Science of Science Research and Tools
Tutorial #07 of 12

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

Invited by Robin M. Wagner, Ph.D., M.S.
Chief Reporting Branch, Division of Information Services
Office of Research Information Systems, Office of Extramural Research
Office of the Director, National Institutes of Health

Suite 4090, 6705 Rockledge Drive, Bethesda, MD 20892
10a-noon, July 19, 2010

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**

12 Tutorials in 12 Days at NIH—Overview

[#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

Recommended Reading

- NWB Team (2009) Network Workbench Tool, User Manual 1.0.0, http://nwb.slis.indiana.edu/Docs/NWBTool-Manual.pdf
- Pat Hanrahan. To Draw a Tree. http://www-graphics.stanford.edu/~hanrahan/talks/todrawatree

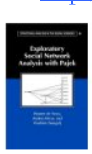
12 Tutorials in 12 Days at NIH—Overview

[#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

Recommended Reading

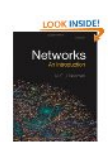
- NWB Team (2009) Network Workbench Tool, User Manual 1.0.0, http://nwb.slis.indiana.edu/Docs/NWBTool-Manual.pdf



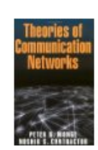
Exploratory Social Network Analysis with Pajek by de Nooy, Wouter
(9)
$35.19

Models and Methods in Social Network Analysis by Peter J. Carrington
(1)
$18.14

Social Network Analysis: Methods and Applications by Katherine Faust
(9)
$31.20



Networks: An Introduction by Mark Newman
$68.90



Theories of Communication Networks by Peter R. Monge
(7)
$19.25

12 Tutorials in 12 Days at NIH—Overview

[#09] Large Network Analysis and Visualization

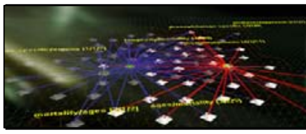
- General Overview
- Designing Effective Network Visualizations
- Sci2-Reading and Modeling Networks
- Sci2-Analysing Large Networks
- Sci2-Visualizing Large Networks and Distributions
- Outlook
- Exercise: Identify Promising Large Network Analyses of NIH Data

Recommended Reading

- NWB Team (2009) Network Workbench Tool, User Manual 1.0.0, http://nwb.slis.indiana.edu/Docs/NWBTool-Manual.pdf
- 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

[#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 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

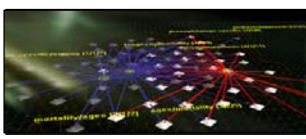
Decision Processes
- Indices or search trees
- Decision trees

Goals & Objectives

Representing hierarchical data
- Structural information
- Content information

Objectives
- Efficient Space Utilization
- Interactivity
- Comprehension
- Esthetics

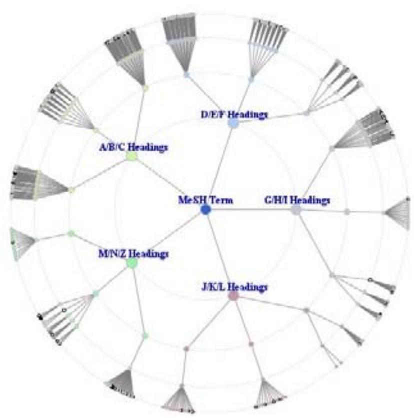
Pat Hanrahan, Stanford U
http://www-graphics.stanford.edu/~hanrahan/talks/todrawatree/



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.



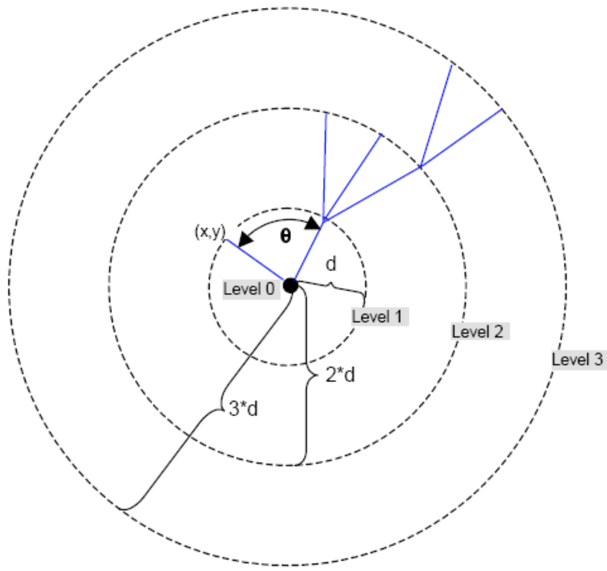
Greg Book & Neeta Keshary (2001) Radial Tree Graph Drawing Algorithm for Representing Large Hierarchies. University of Connecticut Class Project.

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 360º 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

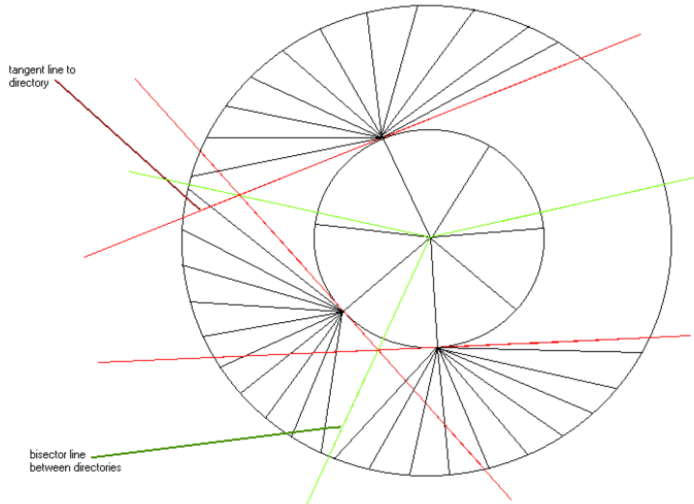
Use information on number of parents, their location, and their space for children to place all level x nodes.

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.

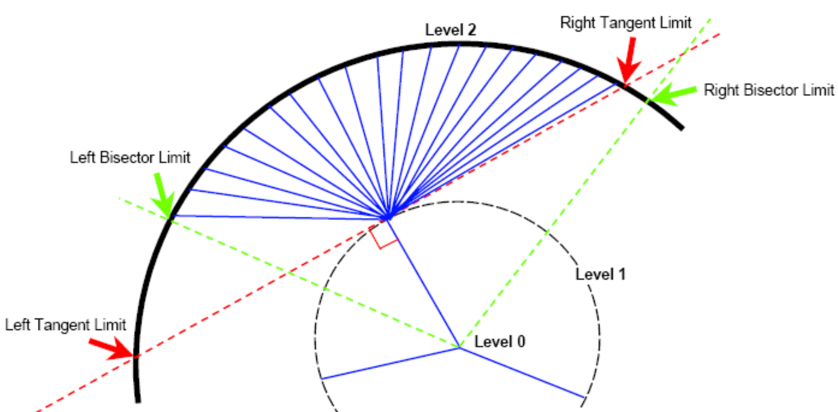
We then iterate through all the nodes at level 1 and calculate the position of the node Bisector Limits



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

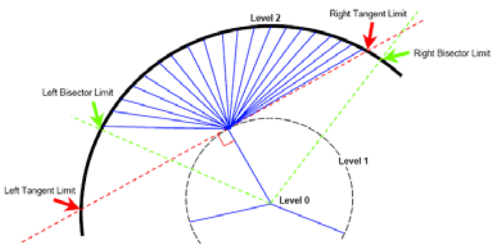
case 0:

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:



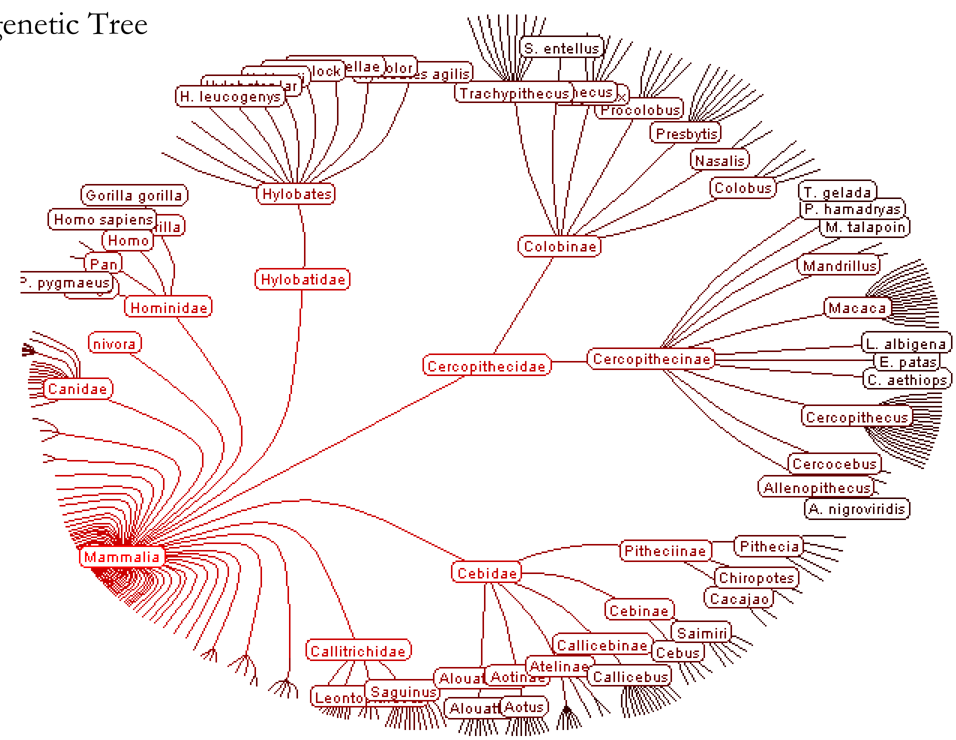
Nodes in levels two and higher must be grouped according to their parent.
Loop through all nodes at this level and get a list of the parent nodes
And get the number of children for each parent
Calculate the AngleSpace for each parent:
AngleSpace = (leftLimit – rightLimit)/NumNodesForThisParent
Foreach 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

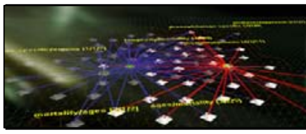
End switch

Hyperbolic Tree – How does it work?

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

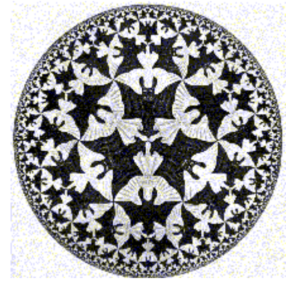
Phylogenetic Tree





Hyperbolic Geometry

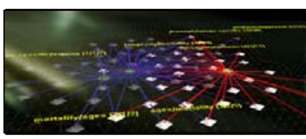
Inspired by Escher's Circle Limit IV (Heaven and Hell), 1960.



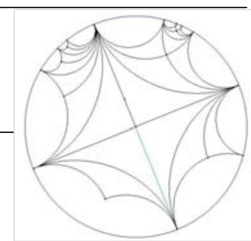
- Focus+context technique for visualizing large hierarchies
- Continuous redirection of the focus possible.

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.)

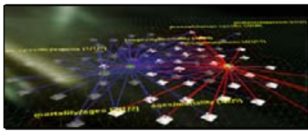
Map hyperbolic plane onto the unit disk

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.

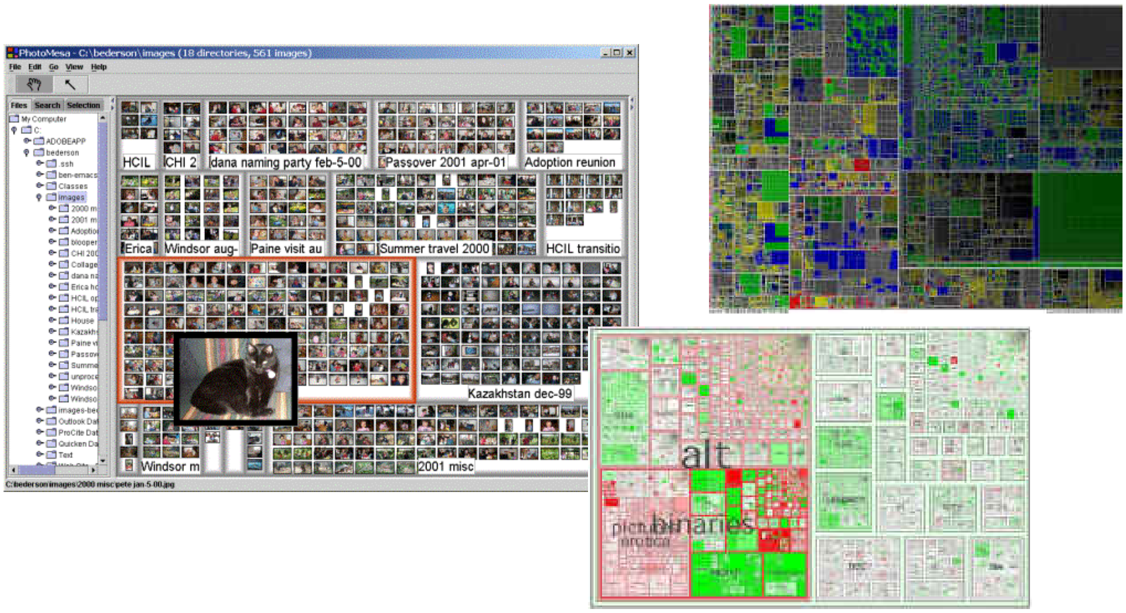
Change of Focus – Animated Transitions

Node & Edge Information

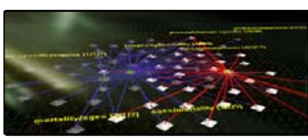


Treemap – How does it work?

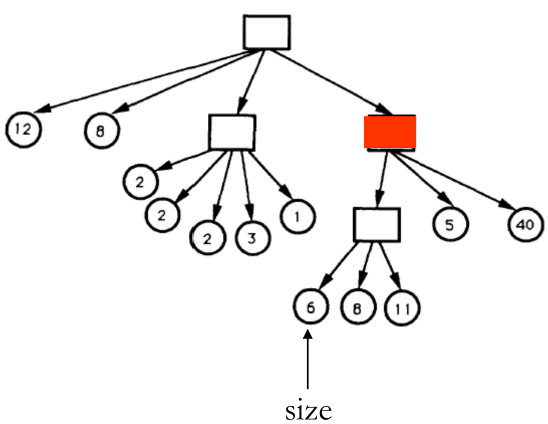
See also http://sw.slis.indiana.edu/sw/treemap.html

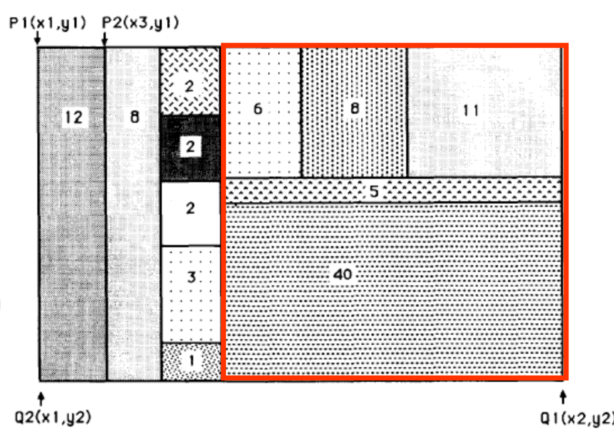


Shneiderman, B. (1992) Tree visualization with tree-maps: 2-d space-filling approach. ACM Transactions on Graphics 11, 1 (Jan. 1992), pp 92 - 99. See also http://www.cs.umd.edu/hcil/treemaps/

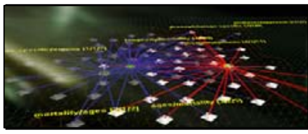


Treemaps – Layout





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

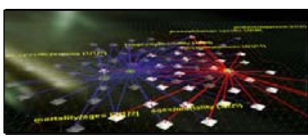


Treemap – Pseudo Code

```
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]);
   }
```

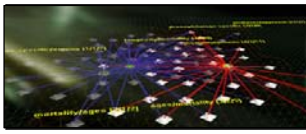
Treemap – Properties

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

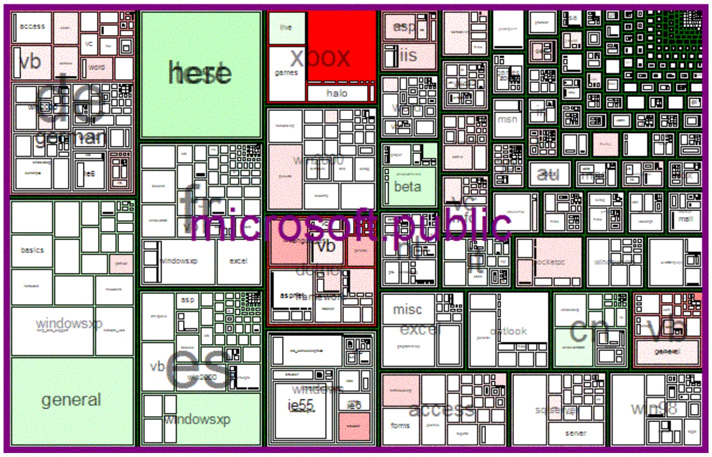
- Size comparison is difficult
- Labeling is a problem.
- Cluttered display
- Difficult to discern boundaries
- Shows only leaf content information



Treemap – Algorithm Improvements

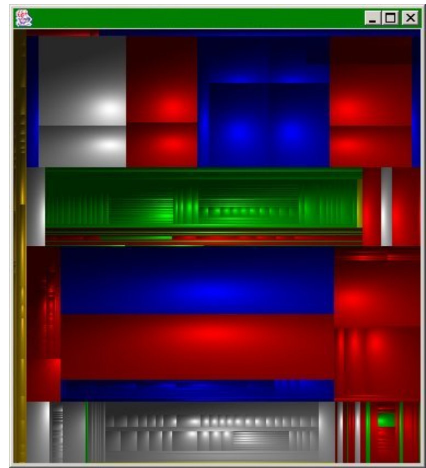
Sorted treemap

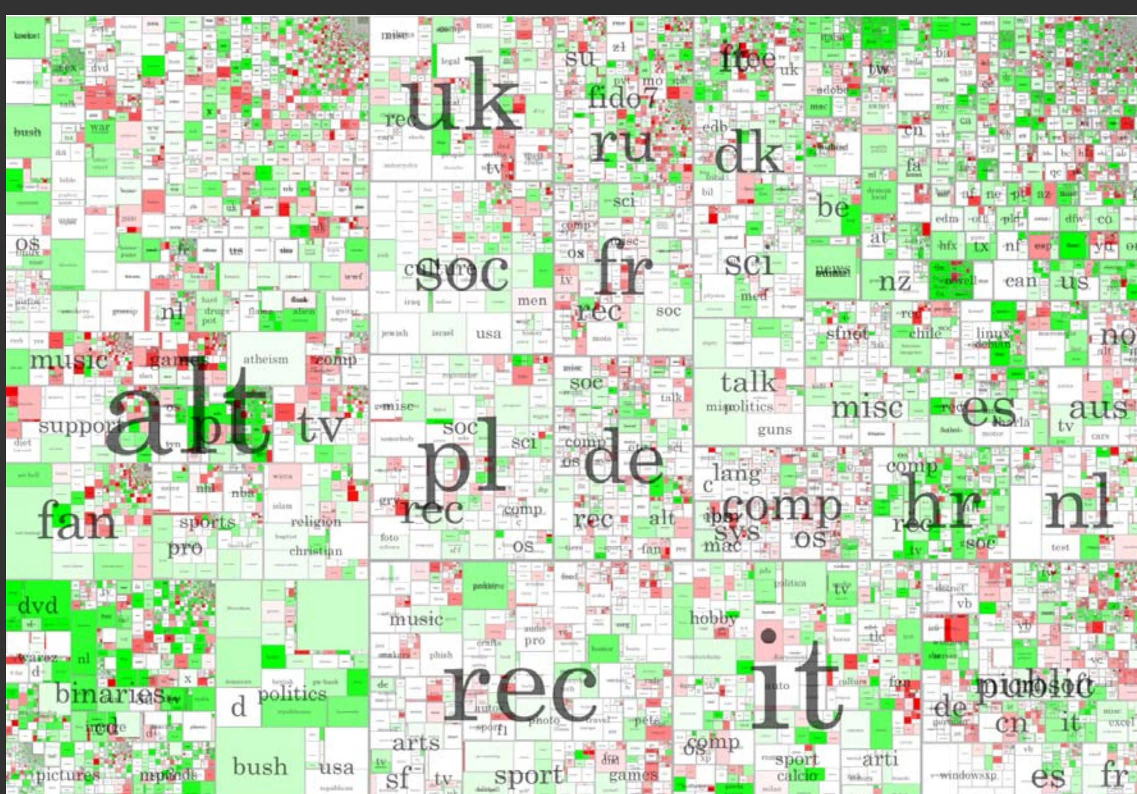
Marc Smith



Cushion treemap

[*http://treemap.sourceforge.net/*](http://treemap.sourceforge.net/)

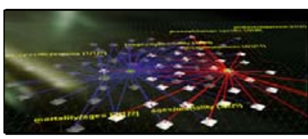




Treemap View of 2004 Usenet Returnees - Marc Smith, Danyel Fisher, Tony Capone - 2005

[#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.

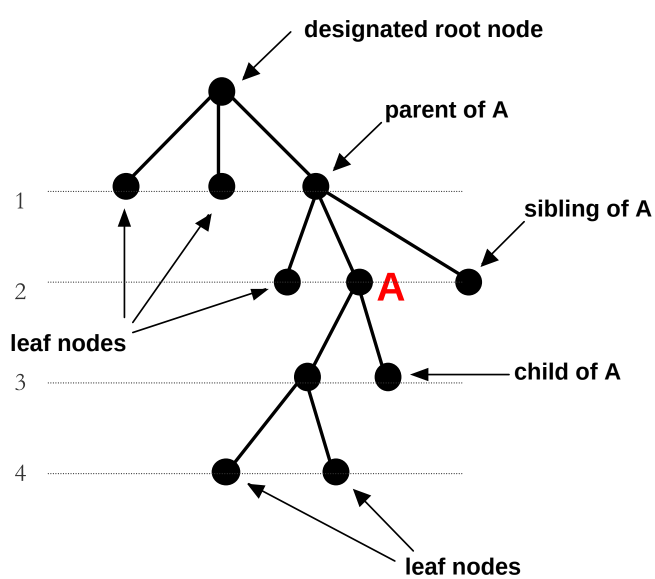
A **leaf node** has no children.

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.

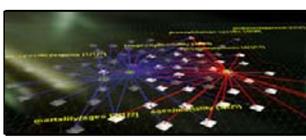
Sample tree of

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

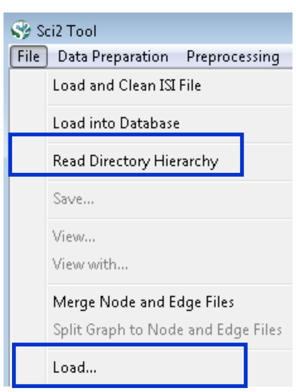


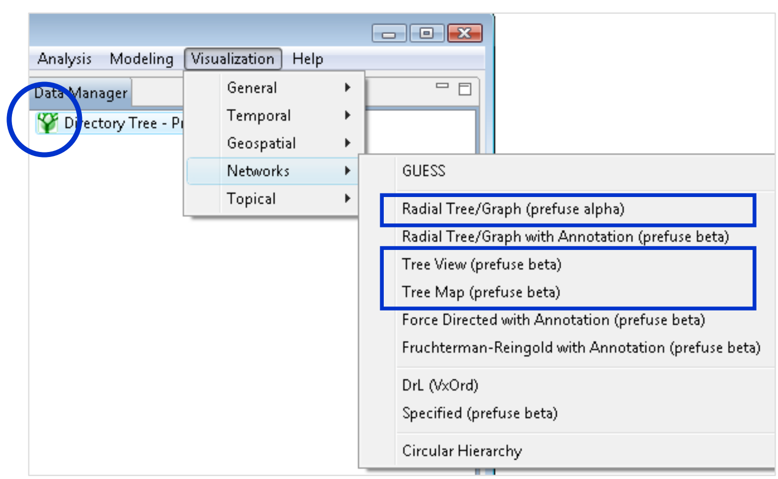
[#07] Tree Analysis and Visualization

- 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

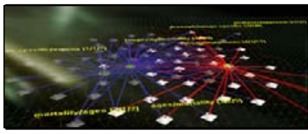


Read and Visualize Trees with Sci2 Tool



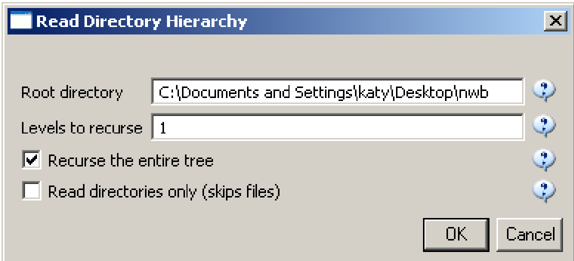


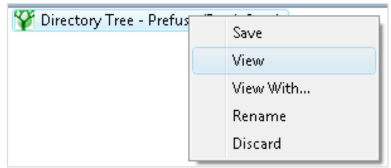
See Science of Science (Sci2) Tool User Manual, Version Alpha 3, Section 3.1 for a listing and brief explanations of all plugins. http://sci.slis.indiana.edu/registration/docs/Sci2_Tutorial.pdf



Sample Tree: Read Directory Hierarchy

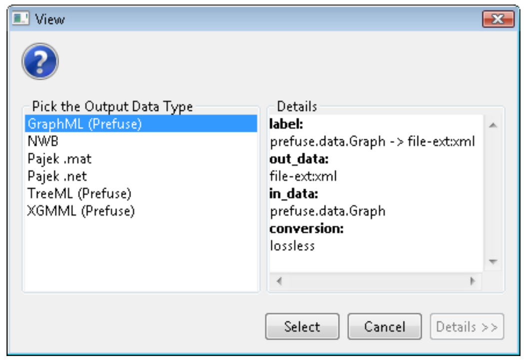
Use *'File > Read Directory Hierarchy'* with parameters

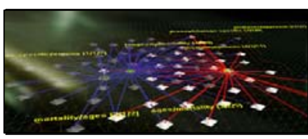




To view file in different formats right click *'Directory Tree - Prefuse (Beta) Graph'* in Data Manager and select *View*.

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

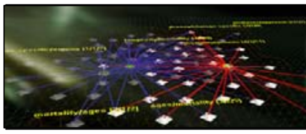
```
*Vertices 568
1 sci2-with-scimaps
2 .eclipseproduct
3 sci2.exe
4 sci2.ini
5 configuration
6 config.ini
7 default_menu.xml
...

*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
```


Network – .NET

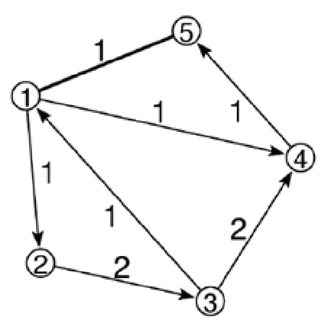
Network can be defined in different ways on input file. Look at three of them:

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
```

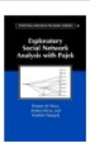

3. Matrix

```
*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)

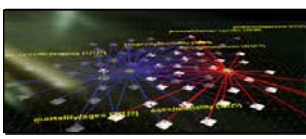
See documentation at https://nwb.slis.indiana.edu/community/?n=DataFormats.HomePage

```
- <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

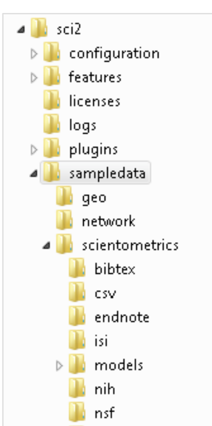


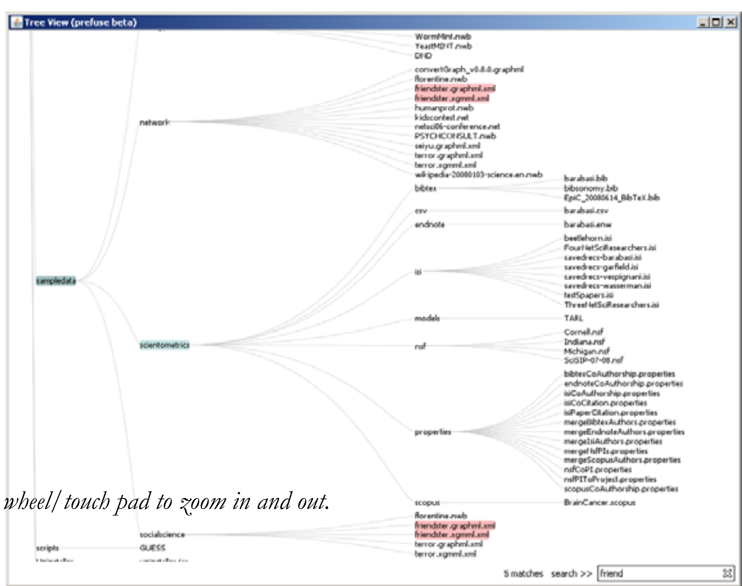
Sample Tree Visualizations

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)'*

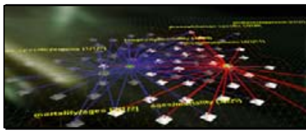




Press right mouse button and use mouse wheel/touch pad to zoom in and out.

Click on directory to expand/collapse.

Use search field to find specific files.

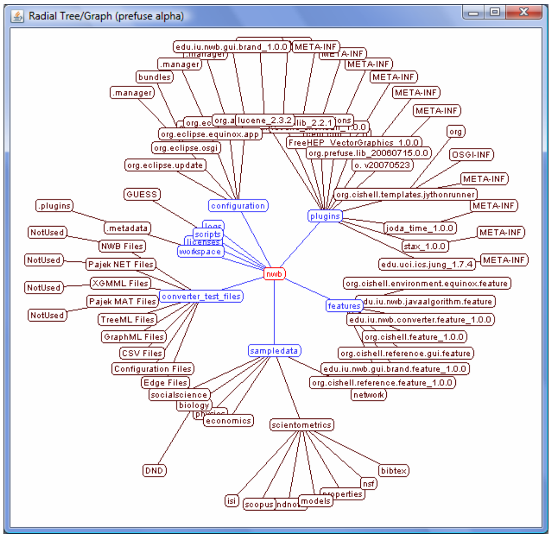


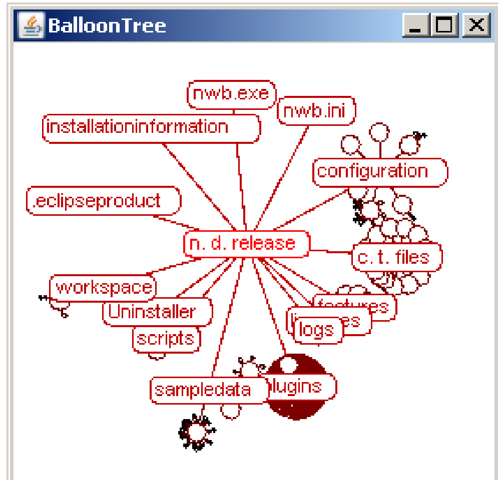
Sample Tree Visualizations

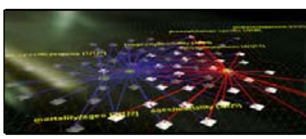
Radial Tree and ***Ballon Tree*** showing the structure of, e.g., directory hierarchies.

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)*





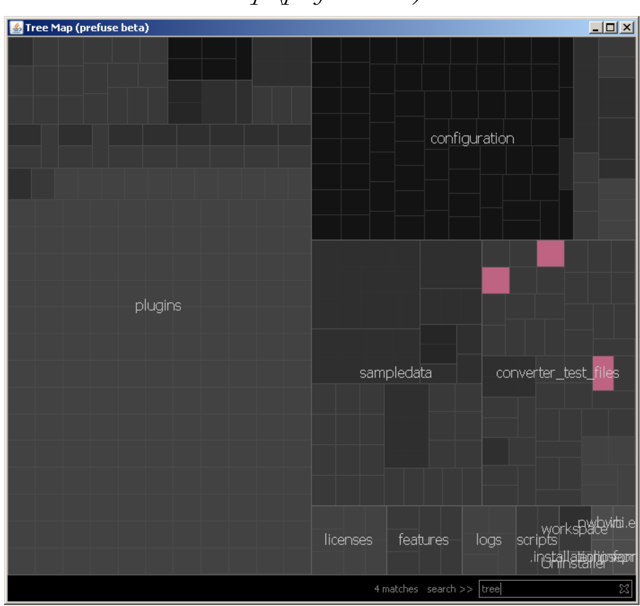


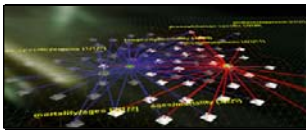
Sample Tree Visualization

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

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

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



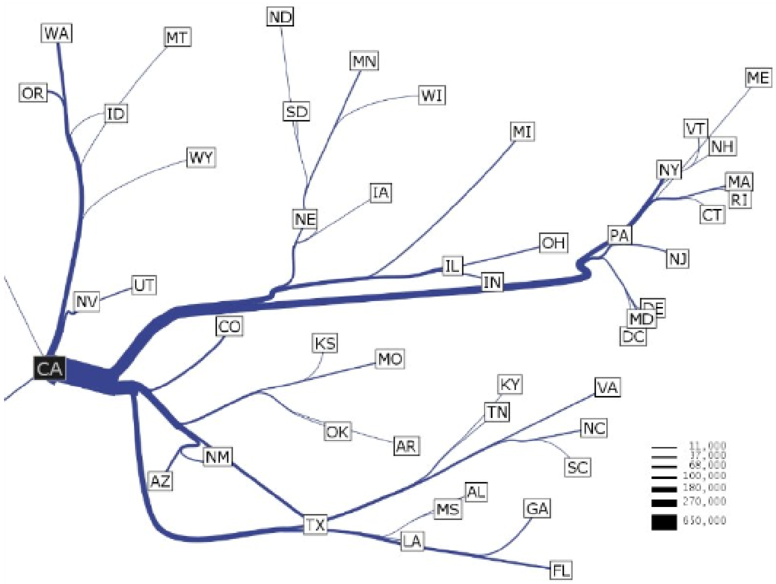


Sample Tree Visualization

Flow Maps showing migration patterns

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

Soon available in Sci2 Tool.



[#07] Tree Analysis and Visualization

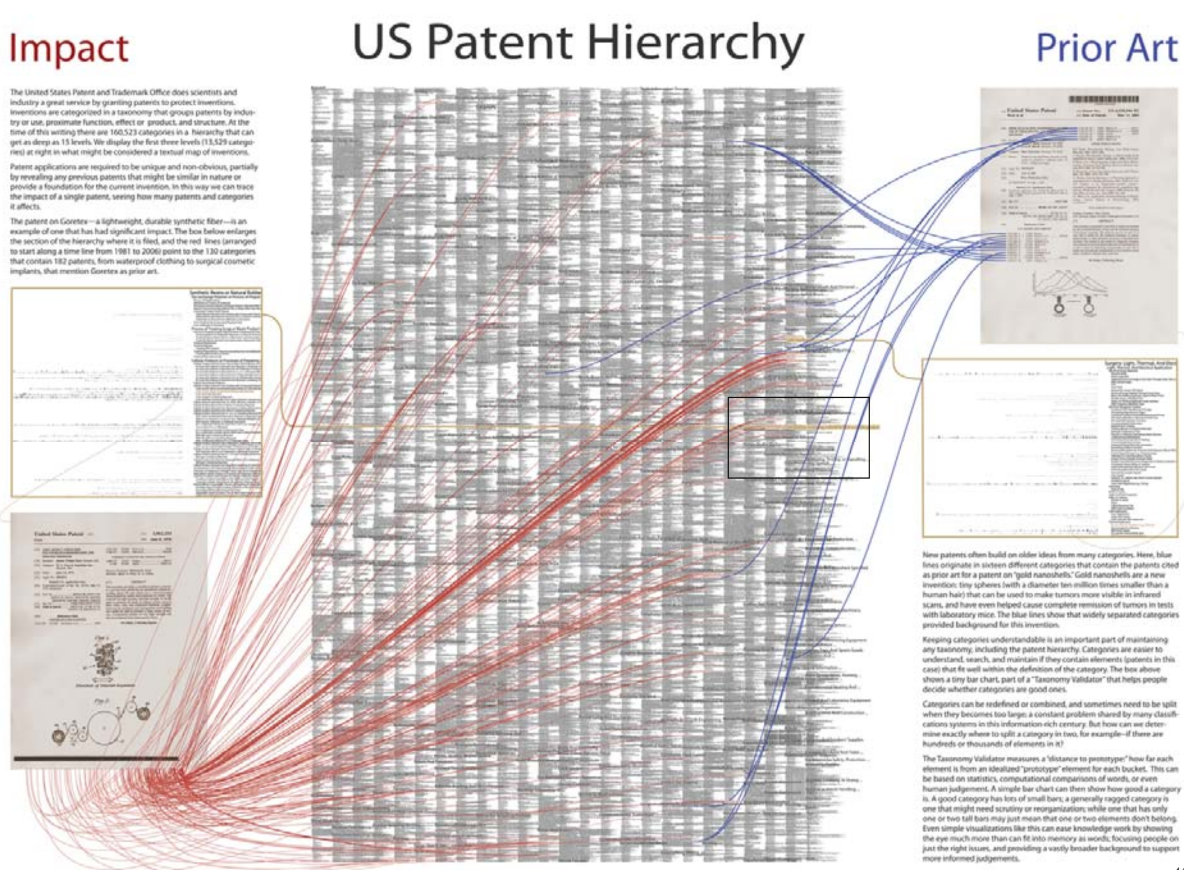
- General Overview
- Designing Effective Tree Visualizations
- Notions and Notations
- Sci2-Reading and Extracting Trees
- Sci2-Visualizing Trees
- Outlook
- Exercise: Identify Promising Tree Analyses of NIH Data

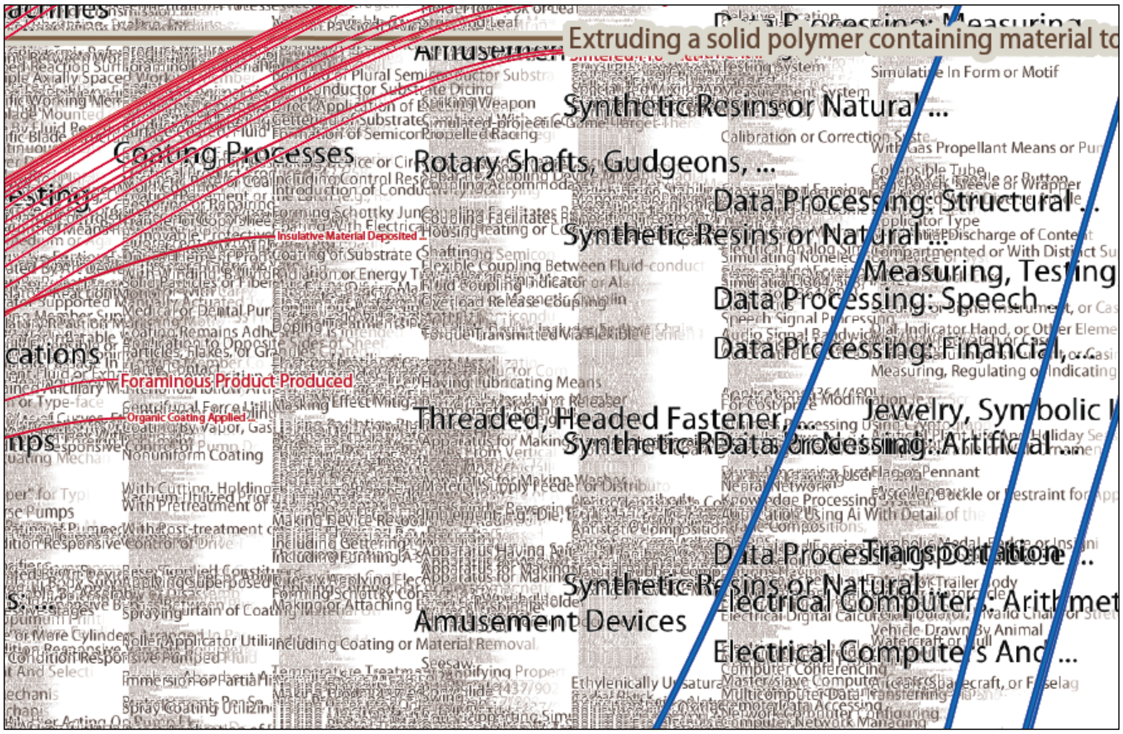
Outlook

Planned extensions of Sci2 Tool:

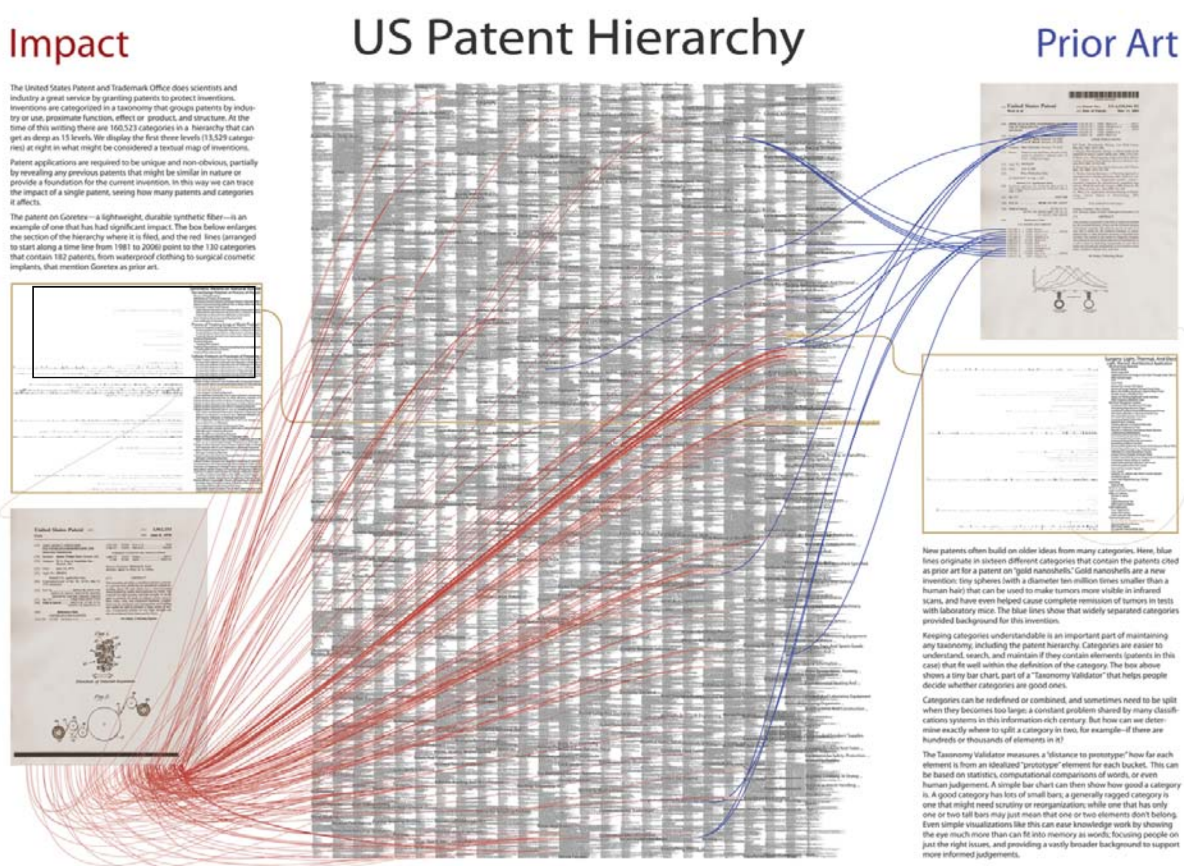
- (Flowmap) tree network overlays for geo maps and science maps.
- Bimodal network visualizations.
- Scalable visualizations of large hierarchies.

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

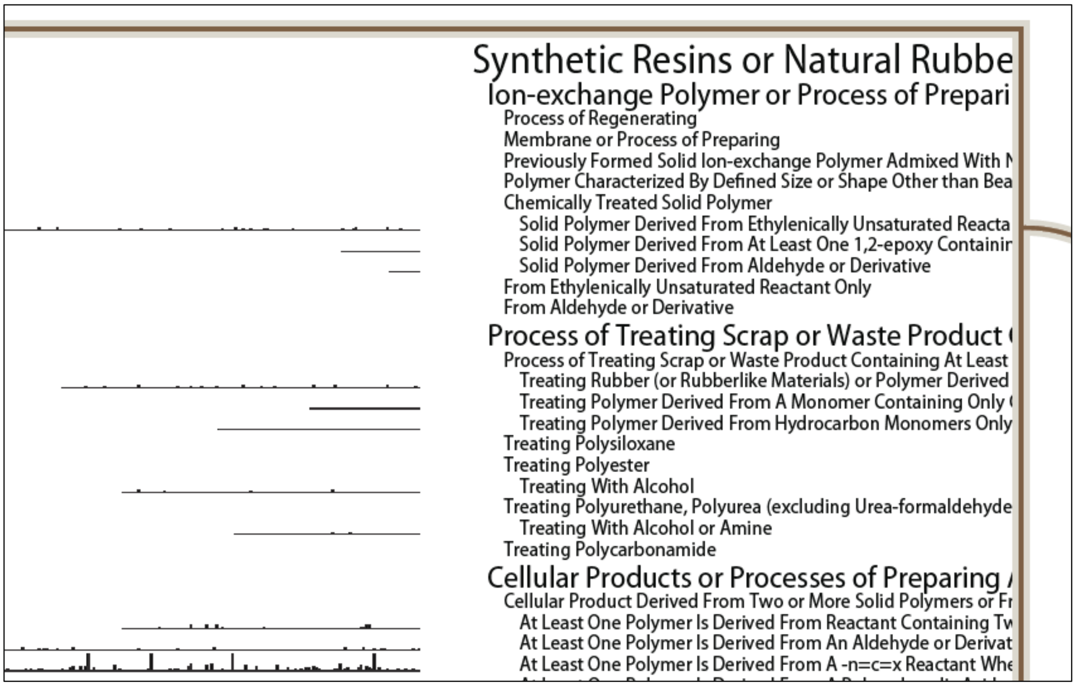




Extruding a solid polymer containing material to
Synthetic Resins or Natural ...
Coating Processes
Rotary Shafts, Gudgeons, ...
Data Processing: Structural ...
Synthetic Resins or Natural ...
Measuring, Testing
Data Processing: Speech
Data Processing: Financial, ...
Foraminous Product Produced
Threaded, Headed Fastener, ...
Jewelry, Symbolic
Amusement Devices
Electrical Computers: Arithmet
Electrical Computers And ...



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.



[#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

Exercise

Please identify a promising tree analyses of NIH data.

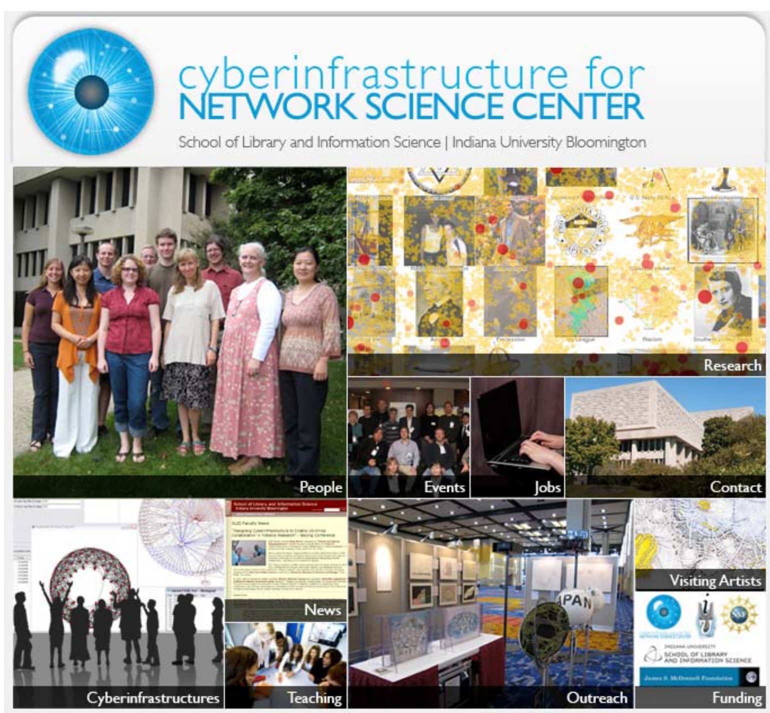
Document it by listing
- Project title
- User, i.e., who would be most interested in the result?
- Insight need addressed, i.e., what would you/user like to understand?
- Data used, be as specific as possible.
- Analysis algorithms used.
- Visualization generated. Please make a sketch with legend.

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



All papers, maps, cyberinfrastructures, talks, press are linked from http://cns.slis.indiana.edu