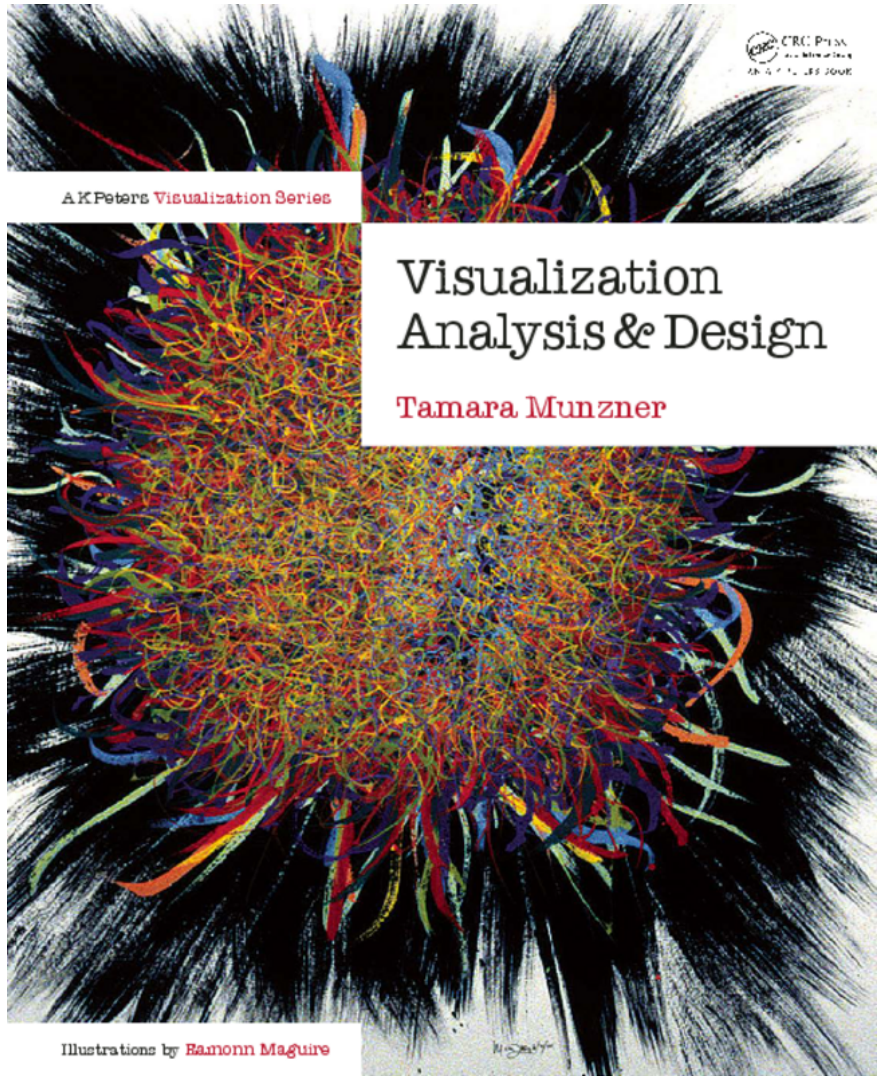
Principles of Visualization Design

- Tamara Munzner. Visualization Analysis and Design. AK Peters Visualization Series. CRC Press, 2014.
  - http://www.cs.ubc.ca/~tmm/vadbook/



Why have a human in the loop?

Computer-based visualization systems provide visual representations of datasets designed to help people carry out tasks more effectively.

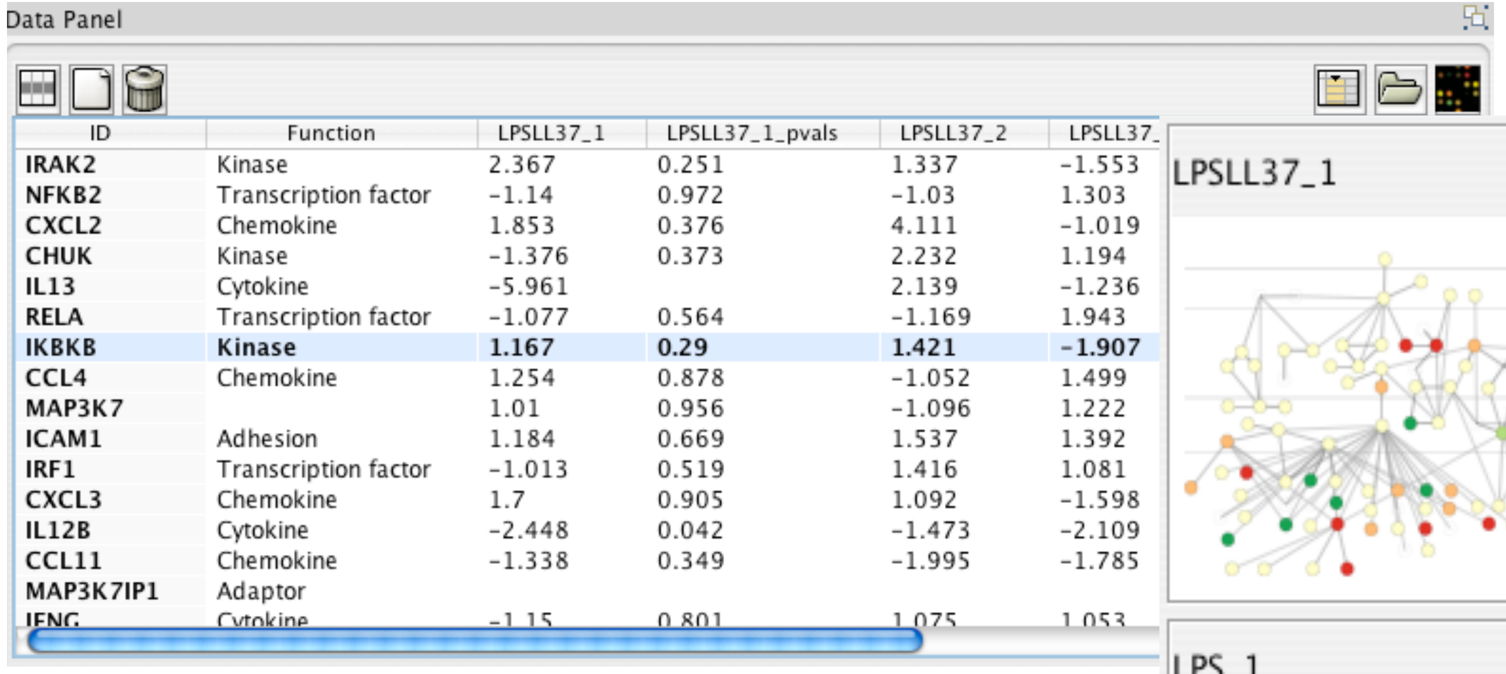
Visualization is suitable when there is a need to augment human capabilities rather than replace people with computational decision-making methods.

- don't need vis when fully automatic solution exists and is trusted
- many analysis problems ill-specified
  - don't know exactly what questions to ask in advance
- possibilities
  - long-term use for end users (e.g. exploratory analysis of scientific data)
  - presentation of known results
  - stepping stone to better understanding of requirements before developing models
  - help developers of automatic solution refine/debug, determine parameters
  - help end users of automatic solutions verify, build trust

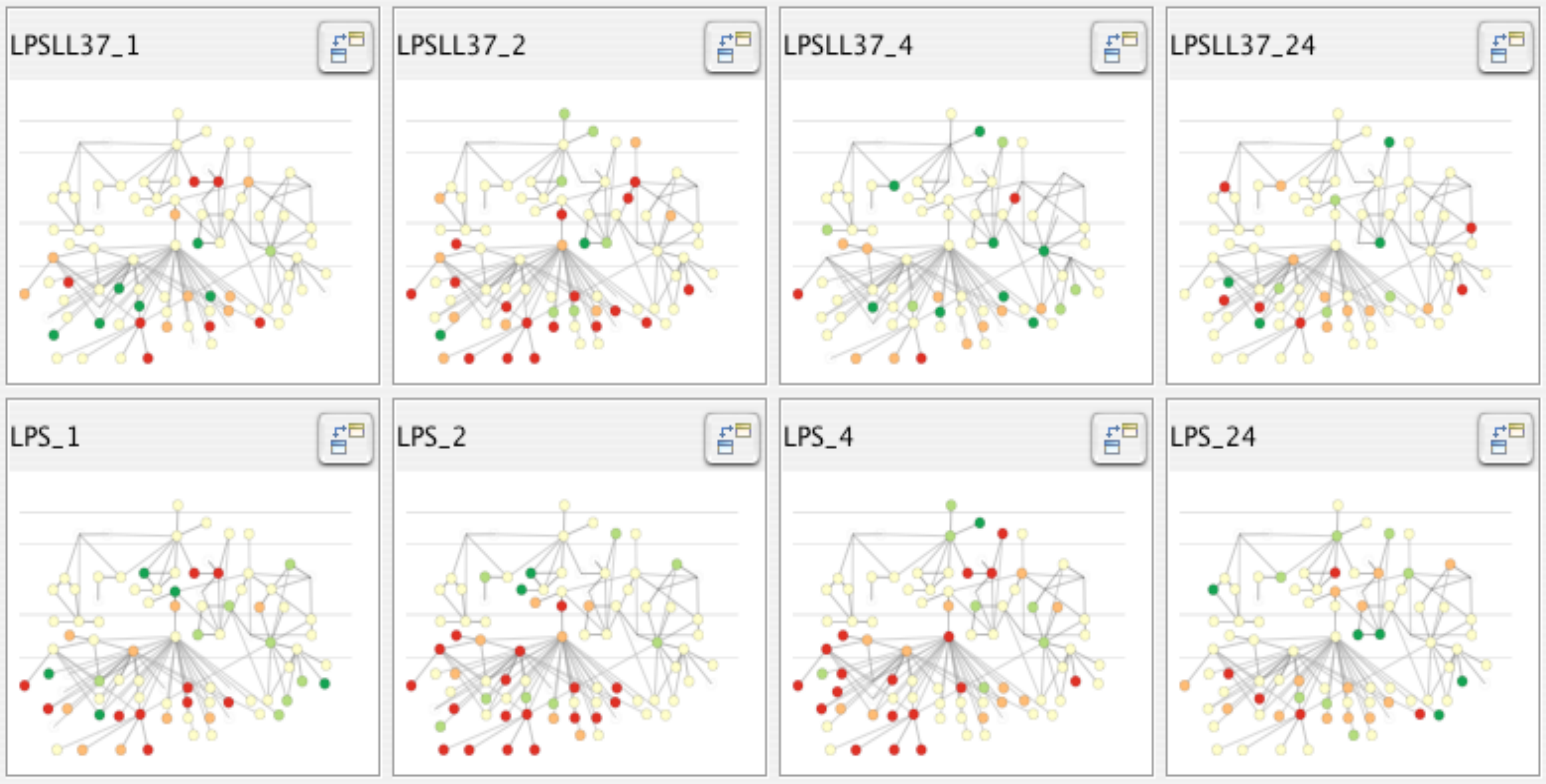
Why use an external representation?

Computer-based visualization systems provide visual representations of datasets designed to help people carry out tasks more effectively.

- external representation: replace cognition with perception



ID	Function	LPSLL37_1	LPSLL37_1_pvals	LPSLL37_2	LPSLL37_
IRAK2	Kinase	2.367	0.251	1.337	-1.553
NFKB2	Transcription factor	-1.14	0.972	-1.03	1.303
CXCL2	Chemokine	1.853	0.376	4.111	-1.019
CHUK	Kinase	-1.376	0.373	2.232	1.194
IL13	Cytokine	-5.961		2.139	-1.236
RELA	Transcription factor	-1.077	0.564	-1.169	1.943
IKBKB	Kinase	1.167	0.29	1.421	-1.907
CCL4	Chemokine	1.254	0.878	-1.052	1.499
MAP3K7		1.01	0.956	-1.096	1.222
ICAM1	Adhesion	1.184	0.669	1.537	1.392
IRF1	Transcription factor	-1.013	0.519	1.416	1.081
CXCL3	Chemokine	1.7	0.905	1.092	-1.598
IL12B	Cytokine	-2.448	0.042	-1.473	-2.109
CCL11	Chemokine	-1.338	0.349	-1.995	-1.785
MAP3K7IP1	Adaptor				
IFNG	Cytokine	-1.15	0.801	1.075	1.053



[Cerebral: Visualizing Multiple Experimental Conditions on a Graph with Biological Context. Barsky, Munzner, Gardy, and Kincaid. IEEE TVCG (Proc. InfoVis) 14(6):1253-1260, 2008.]

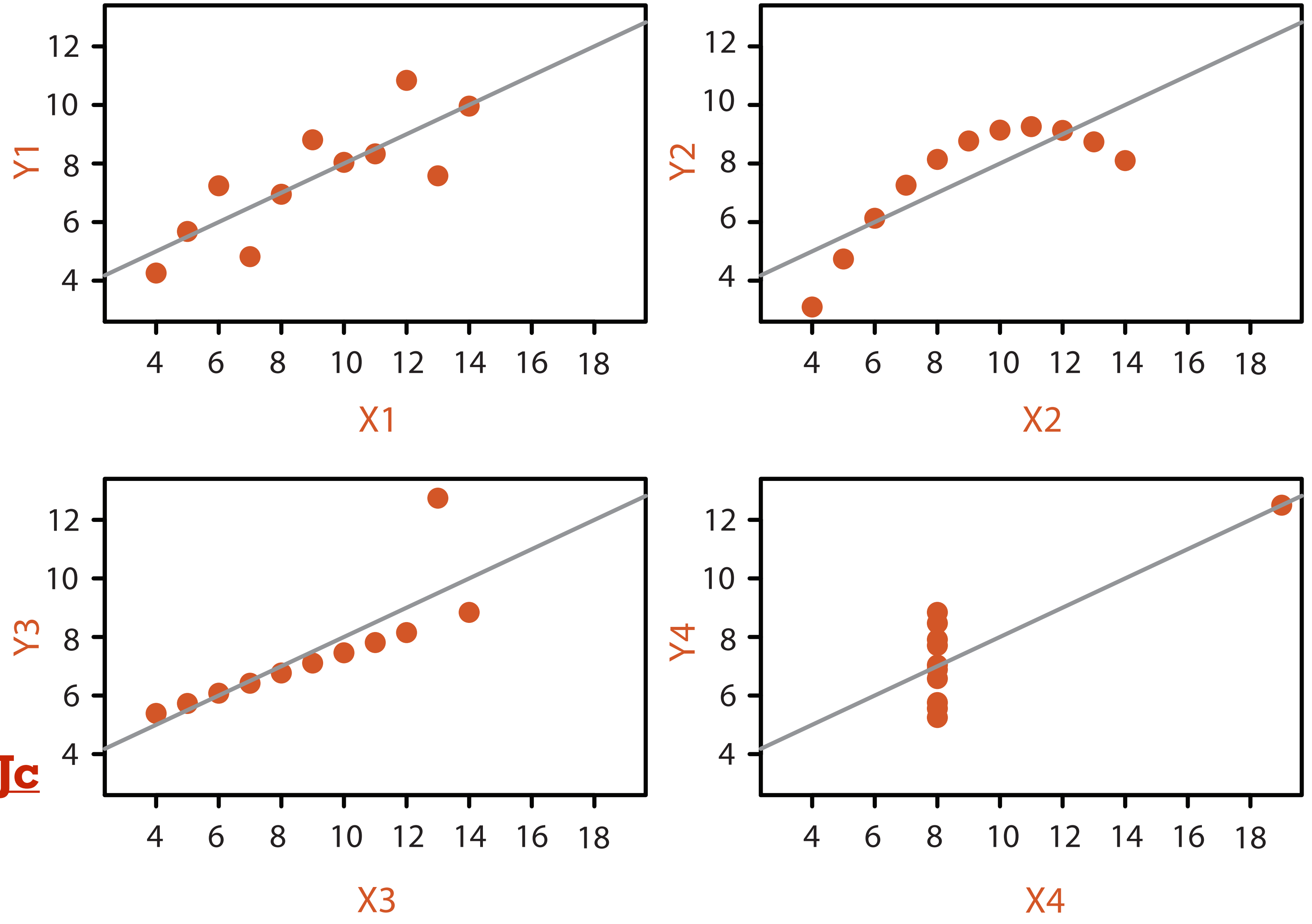
Why represent all the data?

Computer-based visualization systems provide visual representations of datasets designed to help people carry out tasks more effectively.

- summaries lose information, details matter
  - confirm expected and find unexpected patterns
  - assess validity of statistical model

Anscombe's Quartet

Identical statistics	
x mean	9
x variance	10
y mean	7.5
y variance	3.75
x/y correlation	0.816



https://www.youtube.com/watch?v=DbJyPELmhJc

Same Stats, Different Graphs

Why focus on tasks and effectiveness?

Computer-based visualization systems provide visual representations of datasets designed to help people carry out tasks more effectively.

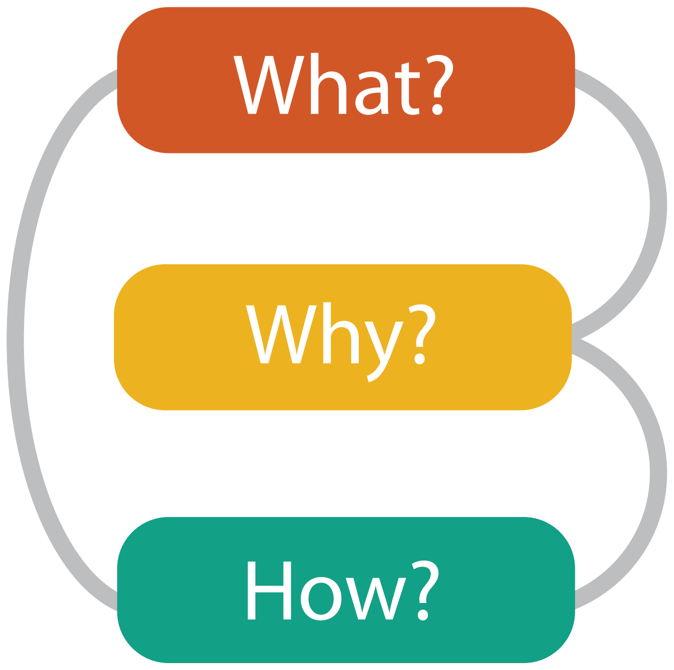
- tasks serve as constraint on design (as does data)
  - idioms do not serve all tasks equally!
  - challenge: recast tasks from domain-specific vocabulary to abstract forms
- most possibilities ineffective
  - validation is necessary, but tricky
  - increases chance of finding good solutions if you understand full space of possibilities
- what counts as effective?
  - novel: enable entirely new kinds of analysis
  - faster: speed up existing workflows

Why are there resource limitations?

Vis designers must take into account three very different kinds of resource limitations: those of computers, of humans, and of displays.

- computational limits
  - processing time
  - system memory
- human limits
  - human attention and memory
- display limits
  - pixels are precious resource, the most constrained resource
  - **information density**: ratio of space used to encode info vs unused whitespace
    - tradeoff between clutter and wasting space, find sweet spot between dense and sparse

Analysis: What, why, and how



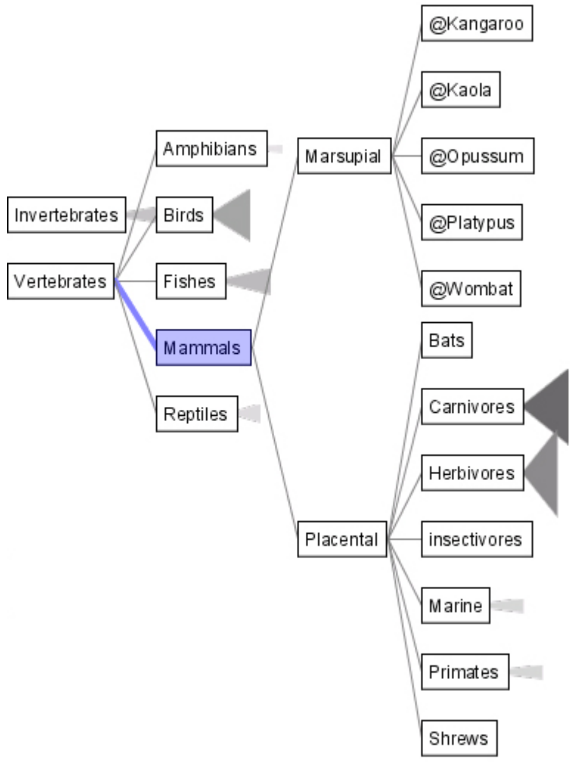
- what is shown?
  - **data** abstraction
- why is the user looking at it?
  - **task** abstraction
- how is it shown?
  - **idiom**: visual encoding and interaction

- abstract vocabulary avoids domain-specific terms
  - translation process iterative, tricky
- what-why-how analysis framework as scaffold to think systematically about design space

Why analyze?

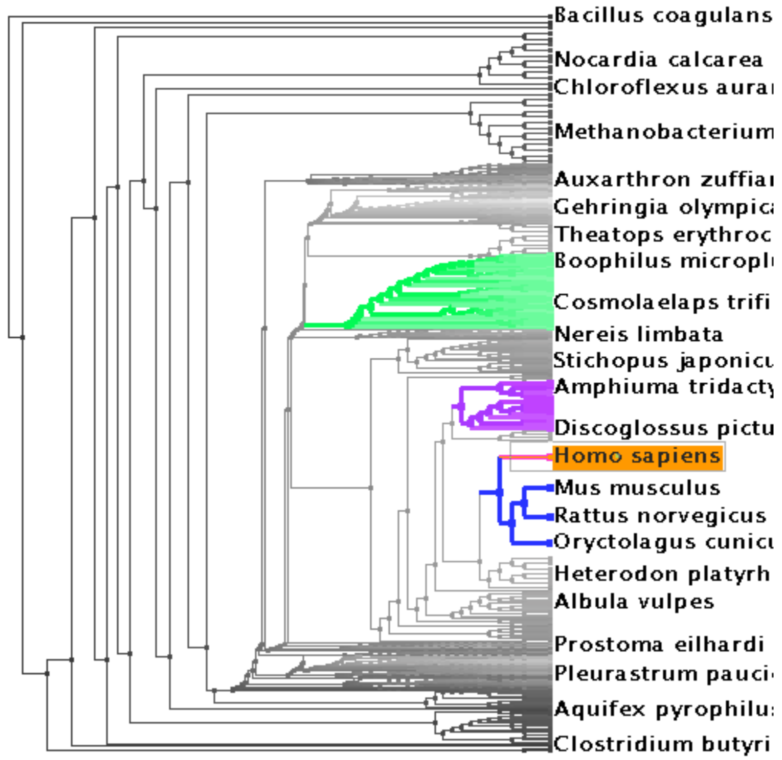
- imposes structure on huge design space
  - scaffold to help you think systematically about choices
  - analyzing existing as stepping stone to designing new
  - most possibilities ineffective for particular task/data combination

SpaceTree

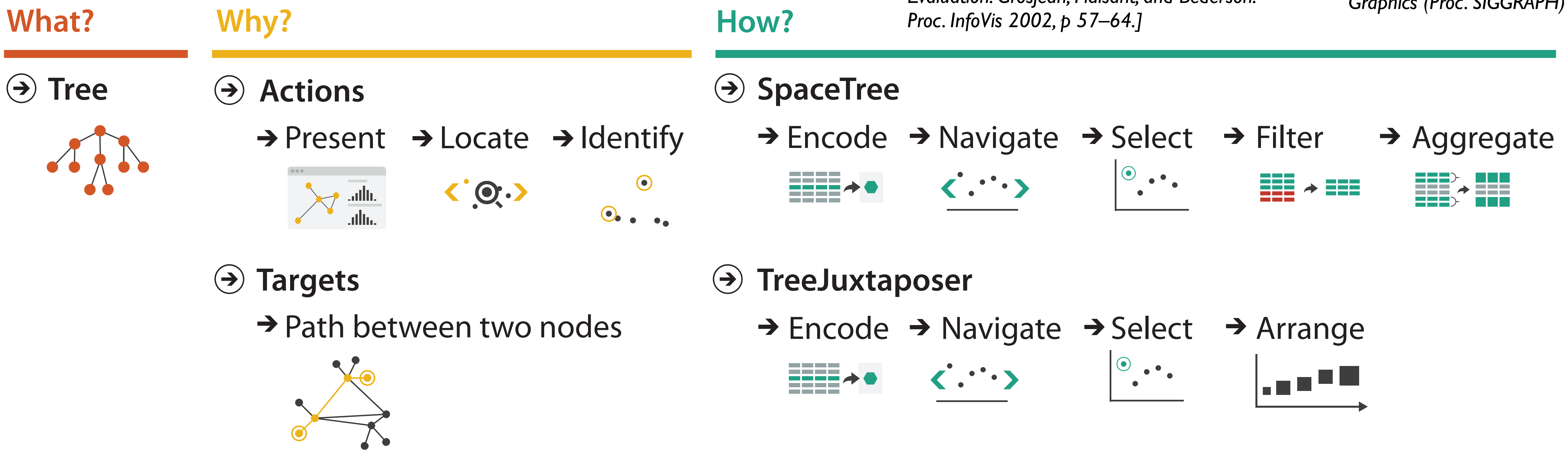


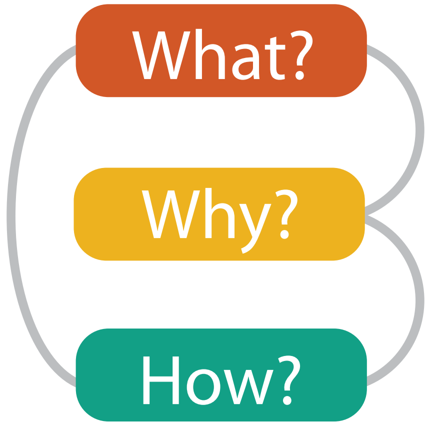
[SpaceTree: Supporting Exploration in Large Node Link Tree, Design Evolution and Empirical Evaluation. Grosjean, Plaisant, and Bederson. Proc. InfoVis 2002, p 57–64.]

TreeJuxtaposer

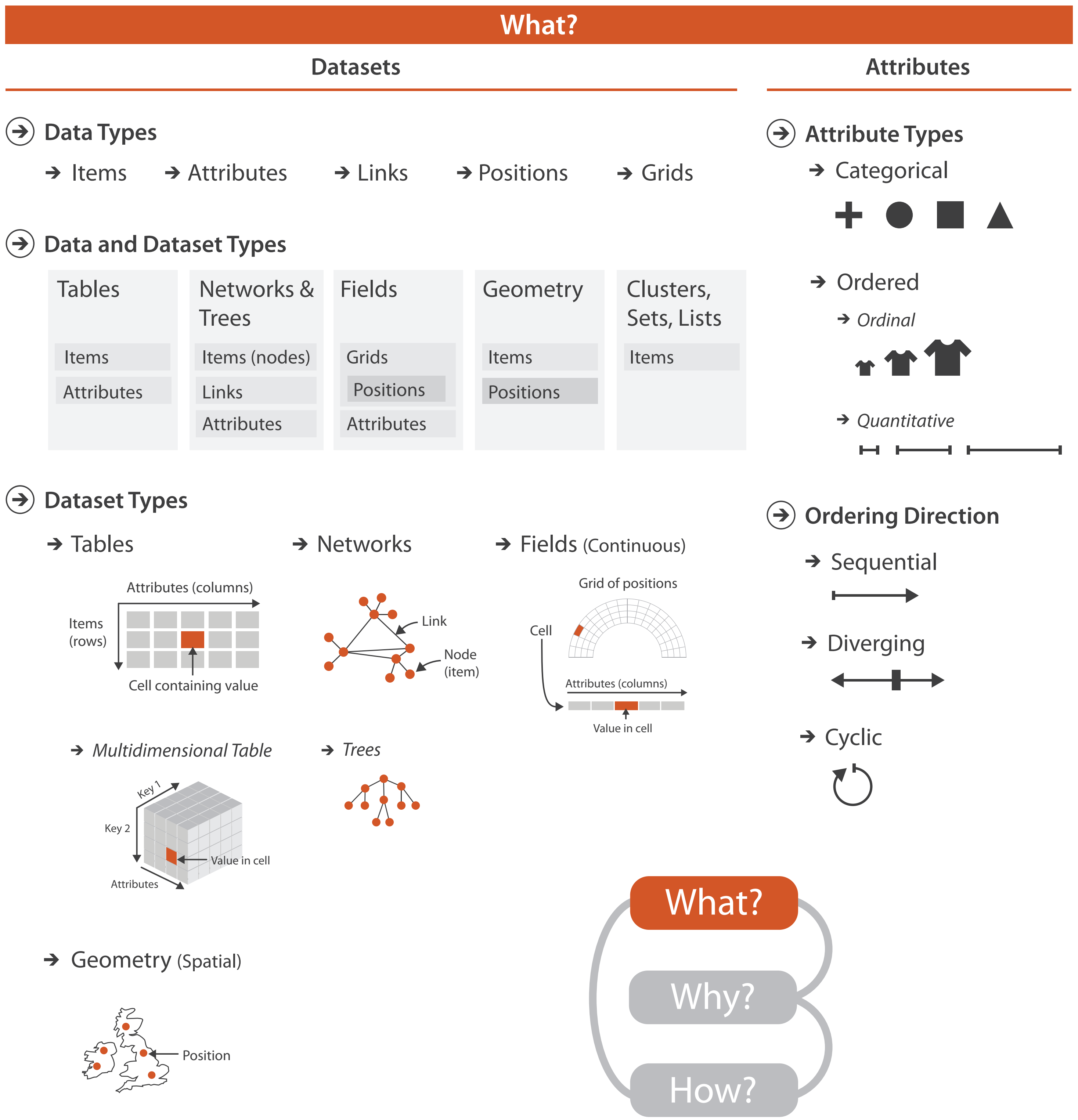


[TreeJuxtaposer: Scalable Tree Comparison Using Focus+Context With Guaranteed Visibility. ACM Trans. on Graphics (Proc. SIGGRAPH) 22:453– 462, 2003.]



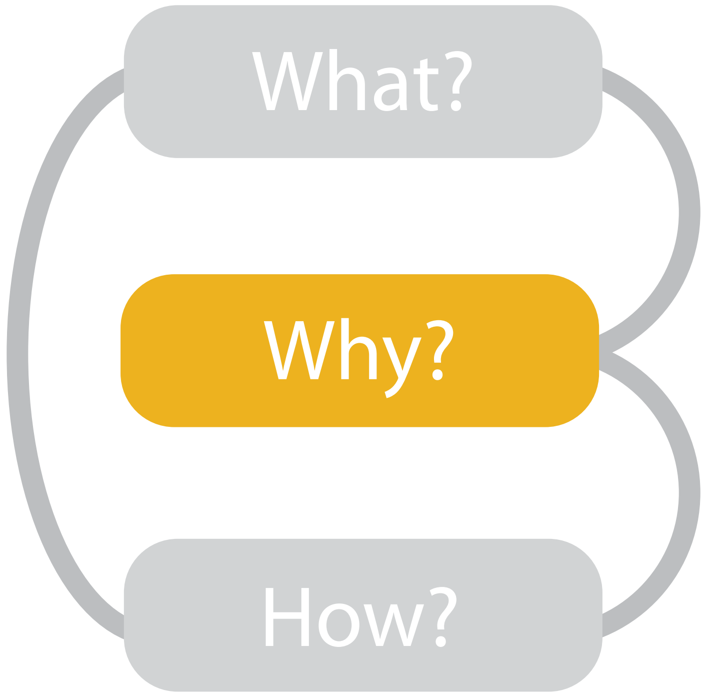


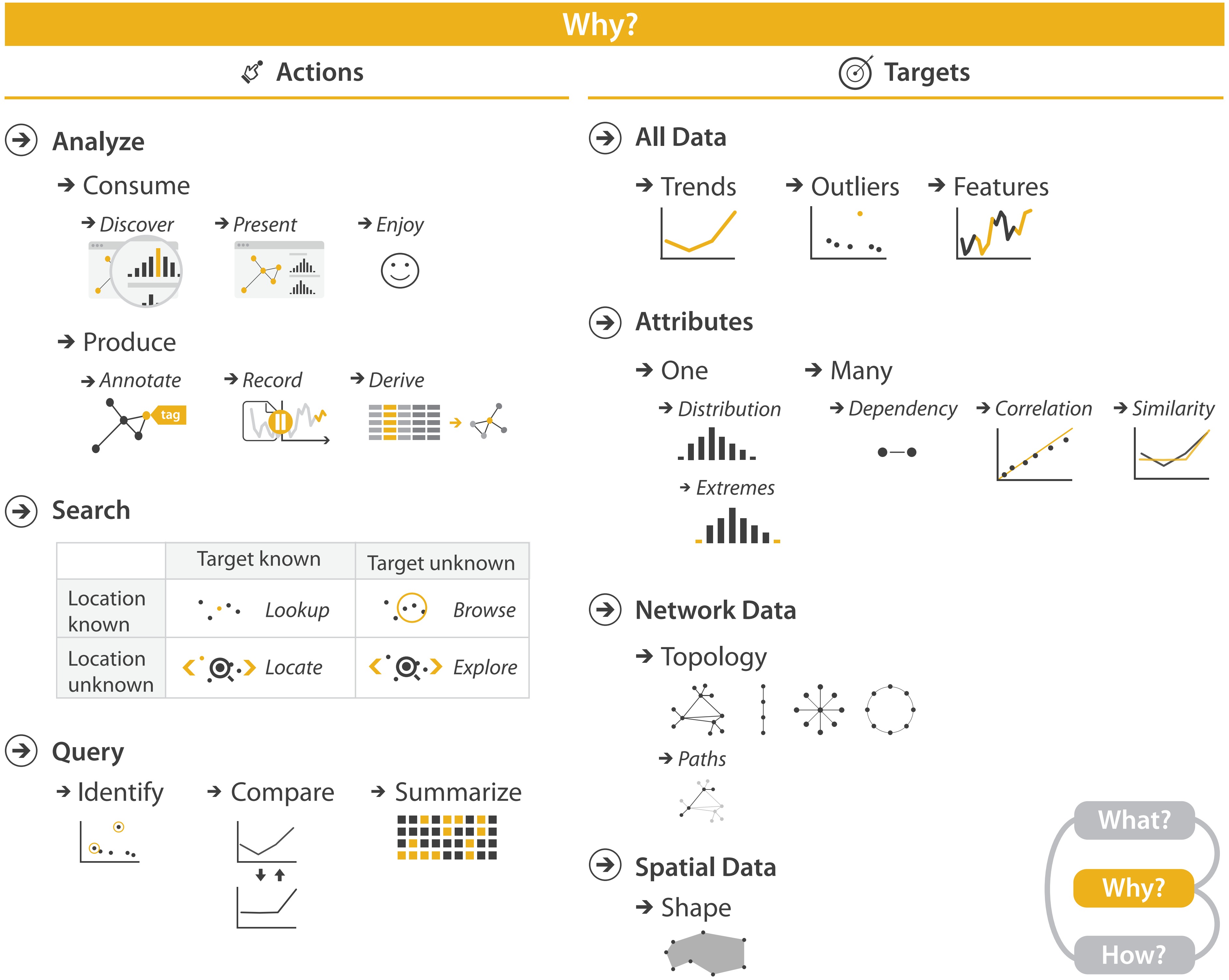
Data Abstraction



[VAD Fig 2.1]

VAD Ch 3: Task Abstraction





- {action, target} pairs
  - *discover distribution*
  - *compare trends*
  - *locate outliers*
  - *browse topology*

[VAD Fig 3.1]

How?

Encode

Arrange
- Express
- Separate
- Order
- Align
- Use

Map
from categorical and ordered attributes

- Color
  - Hue
  - Saturation
  - Luminance
- Size, Angle, Curvature, ...
- Shape
- Motion
  Direction, Rate, Frequency, ...

Manipulate

Change

Select

Navigate

Facet

Juxtapose

Partition

Superimpose

Reduce

Filter

Aggregate

Embed

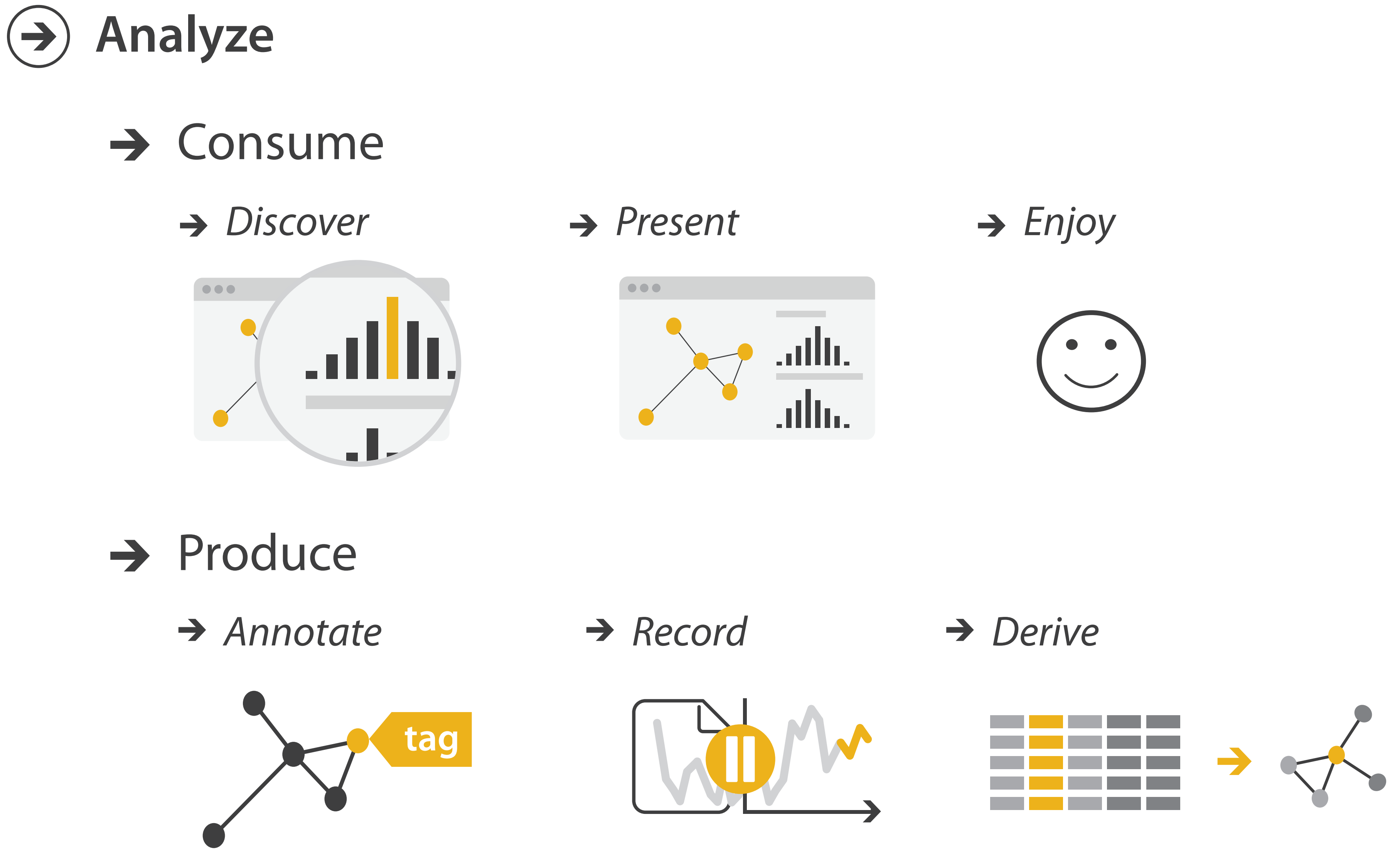
What?

Why?

How?

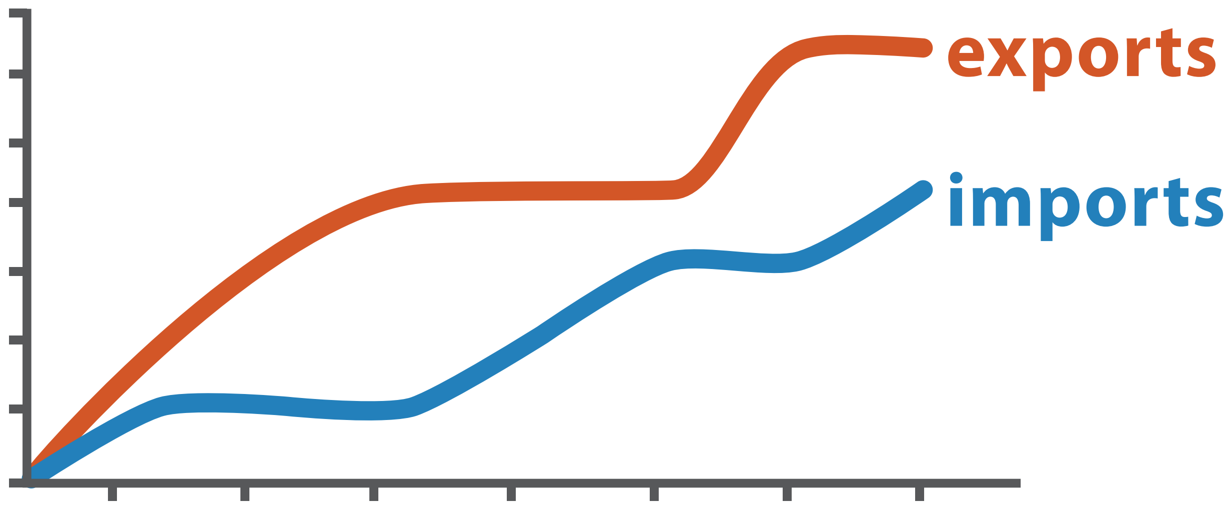
High-level actions: Analyze

- consume
  - discover vs present
    - classic split
    - aka explore vs explain
  - enjoy
    - newcomer
    - aka casual, social
- produce
  - annotate, record
  - derive
    - crucial design choice

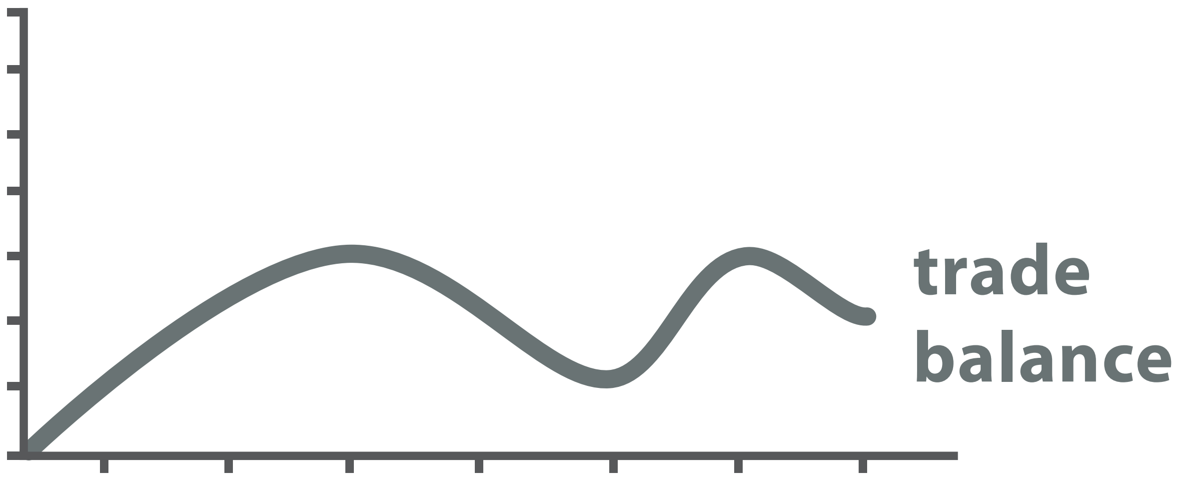


Derive

- don't just draw what you're given!
  - decide what the right thing to show is
  - create it with a series of transformations from the original dataset
  - draw that
- one of the four major strategies for handling complexity



Original Data



trade balance = exports − imports

Derived Data

Actions: Mid-level search, low-level query

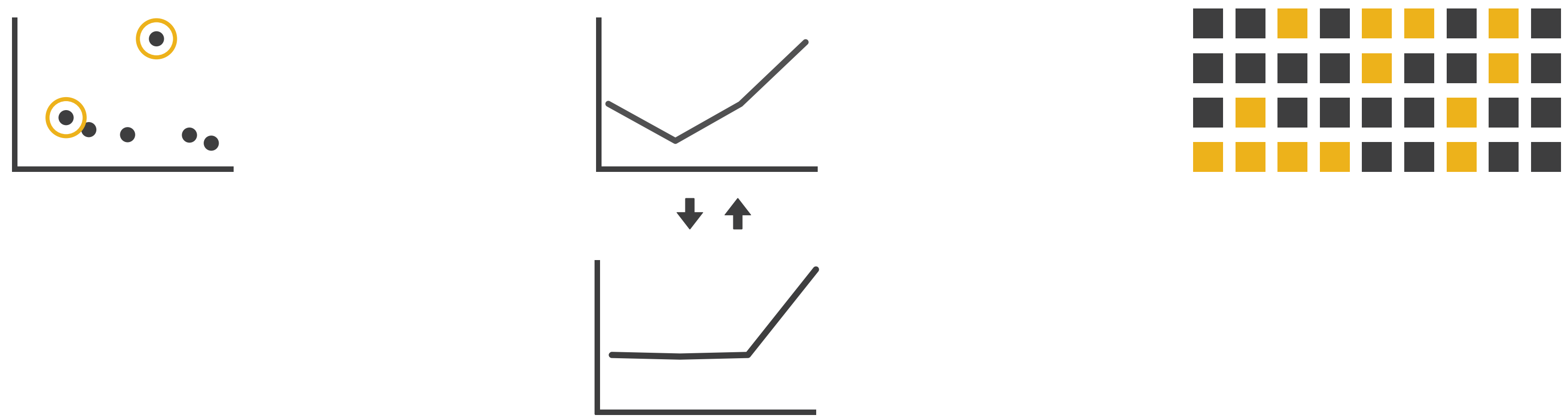
- what does user know?
  - –target, location
- how much of the data matters?
  - –one, some, all
- independent choices, mix & match
  - –analyze, query, search

➔ **Search**

	Target known	Target unknown
Location known	*Lookup*	*Browse*
Location unknown	*Locate*	*Explore*

➔ **Query**

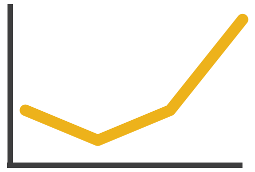
→ Identify → Compare → Summarize



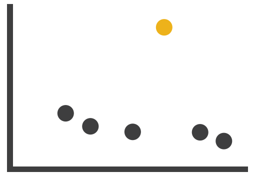
Targets

All Data

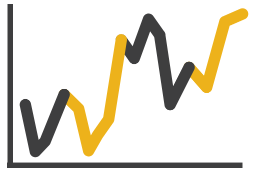
- Trends



- Outliers

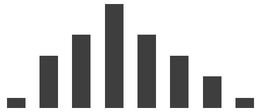


- Features

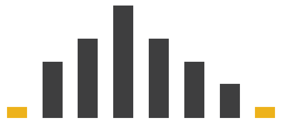


Attributes

- One
  - *Distribution*

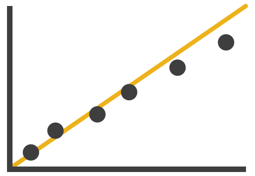


  - *Extremes*

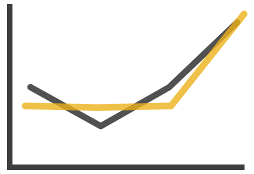


- Many
  - *Dependency*

  - *Correlation*

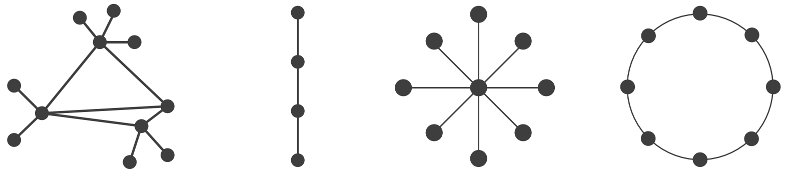


  - *Similarity*

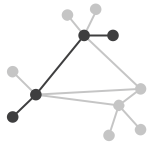


Network Data

- Topology

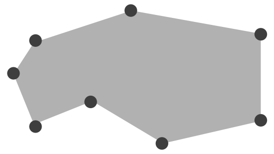


  - *Paths*



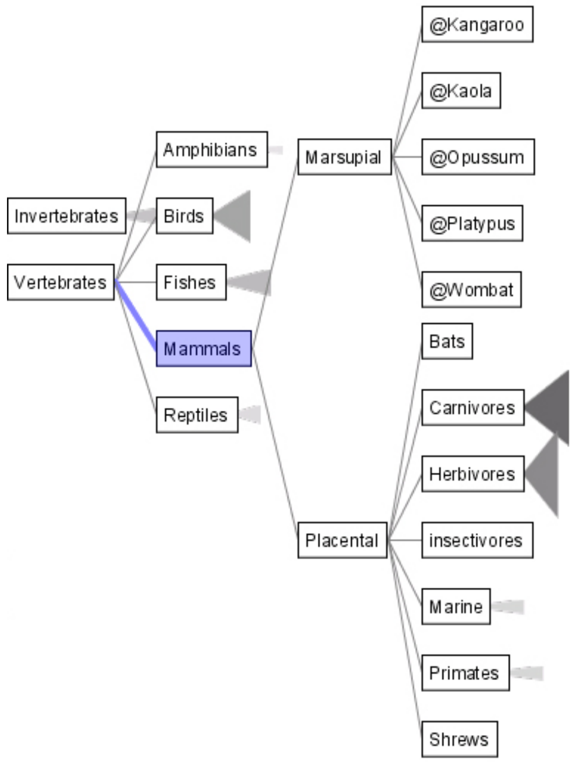
Spatial Data

- Shape

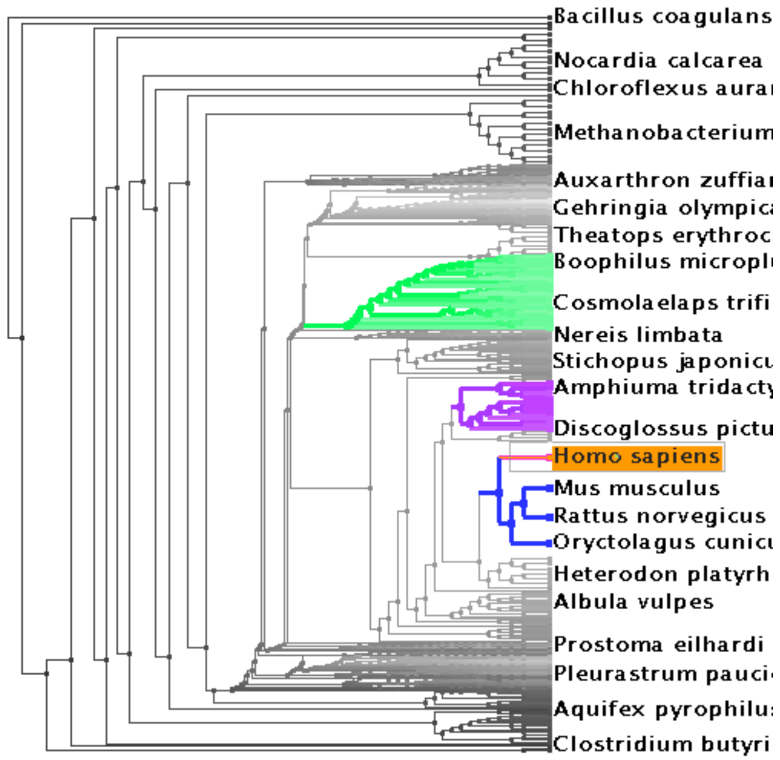


Analysis example: Compare idioms

SpaceTree

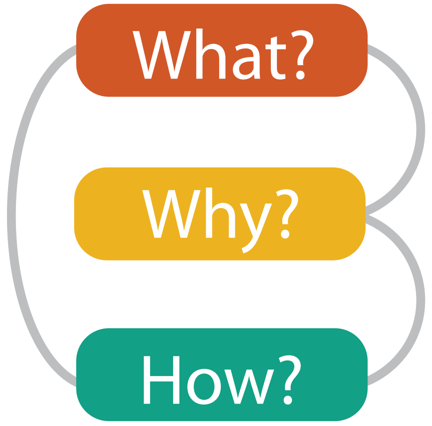


TreeJuxtaposer



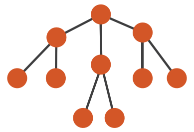
[SpaceTree: Supporting Exploration in Large Node Link Tree, Design Evolution and Empirical Evaluation. Grosjean, Plaisant, and Bederson. Proc. InfoVis 2002, p 57–64.]

[TreeJuxtaposer: Scalable Tree Comparison Using Focus+Context With Guaranteed Visibility. ACM Trans. on Graphics (Proc. SIGGRAPH) 22:453– 462, 2003.]



What?

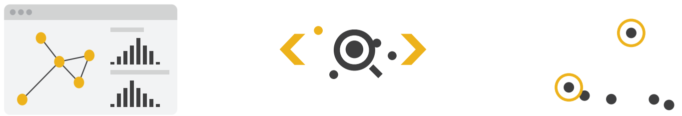
➔ Tree



Why?

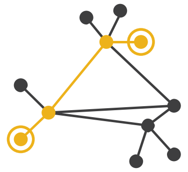
➔ Actions

➔ Present ➔ Locate ➔ Identify



➔ Targets

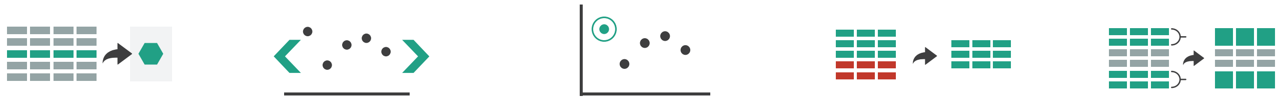
➔ Path between two nodes



How?

➔ SpaceTree

➔ Encode ➔ Navigate ➔ Select ➔ Filter ➔ Aggregate



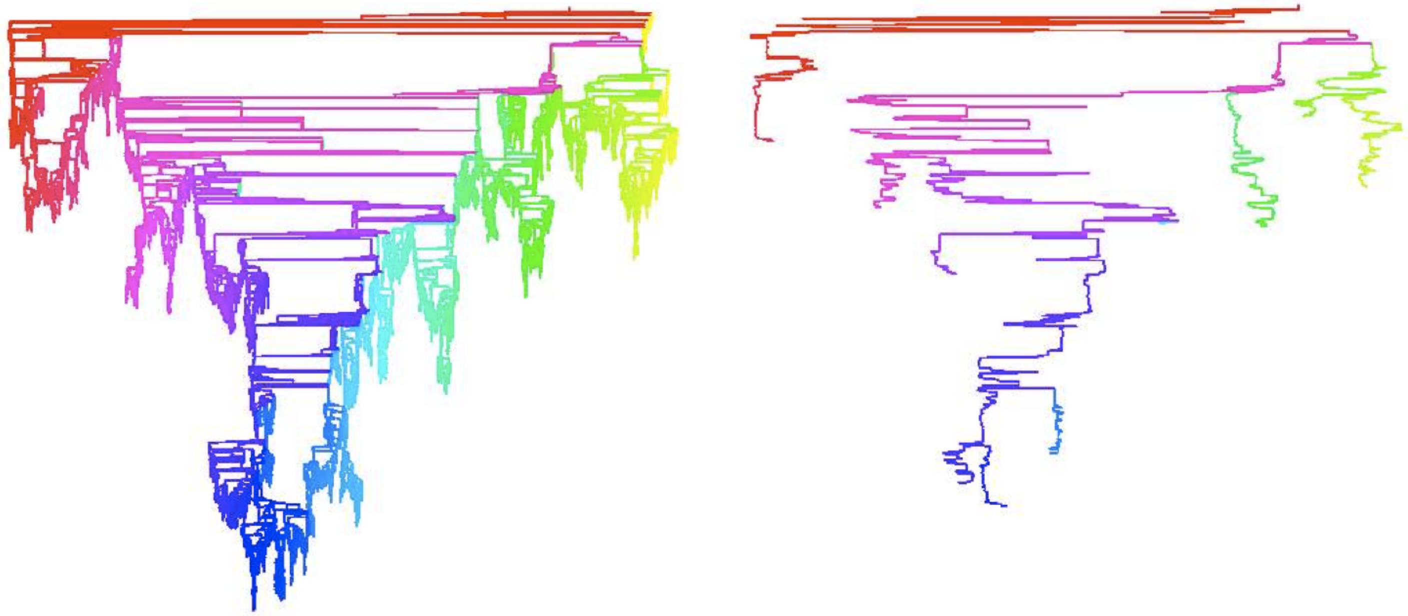
➔ TreeJuxtaposer

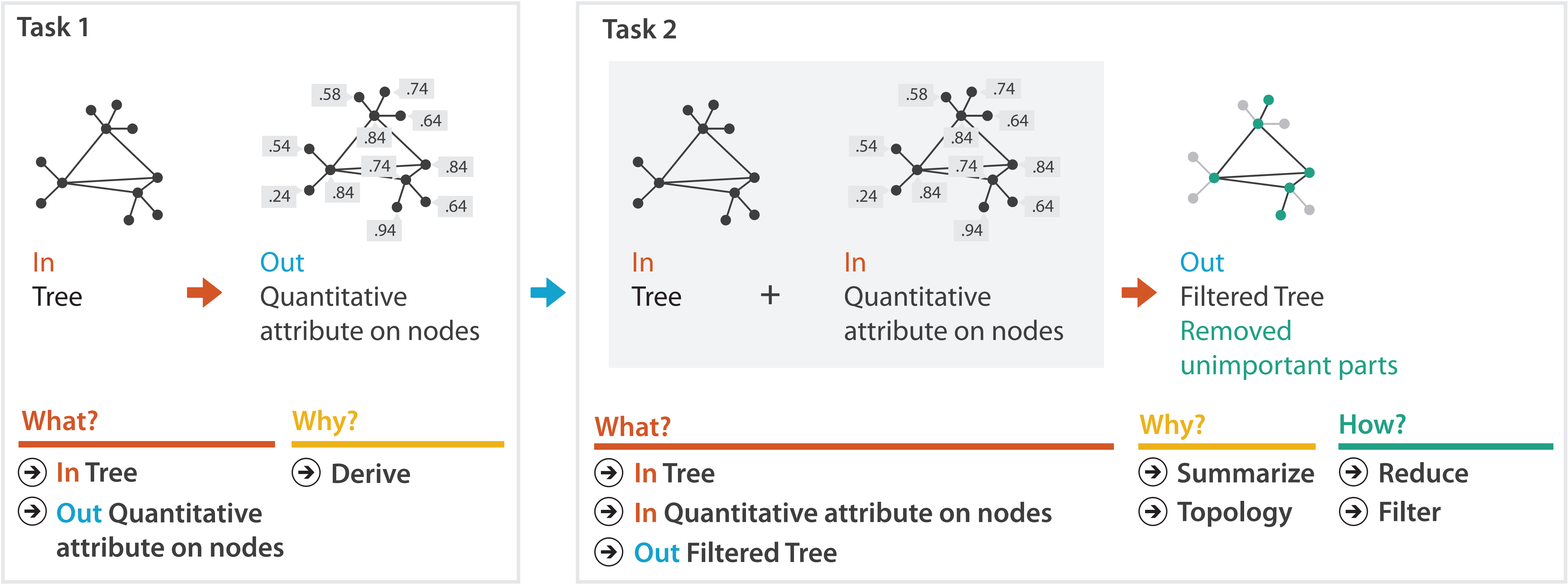
➔ Encode ➔ Navigate ➔ Select ➔ Arrange

Analysis example: Derive one attribute

- Strahler number
  - centrality metric for trees/networks
  - derived quantitative attribute
  - draw top 5K of 500K for good skeleton

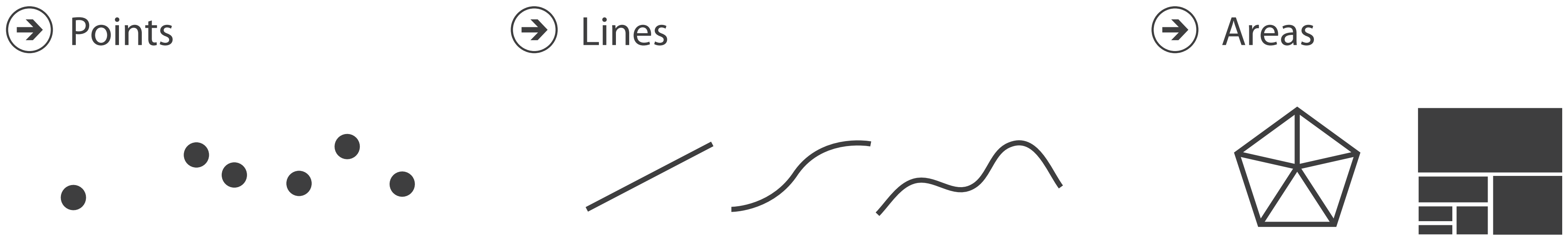
  [Using Strahler numbers for real time visual exploration of huge graphs. Auber. Proc. Intl. Conf. Computer Vision and Graphics, pp. 56–69, 2002.]



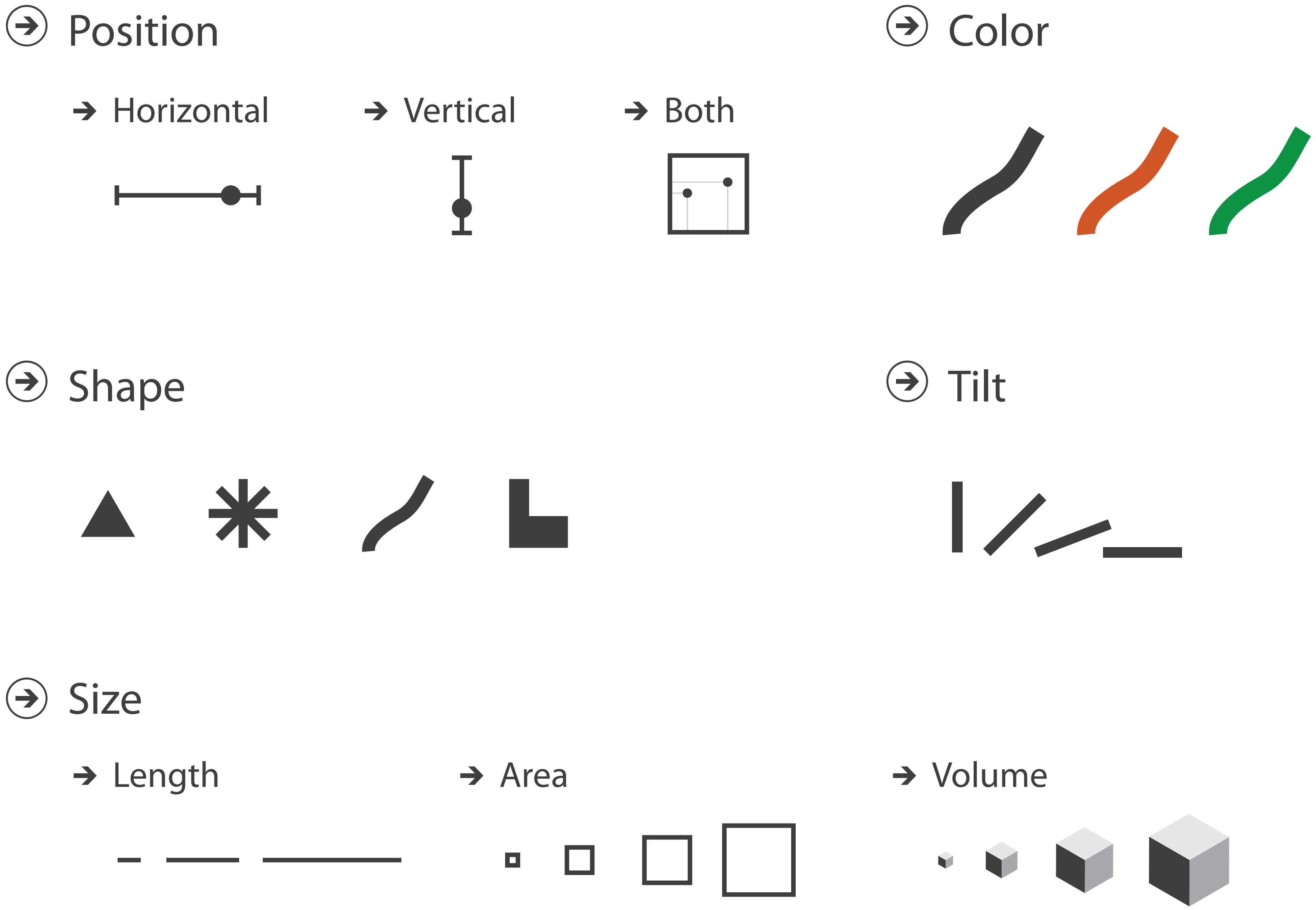


Definitions: Marks and channels

- marks
  - geometric primitives

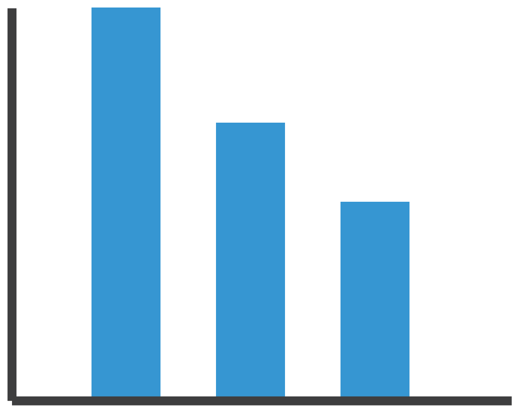


- channels
  - control appearance of marks



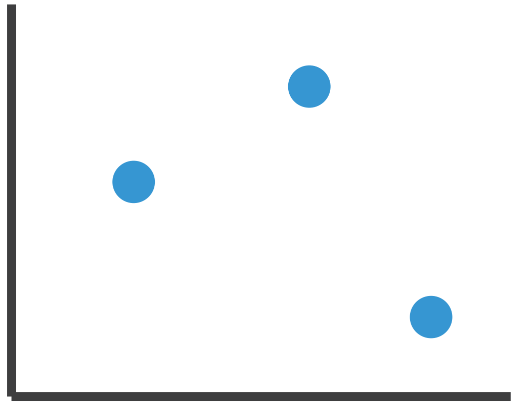
Encoding visually with marks and channels

- analyze idiom structure
  - as combination of marks and channels



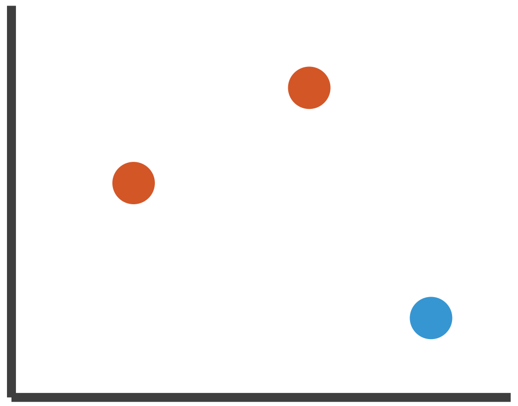
1:
vertical position

mark: line



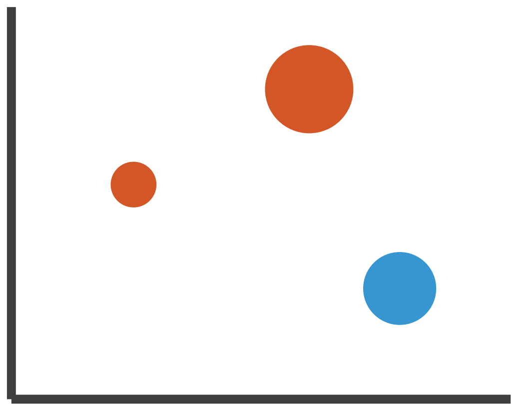
2:
vertical position
horizontal position

mark: point



3:
vertical position
horizontal position
color hue

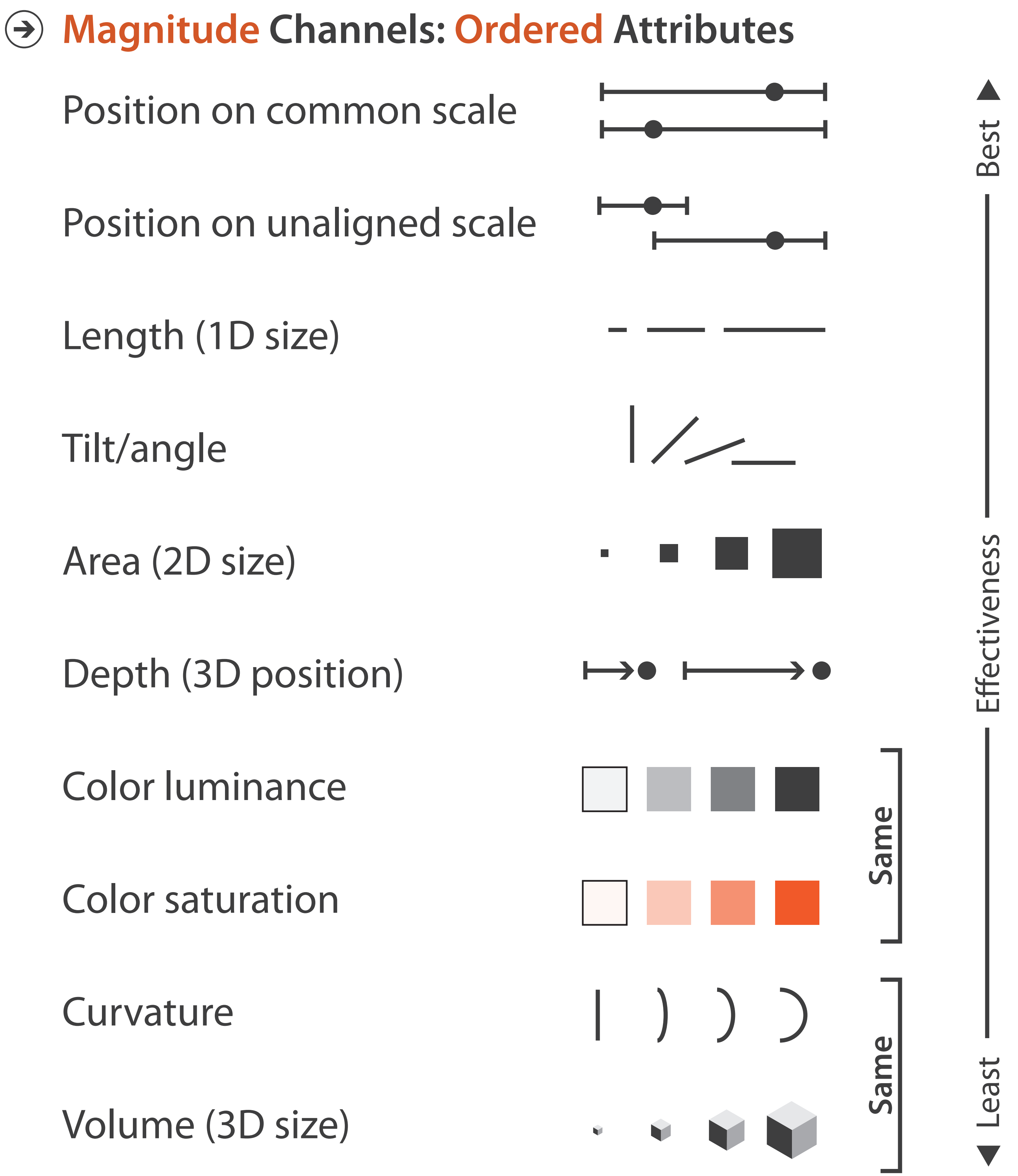
mark: point

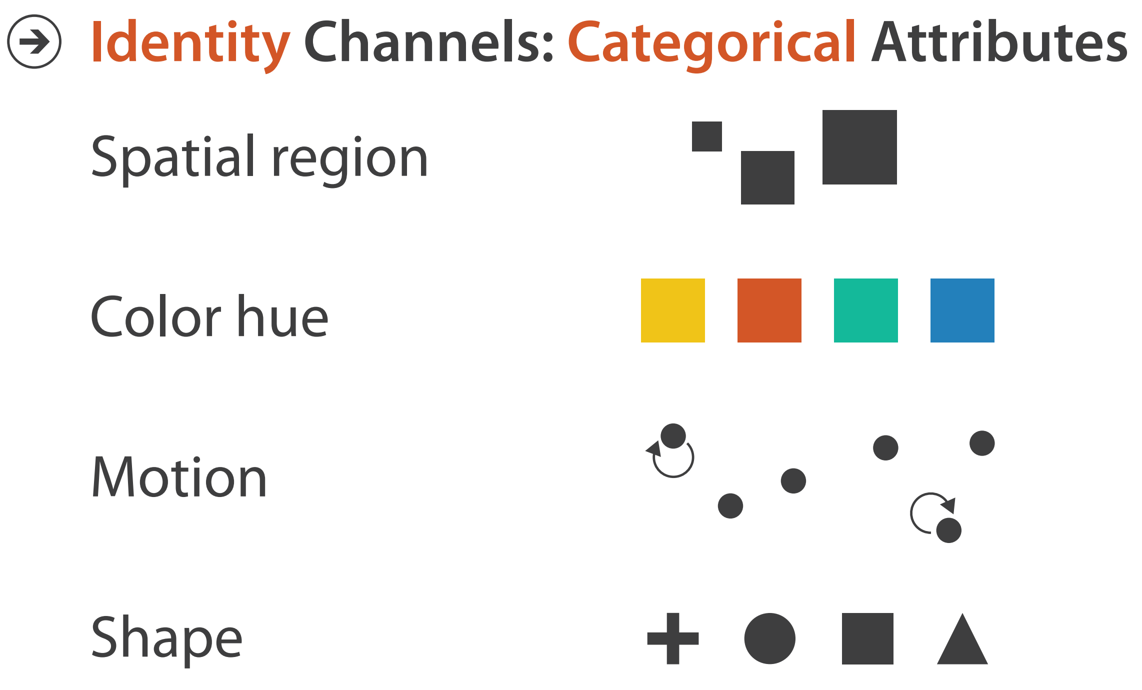


4:
vertical position
horizontal position
color hue
size (area)

mark: point

Channels: Rankings

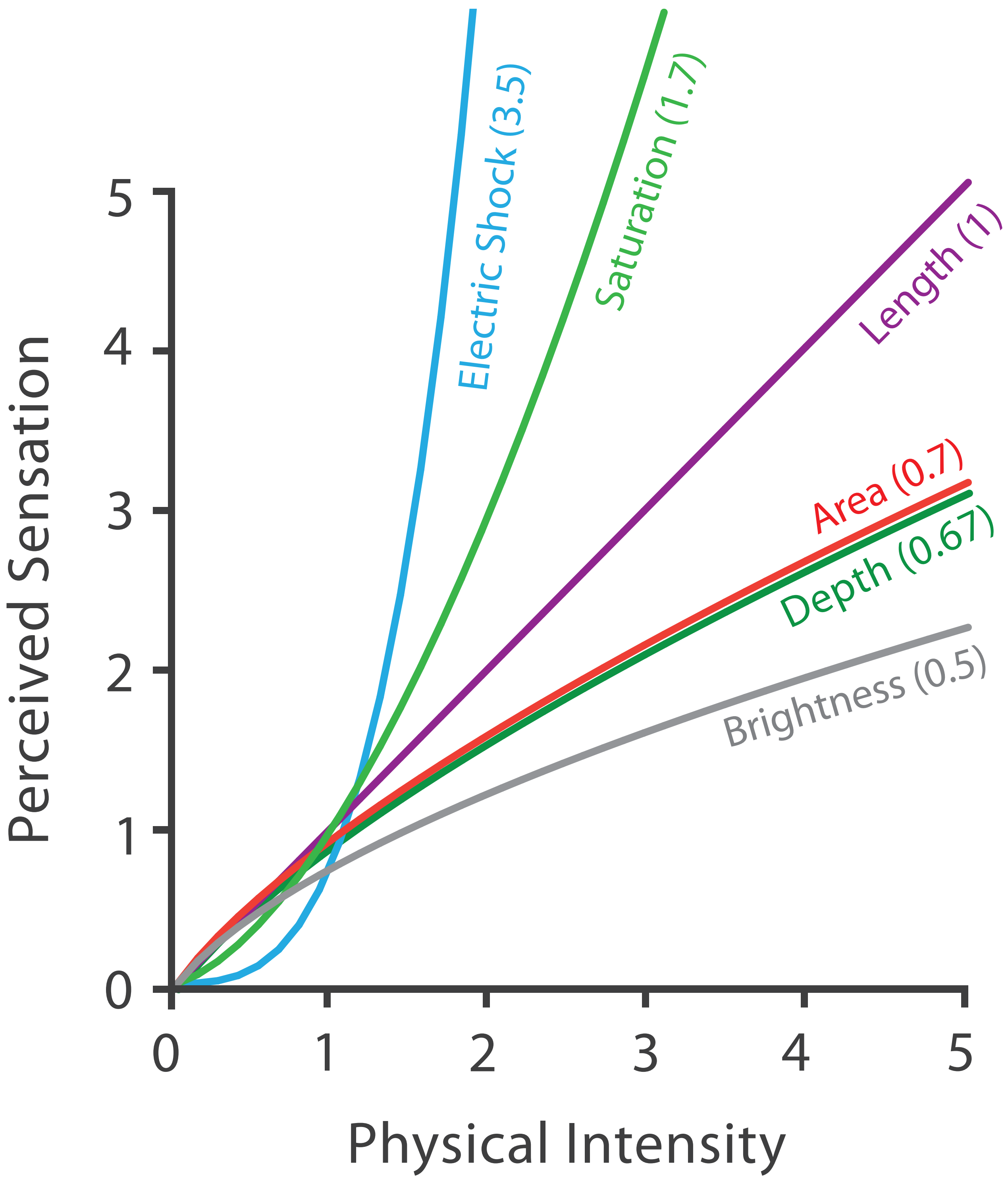




- effectiveness principle
  - encode most important attributes with highest ranked channels
- expressiveness principle
  - match channel and data characteristics

Accuracy: Fundamental Theory

Steven's Psychophysical Power Law: $S = I^N$



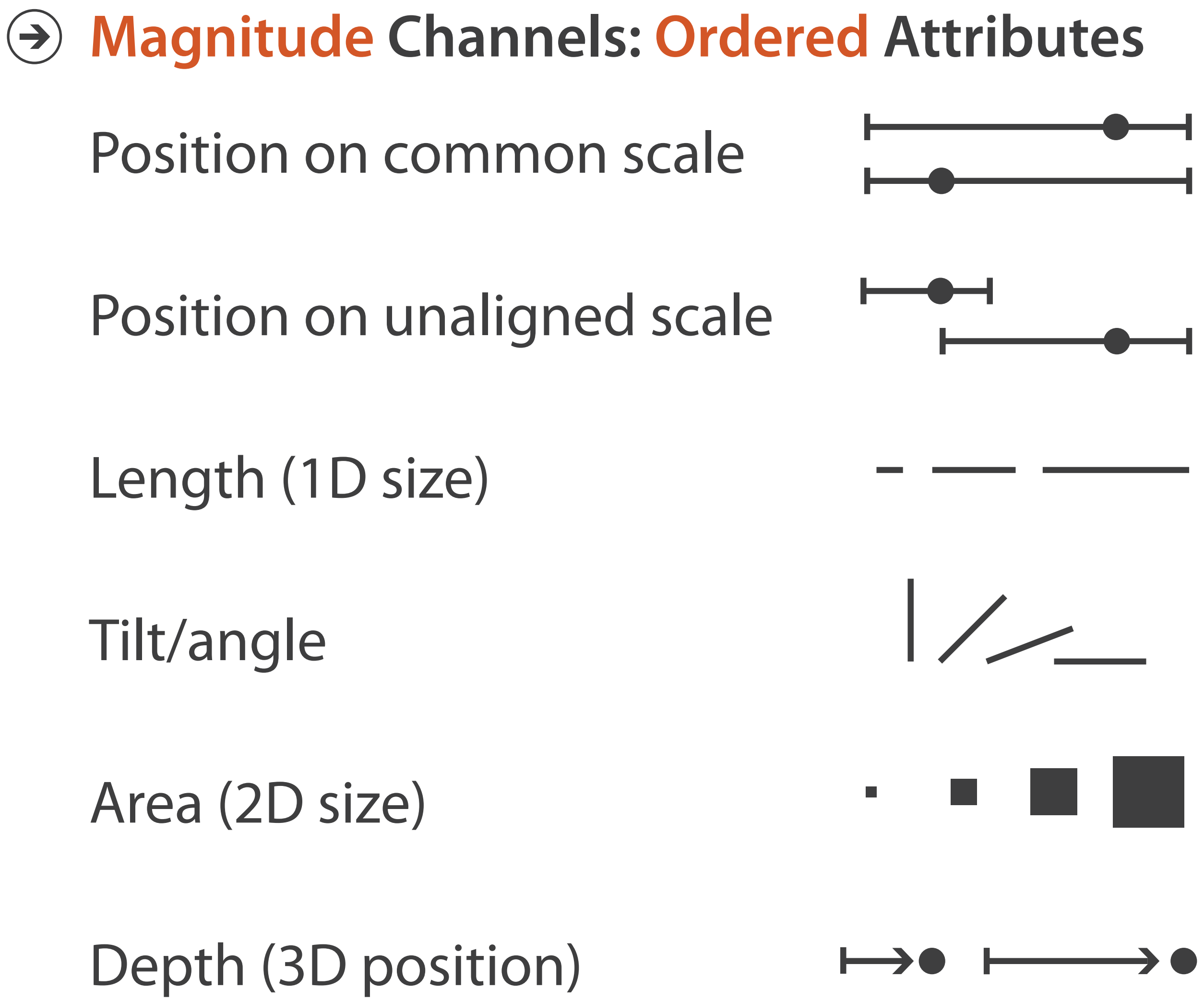
Rules of Thumb

- No unjustified 3D
  - Power of the plane, dangers of depth
  - Occlusion hides information
  - Perspective distortion loses information
  - Tilted text isn't legible
- No unjustified 2D
- Eyes beat memory
- Resolution over immersion
- Overview first, zoom and filter, details on demand
- Function first, form next

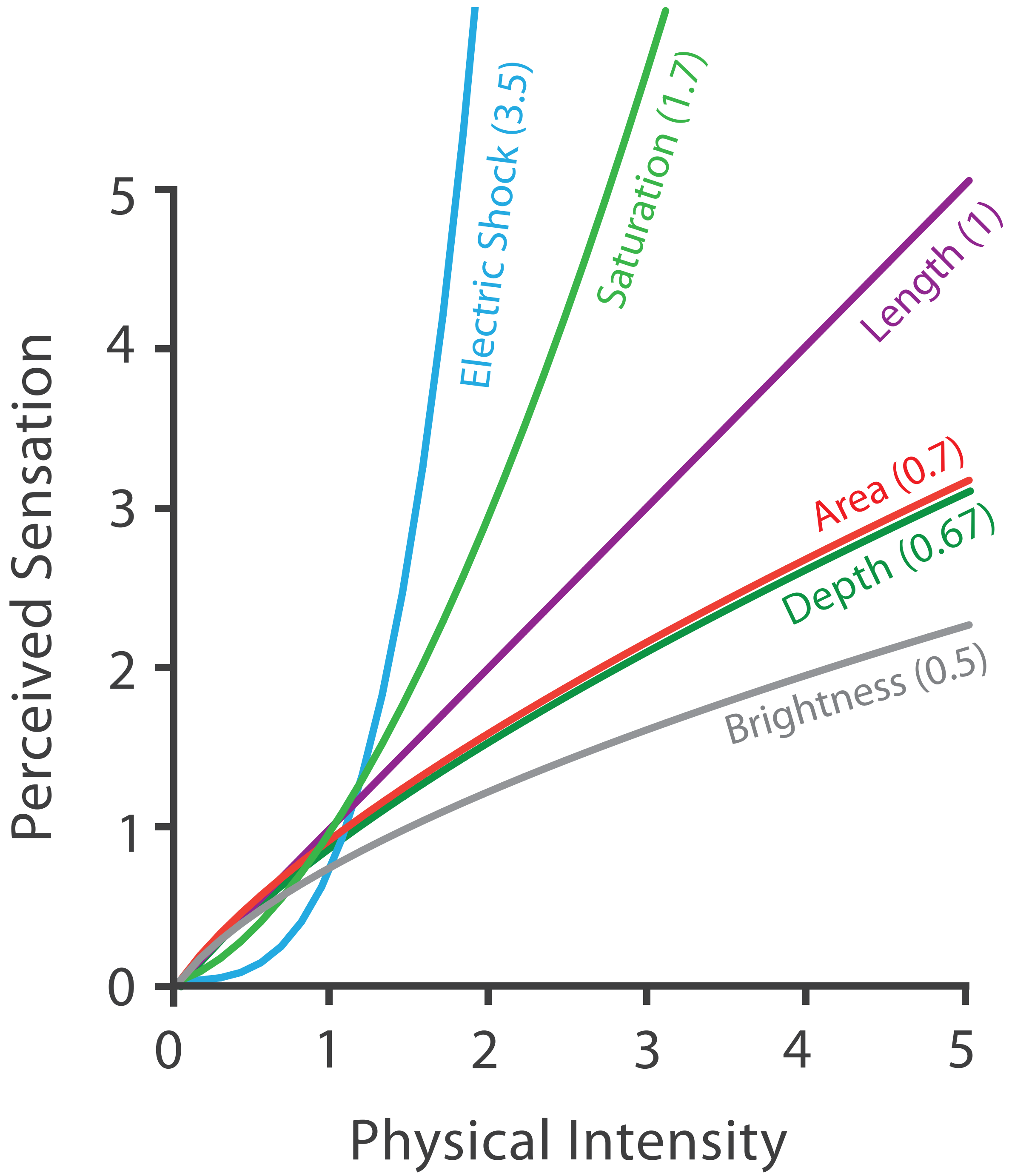
- (Get it right in black and white)

No unjustified 3D: Power of the plane

- high-ranked spatial position channels: **planar** spatial position
  - not depth!

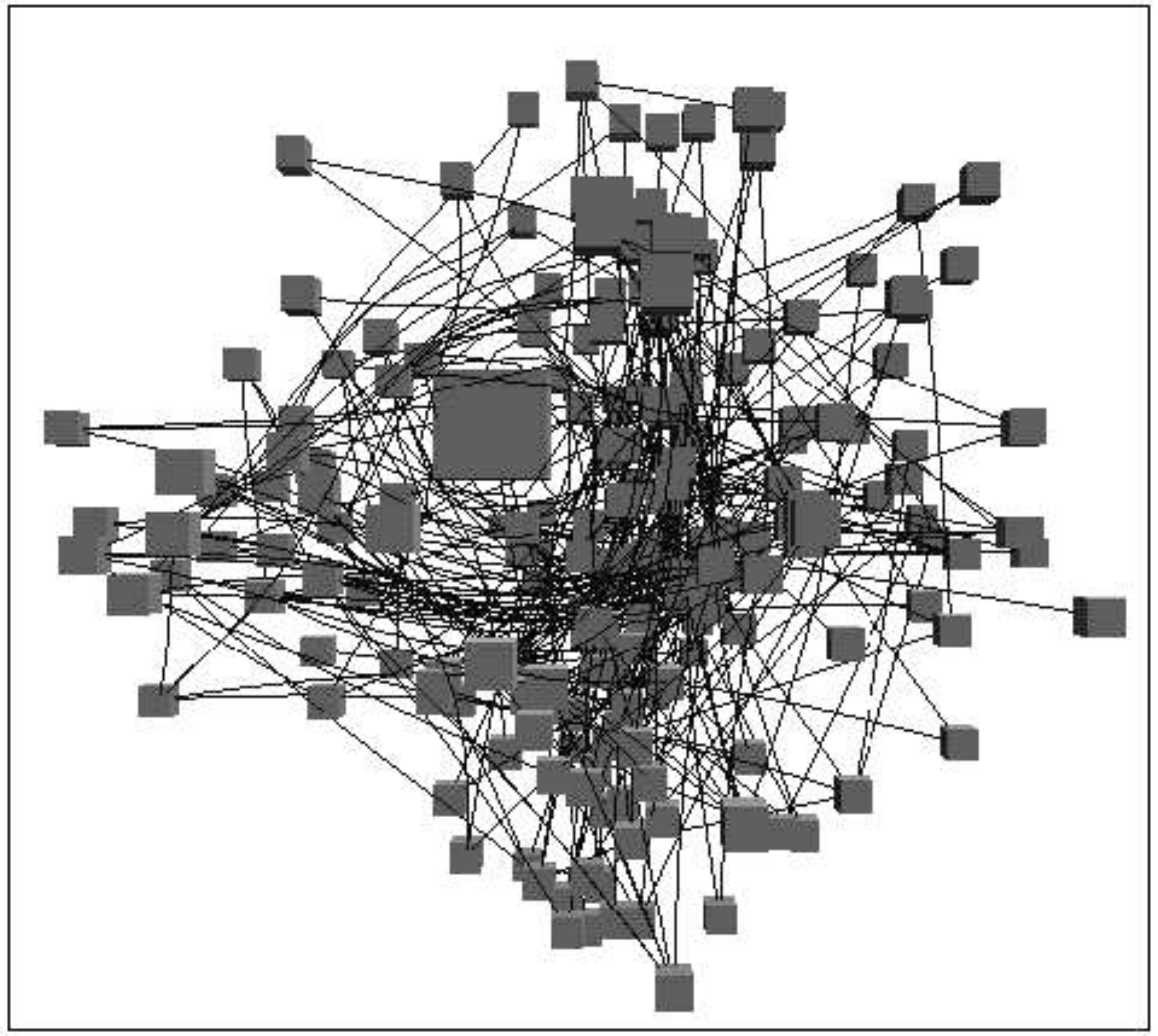


Steven's Psychophysical Power Law: $S = I^N$



Occlusion hides information

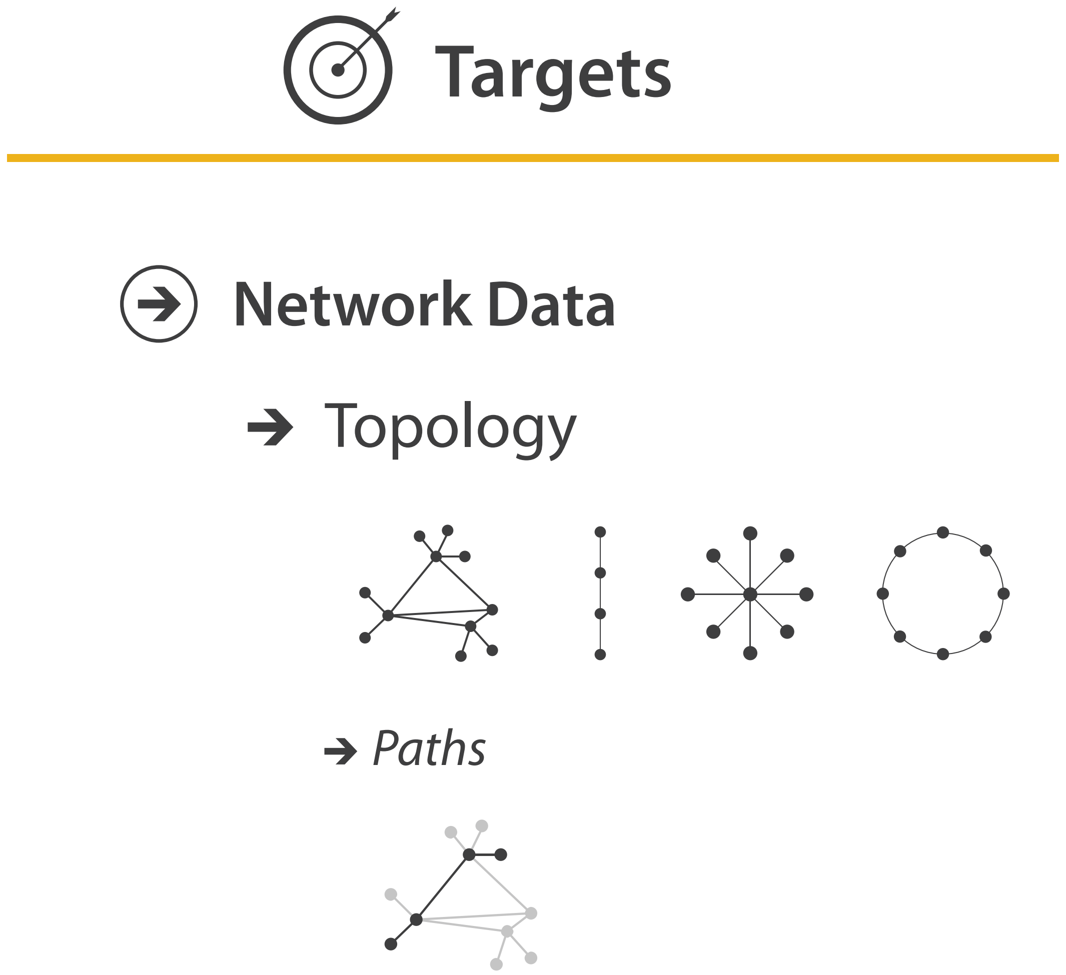
- occlusion
- interaction complexity



[Distortion Viewing Techniques for 3D Data. Carpendale et al. InfoVis1996.]

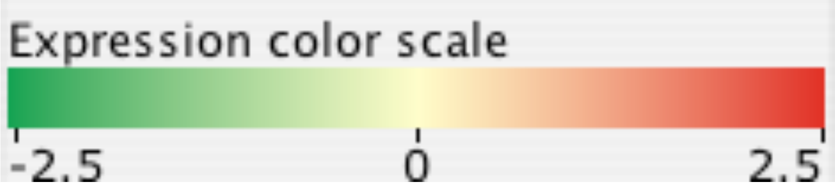
No unjustified 2D

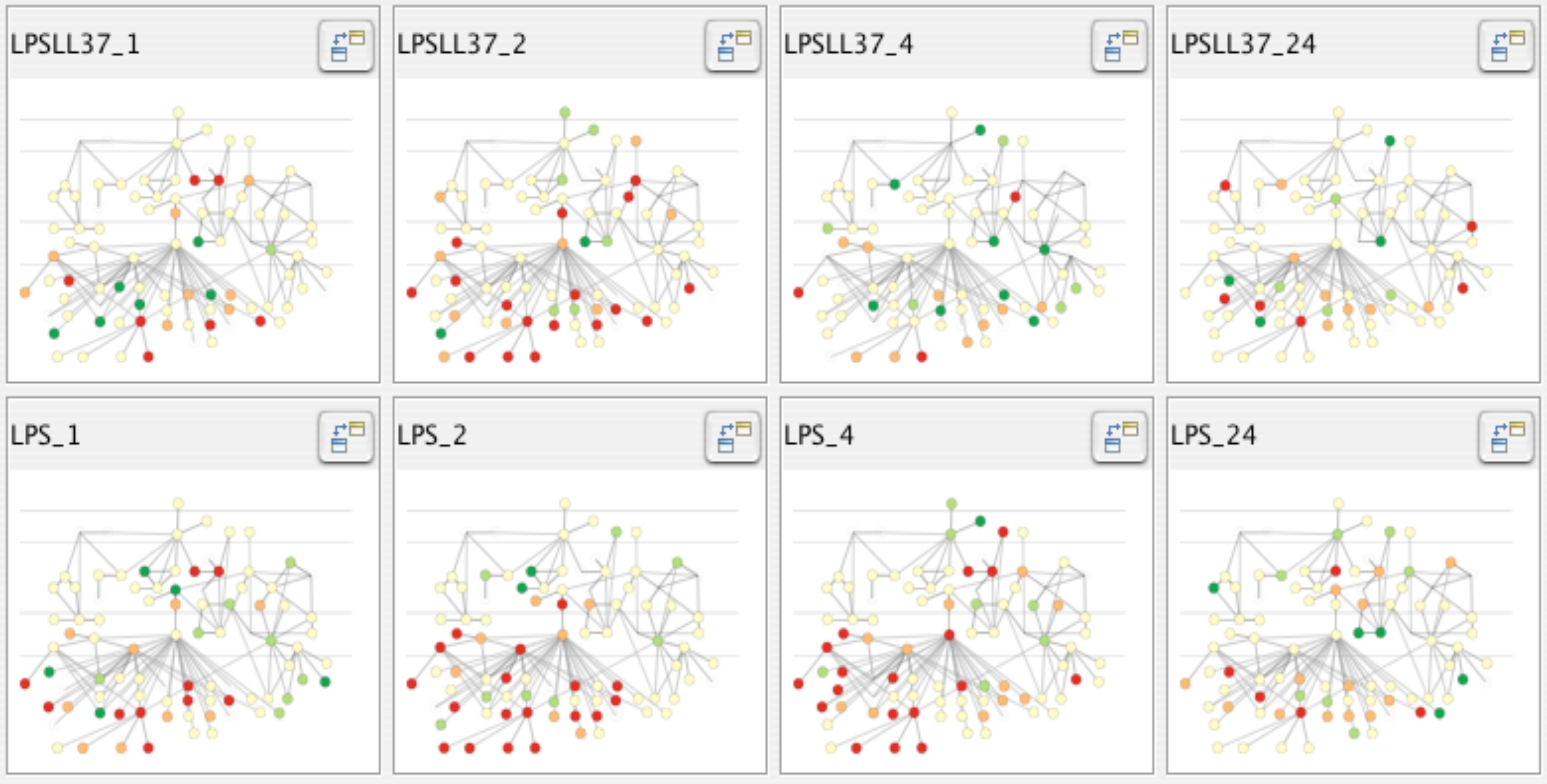
- consider whether network data requires 2D spatial layout
  - especially if reading text is central to task!
  - arranging as network means lower information density and harder label lookup compared to text lists
- benefits outweigh costs when topological structure/context important for task
  - be especially careful for search results, document collections, ontologies



Eyes beat memory example: Cerebral

- small multiples: one graph instance per experimental condition
  - same spatial layout
  - color differently, by condition

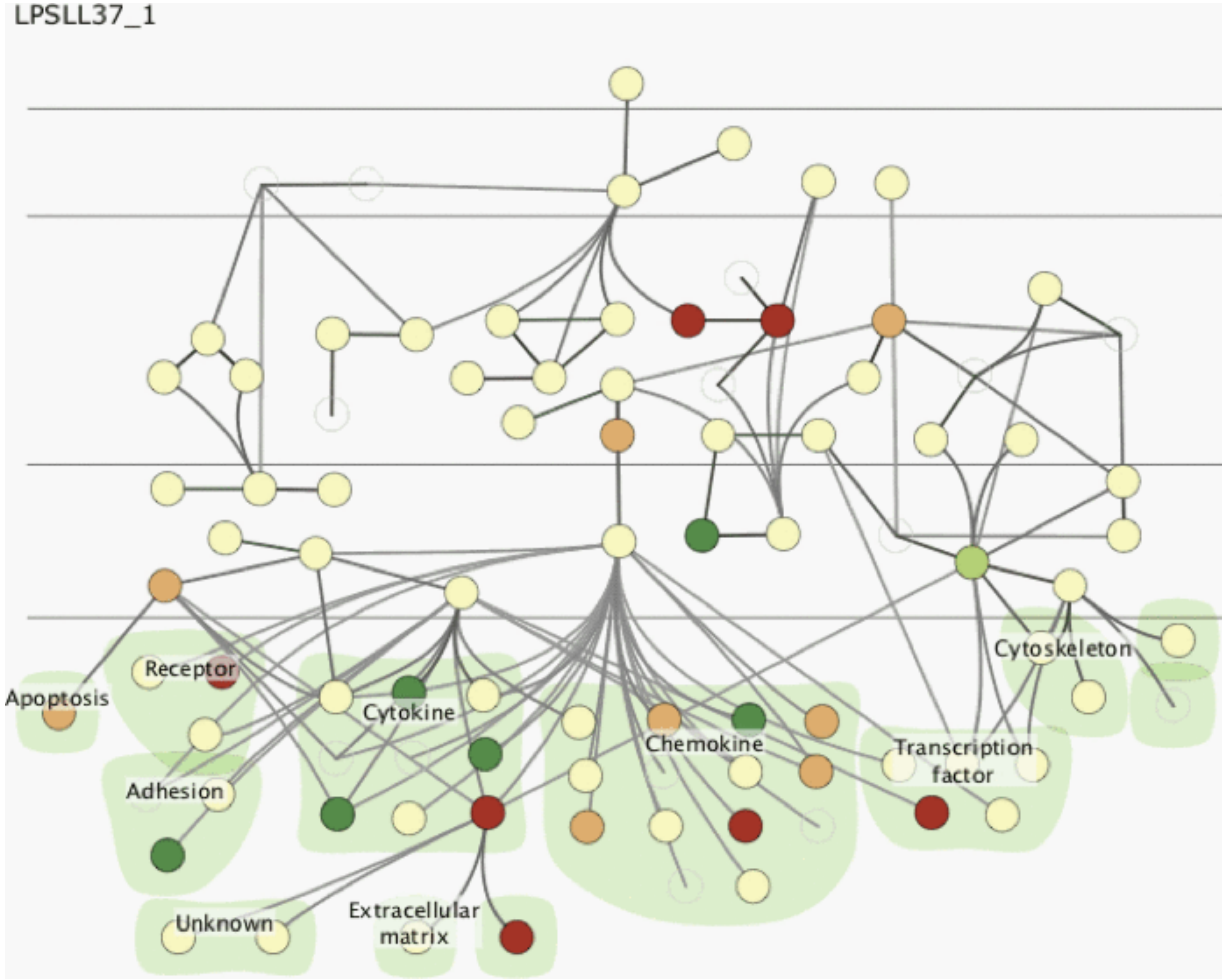




[Cerebral: Visualizing Multiple Experimental Conditions on a Graph with Biological Context. Barsky, Munzner, Gardy, and Kincaid. IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis 2008) 14:6 (2008), 1253–1260.]

Why not animation?

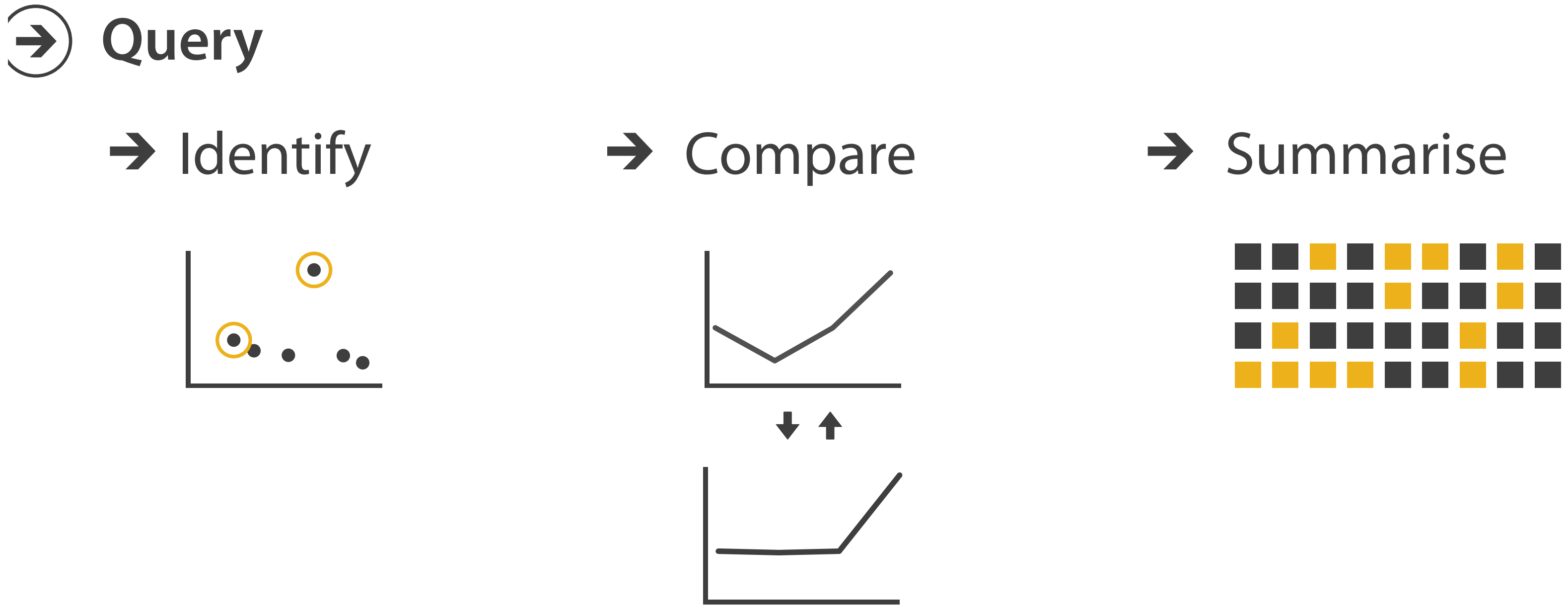
- disparate frames and regions: comparison difficult
  - –vs contiguous frames
  - –vs small region
  - –vs coherent motion of group
- change blindness
  - –even major changes difficult to notice if mental buffer wiped
- safe special case
  - –animated transitions



Overview first, zoom and filter, details on demand

- influential mantra from Shneiderman

[The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. Shneiderman. Proc. IEEE Visual Languages, pp. 336–343, 1996.]



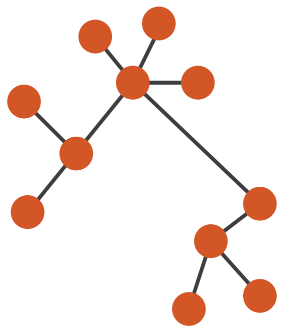
- overview = summary
  – microcosm of full vis design problem
- nuances
  – beyond just two levels: multi-scale structure
  – difficult when scale huge: give up on overview and browse local neighborhoods?

[Search, Show Context, Expand on Demand: Supporting Large Graph Exploration with Degree-of-Interest. van Ham and Perer. IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis 2009) 15:6 (2009), 953–960.]

Arrange networks and trees

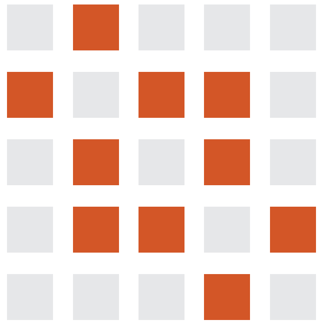
Node–Link Diagrams
Connection Marks

✔ NETWORKS ✔ TREES



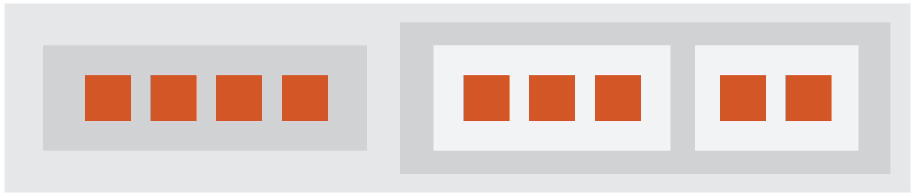
Adjacency Matrix
Derived Table

✔ NETWORKS ✔ TREES



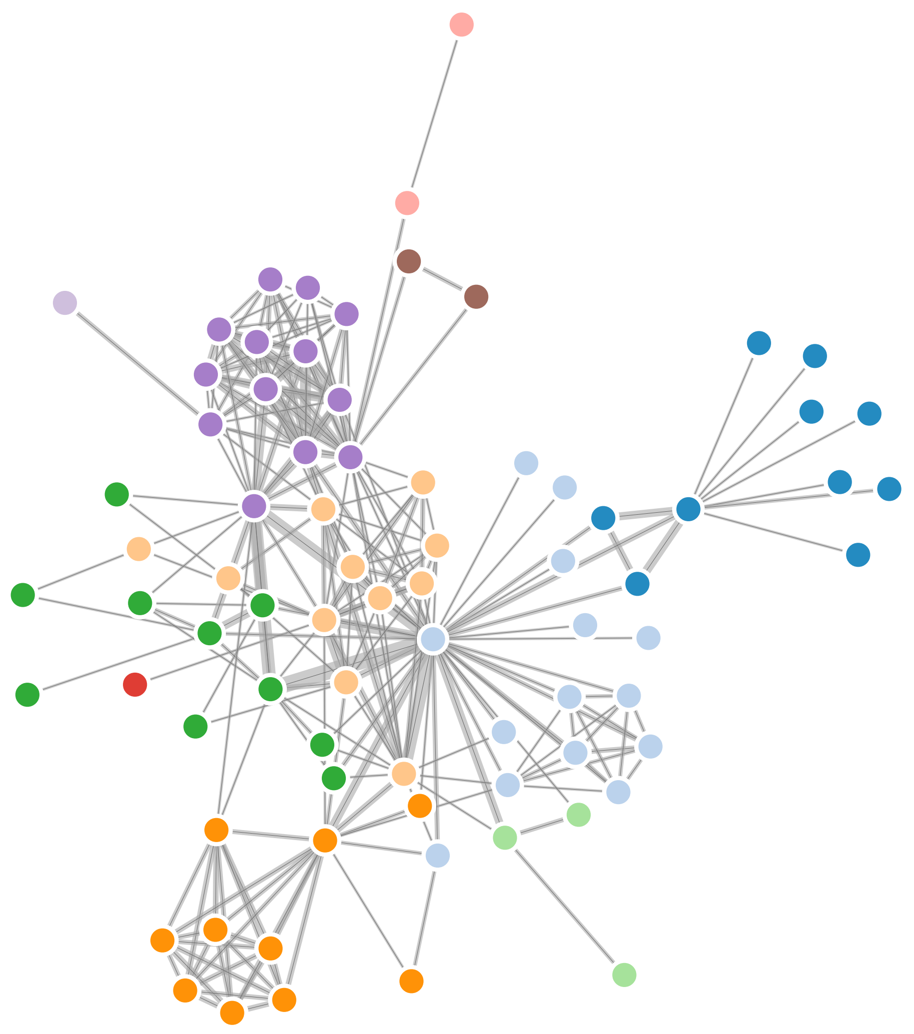
Enclosure
Containment Marks

✕ NETWORKS ✔ TREES



Idiom: **force-directed placement**

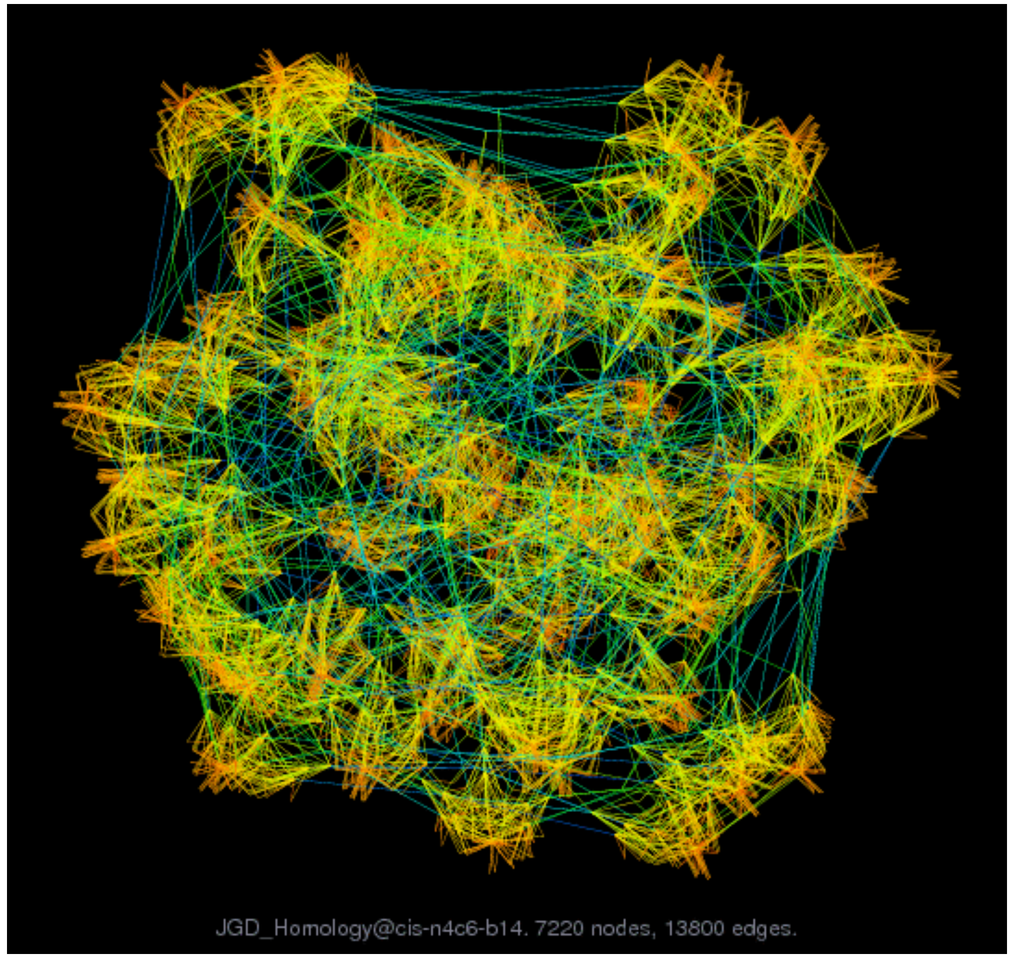
- visual encoding
  - link connection marks, node point marks
- considerations
  - spatial position: no meaning directly encoded
    - left free to minimize crossings
  - proximity semantics?
    - sometimes meaningful
    - sometimes arbitrary, artifact of layout algorithm
    - tension with length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



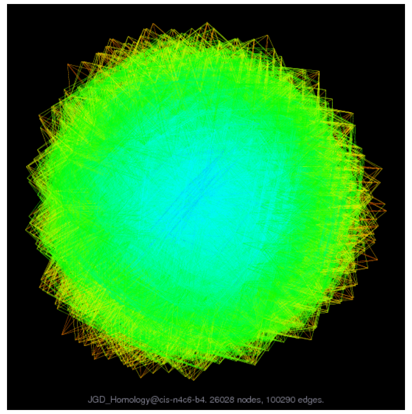
http://mbostock.github.com/d3/ex/force.html

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



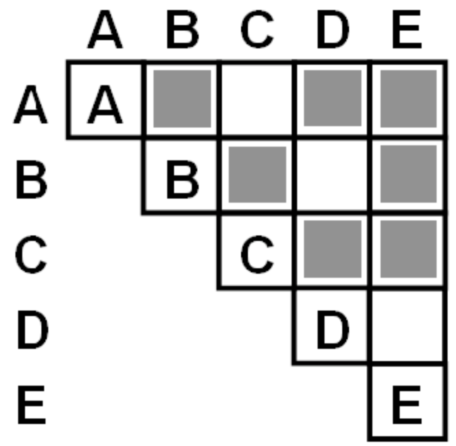
[Efficient and high quality force-directed graph drawing. Hu. The Mathematica Journal 10:37–71, 2005.]

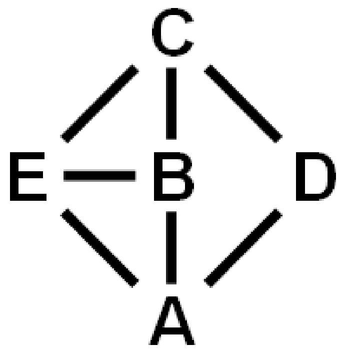


http://www.research.att.com/yifanhu/GALLERY/GRAPHS/index1.html

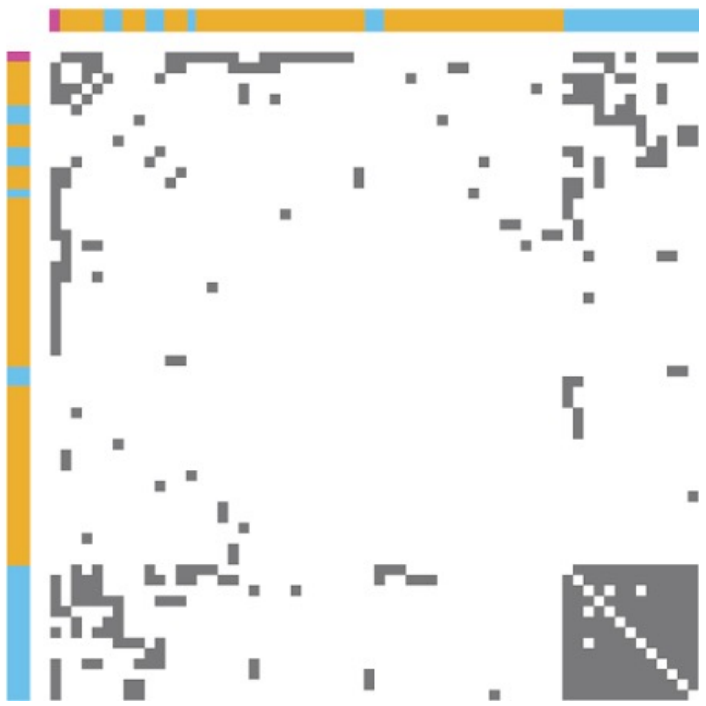
Idiom: **adjacency matrix view**

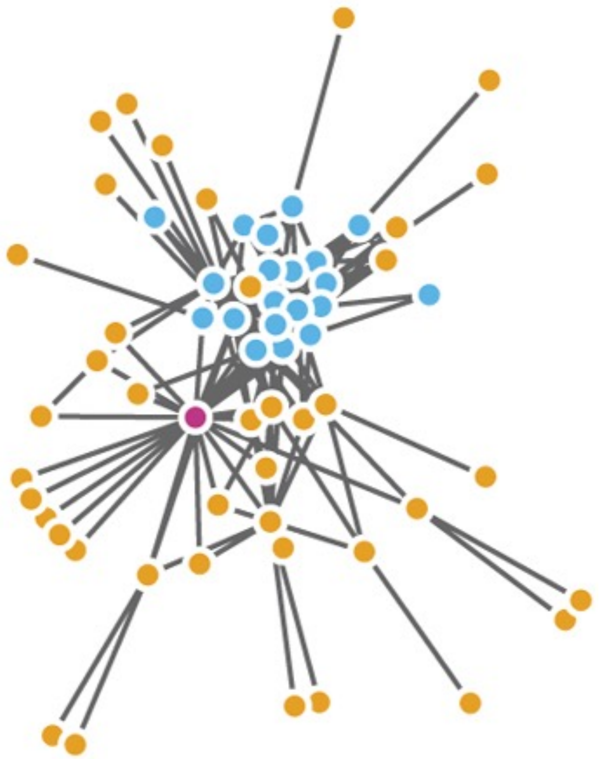
- data: network
  - transform into same data/encoding as heatmap
- derived data: table from network
  - 1 quant attrib
    - weighted edge between nodes
  - 2 categ attribs: node list x 2
- visual encoding
  - cell shows presence/absence of edge
- scalability
  - 1K nodes, 1M edges





[NodeTrix: a Hybrid Visualization of Social Networks. Henry, Fekete, and McGuffin. IEEE TVCG (Proc. InfoVis) 13(6):1302-1309, 2007.]



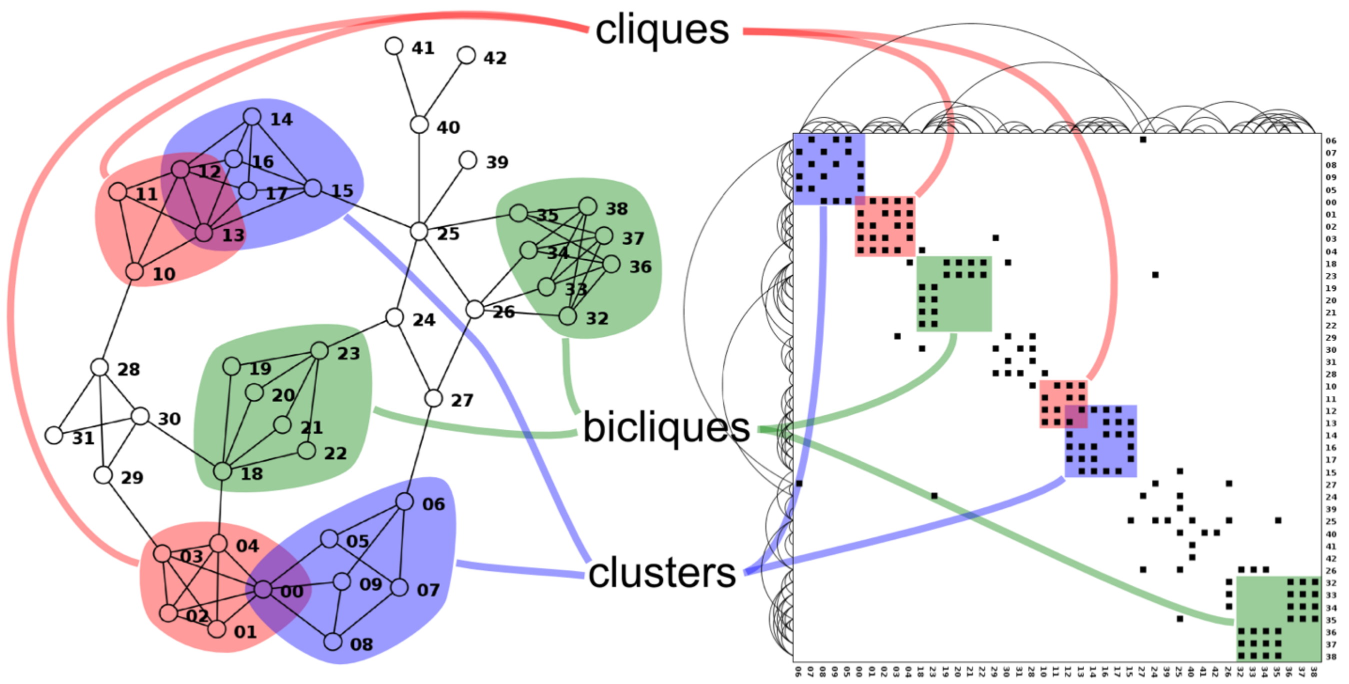


[Points of view: Networks. Gehlenborg and Wong. Nature Methods 9:115.]

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!

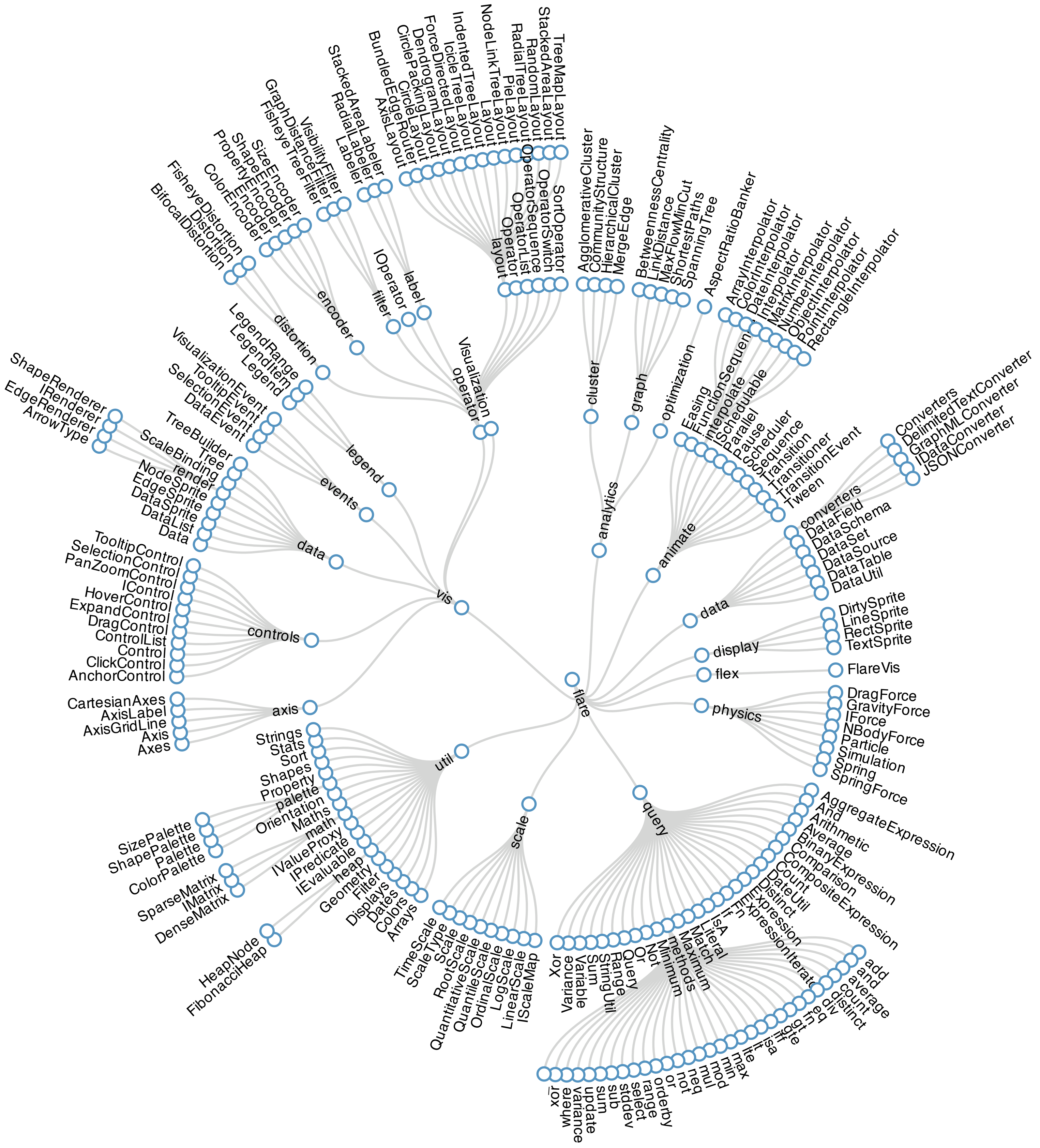
[On the readability of graphs using node-link and matrix-based representations: a controlled experiment and statistical analysis. Ghoniem, Fekete, and Castagliola. Information Visualization 4:2 (2005), 114–135.]



http://www.michaelmcguffin.com/courses/vis/patternsInAdjacencyMatrix.png

Idiom: **radial node-link tree**

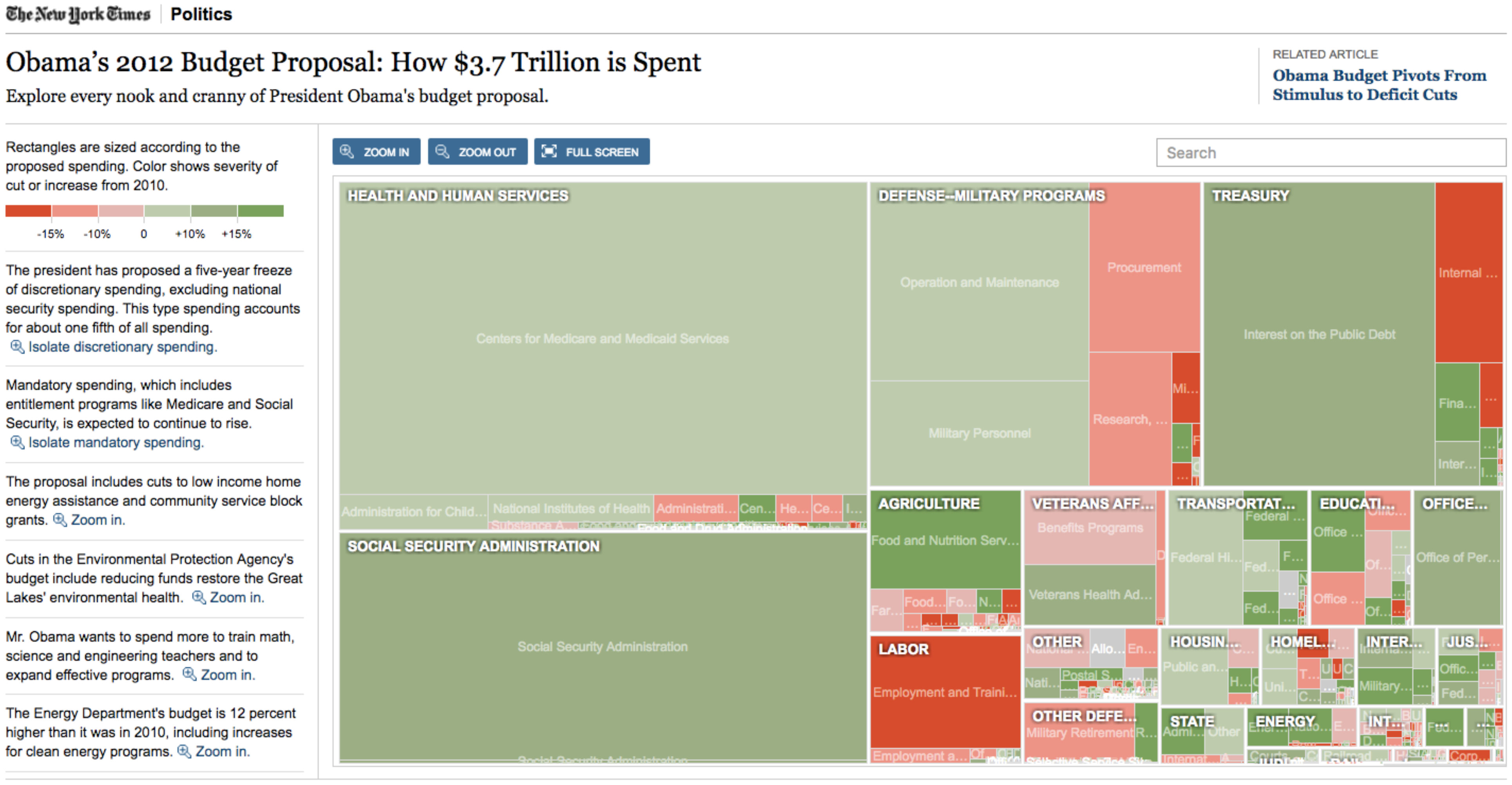
- data
  - tree
- encoding
  - link connection marks
  - point node marks
  - radial axis orientation
    - angular proximity: siblings
    - distance from center: depth in tree
- tasks
  - understanding topology, following paths
- scalability
  - 1K - 10K nodes



http://mbostock.github.com/d3/ex/tree.html

Idiom: **treemap**

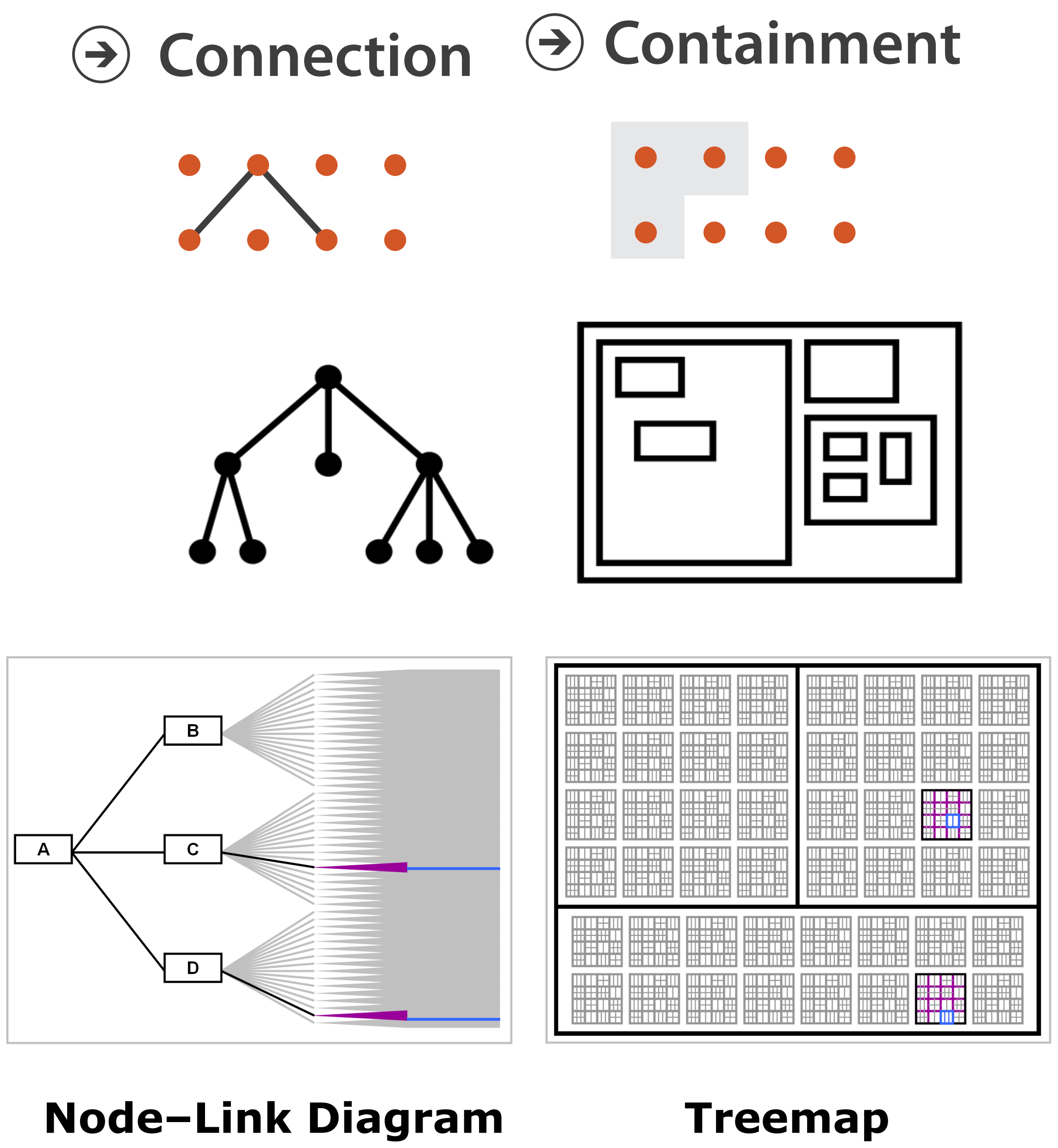
- data
  - tree
  - 1 quant attrib at leaf nodes
- encoding
  - area containment marks for hierarchical structure
  - rectilinear orientation
  - size encodes quant attrib
- tasks
  - query attribute at leaf nodes
- scalability
  - 1M leaf nodes



http://www.nytimes.com/packages/html/newsgraphics/2011/0119-budget/index.html

Link marks: Connection and containment

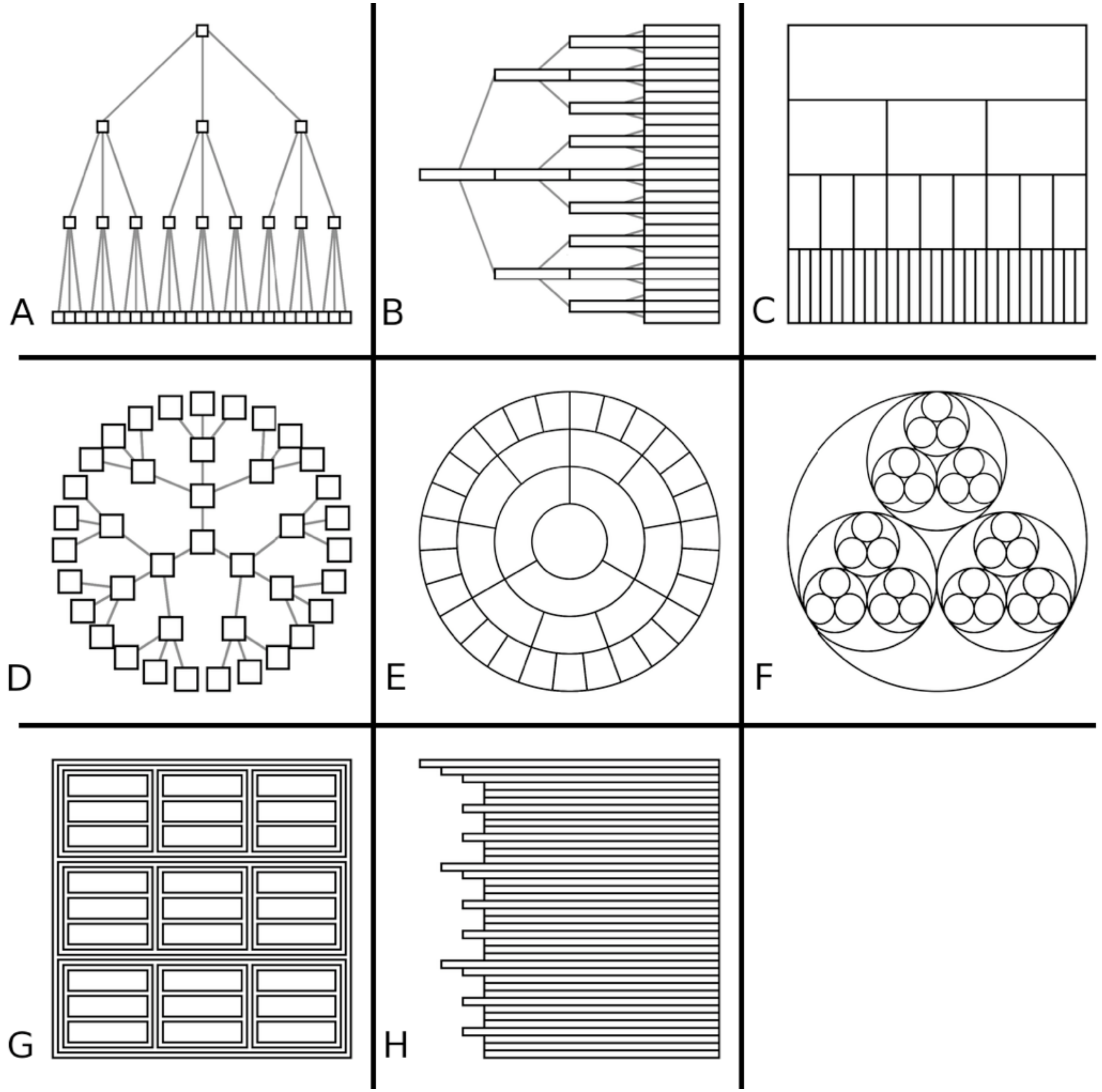
- marks as links (vs. nodes)
  - common case in network drawing
  - 1D case: connection
    - ex: all node-link diagrams
    - emphasizes topology, path tracing
    - networks and trees
  - 2D case: containment
    - ex: all treemap variants
    - emphasizes attribute values at leaves (size coding)
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



[Elastic Hierarchies: Combining Treemaps and Node-Link Diagrams. Dong, McGuffin, and Chignell. Proc. InfoVis 2005, p. 57-64.]

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

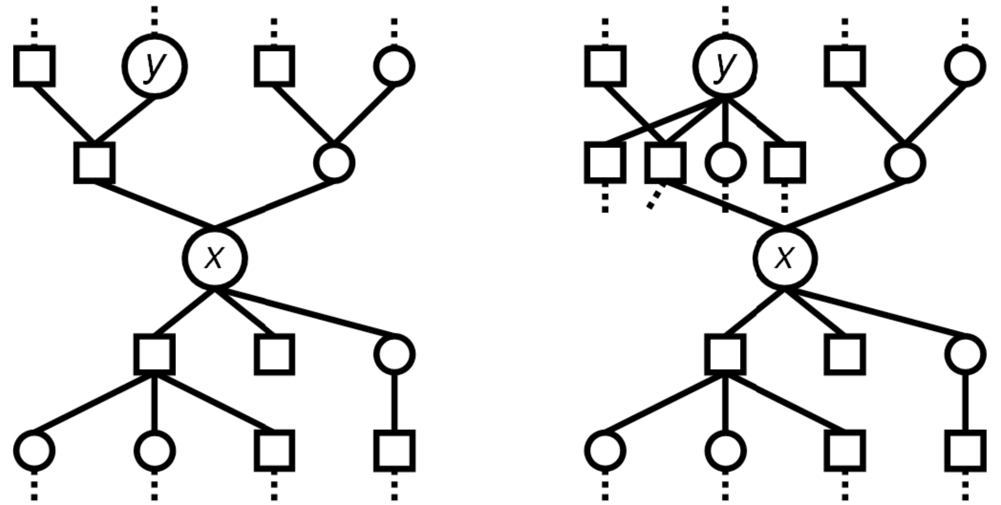


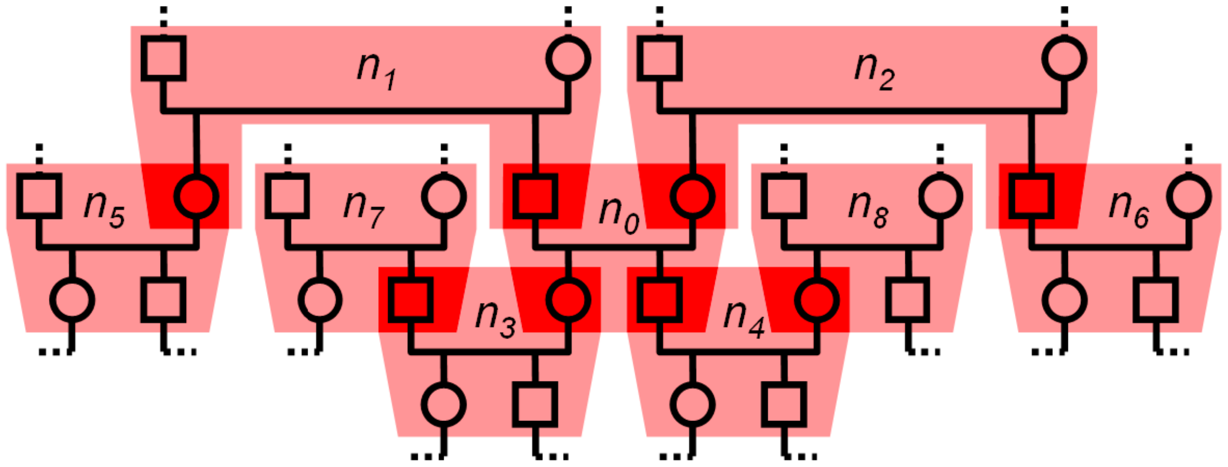
[Quantifying the Space-Efficiency of 2D Graphical Representations of Trees. McGuffin and Robert. Information Visualization 9:2 (2010), 115–140.]

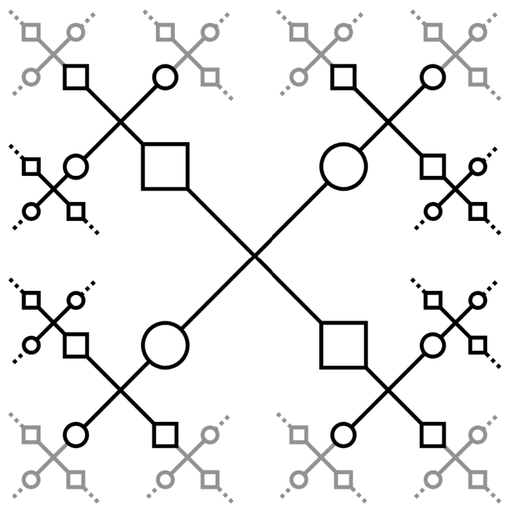
Paper: Genealogical Graphs

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



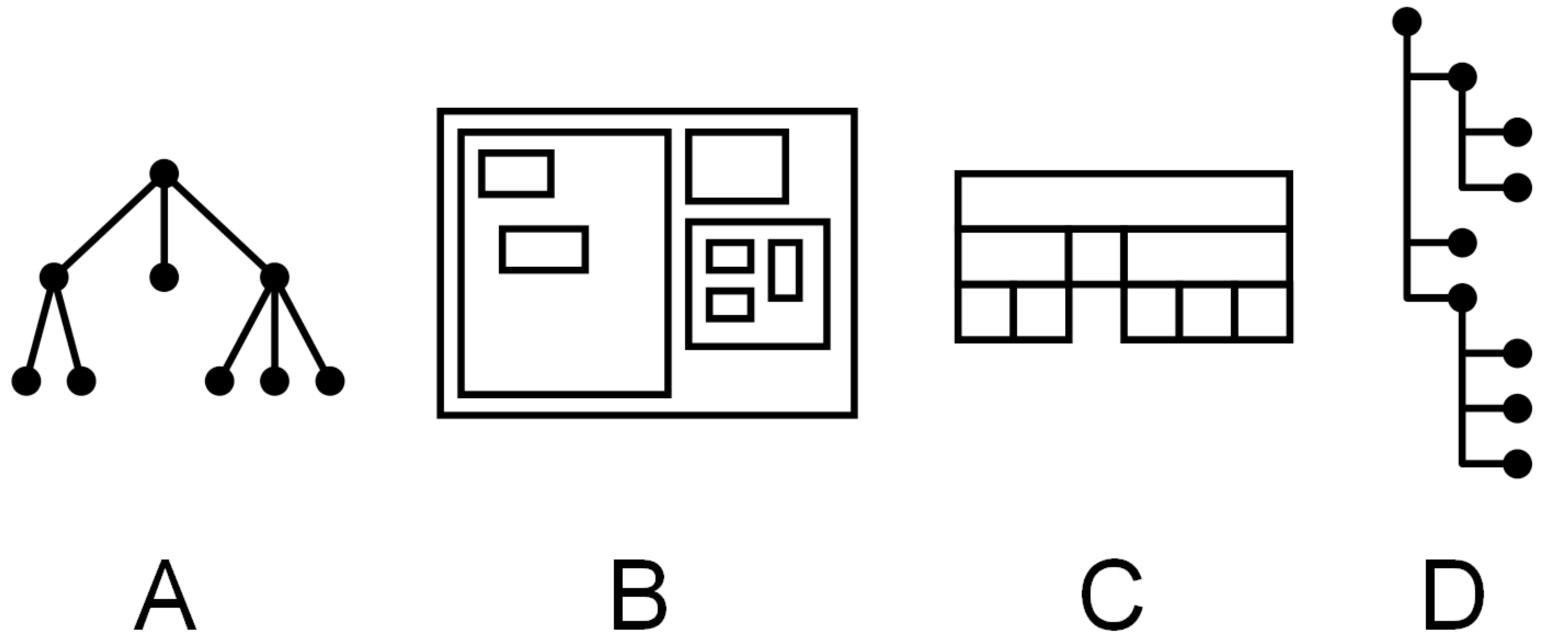




[Fig 2, 6, 7. Interactive Visualization of Genealogical Graphs. Michael J. McGuffin, Ravin Balakrishnan. Proc. InfoVis 2005, pp 17-24.]

Layouts

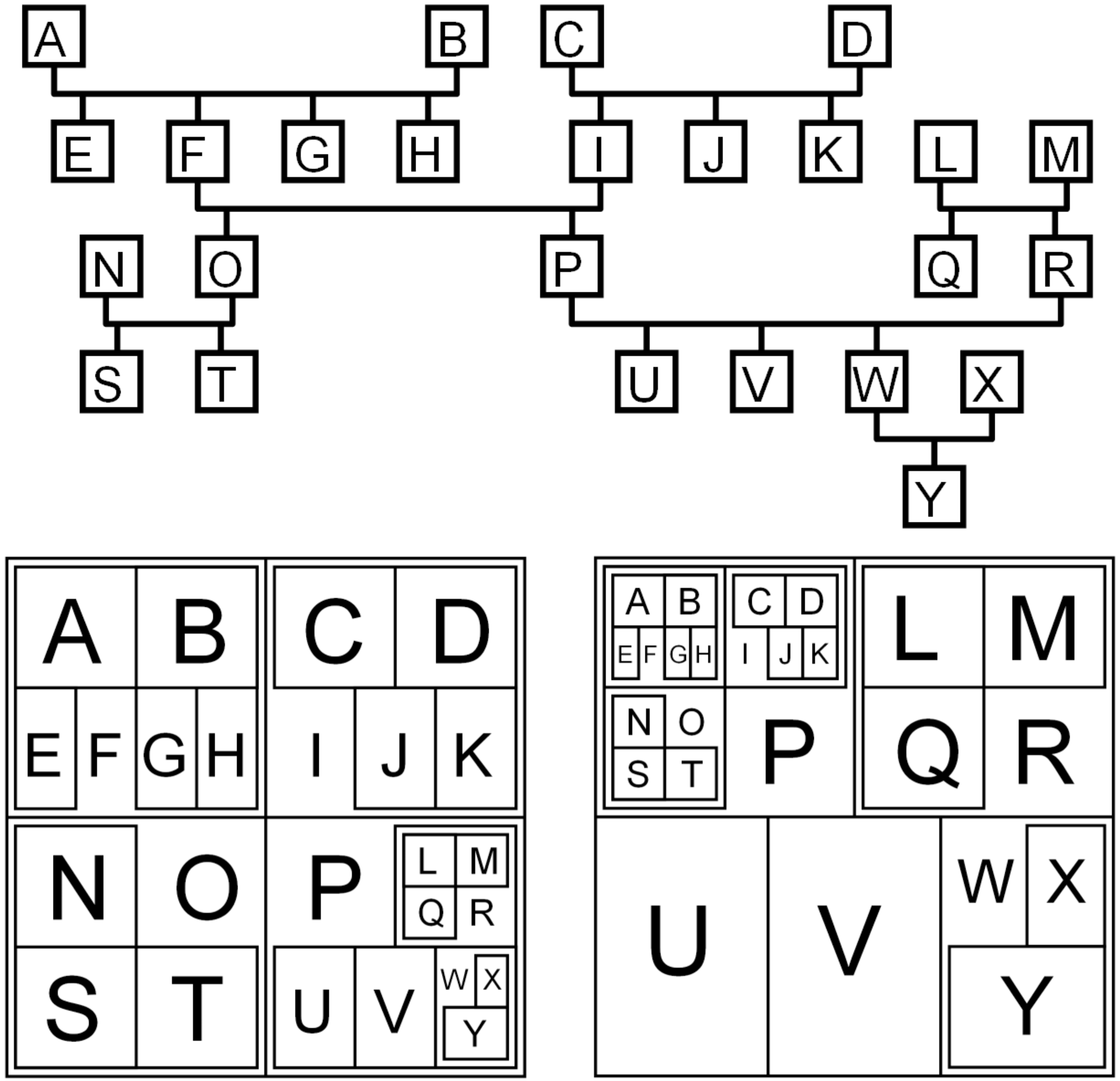
- rooted trees: standard layouts
  - connection
  - containment
  - adjacent aligned position
  - indented position



[Fig 8. Interactive Visualization of Genealogical Graphs. Michael J. McGuffin, Ravin Balakrishnan. Proc. InfoVis 2005, pp 17-24.]

Layouts

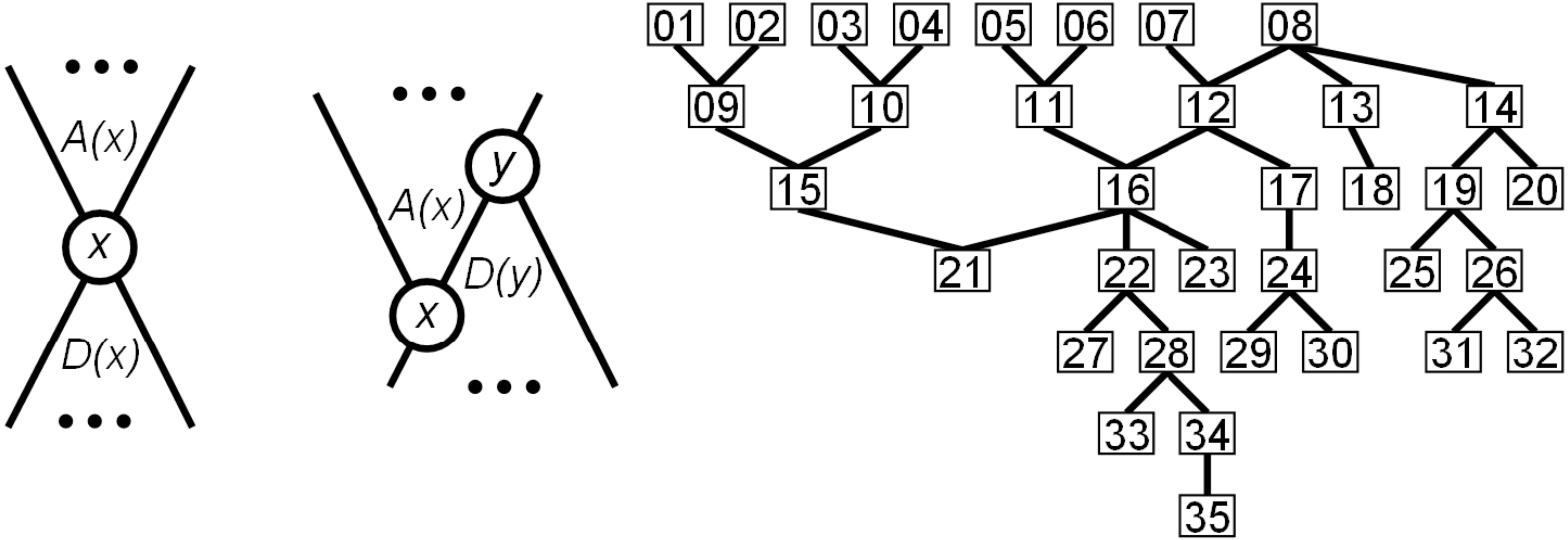
- free trees
  - no root
- adapting rooted methods
  - temporary root for given focus
  - containment (nested)



[Fig 9. Interactive Visualization of Genealogical Graphs. Michael J. McGuffin, Ravin Balakrishnan. Proc. InfoVis 2005, pp 17-24.]

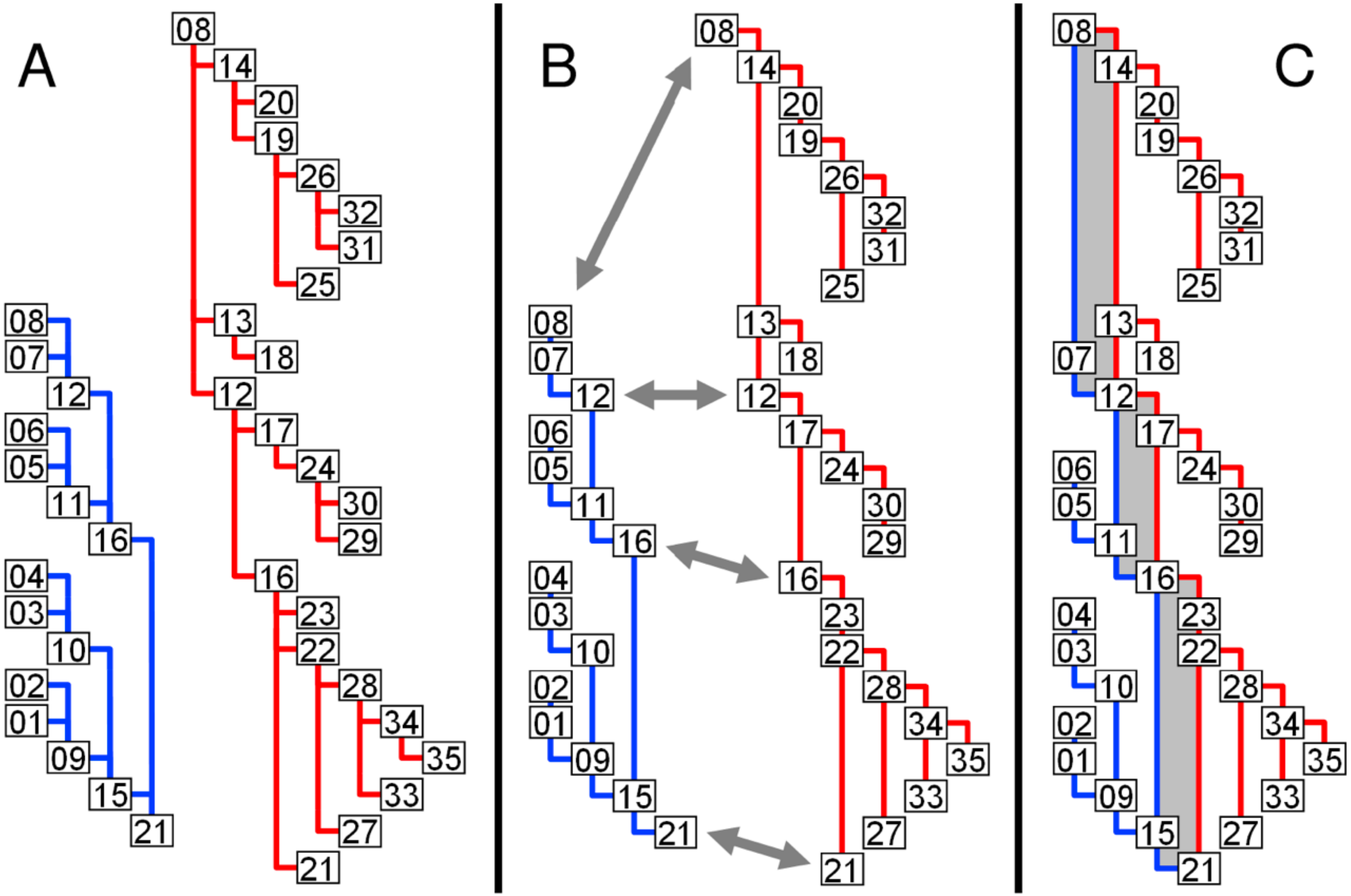
Dual trees abstraction

- 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



[Fig 10. Interactive Visualization of Genealogical Graphs. Michael J. McGuffin, Ravin Balakrishnan. Proc. InfoVis 2005, pp 17-24.]

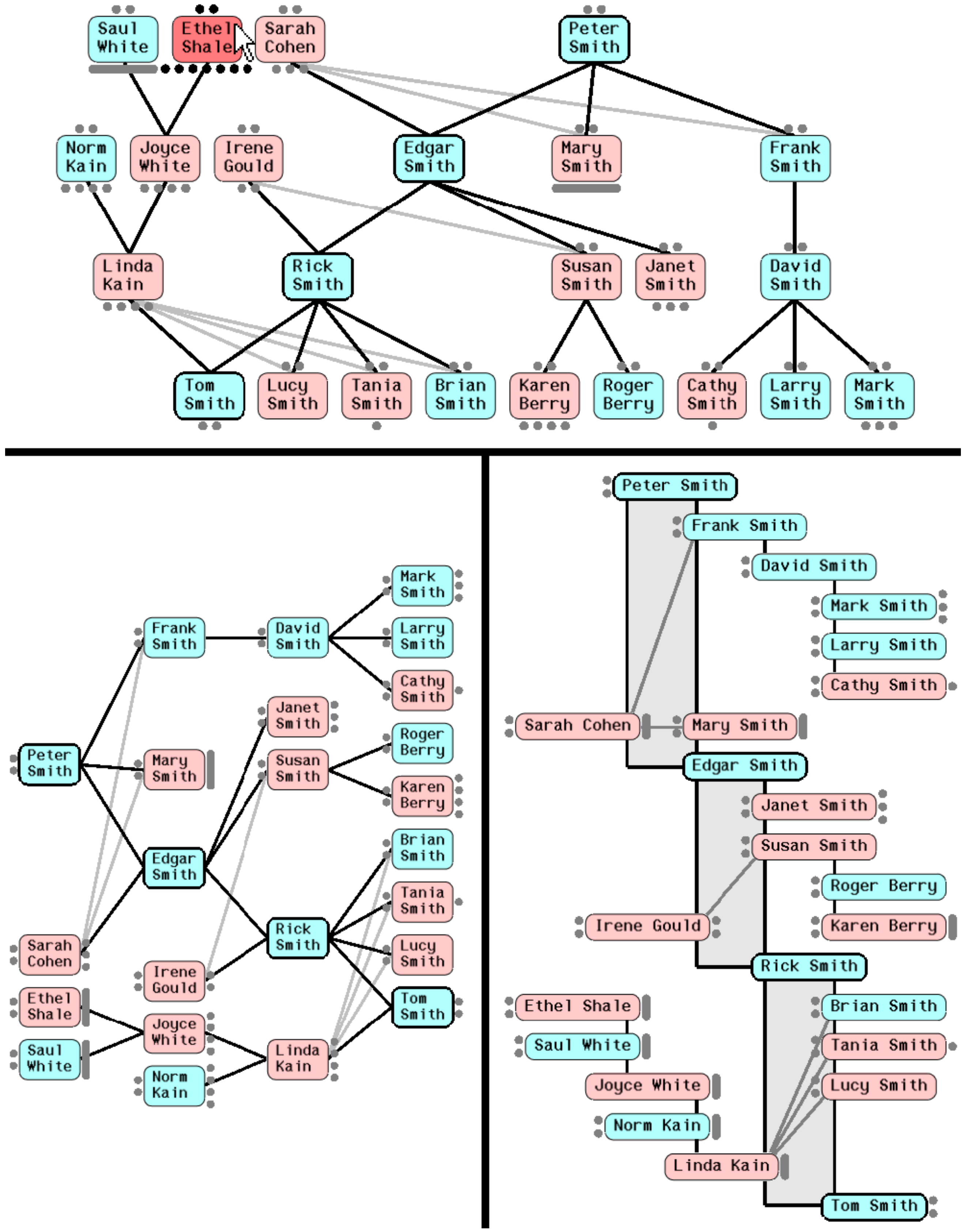
Indented, flipped, combined



[Fig 11. Interactive Visualization of Genealogical Graphs. Michael J. McGuffin, Ravin Balakrishnan. Proc. InfoVis 2005, pp 17-24.]

Another example

- vertical connection
- horizontal connection
- indented



[Fig 13. Interactive Visualization of Genealogical Graphs. Michael J. McGuffin, Ravin Balakrishnan. Proc. InfoVis 2005, pp 17-24.]

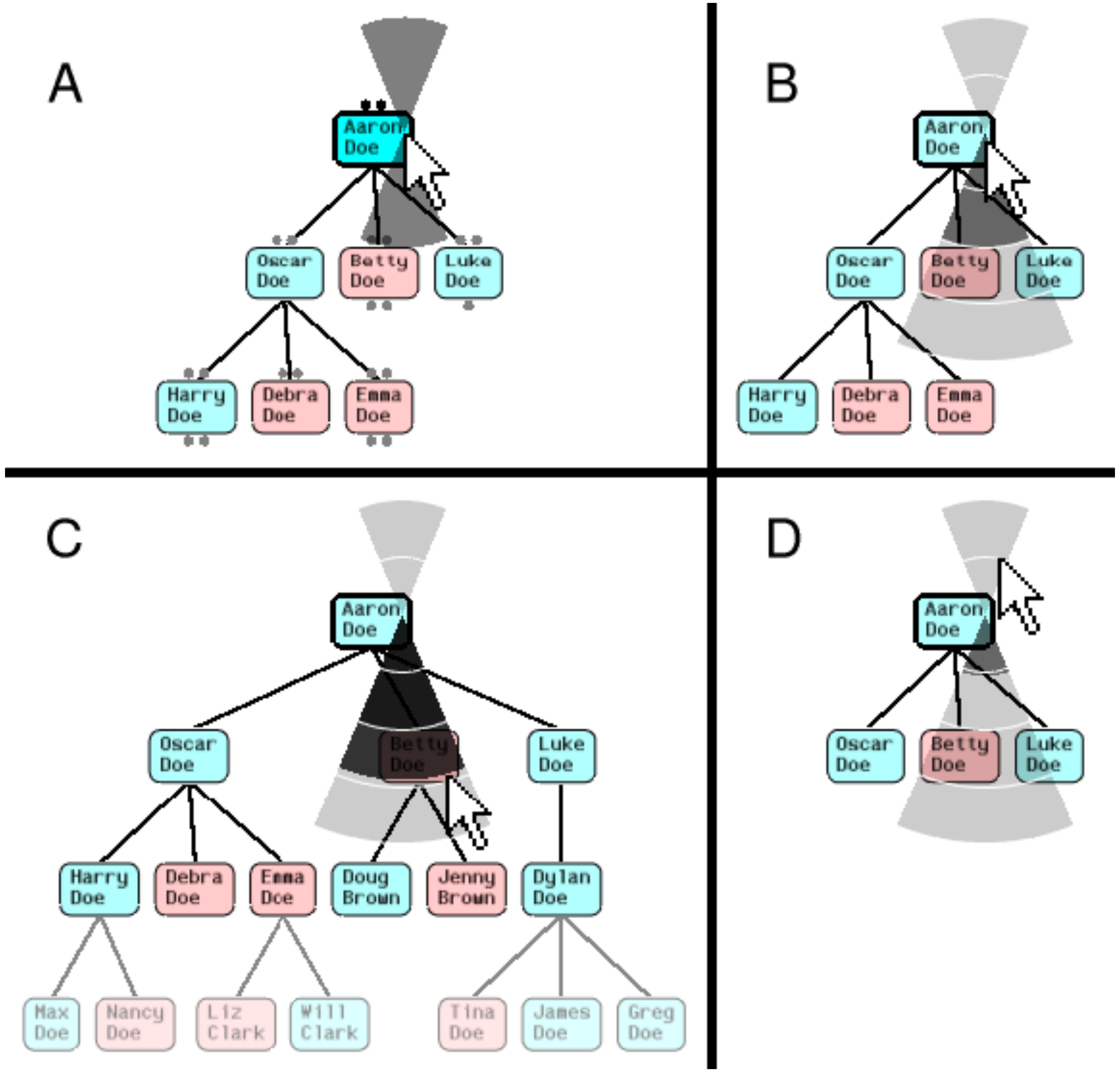
Interaction as fundamental to design

- 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

[Fig 14. Interactive Visualization of Genealogical Graphs. Michael J. McGuffin, Ravin Balakrishnan. Proc. InfoVis 2005, pp 17-24.]

Custom widget

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



[Fig 14. Interactive Visualization of Genealogical Graphs. Michael J. McGuffin, Ravin Balakrishnan. Proc. InfoVis 2005, pp 17-24.]

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

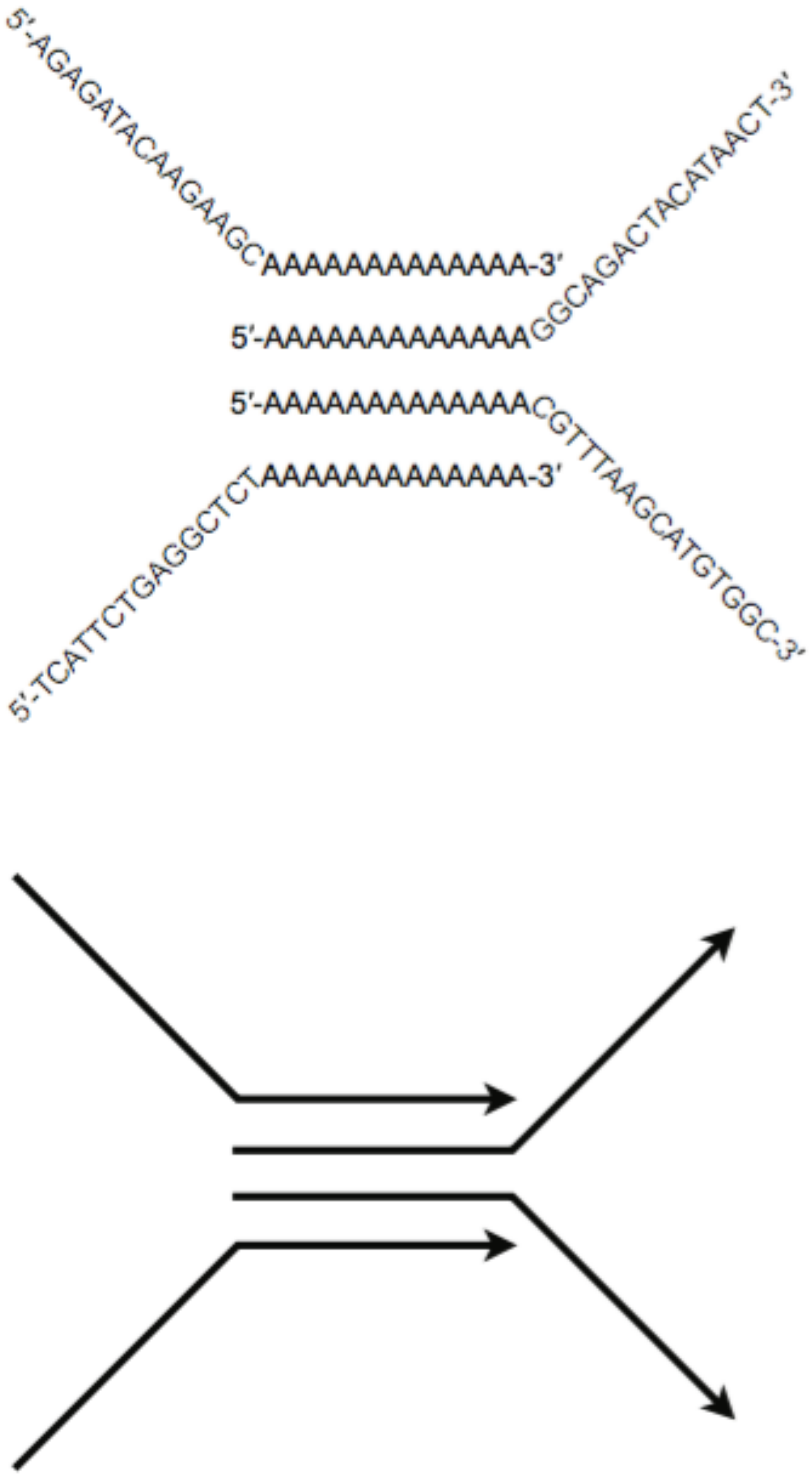


Fig 2. ABySS-Explorer: visualizing genome sequence assemblies. Nielsen, Jackman, Birol, Jones. TVCG 15(6):881-8, 2009 (Proc. InfoVis 2009).

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

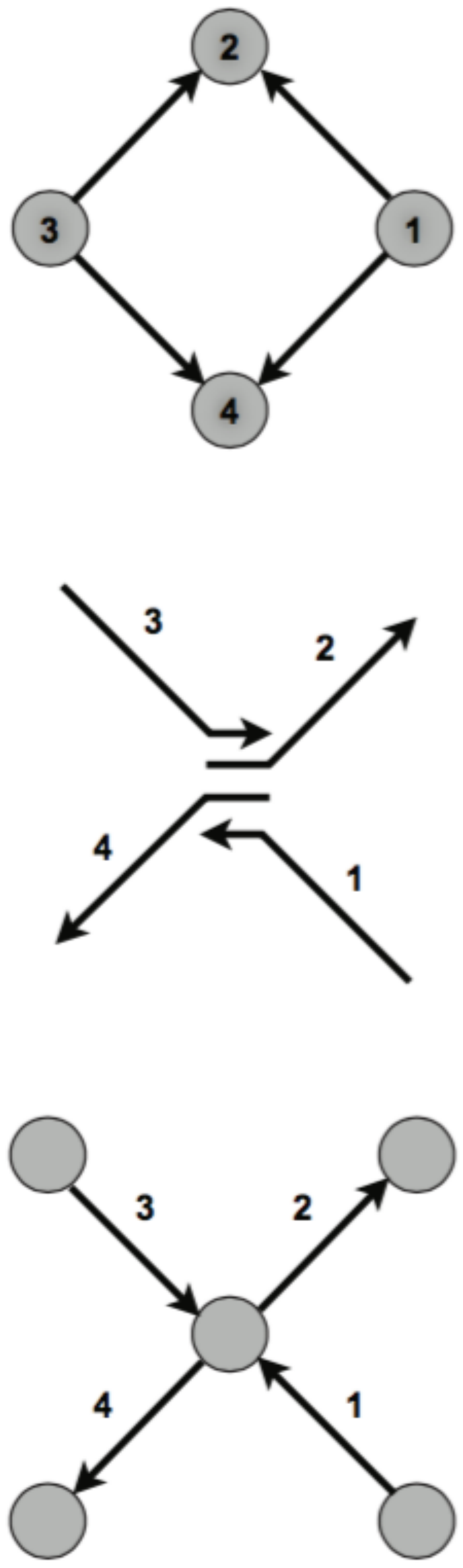


Fig 3. ABySS-Explorer: visualizing genome sequence assemblies. Nielsen, Jackman, Birol, Jones. TVCG 15(6):881-8, 2009 (Proc. InfoVis 2009).

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

Fig 4. ABySS-Explorer: visualizing genome sequence assemblies. Nielsen, Jackman, Birol, Jones. TVCG 15(6):881-8, 2009 (Proc. InfoVis 2009).

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

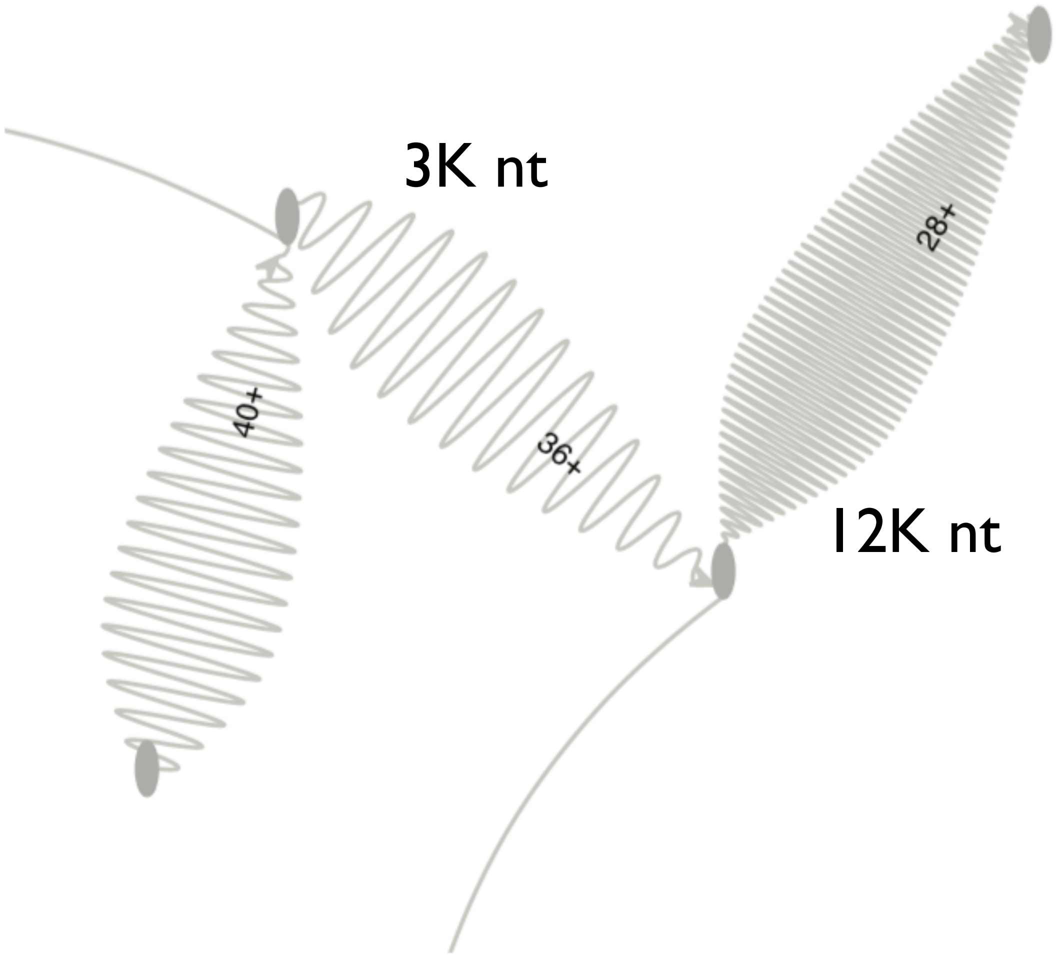


Fig 5. ABySS-Explorer: visualizing genome sequence assemblies. Nielsen, Jackman, Birol, Jones. TVCG 15(6):881-8, 2009 (Proc. InfoVis 2009).

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

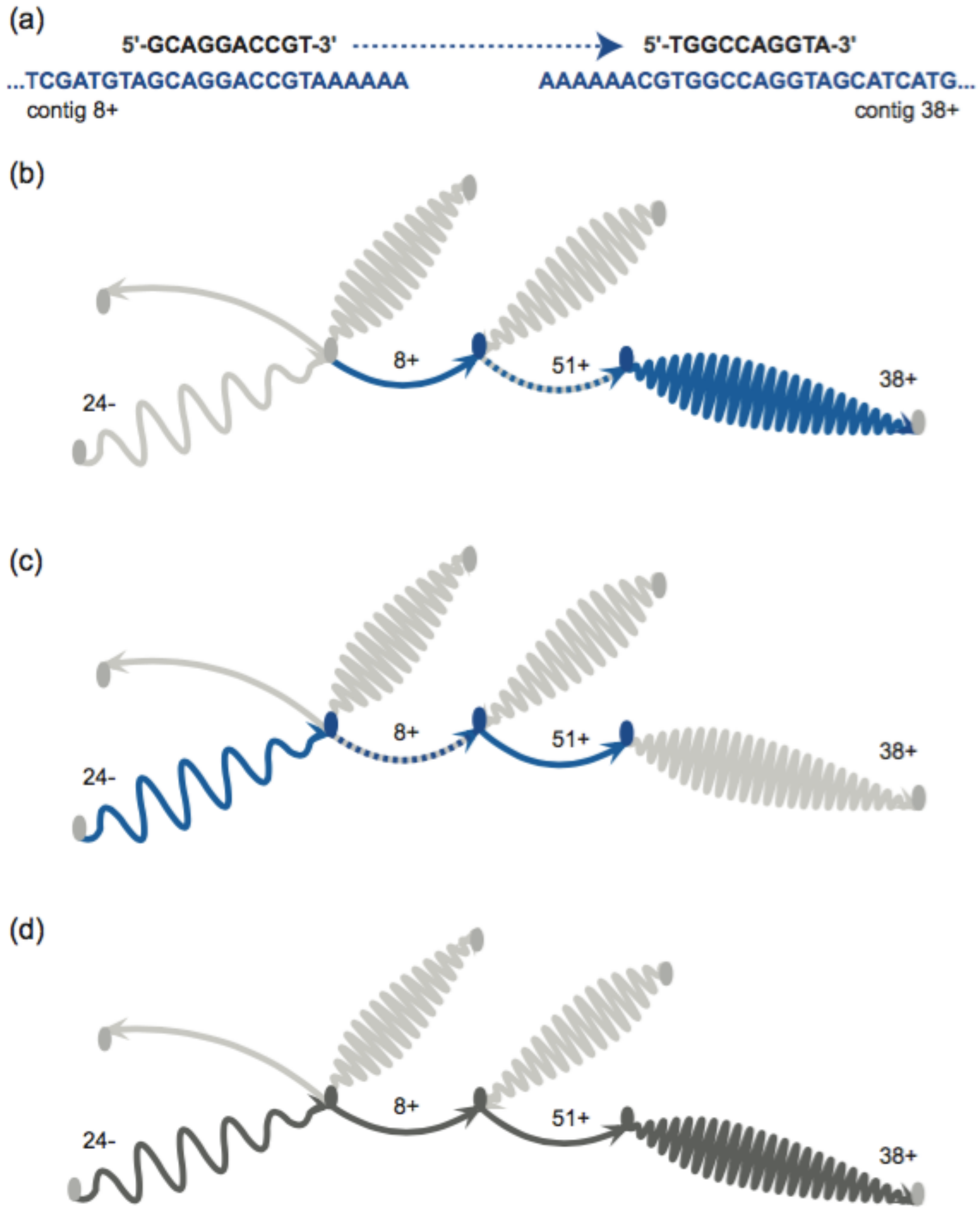
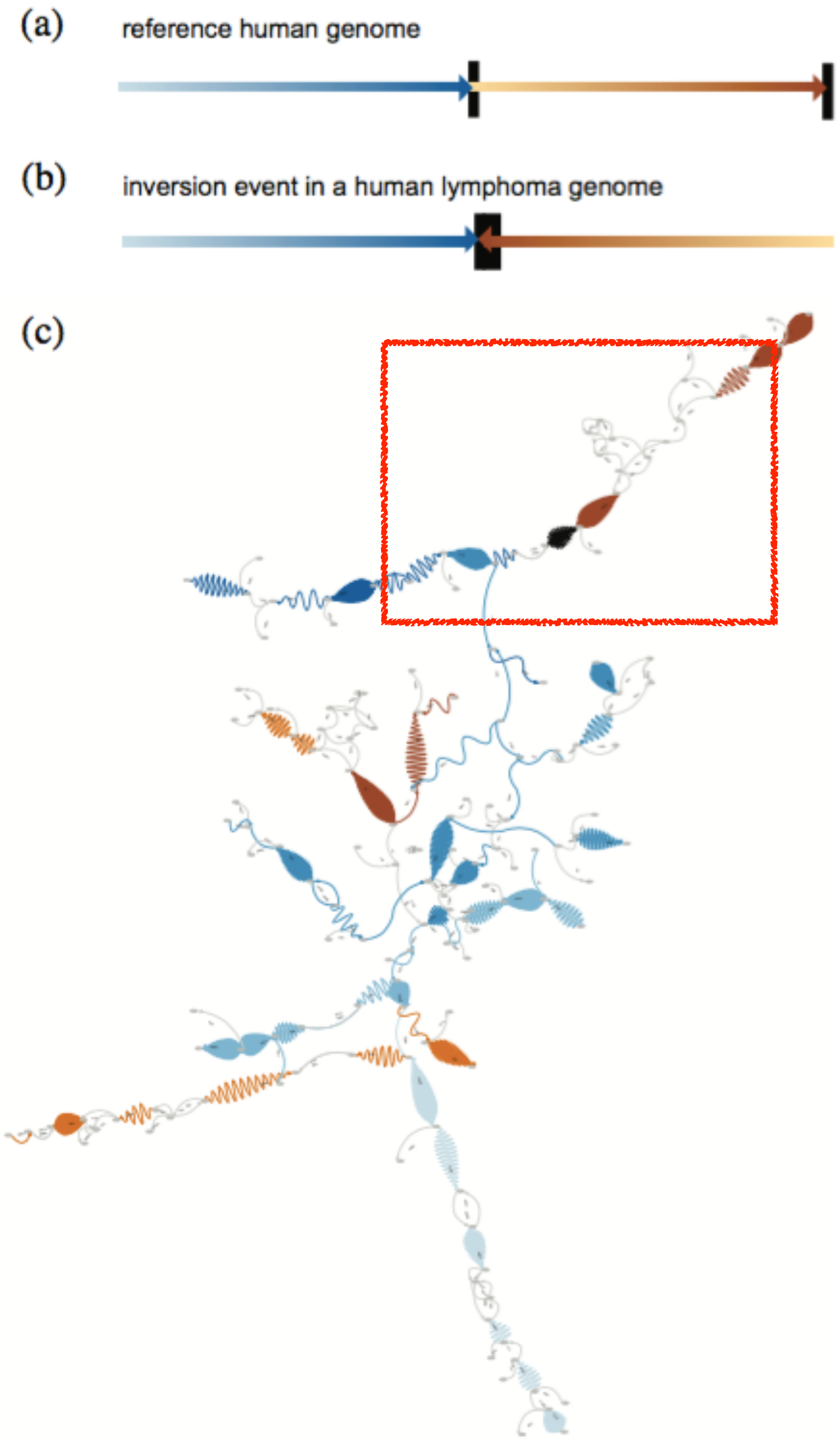


Fig 6. ABySS-Explorer: visualizing genome sequence assemblies. Nielsen, Jackman, Birol, Jones. TVCG 15(6):881-8, 2009 (Proc. InfoVis 2009).

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



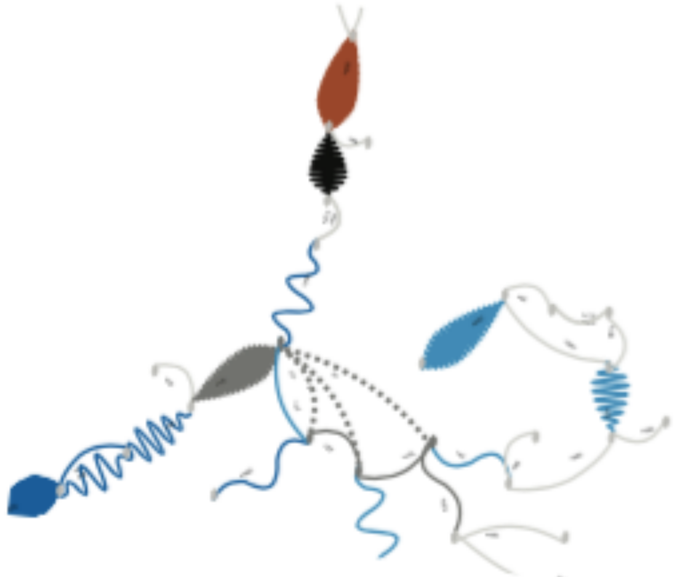


Fig 10. ABySS-Explorer: visualizing genome sequence assemblies. Nielsen, Jackman, Birol, Jones. TVCG 15(6):881-8, 2009 (Proc. InfoVis 2009).

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

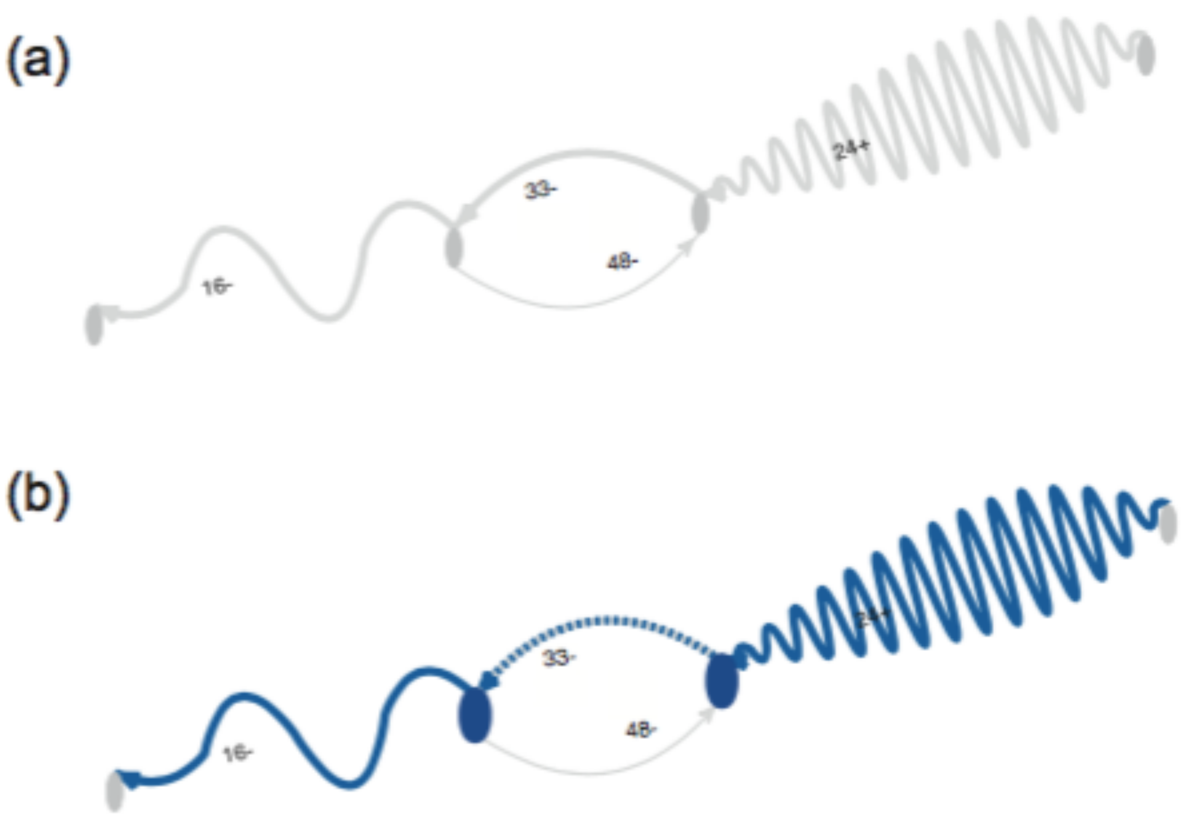


Fig 7/9. ABySS-Explorer: visualizing genome sequence assemblies. Nielsen, Jackman, Birol, Jones. TVCG 15(6):881-8, 2009 (Proc. InfoVis 2009).