

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*

Fall 2017

CS 4460

3

What is a Graph?



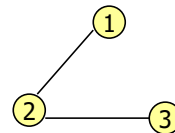
- Vertices (nodes) connected by
- Edges (links)

| | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0 | 1 | 0 |
| 2 | 1 | 0 | 1 |
| 3 | 0 | 1 | 0 |

Adjacency matrix

Adjacency list

1: 2
2: 1, 3
3: 2



Drawing

Fall 2017

CS 4460

4

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)

Fall 2017

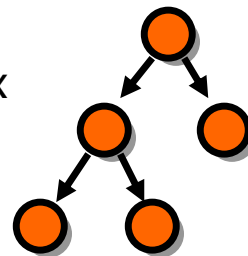
CS 4460

5

Trees are Different



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



Saw last week

Fall 2017

CS 4460

6

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

Fall 2017

CS 4460

9

Navigation/Interaction Challenge



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

Fall 2017

CS 4460

10

Layout Examples



- Homework assignment
- Let's judge!

Results



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

Talk Thursday

2-3 pm
TSRB Auditorium




View this email in your browser

Georgia Tech | Interactive Computing

Visualization Distinguished Lecture

Georgia Tech | College of Computing



with **Steven Drucker** | Microsoft Research
Principal Researcher in the VIBE group

**Data Presentations: Or why
'Letting the Data Speak for Itself
just isn't that easy'**

Thursday, Nov. 9, 2pm / TSRB Auditorium

Fall 2017

CS 4460

13

Quiz 4



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

Name: _____

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

Fall 2017

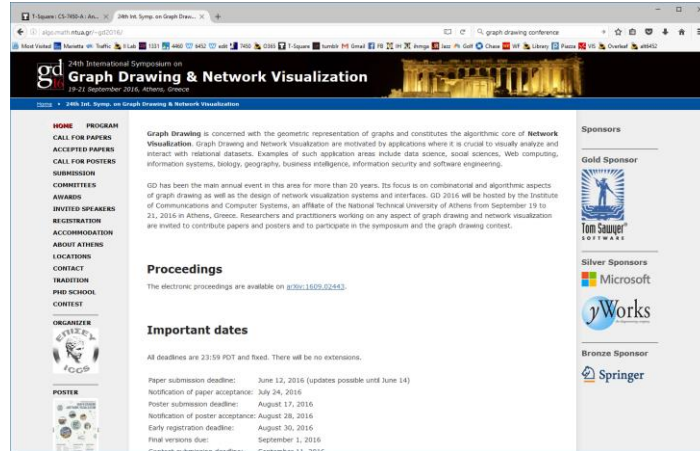
CS 4460

14

Graph Drawing



Entire research community's focus



Fall 2017

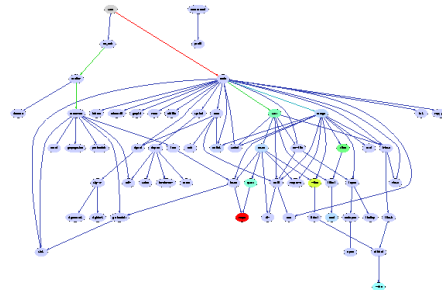
CS 4460

15

Vertex Issues



- Shape
- Color
- Size
- Location
- Label



Fall 2017

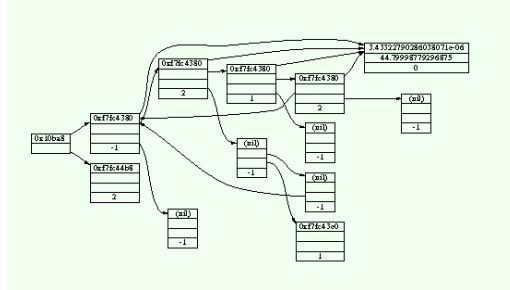
CS 4460

16

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

Fall 2017

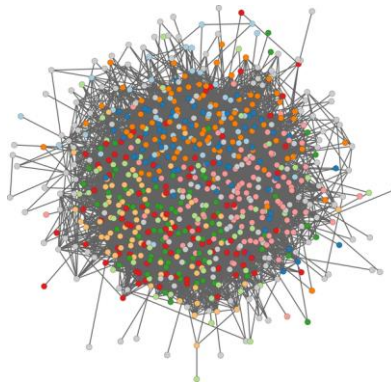
CS 4460

21

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>

Fall 2017

CS 4460

22

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)

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

Fall 2017

CS 4460

25

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

Fall 2017

CS 4460

26

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

Fall 2017

CS 4460

27

Graph Drawing Uses



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

Fall 2017

CS 4460

28

Human Diseases

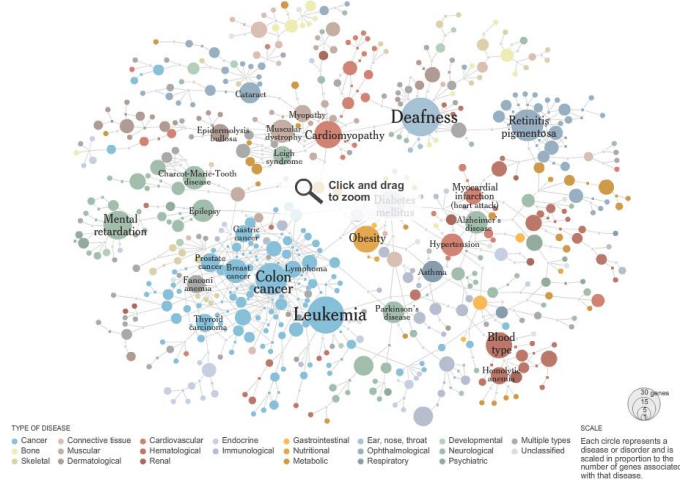
May 5, 2008

SIGN IN TO E-MAIL FEEDBACK



Mapping the Human 'Diseaseome'

Researchers created a map linking different diseases, represented by circles, to the genes they have in common, represented by squares. Related Article: [Redefining Disease, Genes and All](#)



Note the two extra variables per vertex

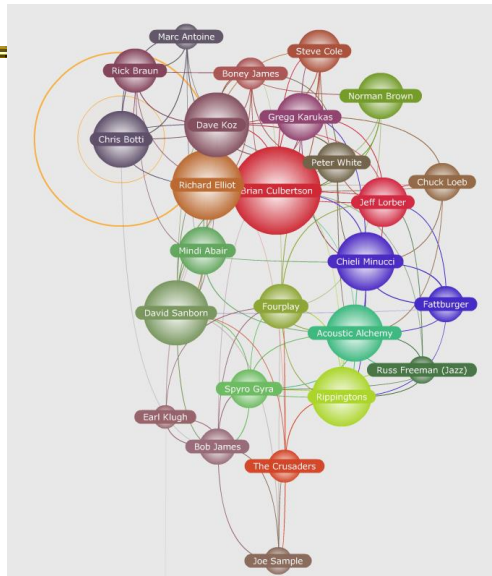
Fall 2017

CS 4460

29

Music Artists

older

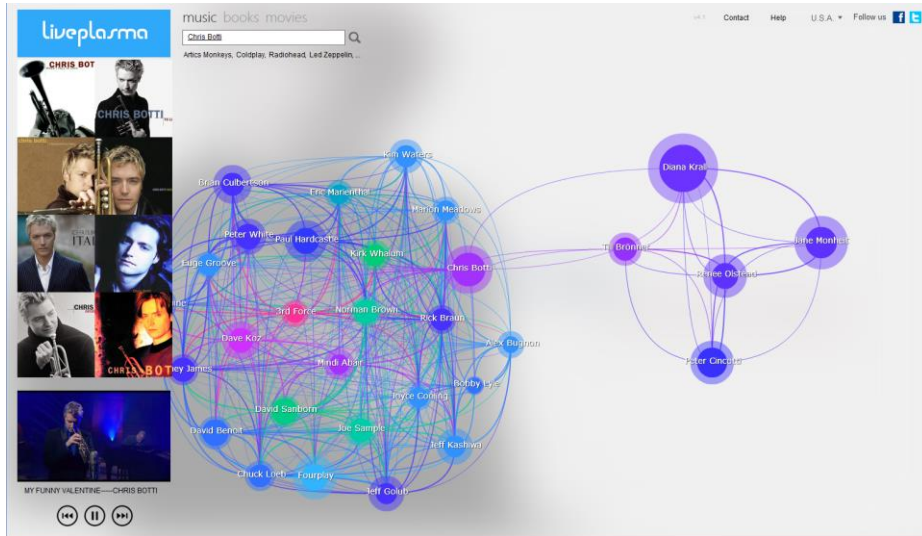


<http://www.liveplasma.com/>

Fall 2017

CS 4460

30



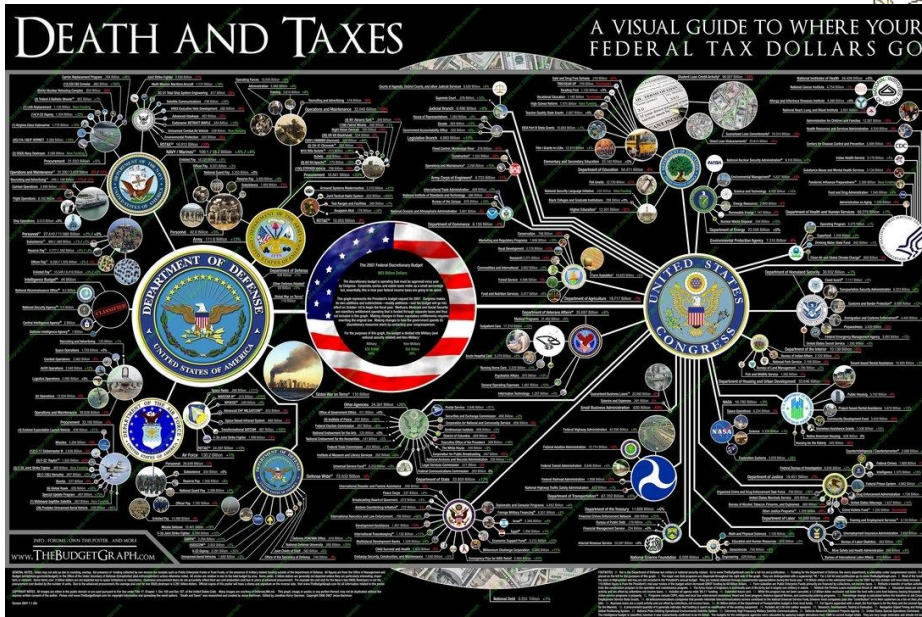
Fall 2017

CS 4460

31

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

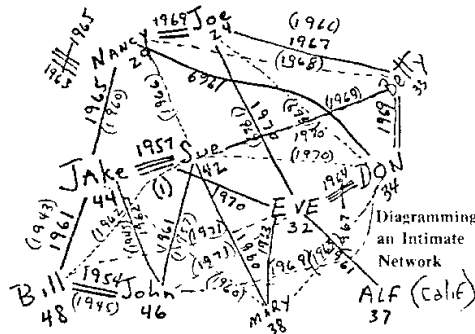
US Budget



Social Network Visualization

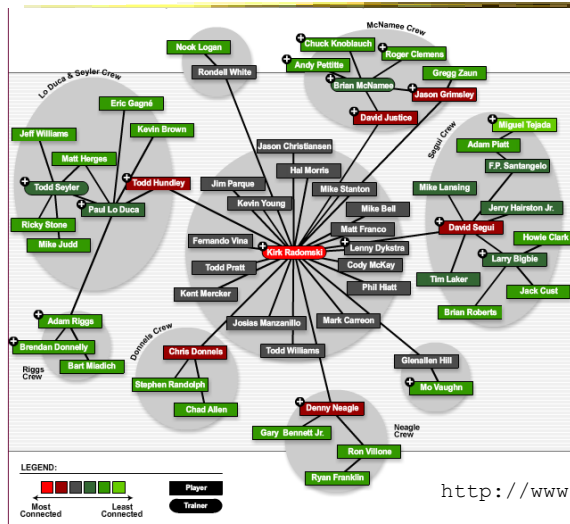


- Social Network Analysis
 - <http://www.insna.org>



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
 - ...

Fall 2017

CS 4460

35

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

Fall 2017

CS 4460

36

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

Fall 2017

CS 4460

37

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

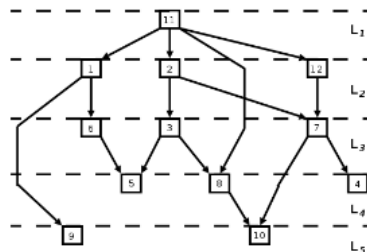


Figure: A graph showing a layered layout, created with the Sugiyama heuristic, with the layers shown. The bends in the edges correspond to dummy nodes.

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

Fall 2017

CS 4460

38

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

Fall 2017

CS 4460

39

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

Fall 2017

CS 4460

40

Force-directed Example

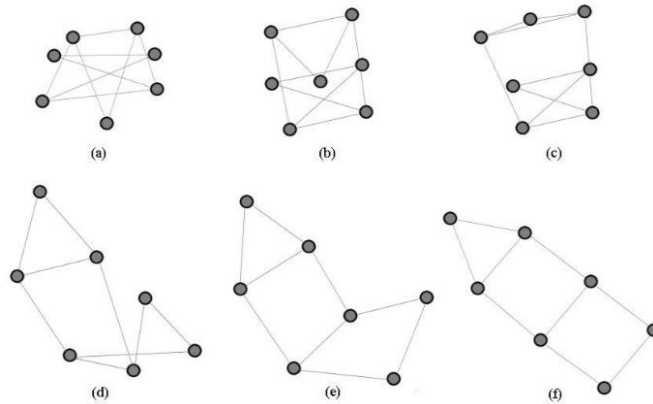


Figure 2: A graph drawing through a number of iterations of a force directed algorithm.

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

Fall 2017

CS 4460

41

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

In Action



Protovis A GRAPHICAL TOOLKIT FOR VISUALIZATION

Overview Examples Documentation Download

Index < Previous / Next >

Force-Directed Layouts

[View full screen](#)

An intuitive approach to network layout is to model the graph as a physical system: nodes are charged particles that repel each other, and links are damped springs that pull related nodes together. A physical simulation of these forces then determines node positions; approximation techniques that avoid computing all pairwise forces enable the layout of large numbers of nodes. In addition, interactivity allows the user to direct the layout and jiggle nodes to disambiguate links. Such a *force-directed layout* is a good starting point for understanding the structure of a general undirected graph.

This network represents character co-occurrence in the chapters of Victor Hugo's classic novel, *Les Misérables*. Node colors depict cluster memberships computed by a community-detection algorithm. Source: Knuth, D. E. 1993. *The Stanford Graphbase: A Platform for*

Fall 2017

CS 4460

42

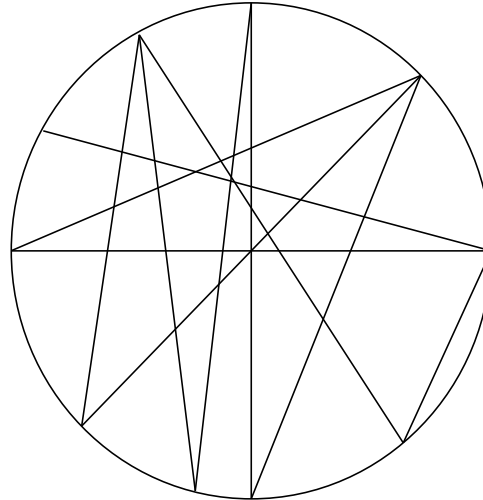
Circular Layout



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"



Fall 2017

CS 4460

43

Circos

<http://circos.ca/>



GET HELP | GET STARTED | BEST PRACTICES | TUTORIALS | COURSE | SAMPLES | DOWNLOAD

GUIDE | IMAGES | SOFTWARE | DOCUMENTATION | PRESENTATIONS | NEWS | CITATIONS | SUPPORT | CIRCOS ONLINE

TEACHING SCIENTIFIC VISUALIZATION SAFETY: LEARN ABOUT THE PLAGIARISM, ABUSE AND MISUSE AT THE [CIRCOS BLOG](#) REPORT

SEARCH

3 ASSEMBLY

BECAUSE DNA IS MILLIONS OF YEARS OLD, THERE ARE USUALLY GAPS IN THE CODE. OUR PALEOGENETICISTS USE DNA FROM RELATED SPECIES, SUCH AS BIRDS AND CROCODILES, TO FILL IN THE MISSING SEQUENCES.

THE TERRIFYING DINOSAUR COGNOME

Artists Entertainment and Legendary Pictures, the studios that produced Jurassic World, try to past genome science into the movie. Unfortunately, since we don't quite know how to construct viable genomes of extinct species, much less grow the creatures themselves, we don't know whether the depiction of the science is right. Perhaps there is exactly what genome we should look for in the film-making facility.

But, we can get fewer things wrong. In the [Circos](#) Lab companion website, a Circos image is used to illustrate a dinosaur genome. Unfortunately, this is an image of the B73 Mouse reference genome (B73 RefGen_v1), as published in Nature's [The B73 Mouse Genome: Context, Contents, and Structure](#).

Schwartz PS, Ware D, Fulton RS et al (2008) The B73 mouse genome: context, contents, and structure. [Genome](#) 109: 1912-1918.

PUBLISHED PAGES | DATA VISUALIZATION | FEATURES | CIRCULAR APPROACH | GENOMIC DATA | GENERAL DATA | TABLEAU VISUALIZATION

WHAT IS CIRCOS?

CIRCULAR VISUALIZATION

Circos is a software package for visualizing data and information. It visualizes data in a circular layout — this makes Circos ideal for exploring relationships between objects or positions. There are [other reasons](#) why a circular layout is advantageous, not the least being the fact that it is attractive.

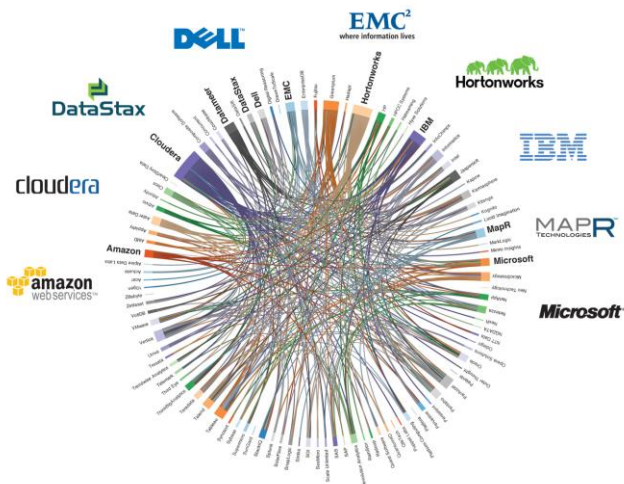
Circos is ideal for creating publication-quality infographics and illustrations with a high [data-to-ink ratio](#), richly layered data and pleasant symmetries. You have fine control each element in the figure to tailor its focus points and detail to your audience.

Fall 2017

CS 4460

44

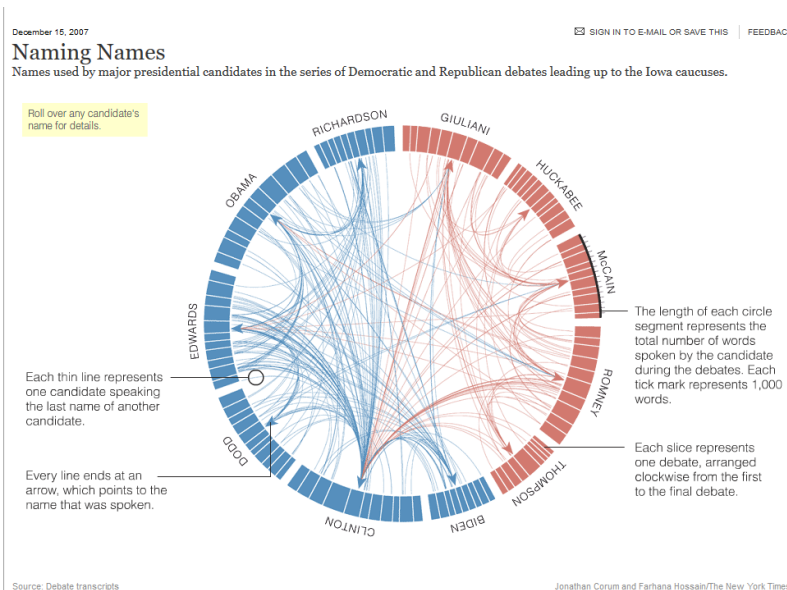
Chord Diagram



Fall 2017

CS 4460

45



Fall 2017

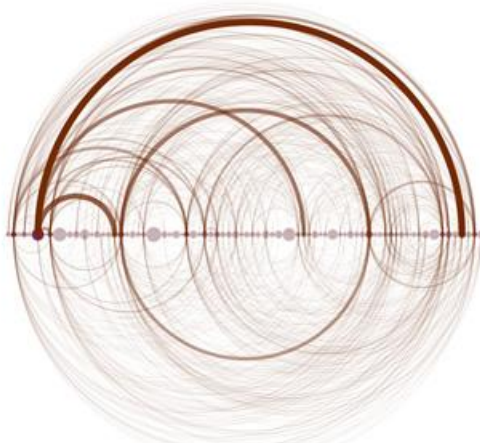
CS 4460

46

Arc Diagram Layout



Wattenberg
InfoVis '02

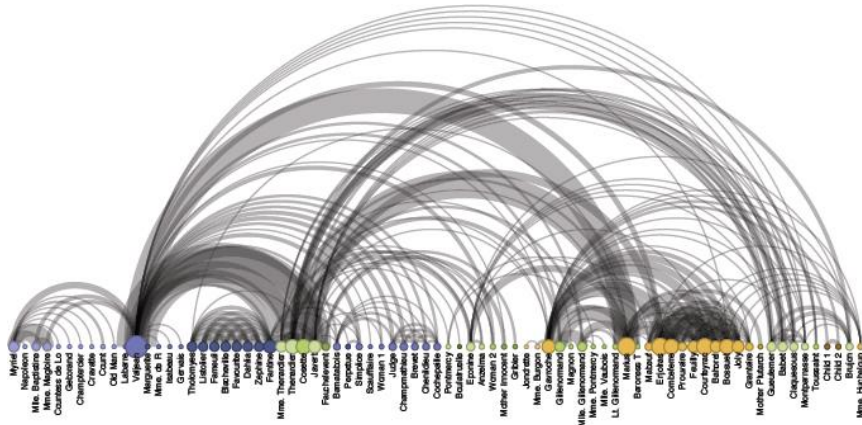


<http://www.visualcomplexity.com/vc/index.cfm?method=Arc%20Diagrams>

Fall 2017

CS 4460

47



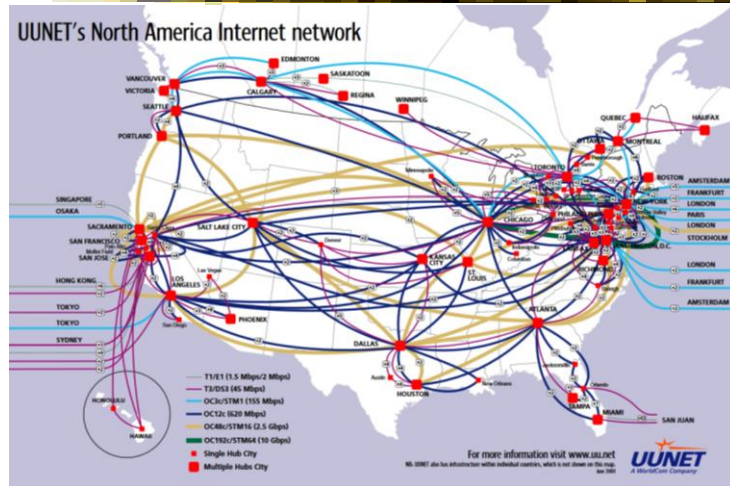
<http://ha.stanford.edu/jheer/files/zoo/ex/networks/arc.html>

Fall 2017

CS 4460

48

Geo/Map Approaches



Maps can easily become networks

Fall 2017

CS 4460

49

Where People Run



CITYLAB | [News](#) | [Commute](#) | [Housing](#) | [Weather](#)

Mapping Where People Run

These graphics chart the most popular routes in 22 cities.

JENNY XIE | @carsonrd | Feb 5, 2014 | 33 Comments

Share on Facebook | [Twitter](#) | [LinkedIn](#) | [Google+](#) | [Print](#)

Love CityLab? Make sure you're signed up for our free e-mail newsletter:

Inspired by a 2011 [project](#) that mapped popular running routes in a few European cities, Nathan Yau at [FlourishData](#) has done the same for 22 major cities, including 18 in the U.S.

To make these maps, Lau simply grabbed public data from the exercise-tracking app [Strava](#). While these visualizations are not representative of all runners in a city, they do offer useful information on urban spaces. For one, we see that people really do love running near water and in parks.

Here's a selection of Yau's work. See the rest [here](#).

Boston

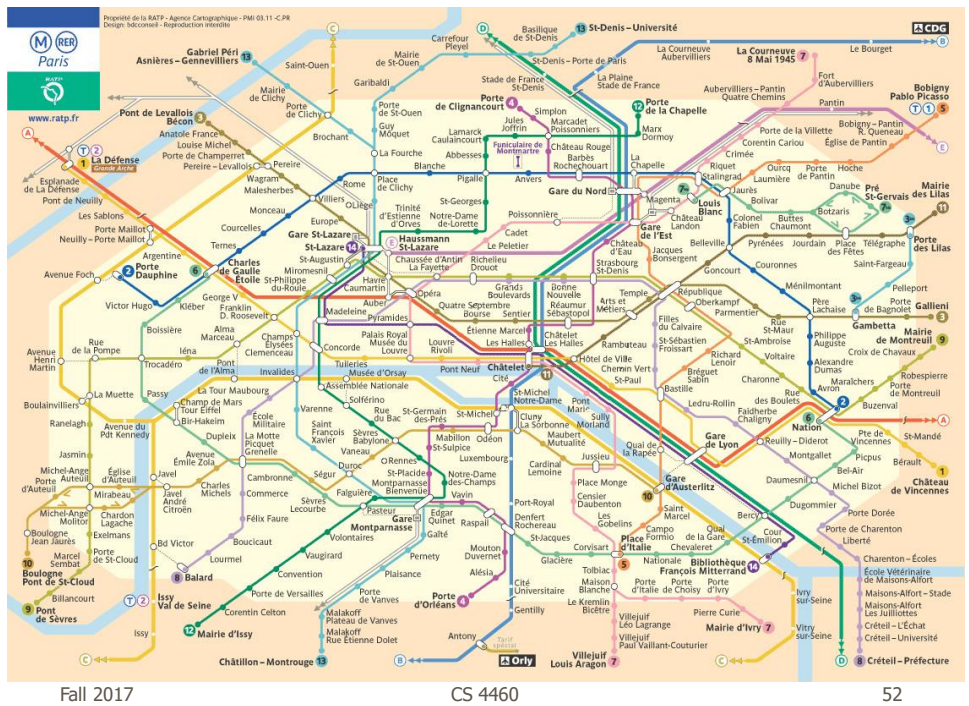
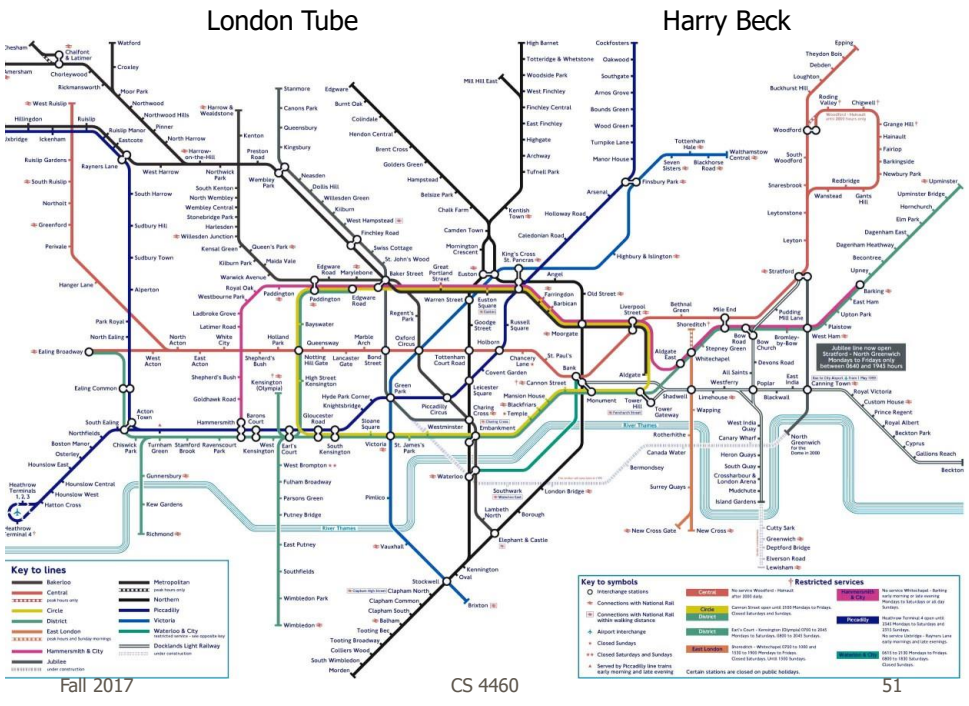
Atlanta

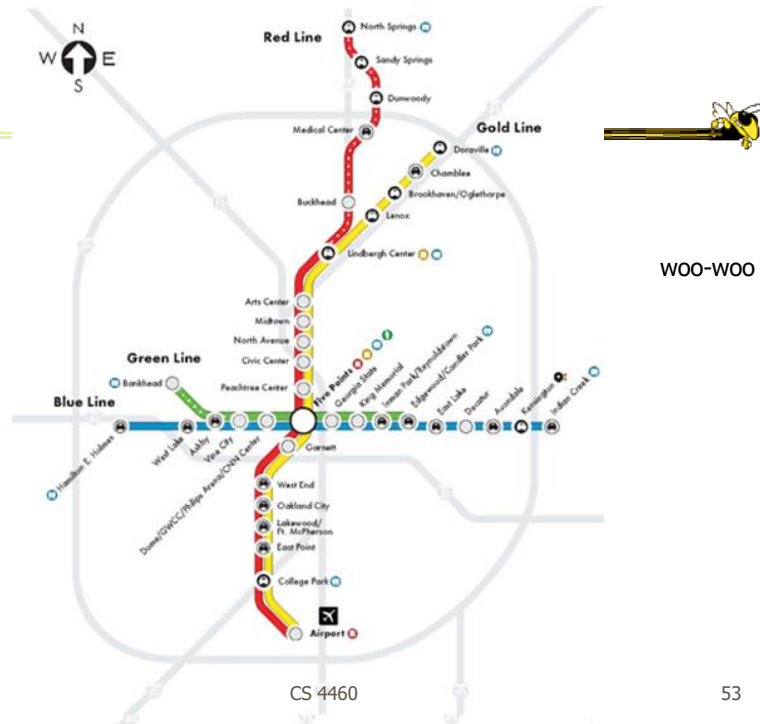


Fall 2017

CS 4460

50





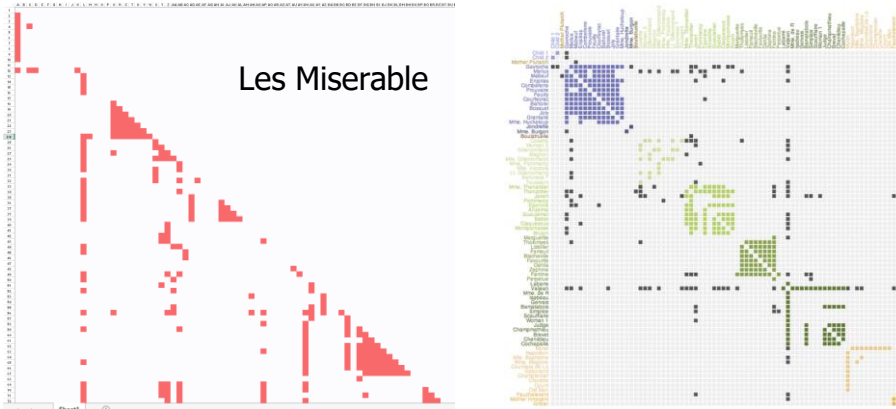
3 Subway Diagrams

- 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/>

Fall 2017

CS 4460

55

Drawing Graphs Better



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

Fall 2017

CS 4460

56

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

Fall 2017

CS 4460

57

Example

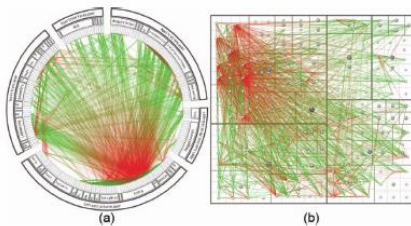


Fig. 11. A software system and its associated call graph (caller = green, callee = red). (a) and (b) show the system without bundling using a radial and a squarified treemap layout (node labels disabled), respectively. (a) and (b) mainly show hot spots; the actual connectivity information is more difficult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

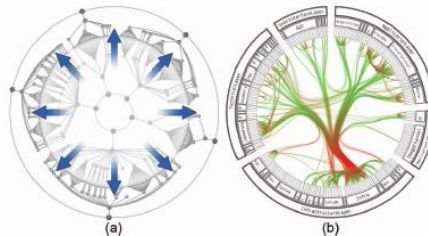


Fig. 12. Radial layout construction. (a) A radial tree layout is used for the inner circle and subsequently mirrored to the outside; (b) the inner layout is hidden and its structure is used to guide the adjacency edges. An icicle plot based on the mirrored layout is used to show the hierarchy.

Fall 2017

CS 4460

58

Example

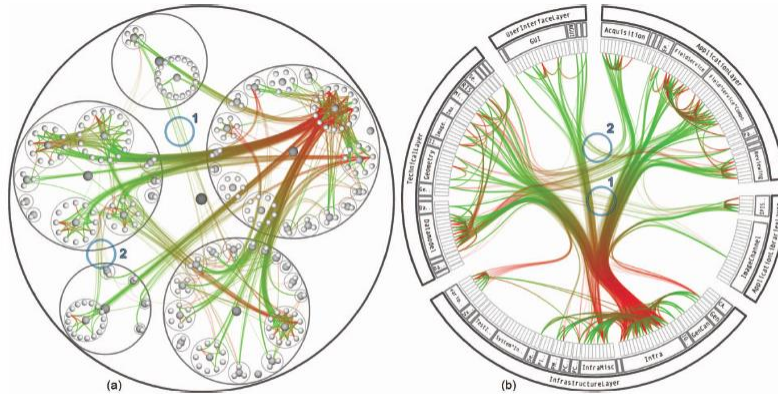


Fig. 13. A software system and its associated call graph (caller = green, callee = red). (a) and (b) show the system with bundling strength $\beta = 0.85$ using a balloon layout (node labels disabled) and a radial layout, respectively. Bundling reduces visual clutter, making it easier to perceive the 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.

Fall 2017

CS 4460

59

Opinion



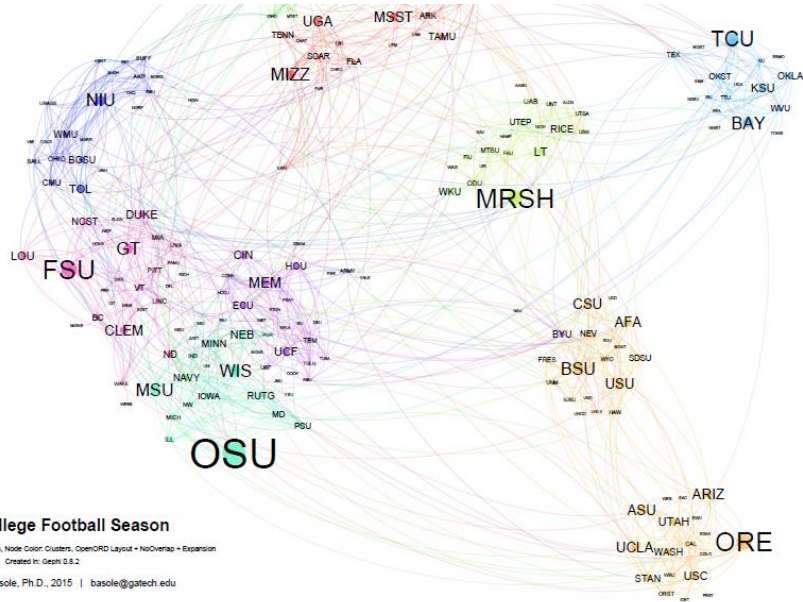
- Many graph drawings/visualizations (particularly the hairballs) provide little insight about the underlying data
 - Many are just "show offs" to make an accompanying visualization

Fall 2017

CS 4460

60

(picking on a friend)



2014 College Football Season

Node Size: # of Wins, Node Color: Clusters, Open/Closed Layout + No/Over + Expansion
Data Source: ESPN Created In: Gephi 0.8.2

(c) Rahul C. Basole, Ph.D., 2015 | basole@gatech.edu

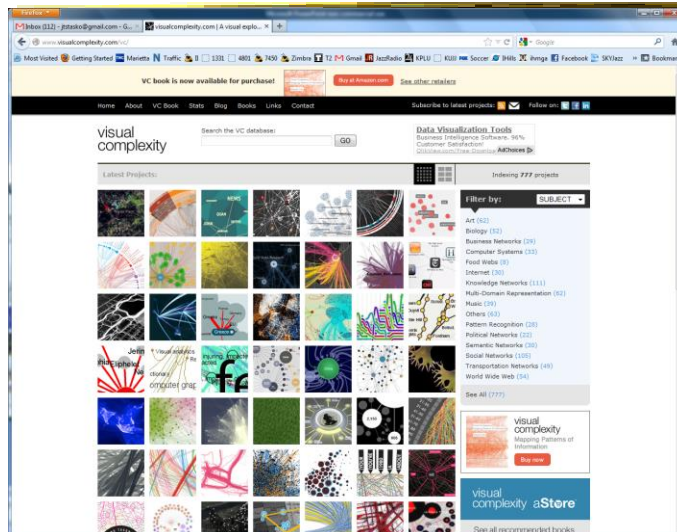
Fall 2017

CS 4460

61

<http://www.visualcomplexity.com>

Many Examples



Fall 2017

CS 4460

62

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...

Fall 2017

CS 4460

63

<http://gephi.org>

Gephi



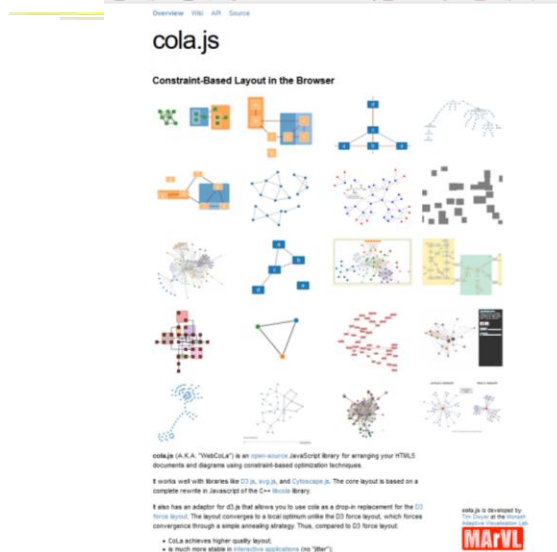
The screenshot shows the Gephi website homepage. At the top, there's a navigation bar with links for Home, Features, Plugins, Users, Developers, and Consortium. The main heading is "The Open Graph Viz Platform". Below this, there's a description of Gephi as an interactive visualization and exploration platform for networks. A "Download FREE" button is prominent. The page also includes sections for "APPLICATIONS" (Exploratory Data Analysis, Link Analysis, Social Network Analysis, Biological Network analysis, Poster creation), "LATEST NEWS", and "PAPERS". A screenshot of the Gephi software interface is shown on the right side of the page.

Fall 2017

CS 4460

64

cola.js



The screenshot shows the cola.js website. At the top, it says "cola.js" and "Constraint-Based Layout in the Browser". Below this is a grid of 16 small images showing various network diagrams and layouts. At the bottom of the grid, there is text describing the library: "cola.js (A.K.A. 'YIMCOLA') is an open source JavaScript library for arranging your HTML5 documents and diagrams using constraint based optimization techniques." It also mentions that it works well with libraries like D3.js, Hq.js, and Cytoscape.js, and that it has an adaptor for d3.js. A MARVL logo is visible in the bottom right corner of the screenshot.

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