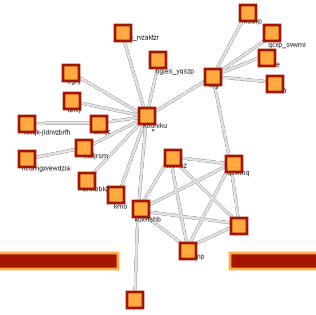


The Java Universal Network/Graph Framework (JUNG): A Brief Tour

Joshua O'Madadhain (UCI)

Danyel Fisher (MSR), Tom Nelson (RABA Technologies)
Scott White (UCI), Yan-Biao Boey (UCI)



What's it for?

phenomena pertaining to relationships/links

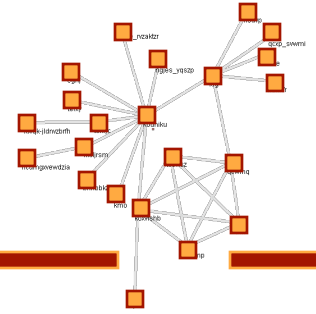
- collaborations, citations/hyperlinks, email, biological systems, friendships, telecom networks, ecological and biological systems, Markov processes, Bayes nets...

▪

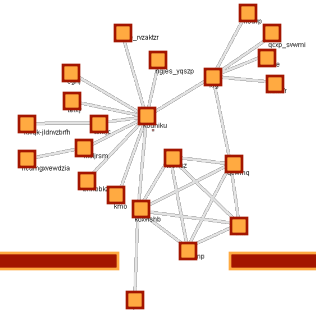
SUNBELT 2000

- 270 People & Structural Holes
- Database *to* UCINET *to* Pajek *to* MAGE

Programmatic Solution



- *Programmable*
Make it easy to do the same thing again
- *Extensible*
If you need a new routine, you can put it in yourself
- *Library*
Can be used in a server, a client, or as part of a larger application
- *Open Source*
Take it apart and put it together some other way
Free!

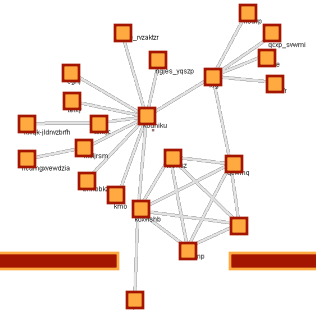


What is JUNG?

Framework for the modeling, analysis, and visualization of graphs in Java

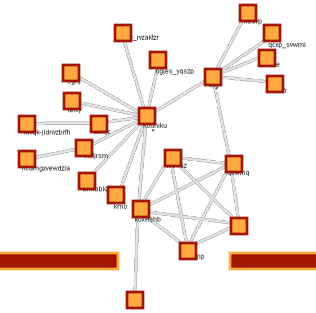
- supports most types of graphs/networks
- separate, flexible visualization framework
- rich attribute structure (metadata)
- network event handling
- predicates (subsets, constraints, filtering, ...)
- extensive (and growing!) library of algorithms

Dependencies



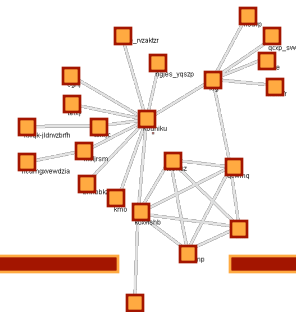
- **Java (1.4.2+)**
- **Commons-Collections API (3.1)**
<http://jakarta.apache.org/commons/collections/>
Predicate, set operations, wrappers, data types (BidiMap)
- **CERN Colt API (1.2.0)**
<http://dsd.lbl.gov/~hoschek/colt/>
matrix operations, statistics
- **Xerces (2.6.2)**
<http://xml.apache.org/xerces2-j/index.html>
XML parsing

Overview



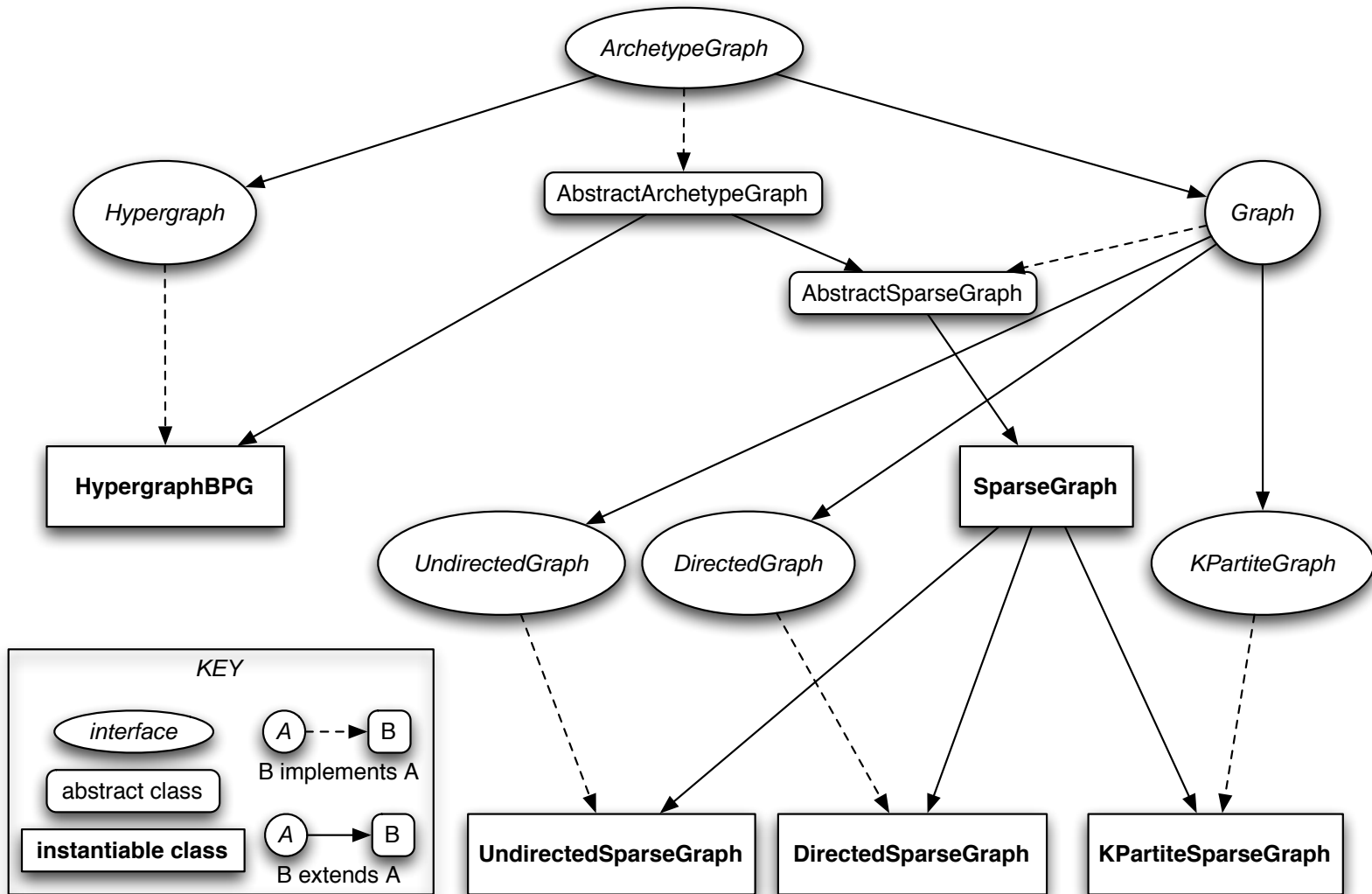
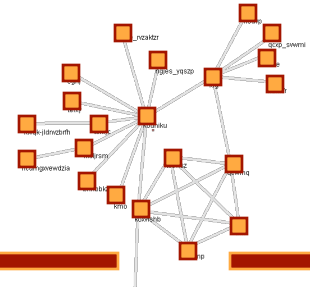
- Creating a graph
- Annotating with data
- Filters
- Graph drawing
- Algorithms
- Demo

Creating a graph

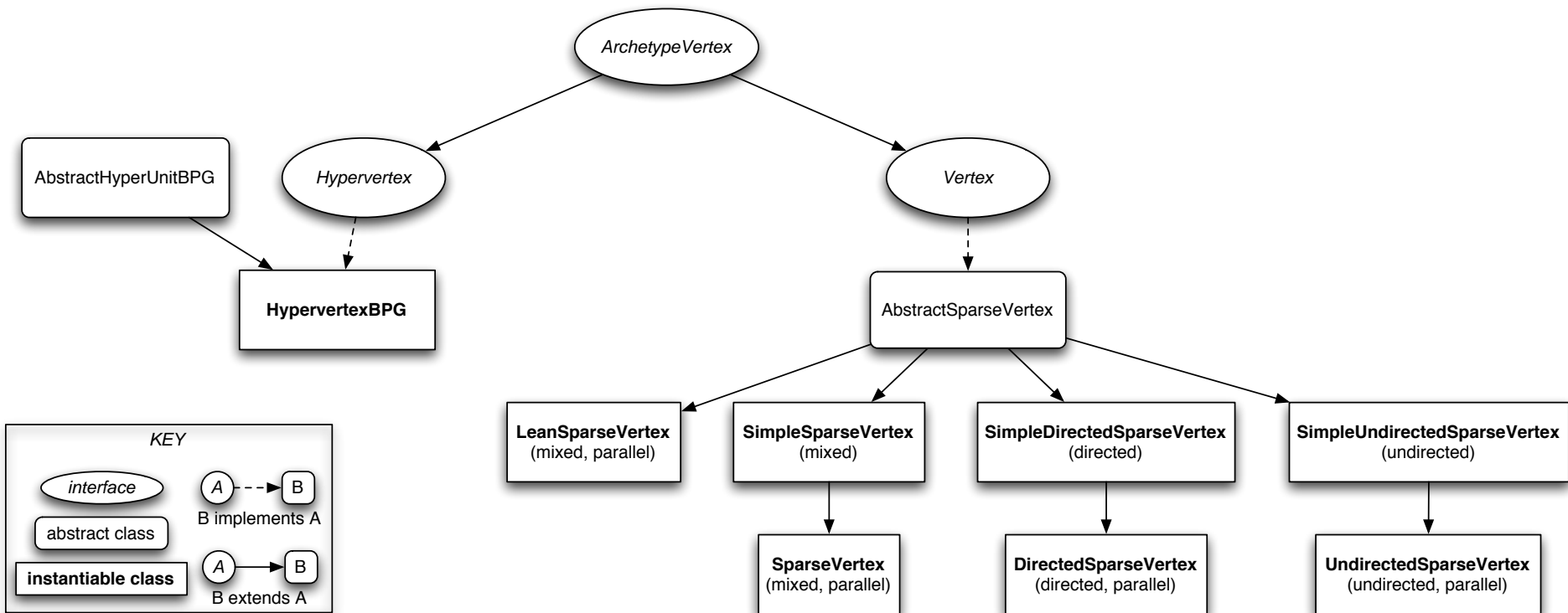
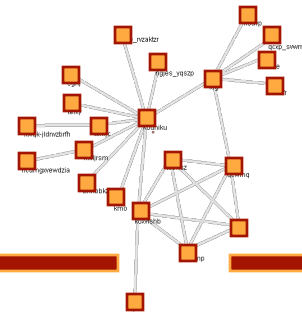


- Load from a file
 - Pajek
 - GraphML (XML)
- Build from scratch
 - Create vertices individually
 - Specify edges
- Generate from an algorithm/model
 - Small-world graphs
 - Power-law graphs

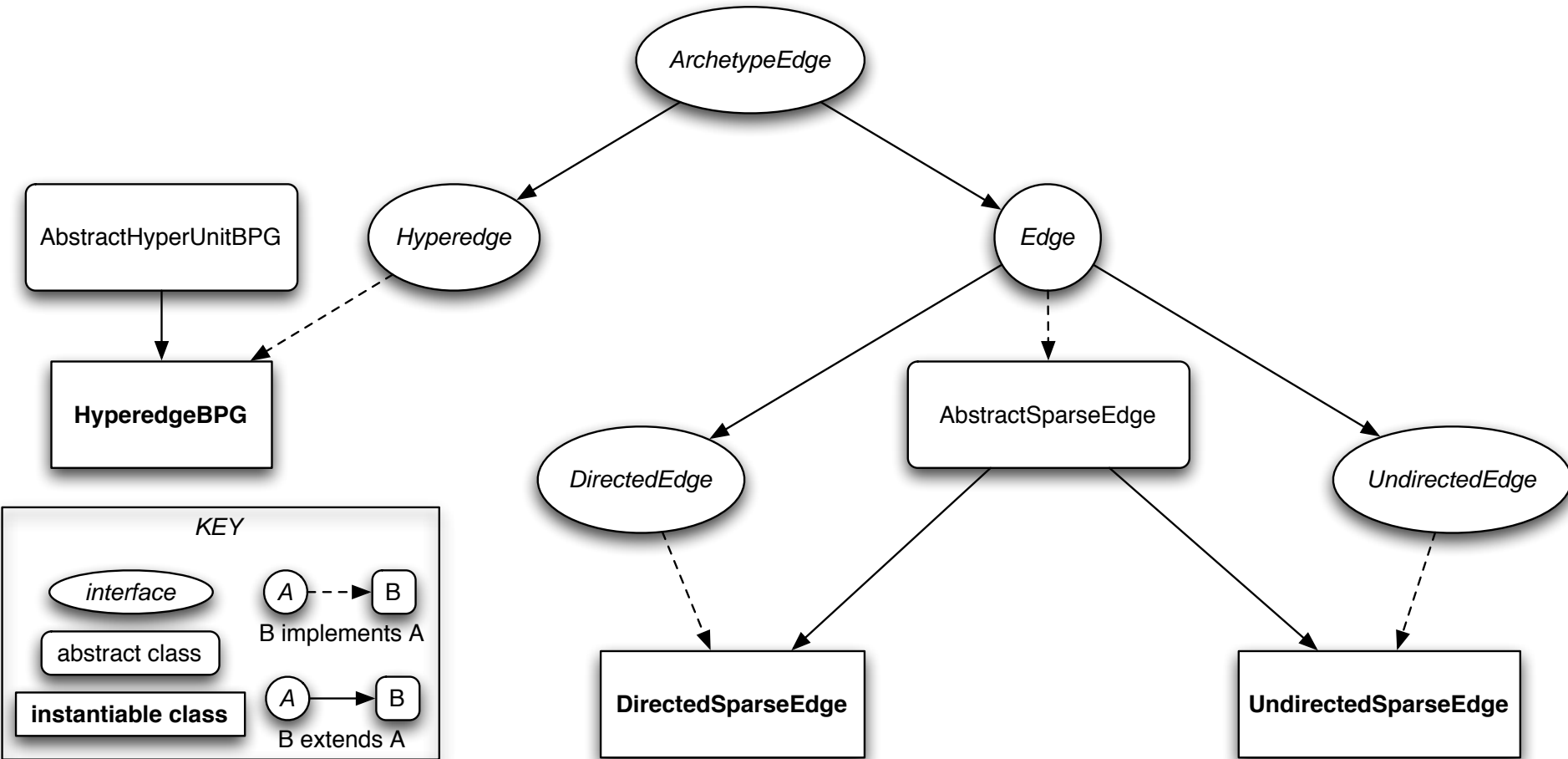
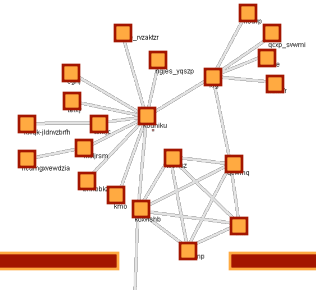
Graph Types

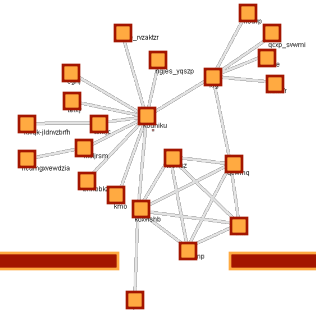


Vertex Types



Edge Types

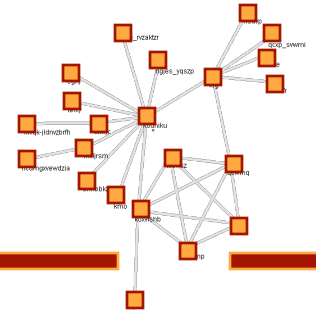




Adding user-defined data

- key-value pairs associated with graphs, edges, vertices
- easy to write algorithms that operate on relational data

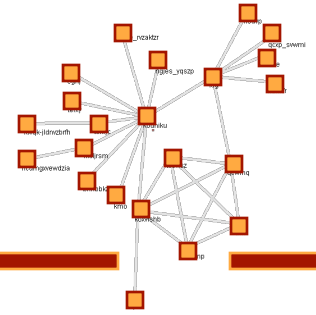
```
String name = vertex.getUserDatum(NAME);  
edge.setUserDatum(DATE, new Date());
```



Copying and Equality

- copies of vertices and edges are “equal to” the originals
 - `equals()`, `hashCode()`
 - subgraphs
 - allows experimentation with mutations to graph, without affecting original

```
v2 = v1.getEqualVertex(g2)
```

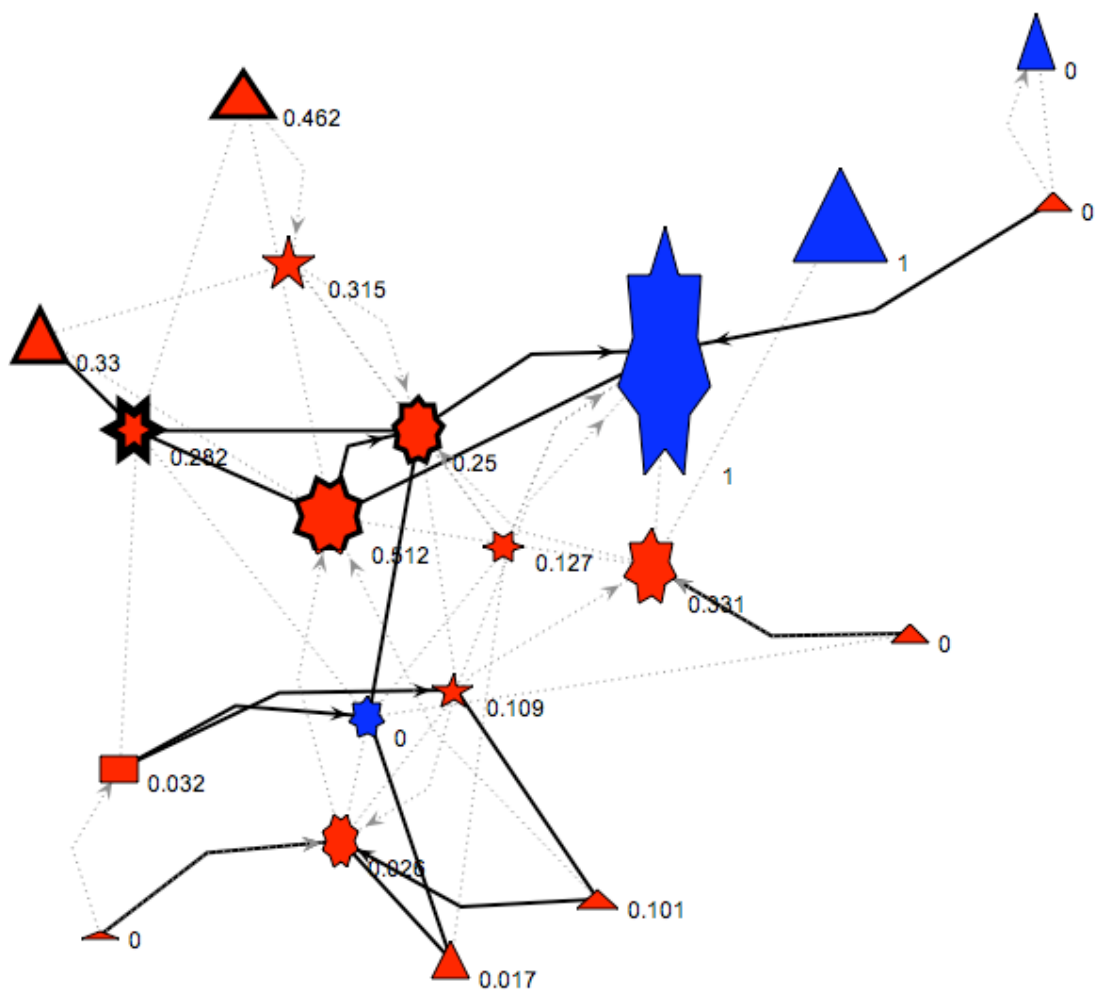



Visualizing graphs

- Basic layout algorithms:
 - Fruchterman-Rheingold
 - Kamada-Kawai
 - Eades
 - Self-Organizing Maps
- Plug and play architecture
 - Drop in your favorite renderer, layout
- Provided rendering engine is Swing-based
 - (but layouts are generic)

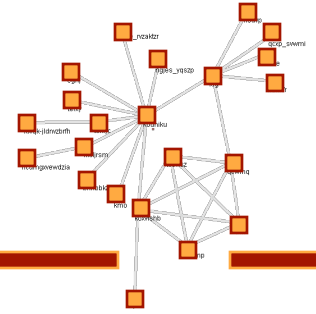


Pluggable Renderer



- vertex seed coloring
- vertex selection stroke highlighting
- show vertex ranks (voltages)
- vertex degree shapes
- vertex voltage size
- vertex degree ratio stretch
- bold text
- edge weight highlighting
- show edge weights
- show undirected edge arrows

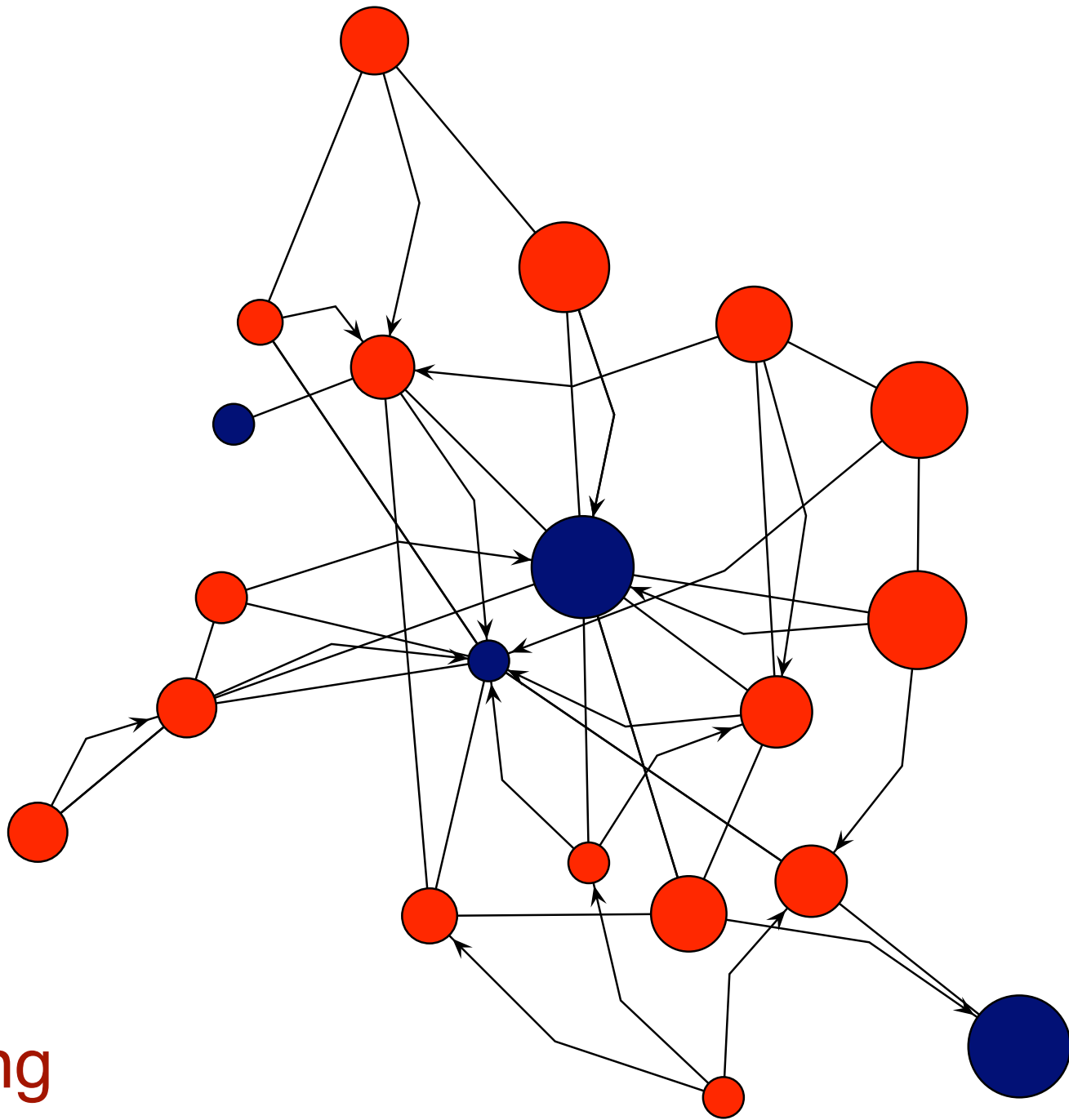
Customizing



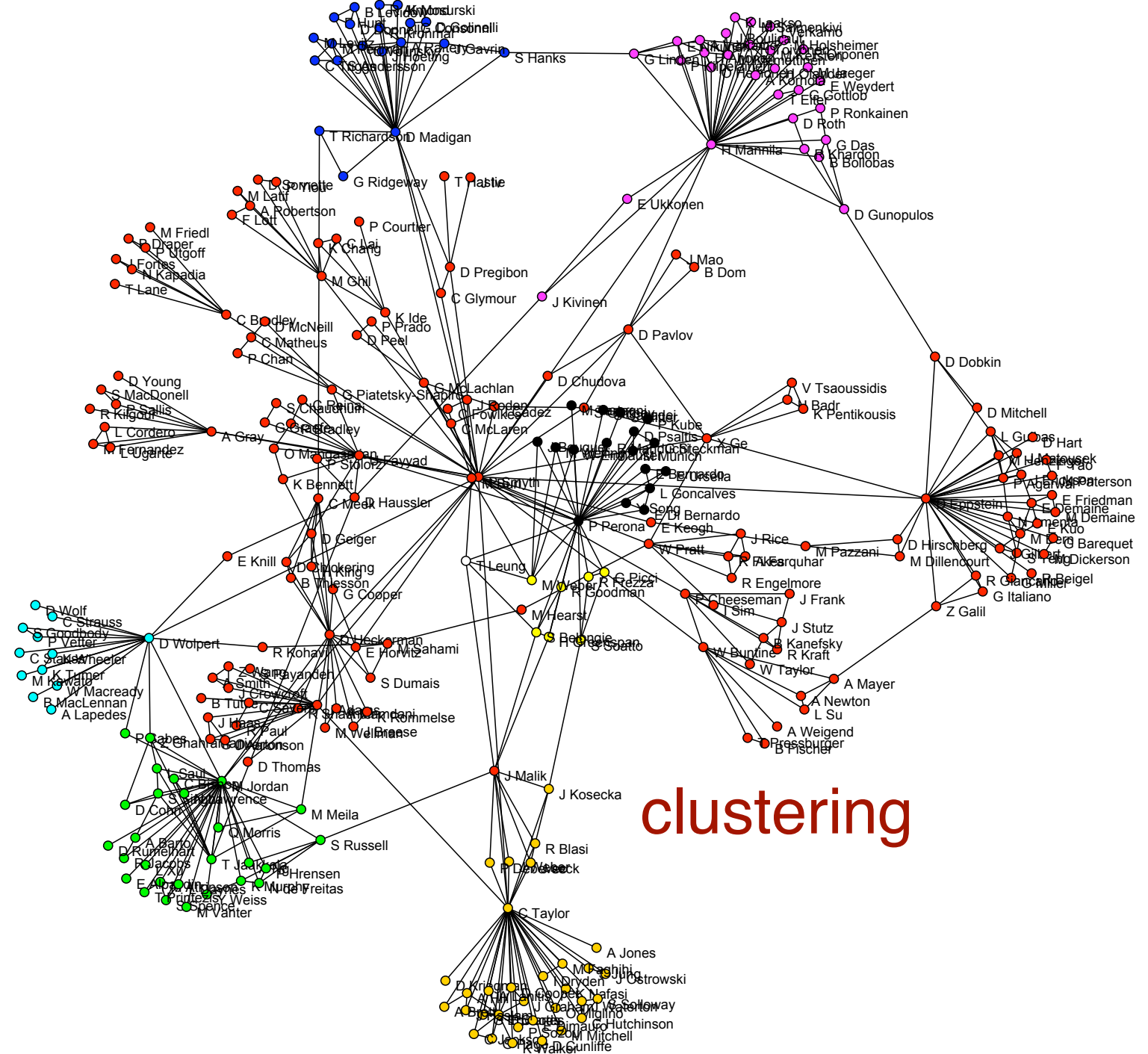
Layout algorithms specify vertex positions
advancePositions()

Renderers draw vertices and edges.

```
paintEdge( g, Edge, x1, y1, x2, y2 )  
paintVertex( g, Vertex, x, y )
```

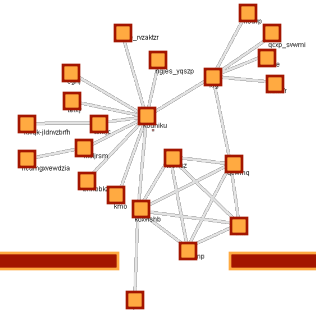



ranking

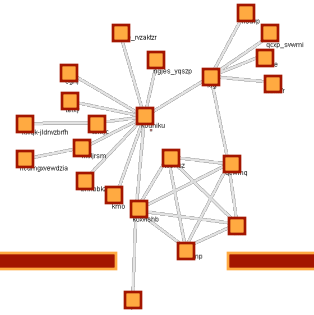


clustering

Statistical Analysis



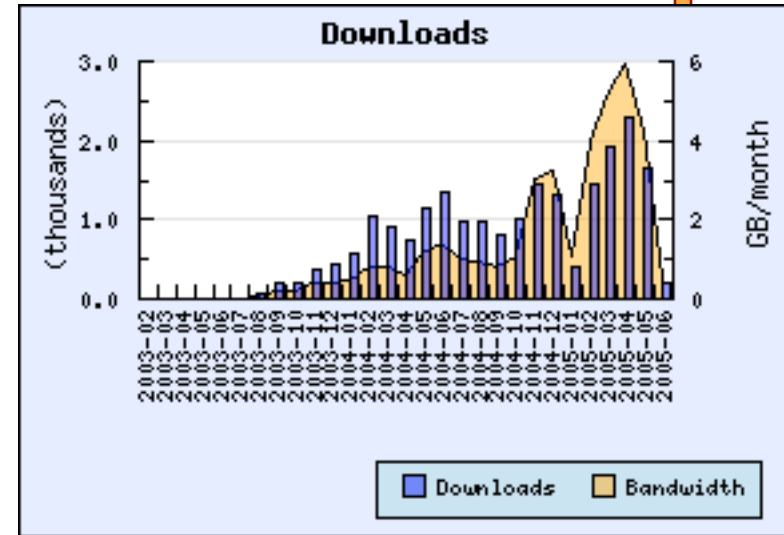
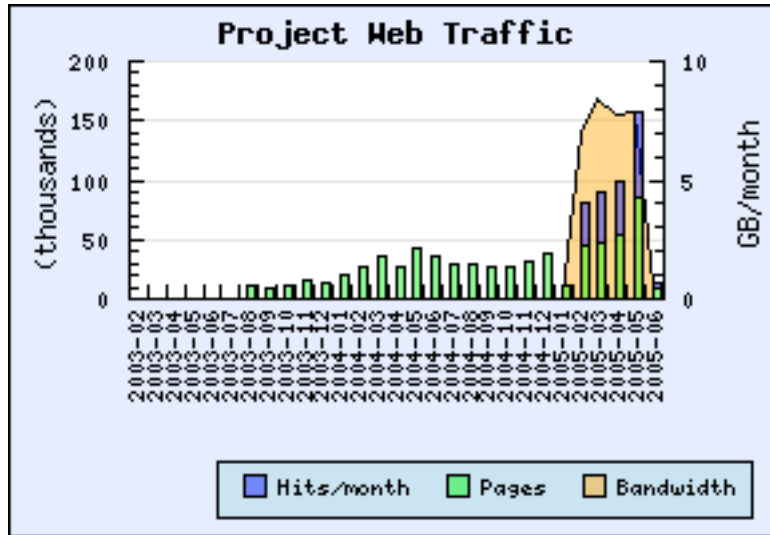
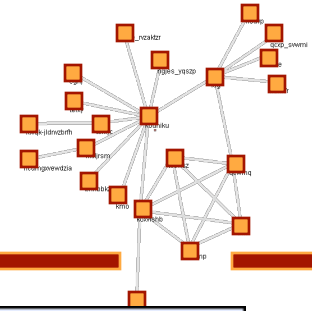
- Graph measures:
 - Degree distributions
 - Average shortest path length, diameter
 - Clustering coefficients
- can use existing Java statistical packages
 - CERN Colt Scientific Library
 - Visual Numeric's JMSL for Java



Conclusion

- Provides a common language for graphs
- Complements rather than replaces other graph packages/network tools
- A powerful framework for working with graphs with rich attribute structure
- Ideally suited for building tools/ applications related to network exploration and data mining

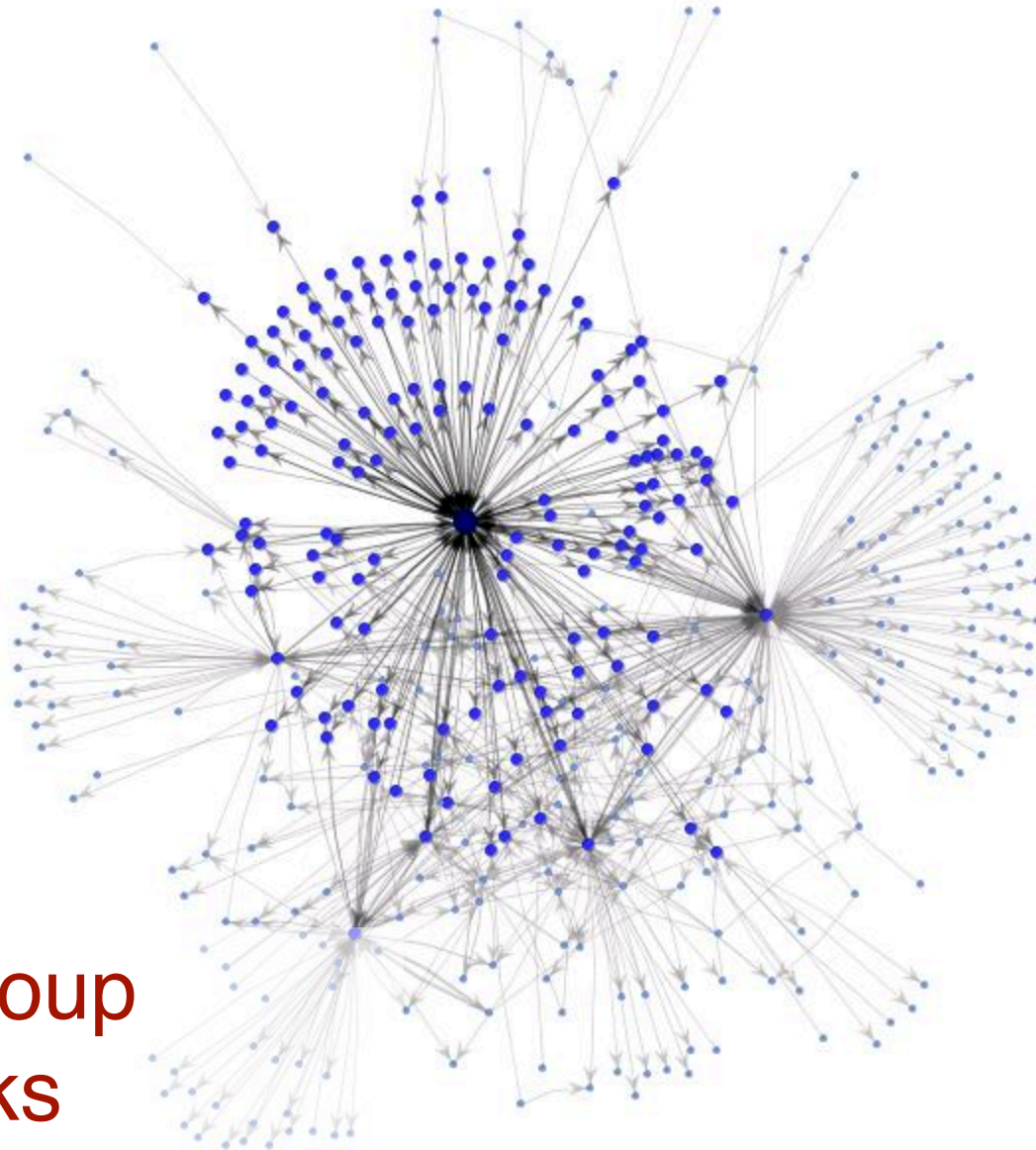
Adoption

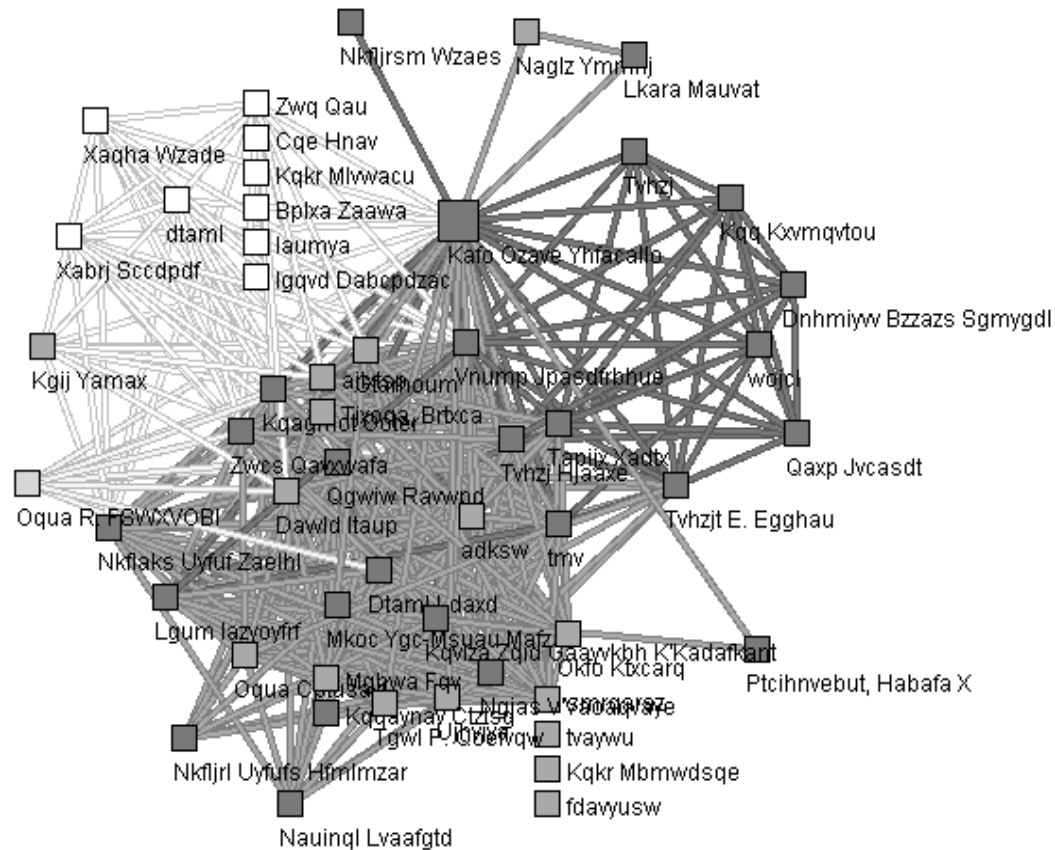


Users are writing

social network analyses, games, animated graph views, transportation network algorithms, Google Cartography, ...

Newsgroup Networks





Temporal Egocentric Communication Networks

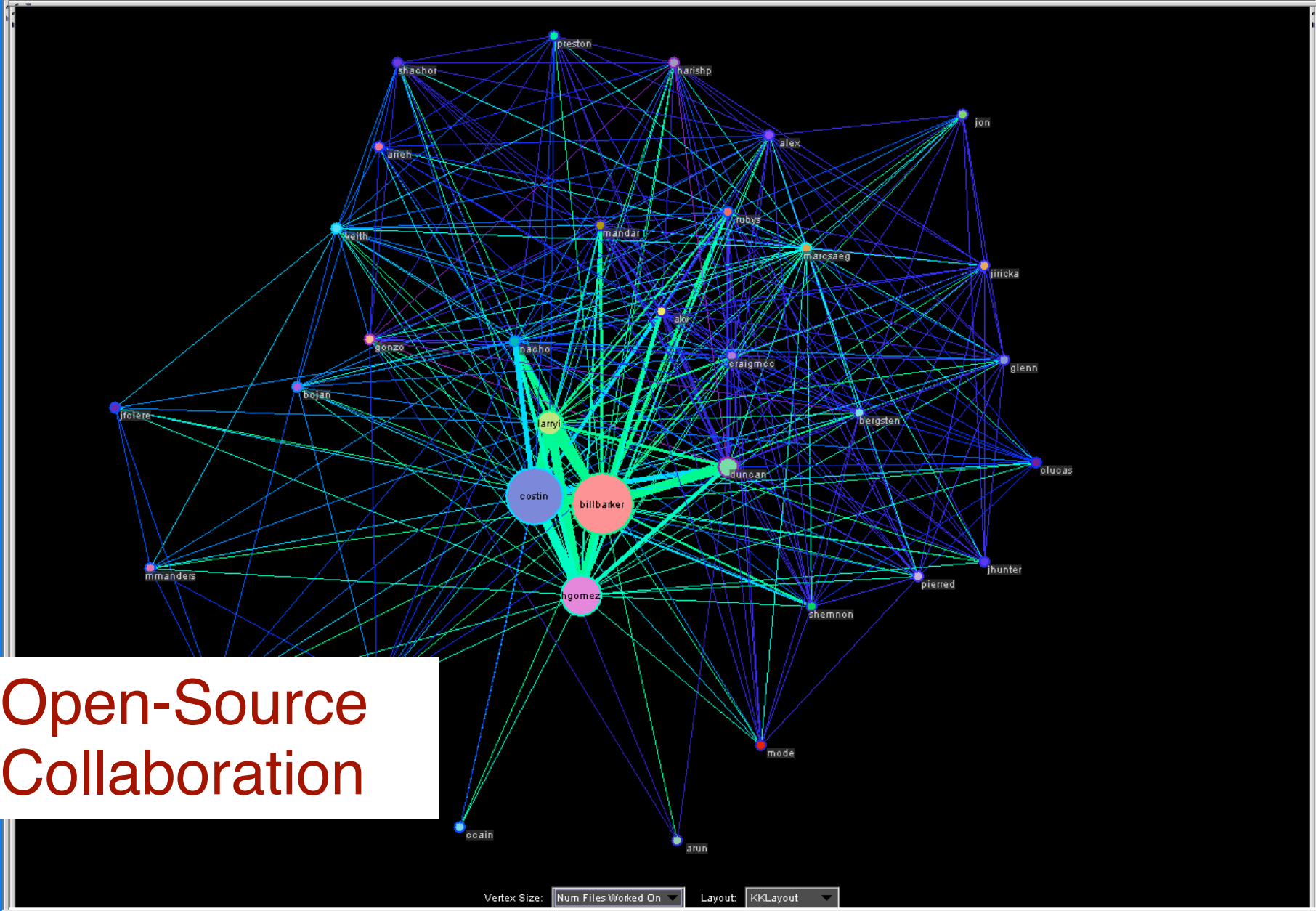
Display By: **Ade**

Indented:

Size:

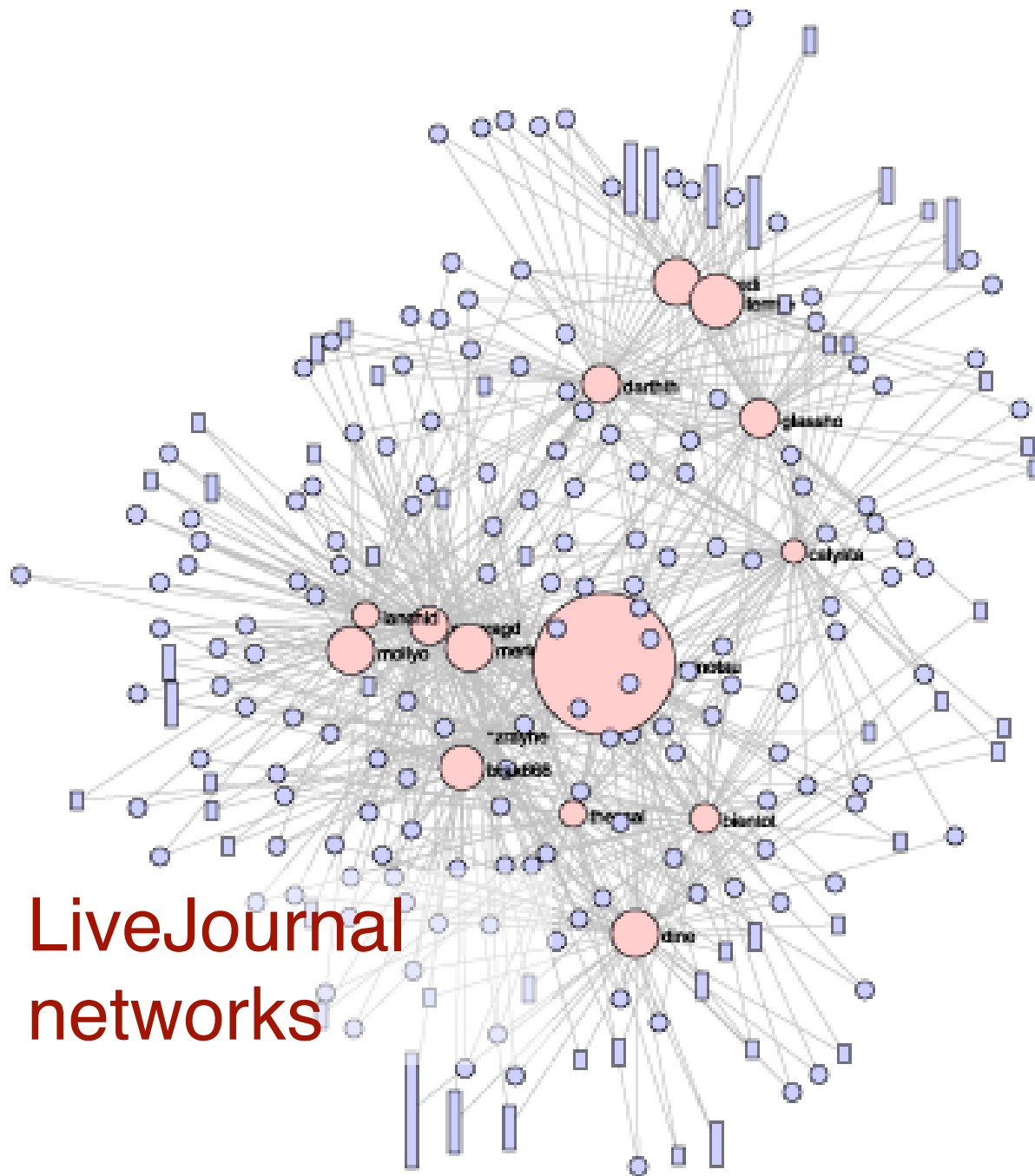
Time:

Past 4 years, 6 months

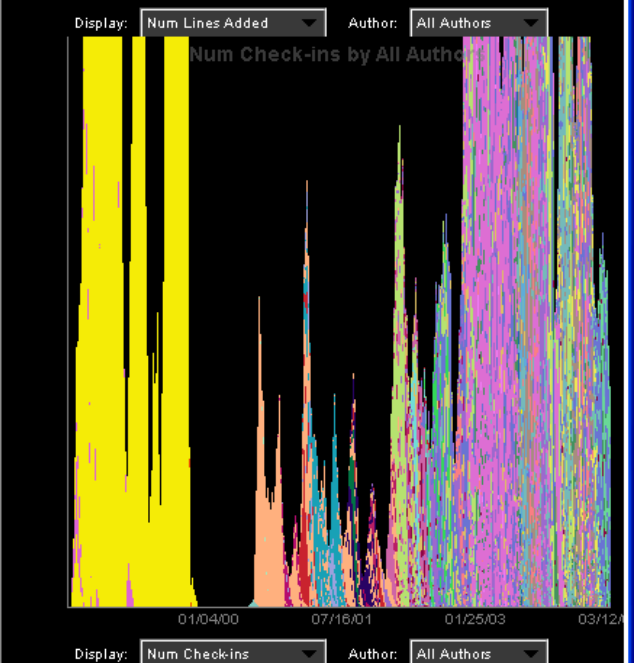
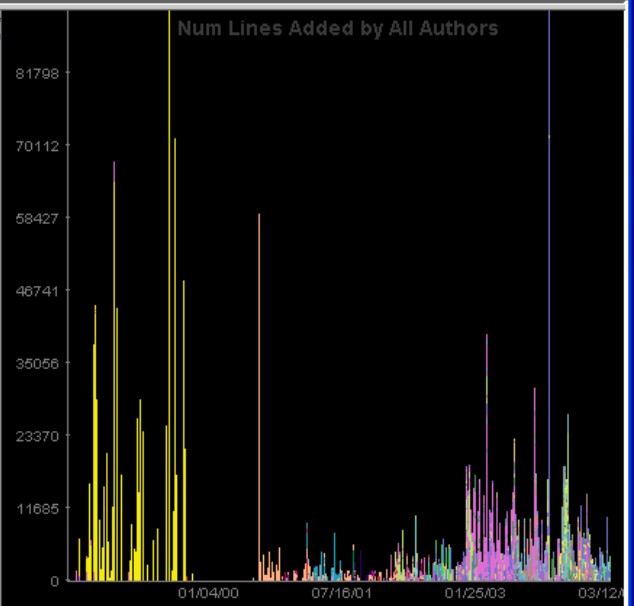
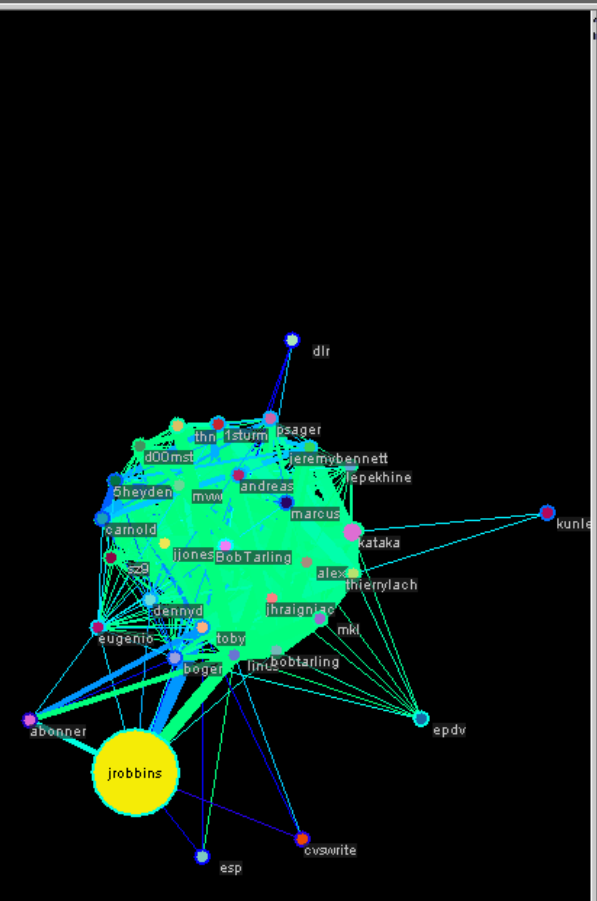
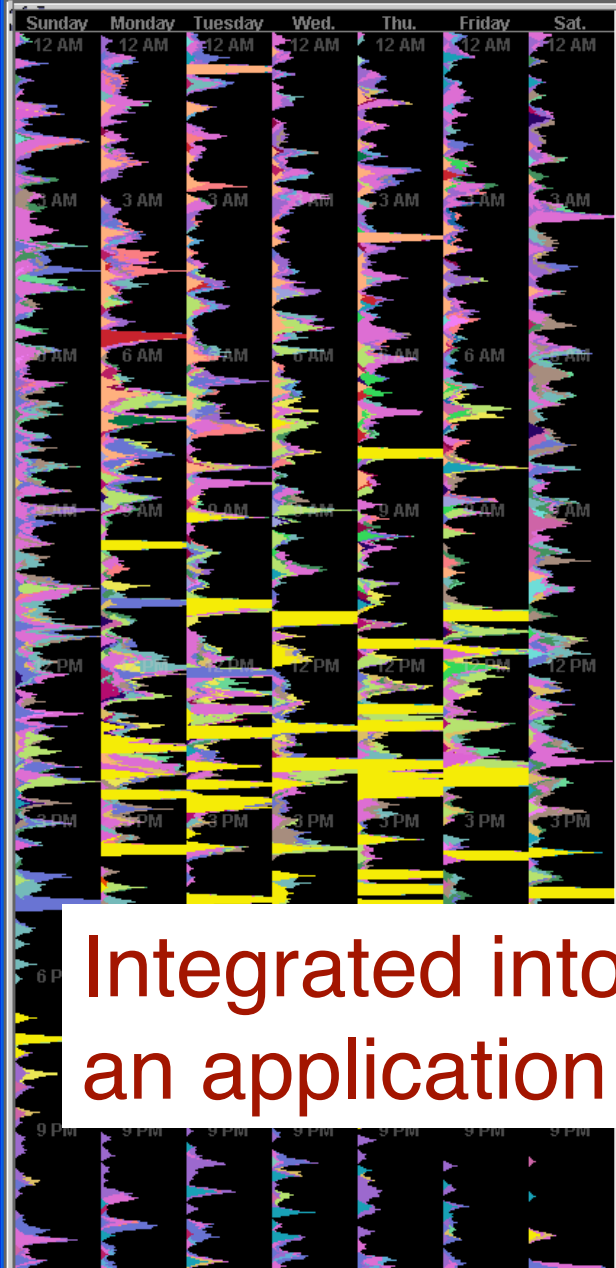


Open-Source Collaboration

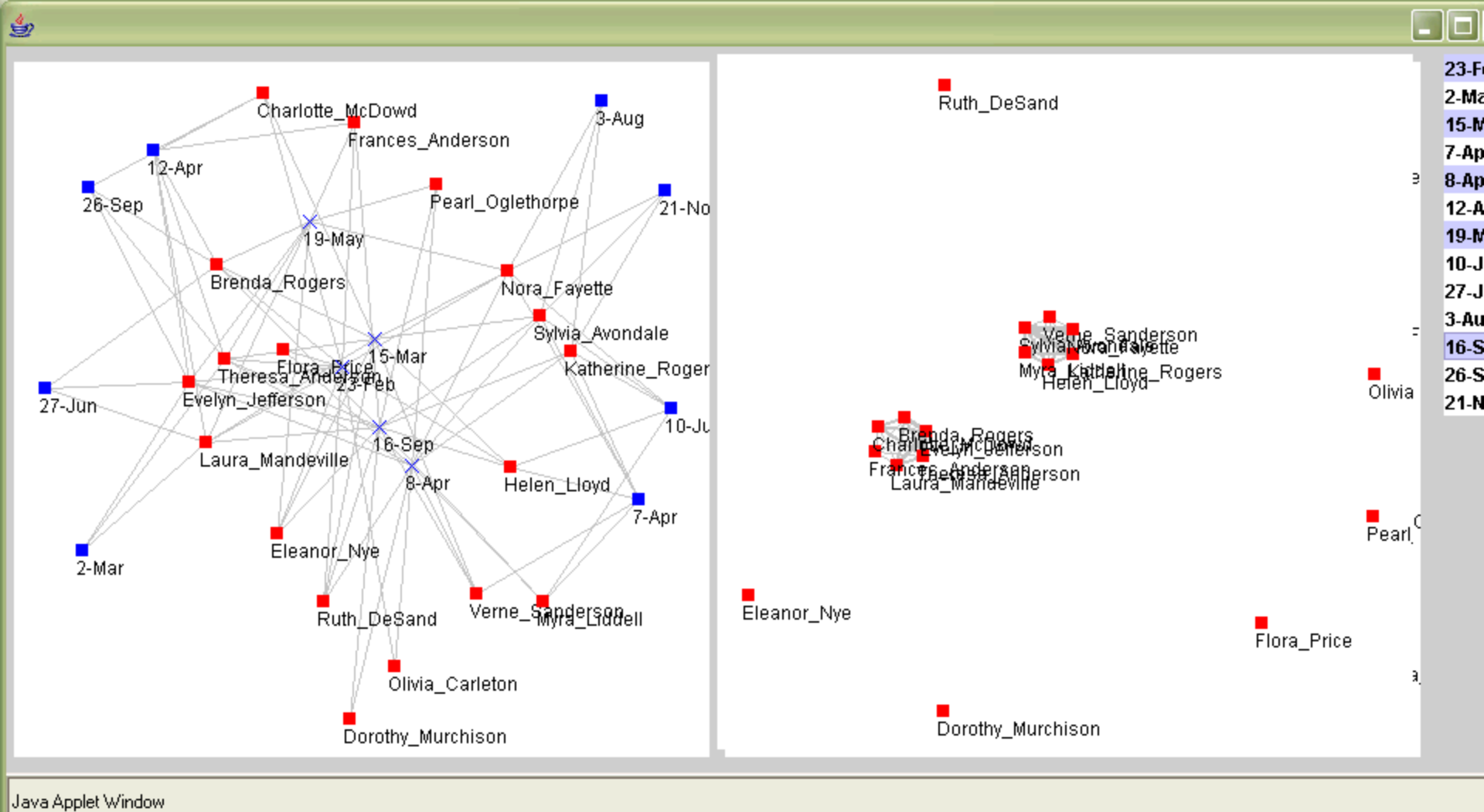
Vertex Size: Num Files Worked On Layout:



LiveJournal
networks

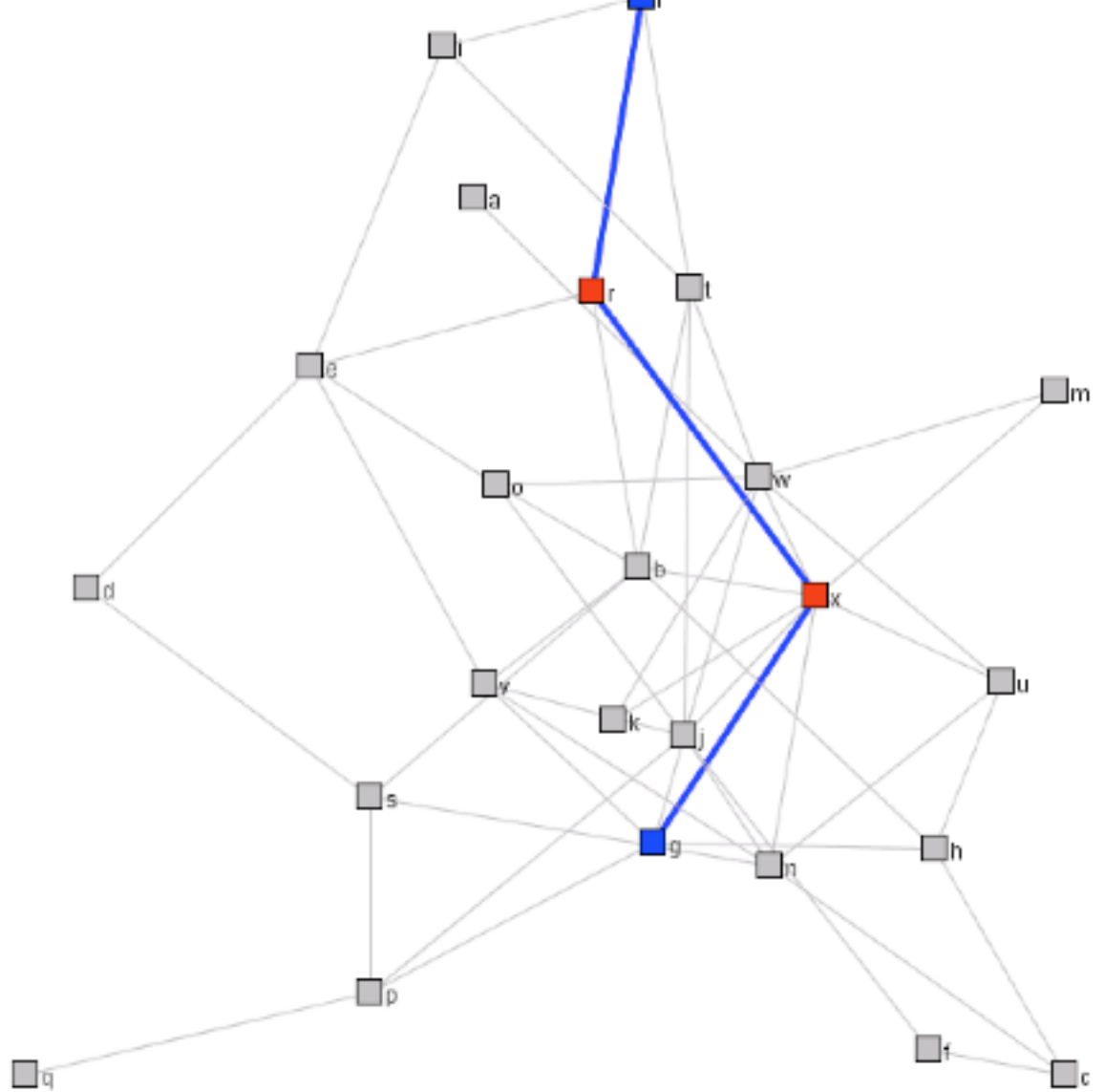


Integrated into an application



Freeman's Southern Women

Shortest Path

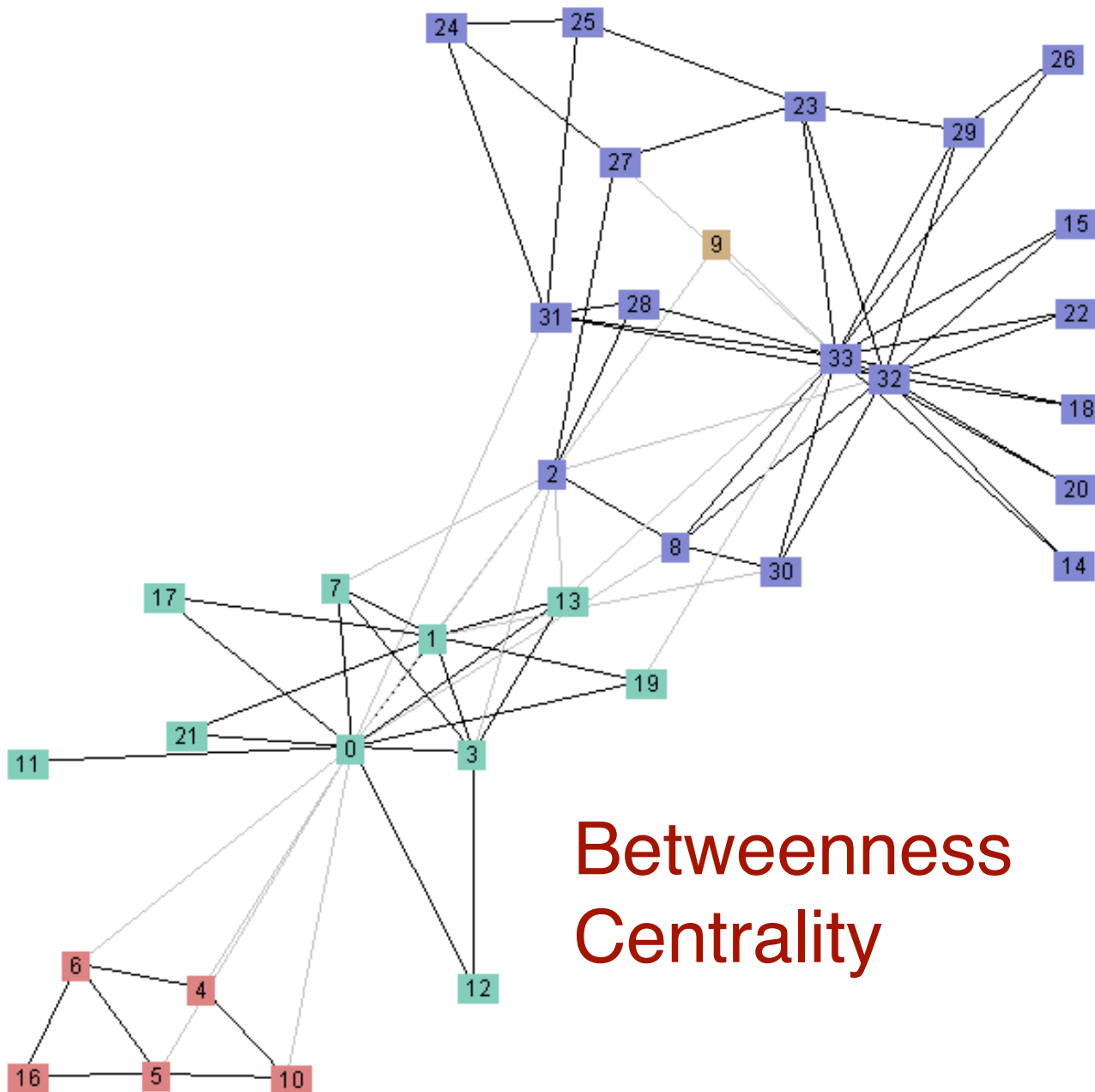


Select a pair of vertices to calculate a shortest path

vertex from


















































































vertex to

Betweenness Centrality



Restart

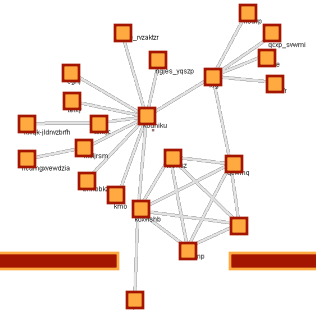
AuthToAuth_AltAlienVampireFlonkFlonkFlonk_Jan2001.txt

| IN_DEG | OUT_DEG | MSG-RC | MSG-SNT | MUTUAL | CENTRAL | DENSITY-W/CTR | DENSITY_W/O_CTR | 1ALL | 1IN | 1OUT | 1MUT | 2ALL | 2IN | 2OUT | MUT2 | Histogram |
|--------|---------|--------|---------|--------|----------|---------------|-----------------|---|---|--|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 0 | 0 | 0.5 | 1 |  |  |  |  |  |  |  |  |  |
| 0 | 1 | 1 | 0 | 0 | 0 | 0.5 | 1 |  |  |  |  |  |  |  |  |  |
| 23 | 20 | 34 | 34 | 9 | 9545.171 | 0.115 | 0.084 |  |  |  |  |  |  |  |  |  |
| 2 | 2 | 4 | 3 | 1 | 72.26 | 0.333 | 0 |  |  |  |  |  |  |  |  |  |
| 22 | 24 | 151 | 142 | 21 | 6899.885 | 0.22 | 0.162 |  |  |  |  |  |  |  |  |  |
| 2 | 2 | 2 | 2 | 1 | 0 | 0.75 | 0.833 |  |  |  |  |  |  |  |  |  |
| 2 | 4 | 20 | 20 | 2 | 534.047 | 0.55 | 0.417 |  |  |  |  |  |  |  |  |  |
| 3 | 3 | 10 | 14 | 2 | 98.094 | 0.55 | 0.417 |  |  |  |  |  |  |  |  |  |
| 4 | 2 | 11 | 19 | 2 | 0 | 0.75 | 0.75 |  |  |  |  |  |  |  |  |  |

Small Multiples

[Sun, E1]

Thanks



<http://jung.sourceforge.net>

Feel free to just play with it...

Or join us! Active collaboration is welcome.

Supported in part by NSF.