Graphs and Networks 1

CS 4460 – Intro. to Information Visualization
November 6, 2017
John Stasko

Learning Objectives

- Define network concepts
  - vertex, edge, cycle, degree, direction
- Describe different node-link design choices
  - color, width, position, shape, size, label, form
- Enumerate primary aesthetic considerations for layouts
  - edge crossings, clusters, symmetry, edge lengths
- List example tasks for network data
- Explain "ball of string/hairball" problem
- List common layout approaches and describe characteristics of each
  - hierarchical, force-directed, circular, geo, matrix
- Define "edge bundling"

Connections

- Connections throughout our lives and the world
  - Circle of friends
  - Delta's flight plans
  - …
- Model connected set as a *Graph*

What is a Graph?

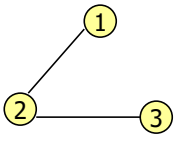
- Vertices (nodes) connected by
- Edges (links)

Adjacency list

1: 2
2: 1, 3
3: 2

	1	2	3
1	0	1	0
2	1	0	1
3	0	1	0

Adjacency matrix



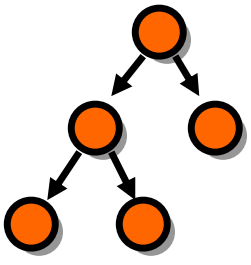
Drawing

Graph Terminology

- Graphs can have *cycles*
- Graph edges can be *directed* or *undirected*
- The *degree* of a vertex is the number of edges connected to it
  - *In-degree* and *out-degree* for directed graphs
- Graph edges can have values (*weights*) on them (nominal, ordinal or quantitative)

Trees are Different

- Subcase of general graph
- No cycles
- Typically directed edges
- Special designated root vertex



Saw last week

Graph Uses

- In information visualization, any number of data sets can be modeled as a graph
  - US telephone system
  - World Wide Web
  - Distribution network for on-line retailer
  - Call graph of a large software system
  - Semantic map in an AI algorithm
  - Set of connected friends
- Graph/network visualization is one of the oldest and most studied areas of InfoVis

Graph Visualization Challenges

- Graph layout and positioning
  - Make a concrete rendering of abstract graph
- Navigation/Interaction
  - How to support user changing focus and moving around the graph

Scale Challenge

- Previous two issues not too bad for small graphs, but large ones are much tougher
- May run out of space for vertices and edges (turns into "ball of string")
- Can really slow down algorithm

- Sometimes use *clustering* to help
  - Extract highly connected sets of vertices
  - Collapse some vertices together

Navigation/Interaction Challenge

- How do we allow a user to query, visit, or move around a graph?
- Changing focus may entail a different rendering

Layout Examples

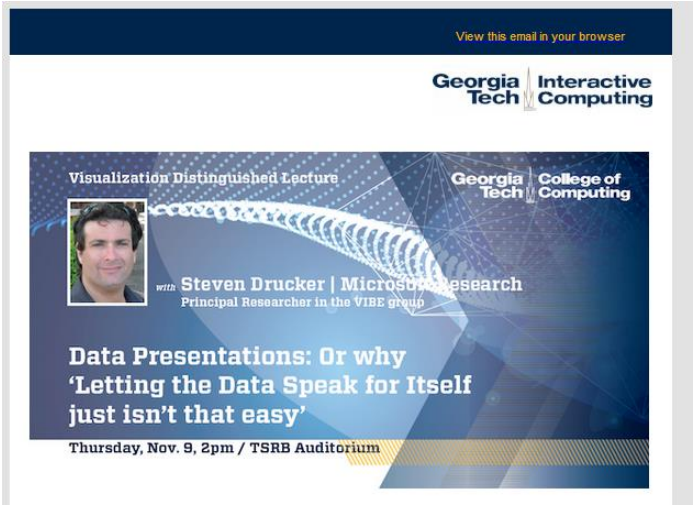
- Homework assignment
- Let's judge!

Results

- What led to particular layouts being liked more?
- Discuss

Talk Thursday

2-3 pm
TSRB Auditorium



Quiz 4

CS 4460, Fall 2017 Name: _____
Pop Quiz 4
11/1/17

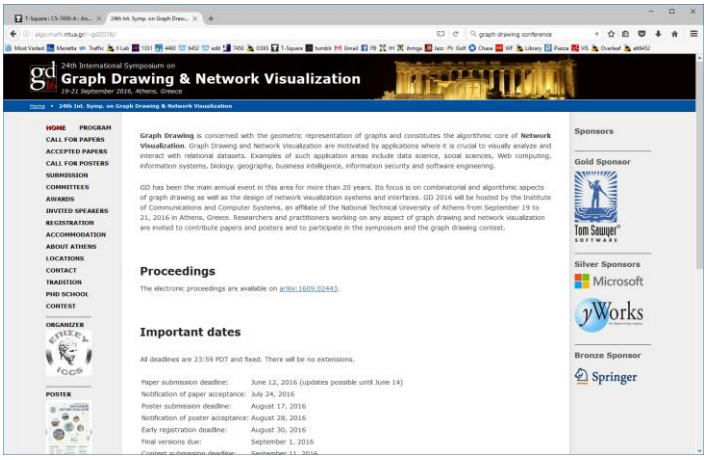
1. Draw the slice-and-dice treemap representation for the hierarchy below. It uses this notation
Node: Child1, Child2, … (order means nothing)

A: B, C, D
B: E, F, G, H
D: I, J
J: K, L

Values:
C-3, E-1, F-1, G-4, H-6, I-3, J-6, K-2, L-4

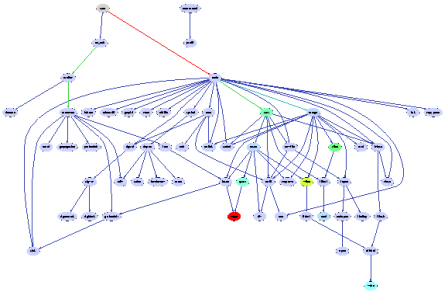
Graph Drawing

Entire research community's focus

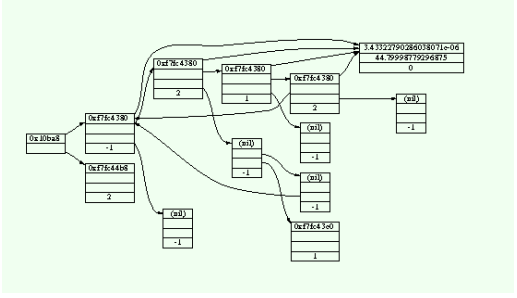


Vertex Issues

- Shape
- Color
- Size
- Location
- Label



Edge Issues



- Color
- Size
- Label
- Form
  - Polyline, straight line, orthogonal, grid, curved, planar, upward/downward, ...

Aesthetic Considerations

- Develop a set of metrics to quantitatively rate the "goodness" of a graph layout
- What metrics would you use?

Aesthetic Considerations

- ***Crossings*** -- minimize towards planar
- ***Total Edge Length*** -- minimize towards proper scale
- ***Area*** -- minimize towards efficiency
- ***Maximum Edge Length*** -- minimize longest edge
- ***Uniform Edge Lengths*** -- minimize variances
- ***Total Bends*** -- minimize orthogonal towards straight-line

Which Matters?

- Various studies examined which of the aesthetic factors matter most and/or what kinds of layout/vis techniques look best
  - Purchase, Graph Drawing '97
  - Ware et al, *Info Vis* 1(2)
  - Ghoniem et al, *Info Vis* 4(2)
  - van Ham & Rogowitz, *TVCG* '08
  - …
- Results mixed: Edge crossings do seem important

Shneiderman's NetViz Nirvana

1) Every node is visible

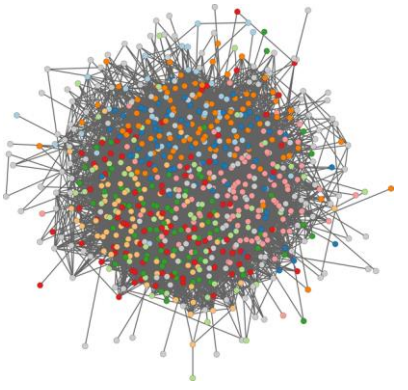
2) For every node you can count its degree

3) For every link you can follow it from source to destination

4) Clusters and outliers are identifiable

Classic Problem

- With enough vertices and enough edges, you get…
- A hairball! (ball-of-string)



http://visone.info/wiki/images/b/b7/Caltech36-hairball.png

But What about User Tasks?

- So what do people want to do with or learn from network visualizations?
  - Recurring theme of this class: Too often this is neglected

Graph Vis Task Taxonomy

- Start with Amar et al '05 low-level tasks (retrieve value, find extreme, sort, etc.)
- Then add four types of other tasks (next pages)

Lee et al
BELIV '06

Graph Vis Task Taxonomy

- 1. Topology-based tasks
  - Adjacency
    - Find the set of nodes adjacent to a node
  - Accessibility
    - Find the set of nodes accessible to a node
  - Common connection
    - Given nodes, find the set of nodes connected to all
  - Connectivity
    - Find shortest path
    - Identify clusters
    - Identify connected components

Graph Vis Task Taxonomy

- 2. Attribute-based tasks
  - On the nodes
    - Find the nodes having a specific attribute value
  - On the edges
    - Given a node, find the nodes connected only by certain kinds of edges

Graph Vis Task Taxonomy

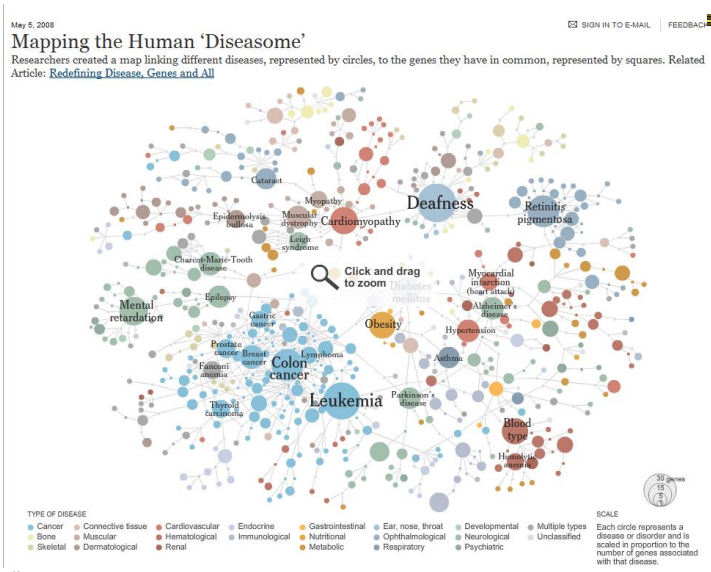
- 3. Browsing tasks
  - Follow path
    - Follow a given path
  - Revisit
    - Return to a previously visited node
- 4. Overview task
  - Compound exploratory task
    - Estimate size of a network
    - Find patterns

Graph Drawing Uses

- Many domains and data sets can benefit significantly from nice graph drawings
- Let's look at some examples…

http://www.nytimes.com/interactive/2008/05/05/science/20080506_DISEASE.html

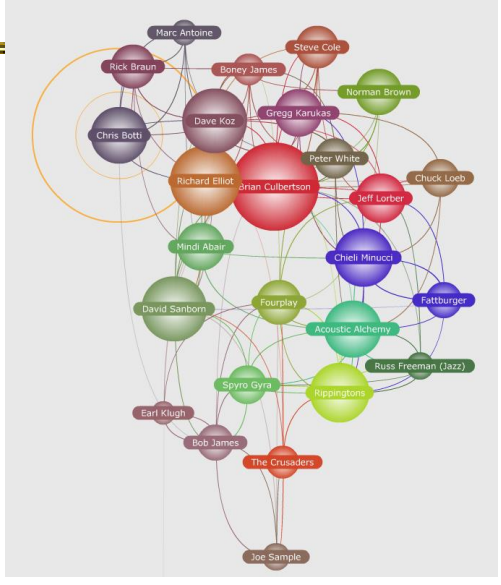
Human Diseases



Note the two extra variables per vertex

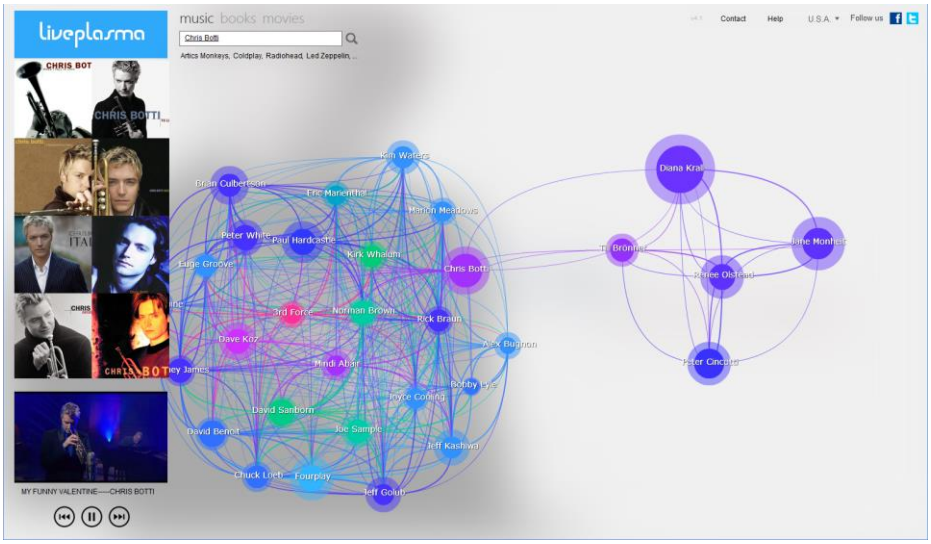
Music Artists

older



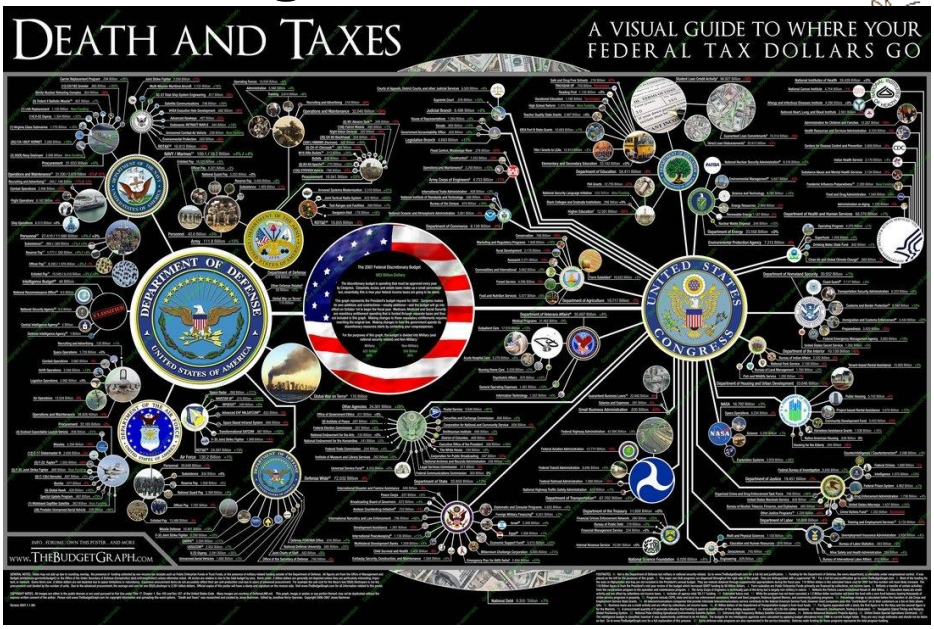
http://www.liveplasma.com/

newer



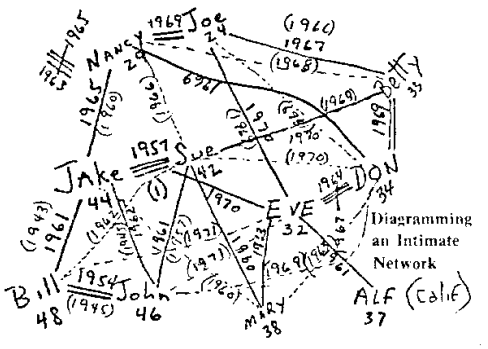
http://mibi.deviantart.com/art/Death-and-Taxes-2007-39894058

US Budget



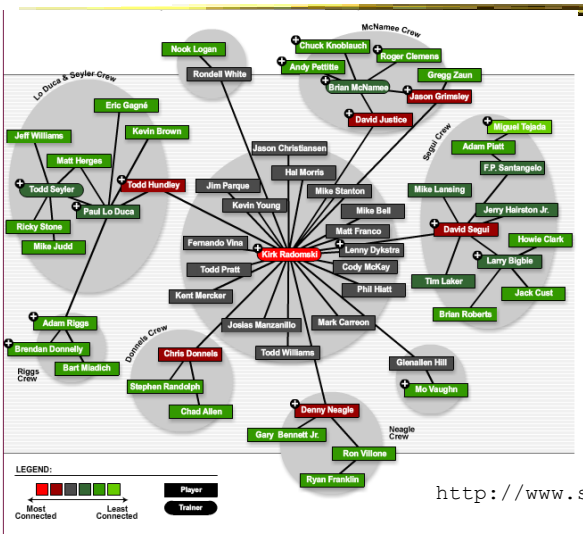
Social Network Visualization

- Social Network Analysis
  - http://www.insna.org



Hot topic again
Why?
Terrorists
Facebook

Steroids in MLB



http://www.slate.com/id/2180392/

Layout Heuristics

- Layout algorithms can be
  - polyline edges
  - planar
    - No edge crossings
  - orthogonal
    - horizontal and vertical lines/polylines
  - grid-based
    - vertices, crossings, edge bends have integer coords
  - curved lines
  - hierarchies
  - circular
  - …

Common Layout Techniques

- Hierarchical
- Force-directed
- Circular
- Geographic-based
- Clustered
- Matrix
- Attribute-based

We will discuss many of these further in the slides to come

Tree Layout (Use Last Week)

- Run a breadth-first search from a vertex
  - This imposes a spanning tree on the graph
- Draw the spanning tree

- Simple and fast, but obviously doesn't represent the whole graph

Hierarchical Layout

Often called Sugiyama layout

Try to impose hierarchy on graph
Reverse edges if needed to remove cycles
Introduce dummy nodes
Put nodes into layers or levels
Order l->r to minimize crossings

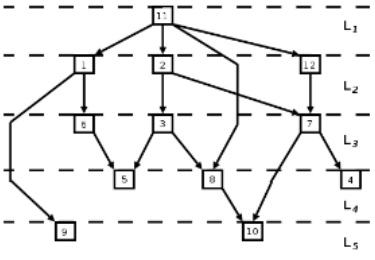


Figure: A graph showing a layered layout, created with the Sugiyama heuristic, with the layers shown. The bends in the edges correspond to dummy nodes.

http://www.csse.monash.edu.au/hons/se-projects/2006/Kieran.Simpson/output/html/node7.html#sugiyamaexample

Force-directed Layout

- Example of constraint-based layout technique
- Impose constraints (objectives) on layout
  - Shorten edges
  - Minimize crossings
  - …
- Define through equations
- Create optimization algorithm that attempts to best satisfy those equations

Force-directed Layout

- Spring model (common)
  - Edges – Springs (gravity attraction)
  - Vertices – Charged particles (repulsion)
- Equations for forces
- Iteratively recalculate to update positions of vertices
- Seeking local minimum of energy
  - Sum of forces on each node is zero

Force-directed Example

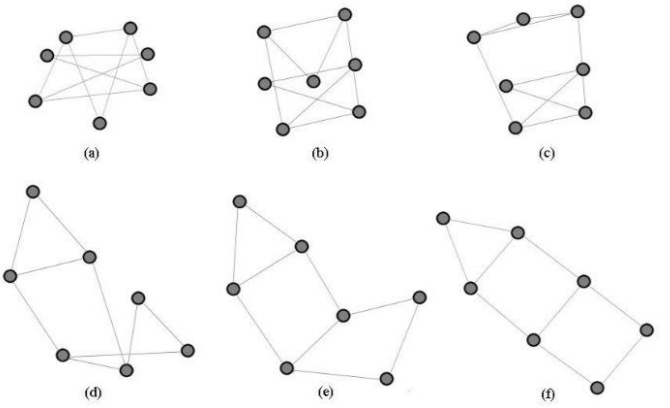
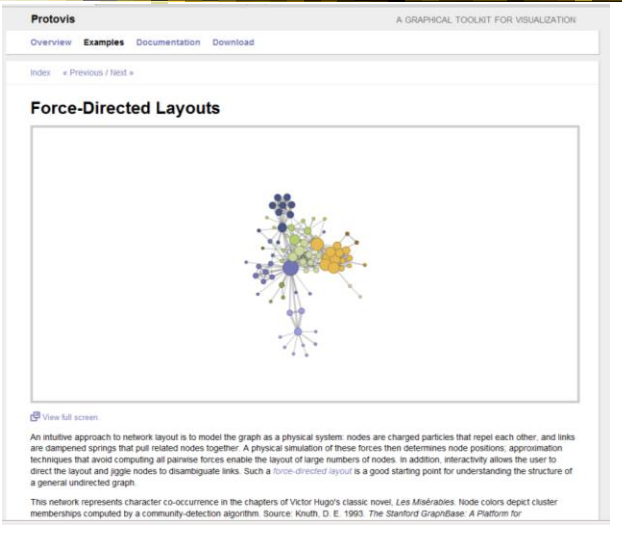


Figure 2: A graph drawing through a number of iterations of a force directed algorithm.

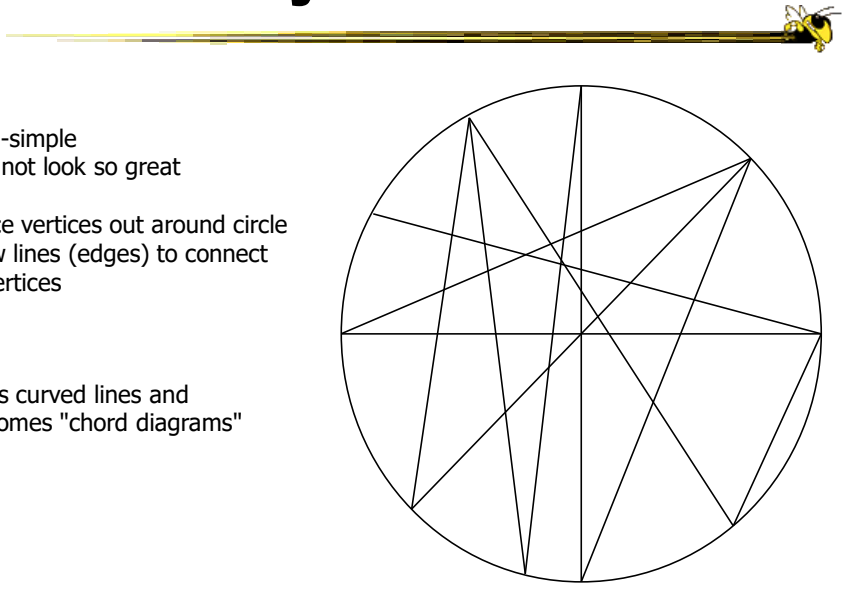
http://www.cs.usyd.edu.au/~aquigley/3dfade/

http://vis.stanford.edu/protovis/ex/force.html

In Action



Circular Layout



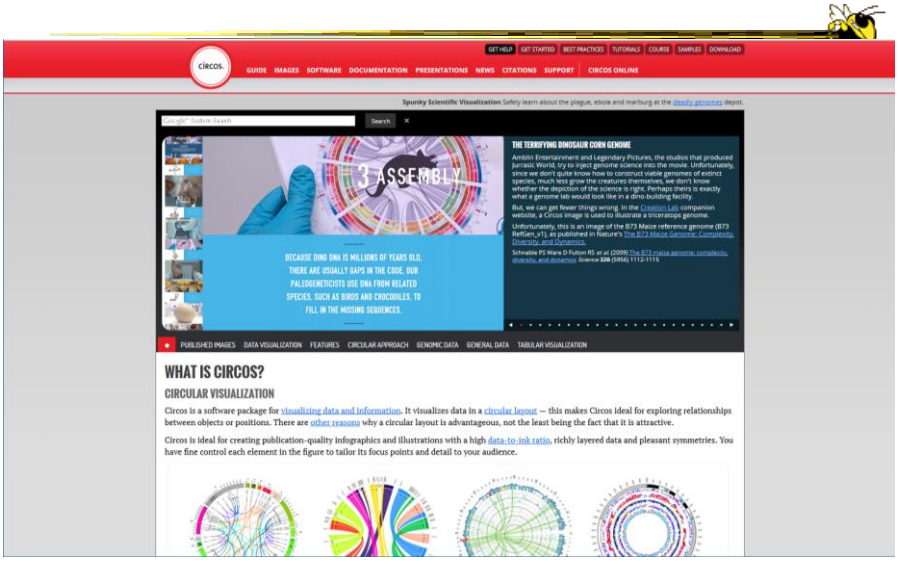
Ultra-simple
May not look so great

Space vertices out around circle
Draw lines (edges) to connect vertices

Uses curved lines and becomes "chord diagrams"

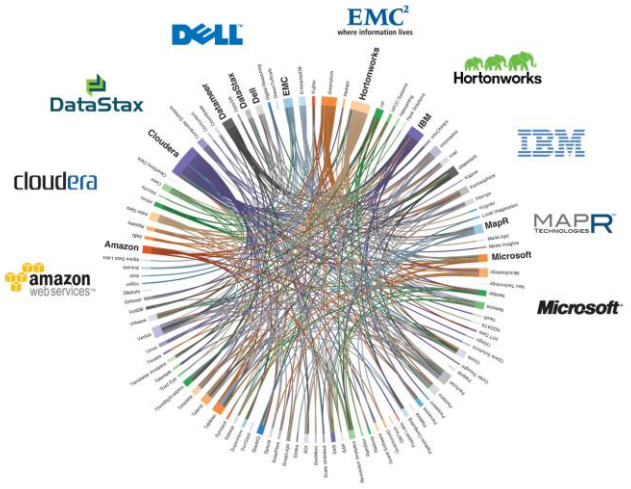
Circos

http://circos.ca/

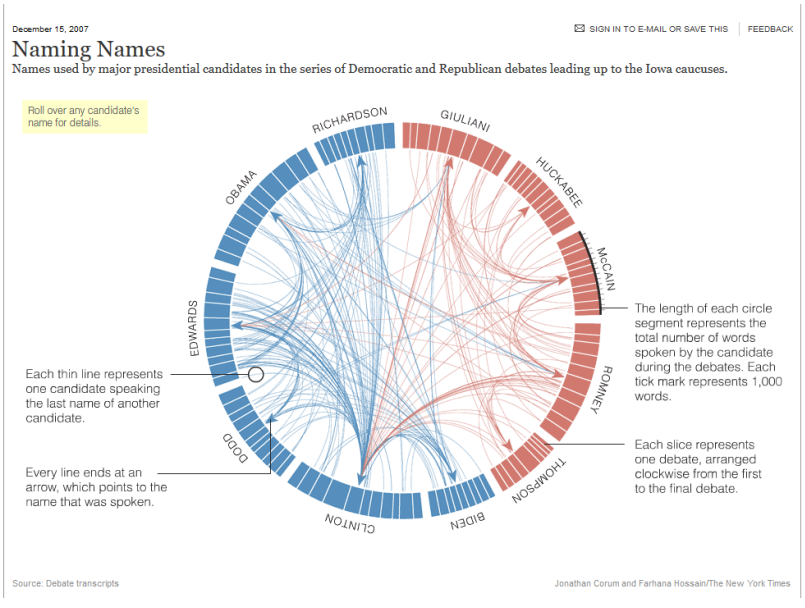


https://gigaom.com/2012/06/29/whos-connected-to-whom-in-hadoop-world-infographic/

Chord Diagram

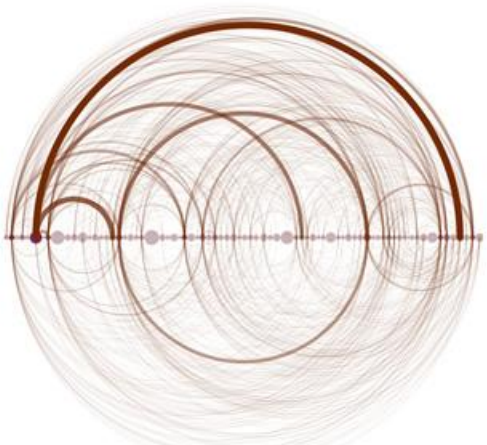


http://www.nytimes.com/interactive/2007/12/15/us/politics/DEBATE.html?_r=0

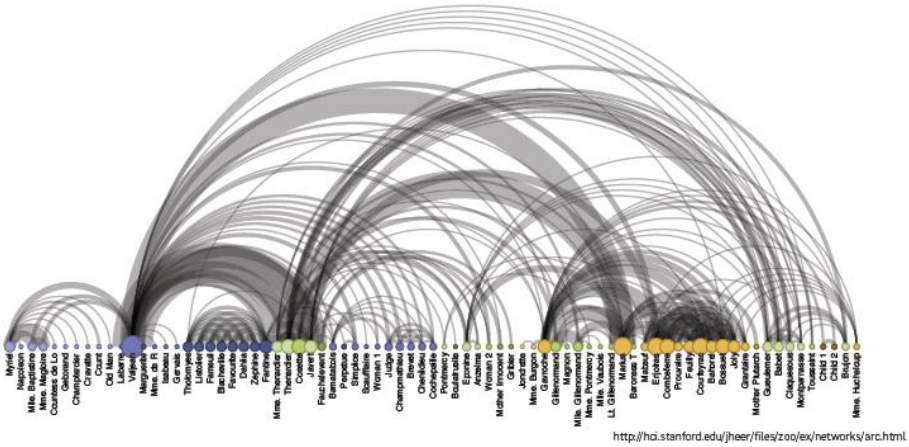


Arc Diagram Layout

Wattenberg
InfoVis '02



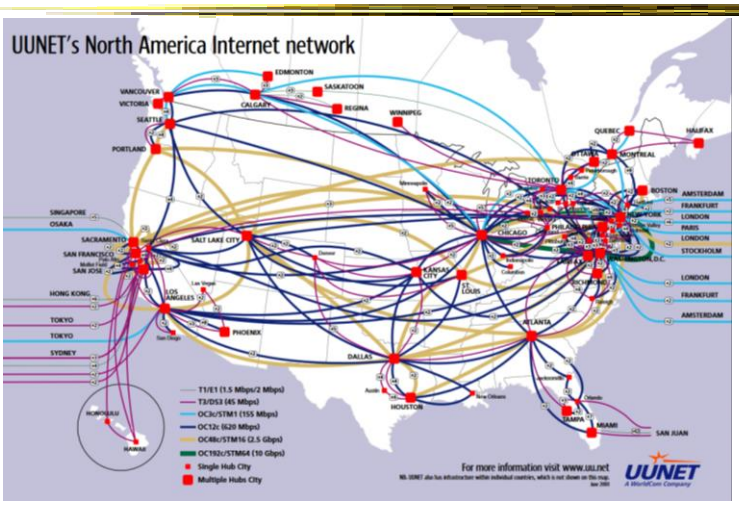
http://www.visualcomplexity.com/vc/index.cfm?method=Arc%20Diagrams



http://hci.stanford.edu/jheer/files/zoo/ex/networks/arc.html

http://www.nthelp.com/images/uunet.pdf

Geo/Map Approaches



Maps can easily become networks

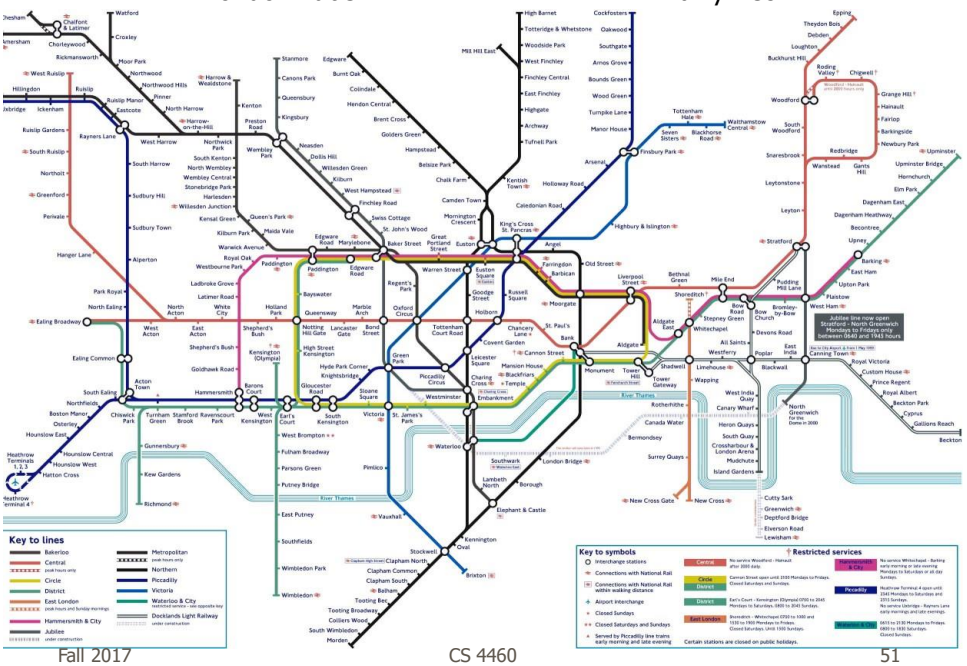
http://www.citylab.com/tech/2014/02/mapping-where-people-run/8313/

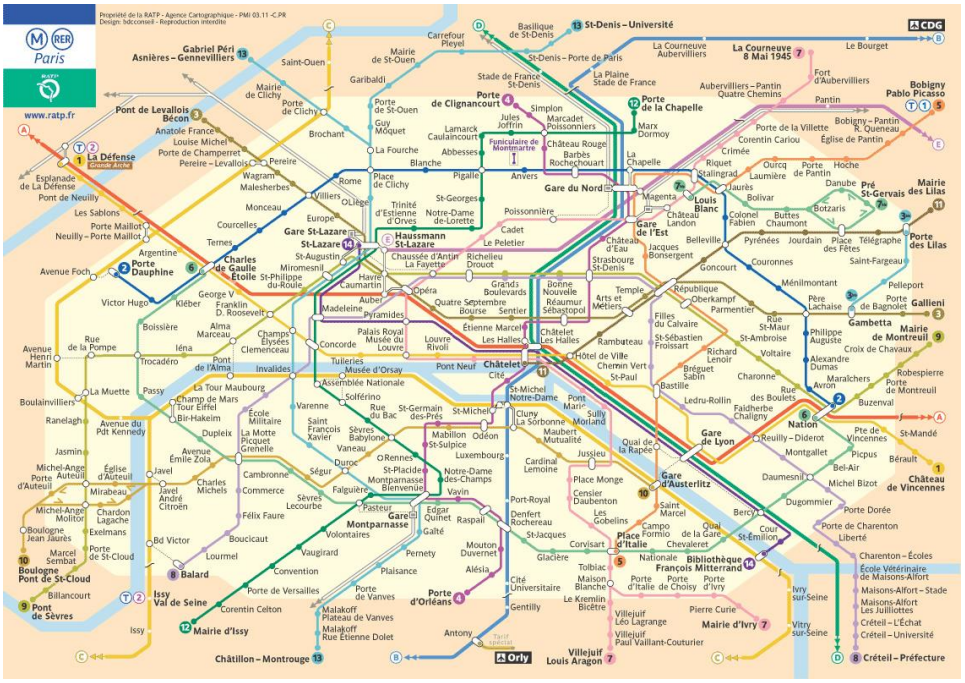
Where People Run

Atlanta

London Tube

Harry Beck





Atlanta
MARTA

WOO-WOO

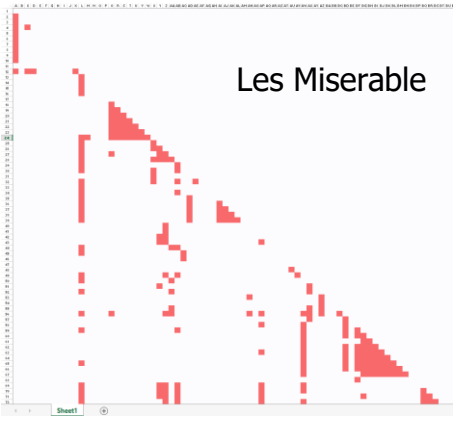
3 Subway Diagrams

- Geographic landmarks largely suppressed on maps, except water (rivers in London & Paris) and asphalt (highways in Atlanta)
  - Rather fitting, no?
- These are more *graphs* than maps!

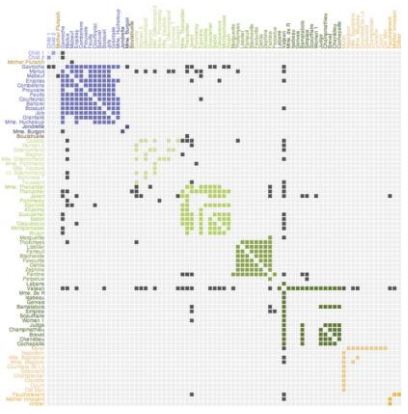
- Subway-style diagrams have become their own genre of network layouts

Matrix Representations

- Forget the node-link approach



http://www.themacroscope.org



https://homes.cs.washington.edu/~jheer/files/zoo/

Drawing Graphs Better

- Can we do clever "tricks" to make dense graphs more readable?

Hierarchical Edge Bundles

- Bundle edges that go from/to similar nodes together
  - Like wires in a house
- Uses B-spline curves for edges
- Reduces the clutter from many edges

Holten
TVCG (InfoVis) '06

Example

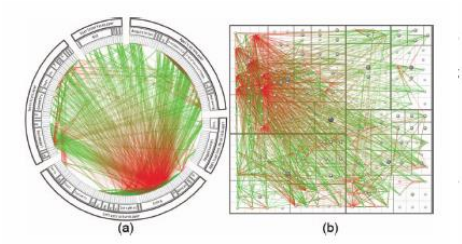


Fig. 11. A software system and its associated call graph (caller = green, callee = red). (a) and (b) show the system without bundling using a radial and a squarified treemap layout (node labels disabled), respectively. (a) and (b) mainly show hot spots; the actual connectivity information is more difficult to discern due to visual clutter.

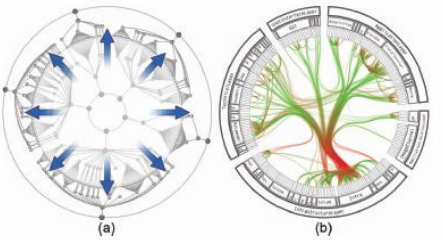


Fig. 12. Radial layout construction. (a) A radial tree layout is used for the inner circle and subsequently mirrored to the outside; (b) the inner layout is hidden and its structure is used to guide the adjacency edges. An icicle plot based on the mirrored layout is used to show the hierarchy.

Example

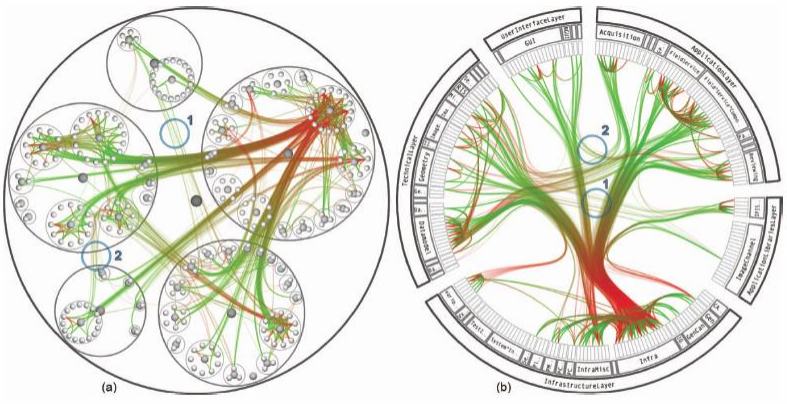
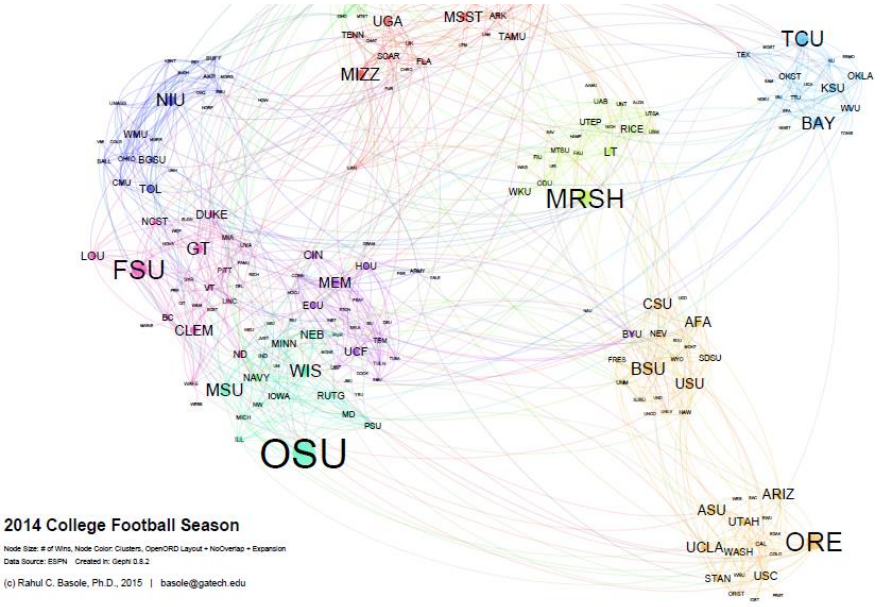


Fig. 13. A software system and its associated call graph (caller = green, callee = red). (a) and (b) show the system with bundling strength $\beta = 0.85$ using a balloon layout (node labels disabled) and a radial layout, respectively. Bundling reduces visual clutter, making it easier to perceive the actual connections than when compared to the non-bundled versions (figures 2a and 11a). Bundled visualizations also show relations between sparsely connected systems more clearly (encircled regions); these are almost completely obscured in the non-bundled versions. The encircled regions highlight identical parts of the system for (a), (b), and figure 15.

Opinion

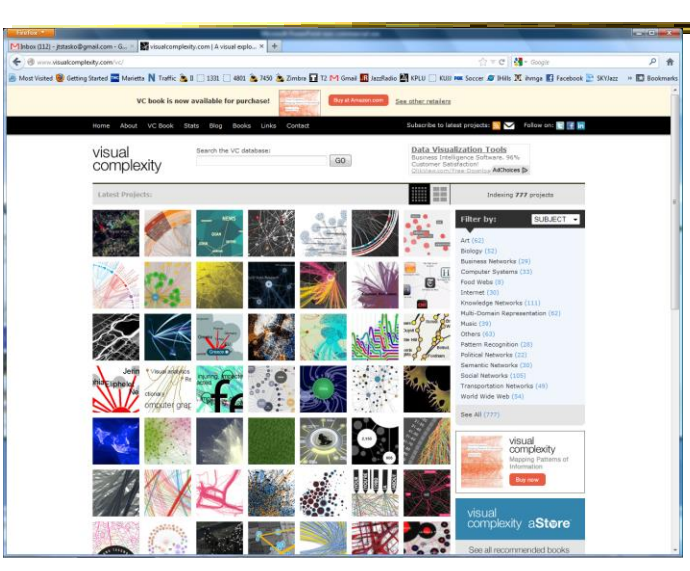
- Many graph drawings/visualizations (particularly the hairballs) provide little insight about the underlying data
  - Many are just "show offs" to make an accompanying visualization

(picking on a friend)



http://www.visualcomplexity.com

Many Examples

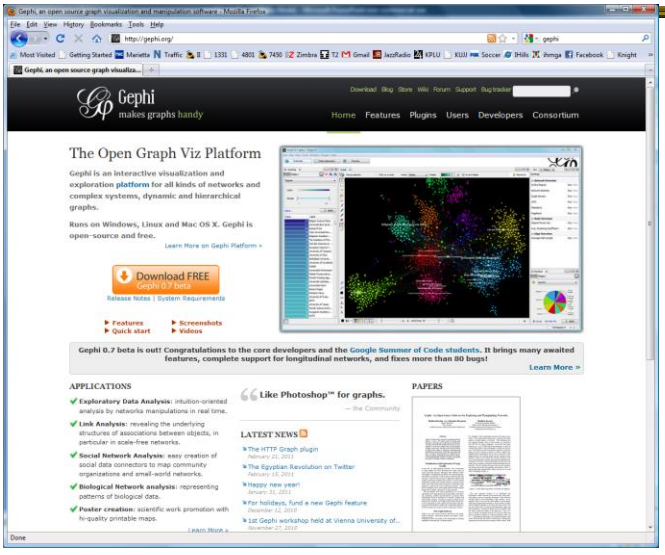


But Is It InfoVis?

- I generally don't consider a pure graph layout (drawing) algorithm to be InfoVis
  - Nothing wrong with that, just an issue of focus
- For InfoVis, I like to see some kind of interaction or a system or an application…
  - Still, understanding the layout algorithms is very important for infovis
  - Let's look at a few…

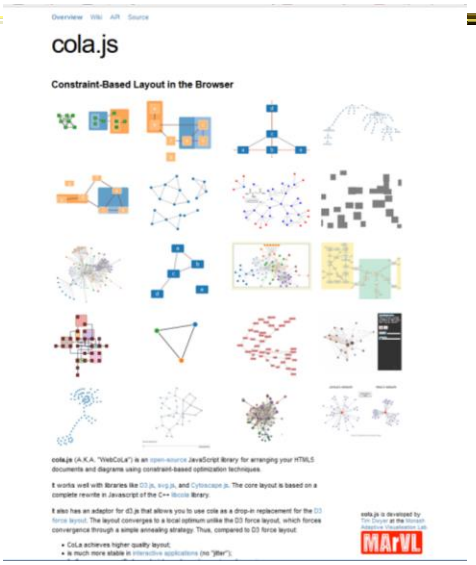
http://gephi.org

Gephi



http://marvl.infotech.monash.edu/webcola/

cola.js



Learning Objectives

- Define network concepts
  - vertex, edge, cycle, degree, direction
- Describe different node-link design choices
  - color, width, position, shape, size, label, form
- Enumerate primary aesthetic considerations for layouts
  - edge crossings, clusters, symmetry, edge lengths
- List example tasks for network data
- Explain "ball of string/hairball" problem
- List common layout approaches and describe characteristics of each
  - hierarchical, force-directed, circular, geo, matrix
- Define "edge bundling"

Upcoming

- Graphs and Networks 2
  - Prep: Try out immersion (website)
- Lab 8: D3 Interactivity 2