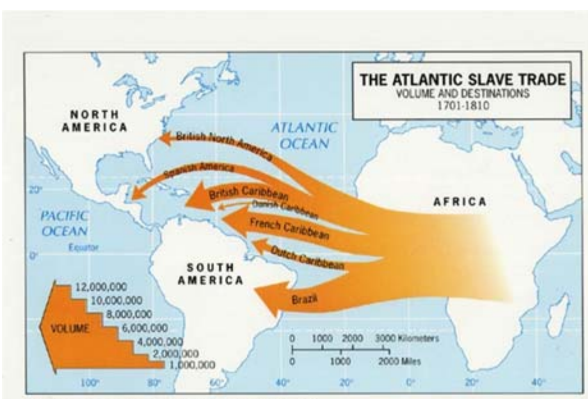
- Communication networks:

  Internet, telephone network, wireless network.
- Network applications:

  The World Wide Web, Email interactions
- Transportation network/road maps
- Relationships between objects in a data base:

  Function/module dependency graphs

  Knowledge bases



http://loadrunner.uits.iu.edu/weathermaps/abilene/

Amsterdam RealTime project, WIRED Magazine, Issue 11.03 - March 2003

Direct Manipulation

Modify focusing parameters while continuously provide visual feedback and update display (fast computer response).

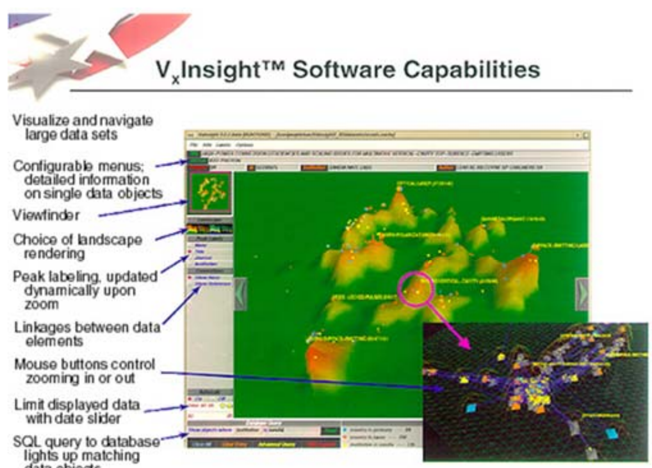
- Conditioning: filter, set background variables and display foreground parameters
- Identification: highlight, color, shape code
- Parameter control: line thickness, length, color legend, time slider, and animation control
- Navigation: Bird's Eye view, zoom, and pan
- Information requests: Mouse over or click on a node to retrieve more details or collapse/expand a subnetwork

See NIH Awards Viewer at http://scimaps.org/maps/nih/2007/

VxInsight Tool

VxInsight is a general purpose knowledge visualization software package developed at Sandia National Laboratories.

It enables researchers, analysts, and decision-makers to accelerate their understanding of large databases.



Davidson, G.S., Hendrickson, B., Johnson, 1 "Knowledge Mining with VxInsight: Discovery through Interaction," Volume 11, Number 3, Journal of Intelligent Information Systems, Special Issue on Integrating Artificial Intelligence and Database Technologies. pp.259-285.)

Other Tools

Tool	Year	Domain	Description	Open Source	Operating System	References
S&T Dynam. Toolbox	1985	Scientom.	Tools from Loet Leydesdorff for organization analysis, and visualization of scholarly data.	No	Windows	(Leydesdorff , 2008)
In Flow	1987	SocSci	Social network analysis software for organizations with support for what-if analysis.	No	Windows	(Krebs, 2008)
Pajek	1996	SocSci	A network analysis and visualization program with many analysis algorithms, particularly for social network analysis.	No	Windows	(Batagelj & Mrvar, 1998)
BibExcel	2000	Scientom	Transforms bibliographic data into forms usable in Excel, Pajek, NetDraw, and other programs.	No	Windows	(Persson, 2008)
Boost Graph Library	2000	CS	Extremely efficient and flexible C++ library for extremely large networks.	Yes	All Major	(Siek et al., 2002)
UCINet	2000	SocSci	Social network analysis software particularly useful for exploratory analysis.	No	Windows	(Borgatti et al., 2002)
Visone	2001	SocSci	Social network analysis tool for research and teaching, with a focus on innovative and advanced visual methods.	No	All Major	(Brandes & Wagner, 2008)
Cytoscape	2002	Bio	Network visualization and analysis tool focusing on biological networks, with particularly nice visualizations.	Yes	All Major	(Cytoscape-Consortium, 2008)

See http://ivl.slis.indiana.edu/km/pub/2010-borner-et-al-nwb.pdf for references.

Other Tools cont.

Tool	Year	Domain	Description	Open Source	Operating System	References
GeoVISTA	2002	Geo	GIS software that can be used to lay out networks on geospatial substrates.	Yes	All Major	(Takatsuka & Gahegan, 2002)
iGraph	2003	CS	A library for classic and cutting edge network analysis usable with many programming languages.	Yes	All Major	(Csárdi & Nepusz, 2006)
Tulip	2003	CS	Graph visualization software for networks over 1,000, 000 elements.	Yes	All Major	(Auber, 2003)
CiteSpace	2004	Scientom	A tool to analyze and visualize scientific literature, particularly co-citation structures.	Yes	All Major	(Chen, 2006)
GraphViz	2004	Networks	Flexible graph visualization software.	Yes	All Major	(AT&T-Research-Group, 2008)
Hittite	2004	Scientom	Analysis and visualization tool for data from the Web of Science.	No	Windows	(Garfield, 2008)
R	2004	Statistics	A statistical computing language with many libraries for sophisticated network analyses.	Yes	All Major	(Ihaka & Gentleman, 1996)
Prefuse	2005	Visualiz.	A general visualization framework with many capabilities to support network visualization and analysis.	Yes	All Major	(Heer et al., 2005)
NWB Tool	2006	Bio, IS, SocSci, Scientom	Network analysis & visualization tool conducive to new algorithms supportive of many data formats.	Yes	All Major	(Huang, 2007)

See http://ivl.slis.indiana.edu/km/pub/2010-borner-et-al-nwb.pdf for references.

[#09] Large Network Analysis and Visualization

- General Overview
- Designing Effective Network Visualizations
- Sci2-Reading Networks
- Sci2-Analyzing Large Networks
- Sci2-Visualizing Large Networks and Distributions
- Outlook
- Exercise: Identify Promising Large Network Analyses of NIH Data

Network Analysis and Visualization – General Workflow

Original Data

	A	B
1	Source Node	Target Nodes
2	A	1;2;3
3	B	3;4
4	C	2;3
5	D	1

	A	B
1	Source Node	Target Nodes
2	A	1
3	A	2
4	A	3
5	B	3
6	B	4
7	C	2
8	C	3
9	D	1

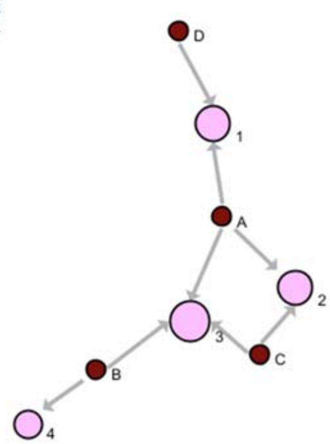
Calculate Node Attributes

```
*Nodes
id*int label*string  bipartitetype*string indegree*int outdegree*int
1     "A" "Source Node" 0 3
2     "3" "Target Nodes" 3 0
3     "2" "Target Nodes" 2 0
4     "1" "Target Nodes" 2 0
5     "B" "Source Node" 0 2
6     "4" "Target Nodes" 1 0
7     "C" "Source Node" 0 2
8     "D" "Source Node" 0 1
*DirectedEdges
source*int target*int
1     2
1     4
1     3
5     6
5     2
7     2
7     3
8     4
```

Extract Network

Extract Bipartite Network was selected.
Input Parameters:
First column: Source Node
Text Delimiter: ;
Second column: Target Nodes

Visualization/Layout



Large Network Analysis & Visualization – General Workflow

Original Data

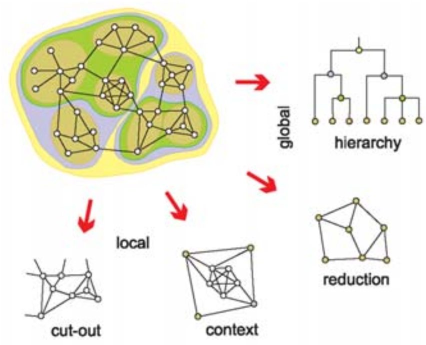
Millions of records, in 100s of columns.
SAS and Excel might not be able to handle these files.
Files are shared between DB and tools as delimited text files (.csv).

Extract Network

It might take several hours to extract a network on a laptop or even on a parallel cluster.

Derived Statistics

Degree distributions
Number of components and their sizes
Extract giant component, subnetworks for further analysis



Visualizations

It is typically not possible to layout the network.
DrL scales to 10 million nodes.

DrL Large Network Layout

See Section 4.9.4.2 in Sci2 Tutorial,
http://sci.slis.indiana.edu/registration/docs/Sci2_Tutorial.pdf

DrL is a force-directed graph layout toolbox for real-world large-scale graphs up to 2 million nodes. It includes:

- Standard force-directed layout of graphs using algorithm based on the popular VxOrd routine (used in the VxInsight program).
- Parallel version of force-directed layout algorithm.
- Recursive multilevel version for obtaining better layouts of very large graphs.
- Ability to add new vertices to a previously drawn graph.

The version of DrL included in Sci2 only does the standard force-directed layout (no recursive or parallel computation).

Davidson, G. S., B. N. Wylie and K. W. Boyack (2001). "Cluster stability and the use of noise in interpretation of clustering." Proc. IEEE Information Visualization 2001: 23-30.

DrL Large Network Layout

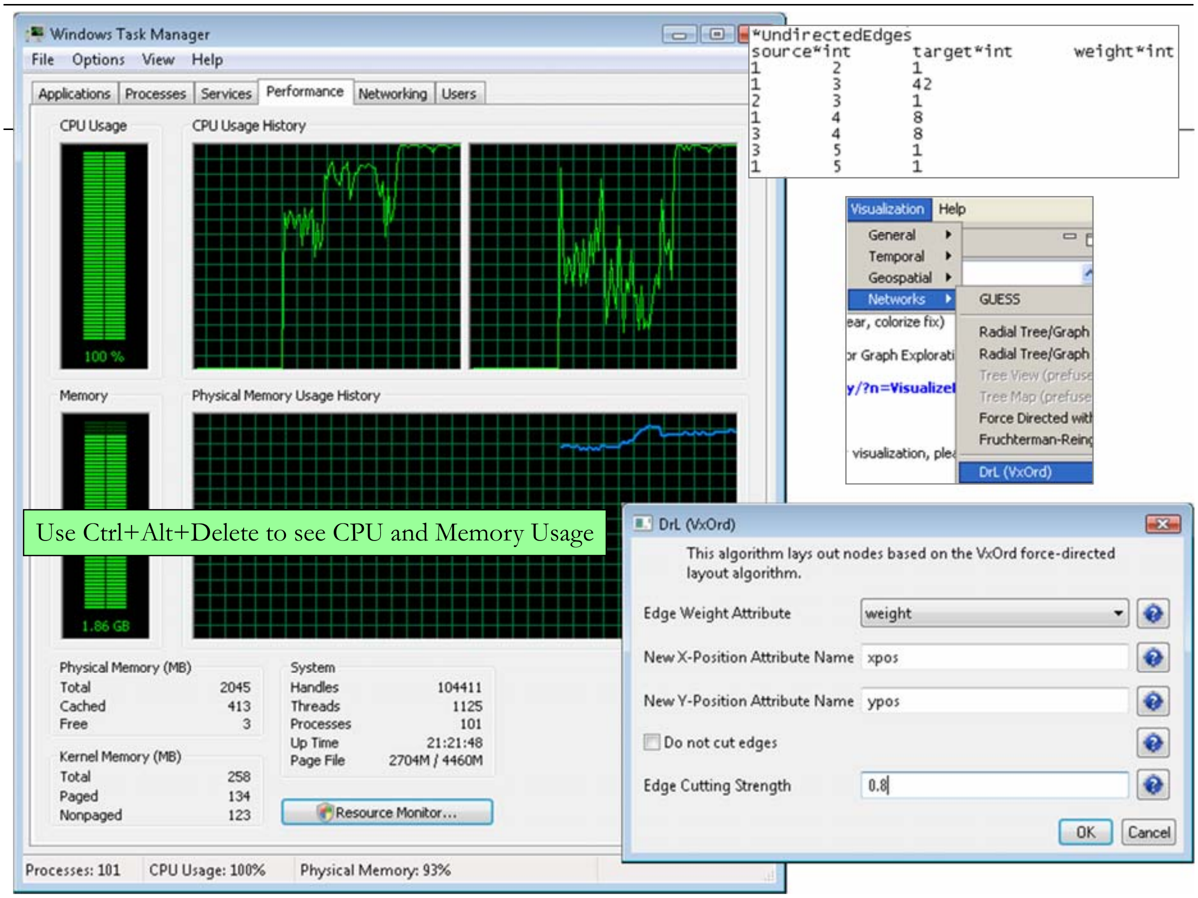
See Section 4.9.4.2 in Sci2 Tutorial,
http://sci.slis.indiana.edu/registration/docs/Sci2_Tutorial.pdf

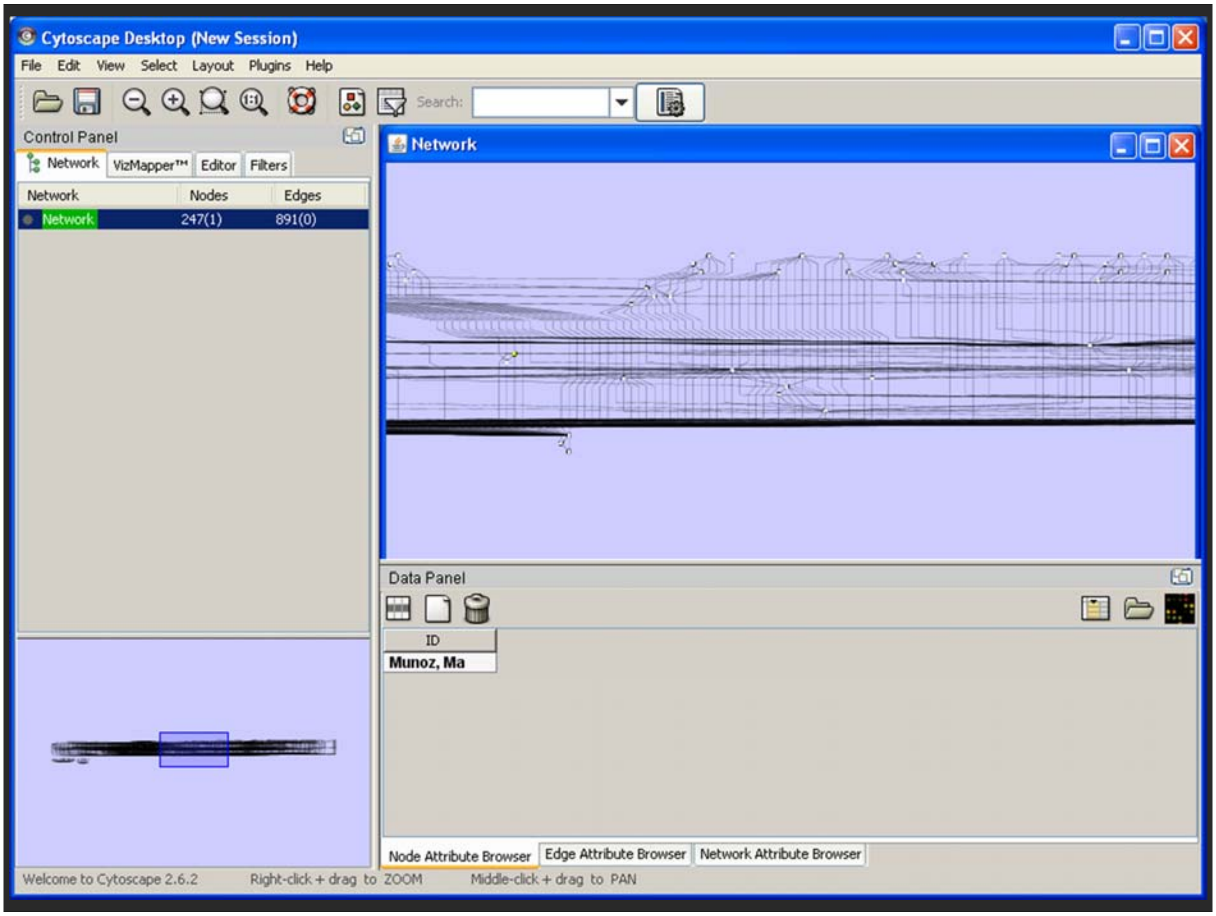
How to use: DrL expects the edges to be *weighted* and *undirected* where the non-zero weight denotes how similar the two nodes are (higher is more similar). Parameters are as follows:

- The **edge cutting parameter** expresses how much automatic edge cutting should be done. 0 means as little as possible, 1 as much as possible. Around .8 is a good value to use.
- The **weight attribute parameter** lets you choose which edge attribute in the network corresponds to the similarity weight. The X and Y parameters let you choose the attribute names to be used in the returned network which corresponds to the X and Y coordinates computed by the layout algorithm for the nodes.

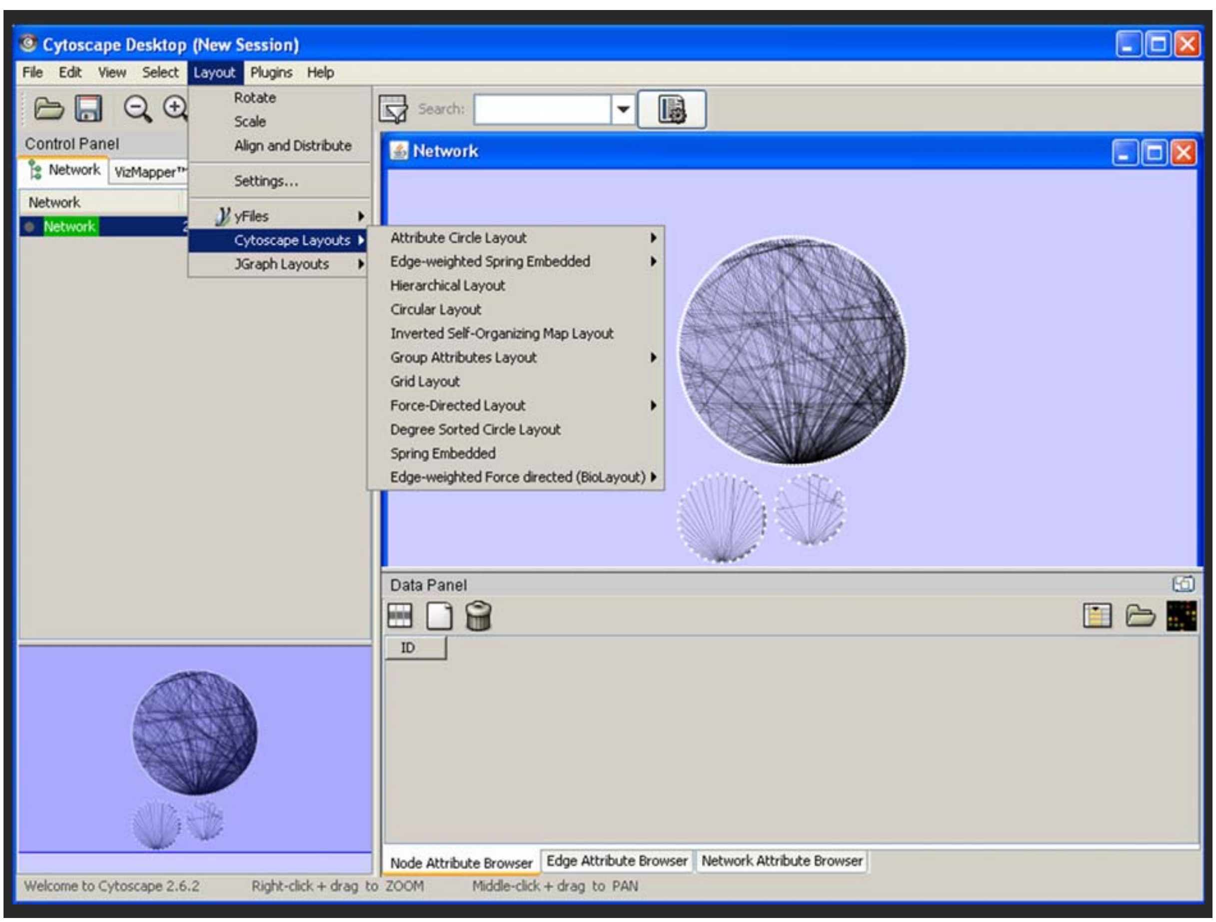
DrL is commonly used to layout large networks, e.g., those derived in co-citation and co-word analyses. In the Sci2 Tool, the results can be viewed in either GUESS or *'Visualization > Specified (prefuse alpha)'*.

See also https://nwb.slis.indiana.edu/community/?n=VisualizeData.DrL



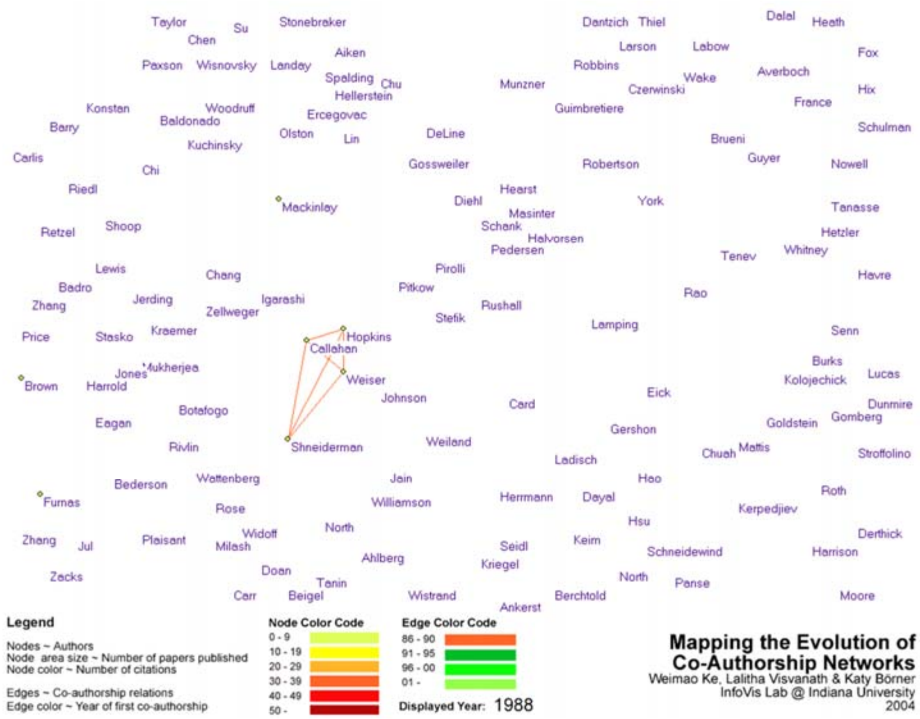


Cytoscape Desktop (New Session)
File Edit View Select Layout Plugins Help
Search:
Control Panel
Network VizMapper™ Editor Filters
Network Nodes Edges
Network 247(1) 891(0)
Network
Data Panel
ID
Munoz, Ma
Node Attribute Browser Edge Attribute Browser Network Attribute Browser
Welcome to Cytoscape 2.6.2 Right-click + drag to ZOOM Middle-click + drag to PAN



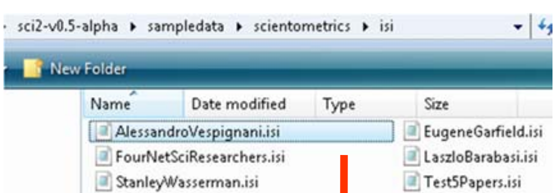
Cytoscape Desktop (New Session)
File Edit View Select Layout Plugins Help
Rotate
Scale
Align and Distribute
Settings...
yFiles
Cytoscape Layouts
JGraph Layouts
Attribute Circle Layout
Edge-weighted Spring Embedded
Hierarchical Layout
Circular Layout
Inverted Self-Organizing Map Layout
Group Attributes Layout
Grid Layout
Force-Directed Layout
Degree Sorted Circle Layout
Spring Embedded
Edge-weighted Force directed (BioLayout)
Search:
Control Panel
Network VizMapper™
Network
Network
Data Panel
ID
Node Attribute Browser Edge Attribute Browser Network Attribute Browser
Welcome to Cytoscape 2.6.2 Right-click + drag to ZOOM Middle-click + drag to PAN

Evolving collaboration networks



Evolving collaboration networks

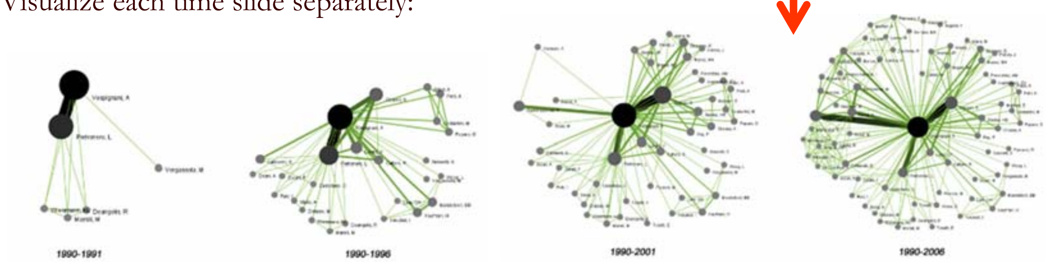
Load isi formatted file



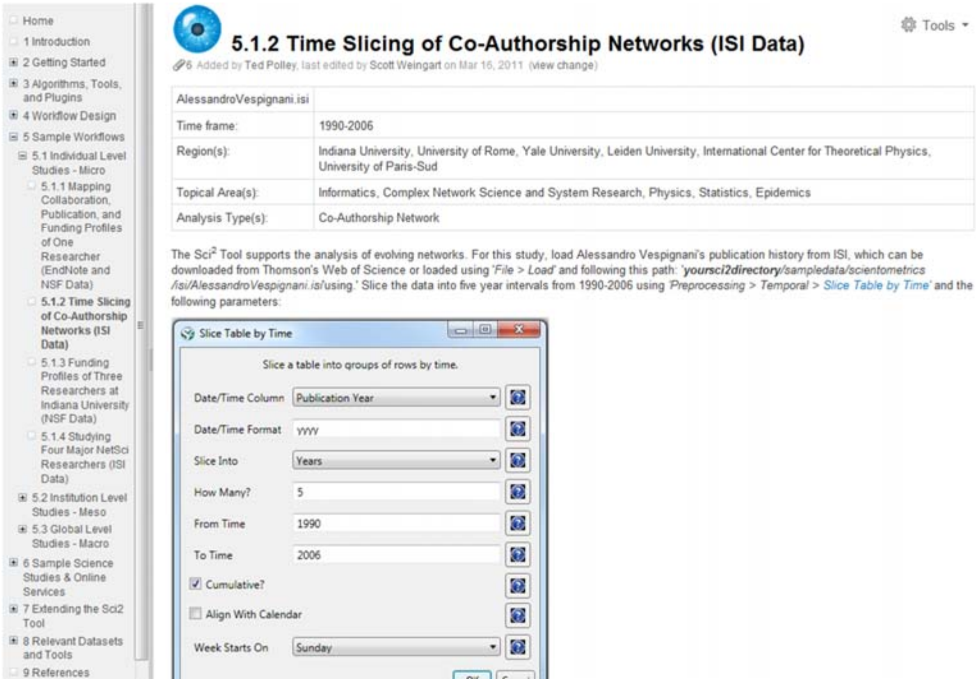
As csv, file looks like:

	A	B	C	D	E	F	G
1	Abstract	Authors	Authors (Full Names)	Beginning	Book Serie	Book Serie	Cited Pate
2	The systematic study of	Colizza, V|Barrat, A|Barthelemy, M|Vespignani, A		2015			
3	Uncovering the hidden r	Colizza, V|Flammini, A|Serrano, MA|Vespignani, A		110			
4	Computer viruses can s	Vespignani, A		135			
5	Mapping the Internet ge	Dall'Asta, L|Alvarez-Hamelin, I|Barrat, A|Vazquez, A|Vespignani, A		140		LECTURE NOTES IN	

Visualize each time slide separately:

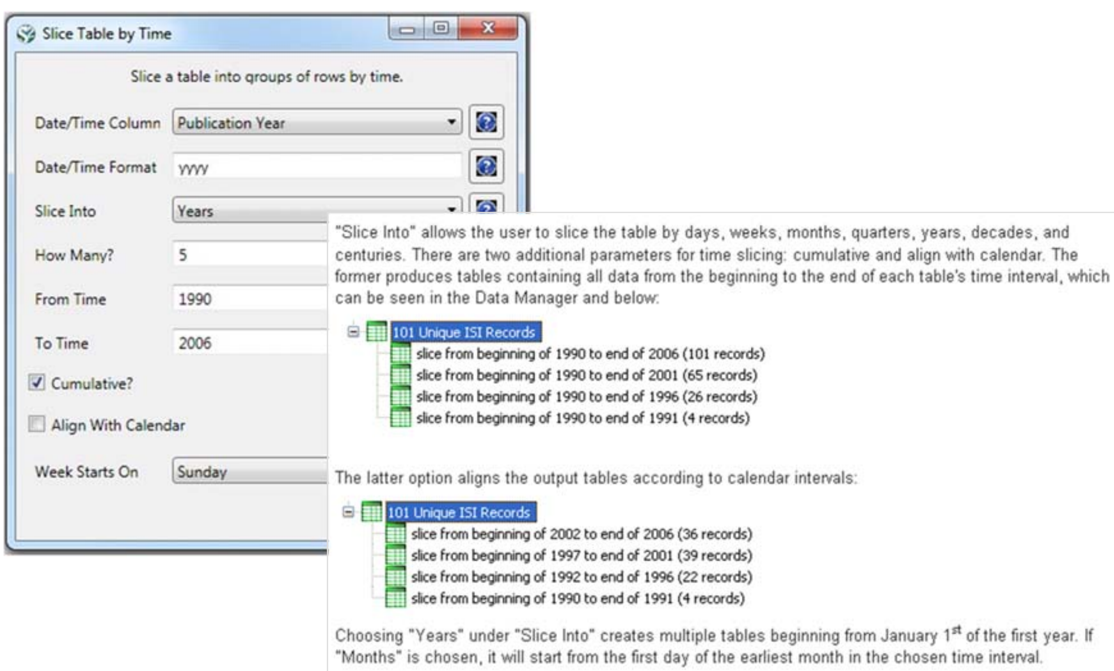


Relevant Sci2 Manual entry



http://sci2.wiki.cns.iu.edu/5.1.2+Time+Slicing+of+Co-Authorship+Networks+(ISI+Data)

Slice Table by Time



http://sci2.wiki.cns.iu.edu/5.1.2+Time+Slicing+of+Co-Authorship+Networks+(ISI+Data)

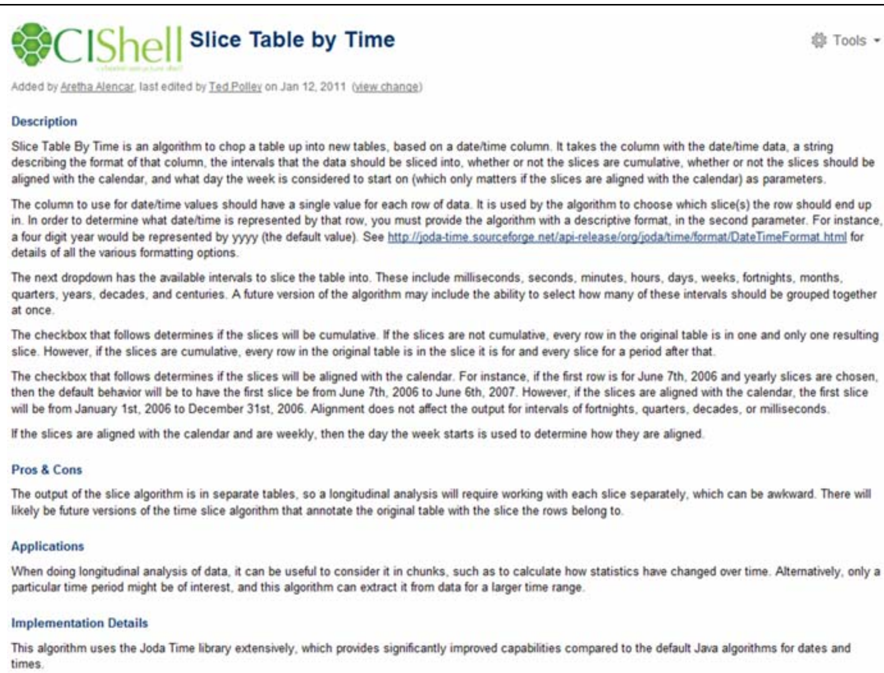
Visualize Each Network, Keep Node Positions

1. To see the evolution of Vespignani's co-authorship network over time, check '*cumulative*'.

2. Extract co-authorship networks one at a time for each sliced time table using *'Data Preparation > Extract Co-Author Network',* making sure to select "ISI" from the pop-up window during the extraction.
3. To view each of the Co-Authorship Networks over time using the same graph layout, begin by clicking on longest slice network (the *'Extracted Co-Authorship Network*' under *'slice from beginning of 1990 to end of 2006 (101 records)')* in the data manager. Visualize it in GUESS using *'Visualization > Networks > GUESS'*.
4. From here, run *'Layout > GEM*' followed by *'Layout > Bin Pack*'. Run *'Script > Run Script* …' and select '*yoursci2directory/scripts/GUESS/co-author-nw.py*'.
5. In order to save the x, y coordinates of each node and to apply them to the other time slices in GUESS, select *'File > Export Node Positions*' and save the result as '*yoursci2directory/NodePositions.csv*'. Load the remaining three networks in GUESS using the steps described above and for each network visualization, run *'File > Import Node Positions'* and open *'yoursci2directory/NodePositions.csv'*.
6. To match the resulting networks stylistically with the original visualization, run *'Script > Run Script …'* and select *'yoursci2directory/scripts/GUESS/co-author-nw.py', followed by 'Layout > Bin Pack',* for each.

http://sci2.wiki.cns.iu.edu/5.1.2+Time+Slicing+of+Co-Authorship+Networks+(ISI+Data)

Relevant CIShell plugin



CIShell **Slice Table by Time** Tools

Added by Aretha Alencar, last edited by Ted Polley on Jan 12, 2011 (view change)

Description

Slice Table By Time is an algorithm to chop a table up into new tables, based on a date/time column. It takes the column with the date/time data, a string describing the format of that column, the intervals that the data should be sliced into, whether or not the slices are cumulative, whether or not the slices should be aligned with the calendar, and what day the week is considered to start on (which only matters if the slices are aligned with the calendar) as parameters.

The column to use for date/time values should have a single value for each row of data. It is used by the algorithm to choose which slice(s) the row should end up in. In order to determine what date/time is represented by that row, you must provide the algorithm with a descriptive format, in the second parameter. For instance, a four digit year would be represented by yyyy (the default value). See http://joda-time.sourceforge.net/api-release/org/joda/time/format/DateTimeFormat.html for details of all the various formatting options.

The next dropdown has the available intervals to slice the table into. These include milliseconds, seconds, minutes, hours, days, weeks, fortnights, months, quarters, years, decades, and centuries. A future version of the algorithm may include the ability to select how many of these intervals should be grouped together at once.

The checkbox that follows determines if the slices will be cumulative. If the slices are not cumulative, every row in the original table is in one and only one resulting slice. However, if the slices are cumulative, every row in the original table is in the slice it is for and every slice for a period after that.

The checkbox that follows determines if the slices will be aligned with the calendar. For instance, if the first row is for June 7th, 2006 and yearly slices are chosen, then the default behavior will be to have the first slice be from June 7th, 2006 to June 6th, 2007. However, if the slices are aligned with the calendar, the first slice will be from January 1st, 2006 to December 31st, 2006. Alignment does not affect the output for intervals of fortnights, quarters, decades, or milliseconds.

If the slices are aligned with the calendar and are weekly, then the day the week starts is used to determine how they are aligned.

Pros & Cons

The output of the slice algorithm is in separate tables, so a longitudinal analysis will require working with each slice separately, which can be awkward. There will likely be future versions of the time slice algorithm that annotate the original table with the slice the rows belong to.

Applications

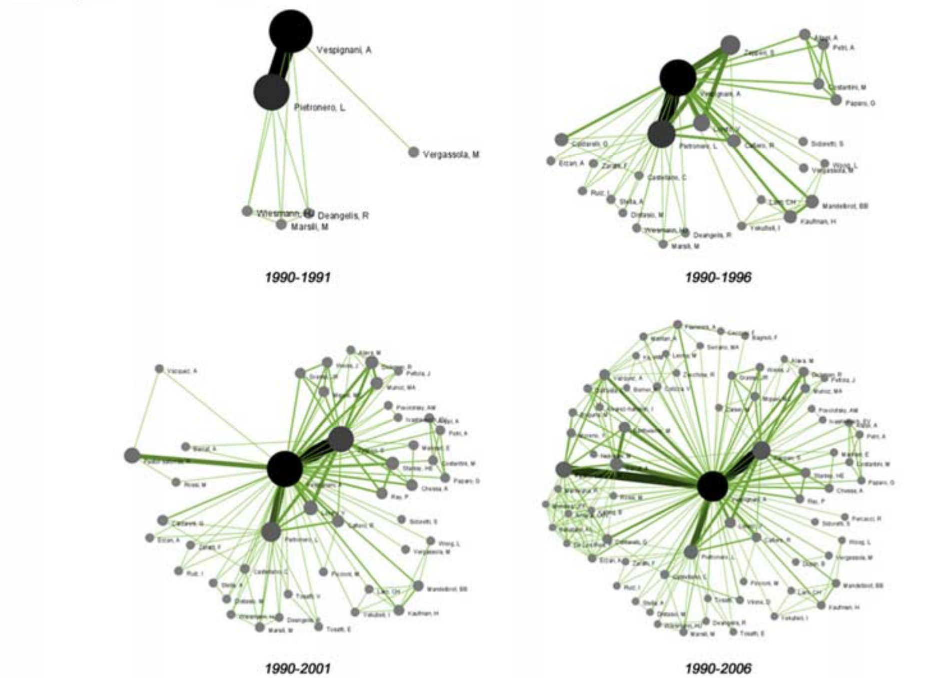
When doing longitudinal analysis of data, it can be useful to consider it in chunks, such as to calculate how statistics have changed over time. Alternatively, only a particular time period might be of interest, and this algorithm can extract it from data for a larger time range.

Implementation Details

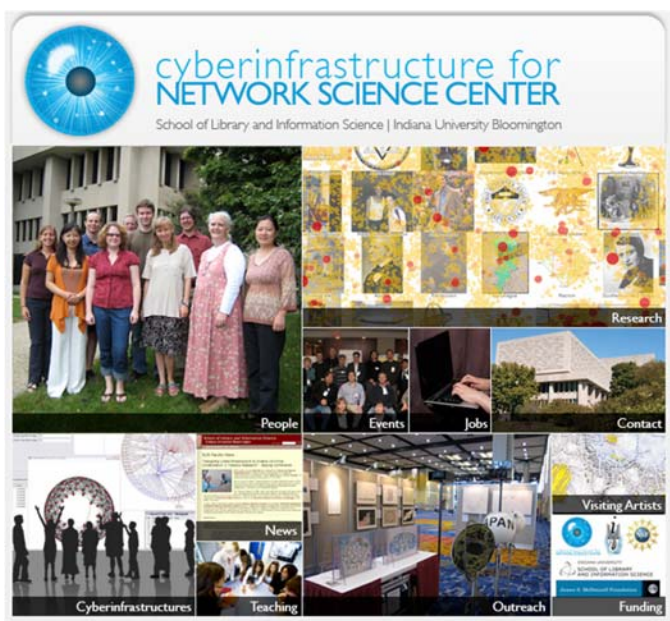
This algorithm uses the Joda Time library extensively, which provides significantly improved capabilities compared to the default Java algorithms for dates and times.

http://cishell.wiki.cns.iu.edu/Slice+Table+by+Time

Visualize Each Network, Keep Node Positions



http://sci2.wiki.cns.iu.edu/5.1.2+Time+Slicing+of+Co-Authorship+Networks+(ISI+Data)



All papers, maps, tools, talks, press are linked from http://cns.iu.edu

CNS Facebook: http://www.facebook.com/cnscenter
Mapping Science Exhibit Facebook: http://www.facebook.com/mappingscience