

# Visualization Analysis & Design

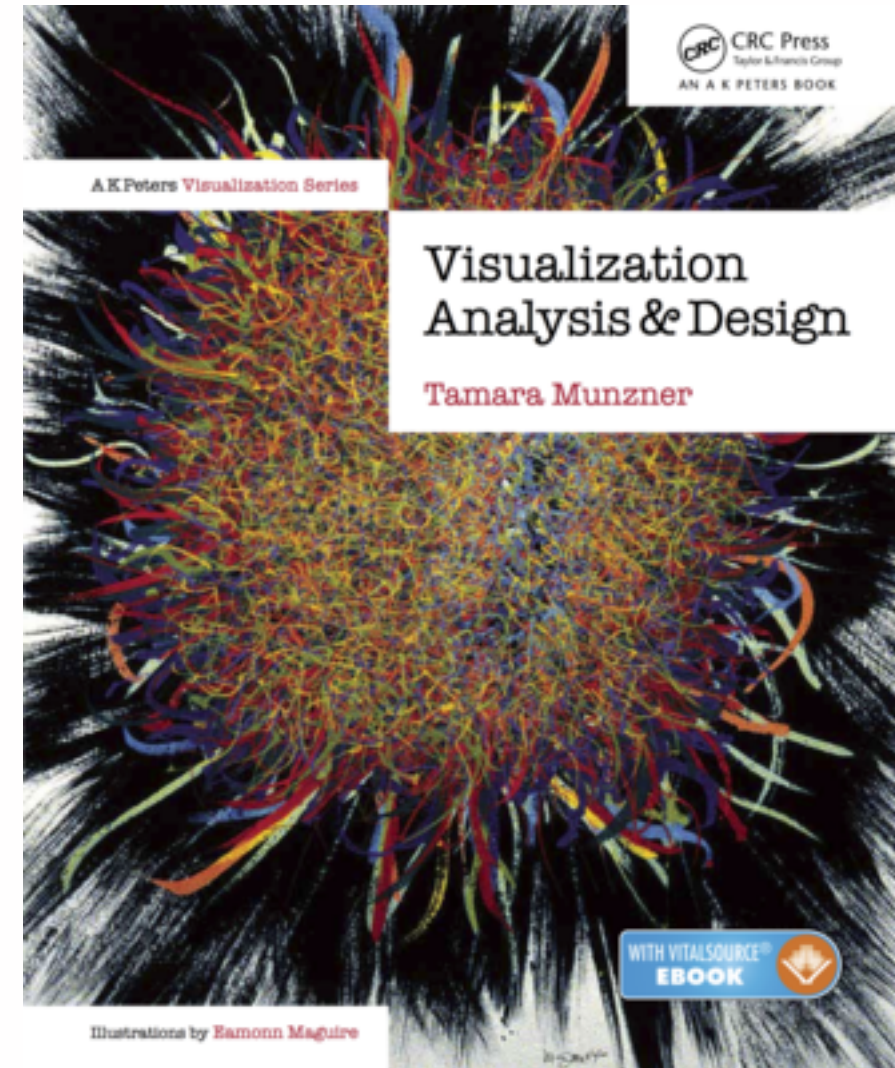
**Tamara Munzner**

Department of Computer Science  
University of British Columbia

*InformationPlus 2016 Keynote  
June 16 2016, Vancouver BC*

<http://www.cs.ubc.ca/~tmm/talks.html#vad16infoplus>

[@tamaramunzner](#)

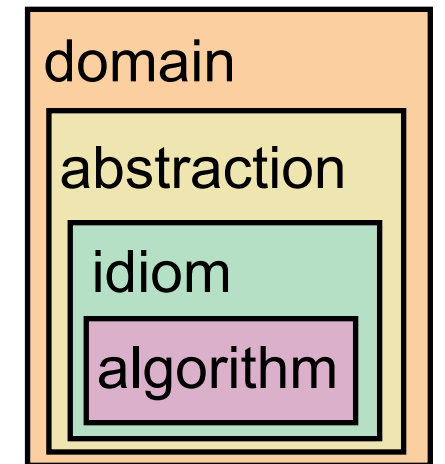


# Why talk about a textbook to a room of experts?

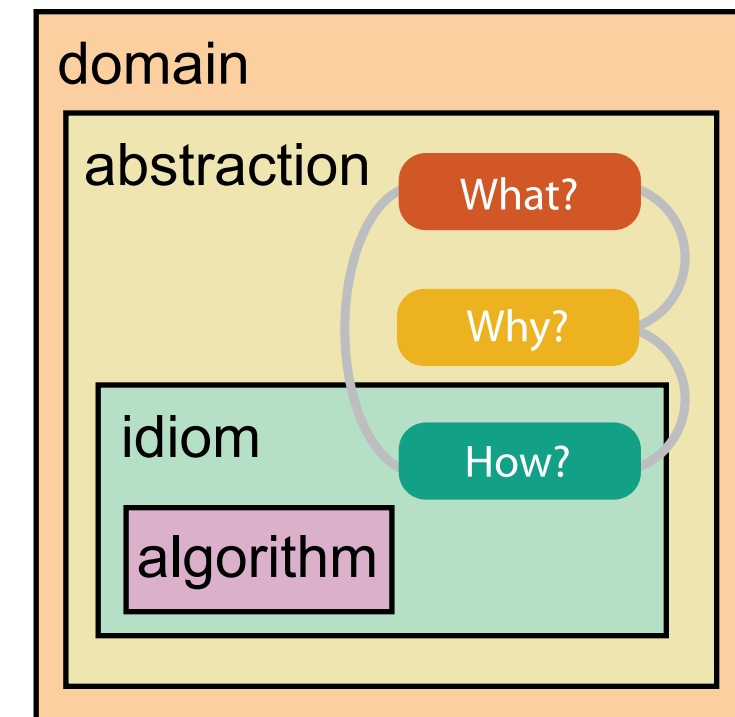
- convince you of the value in thinking systematically about vis design
  - decompose into comprehensive framework of principles and design choices
- provide unified view that crosscuts entire field of visualization
  - infovis and scivis: addressing different kinds of data
  - visual analytics: interweave data analysis & transformation w/ interactive visual exploration
  - caveat: my own background in infovis shines through!
    - my own roots in CS: graphics, later added HCI quant methods, then HCI qual methods

# Analysis framework: Four levels, three questions

- *domain* situation
  - who are the target users?
- *abstraction*
  - translate from specifics of domain to vocabulary of vis
- **what** is shown? **data abstraction**
  - often don't just draw what you're given: transform to new form
- **why** is the user looking at it? **task abstraction**
- *idiom*
  - **how** is it shown?
    - **visual encoding idiom**: how to draw
    - **interaction idiom**: how to manipulate
- *algorithm*
  - efficient computation



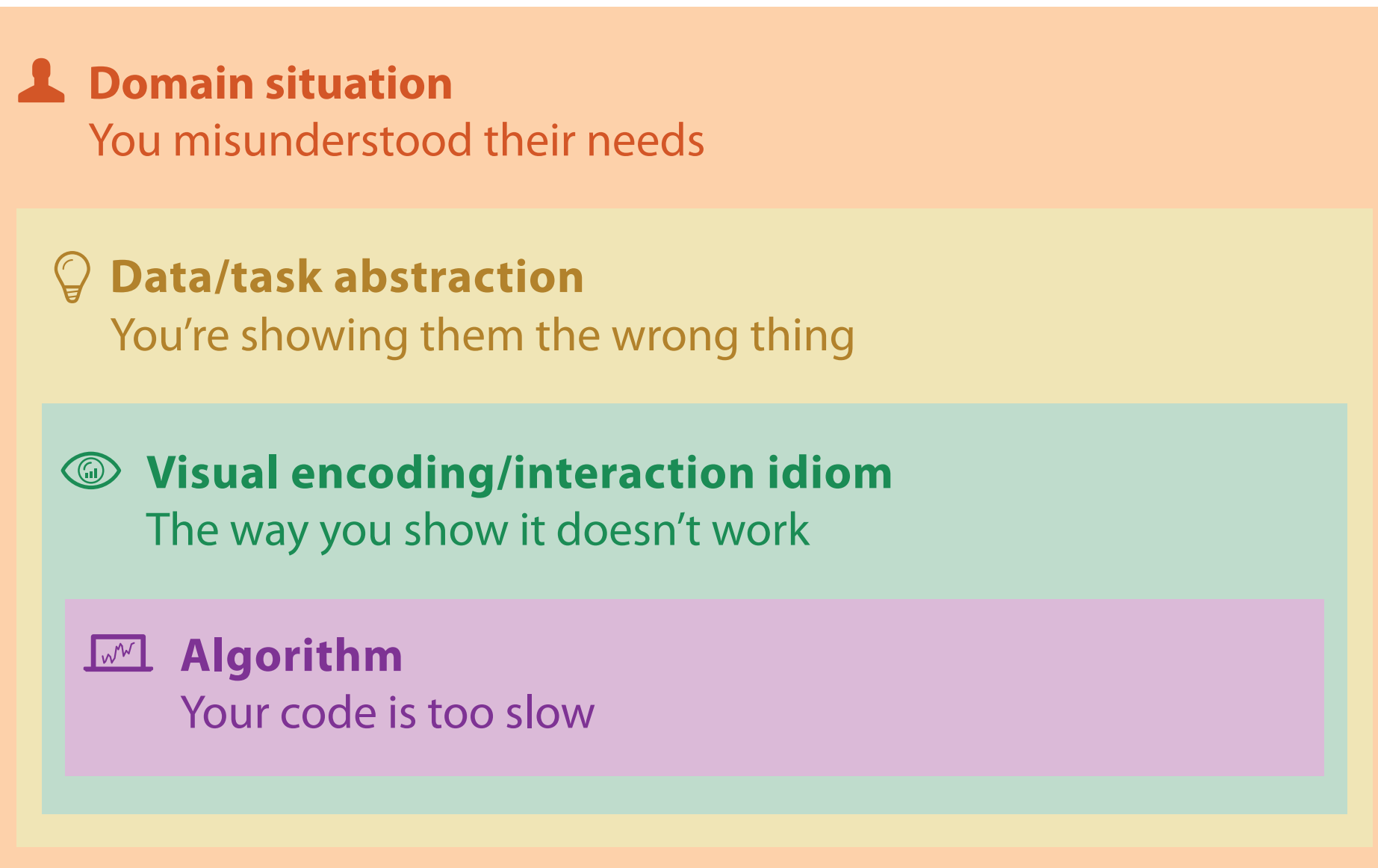
[A Nested Model of Visualization Design and Validation.  
Munzner. *IEEE TVCG* 15(6):921-928, 2009 (Proc. InfoVis 2009).]



[A Multi-Level Typology of Abstract Visualization Tasks  
Brehmer and Munzner. *IEEE TVCG* 19(12):2376-2385, 2013 (Proc. InfoVis 2013).]

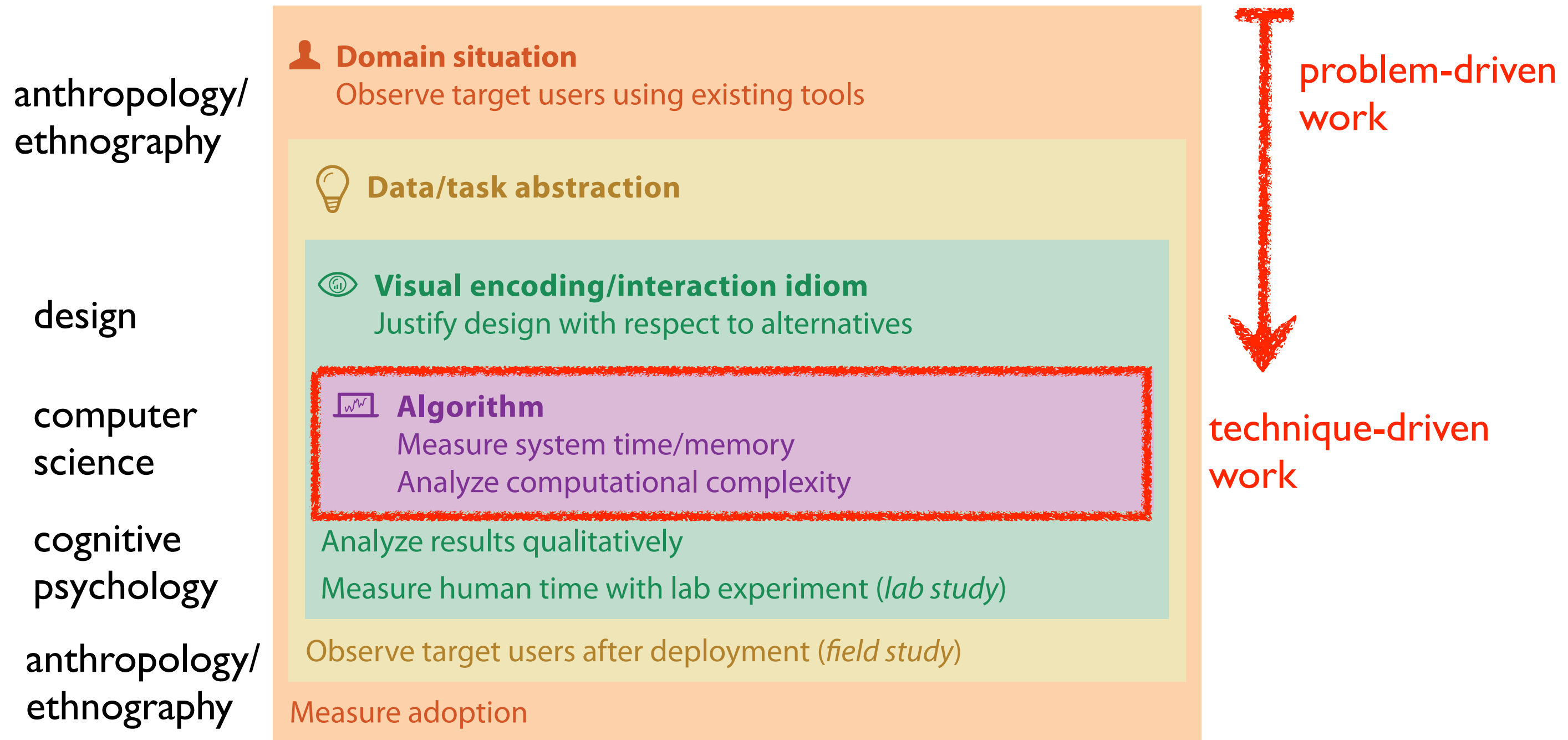
# Why is validation difficult?

- different ways to get it wrong at each level



# Why is validation difficult?

- solution: use methods from different fields at each level



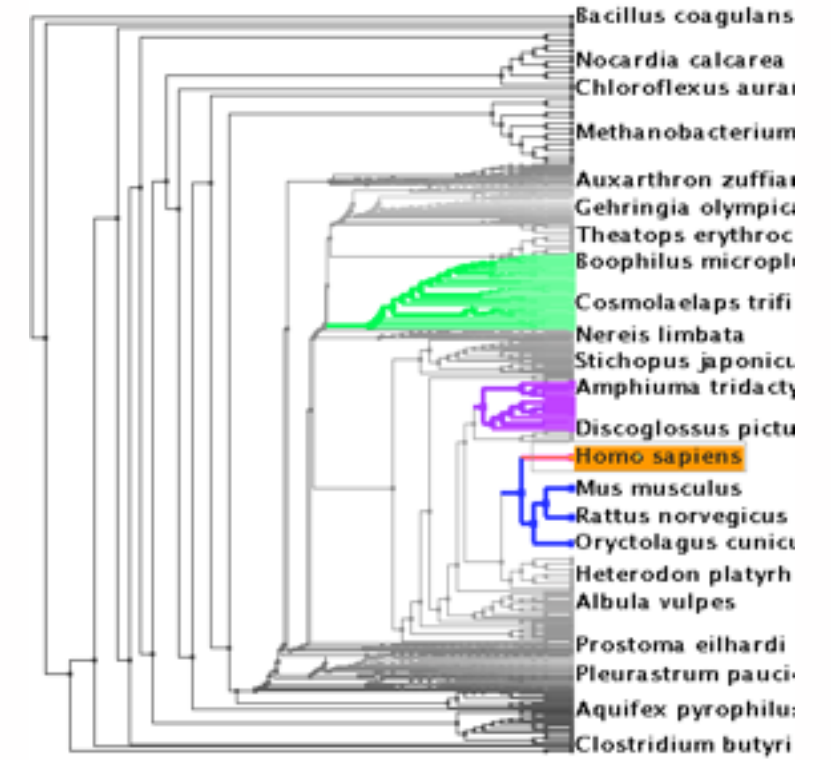
# Why analyze?

- imposes a structure on huge design space
  - scaffold to help you think systematically about choices
  - analyzing existing as stepping stone to designing new

## SpaceTree



## TreeJuxtaposer

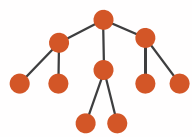


[SpaceTree: Supporting Exploration in Large Node Link Tree, Design Evolution and Empirical Evaluation. Grosjean, Plaisant, and Bederson. Proc. InfoVis 2002, p 57–64.]

[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

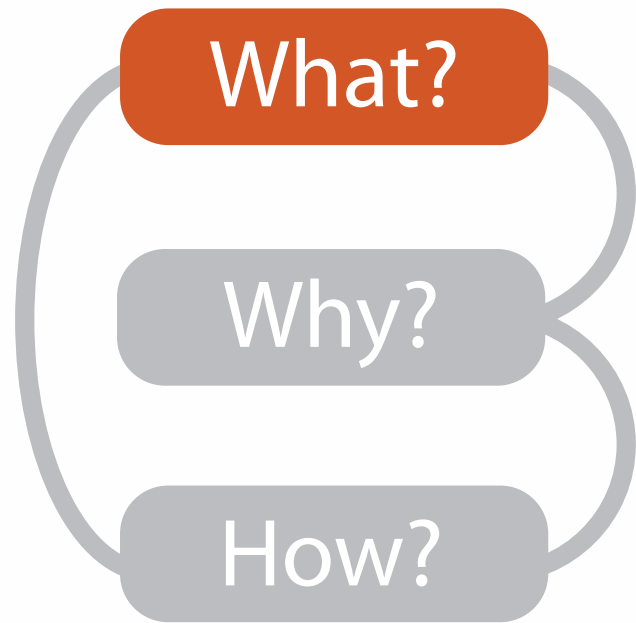
→ Encode → Navigate → Select → Arrange



What?

Why?

How?



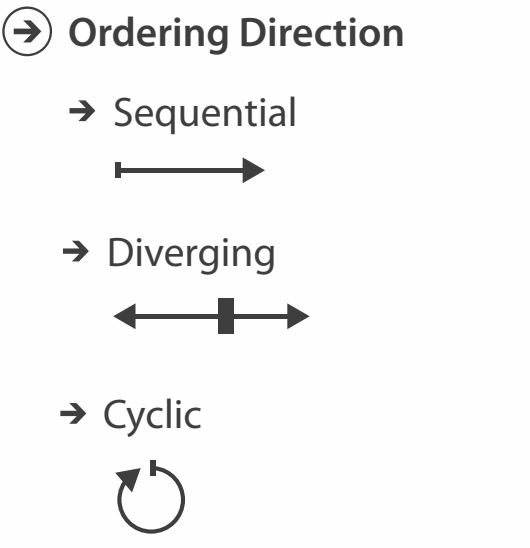
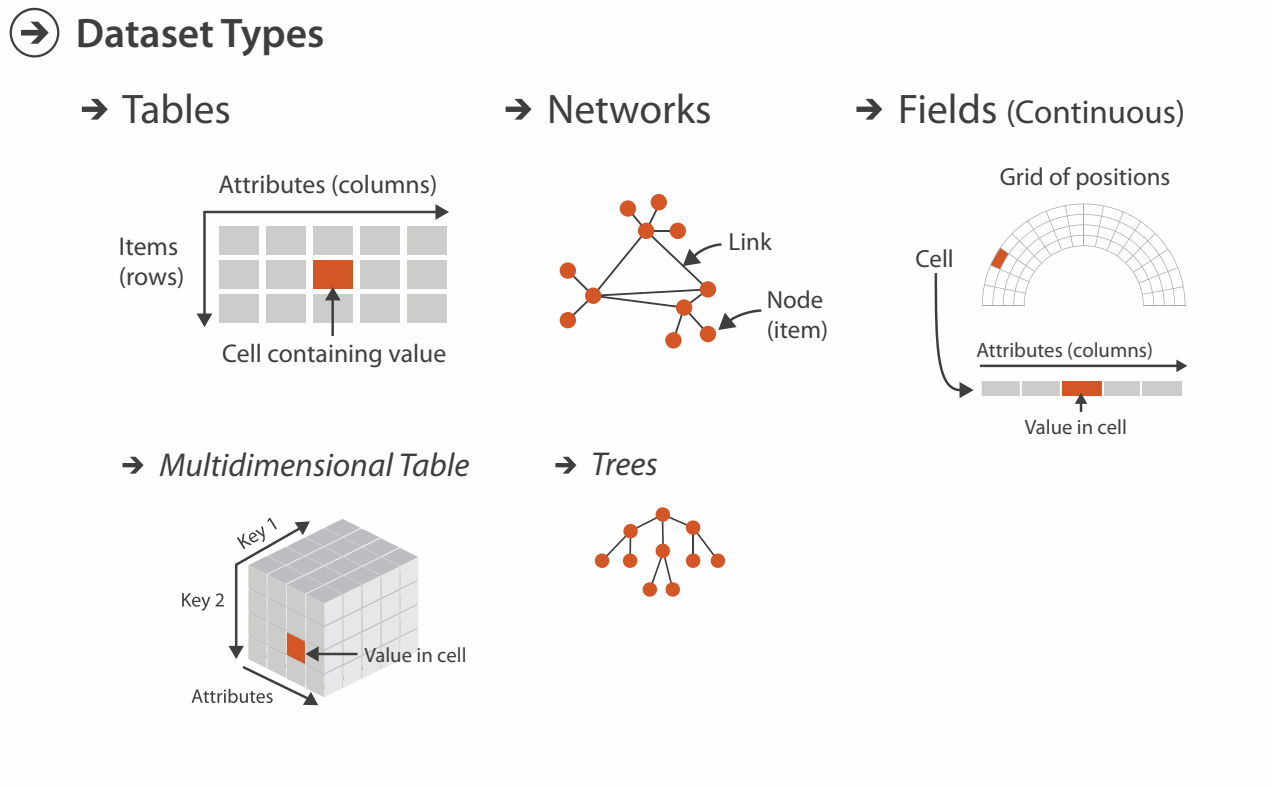
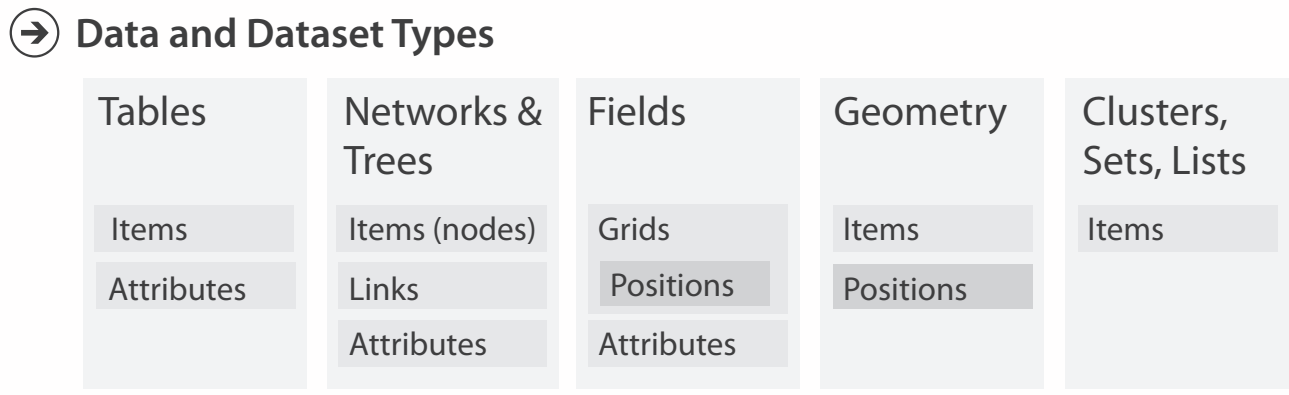
# What?

## Datasets

## Attributes

- ➔ Data Types
  - ➔ Items
  - ➔ Attributes
  - ➔ Links
  - ➔ Positions
  - ➔ Grids

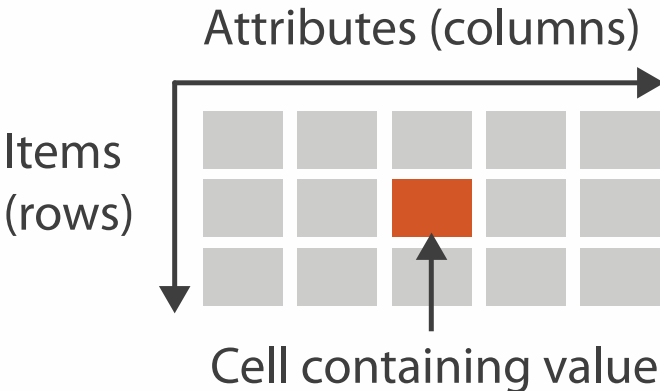
- ➔ Attribute Types
  - ➔ Categorical
    - + ● ■ ▲
  - ➔ Ordered
    - ➔ Ordinal
      - 👕 👕 👕
    - ➔ Quantitative
      - ┆ ┆ ┆



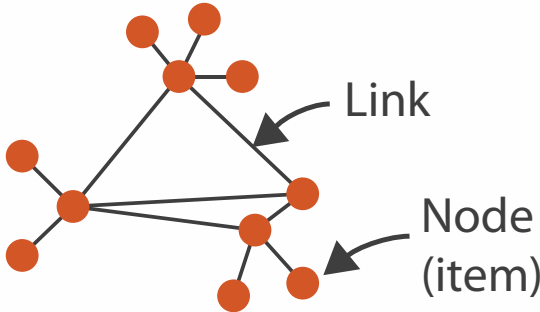
# Types: Datasets and data

## → Dataset Types

→ Tables

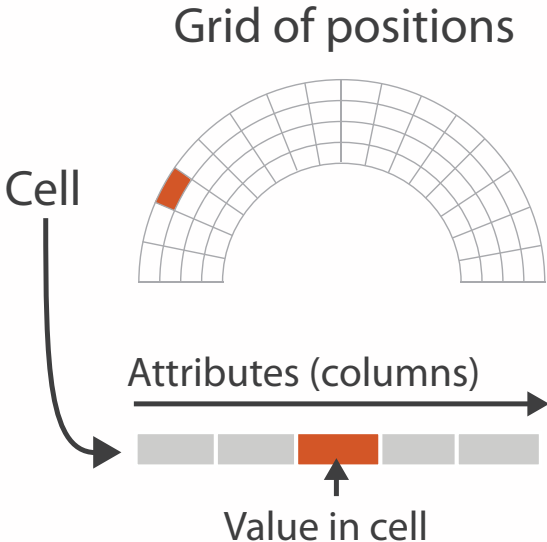


→ Networks

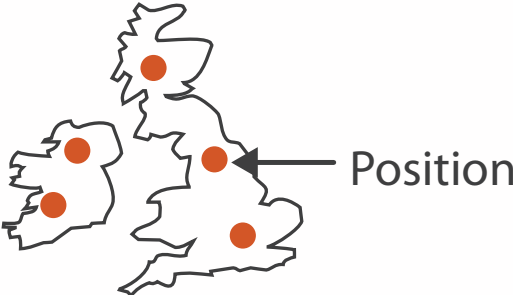


→ Spatial

→ Fields (Continuous)



→ Geometry (Spatial)



## → Attribute Types

→ Categorical

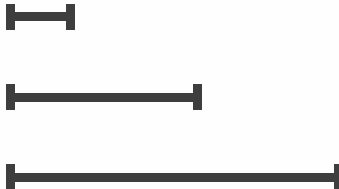


→ Ordered

→ Ordinal



→ Quantitative











# Why?




## 👉 Actions

## 🎯 Targets




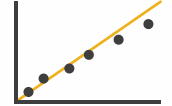
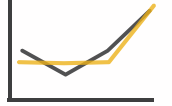
➔ **Analyze**

- ➔ Consume
  - ➔ Discover 
  - ➔ Present 
  - ➔ Enjoy 
- ➔ Produce
  - ➔ Annotate 
  - ➔ Record 
  - ➔ Derive 





➔ **All Data**

- ➔ Trends 
- ➔ Outliers 
- ➔ Features 

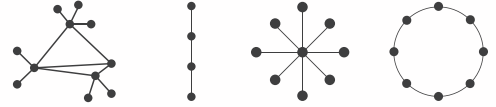

➔ **Attributes**

- ➔ One
  - ➔ Distribution 
  - ➔ Extremes 
- ➔ Many
  - ➔ Dependency 
  - ➔ Correlation 
  - ➔ Similarity 


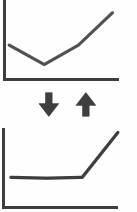

➔ **Search**

	Target known	Target unknown
Location known	 <i>Lookup</i>	 <i>Browse</i>
Location unknown	 <i>Locate</i>	 <i>Explore</i>


➔ **Network Data**

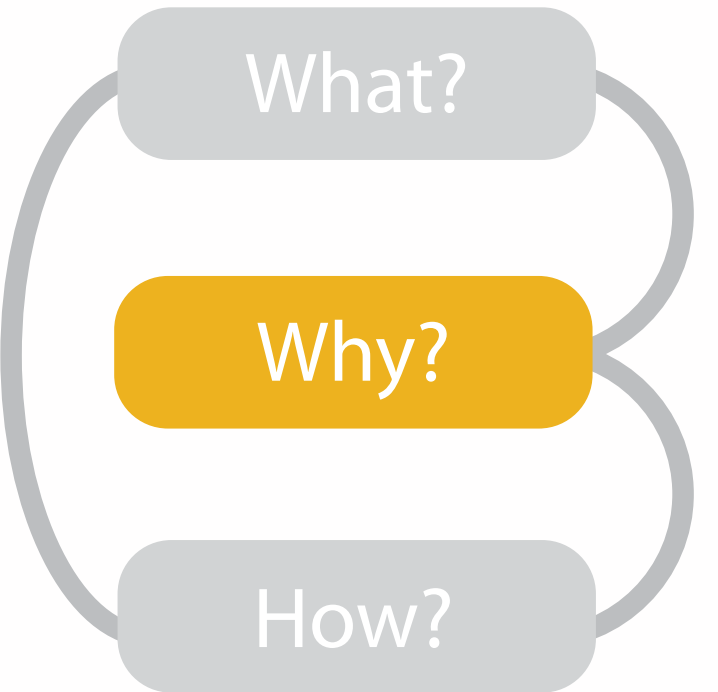
- ➔ Topology
  - 
  - ➔ Paths 

➔ **Query**

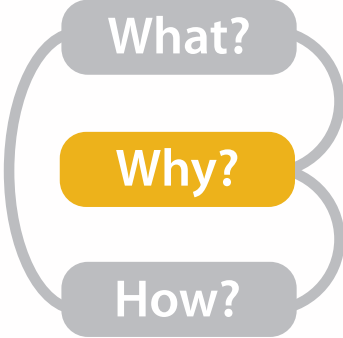
- ➔ Identify 
- ➔ Compare 
- ➔ Summarize 

➔ **Spatial Data**

- ➔ Shape 



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



# Actions: Analyze, Query

- analyze

- consume

- discover vs present

- aka explore vs explain

- enjoy

- aka casual, social

- produce

- annotate, record, derive

- query

- how much data matters?

- one, some, all

- independent choices

- analyze, query, (search)

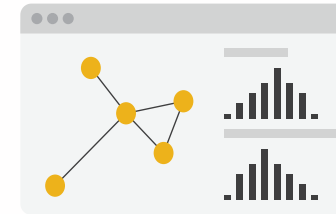
## → Analyze

- Consume

- Discover



- Present

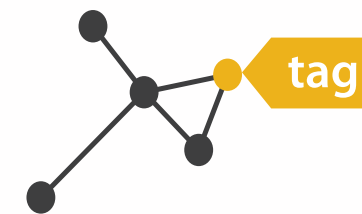


- Enjoy

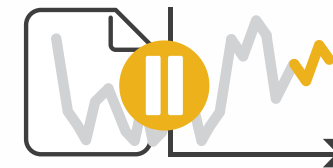


- Produce

- Annotate



- Record

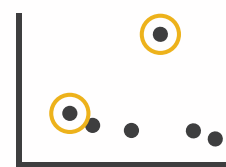


- Derive

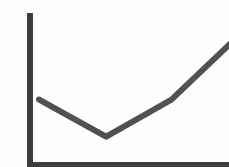


## → Query

- Identify



- Compare

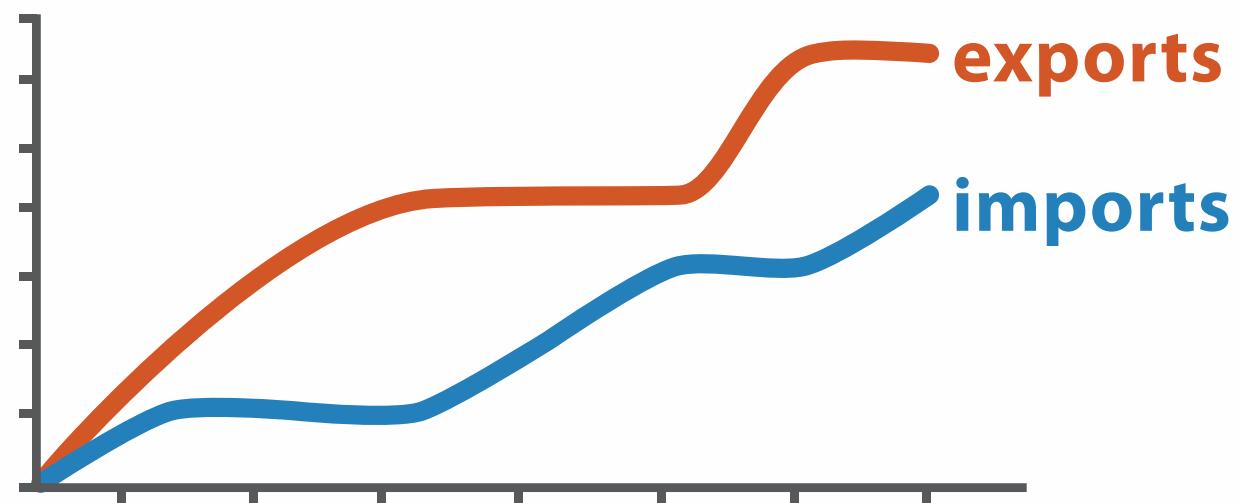


- Summarize

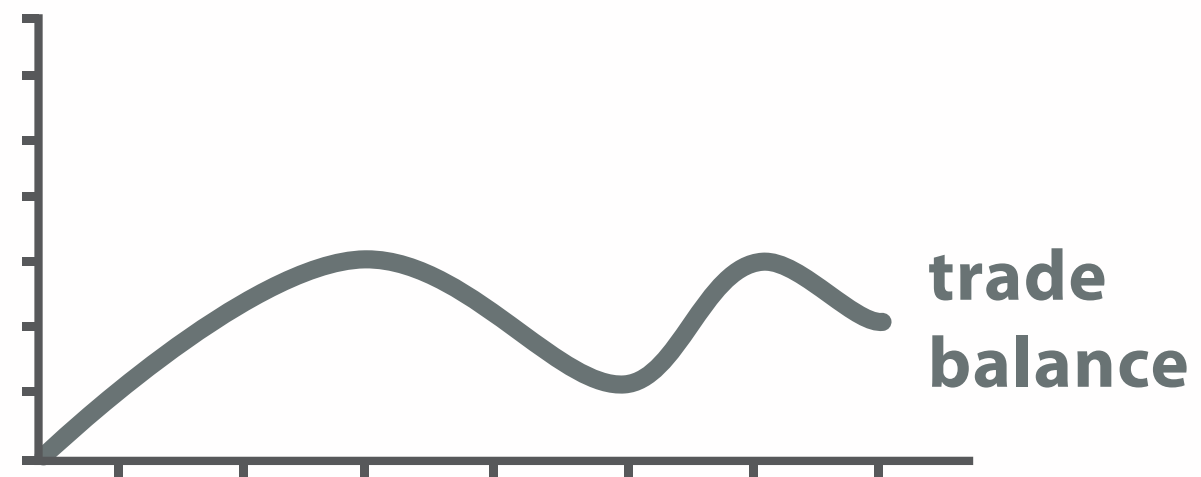


# Derive: Crucial Design Choice

- 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



$$\text{trade balance} = \text{exports} - \text{imports}$$

Derived Data

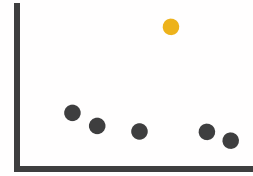
# Targets

## → All Data

→ Trends



→ Outliers



→ Features



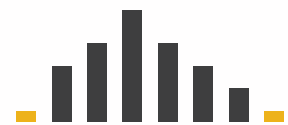
## → Attributes

→ One

→ *Distribution*

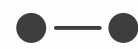


→ *Extremes*

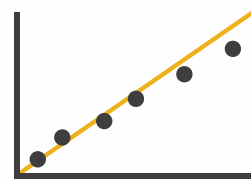


→ Many

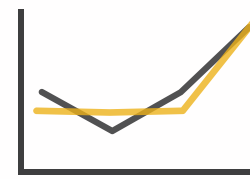
→ *Dependency*



→ *Correlation*

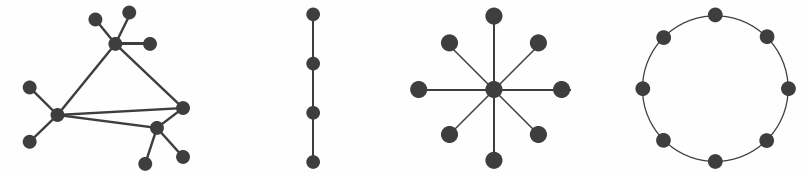


→ *Similarity*

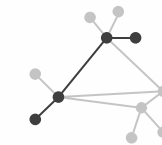


## → Network Data

→ Topology

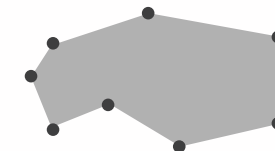


→ *Paths*



## → Spatial Data

→ Shape



# How?

## Encode

### → Arrange

→ Express



→ Separate



→ Order



→ Align



→ Use



### → Map

from **categorical** and **ordered** attributes

→ Color

→ Hue



→ Saturation



→ Luminance



→ Size, Angle, Curvature, ...



→ Shape



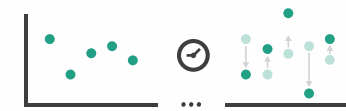
→ Motion

*Direction, Rate, Frequency, ...*

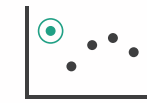


## Manipulate

### → Change



### → Select

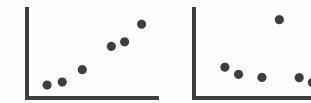


### → Navigate

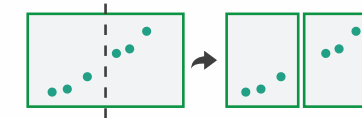


## Facet

### → Juxtapose



### → Partition



### → Superimpose



## Reduce

### → Filter



### → Aggregate



### → Embed



What?

Why?

How?

# How to encode: Arrange space, map channels

## Encode

---

### ② Arrange

→ Express



→ Order



→ Use



→ Separate



→ Align



### ② Map

from **categorical** and **ordered** attributes

→ Color

→ Hue



→ Saturation



→ Luminance



→ Size, Angle, Curvature, ...



→ Shape



→ Motion

*Direction, Rate, Frequency, ...*



# Definitions: Marks and channels

- marks

  - geometric primitives

→ Points



→ Lines



→ Areas



- channels

  - control appearance of marks

→ Position

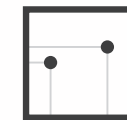
→ Horizontal



→ Vertical



→ Both



→ Color



→ Shape



→ Tilt



→ Size

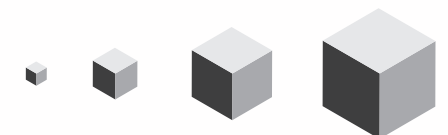
→ Length



→ Area

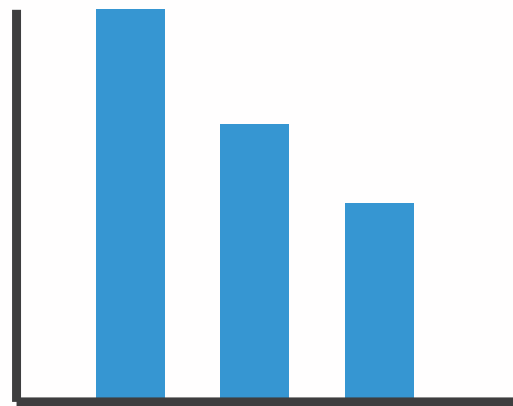


→ Volume



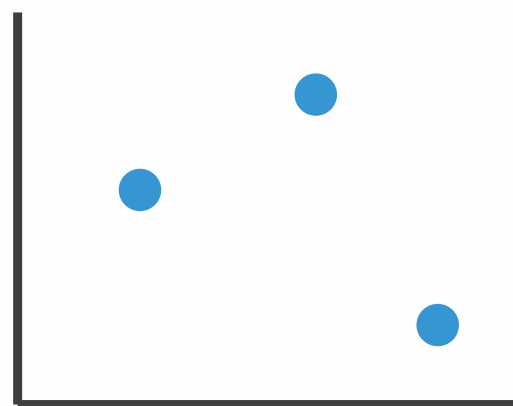
# Encoding visually with marks and channels

- analyze idiom structure
  - as combination of marks and channels



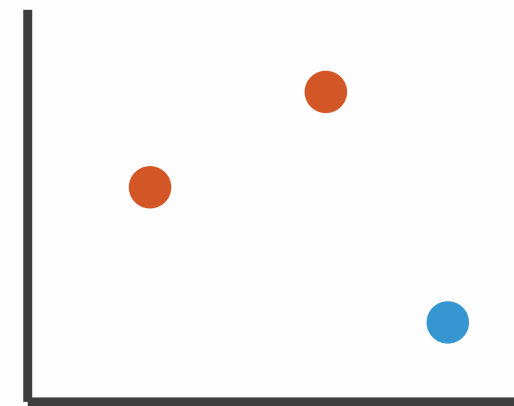
1:  
vertical position

mark: line



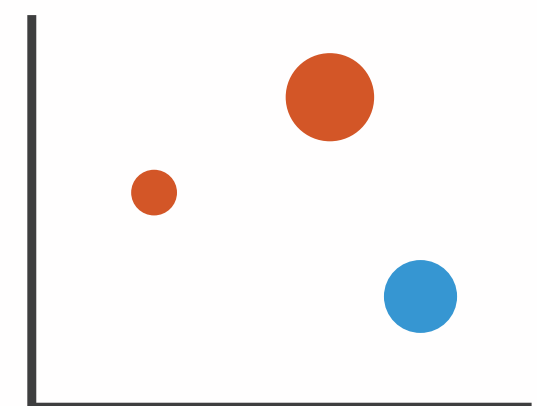
2:  
vertical position  
horizontal position

mark: point



3:  
vertical position  
horizontal position  
color hue

mark: point



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

mark: point



# Channels

Position on common scale



Position on unaligned scale



Length (1D size)



Tilt/angle



Area (2D size)



Depth (3D position)



Color luminance



Color saturation



Curvature



Volume (3D size)



Same

Spatial region



Color hue



Motion



Shape



# Channels: Matching Types

## ➔ Magnitude Channels: Ordered Attributes

Position on common scale 

Position on unaligned scale 

Length (1D size) 

Tilt/angle 

Area (2D size) 

Depth (3D position) 

Color luminance 

Color saturation 

Curvature 

Volume (3D size) 

Same

Same

## ➔ Identity Channels: Categorical Attributes

Spatial region 

Color hue 

Motion 

Shape 

- **expressiveness principle**
  - match channel and data characteristics

# Channels: Rankings

## ➔ Magnitude Channels: Ordered Attributes

Position on common scale 

Position on unaligned scale 

Length (1D size) 

Tilt/angle 

Area (2D size) 

Depth (3D position) 

Color luminance 

Color saturation 

Curvature 

Volume (3D size) 

Same

Best

Effectiveness

Least

## ➔ Identity Channels: Categorical Attributes

Spatial region 

Color hue 

Motion 

Shape 

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

# How?

## Encode

### → Arrange

→ Express



→ Separate



→ Order



→ Align



→ Use



### → Map

from **categorical** and **ordered** attributes

→ Color

→ Hue



→ Saturation



→ Luminance



→ Size, Angle, Curvature, ...



→ Shape



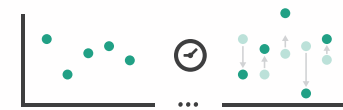
→ Motion

*Direction, Rate, Frequency, ...*

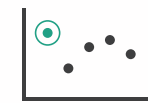


## Manipulate

### → Change



### → Select



### → Navigate

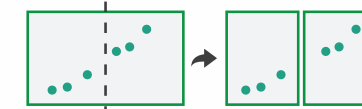


## Facet

### → Juxtapose



### → Partition



### → Superimpose



## Reduce

### → Filter



### → Aggregate



### → Embed



What?

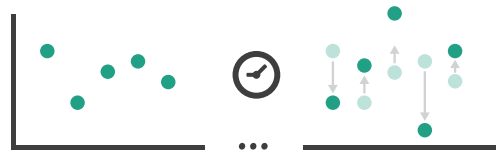
Why?

How?

# How to handle complexity: 3 more strategies + 1 previous

## Manipulate

### ➔ Change



### ➔ Select

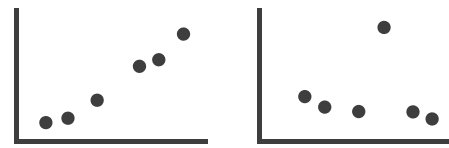


### ➔ Navigate

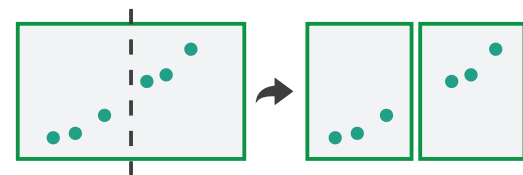


## Facet

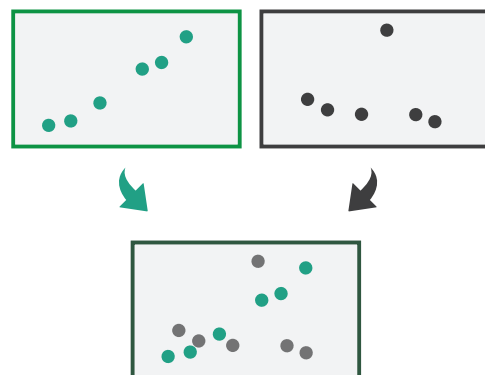
### ➔ Juxtapose



### ➔ Partition



### ➔ Superimpose



## Reduce

### ➔ Filter



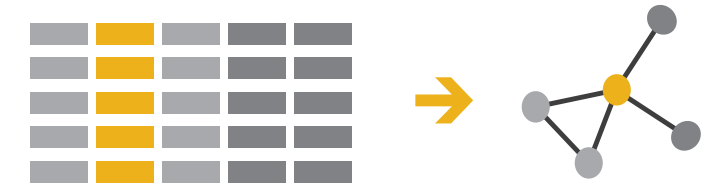
### ➔ Aggregate



### ➔ Embed



➔ *Derive*




- change view over time
- facet across multiple views
- reduce items/attributes within single view
- derive new data to show within view

# How to handle complexity: 3 more strategies

+ 1 previous

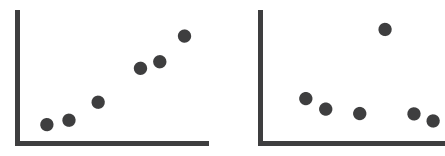
**Manipulate**

→ Change



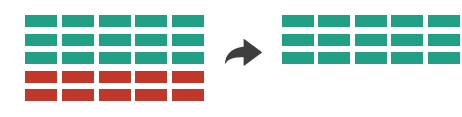
**Facet**

→ Juxtapose

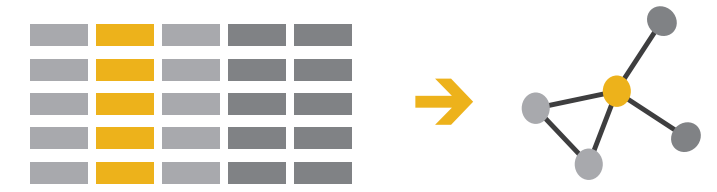


**Reduce**

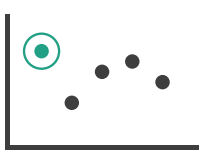
→ Filter



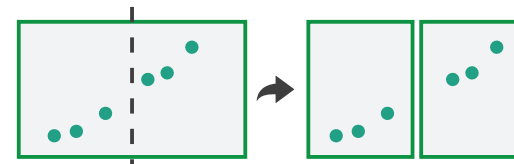
→ *Derive*



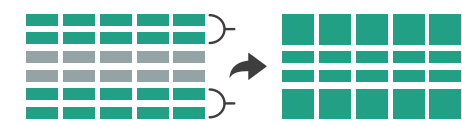
→ Select



→ Partition



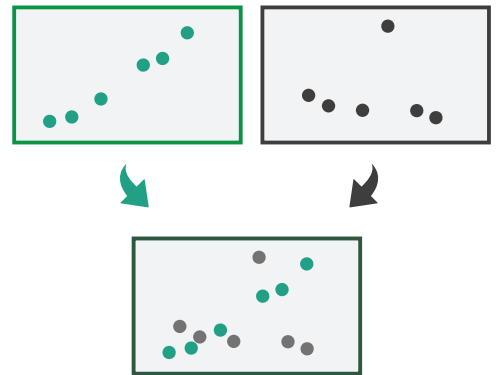
→ Aggregate




→ Navigate



→ Superimpose



→ Embed



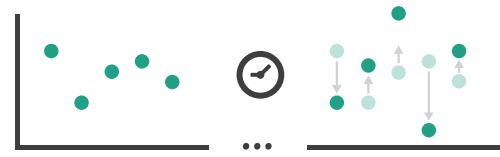
- change over time
- most obvious & flexible of the 4 strategies

# How to handle complexity: 3 more strategies

+ 1 previous

## Manipulate

➔ Change



➔ Select

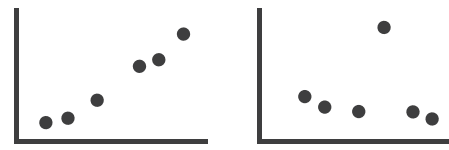


➔ Navigate

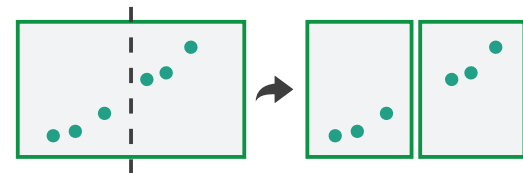


## Facet

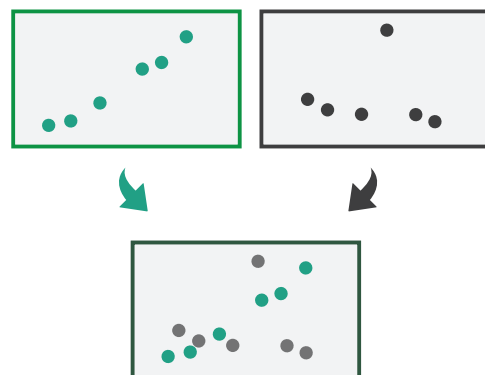
➔ Juxtapose



➔ Partition



➔ Superimpose

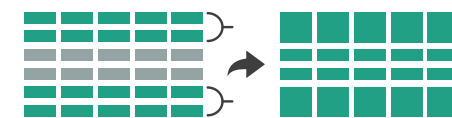


## Reduce

➔ Filter



➔ Aggregate



➔ Embed



➔ *Derive*

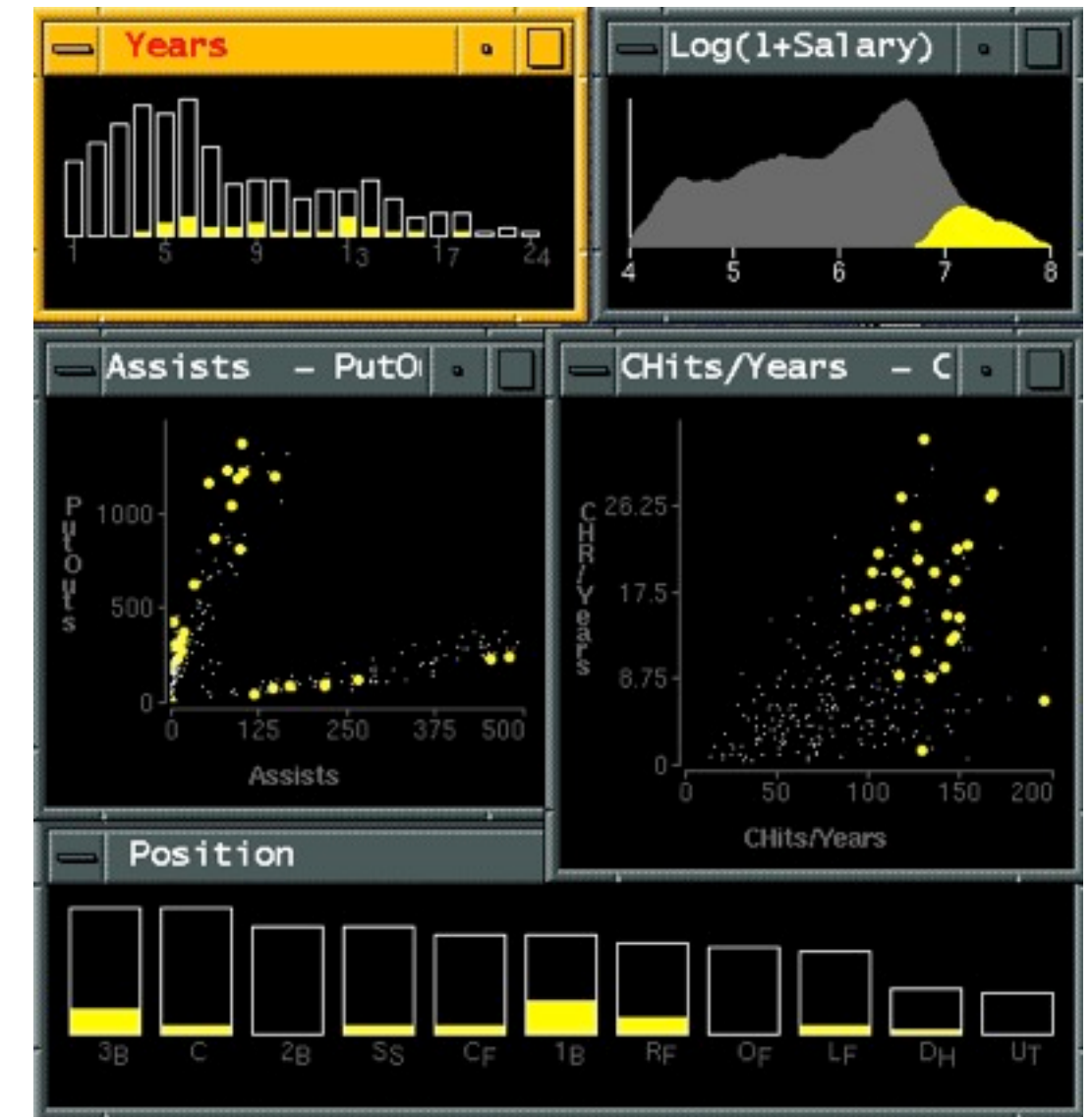


- facet data across multiple views

# Idiom: **Linked highlighting**

System: **EDV**

- see how regions contiguous in one view are distributed within another
  - powerful and pervasive interaction idiom
- encoding: different
- 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: **bird's-eye maps**

# System: **Google Maps**

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

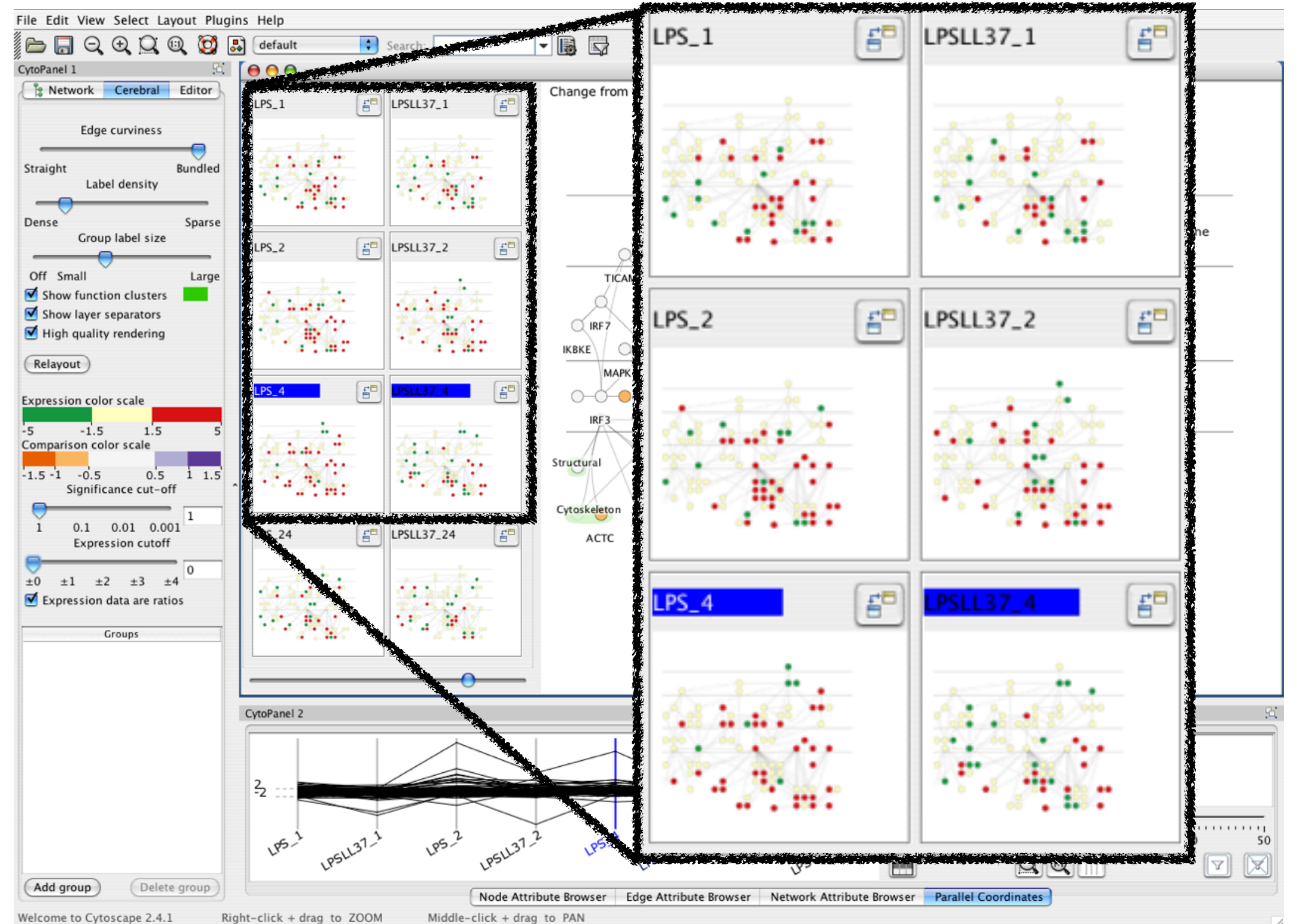


[A Review of Overview+Detail, Zooming, and Focus+Context Interfaces. Cockburn, Karlson, and Bederson. *ACM Computing Surveys* 41:1 (2008), 1–31.]

# Idiom: **Small multiples**

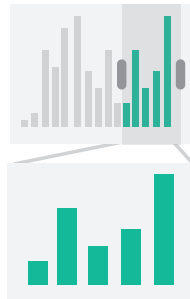
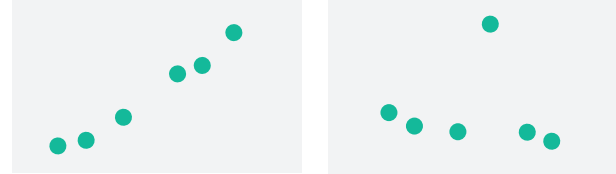


# System: **Cerebral**

- 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.]

# Coordinate views: Design choice interaction

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

- why juxtapose views?

- benefits: eyes vs memory

- lower cognitive load to move eyes between 2 views than remembering previous state with single changing view

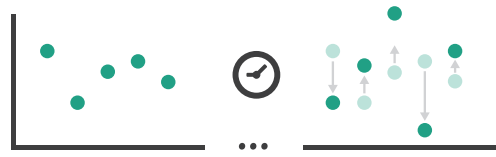
- costs: display area, 2 views side by side each have only half the area of one view

# How to handle complexity: 3 more strategies

+ 1 previous

## Manipulate

➔ Change



➔ Select

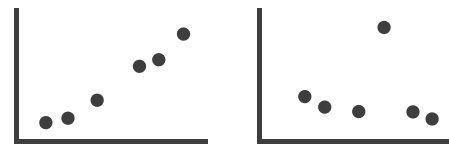


➔ Navigate

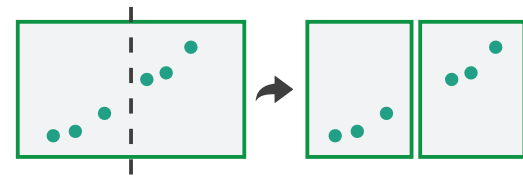


## Facet

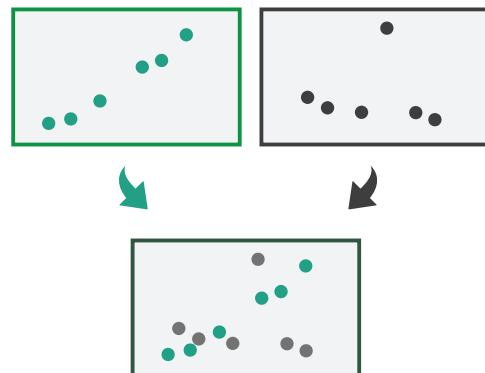
➔ Juxtapose



➔ Partition



➔ Superimpose

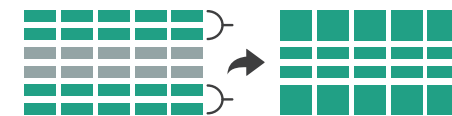


## Reduce

➔ Filter



➔ Aggregate



➔ Embed



➔ *Derive*



- reduce what is shown within single view

# Reduce items and attributes

- reduce/increase: inverses
- filter
  - pro: straightforward and intuitive
    - to understand and compute
  - con: out of sight, out of mind
- aggregation
  - pro: inform about whole set
  - con: difficult to avoid losing signal
- not mutually exclusive
  - combine filter, aggregate
  - combine reduce, facet, change, derive

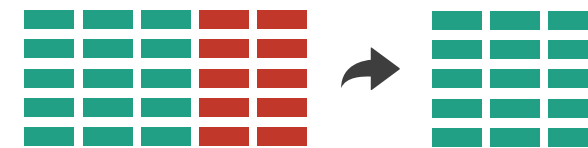
## Reducing Items and Attributes

### → Filter

→ Items

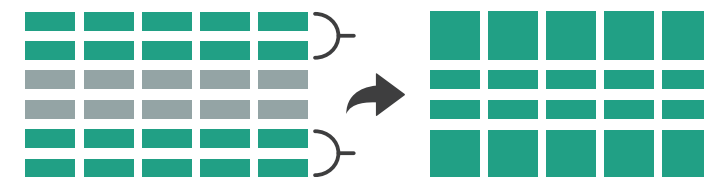


→ Attributes

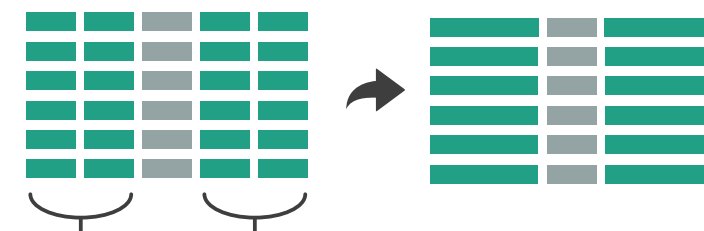


### → Aggregate

→ Items



→ Attributes

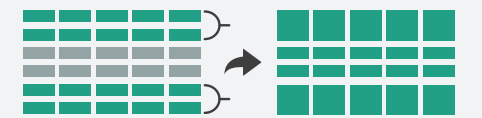


## Reduce

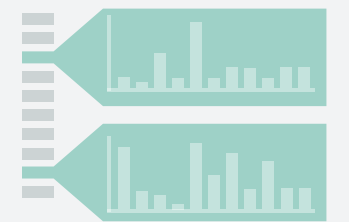
### → Filter



### → Aggregate

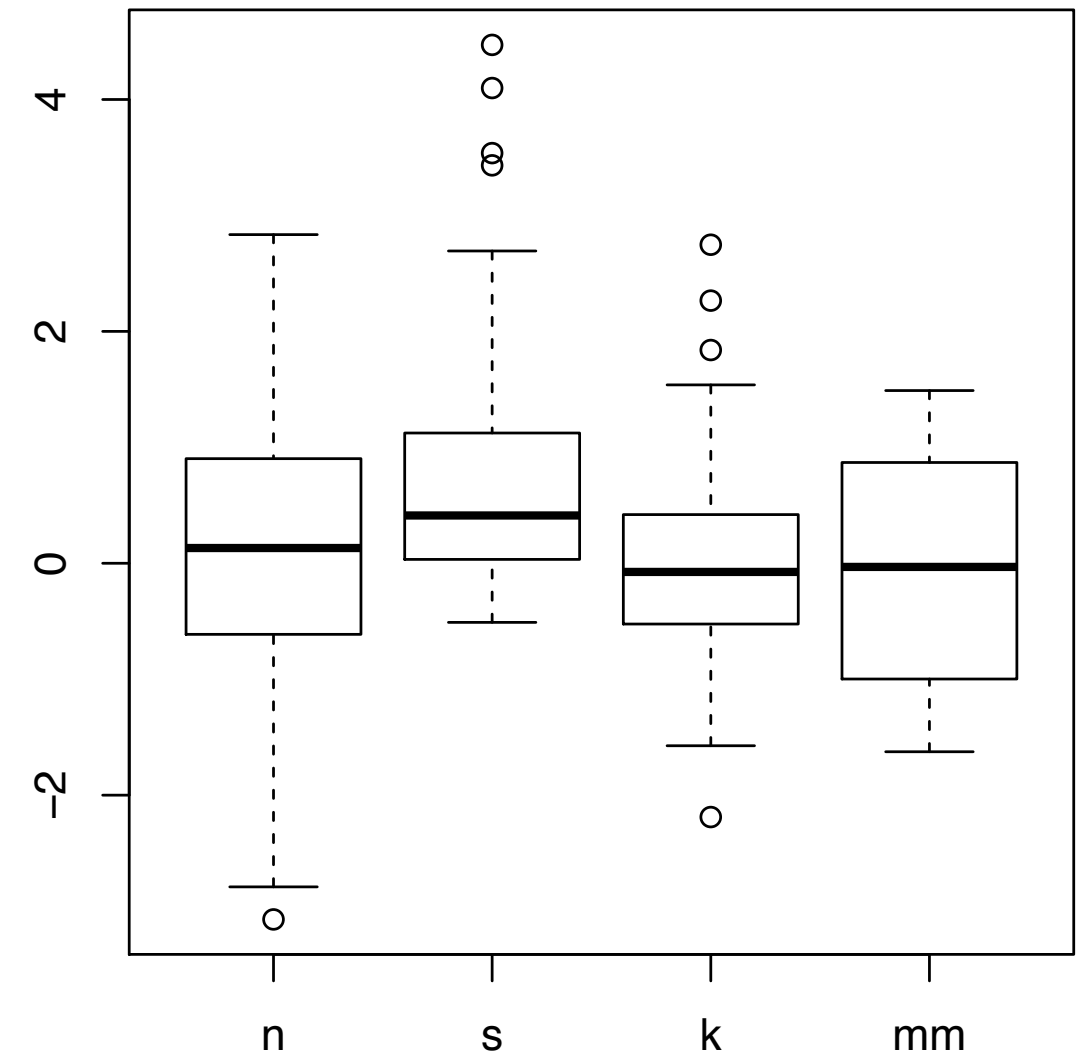


### → Embed



# Idiom: **boxplot**

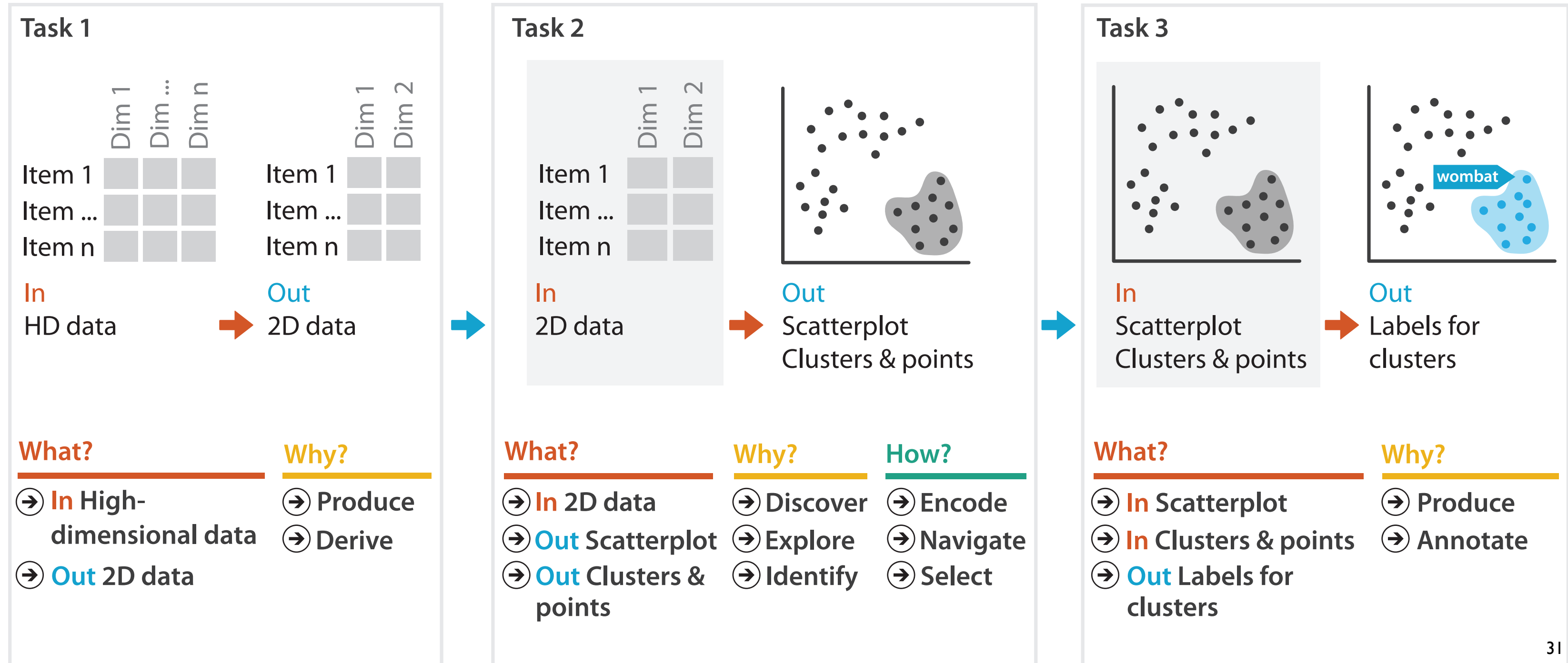
- static item aggregation
- task: find distribution
- data: table
- derived data
  - 5 quant attribs
    - median: central line
    - lower and upper quartile: boxes
    - lower upper fences: whiskers
      - values beyond which items are outliers
  - outliers beyond fence cutoffs explicitly shown



*[40 years of boxplots. Wickham and Stryjewski. 2012. had.co.nz]*

# Idiom: Dimensionality reduction for documents

- attribute aggregation
  - derive low-dimensional target space from high-dimensional measured space



# What?

Datasets

Attributes

domain

abstraction

What?

Why?

idiom

How?

algorithm

## Why?

Actions

Targets

→ Data Types

→ Items

→ Data and D

Tables

Items

Attributes

→ Analyze

→ Consume

→ Discover



→ Present



→ Enjoy



→ All Data

→ Trends



→ Outliers

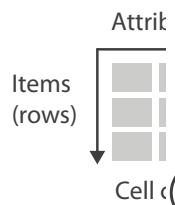


→ Features



→ Dataset Typ

→ Tables



→ Produce

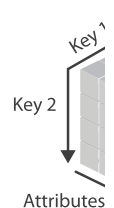
→ Annotate



→ Search

	Tar
Location known	••
Location unknown	<••

→ Multidir



→ Geometr



→ Query

→ Identify



## How?

Encode

Manipulate

Facet

Reduce

→ Arrange

→ Express



→ Separate



→ Order



→ Align



→ Use



→ Map

from **categorical** and **ordered** attributes

→ Color

→ Hue



→ Saturation



→ Luminance



→ Size, Angle, Curvature, ...



→ Shape

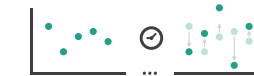


→ Motion

Direction, Rate, Frequency, ...



→ Change



→ Select



→ Navigate



→ Juxtapose



→ Partition



→ Superimpose



→ Filter



→ Aggregate



→ Embed

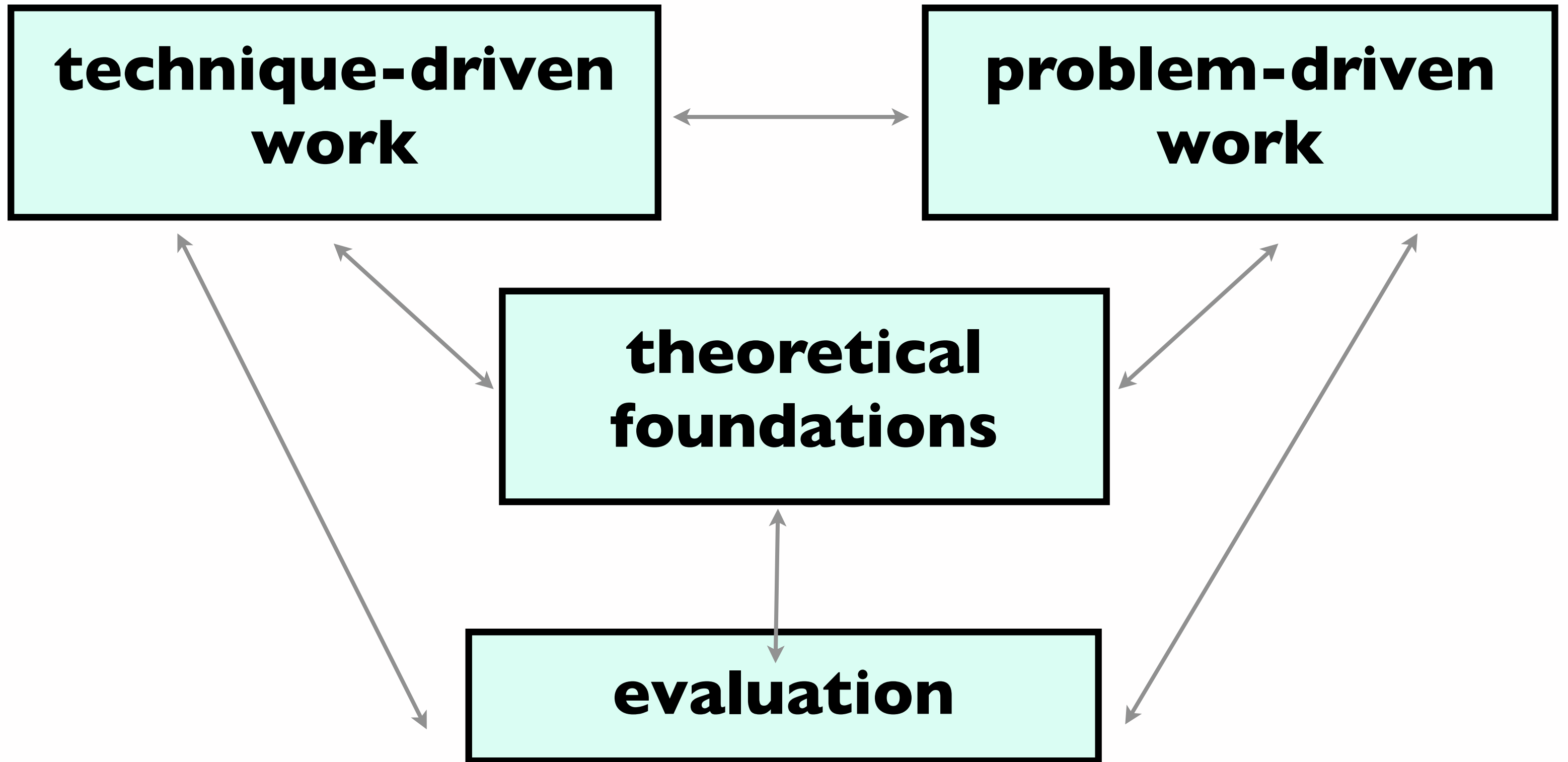


What?

Why?



A quick taste of my own work!



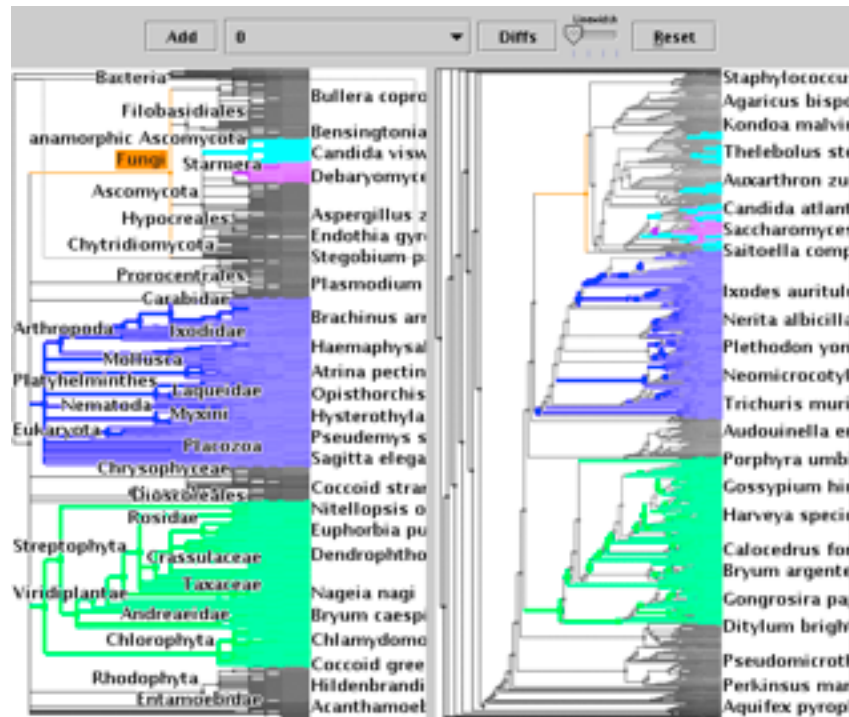
# Technique-driven: Graph drawing

T

P

F

E



James Slack



Kristian Hildebrand

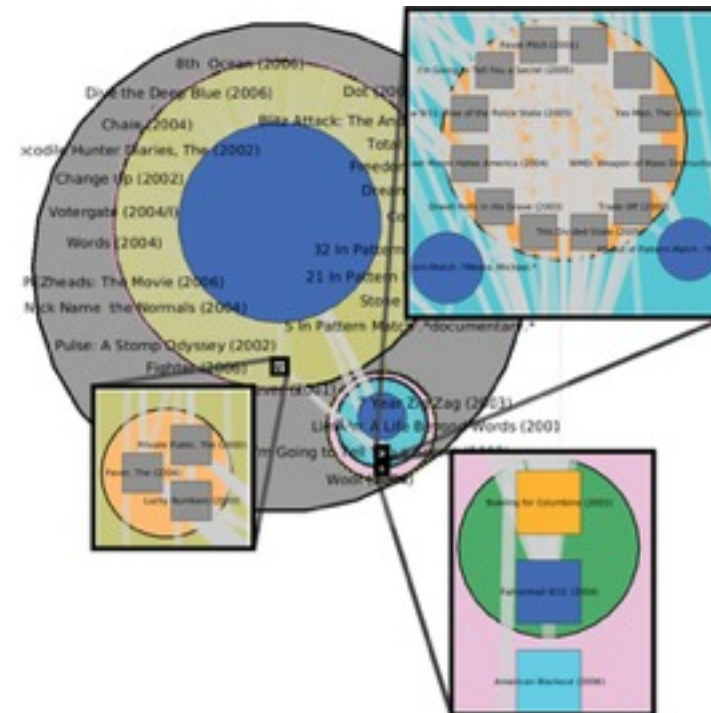


**TreeJuxtaposer**

Daniel Archambault



David Auber  
(Bordeaux)



- TopoLayout**
- SPF**
- Grouse**
- GrouseFlocks**
- TugGraph**

# Evaluation: Graph drawing

T

P

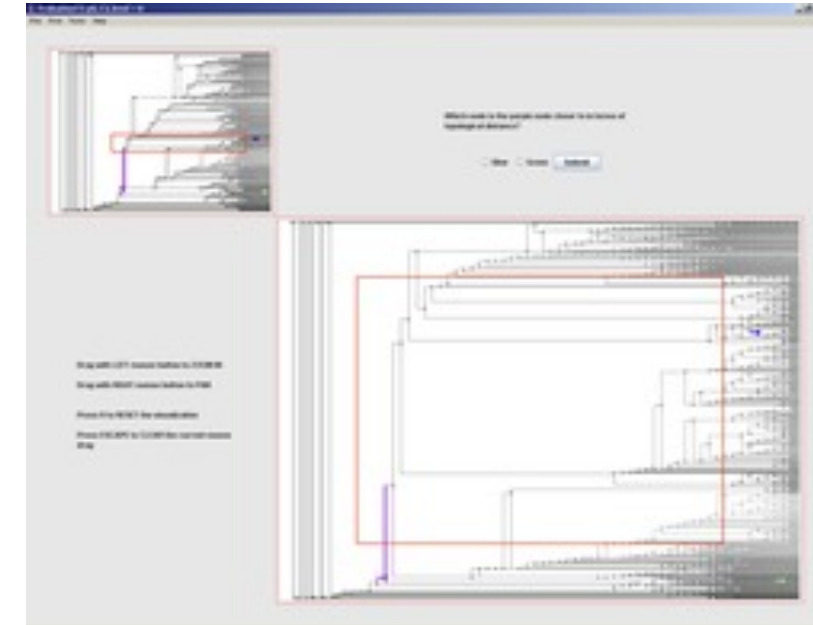
F

E

Dmitry Nekrasovski Adam Bodnar



Joanna McGrenere  
(UBC)

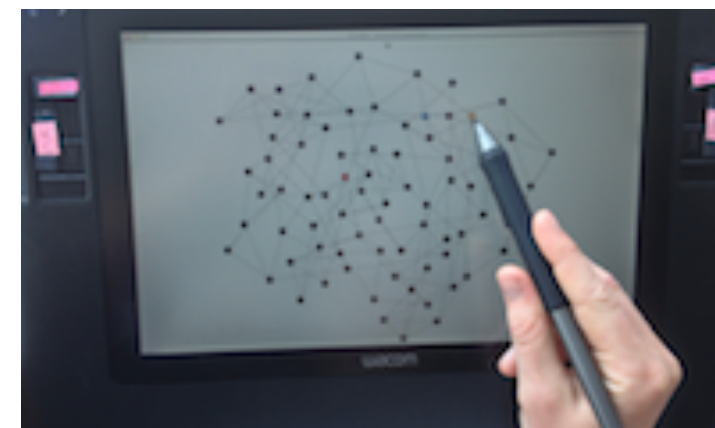


Stretch and squish navigation

Jessica Dawson



Joanna McGrenere  
(UBC)



Search set model of path tracing

# Technique-driven: Dimensionality reduction

T

P

F

E

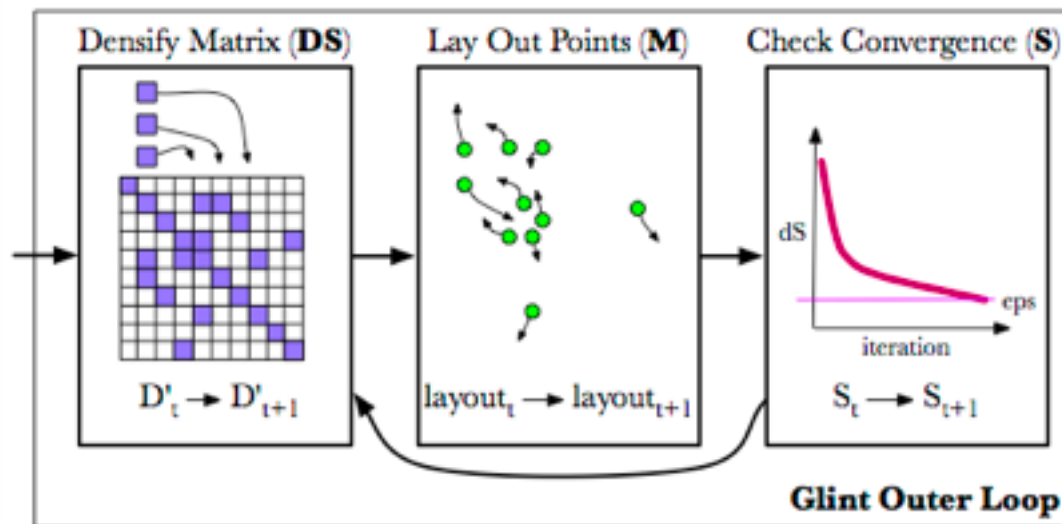
Stephen Ingram



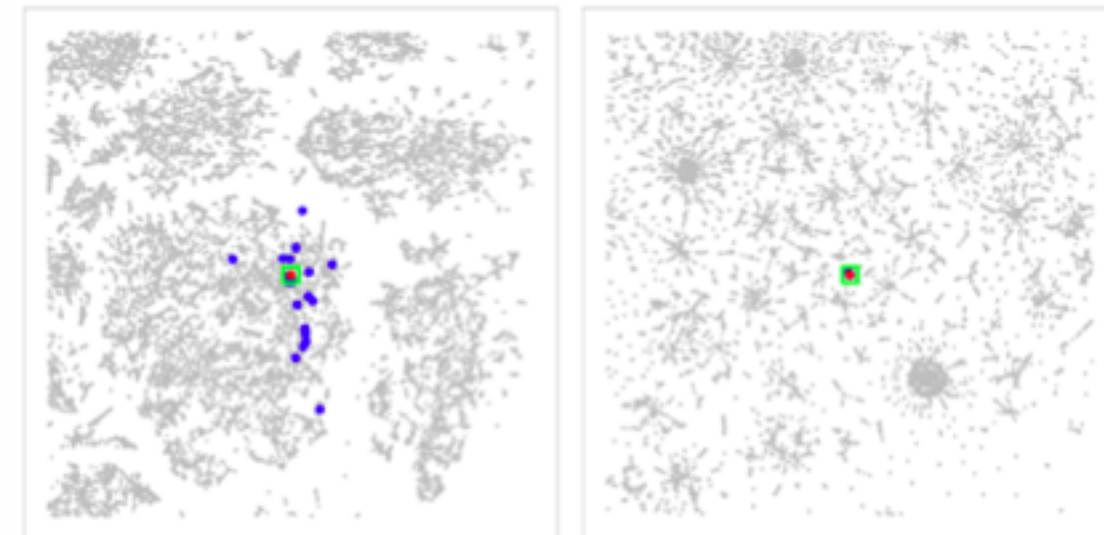
Glimmer



DimStiller



Glint



QSNE

# Evaluation: Dimensionality reduction

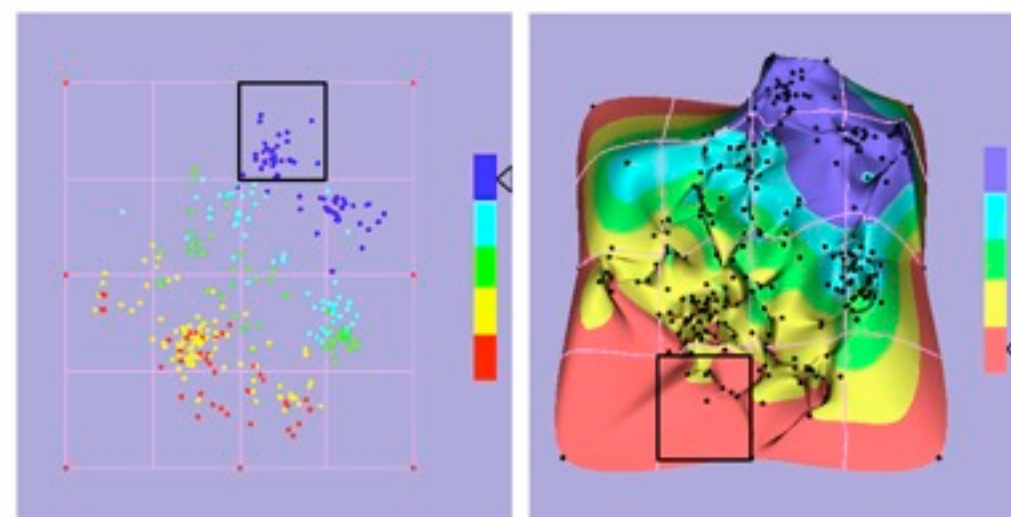
T

P

F

E

Melanie Tory



Points vs landscapes for dimensionally reduced data



Guidance on DR & scatterplot choices

Michael Sedlmair



Melanie Tory (UVic)



Taxonomy of cluster separation factors

# Problem-driven: Genomics

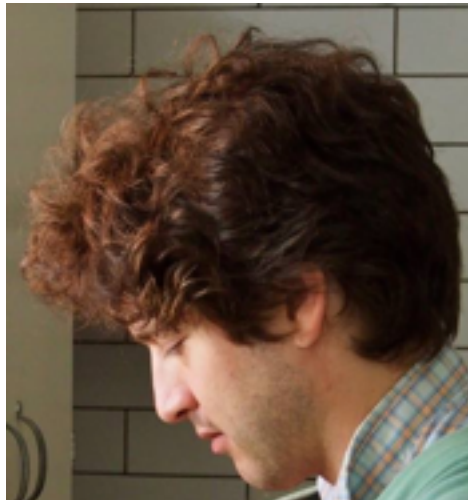
T

P

F

E

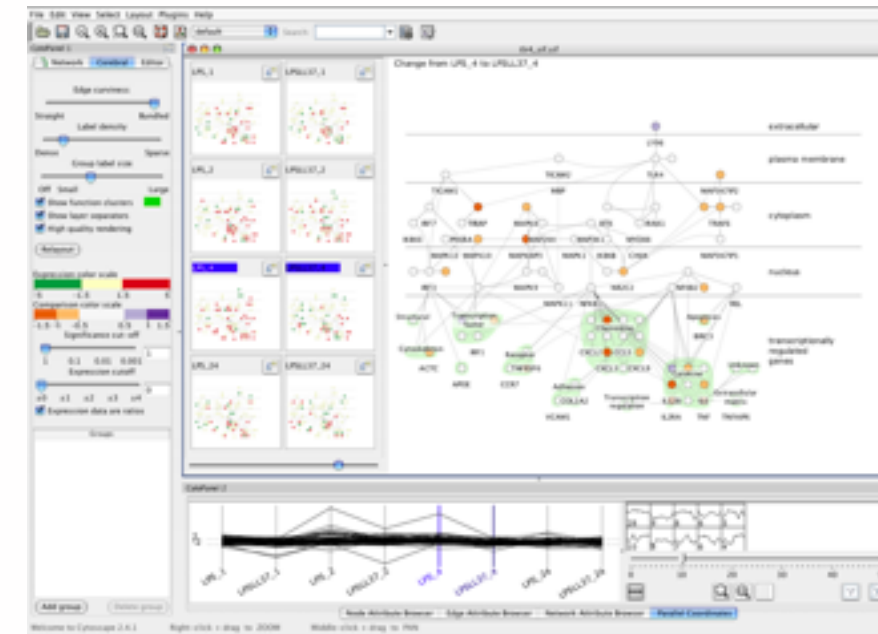
Aaron Barsky



Jenn Gardy  
(Microbio)



Robert Kincaid  
(Agilent)

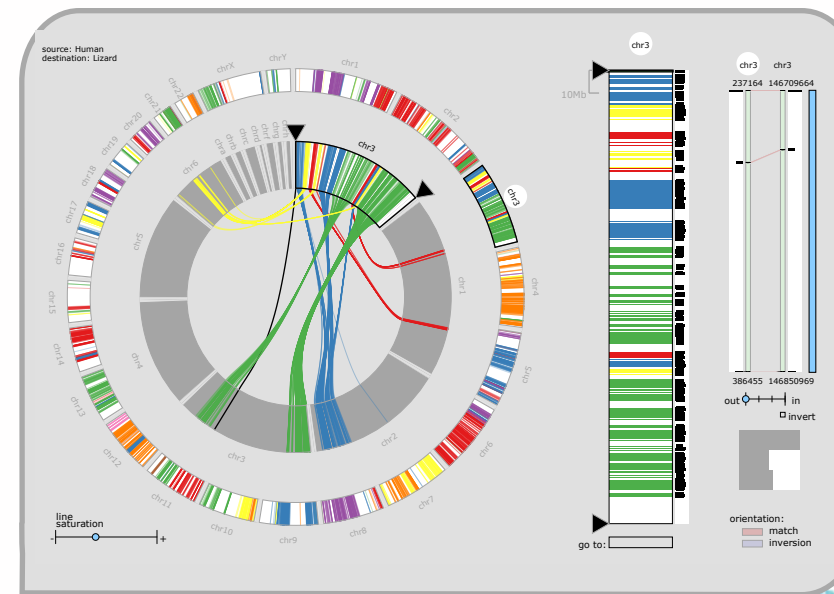


Cerebral

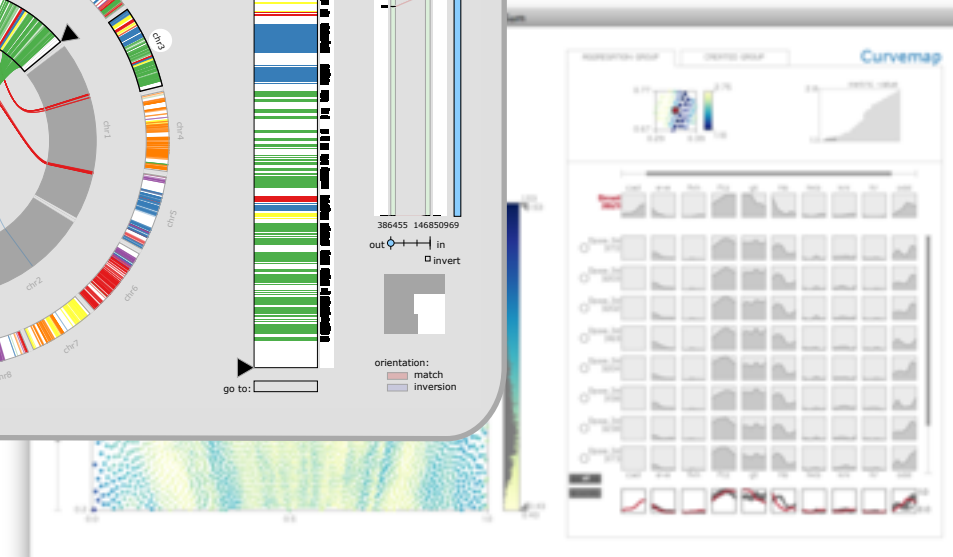
Miriah Meyer



Hanspeter Pfister  
(Harvard)



MizBee



MulteeSum, Pathline

# Problem-driven: Genomics, fisheries

T

P

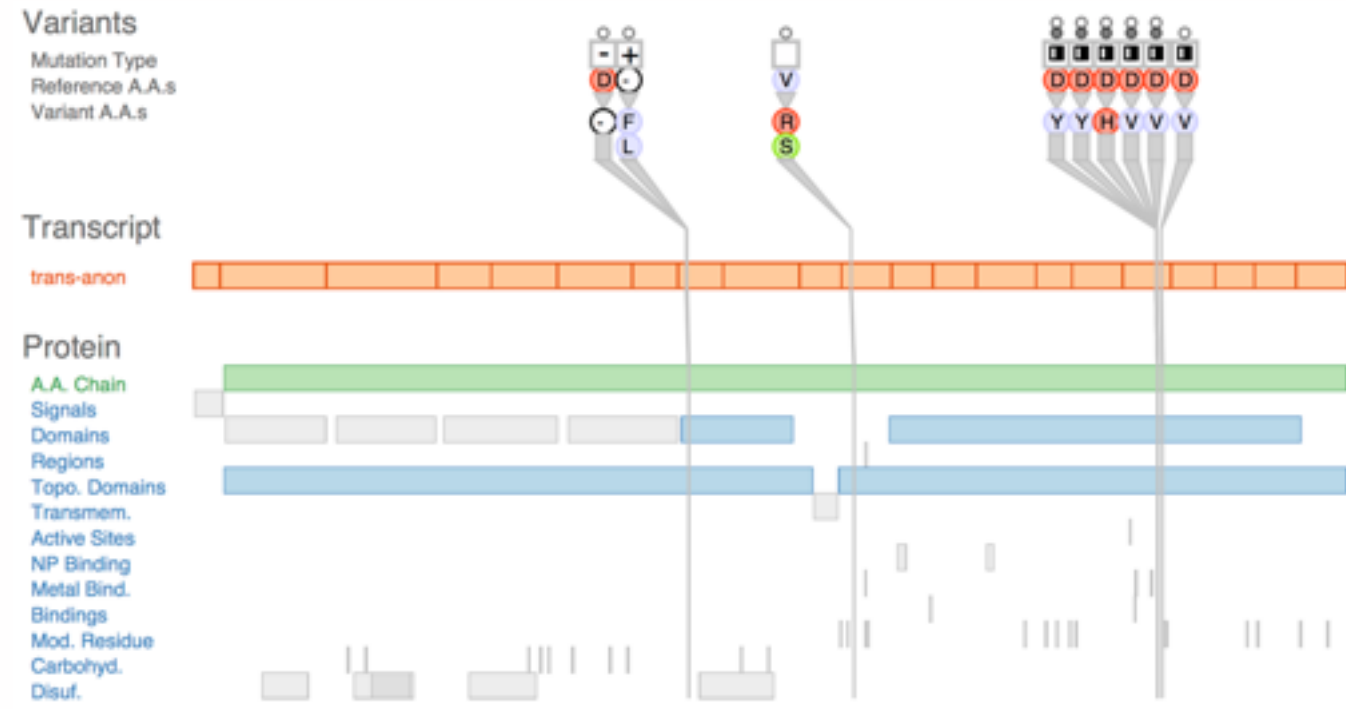
F

E

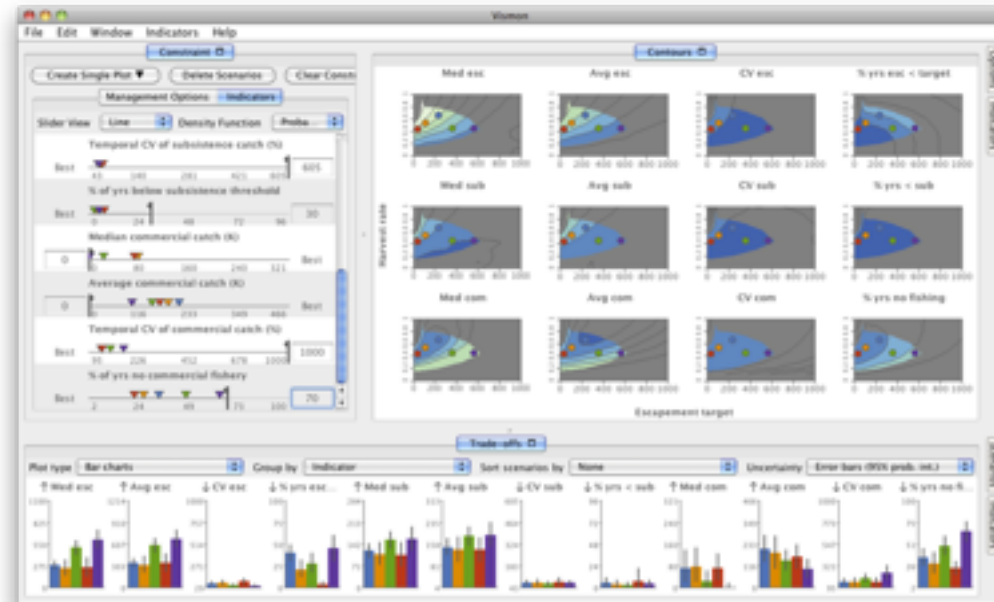
Joel Ferstay



Cydney Nielsen  
(BC Cancer)



Variant View



Vismon

Maryam Booshehrian



Torsten Moeller  
(SFU)



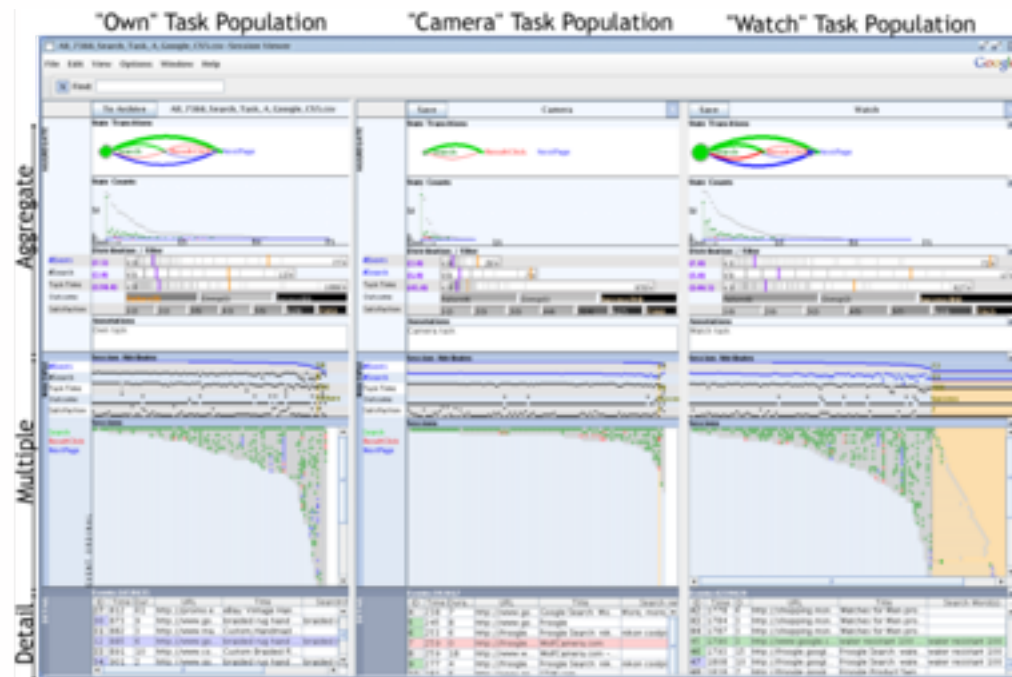
# Problem-driven: Many domains

T

P

F

E



**SessionViewer: web log analysis**

**Heidi Lam**



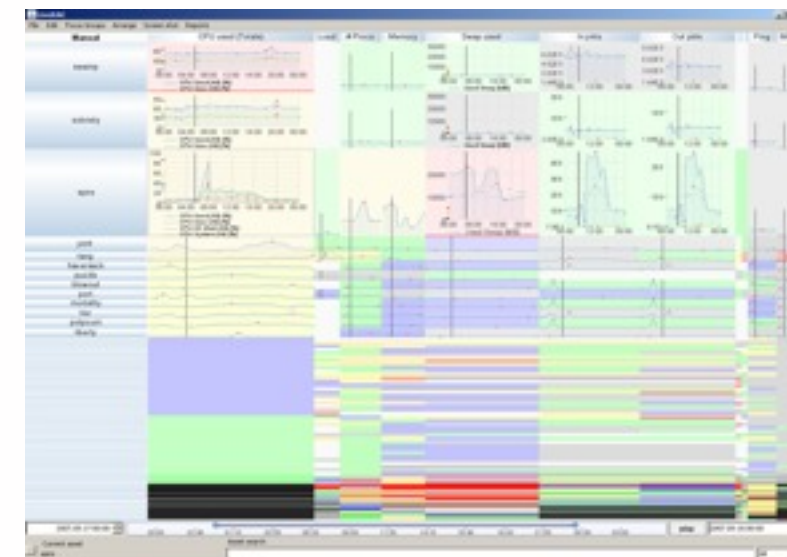
**Diane Tang  
(Google)**



**Peter McLachlan**



**Stephen North  
(AT&T Research)**



**LiveRAC: systems time-series**



# Evaluation: Focus+Context

T

P

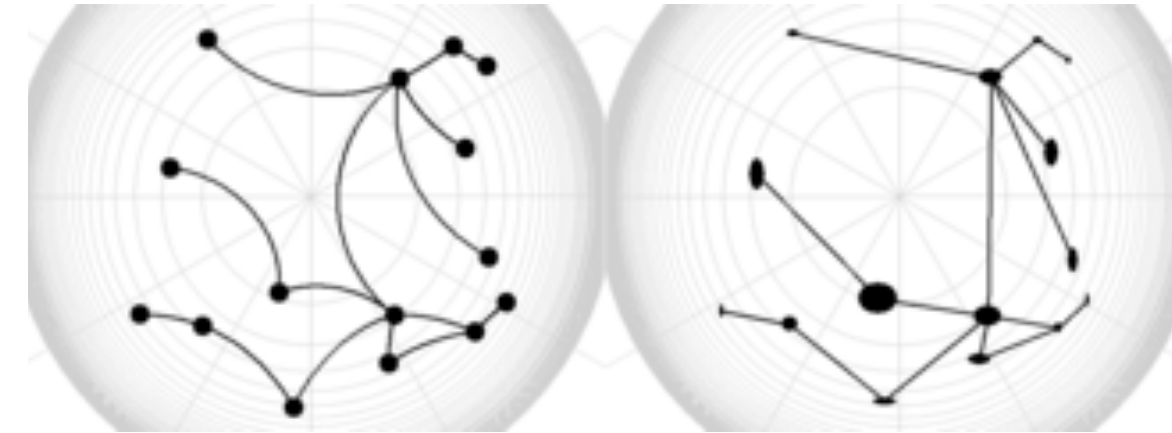
F

E

Heidi Lam



Ron Rensink  
(UBC)

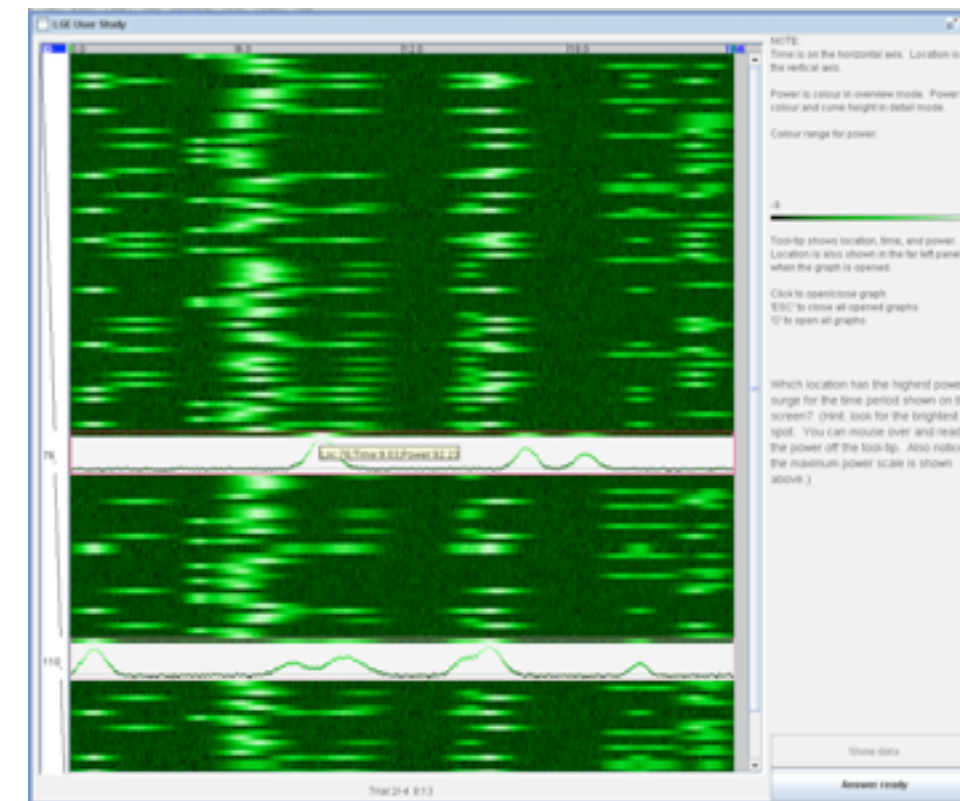


Distortion impact on search/memory

Heidi Lam



Robert Kincaid  
(Agilent)



Separate vs integrated views

# Journalism

T

P

F

E

Matt Brehmer



Stephen Ingram



Jonathan Stray  
(Assoc Press)



Overview

Johanna Fulda  
(Sud. Zeitung)



Matt Brehmer



TimeLineCurator

# Theoretical foundations

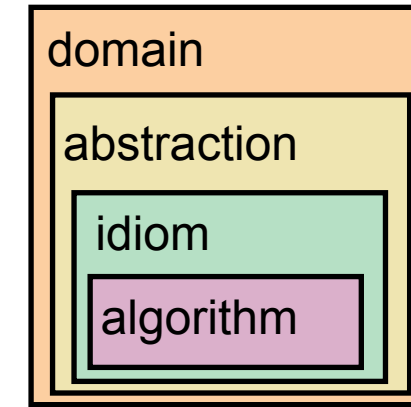
T F P  
E

- Visual Encoding Pitfalls

- Unjustified Visual Encoding
- Hammer In Search Of Nail
- 2D Good, 3D Better
- Color Cacophony
- Rainbows Just Like In The Sky

- Strategy Pitfalls

- What I Did Over My Summer
- Least Publishable Unit
- Dense As Plutonium
- Bad Slice and Dice



**Nested Model**

## Papers Process & Pitfalls

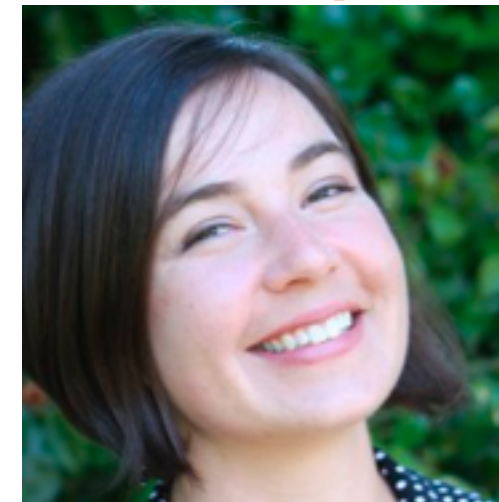


**Design Study Methodology**

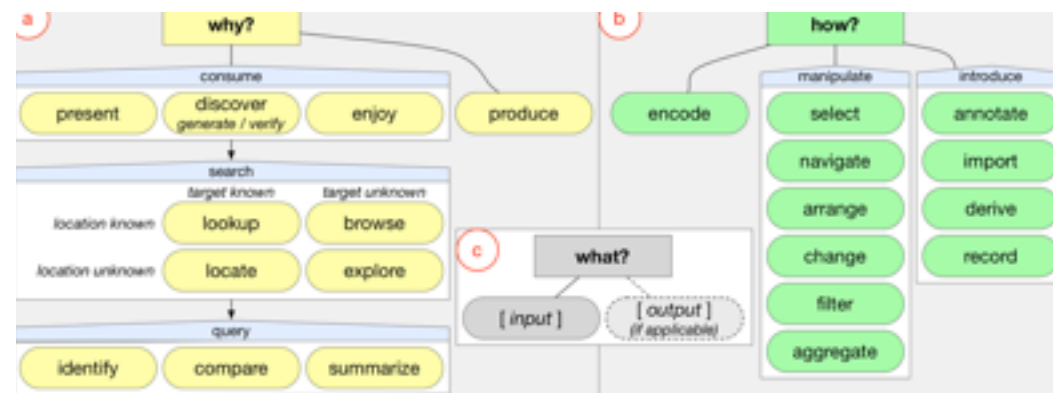
**Michael Sedlmair**



**Miriah Meyer**



**Matt Brehmer**



**Abstract Tasks**

# More Information

[@tamaramunzner](https://twitter.com/tamaramunzner)

- this talk

<http://www.cs.ubc.ca/~tmm/talks.html#vad16infoplus>

- book page (including tutorial lecture slides)

<http://www.cs.ubc.ca/~tmm/vadbook>

– 20% promo code for book+ebook combo:  
HVN17

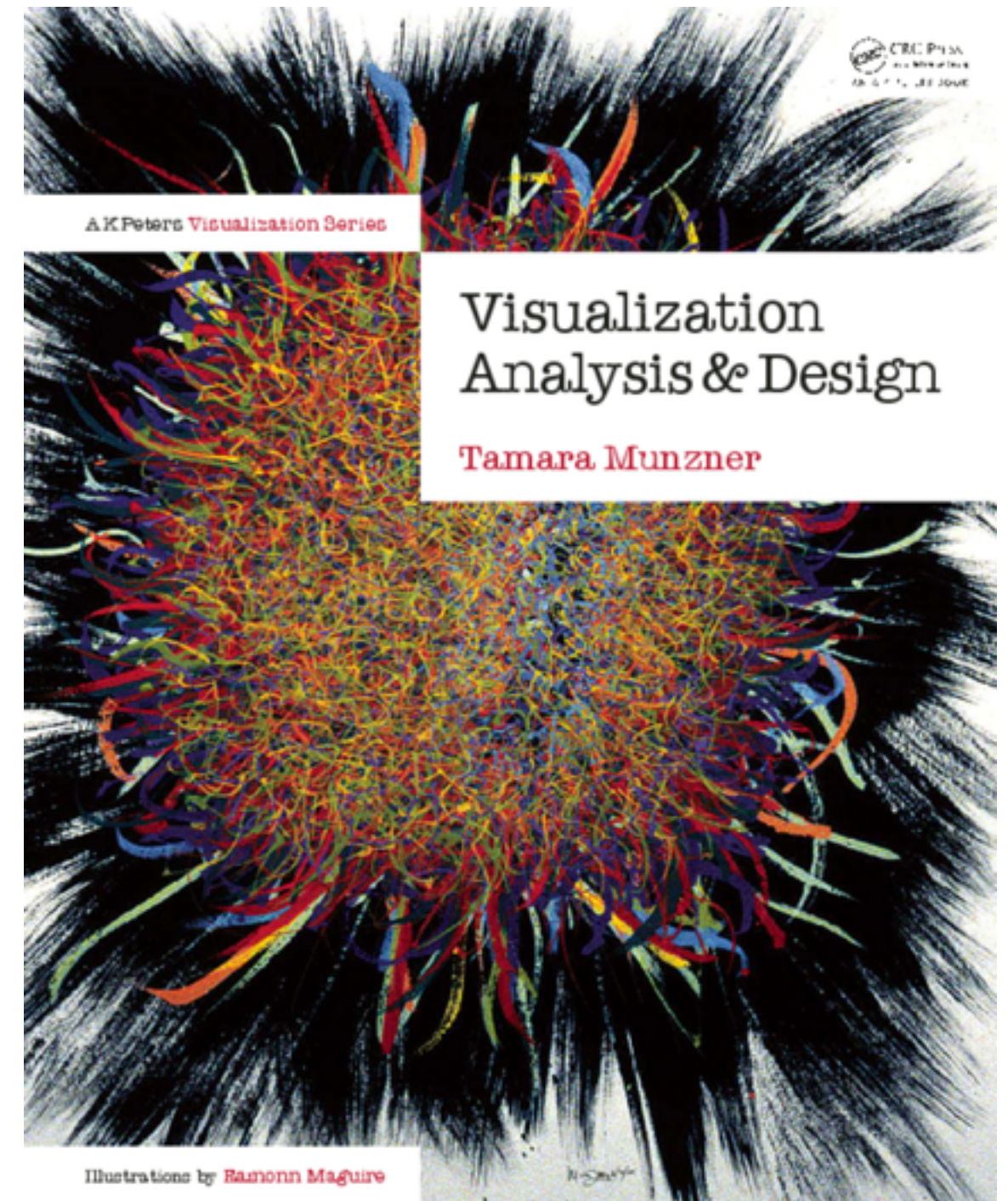
– <http://www.crcpress.com/product/isbn/9781466508910>

– illustrations: Eamonn Maguire

- papers, videos, software, talks, courses

<http://www.cs.ubc.ca/group/infovis>

<http://www.cs.ubc.ca/~tmm>



Visualization Analysis and Design.  
Munzner. A K Peters Visualization Series, CRC Press, Visualization Series, 2014.