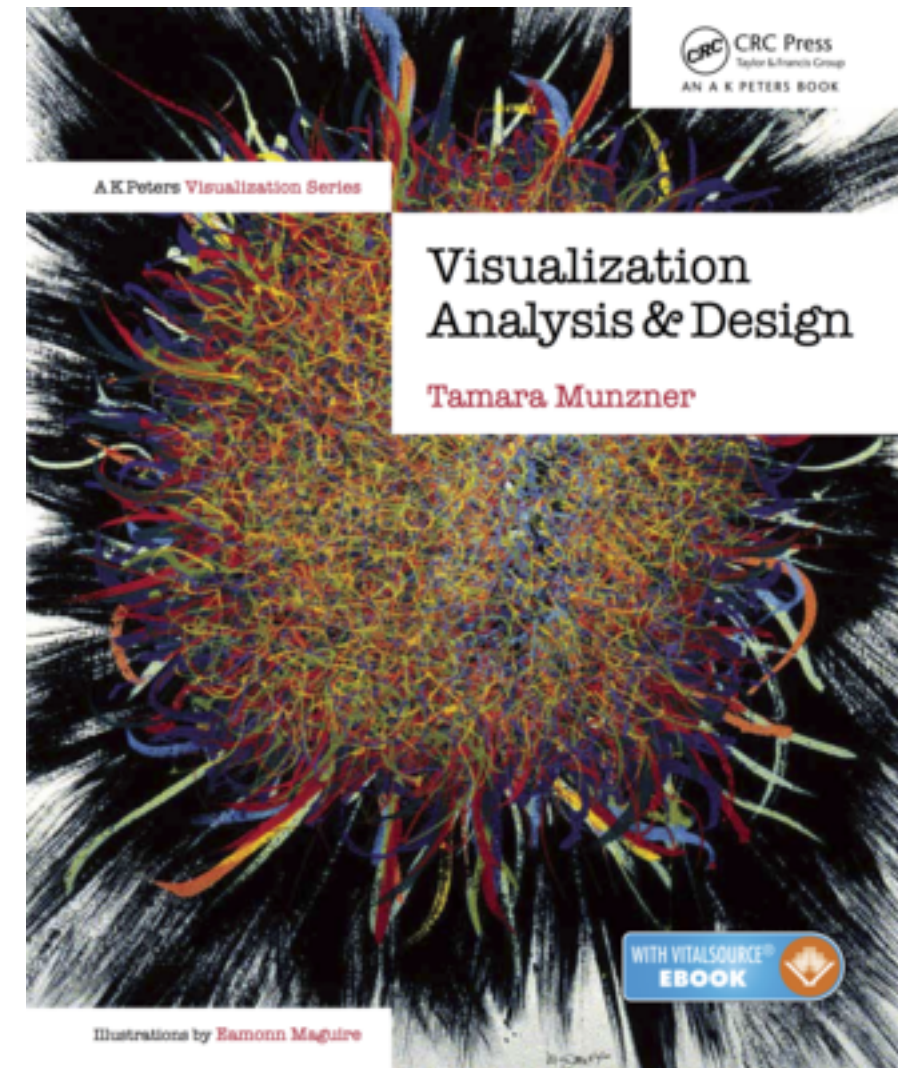# Visualization Analysis & Design

Tamara Munzner
Department of Computer Science
University of British Columbia

NASA Goddard Information Science and Technology Colloquium
December 14 2016, Greenbelt MD

http://www.cs.ubc.ca/~tmm/talks.html#vad16nasa   **@tamaramunzner**

# Visualization (vis) defined & motivated

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.

- human in the loop needs the details
  - doesn't know exactly what questions to ask in advance
  - longterm exploratory analysis
  - presentation of known results
  - stepping stone towards automation: refining, trustbuilding
- external representation: perception vs cognition
- intended task, measurable definitions of effectiveness

more at:
Visualization Analysis and Design, Chapter 1.
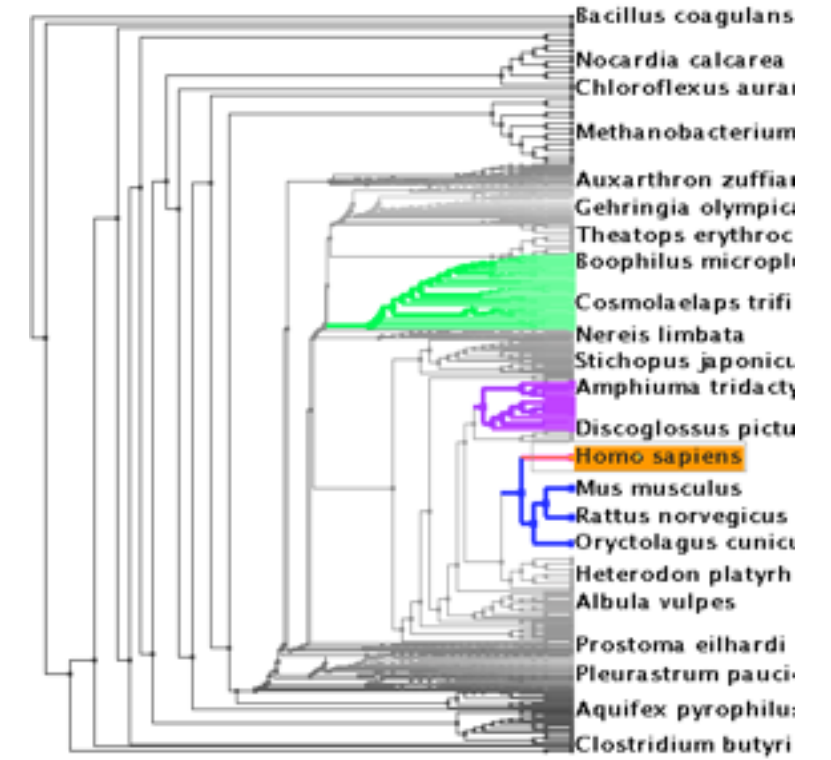*Munzner. AK Peters Visualization Series, CRC Press, 2014.*
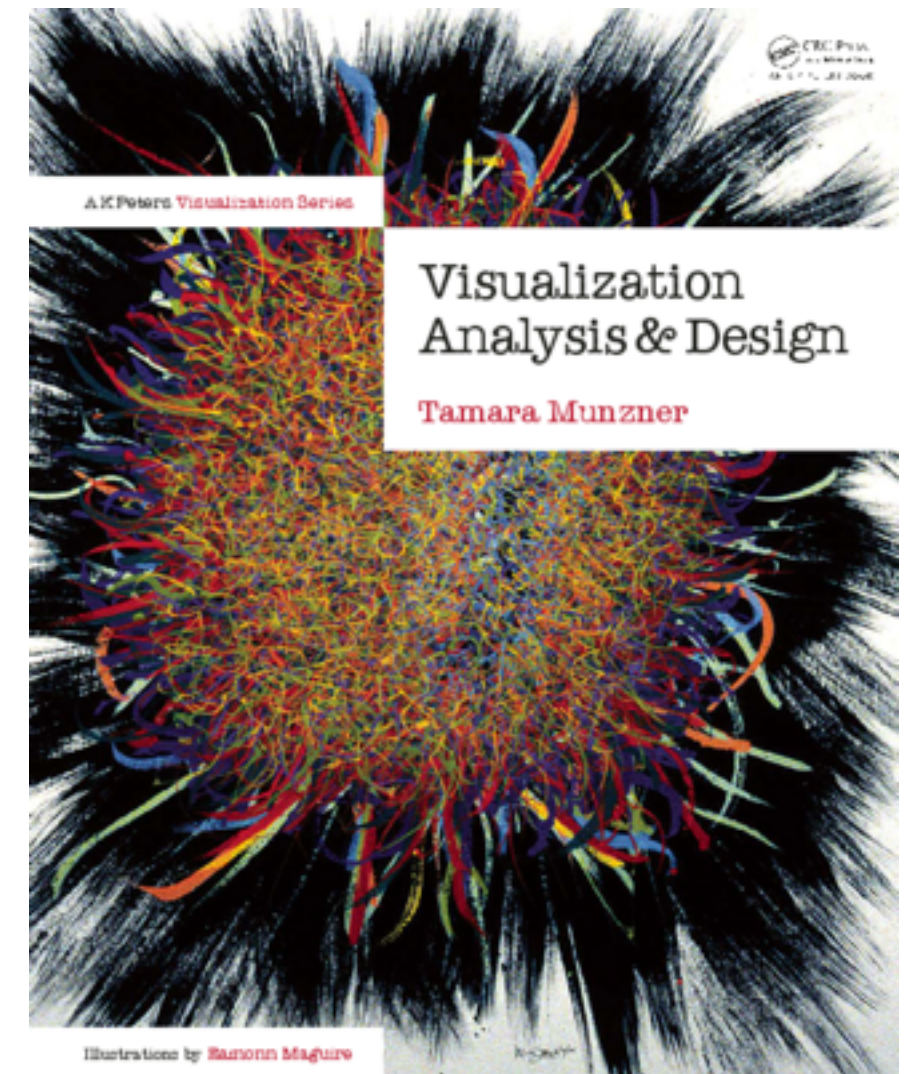
Why analyze?

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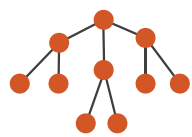
[TreeJuxtaposer: Scalable Tree Comparison Using Focus +Context With Guaranteed Visibility. ACM Trans. on Graphics (Proc. SIGGRAPH) 22:453– 462, 2003.]
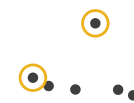
What?
→ Tree

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?

3

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
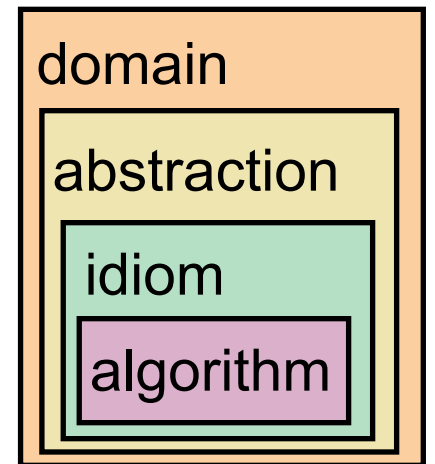
[A Nested Model of Visualization Design and Validation.
Munzner. IEEE TVCG 15(6):921-928, 2009 (Proc. InfoVis 2009).]
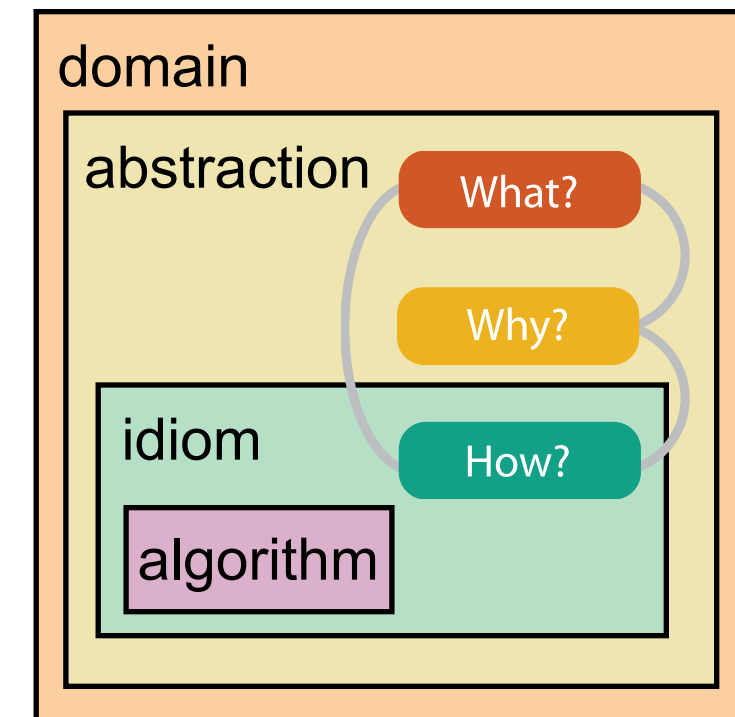
[A Multi-Level Typology of Abstract Visualization Tasks
Brehmer and Munzner. IEEE TVCG 19(12):2376-2385, 2013 (Proc. InfoVis 2013).]
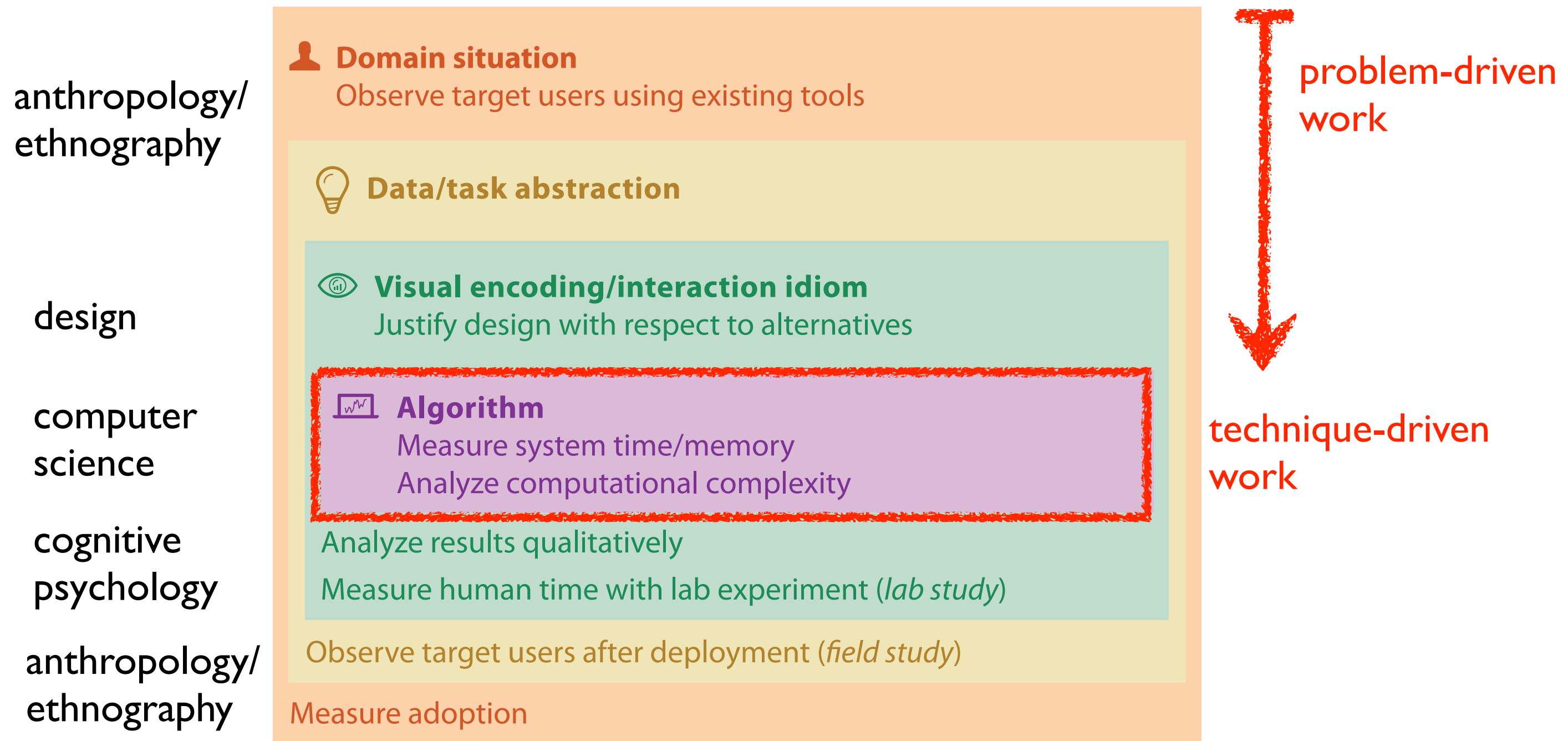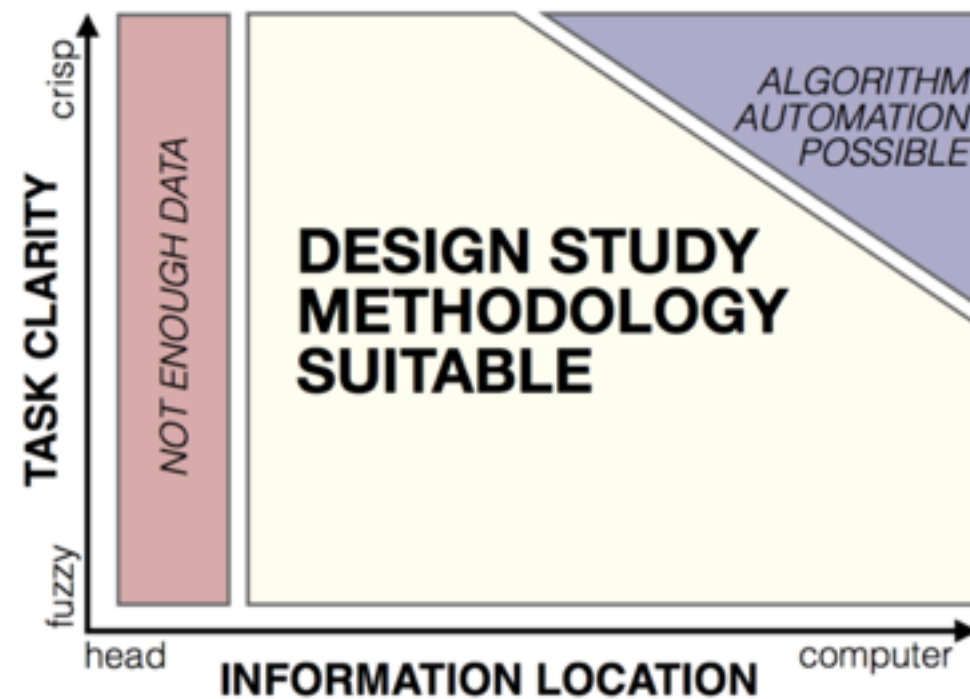
Why is validation difficult?

- different ways to get it wrong at each level

Domain situation
You misunderstood their needs

Data/task abstraction
You're showing them the wrong thing

Visual encoding/interaction idiom
The way you show it doesn't work

Algorithm
Your code is too slow
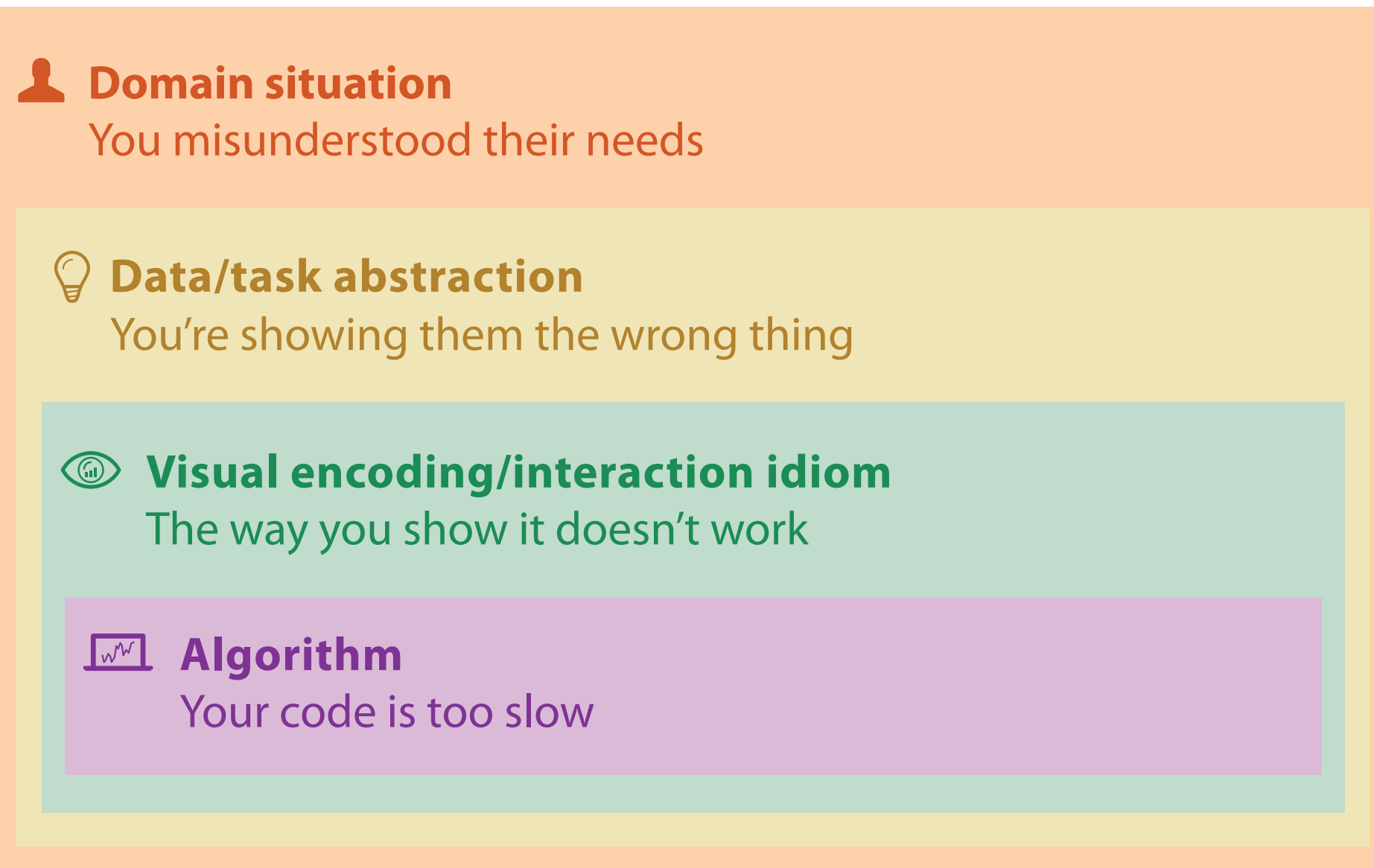
Why is validation difficult?

- solution: use methods from different fields at each level

anthropology/ethnography

Domain situation
Observe target users using existing tools

Data/task abstraction

design

Visual encoding/interaction idiom
Justify design with respect to alternatives

computer science

Algorithm
Measure system time/memory
Analyze computational complexity

cognitive psychology

Analyze results qualitatively
Measure human time with lab experiment (*lab study*)

anthropology/ethnography

Observe target users after deployment (*field study*)

Measure adoption

problem-driven work

technique-driven work

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

Design Study Methodology

Reflections from the Trenches and from the Stacks

http://www.cs.ubc.ca/labs/imager/tr/2012/dsm/

Michael Sedlmair

Miriah Meyer

Tamara Munzner
@tamaramunzner
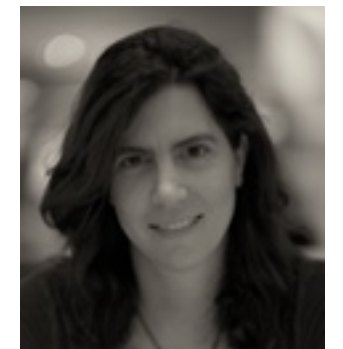
Design Studies: Lessons learned after 21 of them
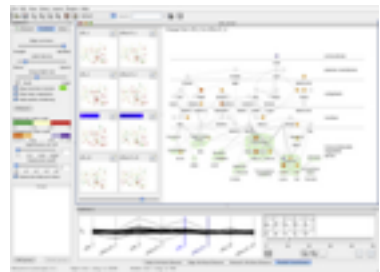
Cerebral
genomics
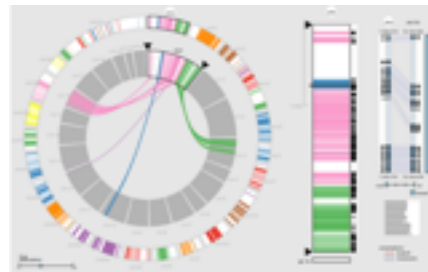
MizBee
genomics
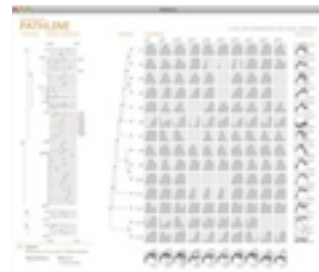
Pathline
genomics
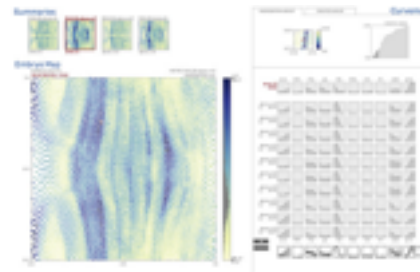
MulteeSum
genomics
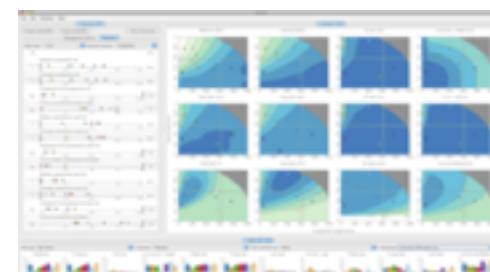
Vismon
fisheries management
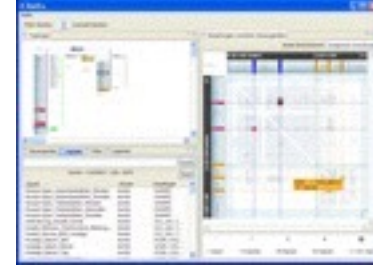
QuestVis
sustainability

WiKeVis
in-car networks
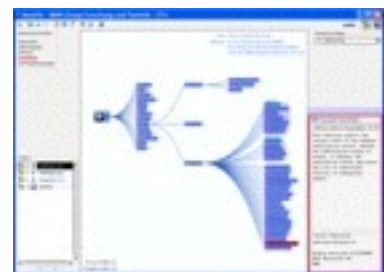
MostVis
in-car networks
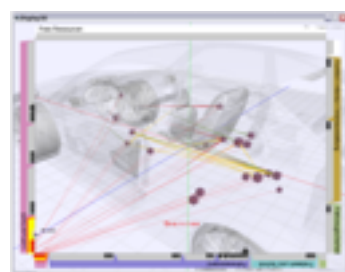
Car-X-Ray
in-car networks

ProgSpy2010
in-car networks

RelEx
in-car networks

Cardiogram
in-car networks

AutobahnVis
in-car networks

VisTra
in-car networks

Constellation
linguistics
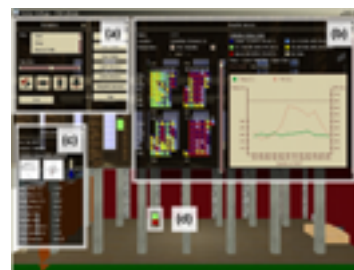
LibVis
cultural heritage
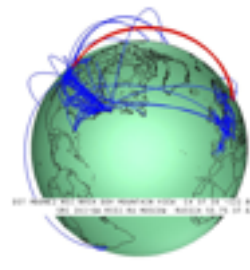
Caidants
multicast

SessionViewer
web log analysis
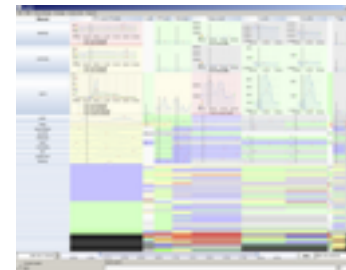
LiveRAC
server hosting
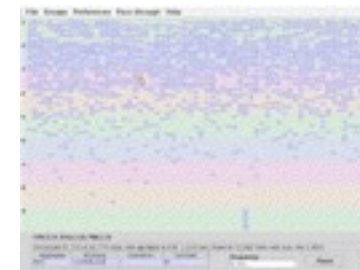
PowerSetViewer
data mining
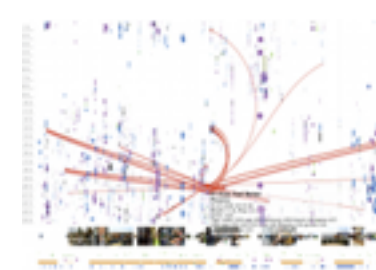
LastHistory
music listening

Methodology for Problem-Driven Work

- definitions

- 9-stage framework

- 32 pitfalls
  and how to avoid them
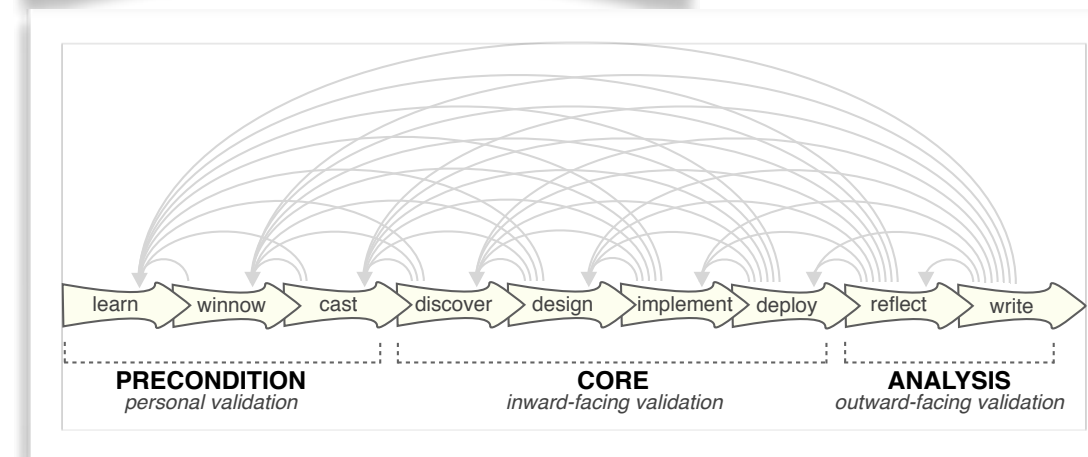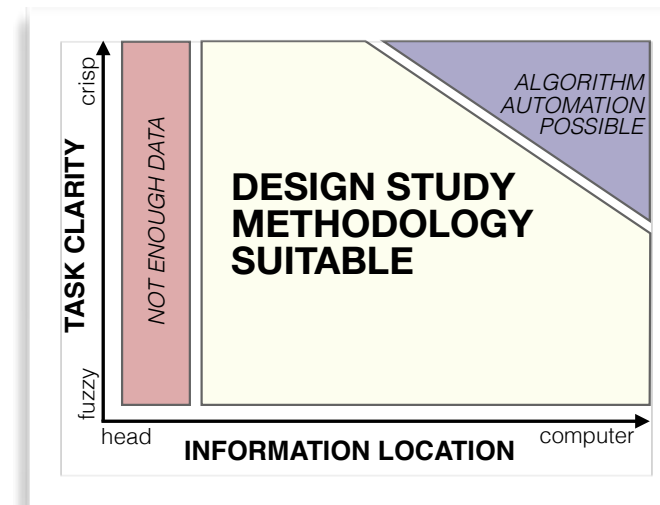
PF-1	premature advance: jumping forward over stages	general
PF-2	premature start: insufficient knowledge of vis literature	learn
PF-3	premature commitment: collaboration with wrong people	winnow
PF-4	no real data available (yet)	winnow
PF-5	insufficient time available from potential collaborators	winnow
PF-6	no need for visualization: problem can be automated	winnow
PF-7	researcher expertise does not match domain problem	winnow
PF-8	no need for research: engineering vs. research project	winnow
PF-9	no need for change: existing tools are good enough	winnow
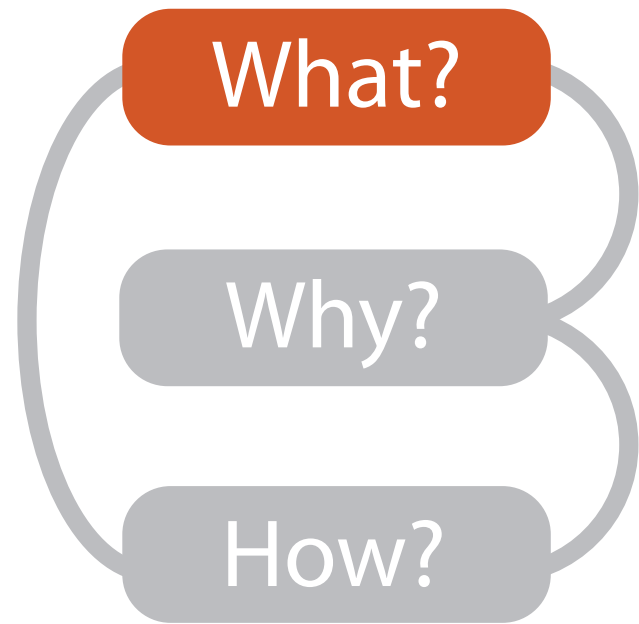
What?

What?
Why?
How?

Datasets

Attributes

➜ **Data Types**

➜ Items   ➜ Attributes   ➜ Links   ➜ Positions   ➜ Grids

➜ **Data and Dataset Types**

Tables	Networks & Trees	Fields	Geometry	Clusters, Sets, Lists
Items	Items (nodes)	Grids	Items	Items
Attributes	Links	Positions	Positions	
	Attributes	Attributes		

➜ **Dataset Types**

➜ Tables

Attributes (columns)
Items (rows)
Cell containing value

➜ Networks

Link
Node (item)

➜ Fields (Continuous)

Grid of positions
Cell
Attributes (columns)
Value in cell

➜ *Multidimensional Table*

Key 1
Key 2
Value in cell
Attributes

➜ *Trees*

➜ Geometry (Spatial)

Position

➜ **Dataset Availability**

➜ Static

➜ Dynamic

➜ **Attribute Types**

➜ Categorical

➜ Ordered

➜ *Ordinal*

➜ *Quantitative*

➜ **Ordering Direction**

➜ Sequential

➜ Diverging

➜ Cyclic

10

Types: Datasets and data

➔ Dataset Types

  ➔ Tables

  ➔ Networks

Attributes (columns)

Items
(rows)

Cell containing value

Link

Node
(item)
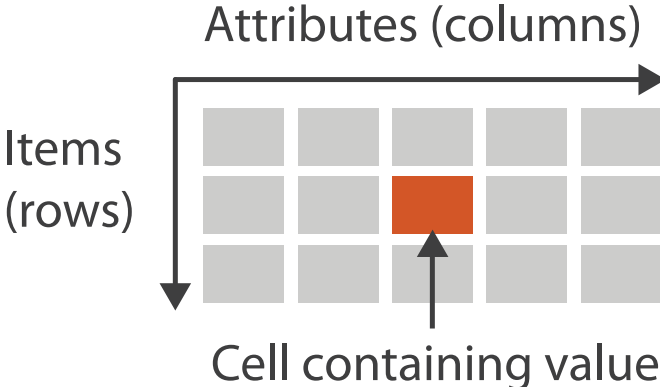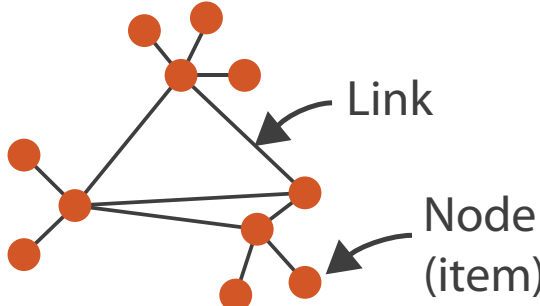
➔ Spatial

➔ Fields (Continuous)

➔ Geometry (Spatial)

Grid of positions

Cell

Attributes (columns)

Value in cell

Position
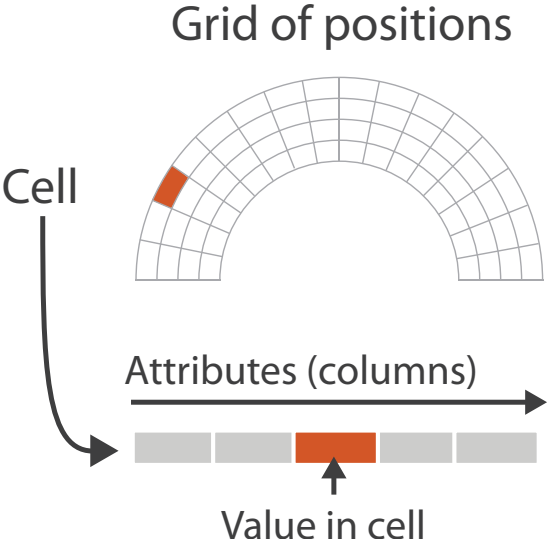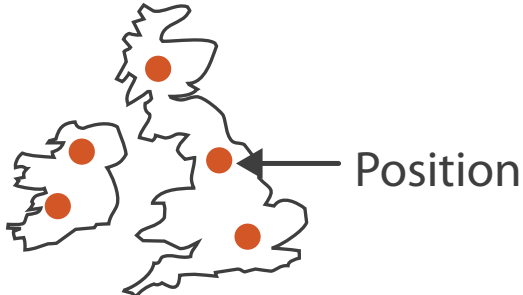
➔ Attribute Types

  ➔ Categorical

  ➔ Ordered

    ➔ *Ordinal*

    ➔ *Quantitative*

Actions

Targets
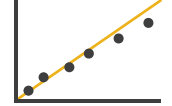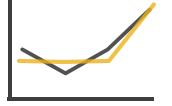
What?

Why?

How?

➔ **Analyze**

➔ Consume

➔ *Discover* ➔ *Present* ➔ *Enjoy*

➔ Produce

➔ *Annotate* ➔ *Record* ➔ *Derive*
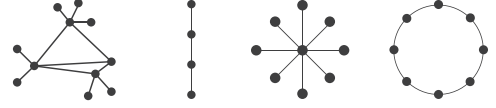
➔ **Search**

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

➔ **Query**
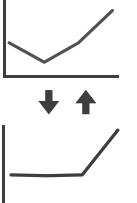
➔ Identify ➔ Compare ➔ Summarize

➔ **All Data**

➔ Trends ➔ Outliers ➔ Features

➔ **Attributes**

➔ One   ➔ Many

➔ *Distribution*   ➔ *Dependency* ➔ *Correlation* ➔ *Similarity*

➔ *Extremes*
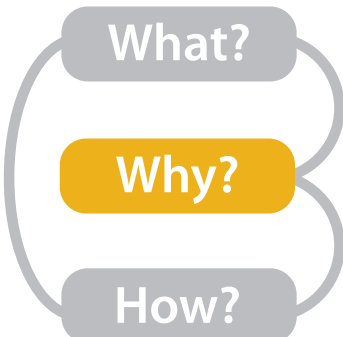
➔ **Network Data**

➔ Topology

➔ *Paths*

➔ **Spatial Data**

➔ Shape
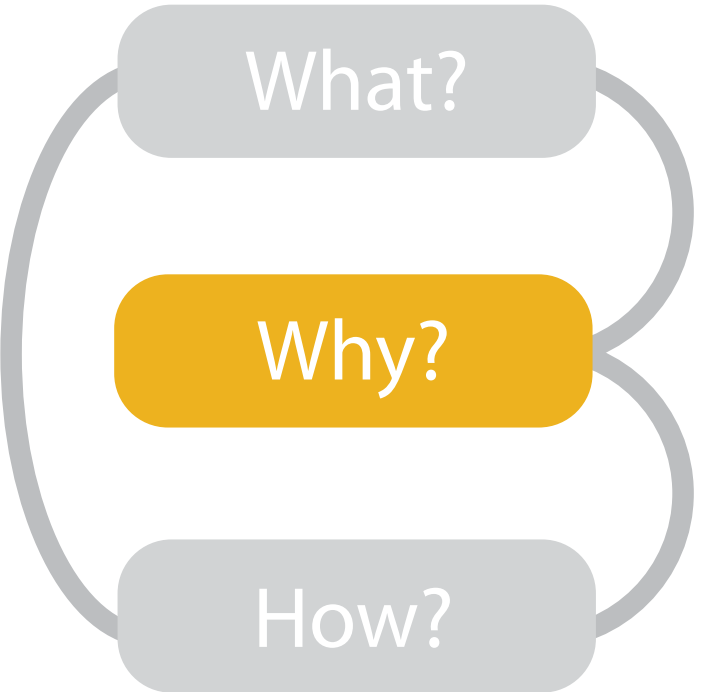
• {action, target} pairs
  – *discover distribution*
  – *compare trends*
  – *locate outliers*
  – *browse topology*

What?

Why?

How?

12

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?
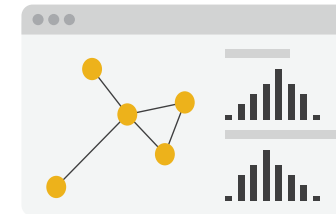   - one, some, all
- independent choices

➔ **Analyze**

   ➔ Consume
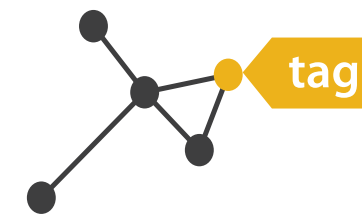
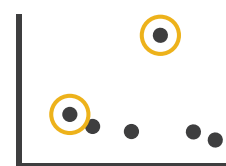   ➔ *Discover*   ➔ *Present*   ➔ *Enjoy*

   ➔ Produce

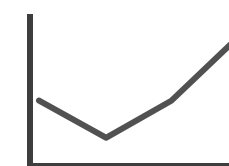   ➔ *Annotate*   ➔ *Record*   ➔ *Derive*

tag

➔ **Query**

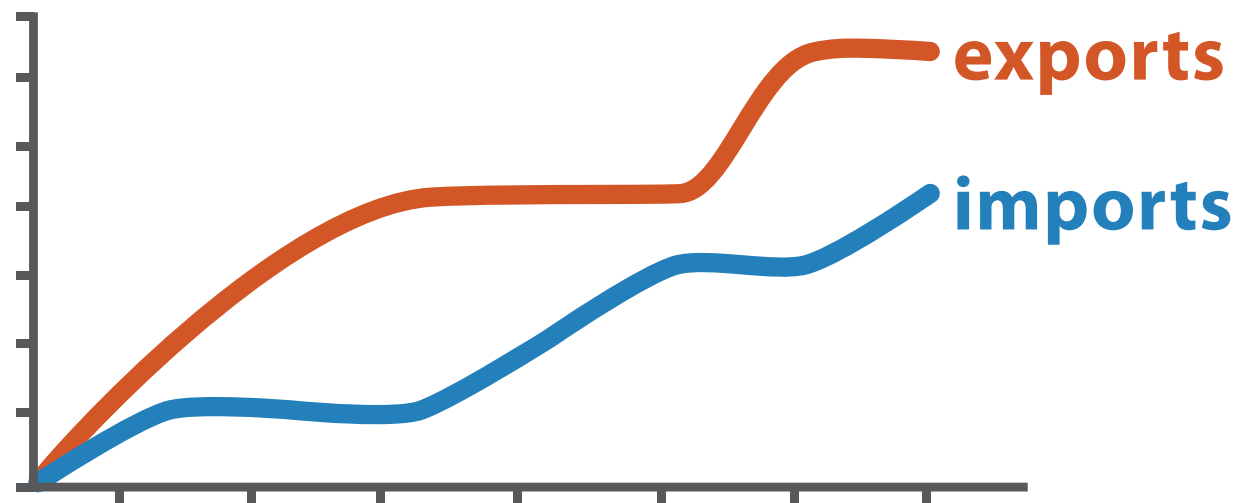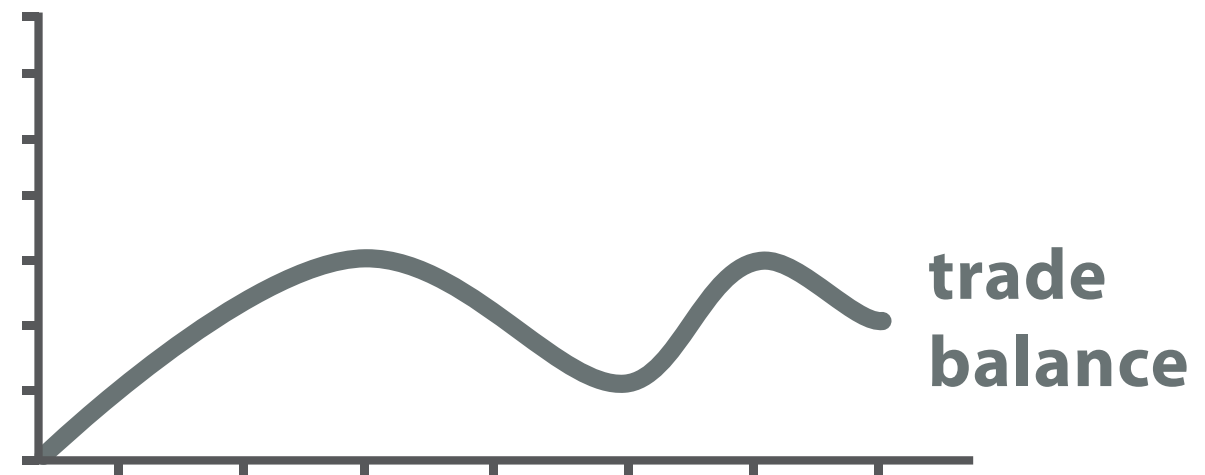   ➔ Identify   ➔ Compare   ➔ Summarize

13

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

*trade balance = exports − imports*

Derived Data

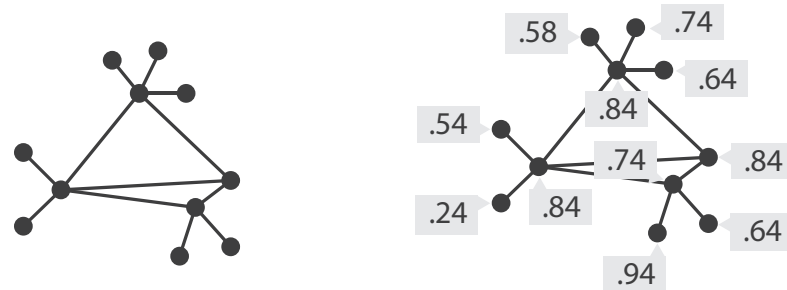# Analysis example: Derive one attribute

- ## 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.]

Task 1

In	Out
Tree	Quantitative attribute on nodes
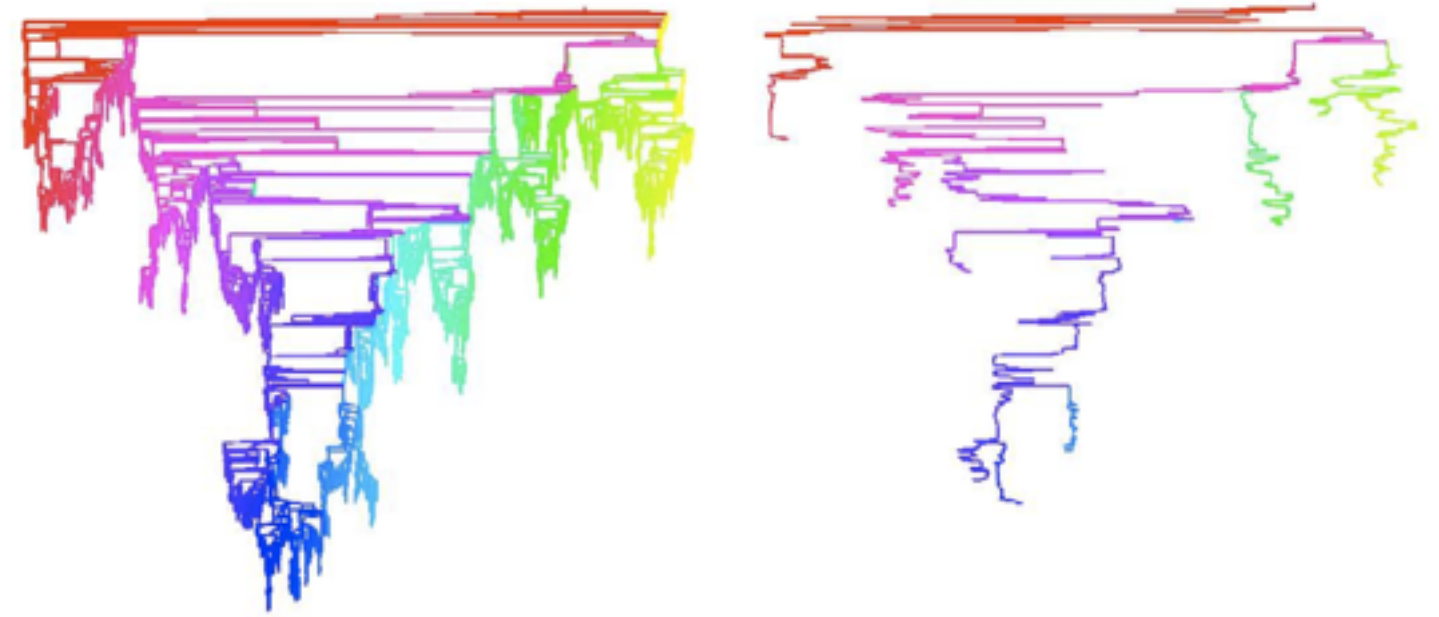
What?

➔ **In** Tree

➔ **Out** Quantitative attribute on nodes

Why?

➔ Derive

Task 2

In		In		Out
Tree	+	Quantitative attribute on nodes		Filtered Tree Removed unimportant parts
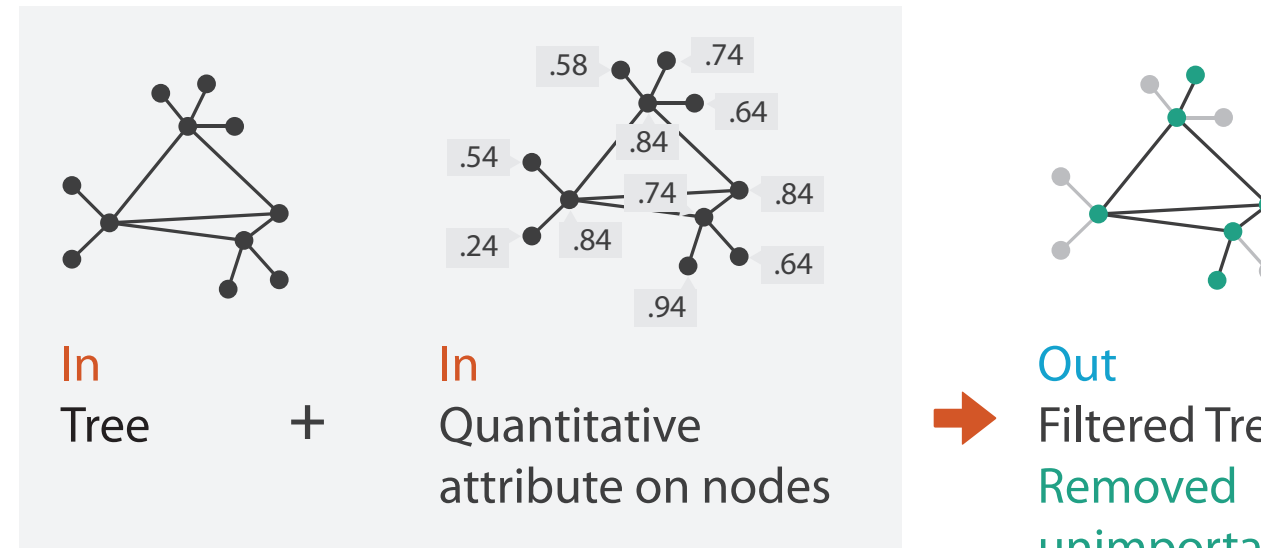
What?

➔ **In** Tree

➔ **In** Quantitative attribute on nodes

➔ **Out** Filtered Tree

Why?

➔ Summarize

➔ Topology

How?

➔ Reduce

➔ Filter

Targets

→ All Data
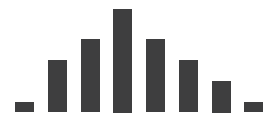
   → Trends   **→ Outliers**   **→ Features**
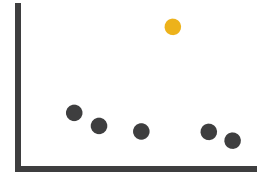
→ Attributes

   → One   **→ Many**

     → Distribution   *→ Dependency*   *→ Correlation*   *→ Similarity*
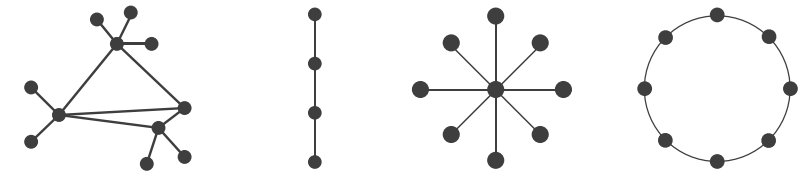
     → Extremes
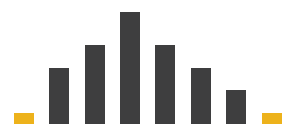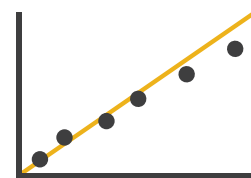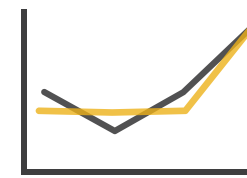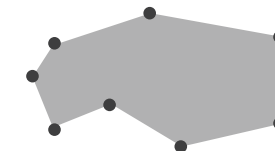
→ Network Data

   → 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
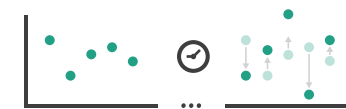
➕ ● ■ ▲

→ 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

→ **Arrange**

➜ Express

➜ Order

➜ Use

➜ Separate

➜ Align

→ **Map**
from categorical and ordered attributes

➜ Color

➜ *Hue* ➜ *Saturation* ➜ *Luminance*

➜ Size, Angle, Curvature, ...

➜ Shape

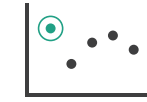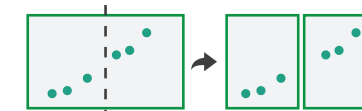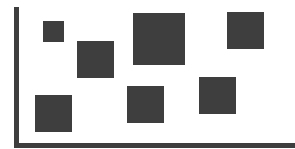➜ Motion
Direction, Rate, Frequency, ...

Definitions: Marks and channels

- **marks**
  - geometric primitives
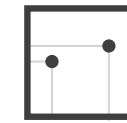
- **channels**
  - control appearance of marks

Points Lines Areas

Position Color

Horizontal Vertical Both
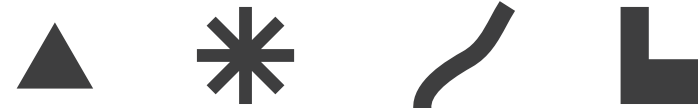
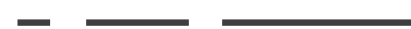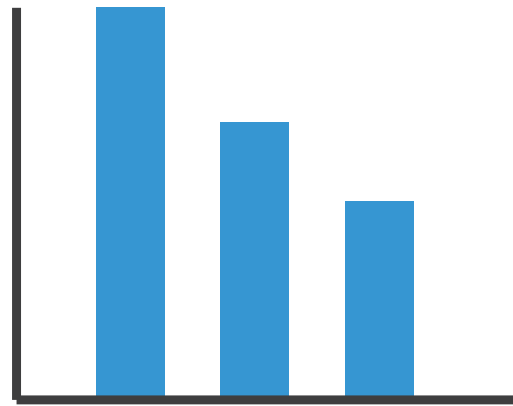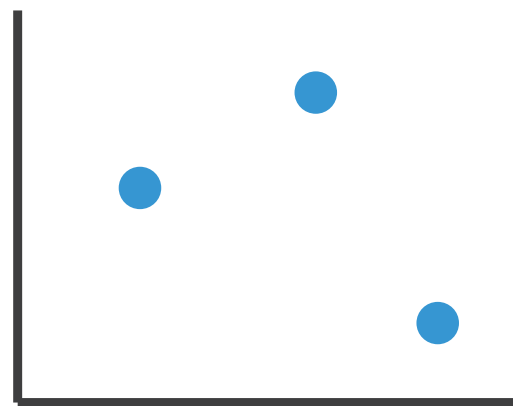Shape Tilt

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
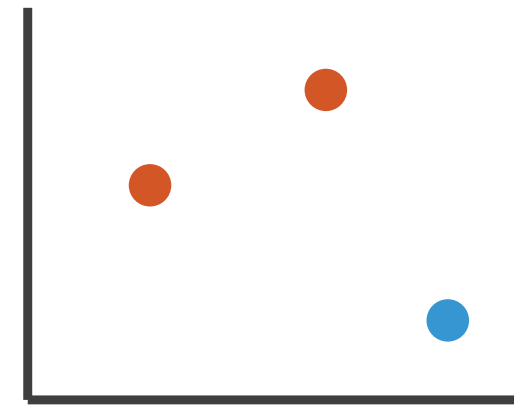
3:
vertical position
horizontal position
color hue

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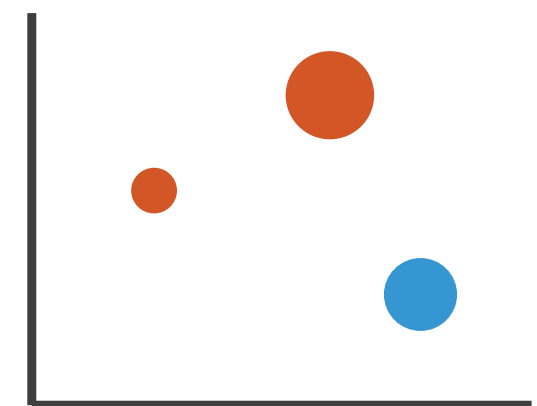
4:
vertical position
horizontal position
color hue
size (area)

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

Same

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

Same

Curvature

Volume (3D size)

Same

→ Identity Channels: Categorical Attributes

Spatial region

Color hue

Motion

Shape

- expressiveness principle
  - match channel and data characteristics

Channels: Rankings

Magnitude Channels: Ordered Attributes

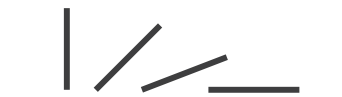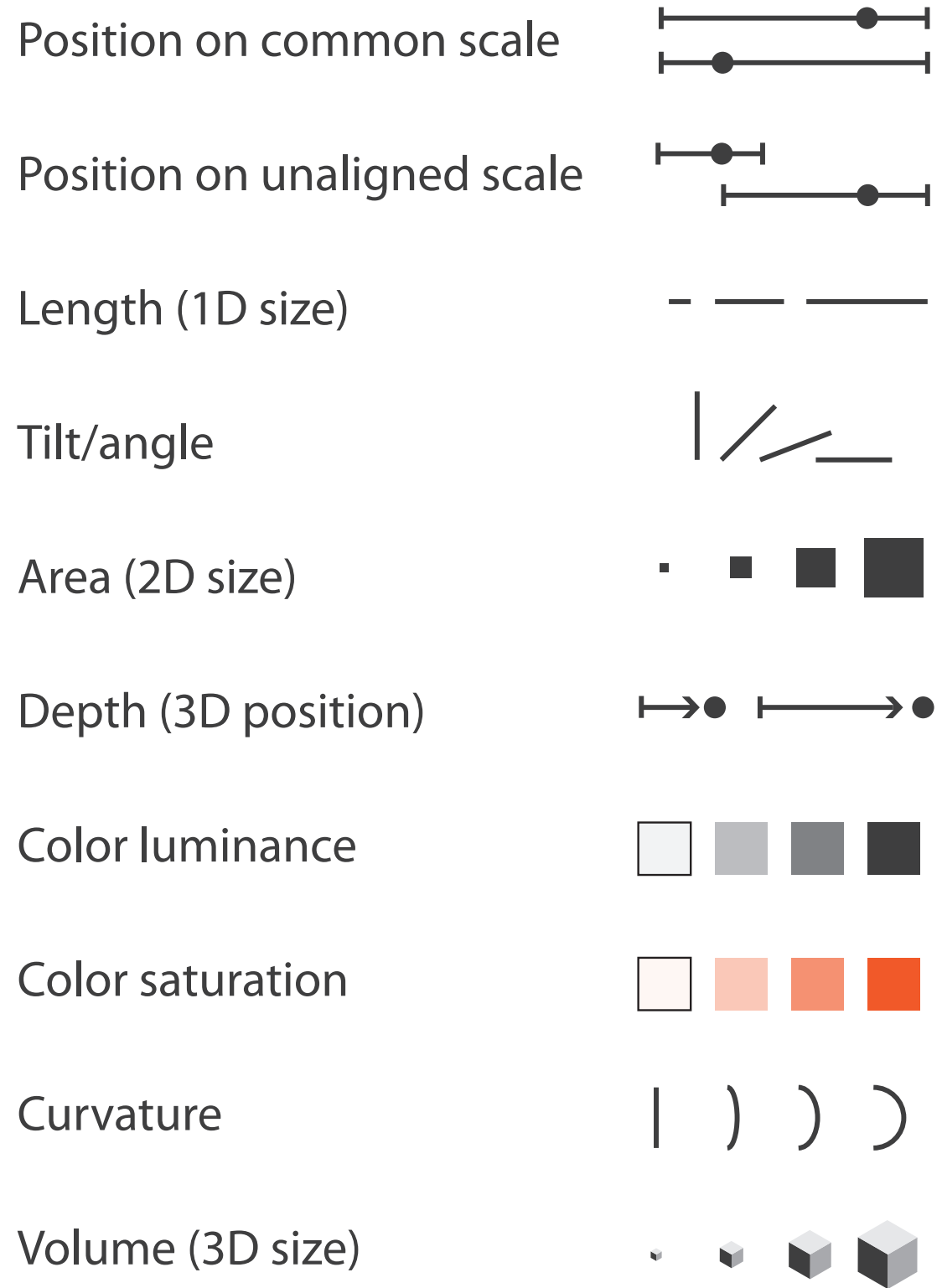Position on common scale

Position on unaligned scale

Length (1D size)

Tilt/angle

Area (2D size)

Depth (3D position)

Color luminance

Color saturation

Curvature

Volume (3D size)

Best ▲

Effectiveness

Least ▼

Same

Same

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*
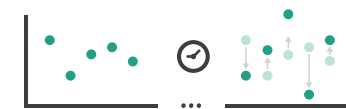
➤ Size, Angle, Curvature, ...

➤ Shape

➤ Motion
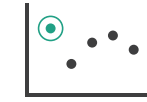*Direction, Rate, Frequency, ...*

Manipulate

➔ Change
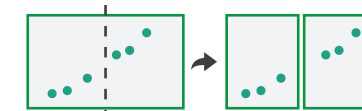
➔ Select

➔ Navigate

Facet

➔ Juxtapose

➔ Partition

➔ Superimpose

Reduce

➔ Filter

➔ Aggregate
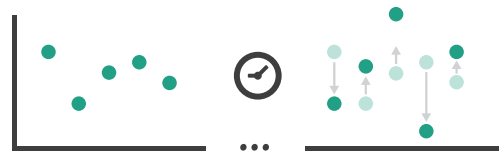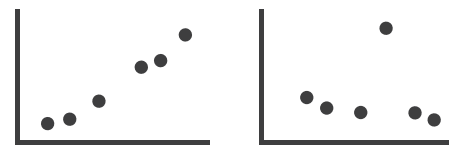
➔ Embed

What?

Why?

How?

How to handle complexity: 3 more strategies   + 1 previous
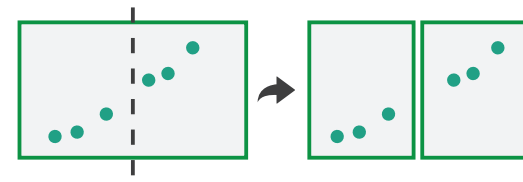

Manipulate

→ **Change**

→ **Select**

→ **Navigate**

Facet

→ **Juxtapose**

→ **Partition**

→ **Superimpose**
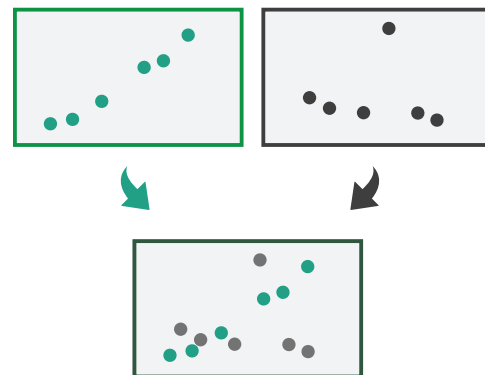
Reduce

→ **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

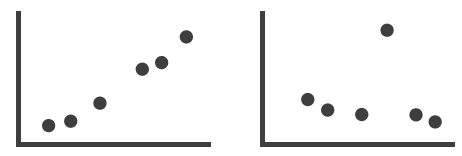# How to handle complexity: 3 more strategies   + 1 previous

Manipulate
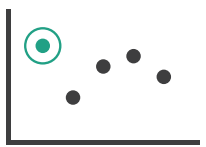- → **Change**
- → **Select**
- → **Navigate**

Facet
- → **Juxtapose**
- → **Partition**
- → **Superimpose**

Reduce
- → **Filter**
- → **Aggregate**
- → **Embed**

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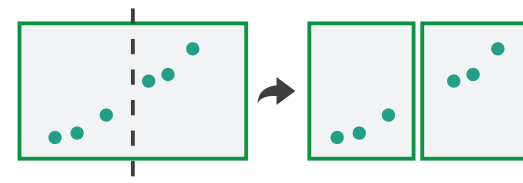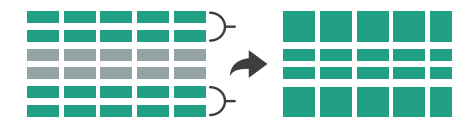
- change over time
  - most obvious & flexible of the 4 strategies

How to handle complexity: 3 more strategies   + 1 previous

Manipulate

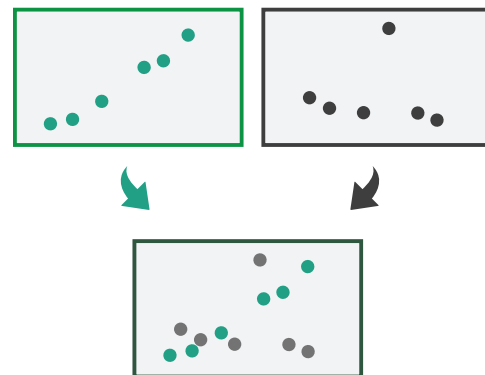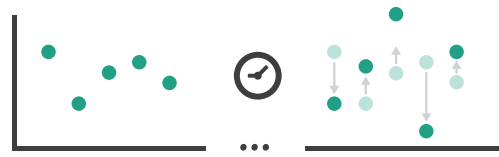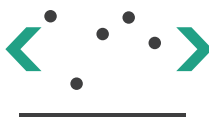→ **Change**

→ **Select**

→ **Navigate**

Facet

→ **Juxtapose**
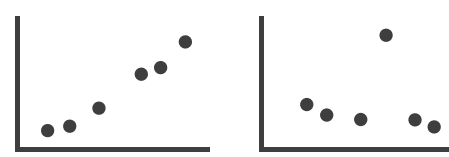
→ Partition
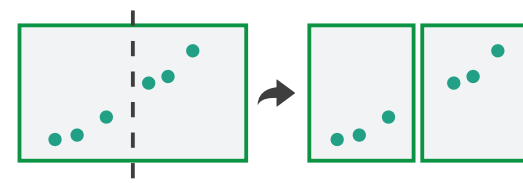
→ Superimpose
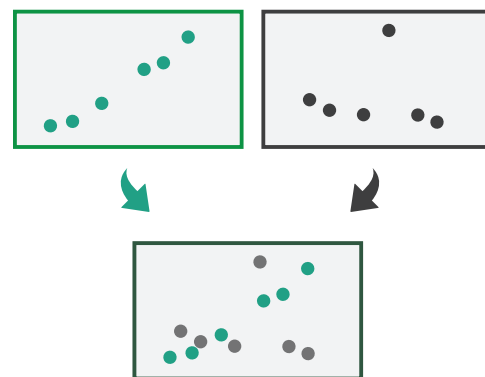
Reduce

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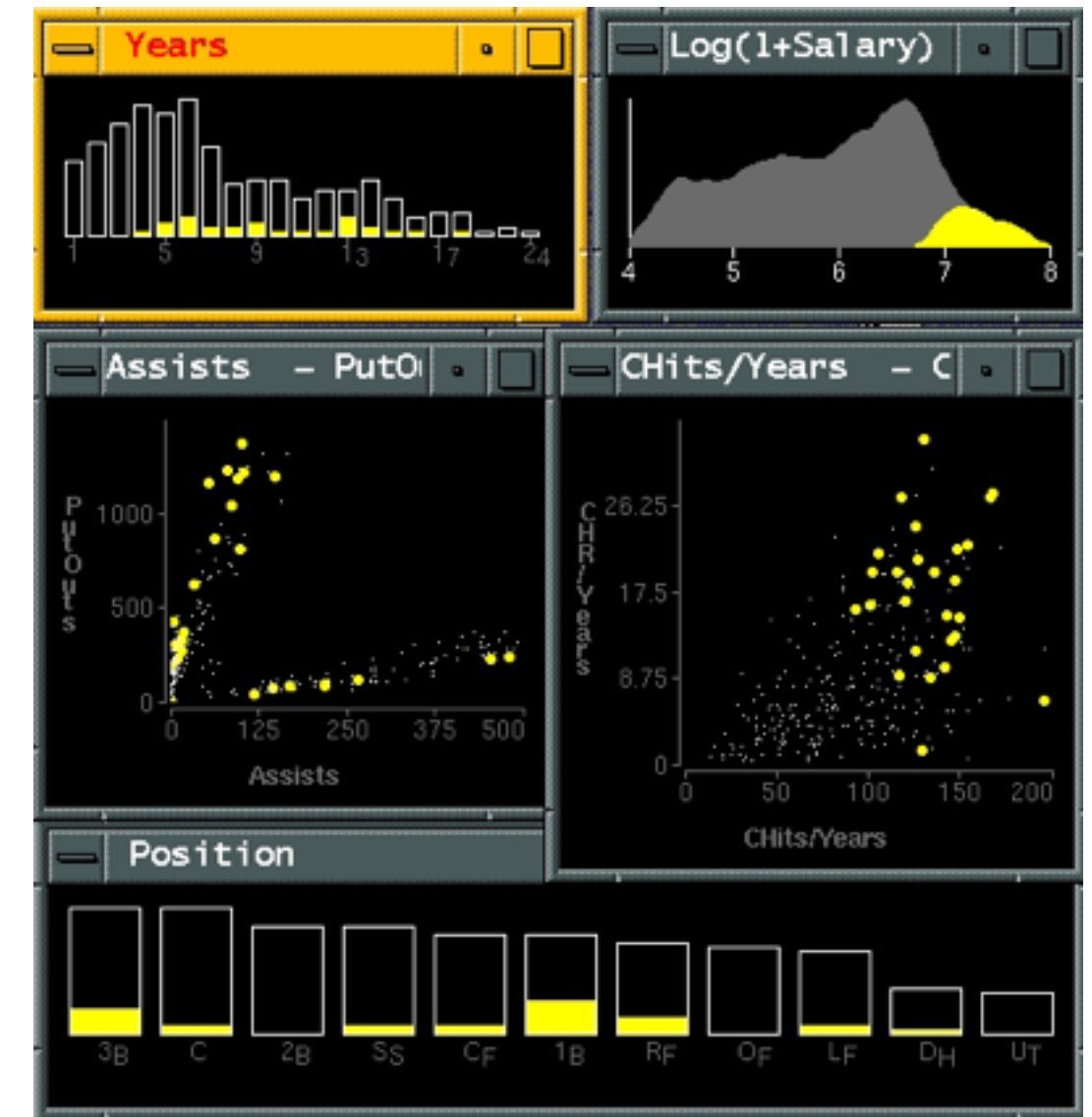
[Visual Exploration of Large Structured Datasets. Wills. Proc. New Techniques and Trends in Statistics (NTTS), pp. 237–246. IOS Press, 1995.]

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

- 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**
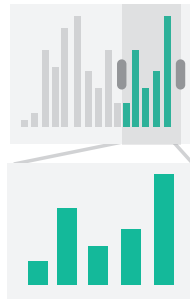
- encoding: same

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

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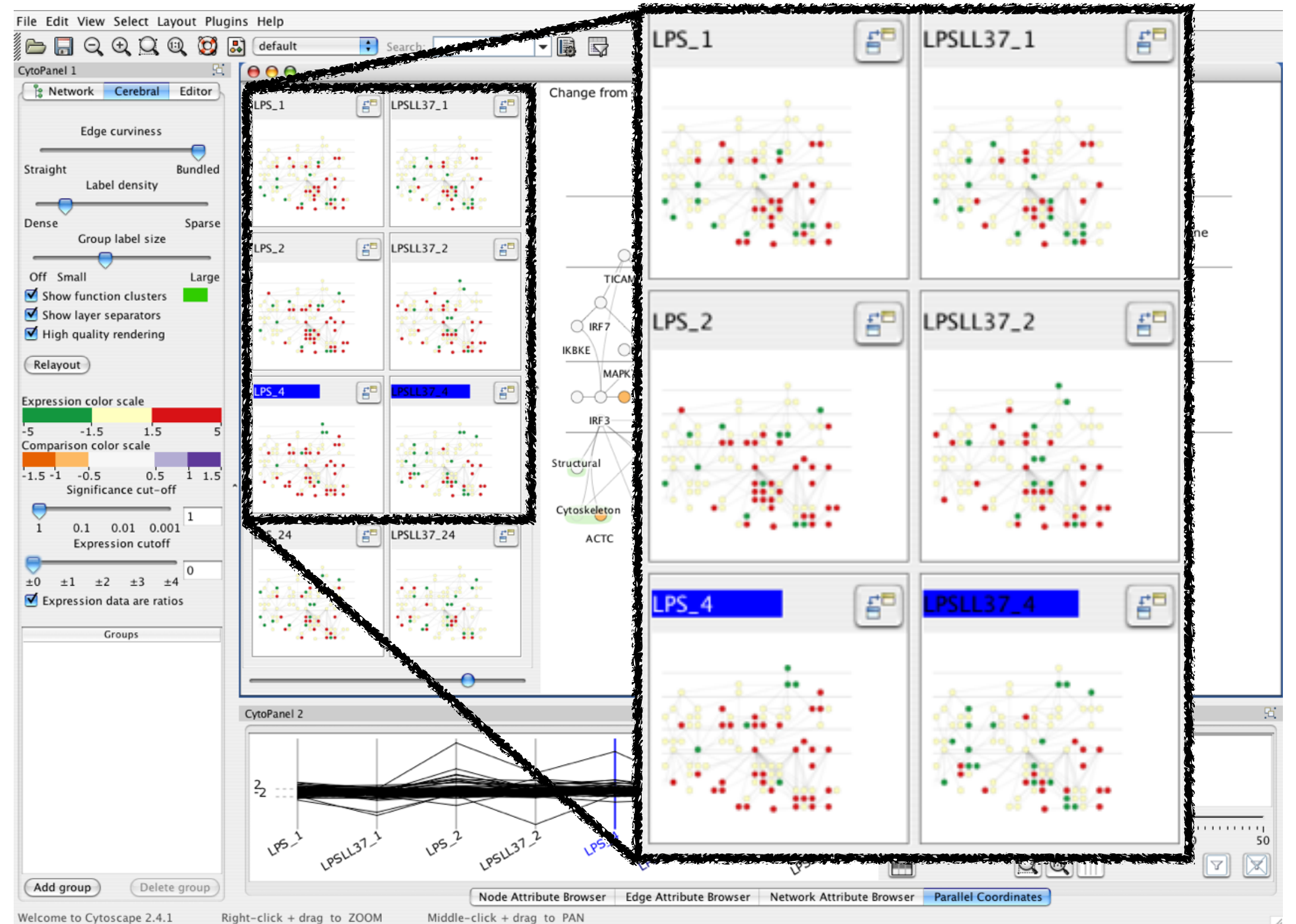
[Cerebral: Visualizing Multiple Experimental Conditions on a Graph with Biological Context. Barsky, Munzner, Gardy, and Kincaid. IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis 2008) 14:6 (2008), 1253–1260.]

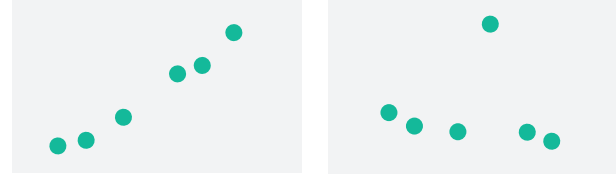# Coordinate views: Design choice interaction

		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

Idiom: Animation (change over time)

- weaknesses
  - widespread changes
  - disparate frames
- strengths
  - choreographed storytelling
  - localized differences between contiguous frames
  - animated transitions between states
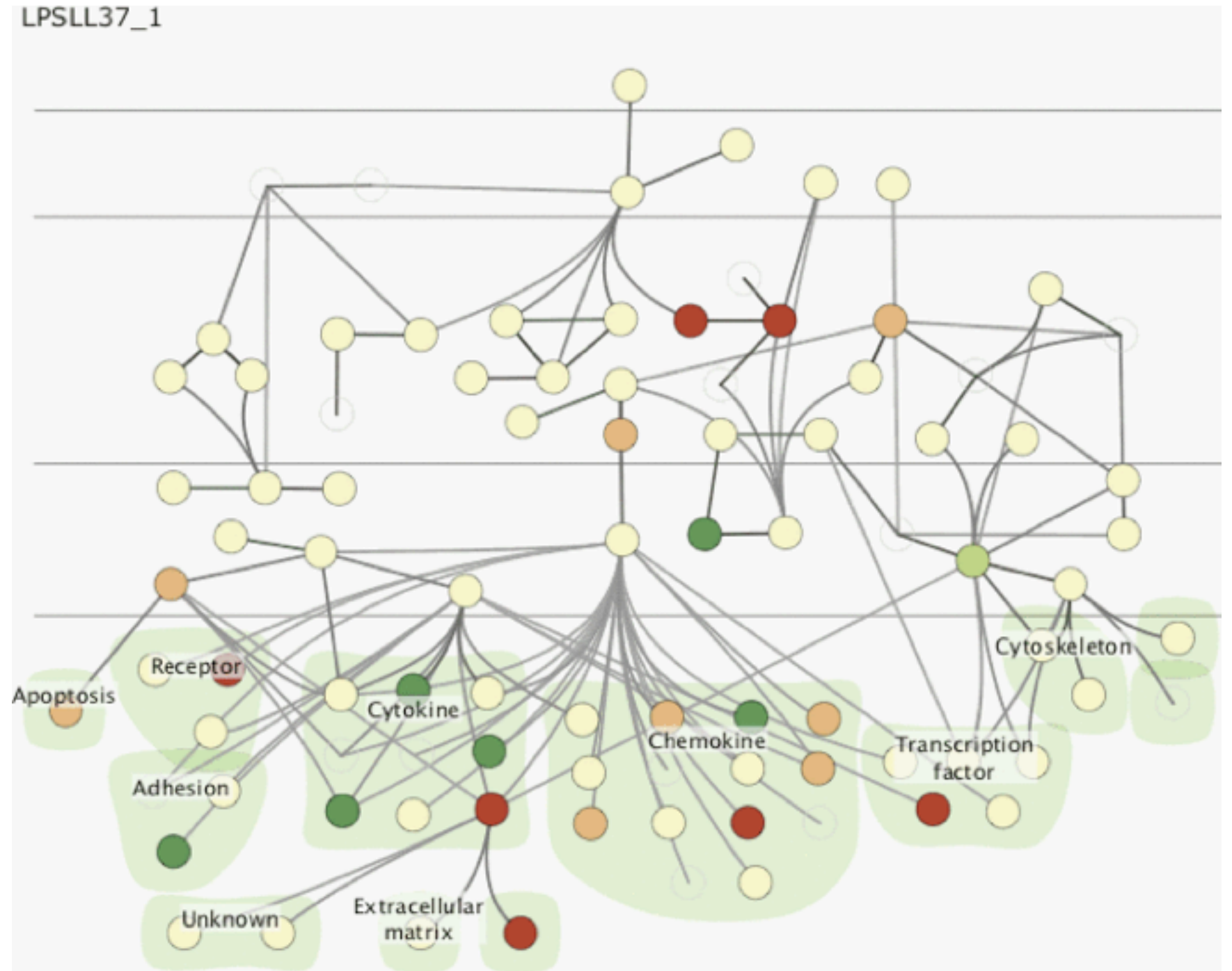
How to handle complexity: 3 more strategies   + 1 previous
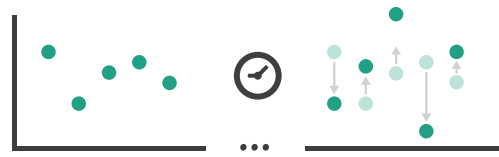
Manipulate
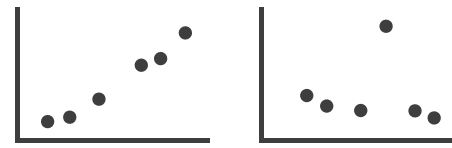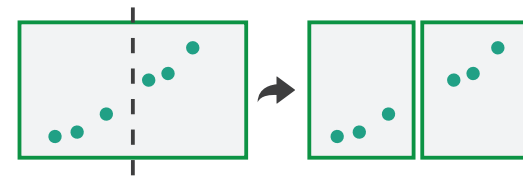- → **Change**
- → **Select**
- → **Navigate**

Facet
- → **Juxtapose**
- → **Partition**
- → **Superimpose**
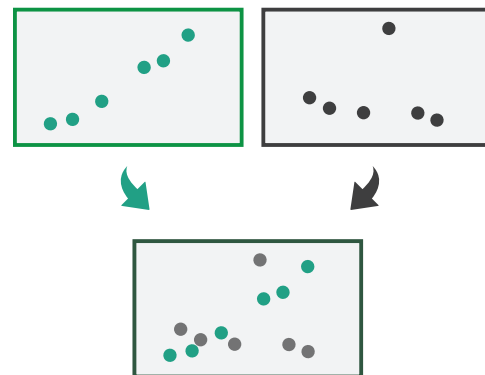
Reduce
- → **Filter**
- → **Aggregate**
- → **Embed**

→ *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

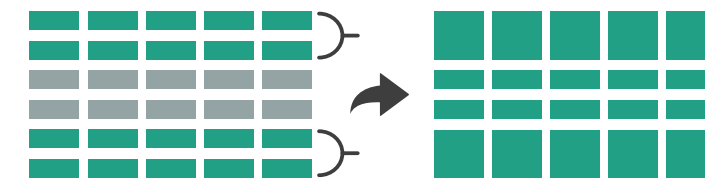**Reducing Items and Attributes**
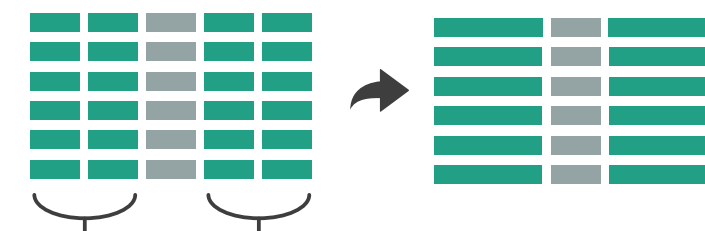
➔ **Filter**
  ➔ Items
  ➔ Attributes

➔ **Aggregate**
  ➔ Items
  ➔ Attributes

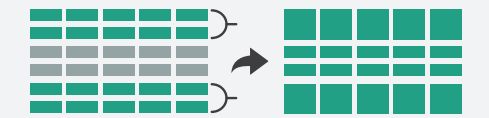**Reduce**

➔ Filter

➔ Aggregate

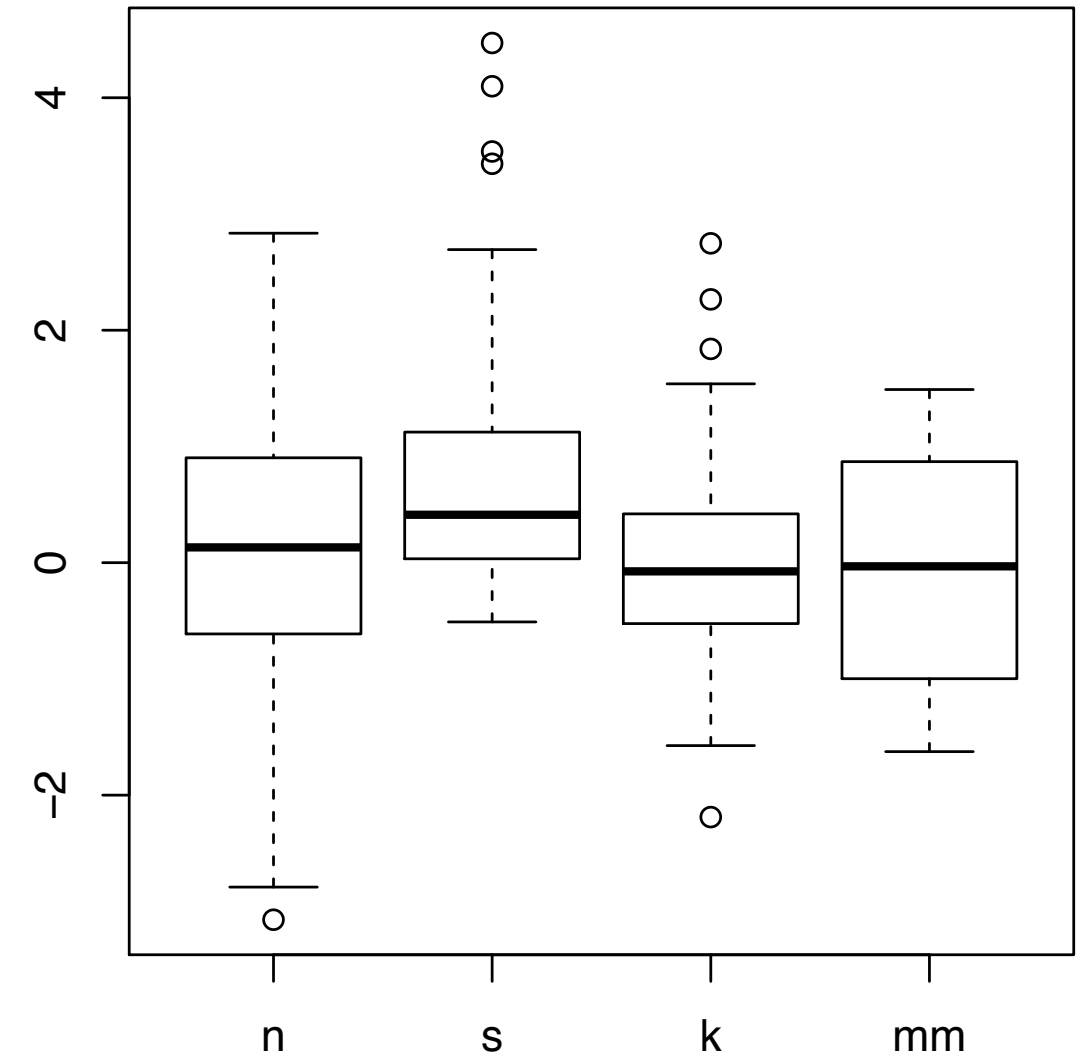➔ Embed

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
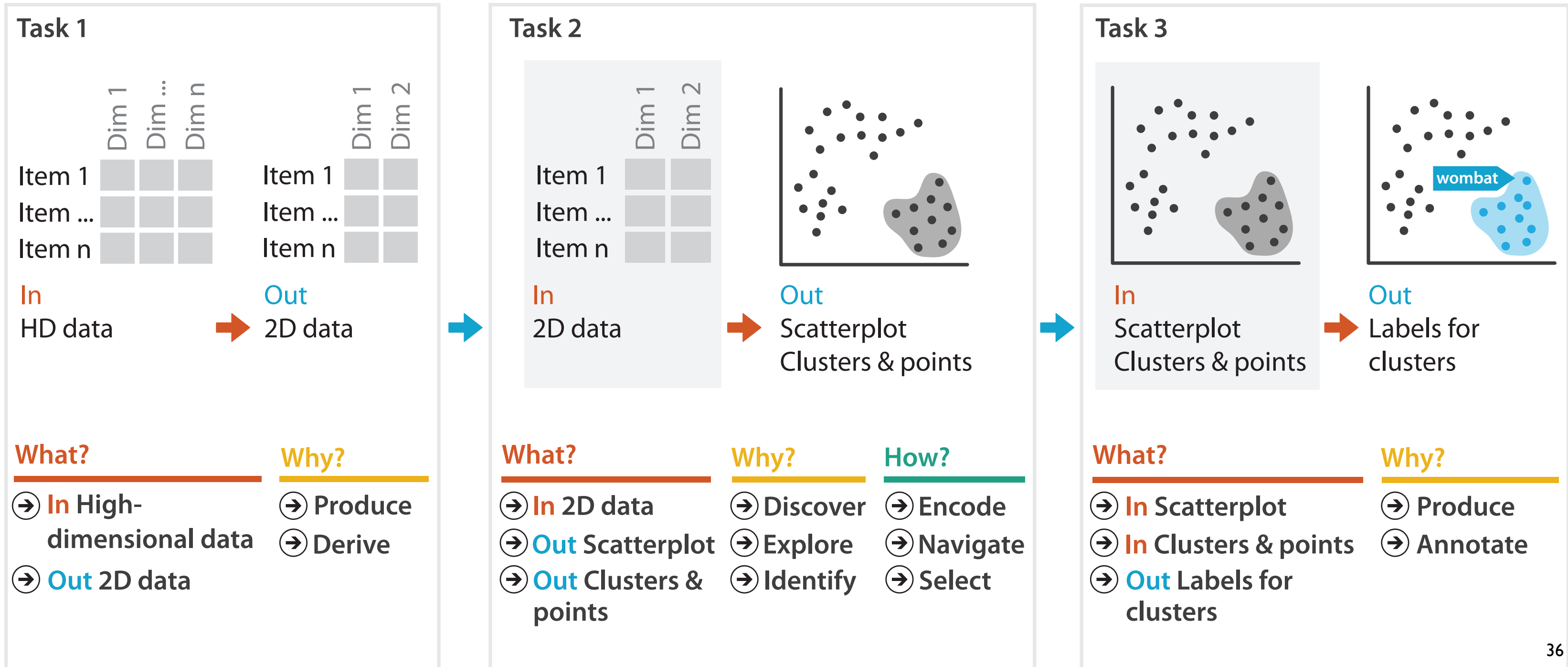 - outliers beyond fence cutoffs explicitly shown

[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

Task 1

In
HD data

Out
2D data

What?
- → **In** High-dimensional data
- → **Out** 2D data

Why?
- → Produce
- → Derive

Task 2

In
2D data

Out
Scatterplot
Clusters & points

What?
- → **In** 2D data
- → **Out** Scatterplot
- → **Out** Clusters & points

Why?
- → Discover
- → Explore
- → Identify

How?
- → Encode
- → Navigate
- → Select

Task 3

wombat

In
Scatterplot
Clusters & points

Out
Labels for clusters

What?
- → **In** Scatterplot
- → **In** Clusters & points
- → **Out** Labels for clusters

Why?
- → Produce
- → Annotate

What?

Datasets | Attributes

→ Data Types
→ Items

→ Data and D...
Tables
Items
Attributes

→ Dataset Typ...
→ Tables

→ Multidim...

→ Geometr...

Why?

🖋 **Actions** 🎯 **Targets**

→ **Analyze** → **All Data**

→ Consume
→ *Discover* → *Present* → *Enjoy*

→ Trends → Outliers → Features

→ **Produce**
→ *Annotate*
tag

How?
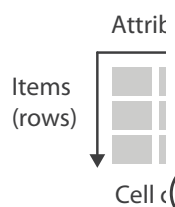
→ **Search**

Location known
Location unknown
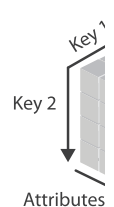
→ **Query**
→ Identify

Encode

→ **Arrange**
→ Express → Separate
→ Order → Align
→ Use

→ **Map**
from categorical and ordered attributes
→ Color
→ *Hue* → *Saturation* → *Luminance*

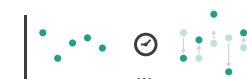→ Size, Angle, Curvature, ...

→ Shape
+ ● ■ ▲

→ Motion
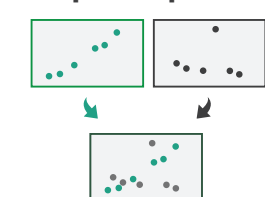*Direction, Rate, Frequency, ...*

Manipulate
→ **Change**
→ **Select**
→ **Navigate**

Facet
→ **Juxtapose**
→ **Partition**
→ **Superimpose**

Reduce
→ **Filter**
→ **Aggregate**
→ **Embed**

domain
abstraction
idiom
algorithm

What?
Why?
How?

37

A quick taste of my own work!

technique-driven work ⟷ problem-driven work
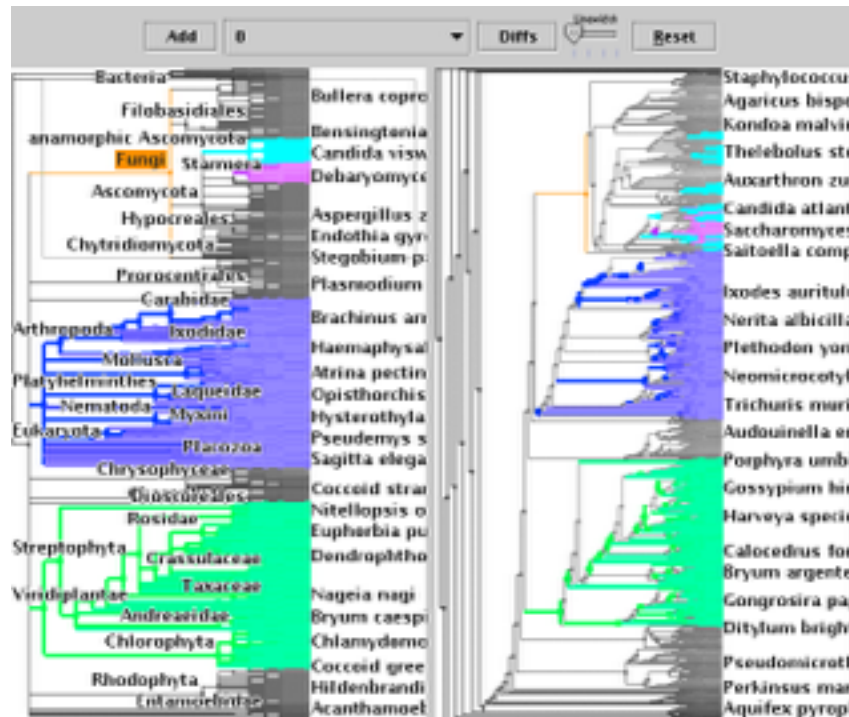
theoretical foundations
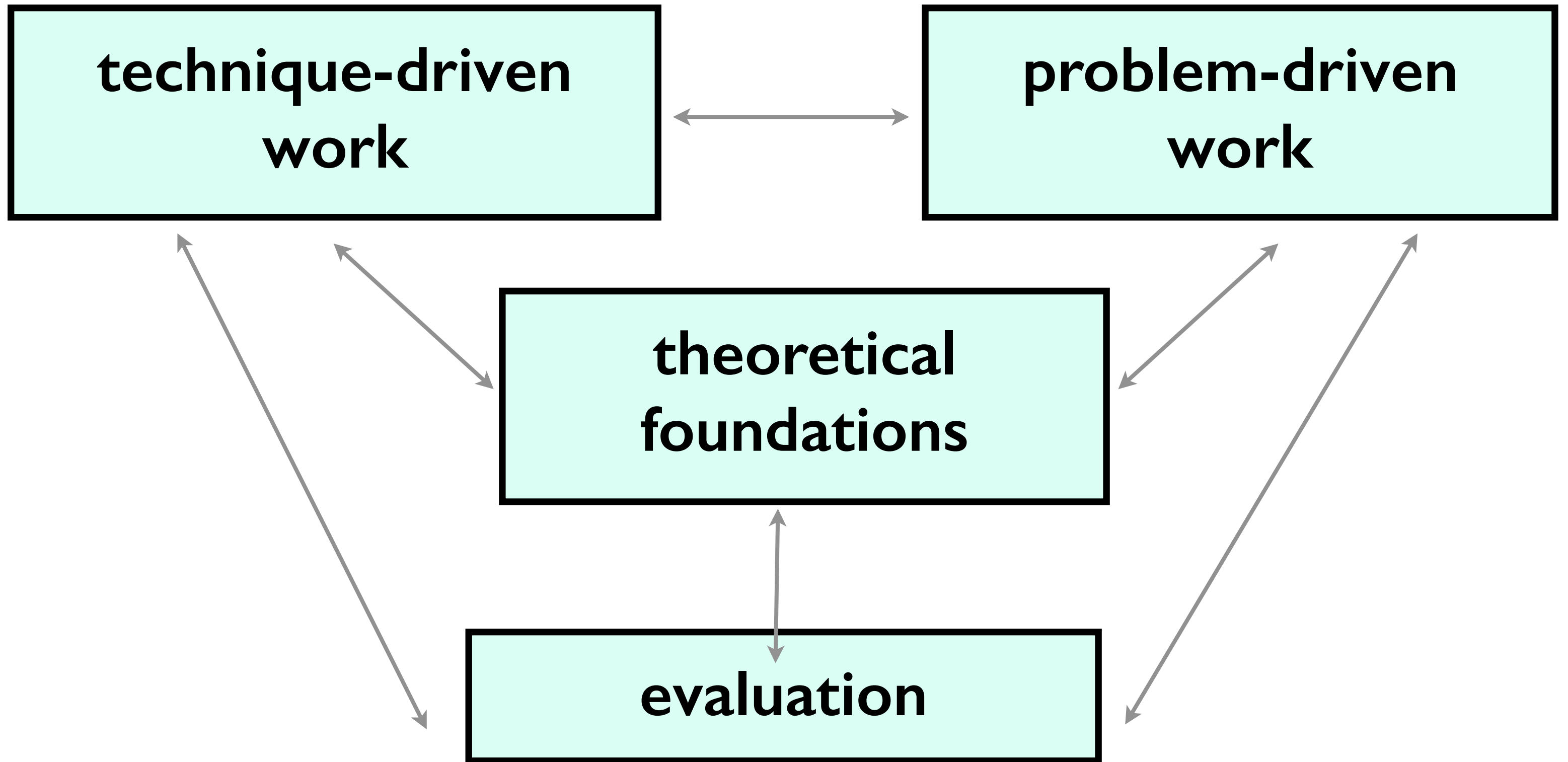
evaluation

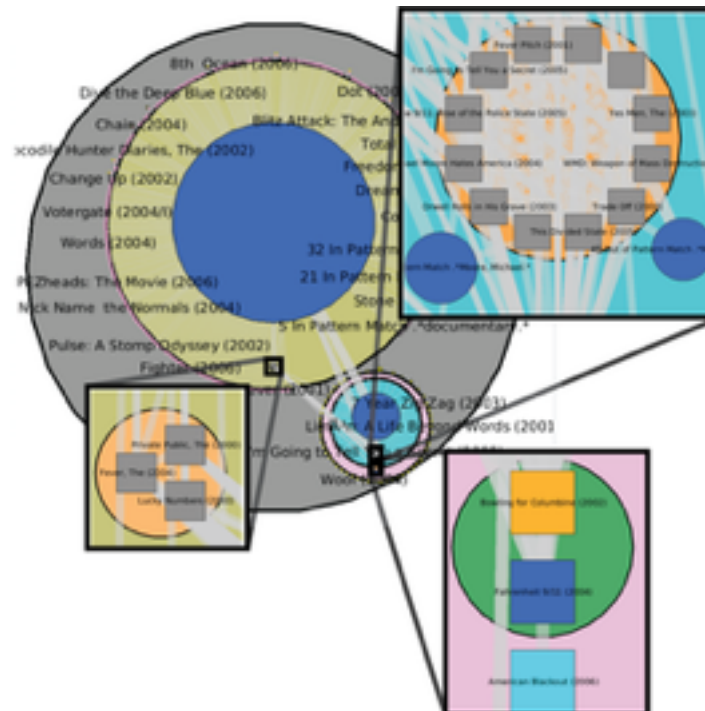# Technique-driven: Graph drawing

TreeJuxtaposer

James Slack

Kristian Hildebrand

Daniel Archambault

David Auber
(Bordeaux)

**TopoLayout
SPF
Grouse
GrouseFlocks
TugGraph**

Evaluation: Graph drawing

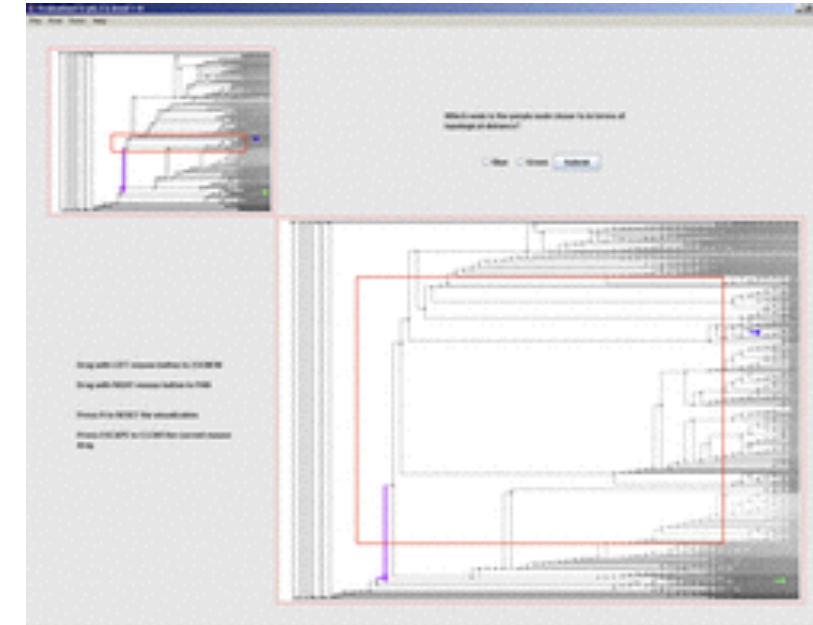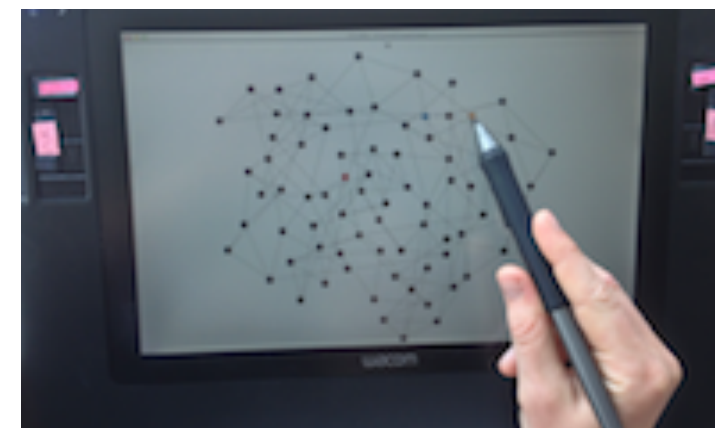Dmitry Nekrasovski   Adam Bodnar   Joanna McGrenere
(UBC)

Stretch and squish navigation

Jessica Dawson   Joanna McGrenere
(UBC)

Search set model of path tracing
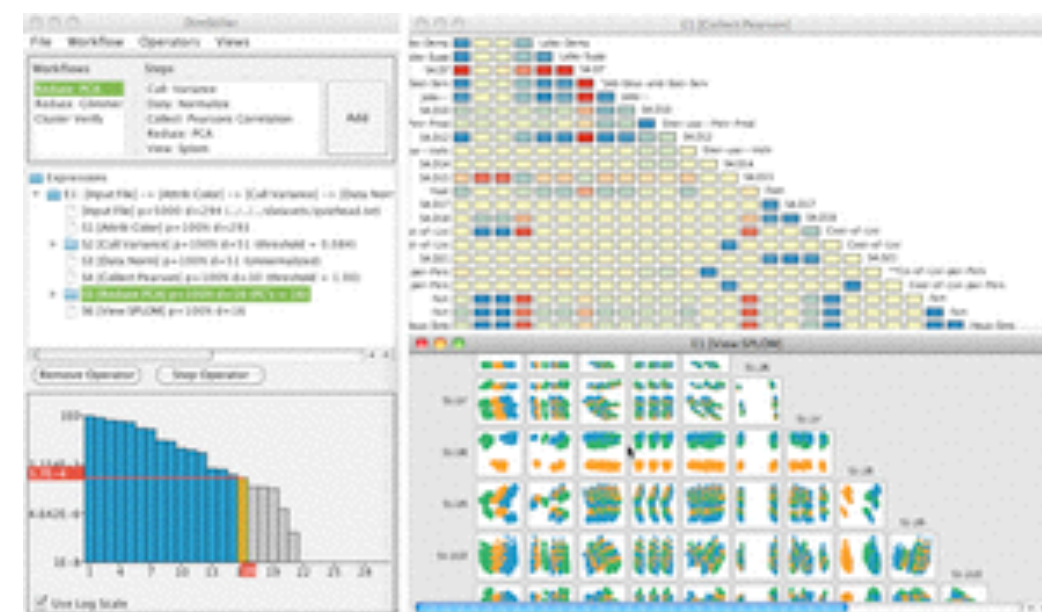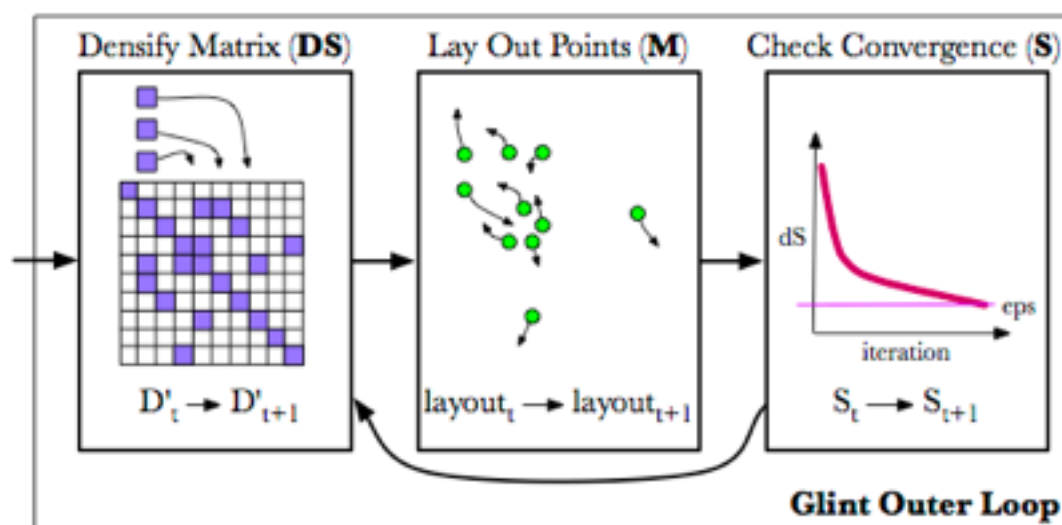
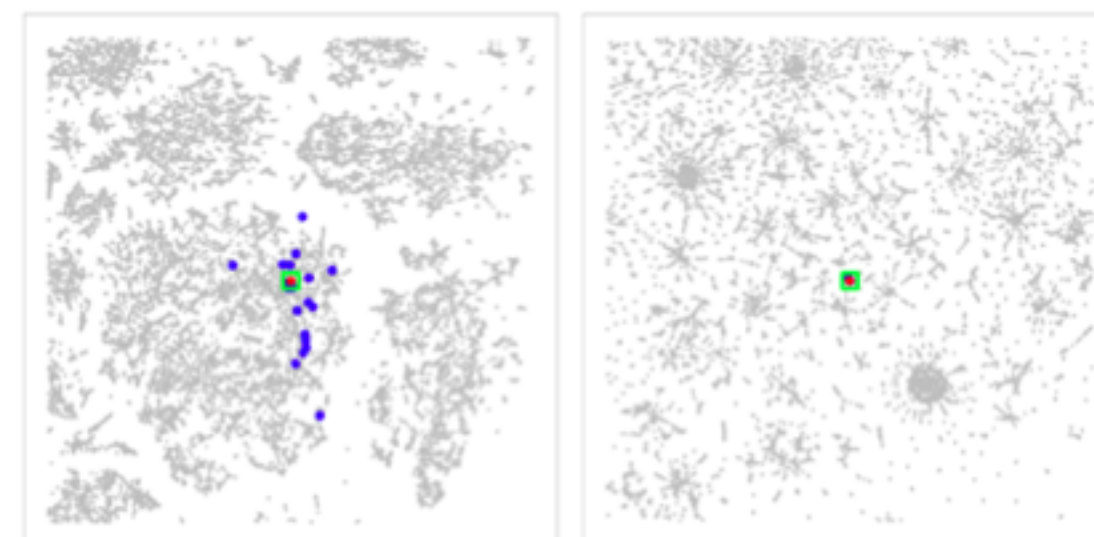# Technique-driven: Dimensionality reduction

Stephen Ingram

Glimmer

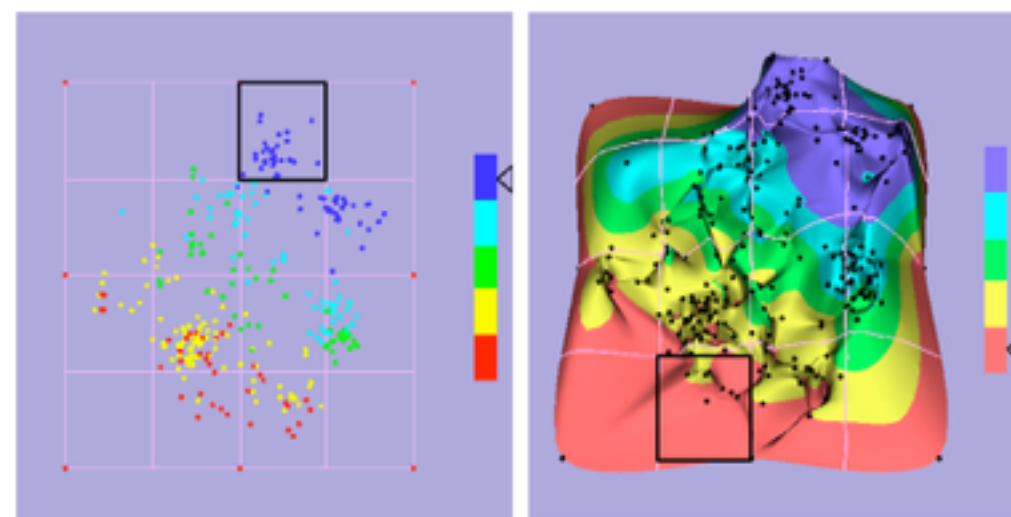**DimStiller**

Glint

QSNE

41

Evaluation: Dimensionality reduction

Melanie Tory

Points vs landscapes for dimensionally reduced data

Guidance on DR & scatterplot choices

Michael Sedlmair

Melanie Tory (UVic)

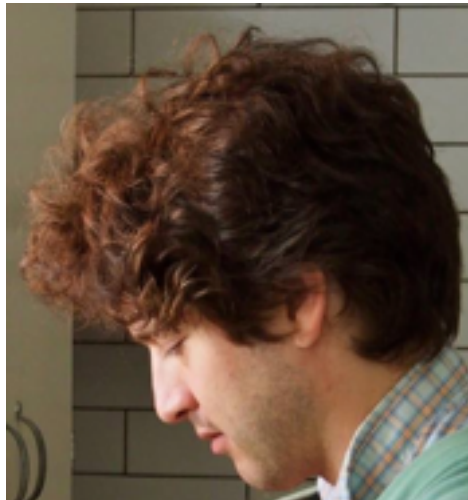**Taxonomy of cluster separation factors**
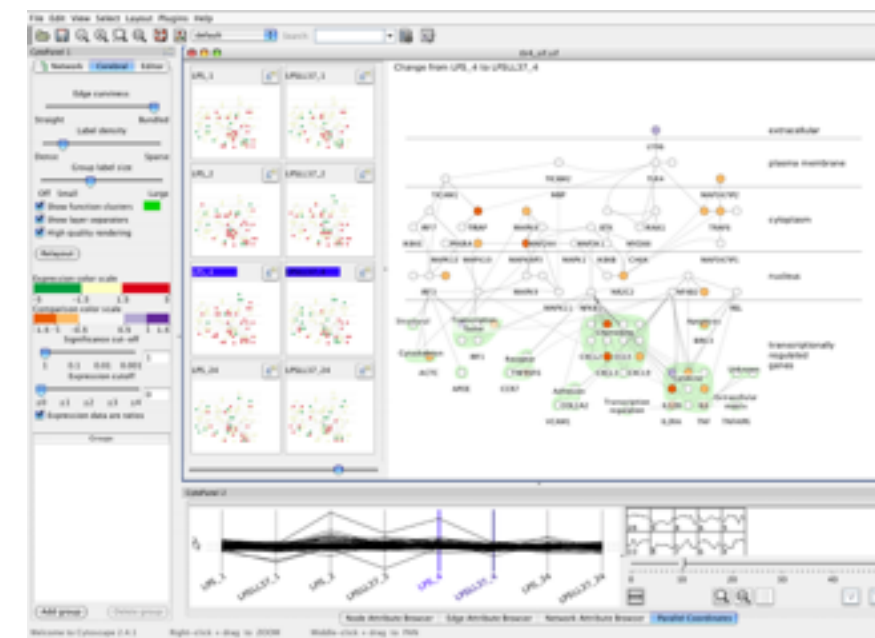
42

Problem-driven: Genomics

Aaron Barsky
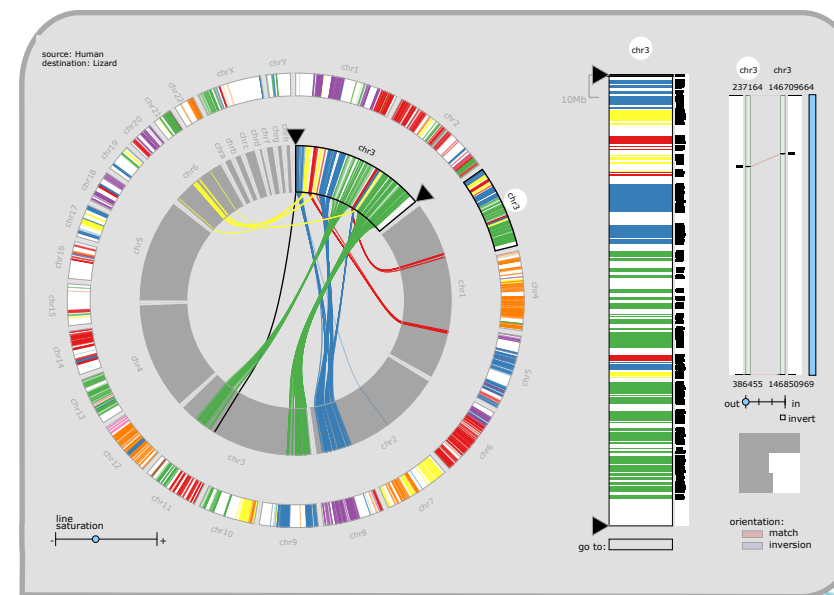
Jenn Gardy (Microbio)

Robert Kincaid (Agilent)

Cerebral

Miriah Meyer
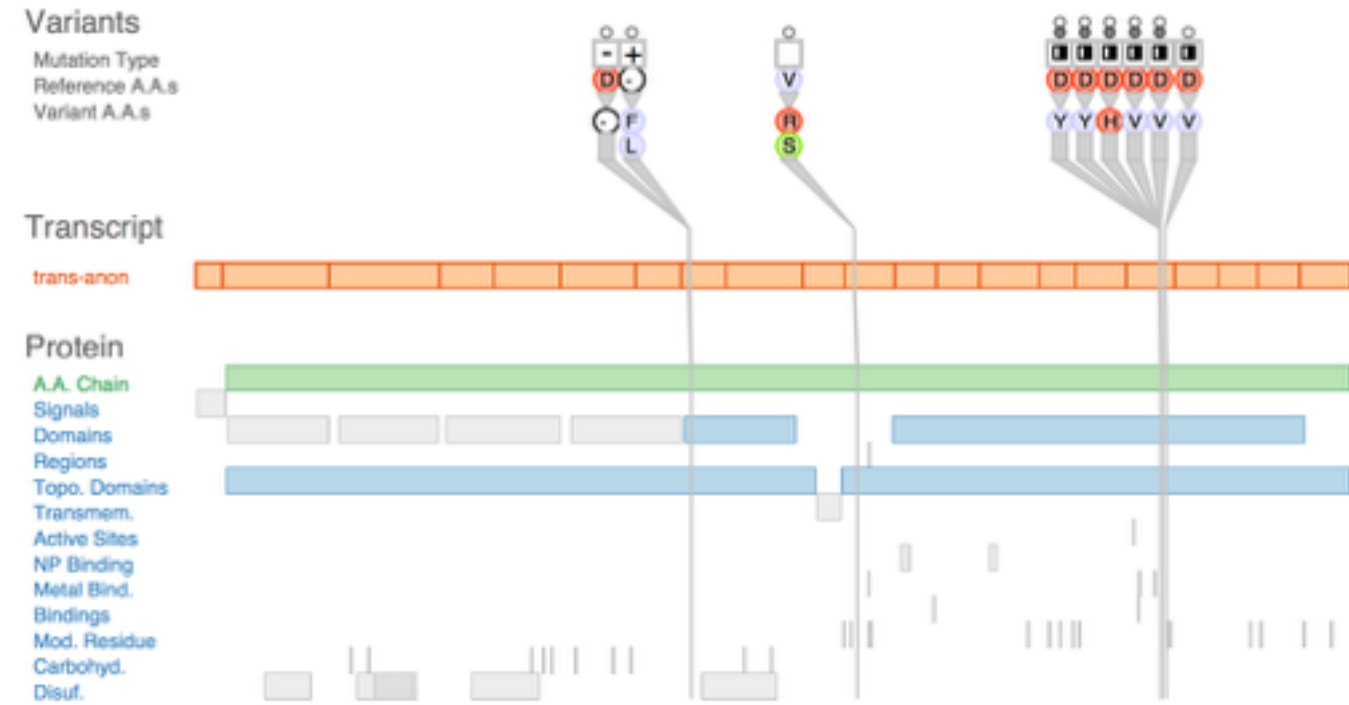
Hanspeter Pfister (Harvard)

MizBee

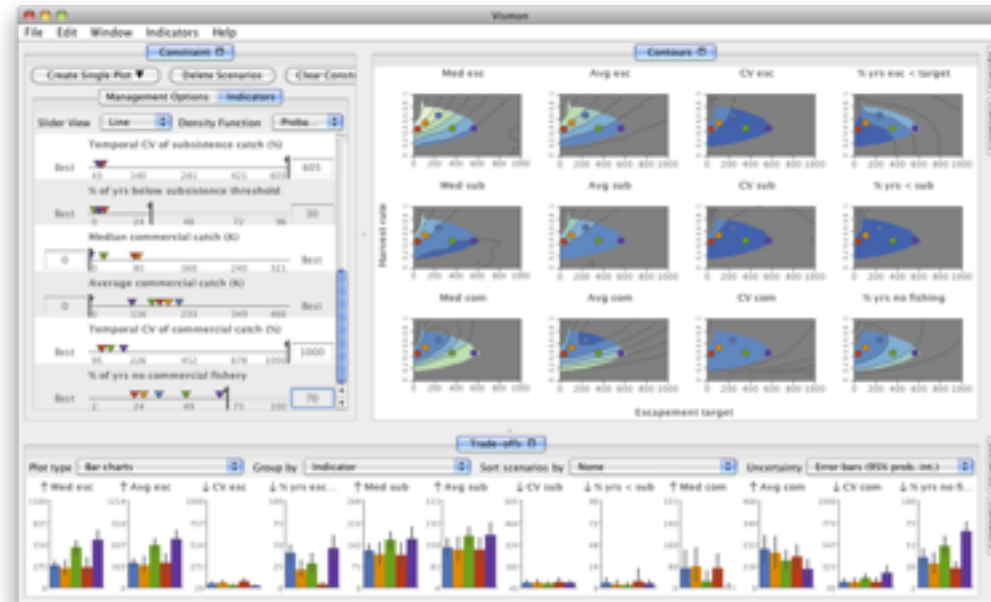MulteeSum, Pathline

43

Problem-driven: Genomics, fisheries

Joel Ferstay

Cydney Nielsen
(BC Cancer)

Variant View
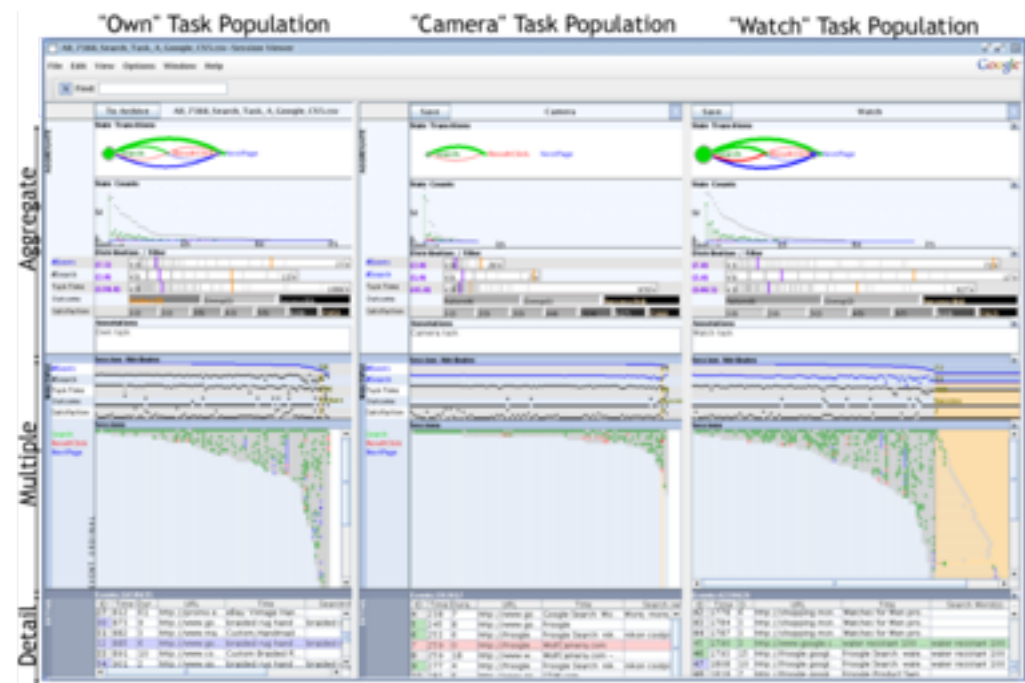

Vismon

Maryam Booshehrian

Torsten Moeller
(SFU)

44

Problem-driven: Many domains

SessionViewer: web log analysis

Heidi Lam
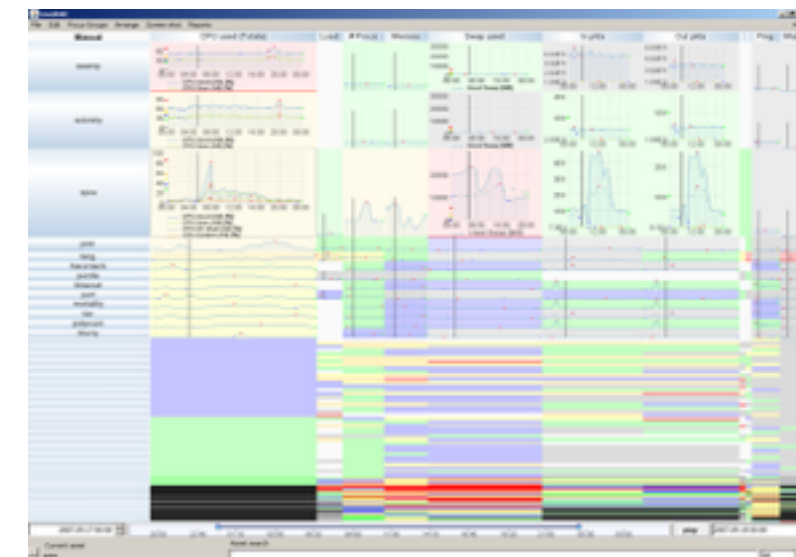
Diane Tang
(Google)

Peter McLachlan

Stephen North
(AT&T Research)

LiveRAC: systems time-series

Evaluation: Focus+Context

Heidi Lam
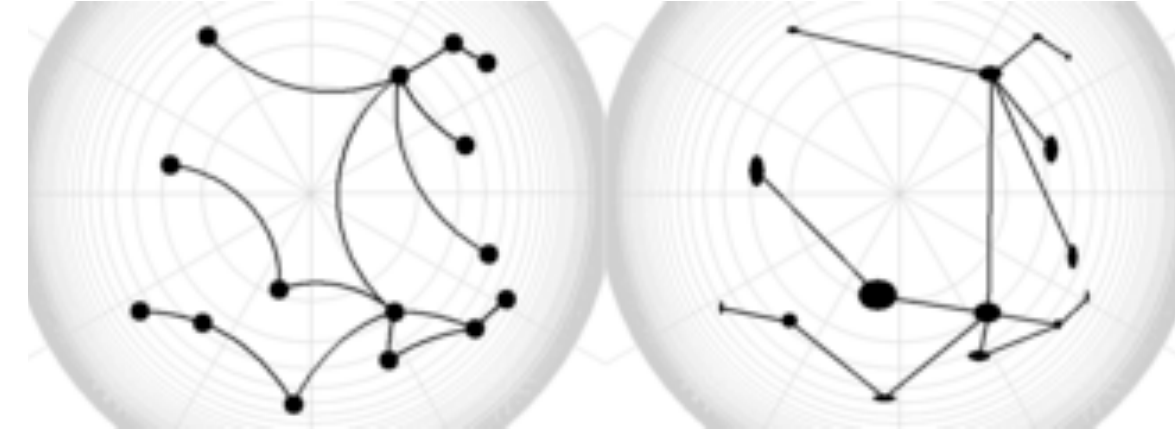
Ron Rensink
(UBC)

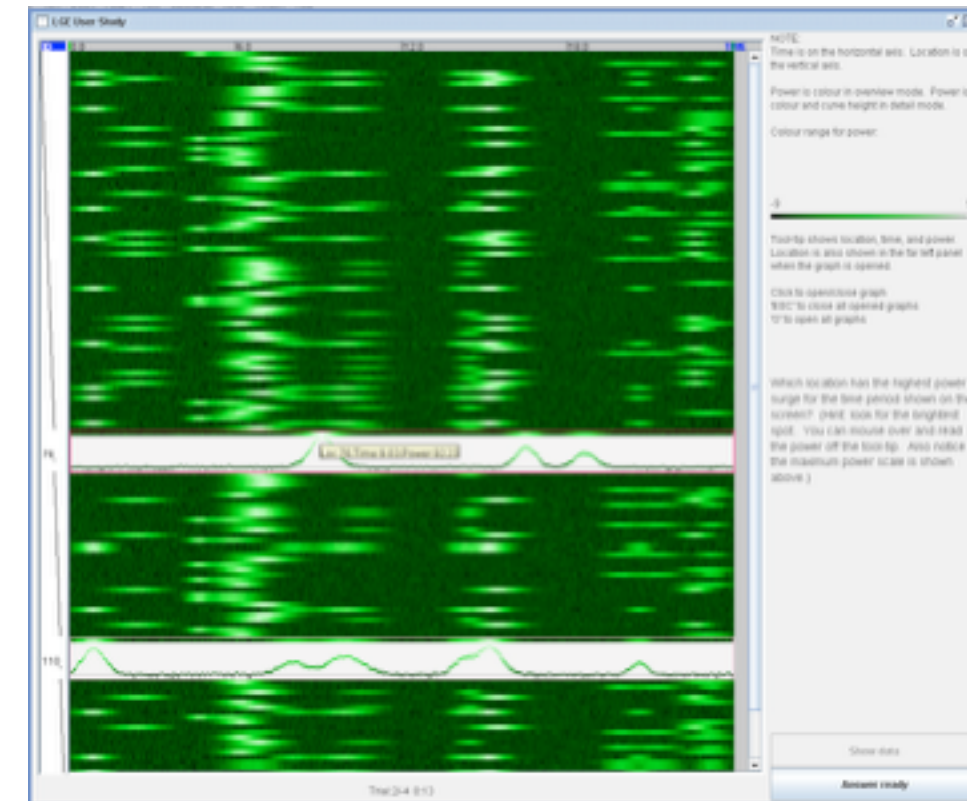**Distortion impact on search/memory**

Heidi Lam

Robert Kincaid
(Agilent)

Separate vs integrated views

46

Journalism

Matt Brehmer

Stephen Ingram

Jonathan Stray
(Assoc Press)

Overview

Johanna Fulda
(Sud. Zeitung)

Matt Brehmer

TimeLineCurator

Theoretical foundations
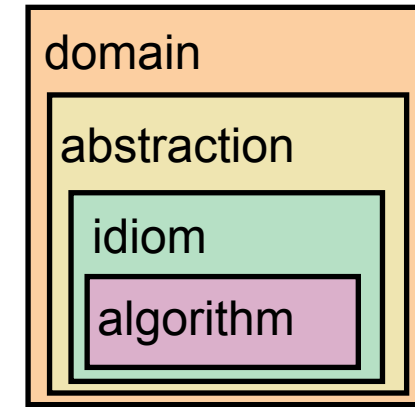
- Visual Encoding Pitfalls
  - Unjustified Visual Encoding
  - Hammer In Search Of Nail
  - 2D Good, 3D Better
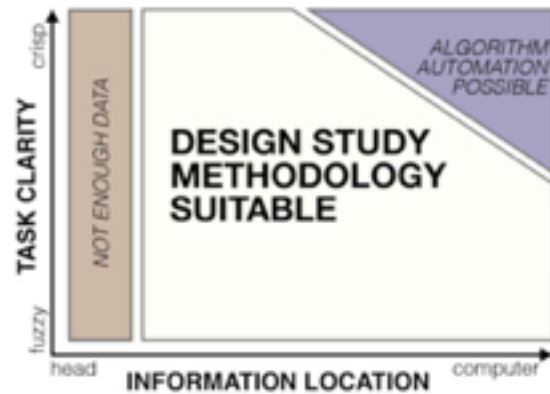 - Color Cacophony
  - Rainbows Just Like In The Sky
- Strategy Pitfalls
  - What I Did Over My Summer
  - Least Publishable Unit
  - Dense As Plutonium
  - Bad Slice and Dice

Papers Process & Pitfalls

Nested Model

Design Study Methodology

Abstract Tasks
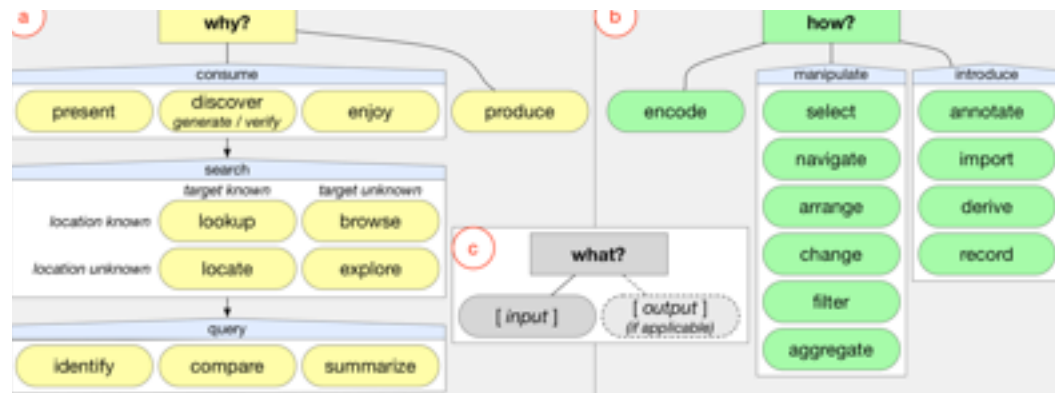
Michael Sedlmair

Miriah Meyer

Matt Brehmer
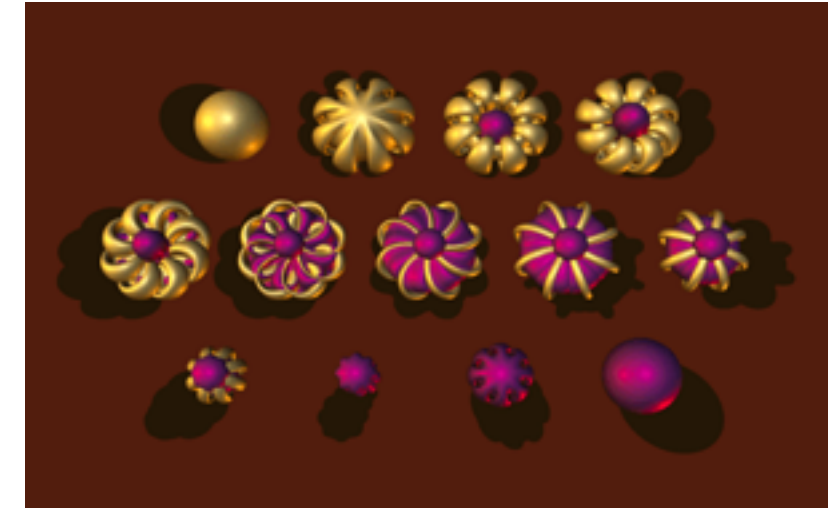
Geometry Center 1990-1995

Geomview

The Shape of Space

Outside In

Charlie Gunn   Stuart Levy   Mark Phillips   Delle Maxwell

More Information

- this talk
  http://www.cs.ubc.ca/~tmm/talks.html#vad16nasa

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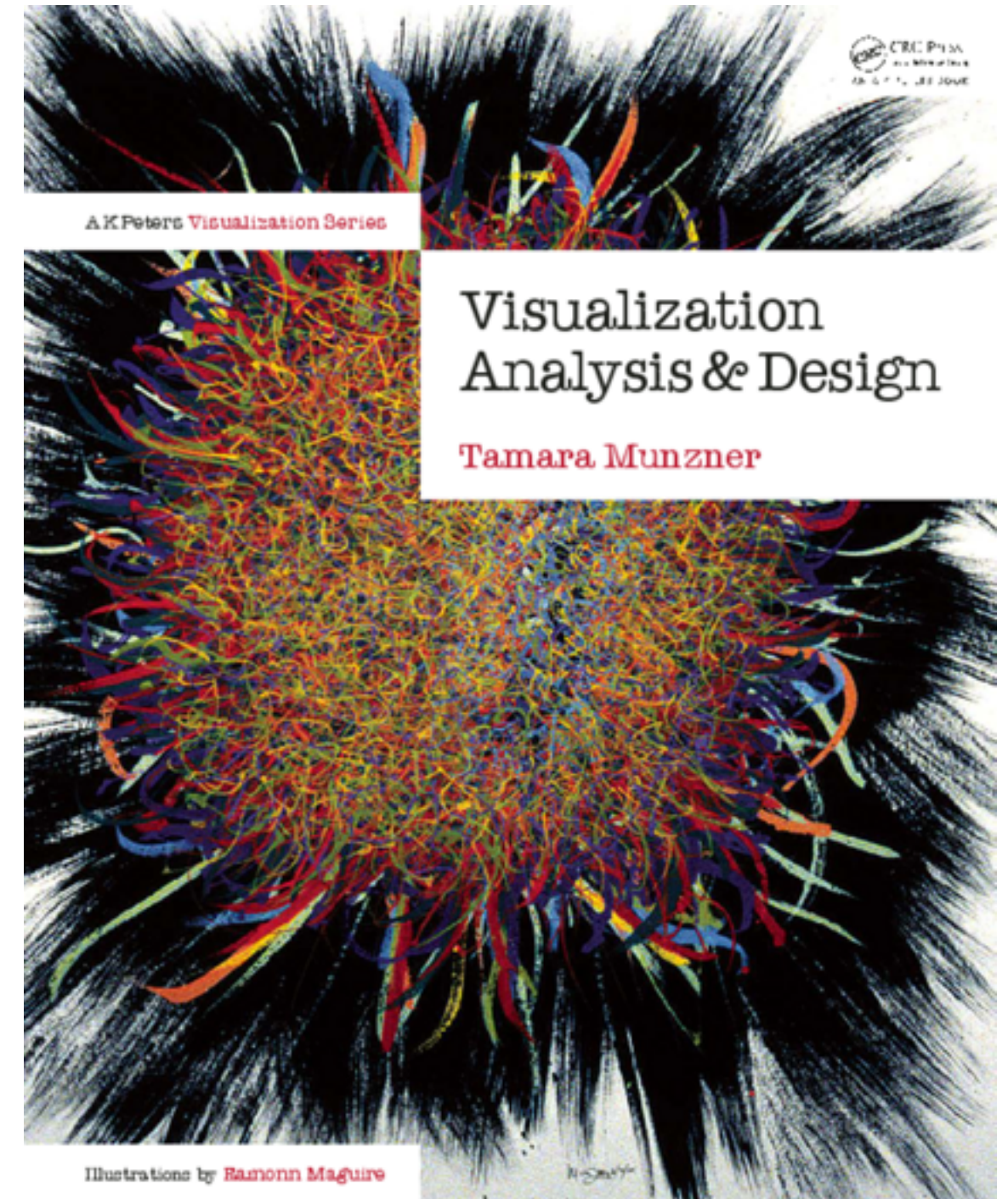
Visualization Analysis and Design.
Munzner. A K Peters Visualization Series, CRC Press, Visualization Series, 2014.

50