NetworkCube

Bringing Dynamic Network Visualizations to Domain Scientists

Benjamin Bach
Microsoft Research-Inria Joint Center
Nathalie Henry Riche
Microsoft Research, Redmond, WA
Roland Fernandez
Microsoft Research, Redmond, WA
Emmanouil Giannisakis
Inria, France
Bongshin Lee
Microsoft Research, Redmond, WA
Jean-Daniel Fekete
Inria, France

Motivation

- *How to provide domain scientists (e.g. researchers in neuroscience, historians) with easy and fast access to visualizations?*
- We prototyped two online platforms for visualizing dynamic multivariate networks.
- From our experience in creating those platforms and discussion with the involved domain experts, we learned:

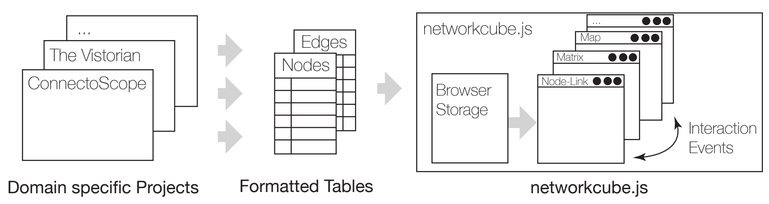
1) Data wrangling is different per application domain, but
2) visualizations are similar similar across domains.
3) Researchers tolerate bugs because few sessions on a prototype can lead to discoveries and save hours of analytical work.
4) Most domain experts are novice visualization users. Standard visualizations, e.g. node-link diagrams, can be instrumental to teach more novel ones via multiple coordinated views.
5) Data have high value and it is often not desirable to upload them onto a server.

- Current software for network visualization, especially from research [1], are sometimes domain-dependent, or are hardly available to practitioners due to lack integration into analysts' workflows.
- Current network visualization platforms [2,3,4] are not always up to date with visualization research, provide little more than node-link diagrams, and almost no visualizations for dynamic networks.

NetworkCube

- NetworkCube is a platform for network visualization, with a focus on dynamic networks.
- It is a javascript library that provides:
  - data import,
  - a dynamic graph API,
  - network visualizations and components (see right),
  - messages that synchronize views after user interaction,
  - general functionalities, such as history, brushing+linking, search, coloring.
- NetworkCube runs locally in a browser; user data is stored solely in the browser's local storage.
- Messages keep views coordinated across iframes, tabs, and even windows.

Usage



1 Format data into node and edge tables. That is the responsibility of the user or his platform (e.g. The Vistorian, ConnectoScope).

2
```
<script src="networkcube.js"></script>
```

3
```
networkcube.importData(dataObject);
```

4
```
networkcube.createVisualizationIFrame(visualizationName,
parentElement, sessionName, dataName)
```

See

Domain scientists can explore their data through domain platforms (see below) that leverage NetworkCube to store data and create visualizations.

Use

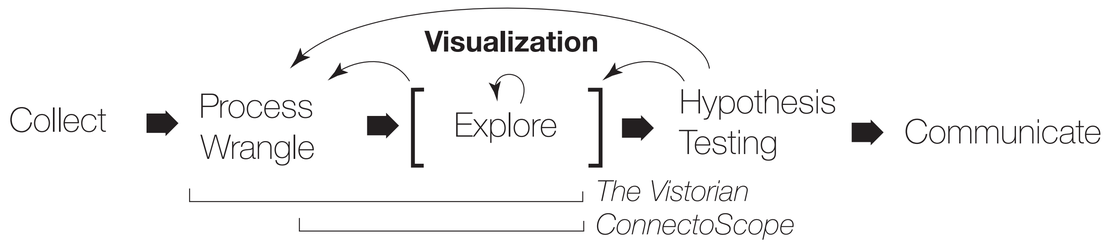
Programmers of domain platforms and websites can use NetworkCube's APIs to import data and integrate their visualizations.

Extend

Visualization researchers can extend NetworkCube, adding visualizations or functionality to investigate visualization usage in the wild.

Data Exploration Process

Data exploration involves several steps (left). NetworkCube currently focuses on providing visual exploration and data import.

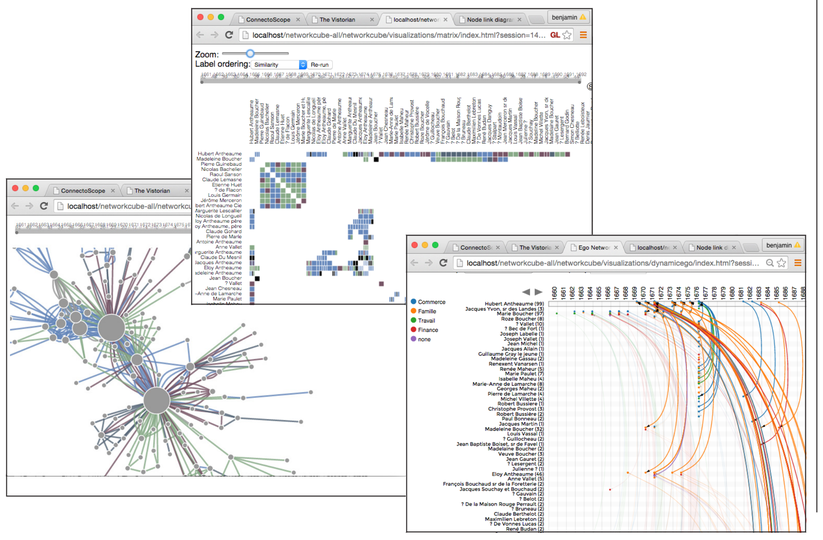


Domain Platforms

Networkcube emerged from designing and prototyping two network-exploration visualization platforms for domain scientists:

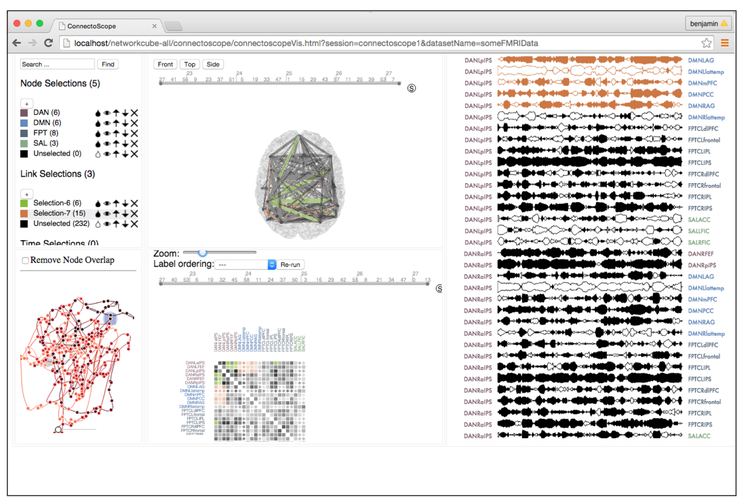
The Vistorian (beta)

The Vistorian is a visualization platform for historians. Users can map their manually assembled data tables to a network structure, by defining columns for source and target node, edge type, and time; then visualizing their data using node-link diagrams, adjacency matrices, and a dynamic ego-network visualization.



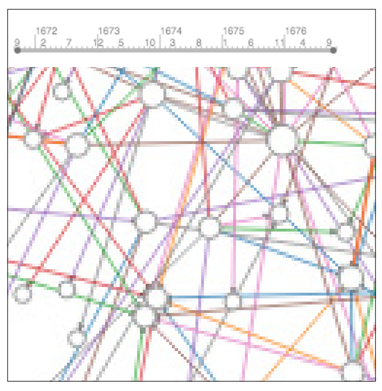
ConnectoScope

ConnectoScope is a visualization platform for brain connectivity in neuroscience. ConnectoScope allows exploring fMRI data from a neuroscience standard format. Brain connectivity networks are automatically extracted and visualized using a 3D glass brain visualization, adjacency matrices, and LinkWave [4]. Views are connected by brushing+linking, and colored selections.

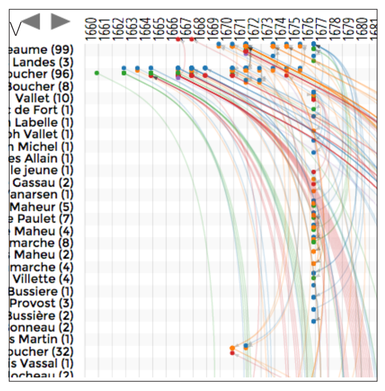


Visualizations

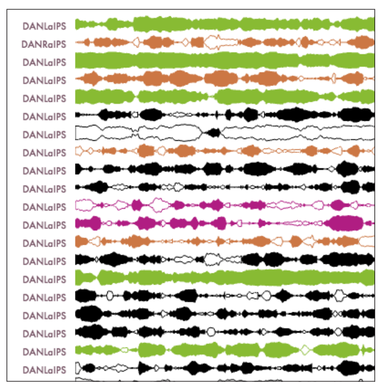
NetworkCube currently supports 6 visualizations for dynamic networks, connected through brushing and linking. All provide time navigation support.



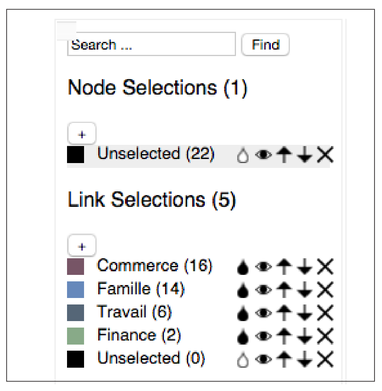
Node Link



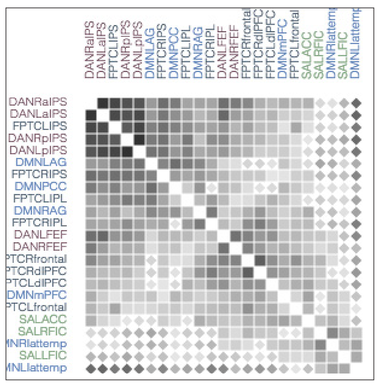
Timeline



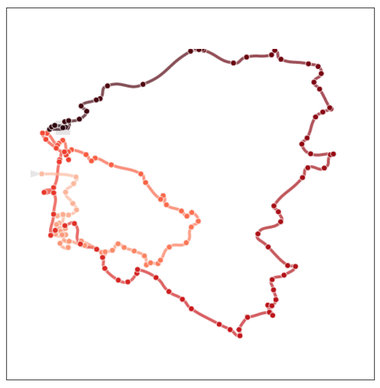
Linkwave [5]



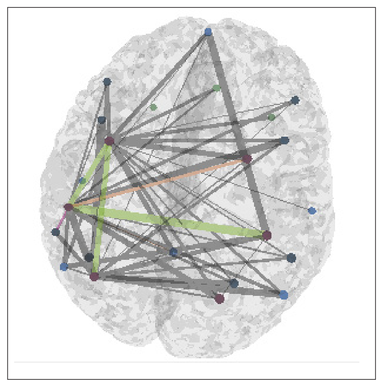
Browser



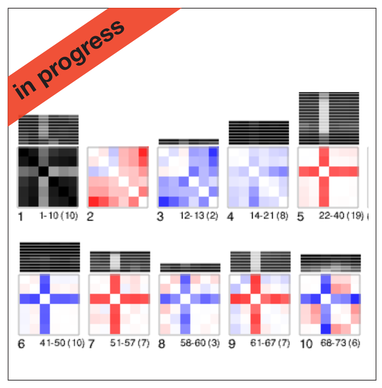
Adjacency Matrix [6,9]



Time Curves [7]



3D Glass Brain



MultiPiles [8,10]

References

[1] Fabian Beck, Michael Burch, Stepan Diehl, Daniel Weiskopf: **The State of the Art in Dynamic Graph Visualization.** *Eurographics Conerence on Visualization (EuroVis), State of the Art Report,* 2014.

[2] NodeXL, `http://nodexl.codeplex.com`

[3] Gephi, `http://gephi.github.io`

[4] Cytoscape, `http://www.cytoscape.org`

[5] Jean-Daniel Fekete: **Reorder.js: A JavaScript Library to Reorder Tables and Networks**. *Posters Compendium of IEEE VIS 2015, Chicago.*

[6] Nathalie Henry Riche, Yann Riche, Nicolas Roussel, Sheelagh Carpendale, Tara Madhyastha, Thomas J. Grabowski. **LinkWave: une Liste d'Adjacence Visuelle Interactive pour explorer les Réseaux Pondérés Dynamiques (LinkWave: an Interactive Visual Adjacency List for Exploring Dynamic Weighted Networks)** . In *Proceedings of Interactional Francophile Conference on Human Computer Interaction (IHM)*, 2014.

[7] Benjamin Bach, Conglei Shi, Nicolas Heulot, Tara Madhyastha, Tom Grabowski, Pierre Dragicevic **Time Curves: Folding Time to Visualize Patterns of Temporal Evolution in Data**, *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2015.

[8] Benjamin Bach, Nathalie Henry Riche, Tim Dwyer, Tara Madhayastha, Jean-Daniel Fekete, Tom Grabowski: **Small MultiPiles: Piling Time to Explore Temporal Patterns in Dynamic Networks,** *Eurographics Conference on Visualization (EuroVis),* 2015.

[9] Basak Alper, Benjamin Bach, Nathalie Henry Riche, Tobias Isenberg, Jean-Daniel Fekete: **Weighted Graph Comparison Techniques for Brain Connectivity Analysis,** Conference on Human Factors in Computing Systems (CHI), 2013

[10] Benjamin Bach, Emmanuel Pietriega, Jean-Daniel Fekete: **Visualizing Dynamic Networks with Matrix Cubes,** Conference on Human Factors in Computing Systems (CHI), Toronto, Canada, 2014,