Interactive Visual Data Analysis

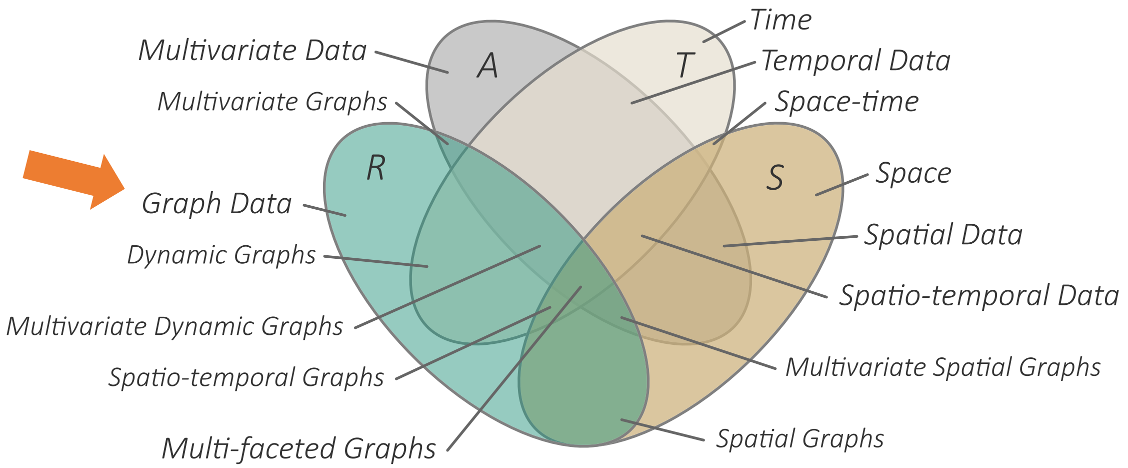
08 – Graph visualization

Objectives

- How to visualize graph data: Learn approaches and techniques for visualizing relations between objects
- How to include additional data facets: Learn basic approaches for visualizing multi-faceted graphs

Overview

- Graph data
- Basic visual representations
- Visualizing multi-faceted graphs

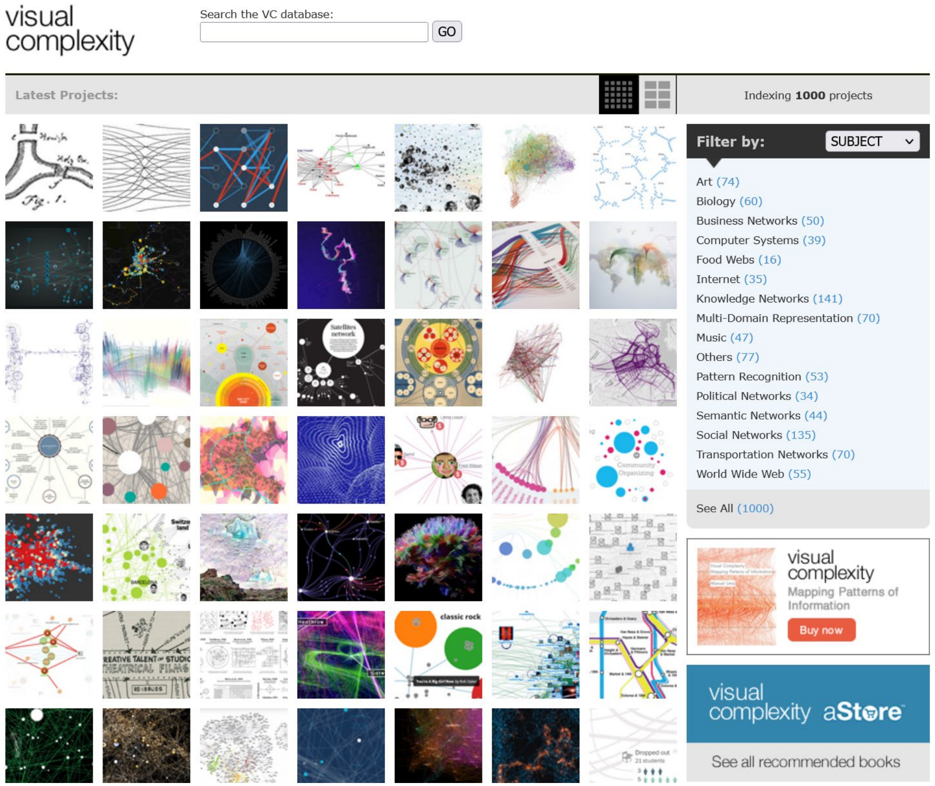


Motivation

- So far, we considered data attributes A in time T and space S
- Now we are concerned with **relations R** between data elements
- Graphs are a universal model for representing relations between entities
- Graphs relevant in many application domains
    - File system hierarchies
    - Computer networks
    - Social networks
    - Biological reaction networks
    - Power grids
    - Think about: Where else are graphs relevant?
- There is a great demand for graph visualization!

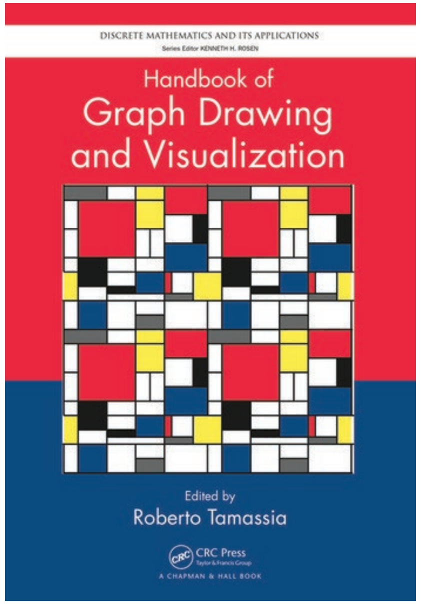
Motivation

- There is a great demand for graph visualization!
- There is also a great supply of graph visualizations!
- www.visualcomplexity.com



Graph visualization

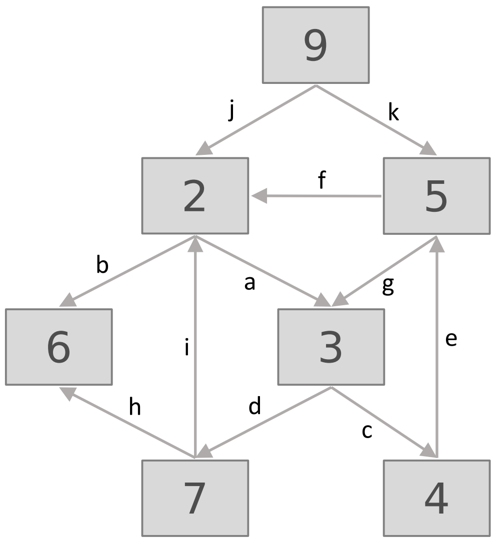
https://cs.brown.edu/people/rtamassi/gdhandbook/



- Graph visualization is concerned with
  - Showing structural relationships R
  - Representing data attributes A associated with R
  - Indicating potential dependencies on time T and space S

- Note: We can fill whole books with the topic of graph visualization alone, other books focus on multivariate graph visualization, and substantial articles take dynamic graph visualization and visualization of geographic networks into account.

Graph data

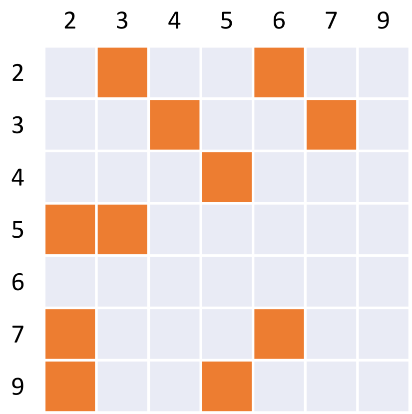


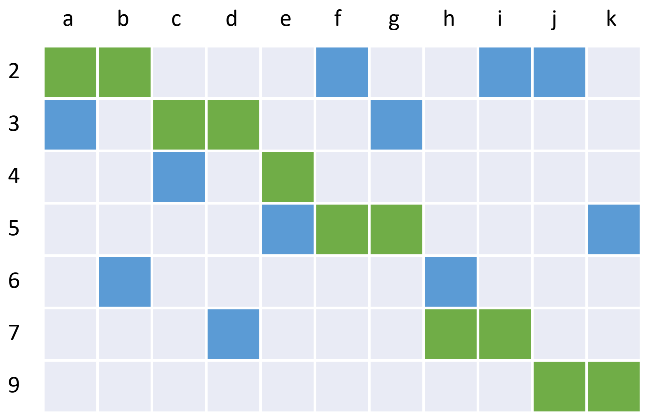
Graphs $G = (V, E)$

- Set of vertices V (the entities or data objects)
- Set of edges E (the relations between entities)
  - Directed pairs $(u, v) \in E : u, v \in V$
  - Undirected tuples $\{u, v\} \in E : u, v \in V$
- Additionally: Derived or associated data attributes

- Graphs stored as
  - **Set representation**
  - Adjacency list
  - Adjacency matrix
  - Incidence matrix

2	→ 3 → 6
3	→ 4 → 7
4	→ 5
5	→ 2 → 3
6	→
7	→ 2 → 6
9	→ 2 → 5





Graph data

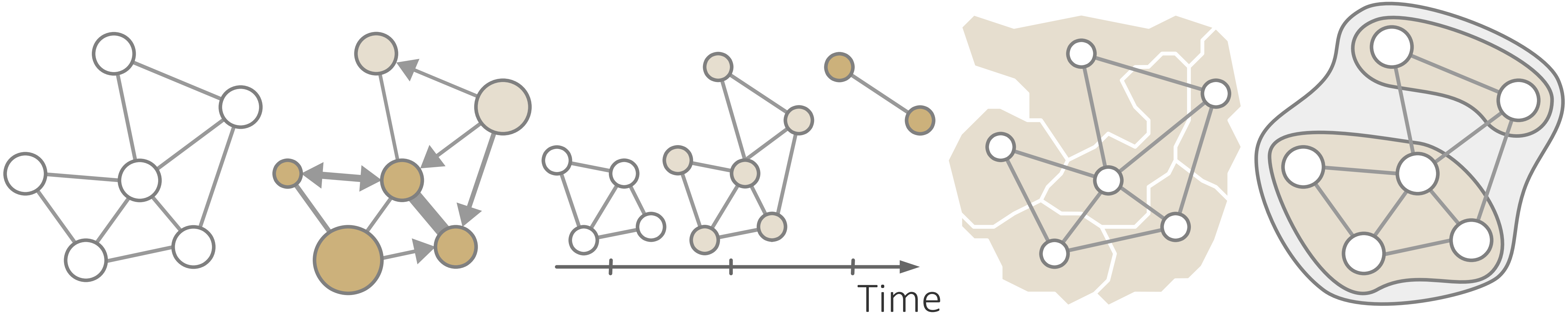
Graphs $G = (V, E)$

- Relevant types of graphs
  - **Networks**
    - No particular constraints for nodes and edges
    - Examples: social networks, computer networks, transportation networks, …
  - **Trees**
    - Connected, acyclic graph of parent-child relations (path exists between any nodes *u* and *v*, no path where start and end node are the same)
    - Designated *root* node, *leaf* nodes with no outgoing edges
    - Examples: File system, organizational hierarchies, …
  - Further:
    - Bi-partite graphs (two distinct subsets of nodes, edges only between subsets)
    - Hypergraphs (edges between more than two nodes)
    - Multi-layer graphs (graphs where nodes and edges exist on different layer)

From a graph-theoretical perspective, there are more than 1.600 graph classes (see https://graphclasses.org/).

Graph data

Facets of graphs

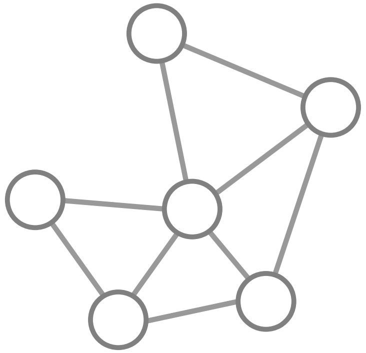


Plain graphs (R) Multivariate graphs (R+A) Dynamic graphs (R+T) Spatial graphs (R+S) Compound graphs (R+H)

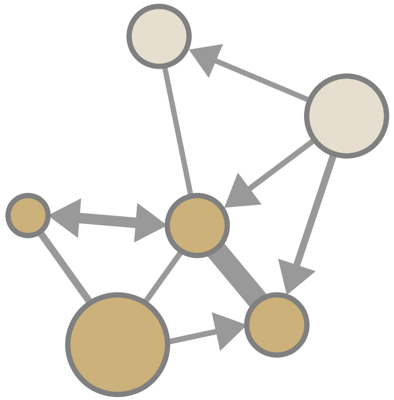
Graph data

Facets of graphs

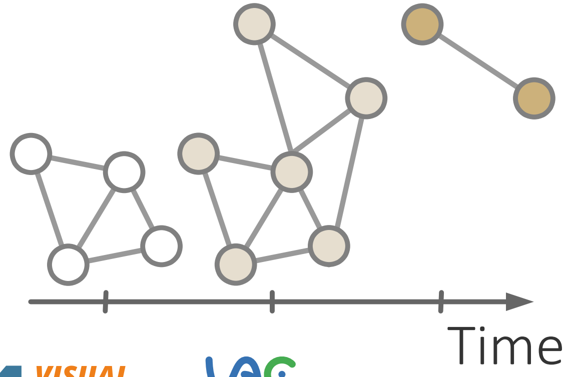
- Plain graphs (R)
  - Structure as primary facet



- Multivariate graphs (R+A)
  - Nodes and edges as data objects with additional given or derived data attributes
  - Example attributes: Node degree, edge weight



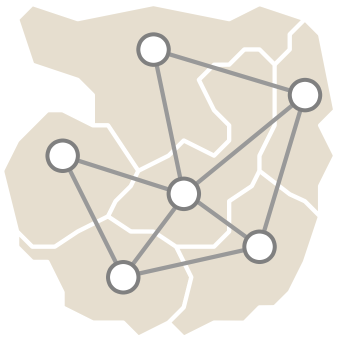
- Dynamic graphs (R+T)
  - Vary over time, where nodes and edges can continue to exists, leave, or enter the graph
  - Example: Social network with new persons and new connections



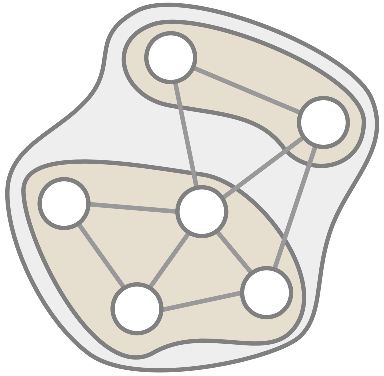
Graph data

Facets of graphs

- Spatial graphs (R+S)
  - Associated with coordinates in space
  - Coordinates may prescribe graph layout and edge routes
  - Example: Flights connecting airports, road network



- Compound graphs (R+H)
  - Nodes and edges partitioned into groups
  - Typically structured in a **nesting hierarchy H**
  - Groups can be expanded or collapsed to create different levels of abstraction
  - Example: Social network with organizational hierarchy on top, clustered graph



Analytical questions for graph data

- Think about: What data objects might be relevant when analyzing graphs?
- Relevant data objects
  - Nodes (degree, centrality, …)
  - Edges (directedness, weight, centrality, …)
  - Paths (length, costs, …)
  - Graphs and subgraphs (node and edge count, number of *, …)
    - Connected components (there is a path between any two nodes)
    - Clusters (or communities) defined by graph structure (e.g., cliques = fully connected subgraphs)
    - Groups of elements with similar properties (e.g., edges with high weights)

Analytical questions for graph data

- Think about: What analytical questions may we ask about graphs?
- Some example questions
  - Who is connected to whom?
  - Is there a connection between two entities?
  - Are there community structures?
  - Who is the central player in a network?
  - Are there critical connections (which cause trouble when broken)?
  - …
- Next, we look at a taxonomy for graph visualization tasks

Analytical questions for graph data

- Task taxonomy for graph visualization (Lee et al., 2006)
  - Topology-based tasks
    - Find the set of nodes adjacent to a node
    - Find the set of nodes accessible from a node
    - Given nodes, find a set of nodes that are connected to all of them
    - Find the shortest path between two nodes
    - …
  - Attribute-based tasks
    - Find the nodes having a specific attribute value
    - Given a node, find the nodes connected only by certain types of links
    - Which node is connected by a link having the largest/smallest value
    - …

Analytical questions for graph data

- Task taxonomy for graph visualization (Lee et al., 2006)
  - Browsing tasks
    - Follow a given path
    - Return to a previously visited node
    - …
  - Overview tasks
    - Compound exploratory tasks for quick estimation of
      - Graph size
      - Number of connected components
      - …

Basic visual representations

- We now know about graph data and corresponding analytical questions, next we study basic visual representations for graphs

Four fundamental visual designs

- Node-link representations
- Matrix representations
- Implicit representations
- Hybrid representations

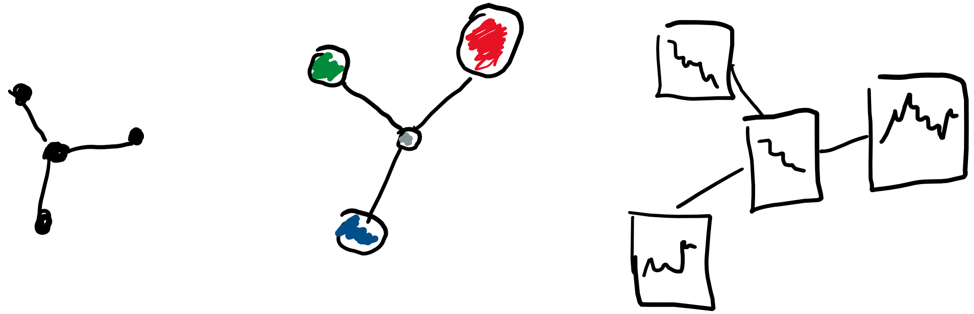
Node-link representations

- Design questions
  - How to represent nodes?
  - How to represent edges?

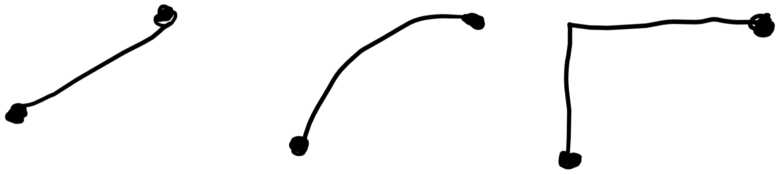
  - How to layout nodes?
  - How to route edges?

Node-link representations

- **How to represent nodes?**
  - Basic dots
  - Size-varying, color-coded circles
  - Glyphs



- **How to represent edges?**
  - Straight lines
  - Curves
  - Orthogonal paths



Node-link representations

- Think about: What are criteria for a good node layout and edge routing?

- **Aesthetics criteria** (Bennett et al., 2007)
  - Node metrics
  - Edge metrics
  - Overall layout metrics

Node-link representations

- **Aesthetics criteria** (Bennett et al., 2007)
  - Node metrics
    - Cluster similar nodes
    - Distribute nodes evenly
    - Keep nodes apart from edges
    - …
  - Edge metrics
    - Minimize edge crossings
    - Keep edge length uniform
    - Minimize edge length
    - Minimize edge bends
    - …

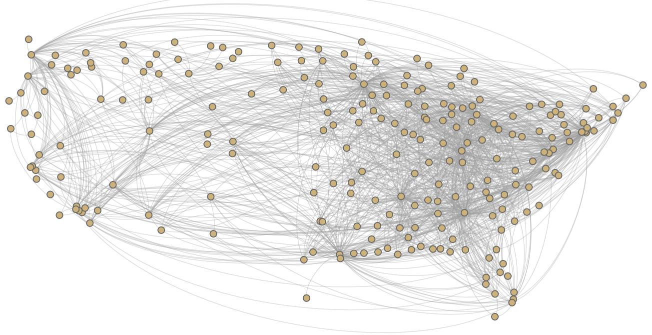
Node-link representations

- **Aesthetics criteria** (Bennett et al., 2007)
  - Overall layout metrics
    - Maximize consistent flow direction
    - Keep correct aspect ratio
    - Minimize area
    - Maximize local/global symmetry
    - …
  - Summary
    - Various partially conflicting aesthetics criteria
    - No silver-bullet solution → **Need to find good compromise**

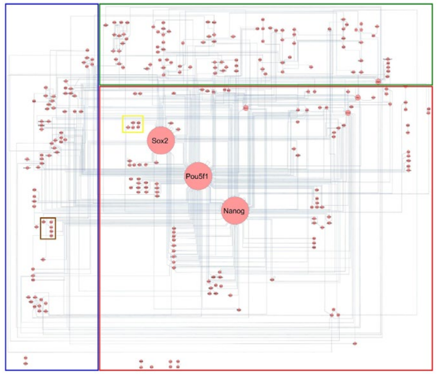
Node-link representations

- **How to layout nodes?**
  - Fixed layout
    - Node positions are predefined and cannot be varied
    - Example: Flight route network
  - Stylized layout
    - Possible positions confined to some predefined scheme (e.g., circle, regions)
    - Node positions may vary only with respect to scheme (e.g., on circle, in region)
    - Examples: **Magic-eye view, Fisheye treeview** for trees
  - Free layout
    - Nodes can be positioned freely
    - Example: **Force-directed layout** for general graphs

Fixed layout of US flight routes



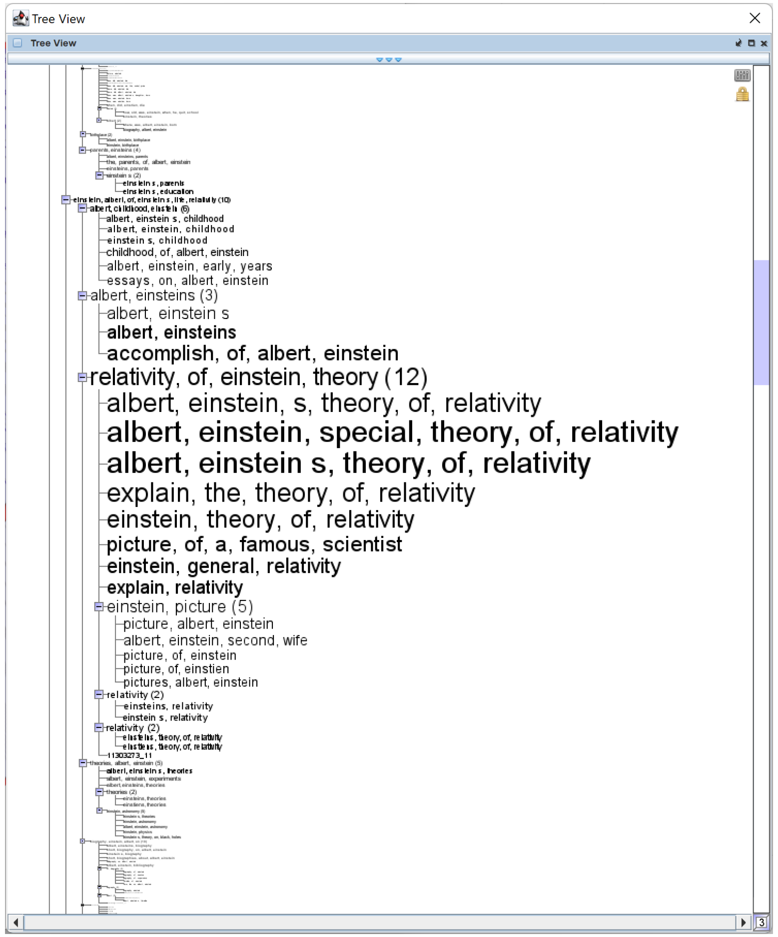
Stylized layout of network of 574 molecular interactions, stimulations and inhibitions



Node-link representations

- **Fisheye treeview** (stylized layout)
  - Classic indentation-based treeview
  - Focused on showing node labels
  - Enhanced with:
    - Zoom
    - Animations
    - Fisheye distortion

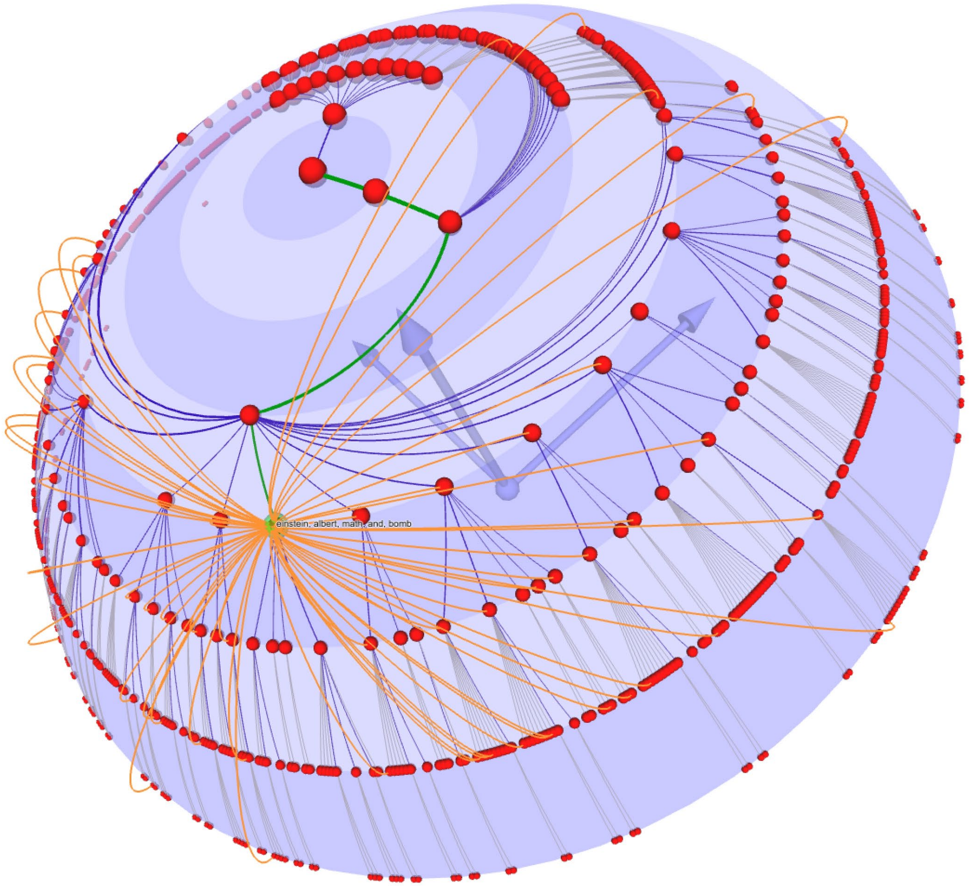
  - Demo



Node-link representations

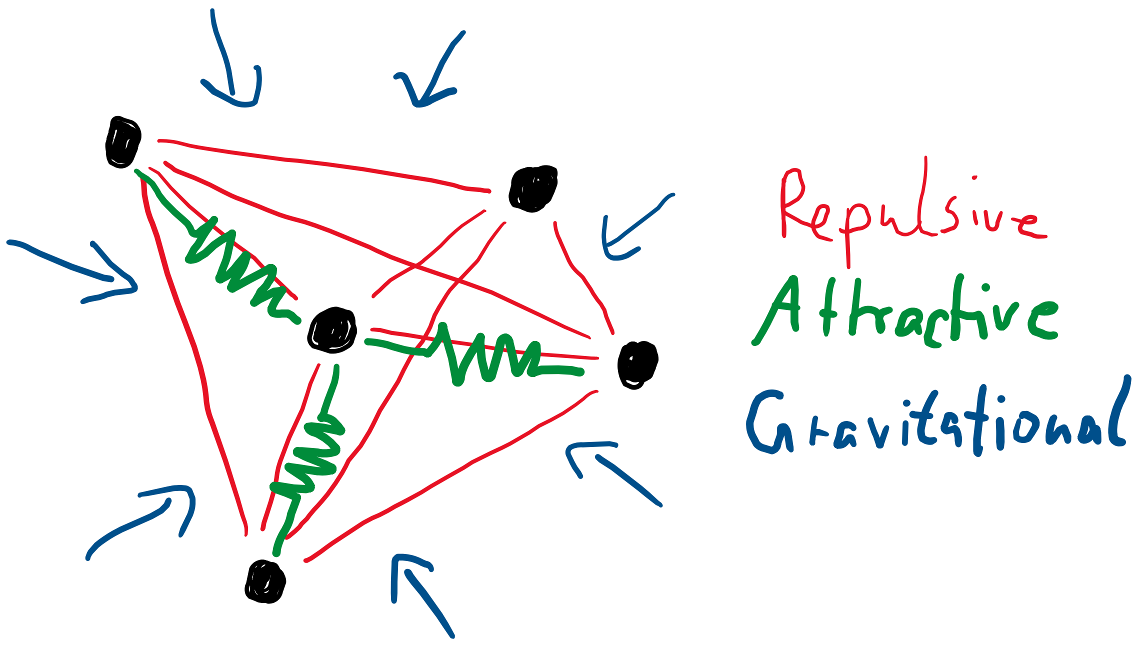
- **Magic Eye View** (stylized layout)
  - Radial variant of Walker's layout
  - Projected onto 3D hemisphere
  - Enhanced with:
    - Cross edges
    - Focus-point adjustment
    - Interactive rotation

  - Demo



Node-link representations

- **Force-directed layout** (free layout)
  - Compute graph layout via physical forces:
    - Nodes as point masses with position, velocity, and acceleration
    - Two primary types of forces:
      - **Repulsive forces** between all nodes
      - **Attractive forces** between connected nodes (e.g., edges as springs)
    - Additional forces:
      - Gravity to keep disconnected nodes and subgraphs from floating away
      - Cluster forces to keep distances between clusters

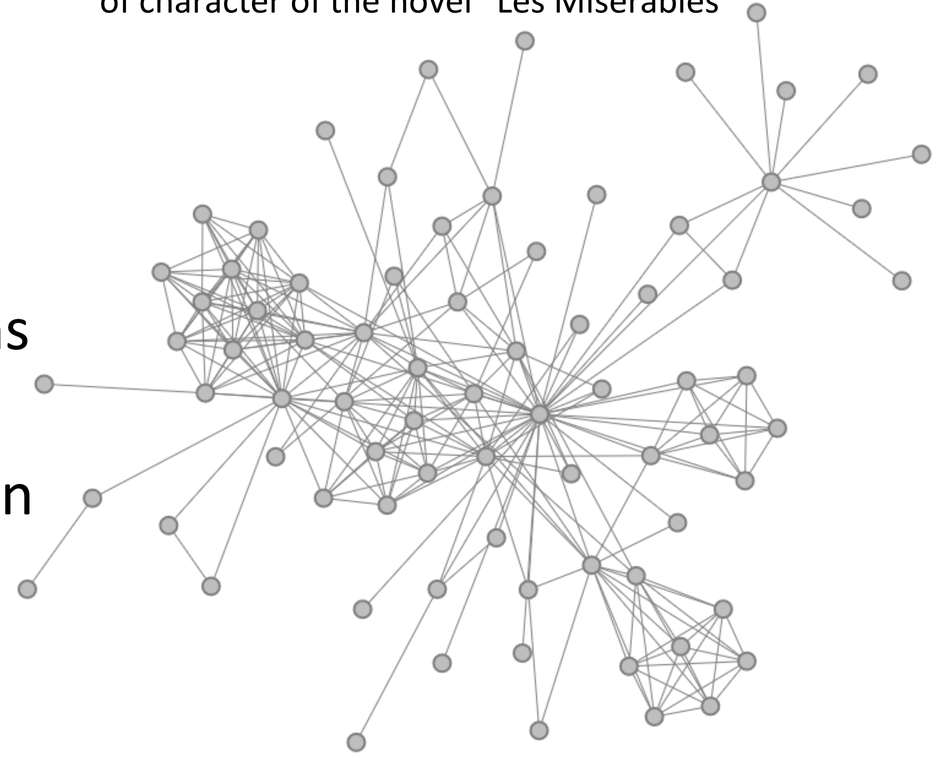


Node-link representations

- **Force-directed layout**

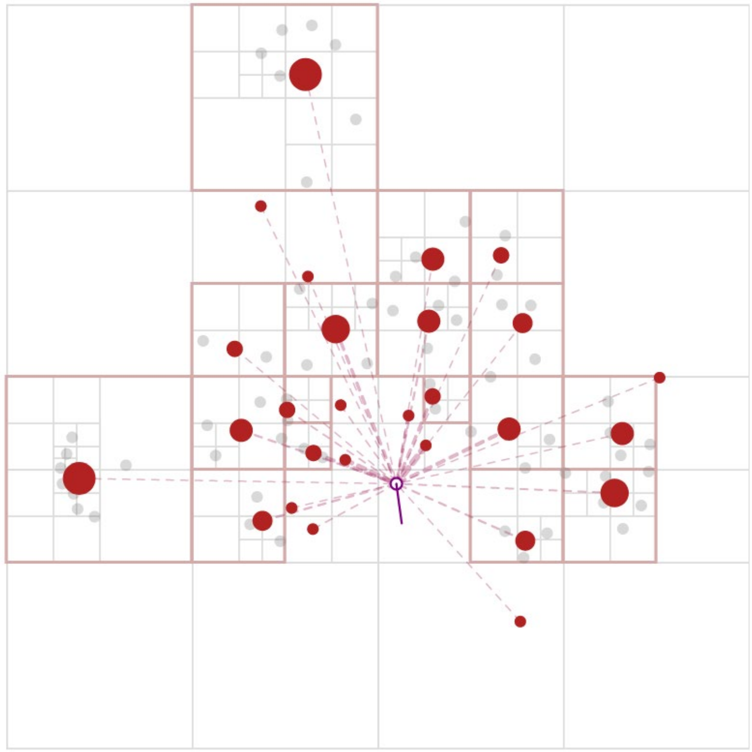
  1. Initialize **positions** (randomly)
  2. Compute **forces** based on current positions
  3. Compute force-induced **accelerations**
  4. Update **velocities** according to acceleration
  5. Update **positions** according to velocity
  6. Repeat 2. until reasonable stop criterion

Force-directed layout of the co-occurrence network of character of the novel "Les Miserables"



Node-link representations

- **Force-directed layout**
  - Think about: What is the computational complexity?
  - Computational complexity
    - Naïve $O(n^2)$
    - Barnes-Hut $O(n \log n)$



https://jheer.github.io/barnes-hut/

Node-link representations

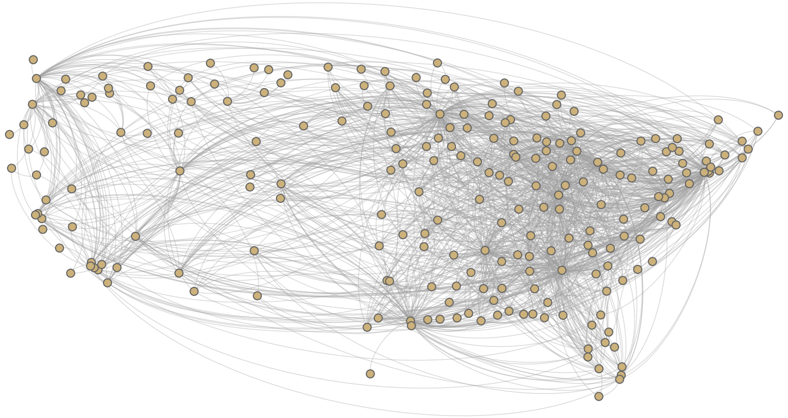
- **Force-directed layout**
  - Different variants of force-directed layout algorithms
    - Eades
    - Fruchterman & Reingold
    - Kamada & Kawai
    - ForceAtlas, ForceAtlas2
    - Yifan Hu's Multi-level layout

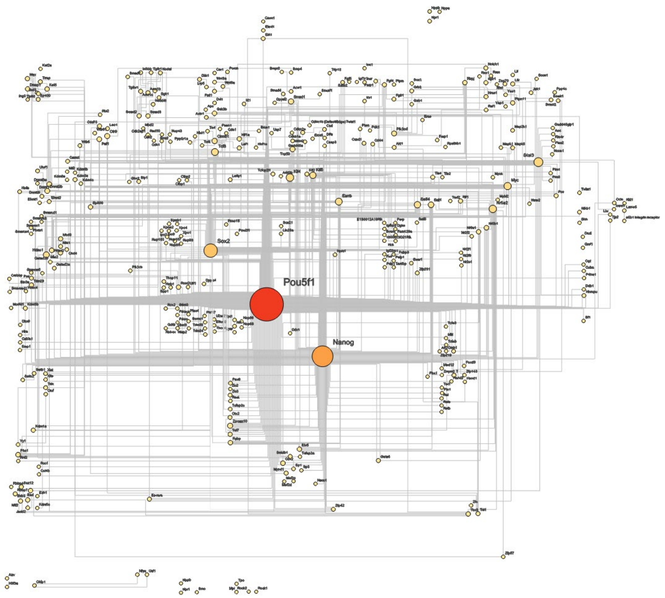
There is a reading assignment that will provide some more information on the Eades, Fruchterman & Reingold and the Kamada & Kawai algorithms

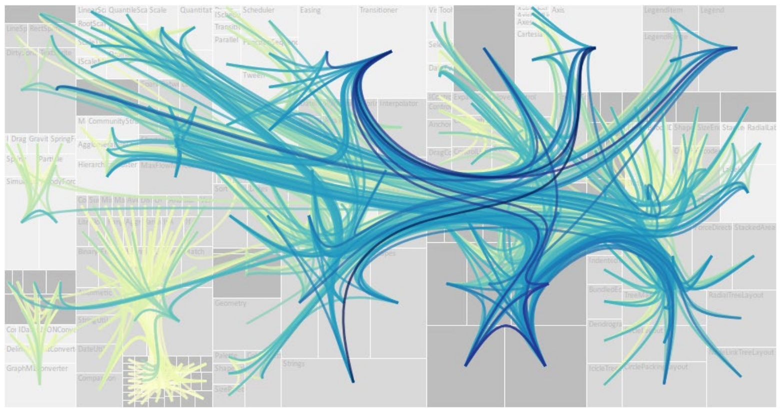
- Gephi software offers different options for testing https://gephi.org

Node-link representations

- **How to route edges?**
  - Straight edges
  - Orthogonal edges
  - Curved edges
  - Edge bundling





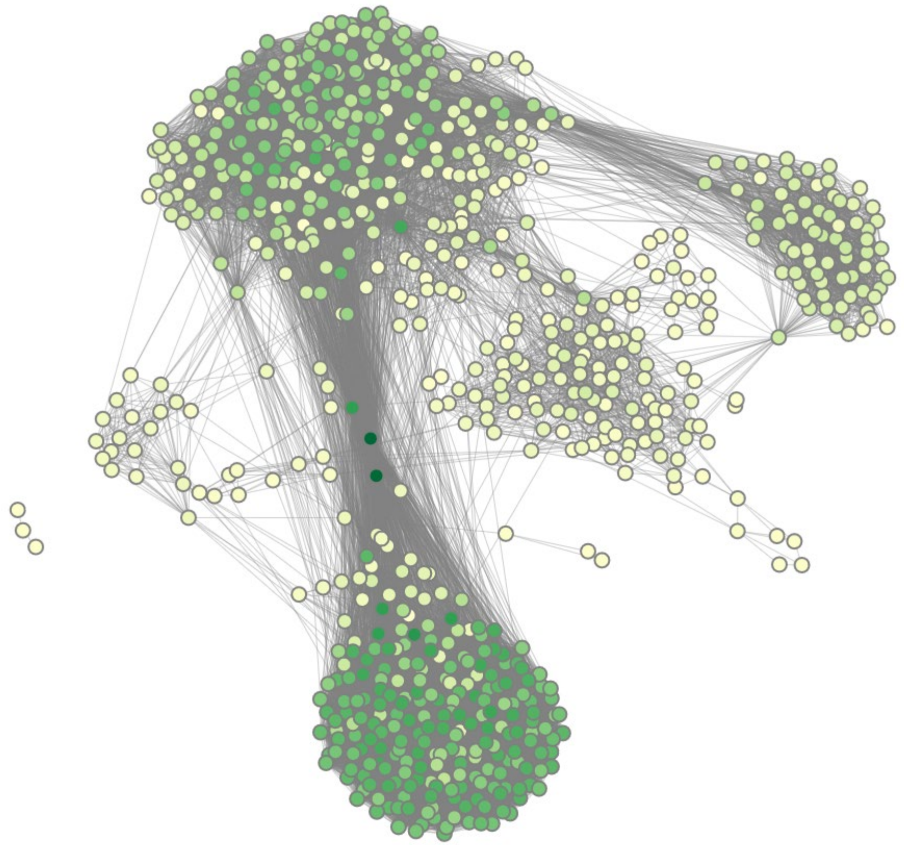


Node-link representations

Summary

- Advantages
  - Good overview of graph structure
  - All graph elements represented explicitly
  - Generally applicable
- Disadvantages
  - Complex layout calculation
  - Numeric (in)stability
  - “Hairball” for large graphs

Force-directed layout of a social network



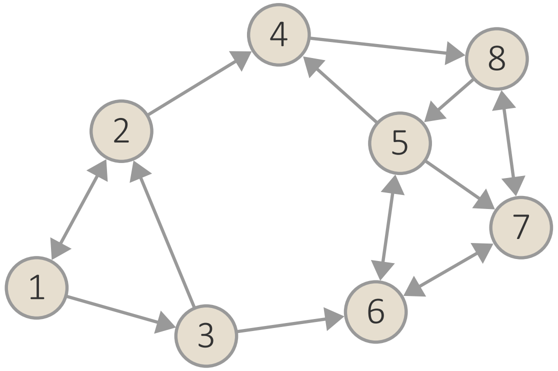
Basic visual representations

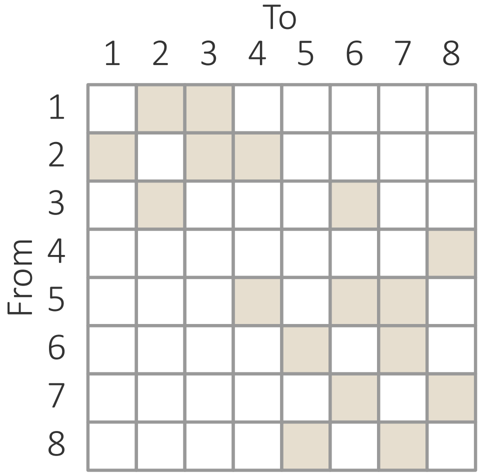
Four fundamental visual designs

- Node-link representations
- Matrix representations
- Implicit representations
- Hybrid representations

Matrix representations

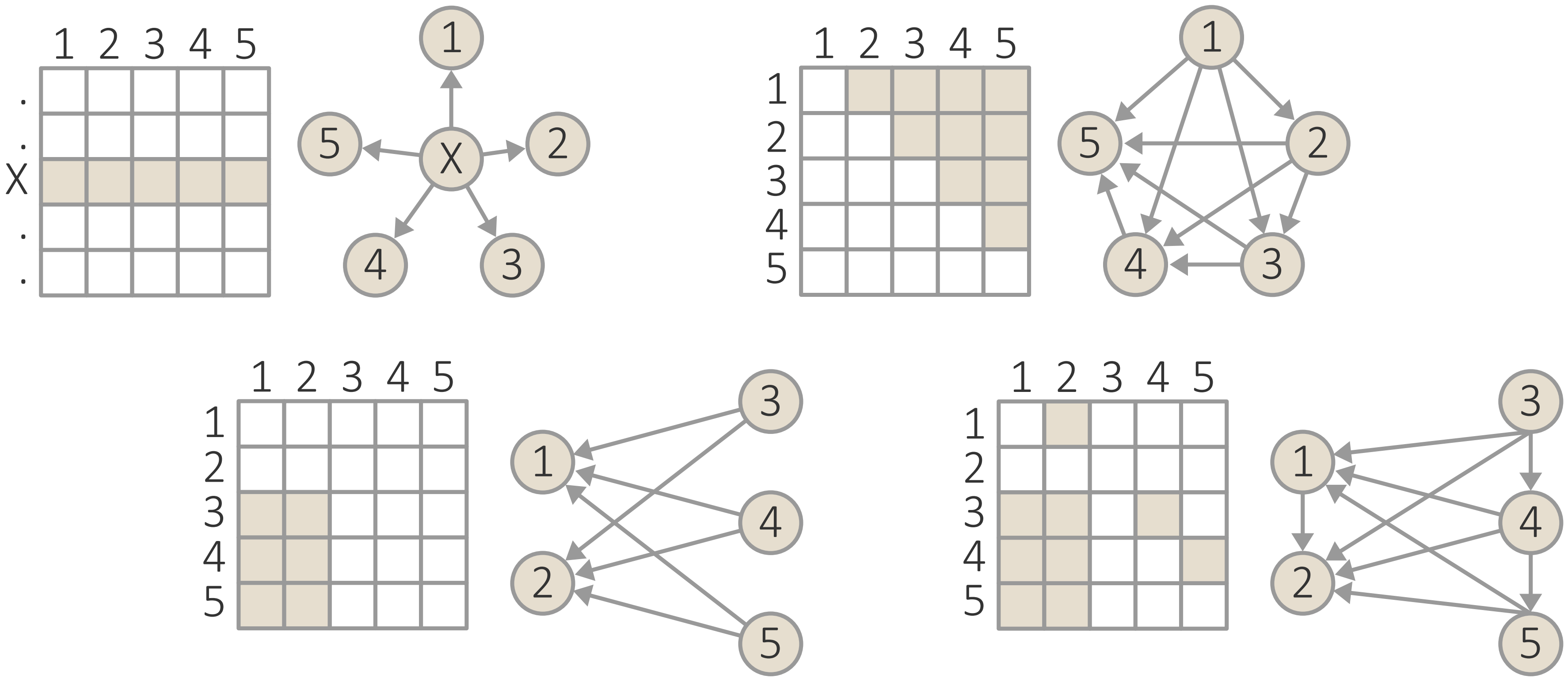
- Graph represented as matrix
- Matrix rows and columns correspond to nodes
- Matrix cells visualize edges
  - Filled cells indicate existence of edge between column node and row node
  - Edge attribute(s) visualized via
    - Color-coding
    - Embedded miniature visualization (e.g., glyphs)





Matrix representations

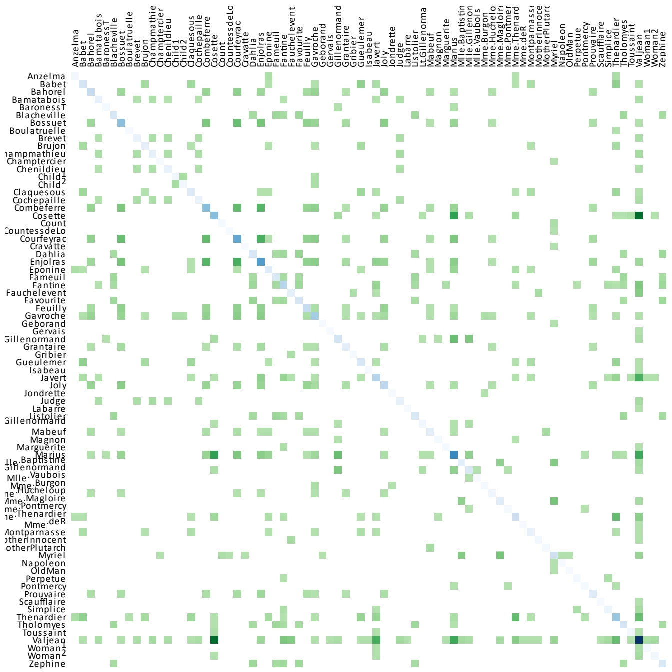
- Visual patterns indicting structural properties of graphs



Matrix representations

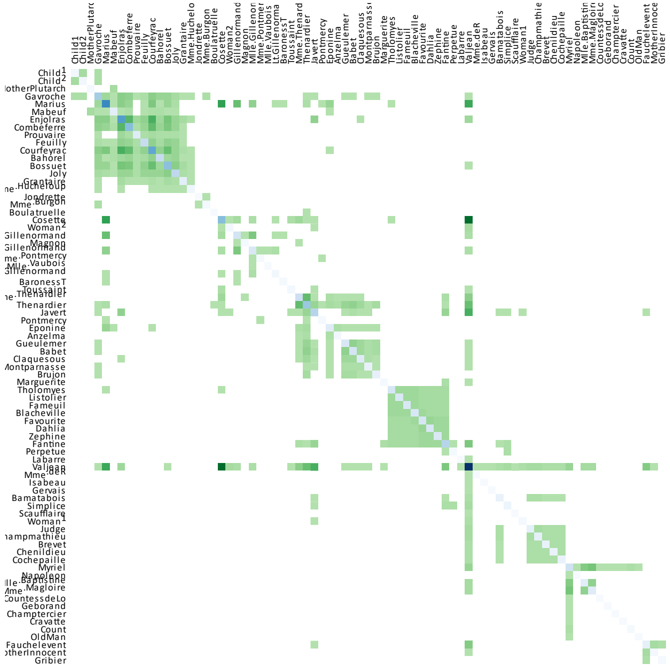
- **Important:** Ordering of rows and columns affects visibility of patterns
- Details on algorithms for matrix re-ordering are described by [Behrisch et al.]

Ordered by name



Ordered by frequency

Ordered by cluster



Demo at https://bost.ocks.org/mike/miserables/

Matrix representations

Summary

- Advantages
  - Focus on edges of graphs
  - No complex layout computations necessary
- Disadvantages
  - Quadratic space requirements
  - Intricate ordering problem

Basic visual representations

Four fundamental visual designs

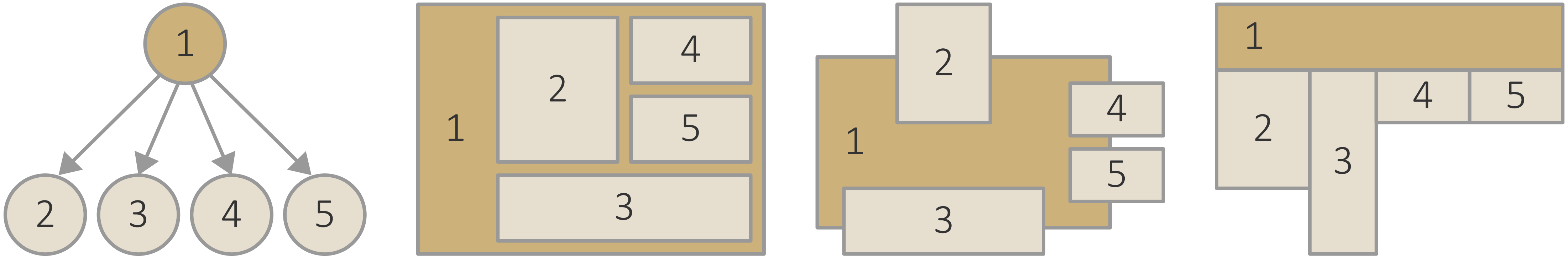
- Node-link representations
- Matrix representations
- Implicit representations
- Hybrid representations

Implicit representations

- Node-link and matrix representations show edges **explicitly** via dedicated graphical primitives
- Implicit representations of graphs lack explicitly drawn edges and instead encode them **implicitly** by the relative position of nodes
- Implicit representation work well only if graphs follow some regularities
  - Hardly applicable to general graphs
  - Well suited for trees with regular parent-child relationships

Implicit representations

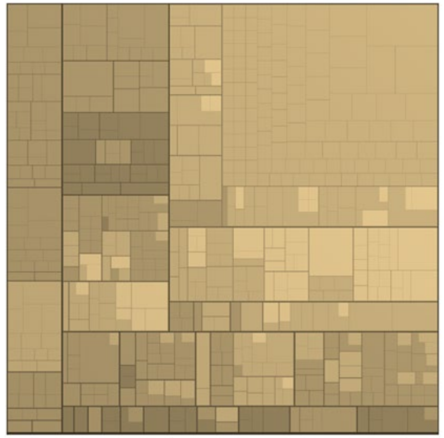
- Basic options for implicit encoding
  - **Inclusion** (children included in parent)
  - **Overlap** (children overlap parent)
  - **Adjacency** (children adjacent to parent)



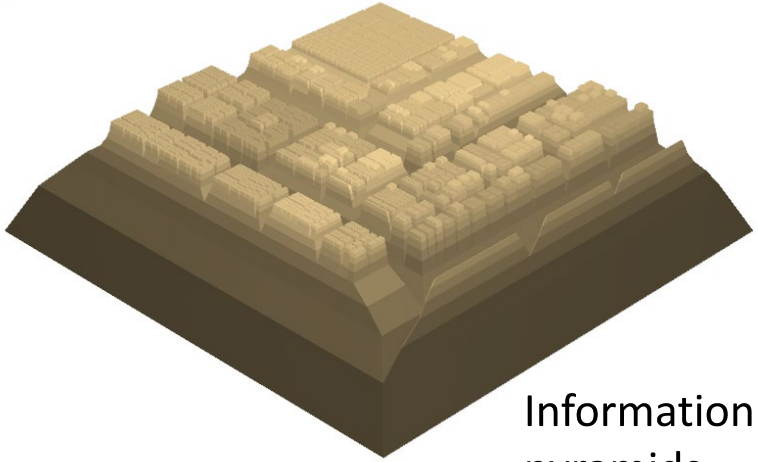
Node-link Inclusion Overlap Adjacency

Implicit representations

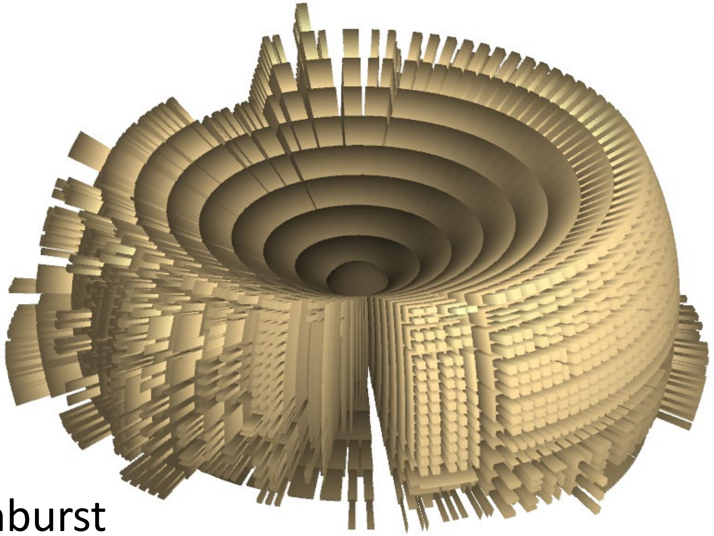
- Many different tree visualizations based on implicit encoding
- Examples
  - Treemap
  - Information pyramids
  - 3D sunburst
  - Many more examples at https://treevis.net
  - Think about: What type of implicit encoding is used in the examples?



Treemap



Information pyramids



3D sunburst

Implicit representations

Summary

- Advantages
  - Focus on nodes
  - Use display space efficiently
- Disadvantages
  - Edge visibility reduced
  - Node visibility may be compromised

Basic visual representations

Four fundamental visual designs

- Node-link representations
- Matrix representations
- Implicit representations
- Hybrid representations

Hybrid representations

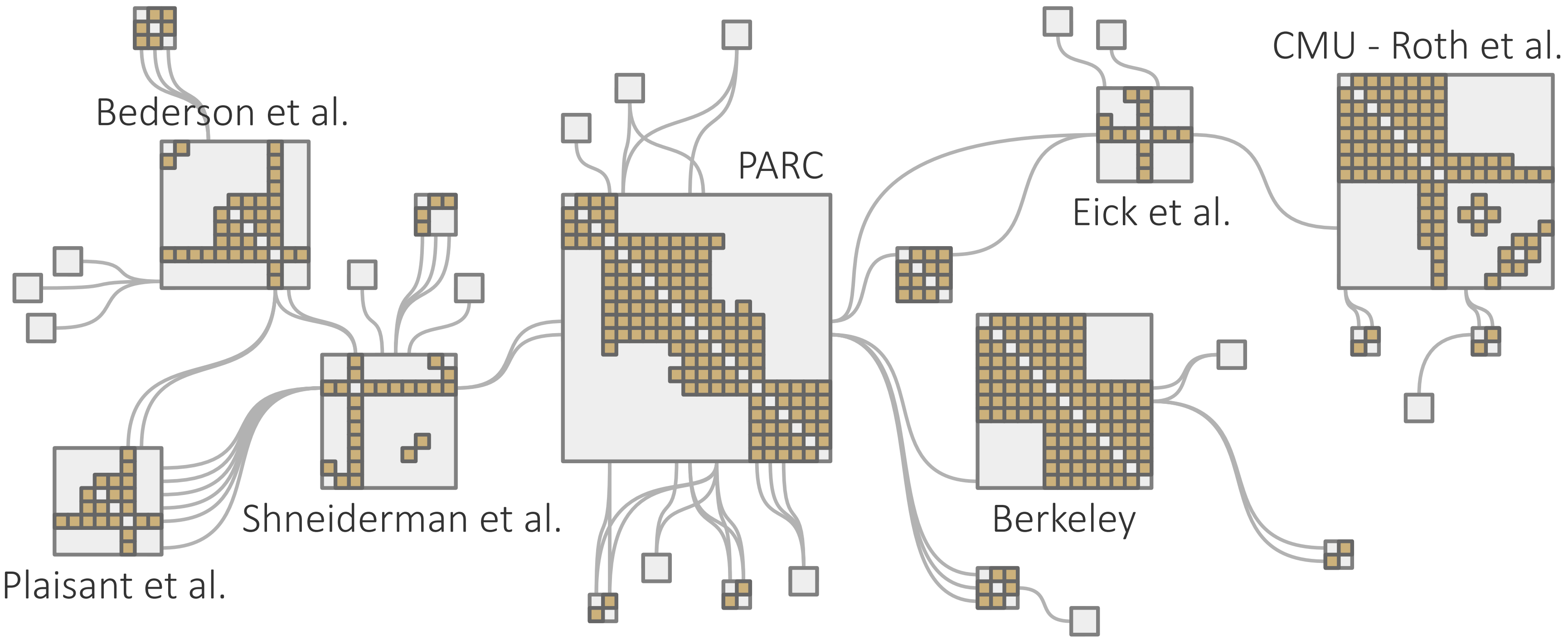
- Combine previous approaches to exploit individual advantages
  - Sparse subgraphs shown via node-link
  - Dense subgraphs shown as matrix
  - Tree-like substructures shown via implicit representations
- Example: **NodeTrix**
  - Sparse parts: node-link representation
  - Dense parts: matrices
  - Dynamic transformation between node-link and matrix representation

> “Each feature is drawn with an algorithm tuned for its topology.
>
> — Archambault et al., 2009

Hybrid representations

- **NodeTrix** (Henry et al., 2007)

Demo video at https://www.youtube.com/watch?v=7G3MxyOcHKQ



Basic visual representations

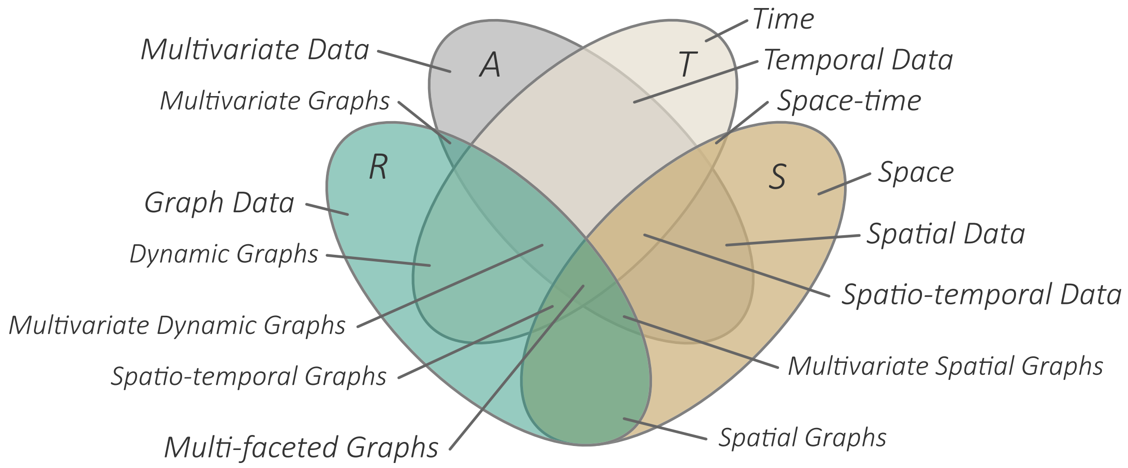
Summary

- Node-link representation
- Matrix representation
- Implicit representation
- Hybrid representation

- Next: Visualization of multi-faceted graphs

Visualization of multi-faceted graphs

- So far, primarily focused on graph structure
- Now, we add additional data facets:
  - Data attributes A
  - Time T
  - Space S



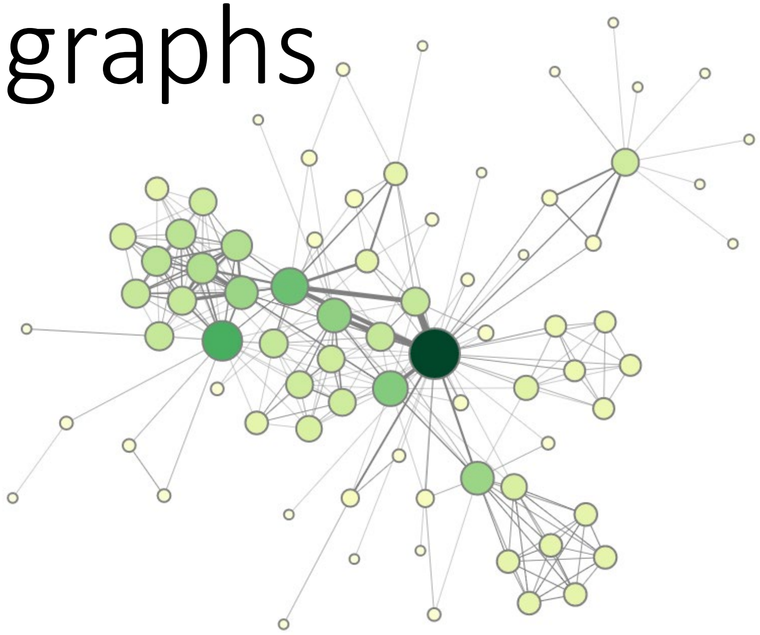
- Two-step design process
  1. Decide primary facet whose depiction governs the overall visualization
  2. Incorporate additional facets into based representation from 1.

Visualization of multi-faceted graphs

- Two-step design process
  1. Decide primary facet whose depiction governs the overall visualization
  2. Incorporate additional facets into based representation from 1.

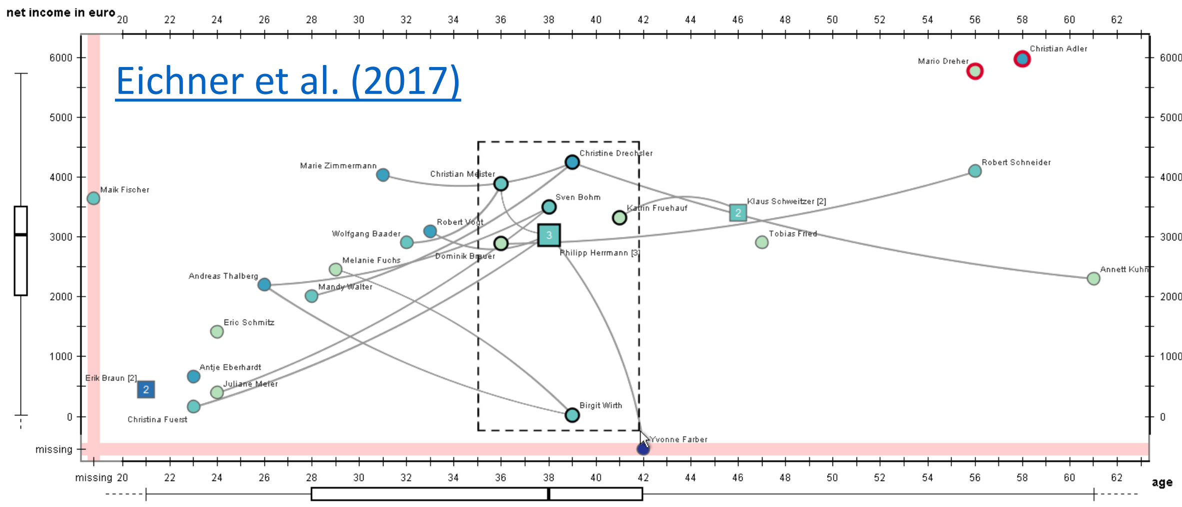
- Examples:
  - R is primary, A is secondary → Topology-driven layout
  - A is primary, R is secondary → Attribute-driven layout

Visualization of multi-faceted graphs



- **Topology-driven layout**
  - R is primary, A is secondary
  - Nodes positioned according to graph structure (e.g., force-directed layout)
  - Attributes encoded via color, size, and width

- **Attribute-driven layout**
  - A is primary, R is secondary
  - Nodes positioned according to attribute values (e.g., scatterplot)
  - Edges represented as links as usual

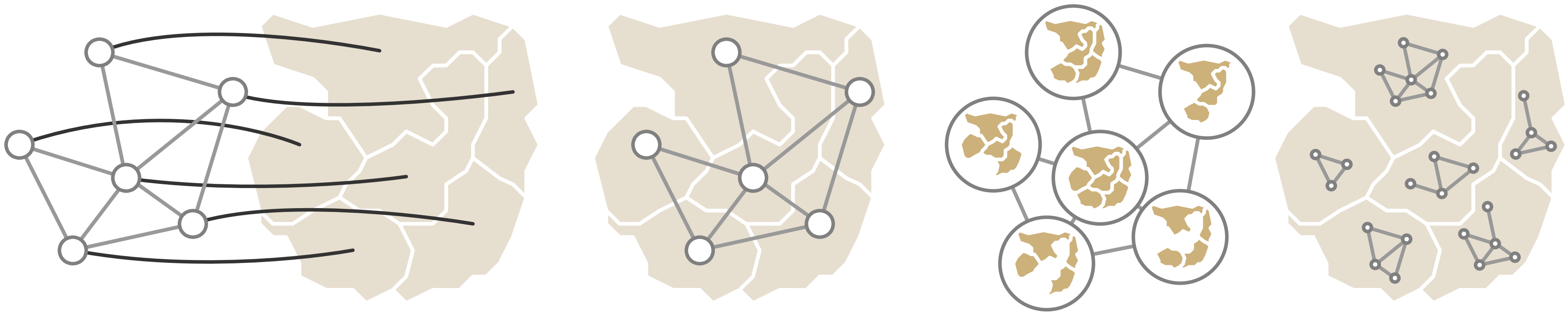


[Eichner et al. (2017)]

Visualization of multi-faceted graphs

- General composition strategies for multiple facets
  - **Temporal composition**
    - Show one data facet after the other
  - **Spatial composition**
    - Show spatial arrangement of multiple facets

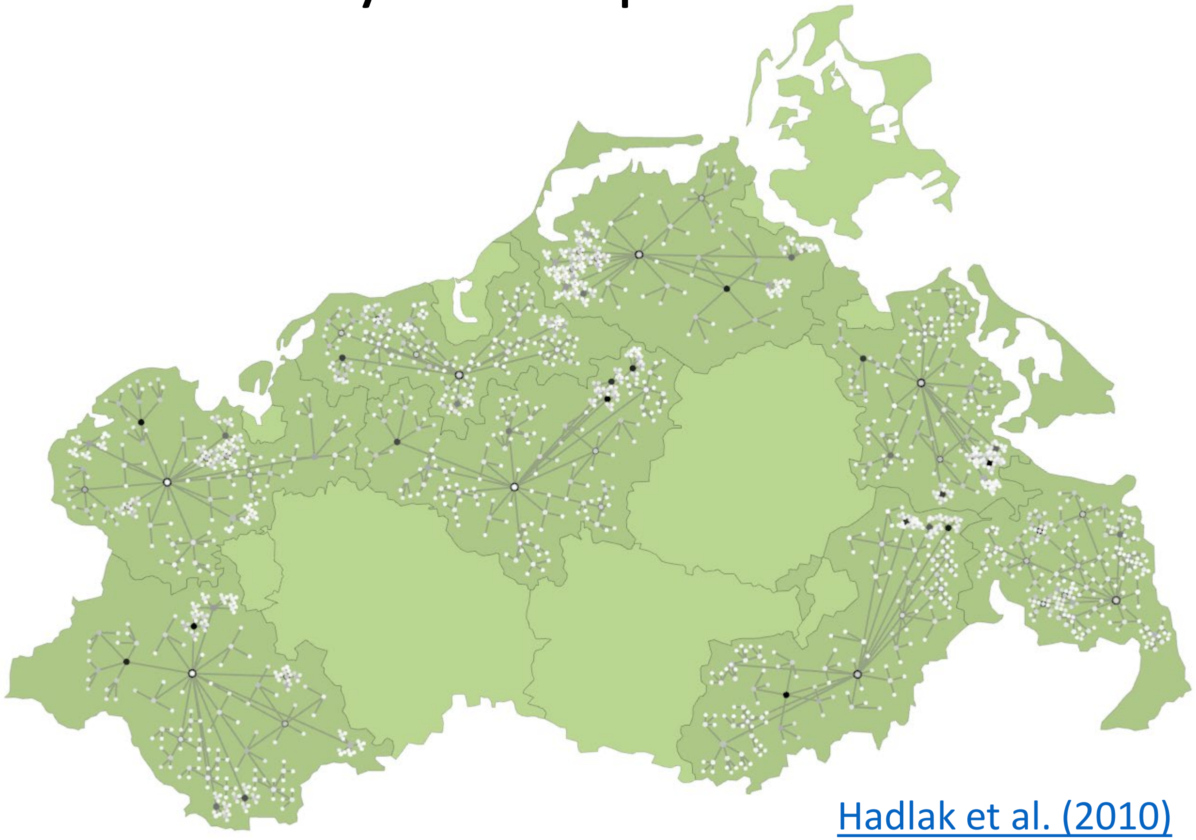
Recall **presenting multiple views in space or in time** from Lecture 4.



Juxtaposition   Superposition   Nesting

Visualization of multi-faceted graphs

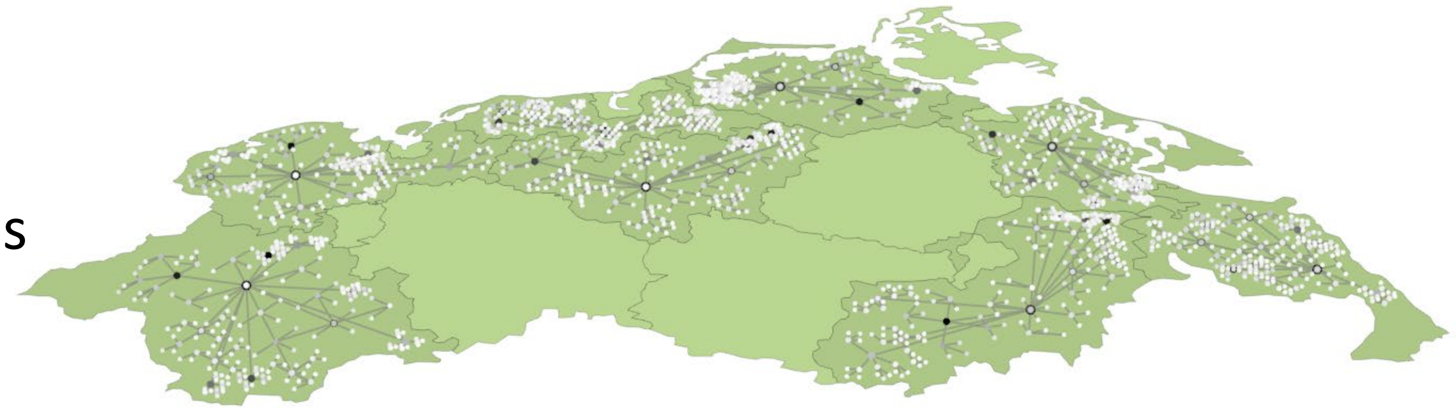
- Example: 3D layered map visualization of dynamic spatial trees
  - Graph facets: R, S, T
  - S represented as map
  - R nested into map regions via point-based layout



Hadlak et al. (2010)

Visualization of multi-faceted graphs

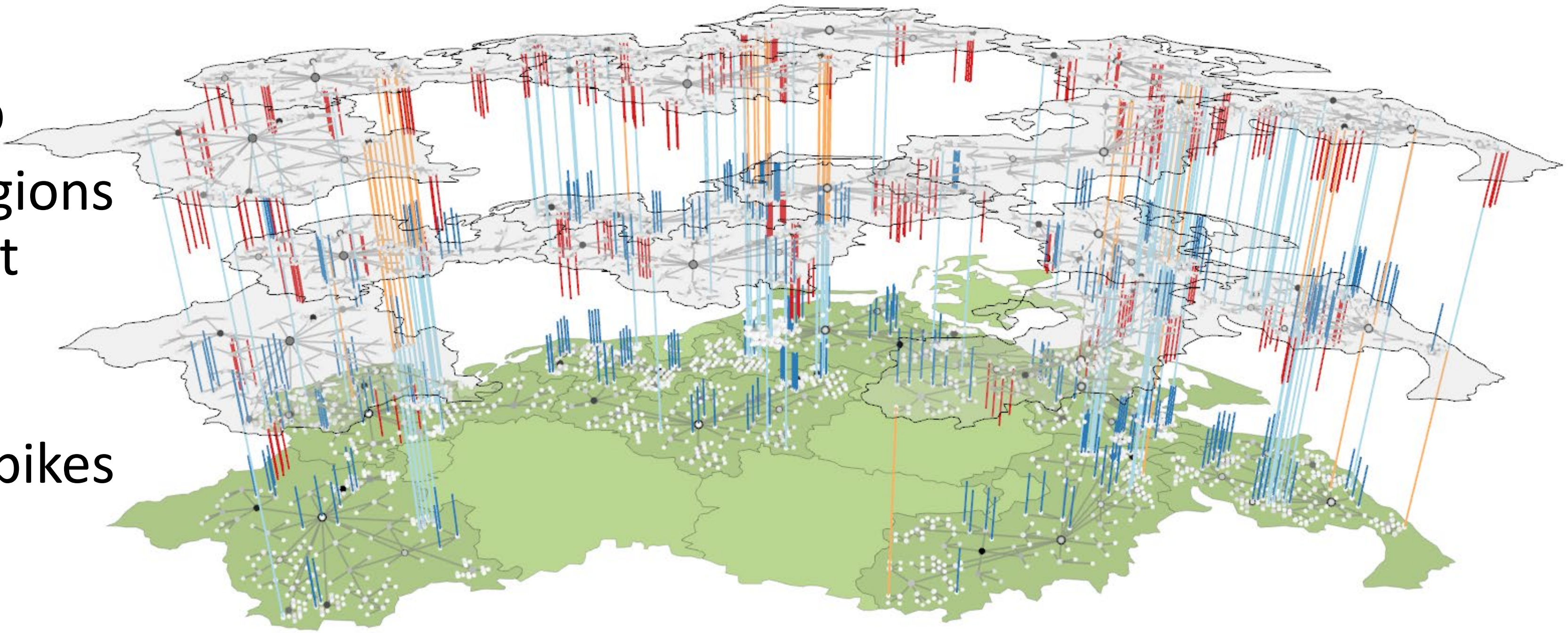
- Example: 3D layered map visualization of dynamic spatial trees
  - Graph facets: R, S, T
  - S represented as map
  - R nested into map regions via point-based layout
  - T represented as stacked 3D layers and encoded as colored spikes



Hadlak et al. (2010)

Visualization of multi-faceted graphs

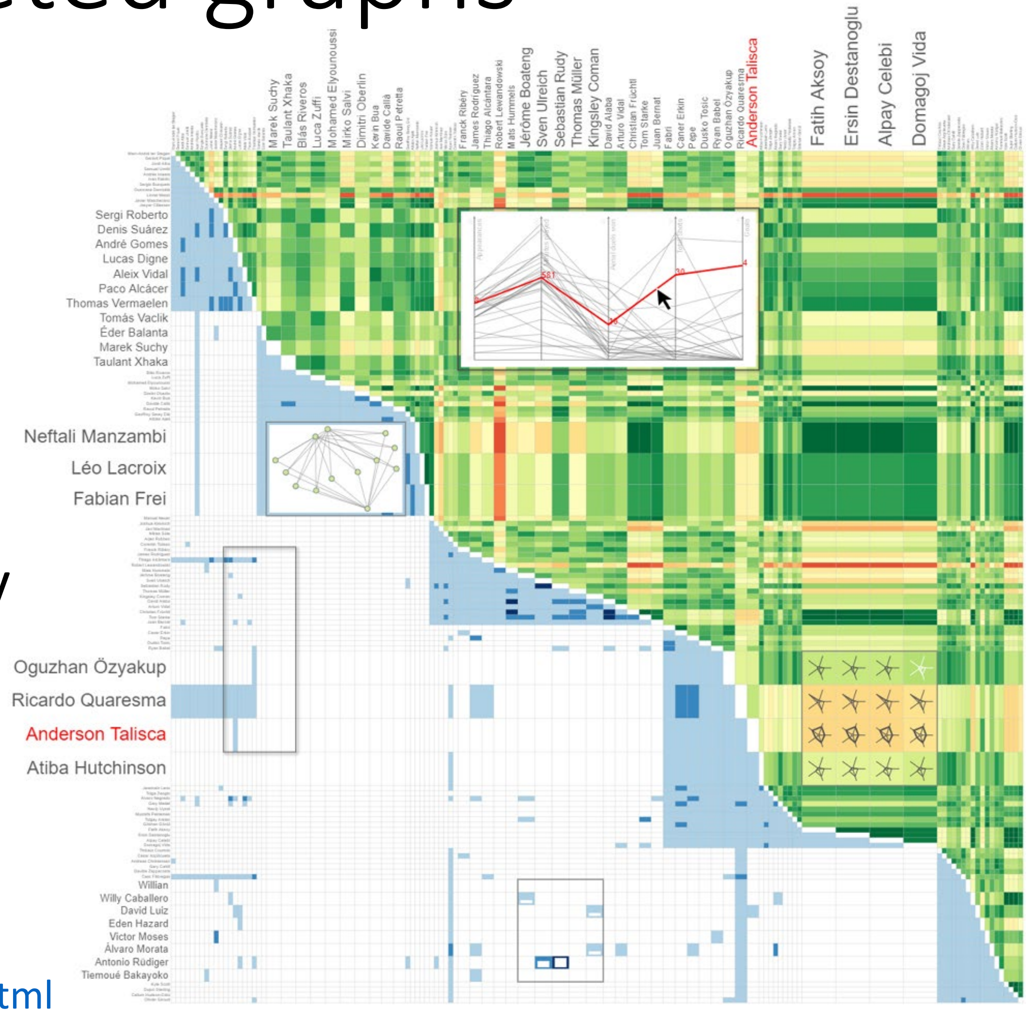
- Example: 3D layered map visualization of dynamic spatial trees
  - Graph facets: R, S, T
  - S represented as map
  - R nested into map regions via point-based layout
  - T represented as stacked 3D layers and encoded as colored spikes



Hadlak et al. (2010)

Visualization of multi-faceted graphs

- Example: Responsive Matrix Cells
  - Graph facets: R, A
  - R represented in lower triangular matrix
  - Overview of A color-coded in upper triangular matrix as aggregated similarity
  - Details of A represented in nested views in matrix cells
  - Nested views respond to size changes



https://vcg.informatik.uni-rostock.de/~ct/software/RMC/index.html

Horak et al. (2021)

Visualization of multi-faceted graphs

- Example: Multi-view visualization of multivariate compound graphs
  - Graph facets: R, A, H (compound hierarchy)
  - Juxtaposition of data facets in multiple views
  - Individual views may use superposition
  - Each view focuses on particular facet
  - Sum of all views covers all facets

Visualization of multi-faceted graphs

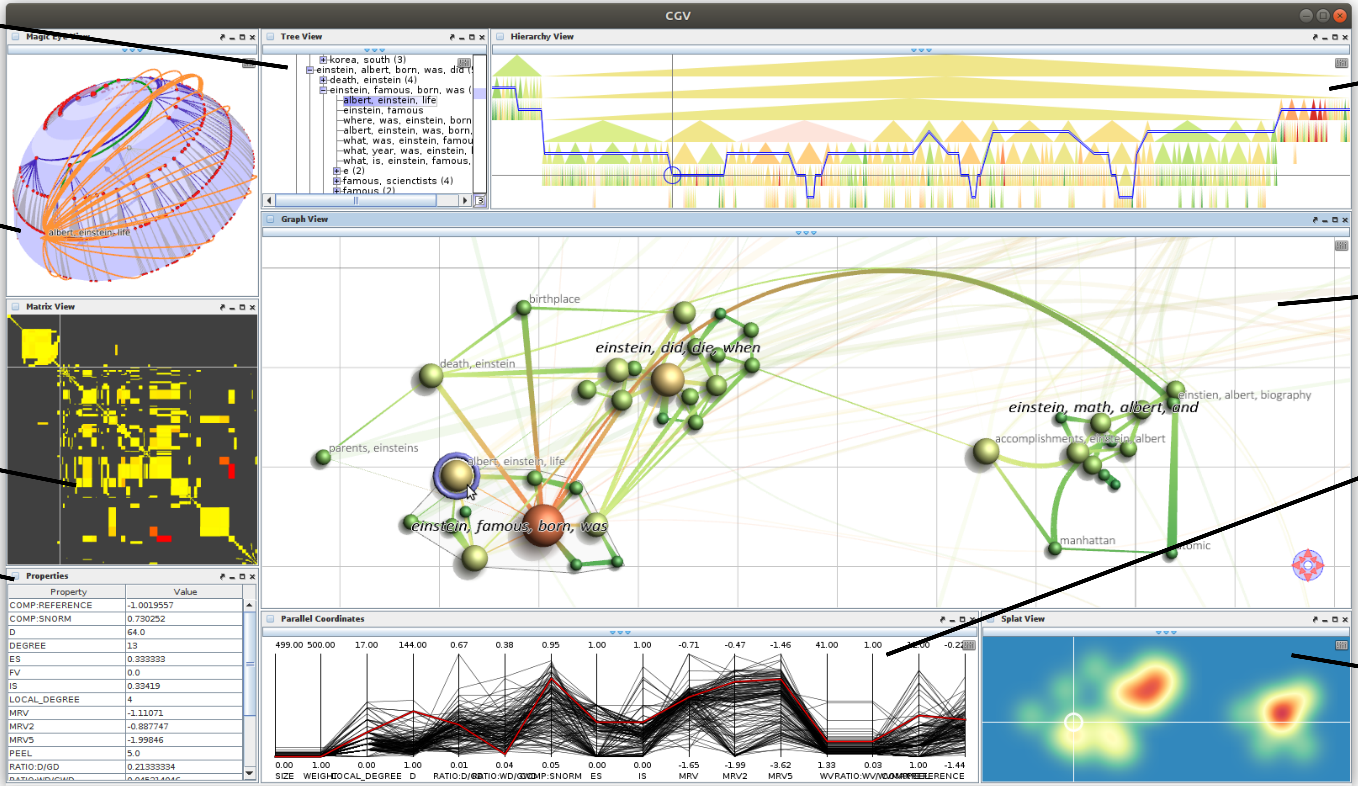
- Example: Multi-view visualization of multivariate compound graphs

Fisheye treeview representing textual labels of the nodes of R

Magic Eye View showing H and selected edges from R

Matrix view focusing on the edges of R and a selected edge attribute A

Property view listing A for a selected node



Tree view showing H and selected attribute A

Graph layout of R with selected attributes A encoded by size and color

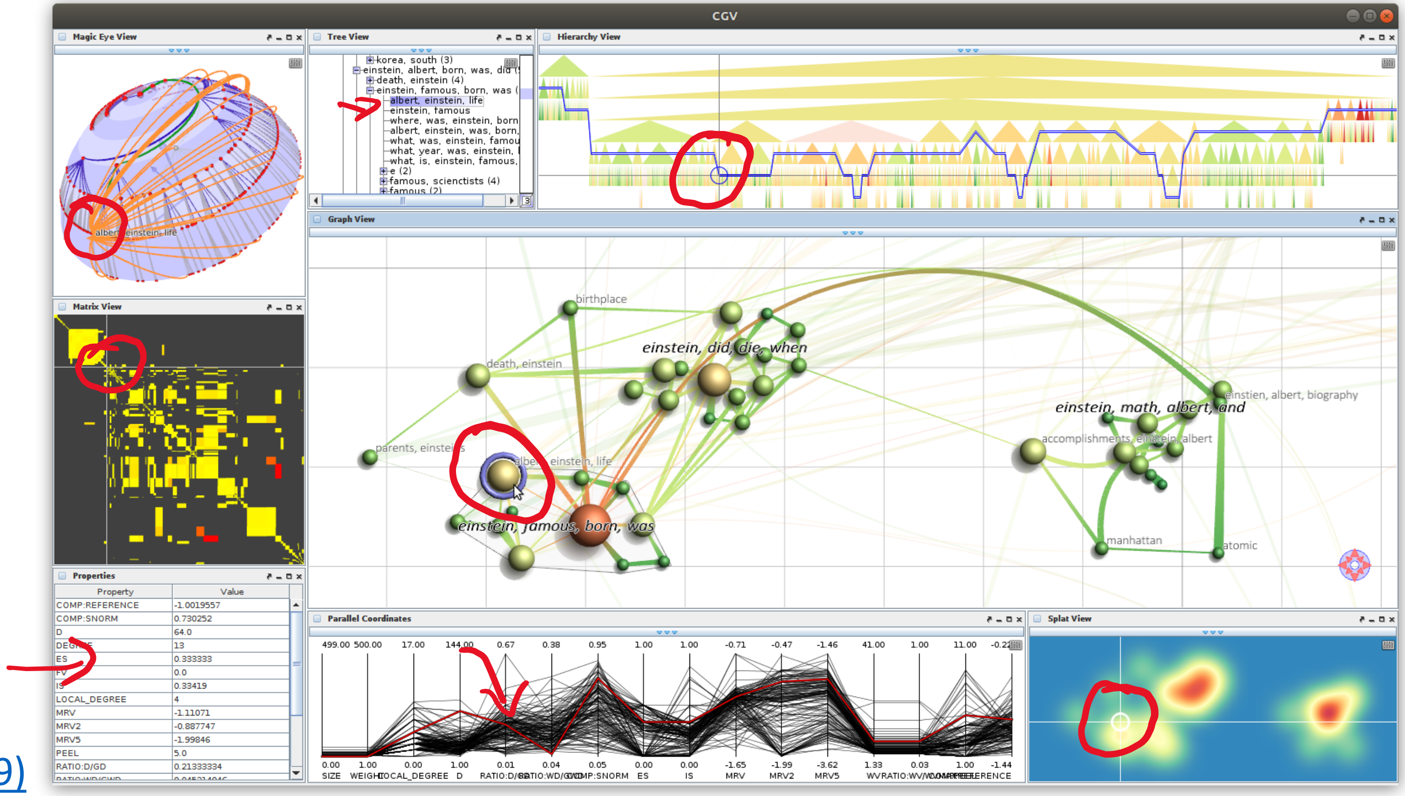
Parallel coordinates view showing attributes A of nodes or edges in detail

Splat view visualizing the density of graph elements in R

Tominski et al. (2009)

Visualization of multi-faceted graphs

- Example: Multi-view visualization of multivariate compound graphs
  - In order to make sense of multi-views visualization, it is necessary to mentally connect the different representations
  - This can be supported by **coordinated highlighting**
  - Focused graph elements are consistently marked in all views



Tominski et al. (2009)

Summary

- Brief introduction to graph data and different facets of graphs
- Basic visual representations
  - Node-link
  - Matrix
  - Implicit
  - Hybrid
- Methods for visualizing multi-faceted graphs
  - Decide primary facet to be communicated
  - Embed secondary facets

Assignments

1. Read about "The Aesthetics of Graph Visualization" in Bennett et al., (2007)!
2. Read about Eades, Fruchterman & Reingold, and Kamada & Kawai algorithms in Sections 12.1, 12.2, and 12.4 from Kobourov's chapter on "Force-Directed Drawing Algorithms"!
3. Browse the comprehensive survey on "The State of the Art in Visualizing Multivariate Networks" by Nobre et al. (2019)!
4. Download Gephi from https://gephi.org/ and create some graph visualizations with different layouts!
5. Try different layout styles of yEd at https://www.yworks.com/yed-live/!

Questions

1. How are graphs defined, what graph objects can be relevant to analyze?
2. What data facets can play a role when visualizing graphs?
3. Name relevant analytical questions for interactive visual graph analysis!
4. What are the four fundamental visual representations for graphs?
5. Outline the basic procedure of force-directed graph layout!
6. Explain the basic idea of matrix representations!
7. Draw a small graph as node-link and matrix representation!
8. What is the difference between explicit and implicit representations?
9. What is a NodeTrix visualization?
10. What are general composition strategies for multi-faceted graphs?
11. What is the difference between topology-driven layouts and attribute-driven layouts, and what are they good for?