12: Facet Across Multiple Views

Enrico Puppo
Department of Computer Science, Bioengineering, Robotics and Systems Engineering
University of Genova

90529 Data Visualization
3-7 December 2020

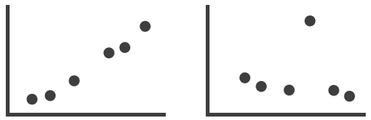
https://2020.aulaweb.unige.it/course/view.php?id=4293

Credits:
material in these slides is partially taken from
- T. Munzner, University of British Columbia
- A. Lex, University of Utah

Facet

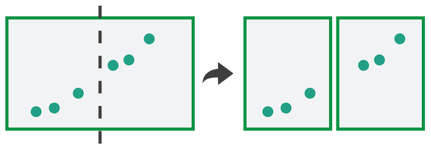
Ways of combining views:

Juxtapose



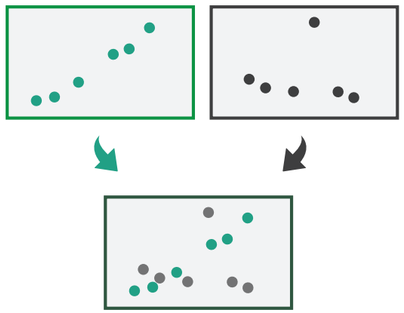
- Place different views side by side

Partition



- Use different views for different portions of data

Superimpose



- Place different data in the same view

Juxtapose and coordinate views

- Share Encoding: Same/Different
  - *Linked Highlighting*

- Share Data: All/Subset/None

- Share Navigation

Various combinations of

- encoding within views
- Boolean relations between datasets
- coordination between views in highlighting and navigation

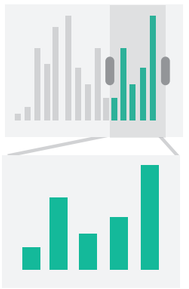
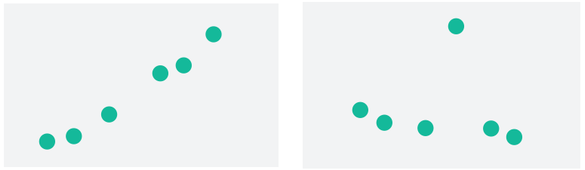
Juxtapose views: Key concepts

- Multiform design:
  - use *different encoding* in different views for the *same data*
  - rationale: single view can show a limited number of attributes simultaneously

- Linked highlight:
  - highlight the same items/attribute/links in the different views
  - easier to locate entities
  - effective to show coherence of neighbourhoods across different embedding spaces

- Small multiples:
  - each view contains a number of items sufficiently small to be clearly read
  - avoids visual cluttering

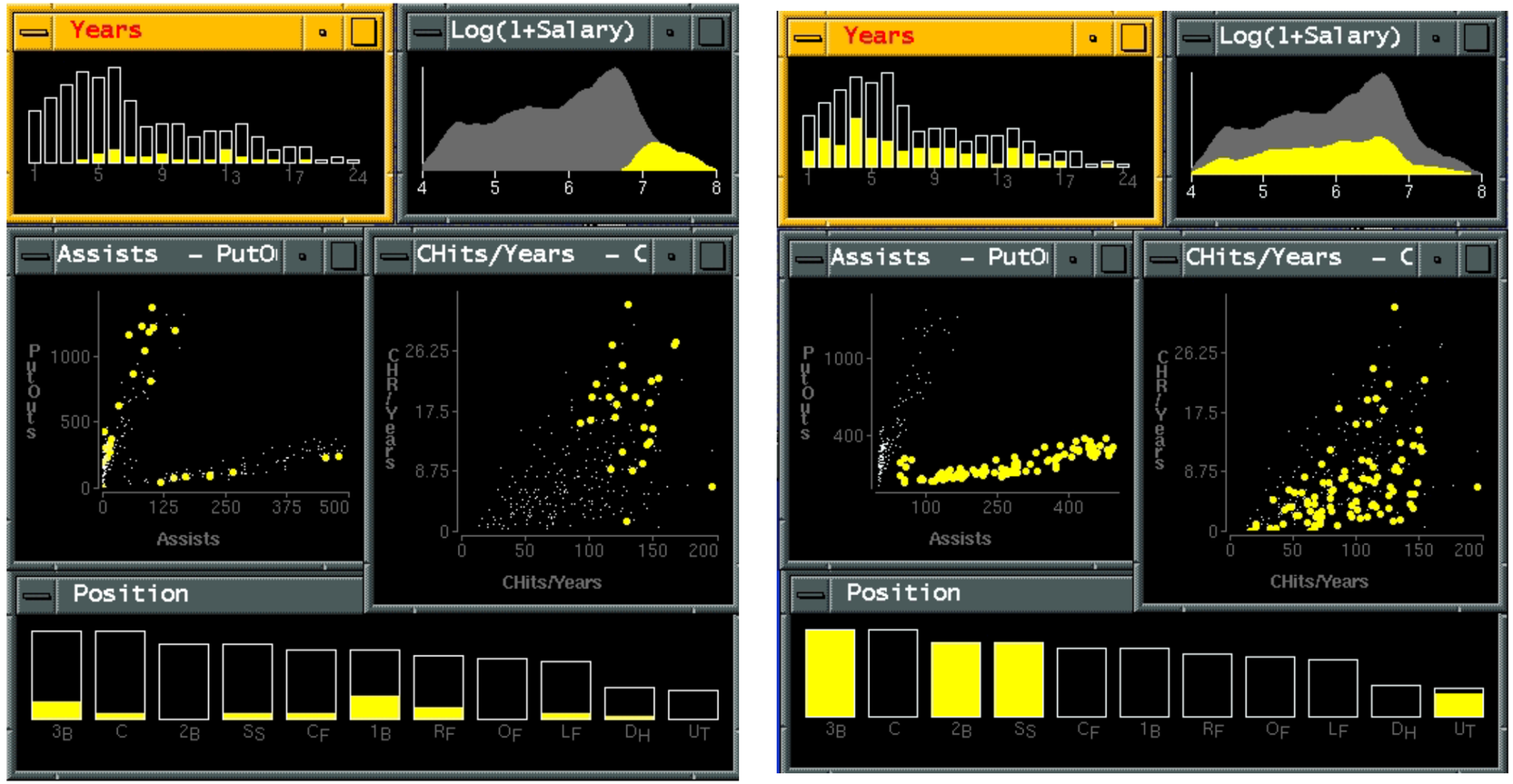
Coordinate views: Design choice interaction

		Data		
		All	Subset	None
Encoding	Same	Redundant	 Overview/ Detail	 Small Multiples
	Different	 Multiform	 Multiform, Overview/ Detail	No Linkage

Idiom: **Linked highlighting**

System: **EDV**

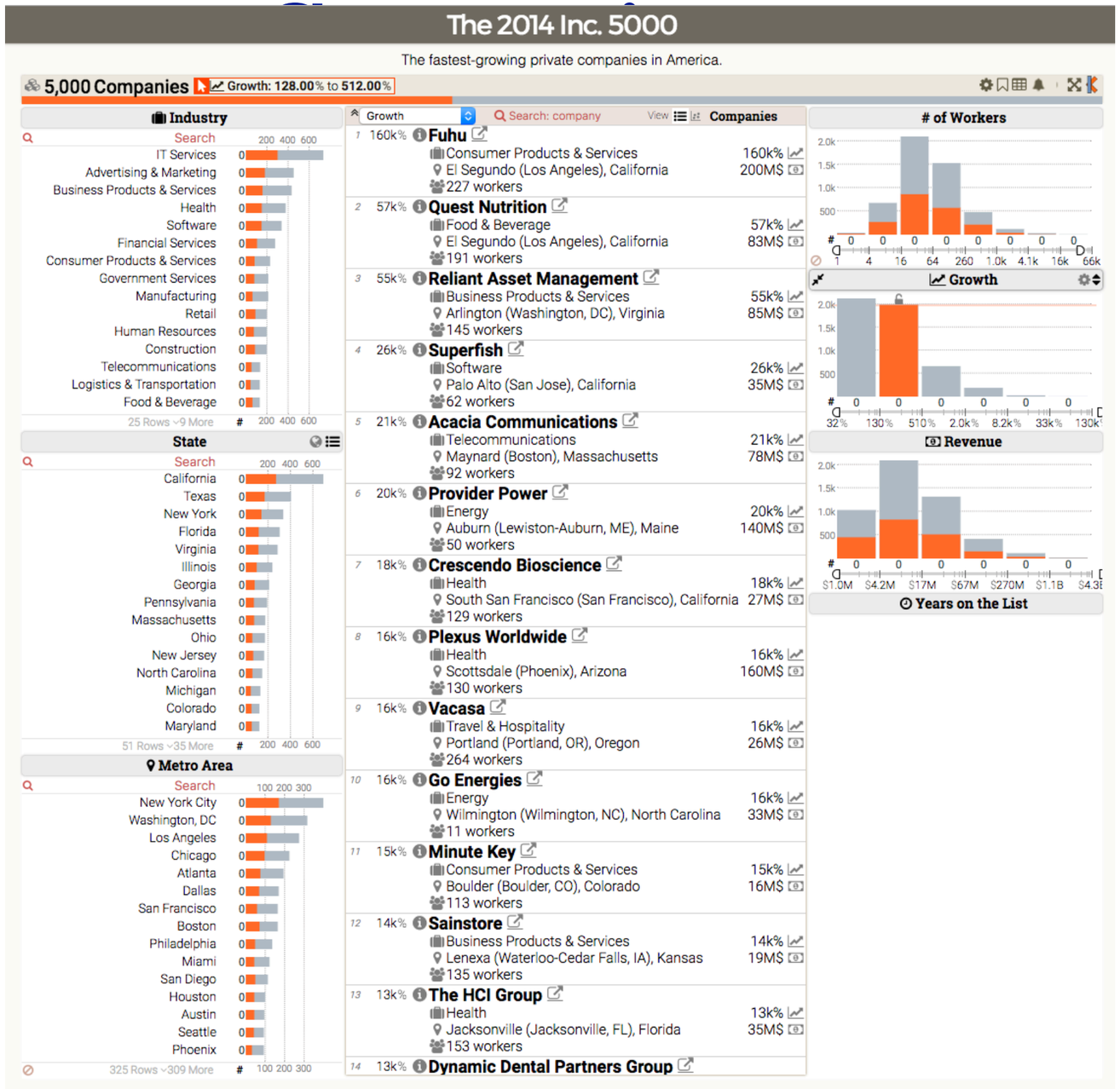
- see how regions contiguous in one view are distributed within another
  - powerful and pervasive interaction idiom
- encoding: different
  - ***multiform***
- data: all shared



[Visual Exploration of Large Structured Datasets. Wills. Proc. New Techniques and Trends in Statistics (NTTS), pp. 237–246. IOS Press, 1995.]

Idiom: **Linked highlighting** System: **5000**

- different views for:
  - detail
  - aggregated data
  - derived data
- encoding: different
  - *multiform*
- data: all shared



https://keshif.me/demo/inc5000

Idiom: **Linked highlighting**

System: **Buckets**

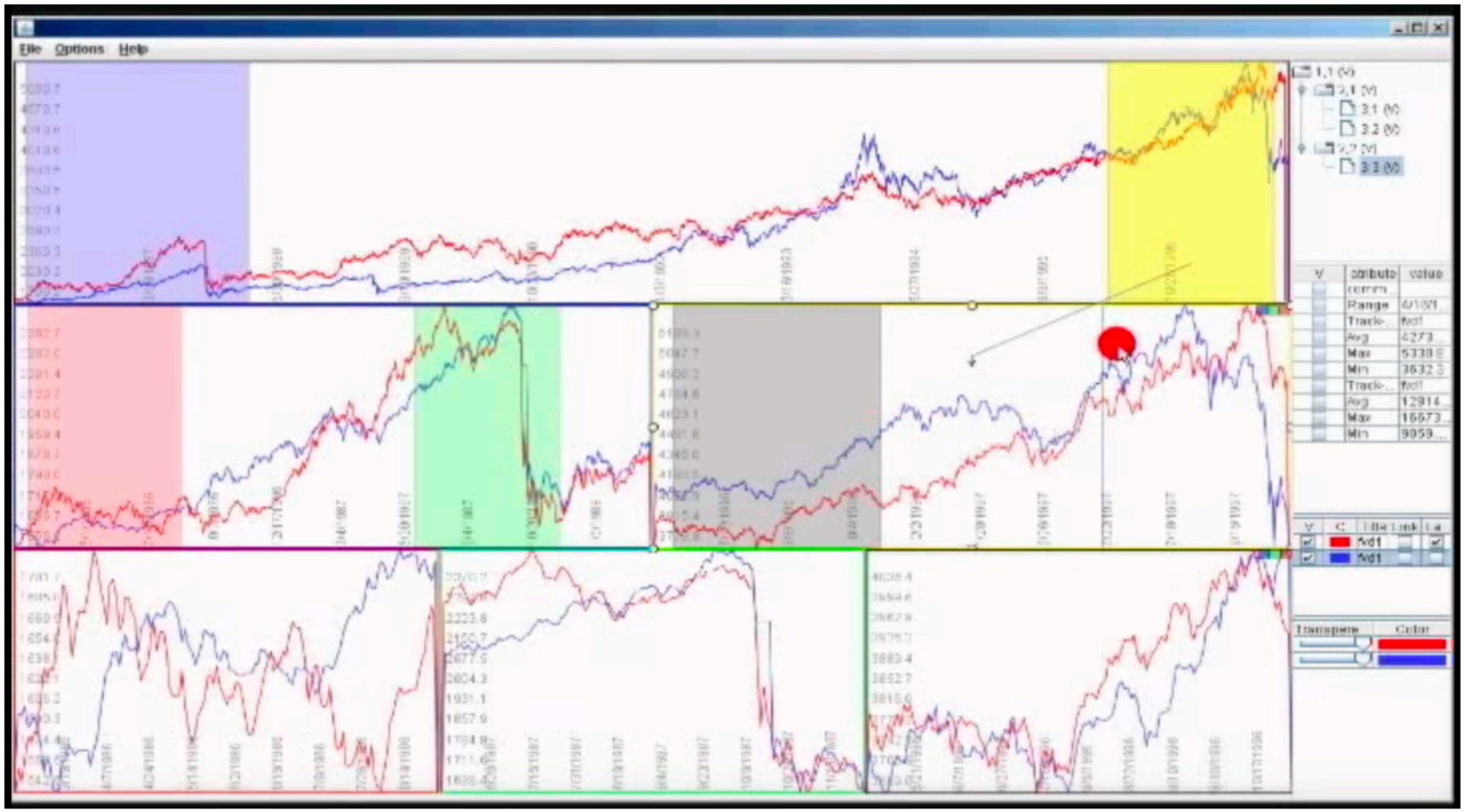
- see correlation of different attributes w.r.t. common key
- encoding: different
  - *multiform*
- data: all shared

buckets.peterbeshai.com/app/#/playerView/201935_2015

Idiom: **Stack zooming**

- Same encoding
- Different resolution
  - **whole - subsets**
- Linked highlight
- Shared navigation

https://www.youtube.com/watch?v=dK0De4XPm5Y



[Waqas Javed and Niklas Elmqvist, Stack zooming for multifocus interaction in time-series data visualization, Proc. IEEE Pacific Vis. Symp., 2010.]

Idiom: **bird's-eye maps**

System: **Google Maps**

- encoding: same
- data: subset shared
- navigation: shared
  - bidirectional linking
- differences
  - viewpoint
  - (size)
- ***overview-detail***

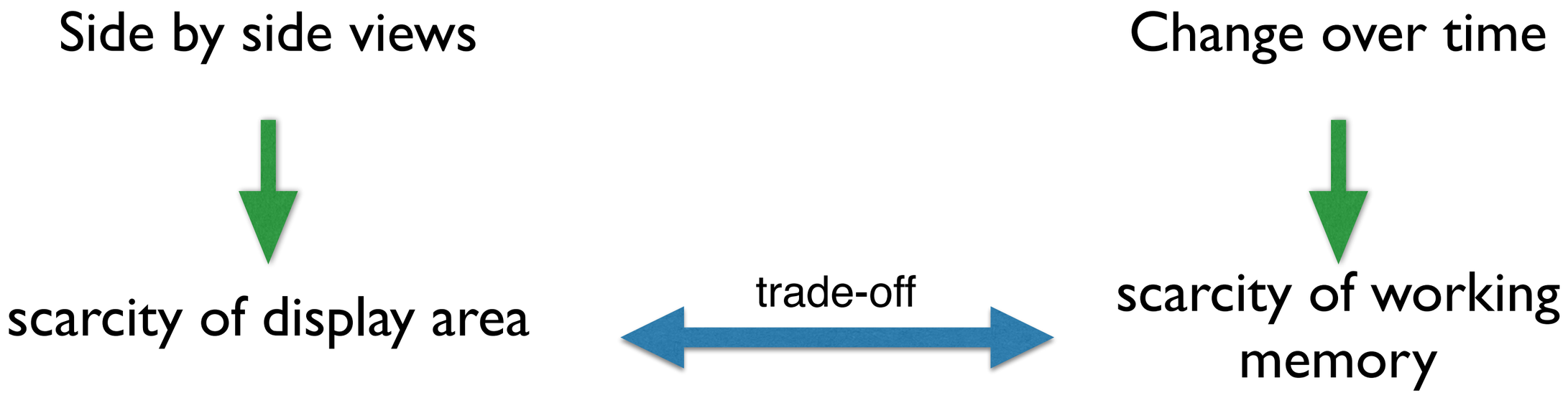
[A Review of Overview+Detail, Zooming, and Focus+Context Interfaces. Cockburn, Karlson, and Bederson. ACM Computing Surveys 41:1 (2008), 1–31.]

Juxtapose views: Design choices

- view count
  - few vs many
    - how many is too many? open research question
- view visibility
  - always side by side vs temporary popups
- view arrangement
  - user managed vs system arranges/aligns
- why juxtapose views?
  - benefits: eyes vs memory
    - lower cognitive load to move eyes between 2 views than remembering previous state with 1 view
  - costs: display area
    - 2 views side by side each have only half the area of 1 view

Juxtapose views vs Dynamic views

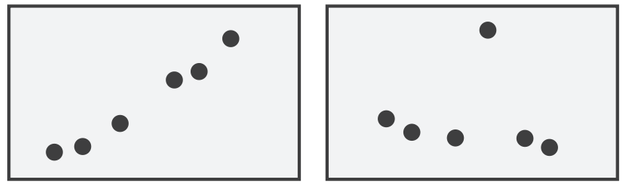
- benefits: eyes vs memory
  - lower cognitive load to move eyes between 2 views than remembering previous state with 1 view
- costs: display area
  - 2 views side by side each have only half the area of 1 view



Partition into views

- separate data into groups
  - how? rationale for partitioning
- generate one view per group
  - encoding?
- arrange views in space
  - order, align
  - flat, hierarchical, recursive

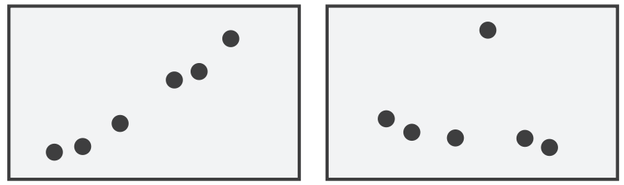
Partition into Side-by-Side Views



Partition into views

- how to divide data between views
  - encodes association between items using spatial proximity
  - major implications for what patterns are visible
  - split according to attributes
- design choices
  - how many splits
    - all the way down: one mark per region?
    - stop earlier, for more complex structure within region?
  - order in which attributes are used to split
  - how many views

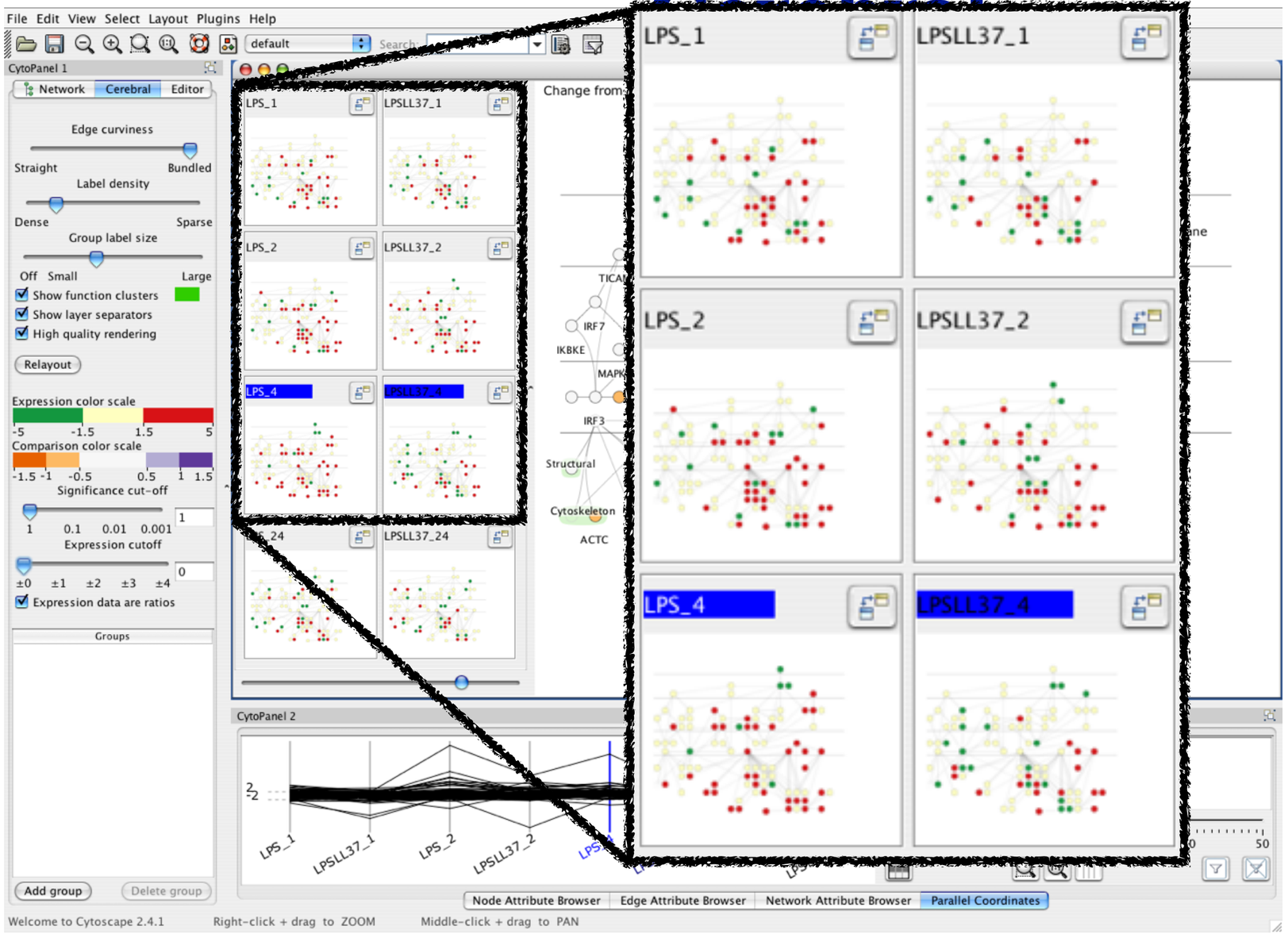
Partition into Side-by-Side Views



Idiom: **Small multiples**

System: Cerebral

- encoding: same
- data: none shared
  - different attributes for node colors
  - (same network layout)
- navigation: shared

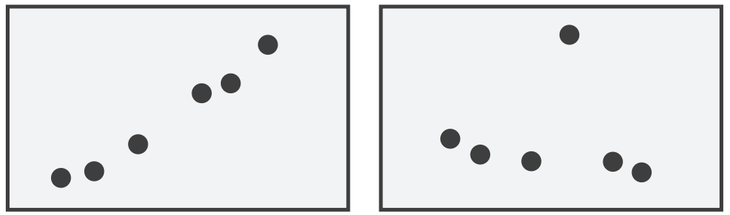


[Cerebral:Visualizing Multiple Experimental Conditions on a Graph with Biological Context. Barsky, Munzner, Gardy, and Kincaid. IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis 2008) 14:6 (2008), 1253–1260.]

Views, glyphs and marks

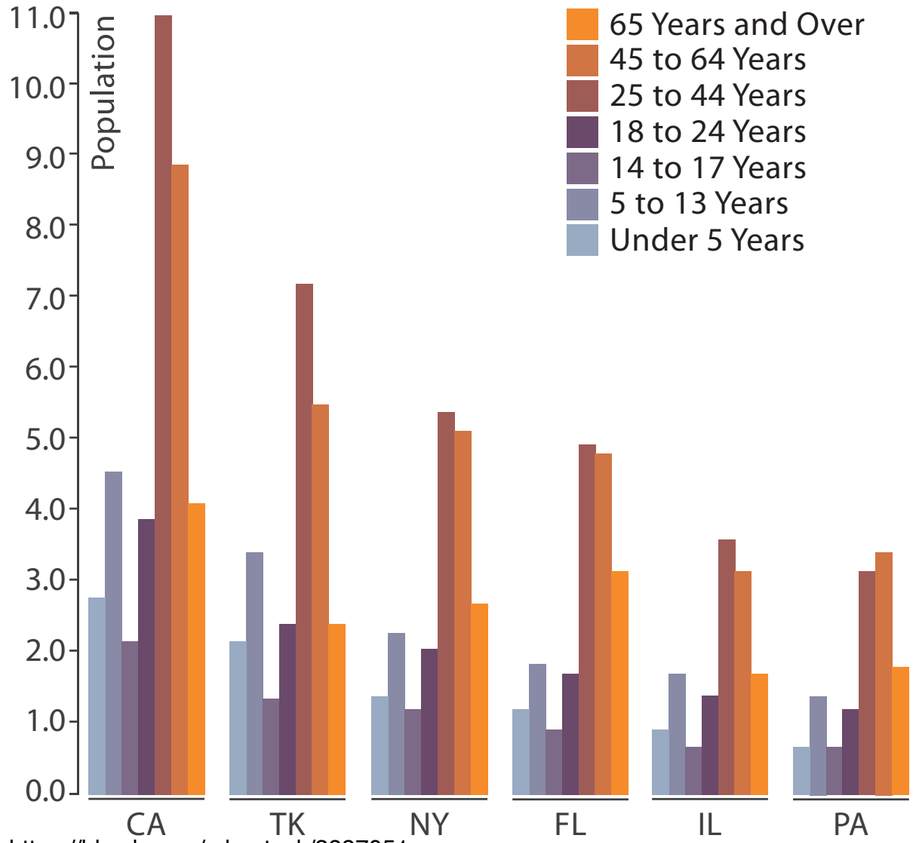
- ***view***
  - contiguous region in which visually encoded data is shown on the display
- ***glyph***
  - object with internal structure that arises from multiple marks
- no strict dividing line
  - view: big/detailed
  - glyph: small/iconic
- ***mark***
  - the simplest glyph

➔ **Partition into Side-by-Side Views**



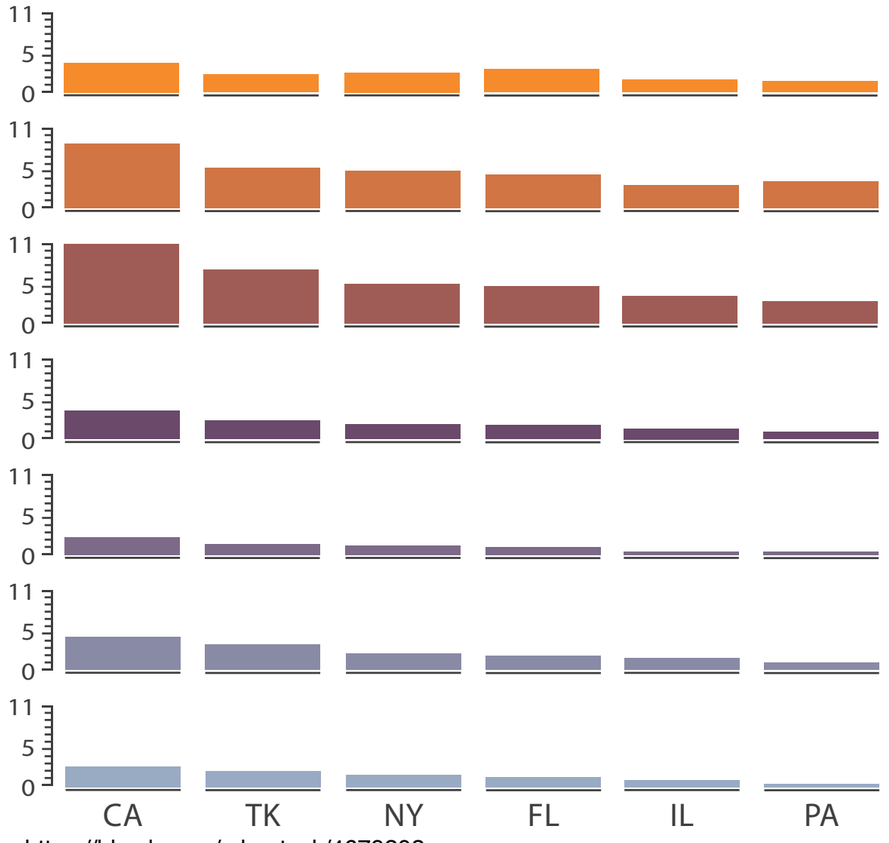
Partitioning: List alignment

- single bar chart with grouped bars
  - split by state into regions
    - complex glyph within each region showing all ages
  - compare: easy within state, hard across ages



https://bl.ocks.org/mbostock/3887051

- small-multiple bar charts
  - split by age into regions
    - one chart per region
  - compare: easy within age, harder across states



https://bl.ocks.org/mbostock/4679202

Multiple views for multiple keys

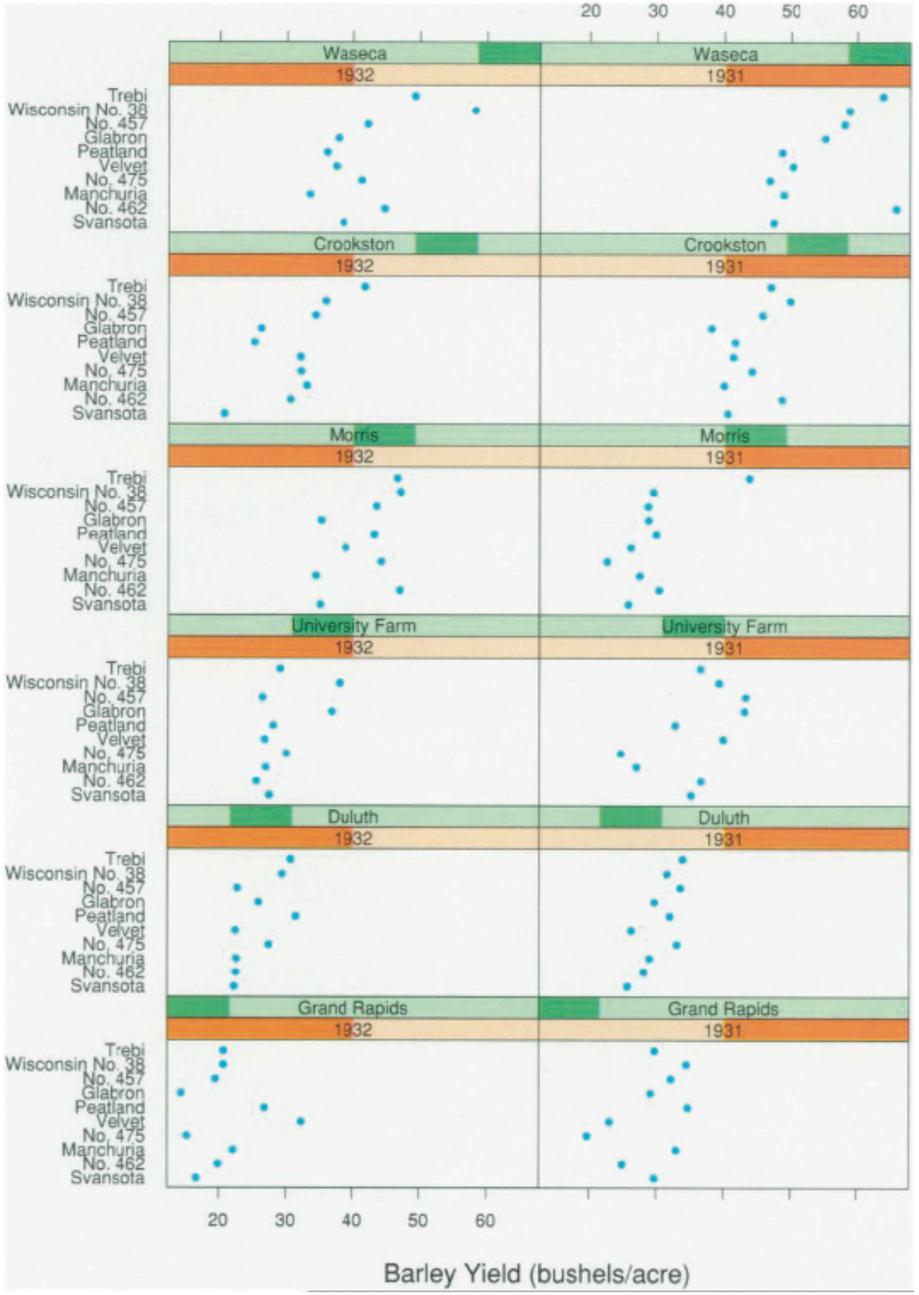
- Datasets with more than 2 keys are difficult to visualize
- Ideas:
  - matrix alignment
    - use some keys to partition data into different views
    - arrange views in matrix form
    - use remaining keys within each view
    - display values
  - recursive partitioning
    - generate as many views as the levels of a selected key
    - arrange them in space
    - within each view, repeat recursively for another key
  - *order of keys* and *arrangement* are crucial choices

Partitioning: matrix arrangement

System: **Trellis**

- 3 keys, 1 value
- 2 keys to set matrix arrangement
- third key indexes rows of a nested dot chart
- values expressed by dot marks on the abscissa

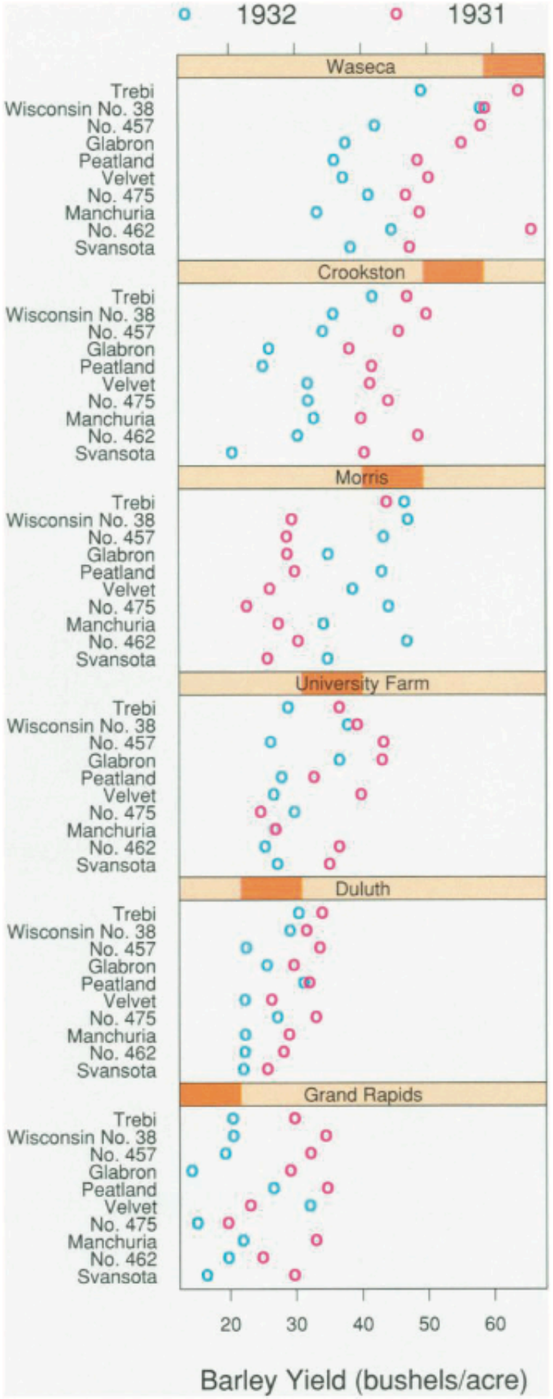
- *Ordering by main-effects* to reveal trends and outliers



[Ronald A. Becker, William S. Cleveland, and Ming-Jen Shyu. "The Visual Design and Control of Trellis Display." Journal of Computational and Statistical Graphics 5:2 (1996), 123–155.]

Partitioning: matrix arrangement

- 3 keys, 1 value
- 1 key to set matrix (column) arrangement
- 1 key indexes rows of a nested dot chart with two layers
- values expressed by dot marks on the abscissa
- third key expressed by hue within the same chart



System: **Trellis**

Alternative arrangement

Partitioning: Recursive subdivision

System: **HIVE**

Example: property transactions over London area in a given period

- 3 keys:
  - type: flat, terrace house, semi-detached house, detached house
  - neighborhood: 33 zones of London
  - time: year and month
- 1 value:
  - price

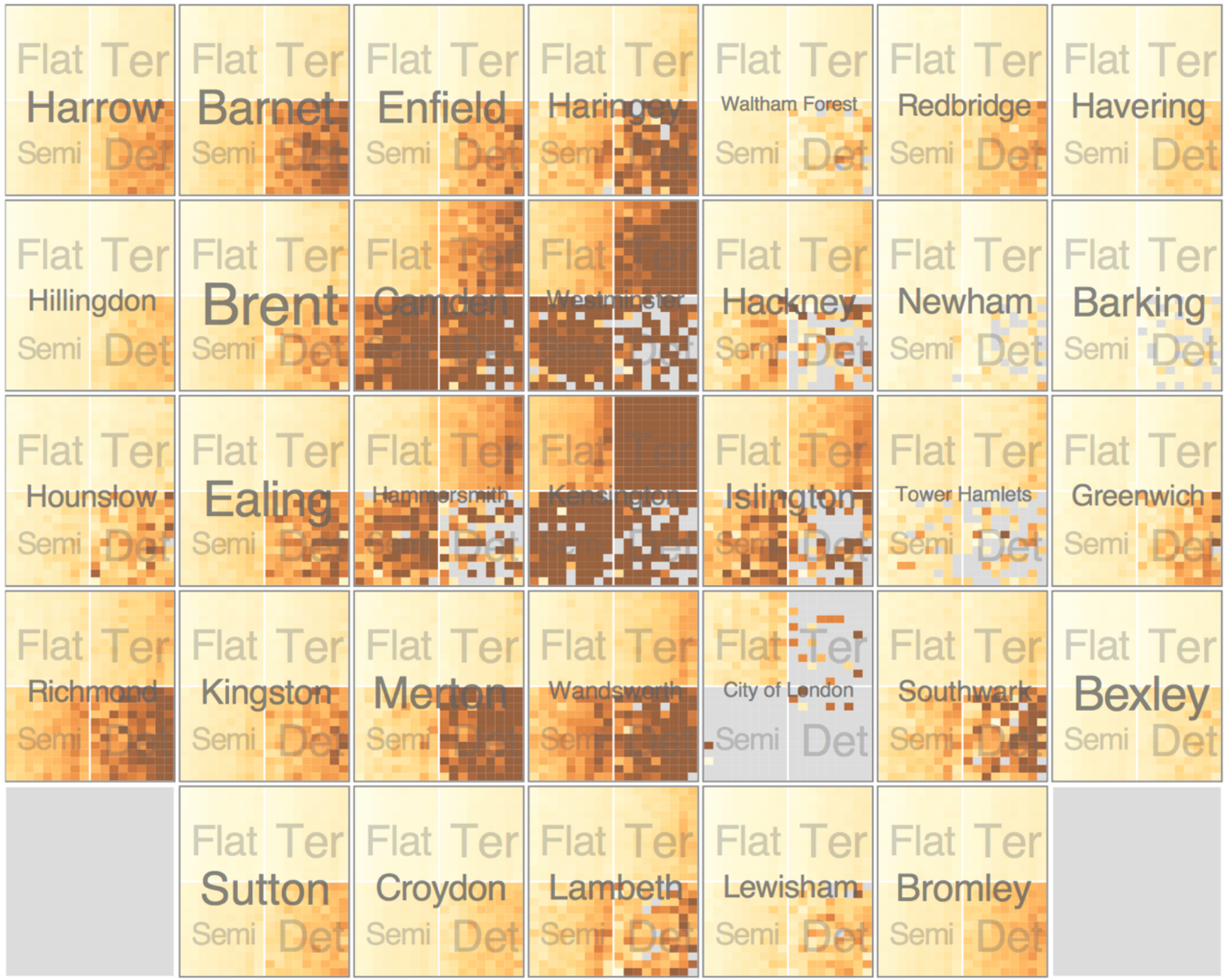
[Configuring Hierarchical Layouts to Address Research Questions. Slingsby, Dykes, and Wood. IEEE Transactions on Visualization and Computer Graphics (Proc. InfoVis 2009) 15:6 (2009), 977–984.]

Partitioning: Recursive subdivision

System: **HIVE**

- split by neighborhood
- then by type
- then time
  - years as rows
  - months as columns
- color by price

- neighborhood patterns
  - where it's expensive
  - where you pay much more for detached type



[Configuring Hierarchical Layouts to Address Research Questions. Slingsby, Dykes, and Wood. IEEE Transactions on Visualization and Computer Graphics (Proc. InfoVis 2009) 15:6 (2009), 977–984.]

Partitioning: Recursive subdivision

System: **HIVE**

- switch order of splits
  - type then neighborhood
- switch color
  - by price variation
- type patterns
  - within specific type, which neighborhoods inconsistent

Alternative interpretation:

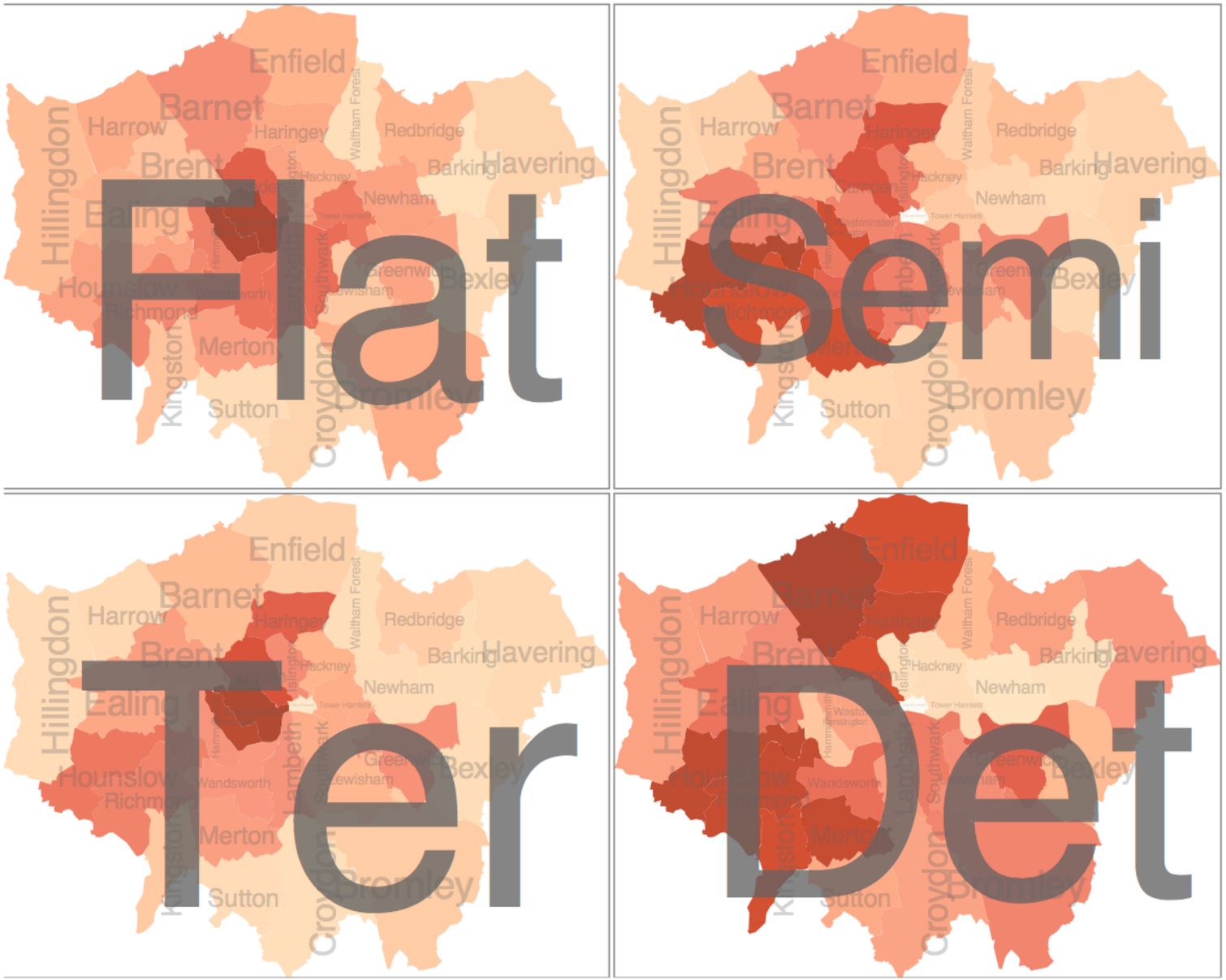
- 4 views
- 33 small multiples
- heat map inside each

[Configuring Hierarchical Layouts to Address Research Questions. Slingsby, Dykes, and Wood. IEEE Transactions on Visualization and Computer Graphics (Proc. InfoVis 2009) 15:6 (2009), 977–984.]

Partitioning: Recursive subdivision

System: **HIVE**

- different encoding for second-level regions
  - choropleth maps
  - better spatial location
  - coarser aggregation of values



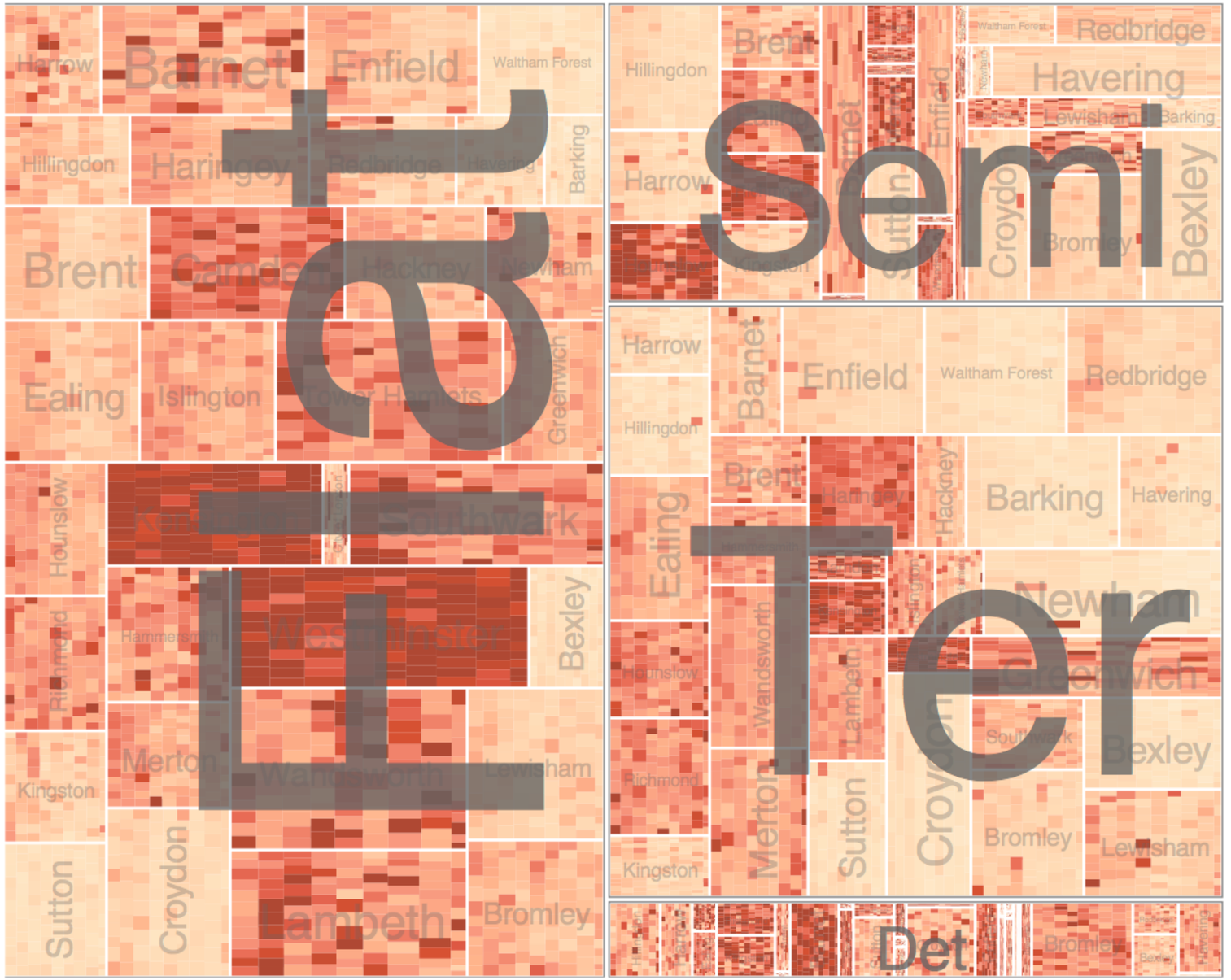
[Configuring Hierarchical Layouts to Address Research Questions. Slingsby, Dykes, and Wood. IEEE Transactions on Visualization and Computer Graphics (Proc. InfoVis 2009) 15:6 (2009), 977–984.]

Partitioning: Recursive subdivision

System: **HIVE**

- size regions by sale counts
  - –not uniformly
- result: treemap

- proportions much clearer
- locating neighborhood is difficult

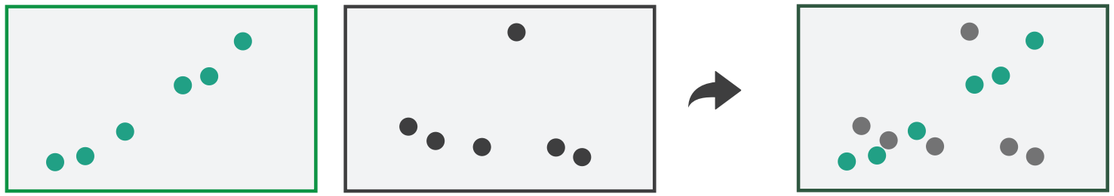


[Configuring Hierarchical Layouts to Address Research Questions. Slingsby, Dykes, and Wood. IEEE Transactions on Visualization and Computer Graphics (Proc. InfoVis 2009) 15:6 (2009), 977–984.]

Superimpose layers

- ***layer***: set of objects spread out over region
  - each set is a visually distinguishable group
  - extent: whole view
- design choices
  - how many layers?
  - how are layers distinguished?
  - small static set or dynamic from many possible?
  - how partitioned?
    - heavyweight with attributes vs lightweight with selection
- distinguishable layers
  - encode with different, non-overlapping channels
    - two layers achievable, three with careful design

Superimpose Layers



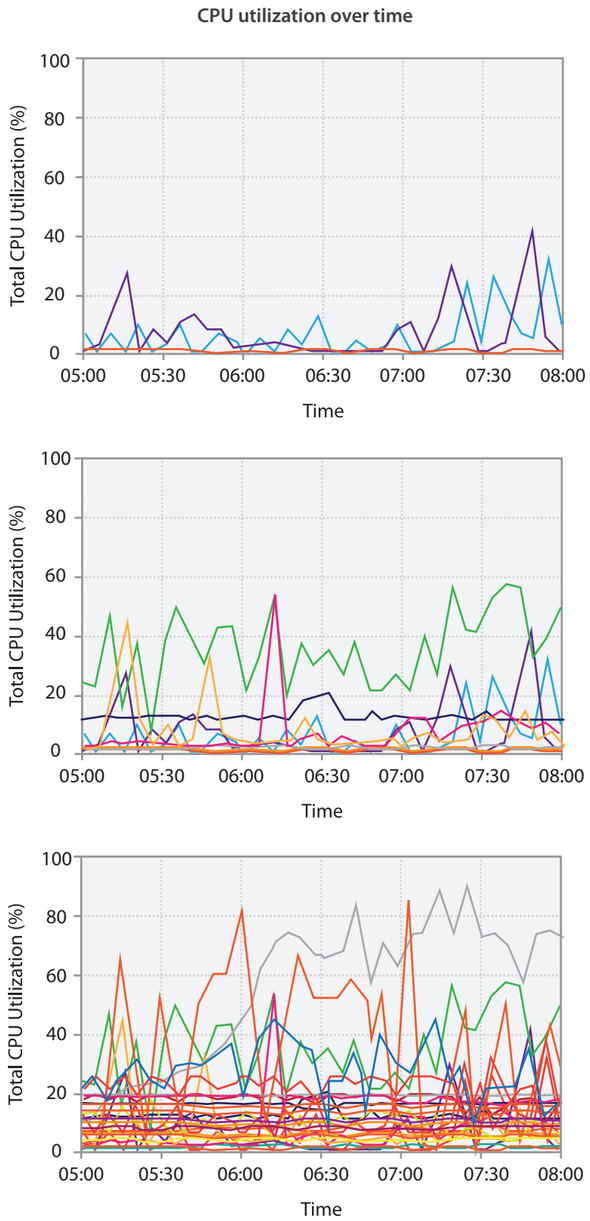
Static visual layering: cartography

- foreground layer: roads
  - hue, size distinguishing main from minor
  - high luminance contrast from background
- background layer: regions
  - desaturated colors for water, parks, land areas
- user can selectively focus attention
- "get it right in black and white"
  - check luminance contrast with greyscale view

[Get it right in black and white. Stone. 2010. http://www.stonesc.com/wordpress/2010/03/get-it-right-in-black-and-white]

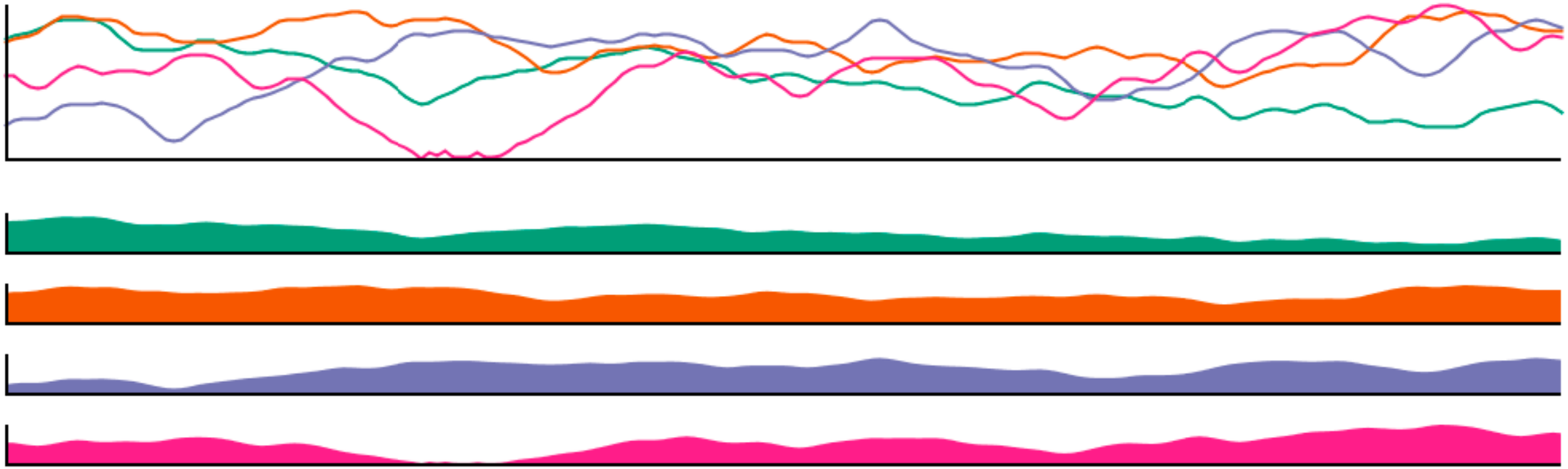
Superimposed line charts

- working well as long as the amount of overlap between different lines is small
  - many lines, but not hundreds
  - visual cluttering depends on difference between charts
- hue and/or line style to distinguish different marks
  - just several levels available



Superimpose vs juxtapose

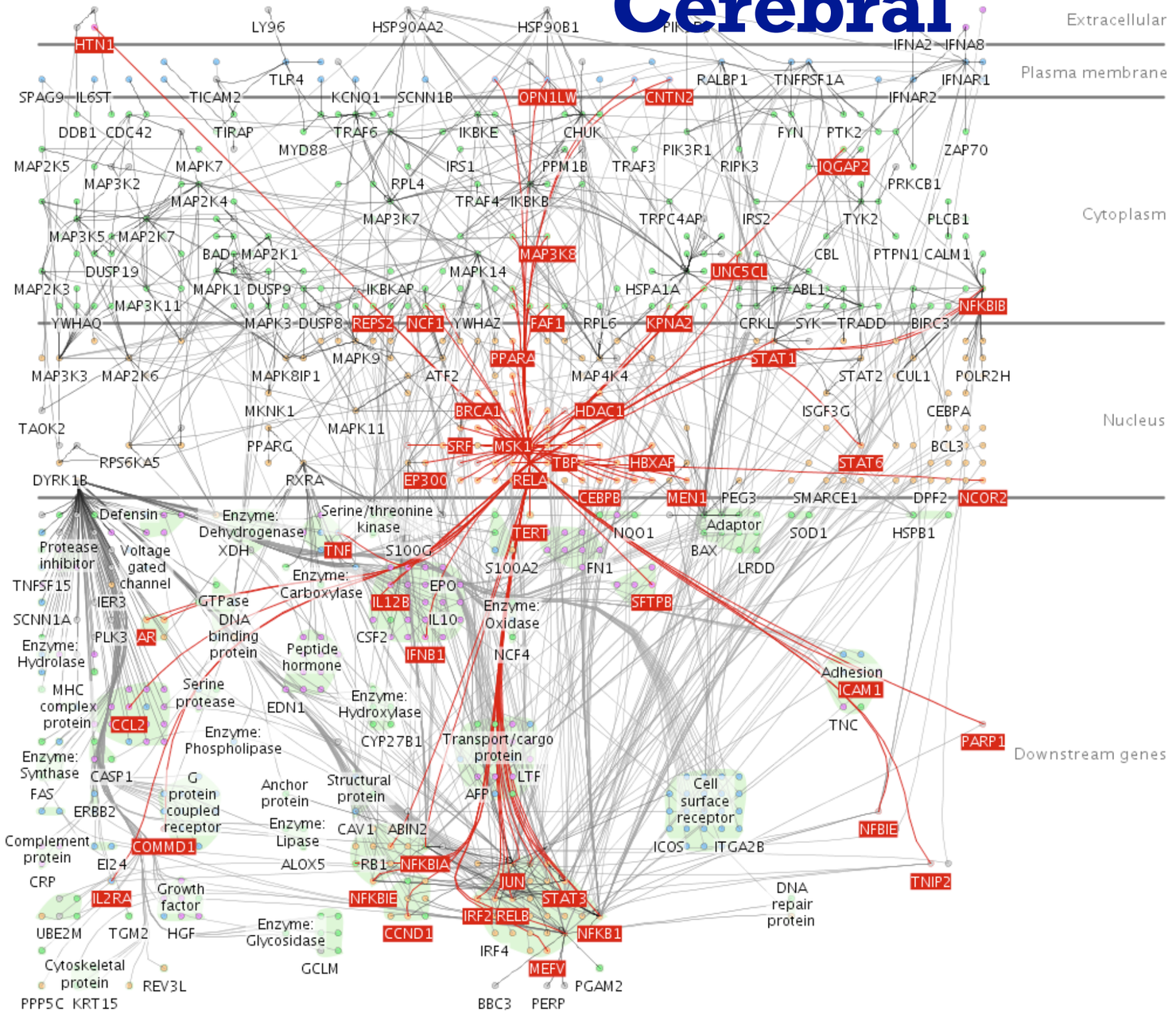
- same screen space for all multiples, single superimposed
  - superimposed: finer vertical resolution, but visual cluttering
  - juxtaposed: clearer charts, but lower vertical resolution
- tasks
  - local: maximum - superimposed is best
  - global: slope, discrimination - juxtaposed is best



[Graphical Perception of Multiple Time Series. Javed, McDonnel, and Elmqvist. IEEE Transactions on Visualization and Computer Graphics (Proc. IEEE InfoVis 2010) 16:6 (2010), 927–934.]

Dynamic visual layering

- interactive, from selection
  - –lightweight: click
  - –very lightweight: hover
- ex: 1-hop neighbors



[Cerebral: a Cytoscape plugin for layout of and interaction with biological networks using subcellular localization annotation. Barsky, Gardy, Hancock, and Munzner. Bioinformatics 23:8 (2007), 1040–1042.]