Ch 8/9: Spatial Data, Networks
Paper: Genealogical Graphs
Paper: ABysS-Explorer

Tamara Munzner
Department of Computer Science
University of British Columbia

CPSC 547, Information Visualization
Week 6: 17 October 2017

www.cs.ubc.ca/~tmm/courses/547-17F
News

• marks for previous 2 weeks published
  – first week was pass/fail for having anything
  – now more fine-grained guidance about expectations with comments
    • if you didn’t get full credit
  – in general: don’t just summarize

• today
  – pitches first
  – Q&A, lecture second
Ch 8: Arrange Spatial Data
Arrange spatial data

→ **Use Given**

→ **Geometry**
  → *Geographic*
  → *Other Derived*

→ **Spatial Fields**

→ *Scalar Fields (one value per cell)*
  → *Isocontours*
  → *Direct Volume Rendering*

→ *Vector and Tensor Fields (many values per cell)*
  → *Flow Glyphs (local)*
  → *Geometric (sparse seeds)*
  → *Textures (dense seeds)*
  → *Features (globally derived)*
Idiom: **choropleth map**

- **use** given spatial data
  - when central task is understanding spatial relationships

- **data**
  - geographic geometry
  - table with 1 quant attribute per region

- **encoding**
  - use given geometry for area mark boundaries
  - sequential segmented colormap [*more later*]
  -(geographic heat map)

http://bl.ocks.org/mbostock/4060606
Population maps trickiness

• beware!
• absolute vs relative again
  • population density vs per capita
• investigate with Ben Jones Tableau Public demo
  • http://public.tableau.com/profile/ben.jones#!/vizhome/PopVsFin/PopVsFin

Are Maps of Financial Variables just Population Maps?
  • yes, unless you look at per capita (relative) numbers

[ https://xkcd.com/1138 ]
Idiom: Bayesian surprise maps

• use models of expectations to highlight surprising values
• confounds (population) and variance (sparsity)

[Surprise! Bayesian Weighting for De-Biasing Thematic Maps. Correll and Heer. Proc InfoVis 2016]
Idiom: **topographic map**

- **data**
  - geographic geometry
  - scalar spatial field
    - 1 quant attribute per grid cell
- **derived data**
  - isoline geometry
    - isocontours computed for specific levels of scalar values

*Land Information New Zealand Data Service*
Idioms: **isosurfaces, direct volume rendering**

- **data**
  - scalar spatial field
    - 1 quant attribute per grid cell
- **task**
  - shape understanding, spatial relationships
- **isosurface**
  - derived data: isocontours computed for specific levels of scalar values
- **direct volume rendering**
  - transfer function maps scalar values to color, opacity

---


Vector and tensor fields

• data
  – many attribs per cell

• idiom families
  – flow glyphs
    • purely local
  – geometric flow
    • derived data from tracing particle trajectories
    • sparse set of seed points
  – texture flow
    • derived data, dense seeds
  – feature flow
    • global computation to detect features
      – encoded with one of methods above


Vector fields

• empirical study tasks
  – finding critical points, identifying their types
  – identifying what type of critical point is at a specific location
  – predicting where a particle starting at a specified point will end up (advection)


Idiom: similarity-clustered streamlines

• data
  – 3D vector field

• derived data (from field)
  – streamlines: trajectory particle will follow

• derived data (per streamline)
  – curvature, torsion, tortuosity
  – signature: complex weighted combination
  – compute cluster hierarchy across all signatures
  – encode: color and opacity by cluster

• tasks
  – find features, query shape

• scalability
  – millions of samples, hundreds of streamlines

Ch 9: Arrange Network Data
Arrange networks and trees

- **Node–Link Diagrams**
  - Connection Marks
  - Connection Marks
  - Connection Marks

- **Adjacency Matrix**
  - Derived Table
  - Derived Table
  - Derived Table

- **Enclosure**
  - Containment Marks
  - Containment Marks
  - Containment Marks
**Idiom: force-directed placement**

- visual encoding
  - link connection marks, node point marks
- considerations
  - spatial position: no meaning directly encoded
    - left free to minimize crossings
  - proximity semantics?
    - sometimes meaningful
    - sometimes arbitrary, artifact of layout algorithm
    - tension with length
      - long edges more visually salient than short
- tasks
  - explore topology; locate paths, clusters
- scalability
  - node/edge density $E < 4N$

[Link to visualization](http://mbostock.github.com/d3/ex/force.html)
Idiom: **sfdp** (multi-level force-directed placement)

- **data**
  - original: network
  - derived: cluster hierarchy atop it

- **considerations**
  - better algorithm for same encoding technique
    - same: fundamental use of space
    - hierarchy used for algorithm speed/quality but not shown explicitly
    - (more on algorithm vs encoding in afternoon)

- **scalability**
  - nodes, edges: 1K-10K
  - hairball problem eventually hits


Idiom: **adjacency matrix view**

- **data: network**
  - transform into same data/encoding as heatmap
- **derived data: table from network**
  - 1 quant attrib
    - weighted edge between nodes
  - 2 categ attribs: node list x 2
- **visual encoding**
  - cell shows presence/absence of edge
- **scalability**
  - 1K nodes, 1M edges

---

Figure 7.5: Comparing matrix and node-link views of a five-node network.

(a) Matrix view. (b) Node-link view. From [Henry et al. 07], Figure 3b and 3a.

(Permission needed.)

Matrix views of networks can achieve very high information density, up to a limit of one thousand nodes and one million edges, just like cluster heatmaps and all other matrix views that uses small area marks.

For undirected networks where links are symmetric, only half of the matrix needs to be shown, above or below the diagonal, because a link from node A to node B necessarily implies a link from B to A. For directed networks, the full square matrix has meaning, because links can be asymmetric. Figure 7.5 shows a simple example of an undirected network, with a matrix view of the five-node dataset in Figure 7.5a and a corresponding node-link view in Figure 7.5b.

Matrix views of networks can achieve very high information density, up to a limit of one thousand nodes and one million edges, just like cluster heatmaps and all other matrix views that uses small area marks.

---


Connection vs. adjacency comparison

- adjacency matrix strengths
  - predictability, scalability, supports reordering
  - some topology tasks trainable
- node-link diagram strengths
  - topology understanding, path tracing
  - intuitive, no training needed
- empirical study
  - node-link best for small networks
  - matrix best for large networks
  - if tasks don’t involve topological structure!

Idiom: **radial node-link tree**

- **data**
  - tree

- **encoding**
  - link connection marks
  - point node marks
  - radial axis orientation
    - angular proximity: siblings
    - distance from center: depth in tree

- **tasks**
  - understanding topology, following paths

- **scalability**
  - 1K - 10K nodes

Idiom: **treemap**

- **data**
  - tree
  - 1 quant attrib at leaf nodes
- **encoding**
  - area containment marks for hierarchical structure
  - rectilinear orientation
  - size encodes quant attrib
- **tasks**
  - query attribute at leaf nodes
- **scalability**
  - 1M leaf nodes

Link marks: Connection and containment

• marks as links (vs. nodes)
  – common case in network drawing
  – 1D case: connection
    • ex: all node-link diagrams
    • emphasizes topology, path tracing
    • networks and trees
  – 2D case: containment
    • ex: all treemap variants
    • emphasizes attribute values at leaves (size coding)
    • only trees

Tree drawing idioms comparison

- data shown
  - link relationships
  - tree depth
  - sibling order

- design choices
  - connection vs containment link marks
  - rectilinear vs radial layout
  - spatial position channels

- considerations
  - redundant? arbitrary?
  - information density?
  - avoid wasting space

Paper: Genealogical Graphs
Genealogical graphs: Technique paper

• family tree is a misnomer
  – single person has tree of ancestors, tree of descendants
  – pedigree collapse inevitable
    • diamond in ancestor graph

• crowding problem
  – exponential

• fractal layout
  – poor info density
  – no spatial ordering for generations

Layouts

- rooted trees: standard layouts
  - connection
  - containment
  - adjacent aligned position
  - indented position

Layouts

• free trees
  – no root

• adapting rooted methods
  – temporary root for given focus
  – containment (nested)

Dual trees abstraction

• explore canonical subsets and combinations, easy to interpret, scales well
• no crossings, nodes ordered by generation
• doubly rooted: x leftmost descend, y rightmost ancestor
  – offset roots from hourglass diagram

Another example

• vertical connection
• horizontal connection
• indented

• upcoming chapters
  – layering
  – aggregation

Interaction as fundamental to design

• navigation
  – topological navigation via collapse/expand on selection
    • parents, children
    • expand can trigger rotation
      – collapsing others
      – layout driven by navigation
  – geometric zoom/pan
  – constrained navigation: automatic camera framing

• animated transitions
  – 3 phases: fade out, move, fade in

• mouseover hover
  – preview dots: expand if collapsed

Custom widget

- popup marking menu
  - flick up or down, ballistic
  - subtree drag-out widget

Paper: ABySS-Explorer
ABySS-Explorer: Design study

- reconstructing genome with ABySS algorithm (Assembly By Short Sequences)
- domain task
  - go from short subsequences to contigs, long contiguous sequences
  - extensive automatic support, but still human in the loop for visual inspection and manual editing
  - ambiguities, like repetitions longer than read length
- data, domain: abstract
  - millions of reads of 25-100 nucleotides (nt): strings
  - read coverage, proxy for quality: quant attrib
  - read pairing distances, proxy for size distribution: quant

Contigs: abstraction as derived network data

• derived data: de Bruijn graph/network
  – directed network, compact representation of sequence overlaps
  – node: contig
  – edge: overlap of k − 1 nt between two contigs
  – good for computing, bad for reasoning about sequence space

• derived data: dual de Bruijn graph
  – node: points of contig overlap
  – edge: contig
  – better match for arrow diagrams used in hand drawn sketches

• base layout: force-directed

DNA as double stranded: idiom for encoding & interaction

• rejected option: 2 nodes per contig
  – excess clutter if one for each direction
  – choice at data abstraction level

• encoding & interaction idiom: polar node
  – encoding: upper vs lower attachment point
    • redundant with arc direction
      – large-scale visibility, without need to zoom
    • arbitrary but consistent
  – interaction: click to reverse direction
    • switches polarity of vertex connections
    • changes sign of label

Contig length: encoding

• rejected option: scale edge lengths by sequence lengths
  – short contigs are important sources of ambiguity, would be hard to distinguish
  – task guidance: only low-res judgements needed, relatively long or short

• encoding idiom: wave pattern
  – oscillation shows fixed number, shapes distinguishable
  – min amplitude at connections so edges visible
  – orientation with max amplitude asymmetric wrt start
  • rejected initial option: max in middle
  • rejected options:
    – color (keep for other attribute)
    – half-lines
    – curvature (used for polar nodes)
  • aligned with empirical guidance for tapered edges

Contig coverage: encoding

- rejected options: luminance/lightness
  - not distinguishable given denseness variation from wave shapes
  - also problematic with desire for separable color/hue encoding
- chosen: line thickness
  - not distinguishable for extremely long contigs
  - can address by adjusting oscillation frequency to suitable size
Read pairs: encoding

• data:
  – distance estimate
  – orientation

• encoding:
  – dashed line (shape channel for line mark)
    • implying inferred vs observed sequences
  – color for both dashed line and contig leaf
  – [same length as for contigs]
  – rejected initial option: line color alone
    • too ambiguous
  – interaction to fully resolve remaining ambiguity
  – or color by unambiguous paths in grey

Displaying meta-data

• reserve color for additional attributes
• ex: color to compare reference human to lymphoma genome
  – inconsistencies visible as interconnections between different colors
  – inversion breakpoint visible
  – interaction to check if error in metadata from experiments vs assembly
    • read pair info supports metadata
      – speedup claim vs prev work

Assembly examples

- ideal: single large contig
  - overview/gist: many small contigs remain
- interaction to resolve
  - integrate paired read highlighting on top of contig paths structure