Visualization Analysis & Design for Biology
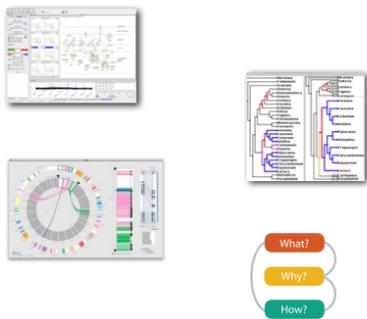
Tamara Munzner
Department of Computer Science
University of British Columbia

Symposium on Biological Data Visualization (BioVis) Keynote
in conjunction with ISMB 2014
11 July 2014, Boston MA

http://www.cs.ubc.ca/~tmm/talks.html#biovis14

Outline

• introduction

• Cerebral

• TreeJuxtaposer

• MizBee

• summary and conclusions

What?
Why?
How?

Defining visualization (vis)

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

Why?...
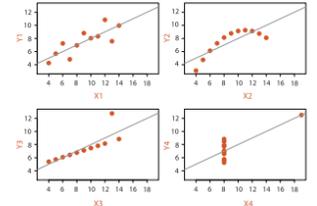

Why have a human in the loop?

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

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

• many analysis problems ill-specified, not clear what questions to ask in advance
  – don't need vis when fully automatic solution exists and is trusted
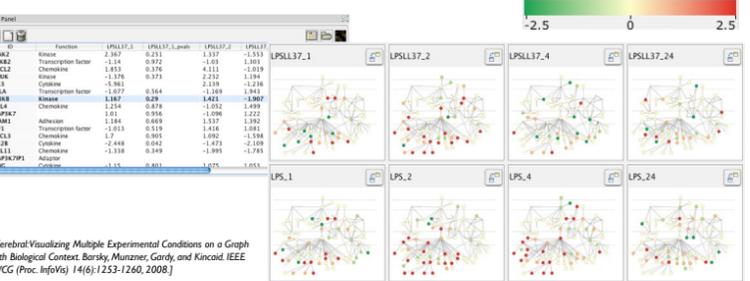
Anscombe's Quartet

Identical statistics	
x mean	9
x variance	10
y mean	8
y variance	4
x/y correlation	1

Why use an external representation?

Computer-based visualization systems provide visual representations of datasets designed to help people carry out tasks more effectively.
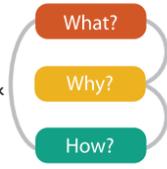
• external representation: replace cognition with perception

Expression color scale
-2.5   0   2.5

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


Why analyze?

Vis usage can be analyzed in terms of what data is shown, why the user needs it, and how the idiom is designed.

• abstractions
  – translate from specifics of domain to vocabulary of vis
    • *data abstraction*: **what** to show
      – might not draw what you're given: transform data into form useful for task
    • *task abstraction*: **why** they're looking at it
• idioms
  – *visual encoding idiom*: **how** to draw
  – *interaction idiom*: **how** to manipulate
• analysis framework: scaffold to think systematically about design space
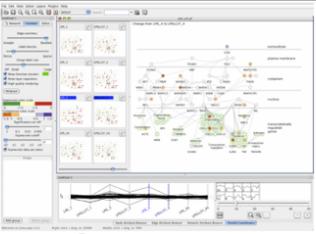 • huge, and most possibilities ineffective for particular task/data combination

What?
Why?
How?

Outline

• introduction

• Cerebral

• TreeJuxtaposer

• MizBee

• summary and conclusions

What?
Why?
How?
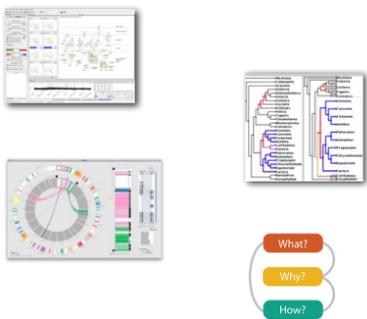
Cerebral
Visualizing Multiple Experimental Conditions on a Graph with Biological Context

joint work with:
Aaron Barsky, Jennifer Gardy, Robert Kincaid
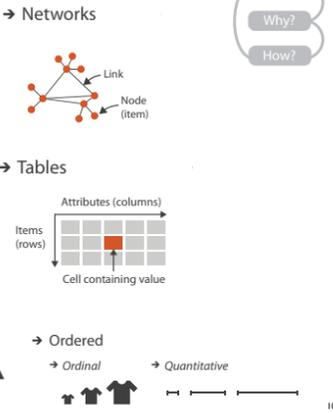
http://www.pathogenomics.ca/cerebral/

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

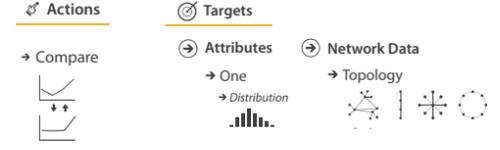## Cerebral video

---
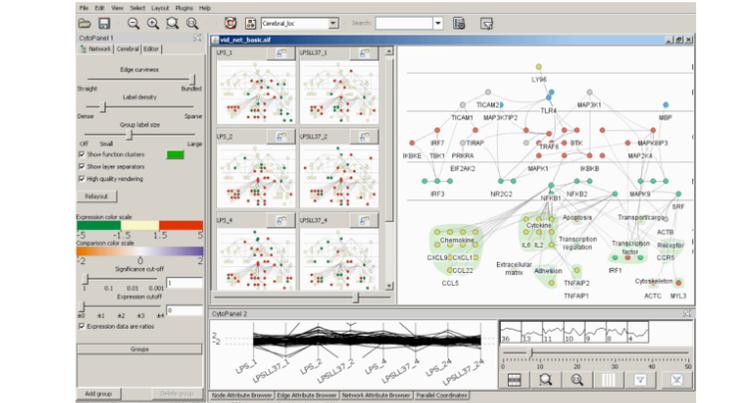
What: Data abstraction

• dataset types
  – network
    • nodes: genes
    • links: known interactions between genes
  – table
    • quantitative attributes
      – gene expression levels for nodes across different experimental conditions
    • categorical attributes
      – subcellular location of interaction
      – functional groups

→ Networks
→ Link
→ Node (item)

→ Tables
Attributes (columns)
Items (rows)
Cell containing value

→ Attribute Types
→ Categorical
→ Ordered
  → Ordinal
  → Quantitative

What?
Why?
How?

Why: Task abstraction

• task: interpret microarray experiment results with respect to gene network
  – goal: accelerate existing discovery workflow
  – compare distributions between attributes
    • experimental conditions
 – interpret attributes in context of current network topological structure

→ Discover

♂ Actions
→ Compare

◎ Targets
→ Attributes
  → One
    → Distribution
→ Network Data
  → Topology

What?
Why?
How?

How: Idiom design decisions

• arrange space for networks
  – custom node-link diagram layout
    • points for nodes
    • connection marks for links
  – vertical compartment according to subcellular location attribute
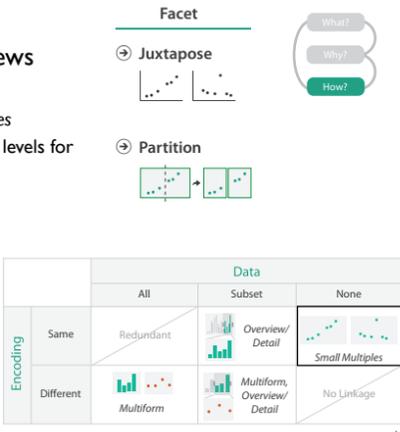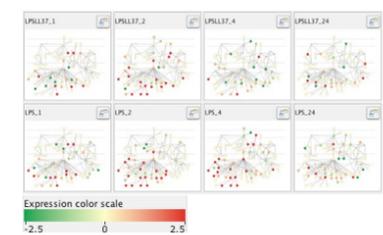 – cluster according to functional grouping

Arrange Networks And Trees
→ Node-link Diagrams
Connections and Marks
✓ NETWORKS ✓ TREES

What?
Why?
How?

How: Arrange space

• automatic layout similar to hand-drawn diagrams
  – vertical compartment according to subcellular location attribute

www.nature.com/nri/focus/tlr/nri1397.html

What?
Why?
How?
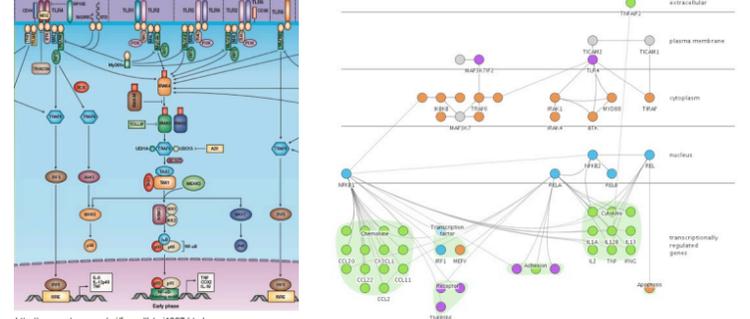
How: Idiom design decisions

• facet: partition data into multiple views
  – juxtapose views side by side
    • same encoding, different data: *small multiples*
    • nodes in each view colored by expression levels for experimental condition

Facet
→ Juxtapose
→ Partition

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

Expression color scale
-2.5   0   2.5

What?
Why?
How?

How: Juxtapose vs. animate

Manipulate
→ Change

LPSLL37_1

• comparison difficult across many frames with with many changes everywhere

• rule of thumb: eyes beat memory
  – principle: external cognition vs. internal memory
    • easy to compare by moving eyes between side-by-side views
   • harder to compare memory of what you saw to visible view

---
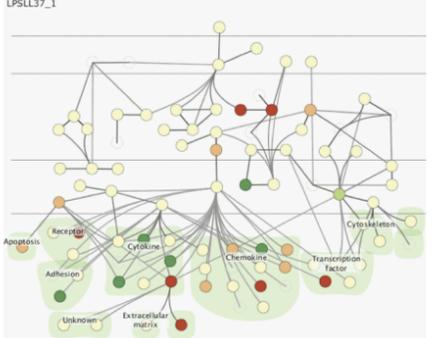
Cerebral contributions

• multiple juxtaposed views support interactive comparison between gene expression level experimental data and network context

• automatic network layout algorithm in spirit of hand drawn diagrams
  – localization and functional group attributes affect spatial position

• open source
  – Cytoscape plugin
  – InnateDB database integration

  http://www.pathogenomics.ca/cerebral/

Outline

- introduction
- Cerebral
- TreeJuxtaposer
- MizBee
- summary and conclusions
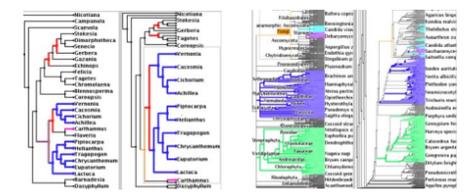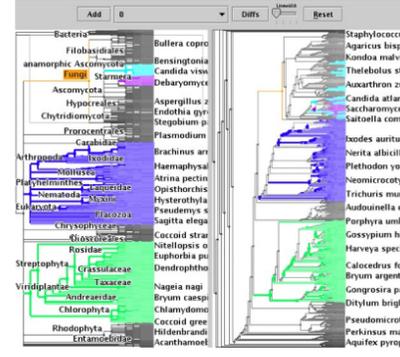

What?
Why?
How?

17
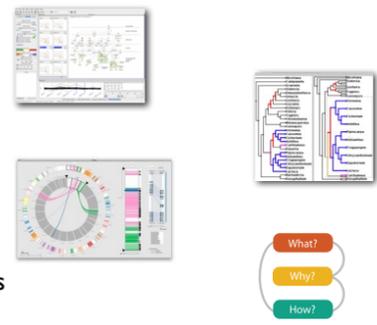
TreeJuxtaposer

Scalable Tree Comparison using Focus+Context with Guaranteed Visibility

joint work with:
François Guimbretière, Serdar Tasiran, Li Zhang, Yunhong Zhou

http://www.cs.ubc.ca/labs/imager/tr/2003/tj/

18

## TreeJuxtaposer video

19

What and why: Data and task abstraction

- data: trees
  - phylogenetic tree reconstruction
    - siblings unordered, interior nodes inferred
- task: compare topological structure
  - larger query scopes require more explicit tool support
    - compare several is more difficult than identify/inspect one
      - even trickier: summarize all
- derived data: structural differences
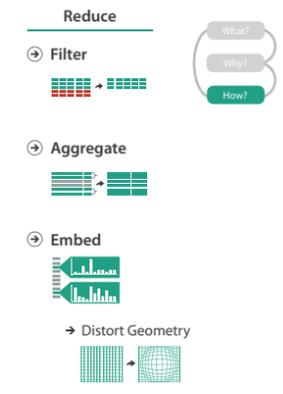 - best corresponding node in other tree

20
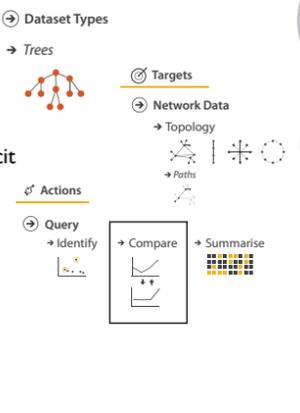
How: Idiom design decisions

- juxtapose linked views
  - show two tree layouts side by side
  - linked navigation
- encode with color: linked highlighting
  - structural differences
  - corresponding subtree (click select)
  - best corresponding node (hover select)

21
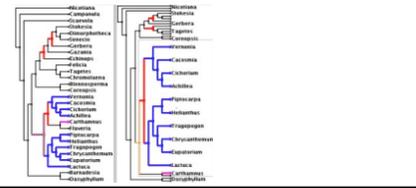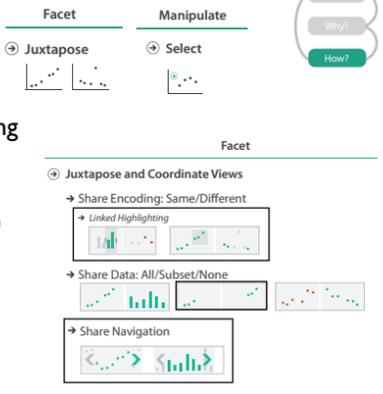
How: Idiom design decisions

- embed focus+context in single view
  - reduce with complex combination of filtering and aggregation
- distort geometry
  - metaphor: stretch and squish navigation
  - shape: rectilinear
  - foci: multiple
  - impact: global

Reduce
- Filter
- Aggregate
- Embed

22

TreeJuxtaposer contributions

- first interactive tree comparison system
  - derive structural difference data to support comparison task
    - subquadratic algorithm: best corresponding node
  - juxtapose views with cross-dataset linked highlighting
- embed focus+context information in single view with stretch and squish navigation
  - sublinear algorithm: guaranteed visibility of structure marks even when squished

- open source
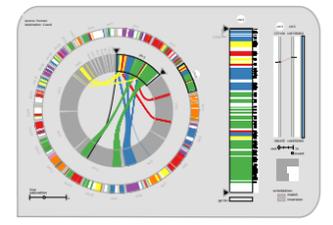 http://olduvai.sf.net/tj

23

Outline

- introduction
- Cerebral
- TreeJuxtaposer
- MizBee
- summary and conclusions

What?
Why?
How?

24
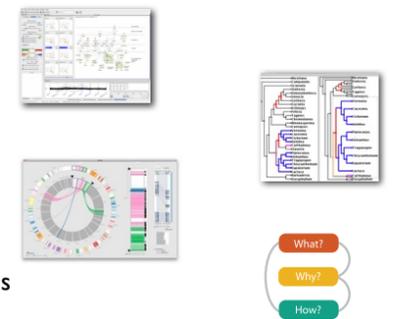
MizBee

A Multiscale Synteny Browser

joint work with:
Miriah Meyer, Hanspeter Pfister

http://www.cs.utah.edu/~miriah/mizbee

25

MizBee video

26
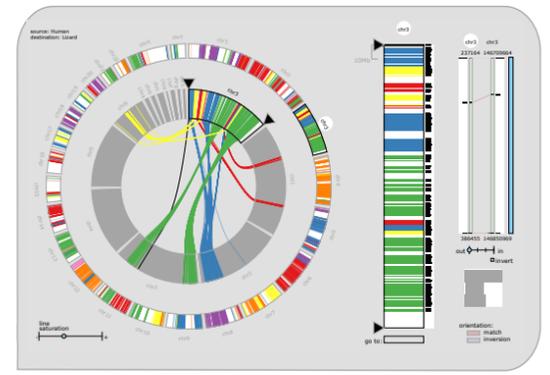
What: Data abstraction

- data: multiscale lists
  - features: hundreds of thousands
    - ordered attribute: position in chromosome sequence coordinates
    - categorical attributes: orientation, chromosome of matching feature
    - quantitative attributes: length, similarity score
  - syntenic blocks: thousands
    - contiguous sets of features on same chromosome
    - combine thresholded features if
      - destination chromosome and orientation match
      - close together
  - chromosomes: dozens
  - genomes: two

27
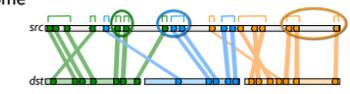
Why: Tasks in domain language

- analyze conservation (similarity) relationships between genomic features
  - high-level biology questions
    - evolution
      - how long ago did two species share common ancestor?
    - function
      - which segment of the genome is responsible for specific function in the cell?
    - ...
  - low-level data-centric questions
    - algorithm refinement
      - are paired features within a block contiguous?
      - which chromosomes share conserved blocks?
      - are similarity scores alike within block?
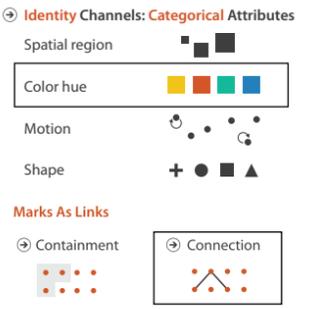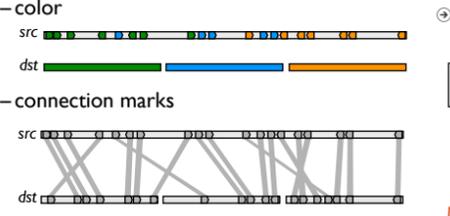   - ...

28

Why: Tasks abstraction

- relationship types: proximity, size, orientation, similarity
- data scales: genome, chromosome, block, feature
- topics: algorithm in/out, block reliability, high-level science

	genome	chromosome	block	feature	proximity / location	size	orientation	similarity
Which chromosomes share conserved blocks?		x				x		
For one chromosome, how many other chromosomes does it share blocks with?	x	x				x		
What is the density of coverage and where are the gaps on: chromosomes? blocks?	x	x	x			x		
Where are the blocks: on chromosomes? around a specific location on a chromosome?	x	x	x			x		
What are the sizes and locations of other genomic features near a block?				x		x	x	
How large are the blocks?			x			x		
Do neighboring blocks go to the same: chromosomes? relative location on a chromosome?	x	x			x			
Are the orientations matched or inverted for: block pairs? feature pairs?			x	x			x	
Do the orientations match for pairs of: neighboring blocks? features within a block?			x	x			x	
Are similarity scores alike: with respect to neighboring blocks? within a block?			x	x				x
Are the paired features within a block contiguous?				x	x			
How large is a feature relative to other genes within a block?				x		x		
What are the sizes, locations, and names of features within a block?				x	x	x		
What are the differences between individual nucleotides of feature pairs?				x	x			x

29

How: Idiom design choices

- encode match relationships between chromosome segments with both
  - color
  - connection marks

Identity Channels: Categorical Attributes

Spatial region

Color hue

Motion

Shape

Marks As Links

- Containment
- Connection

30

How: Arrange space

- design space of arrangements

Mauve [Darling04]

Cineteny

MizBee

	contiguous	circular	discrete
separate			
combined			

Apollo [Lewis02]

31
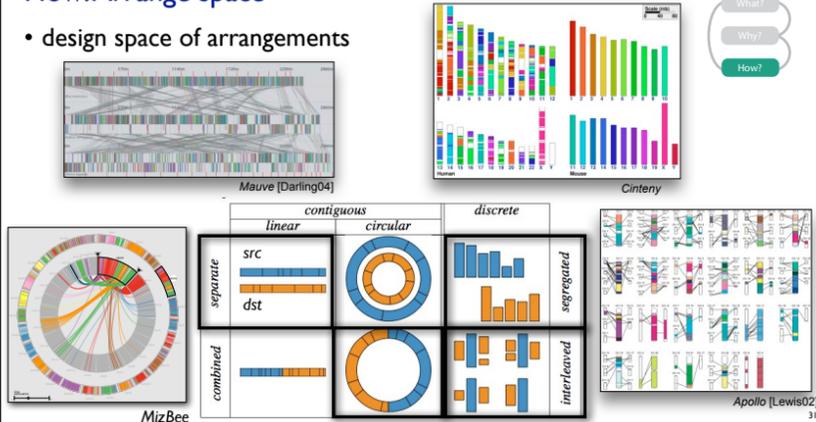
How: Idiom design choices

- juxtapose linked views
  - *multiform overview-detail*
    - three views: genome, chromosome, block
    - different visual encoding in each

Facet
- Juxtapose

32
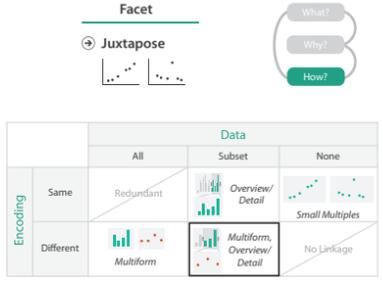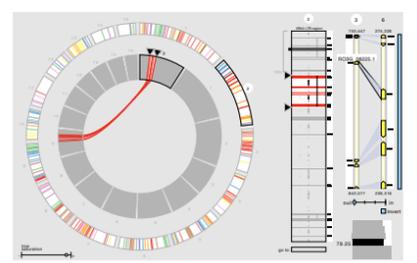
How: Idiom design choices

- axis orientation
  - radial: genome
  - rectilinear: chromosome, block
    - aligned position more accurate than angle

Arrange

→ Axis Orientation

→ Rectilinear | → Parallel | → Radial

→ **Magnitude** Channels: **Ordered** Attributes

Position on common scale

Position on unaligned scale

Length (1D size)

Tilt/angle

What? / Why? / How?

33

How: Idiom design choices

- filter

Reduce

→ Filter

What? / Why? / How?

34

How: Idiom design choices

- outer ring: summarize relationships with color
  - select one chromosome from set of source chromosomes
- inner ring:
  - destination chromosomes around copy of selected source chromosome
  - show relationship details with connection marks as well as color

Manipulate

→ Select

♂ **Actions**

→ Query

→ Identify | → Compare | → Summarise

What? / Why? / How?

35

MizBee contributions

- first synteny browser with side-by-side linked views
  - across the range of scales
  - encoding all four conservation relationship types
    - proximity, size, orientation, similarity

- open source
  http://www.cs.utah.edu/~miriah/mizbee

36

Outline

- introduction
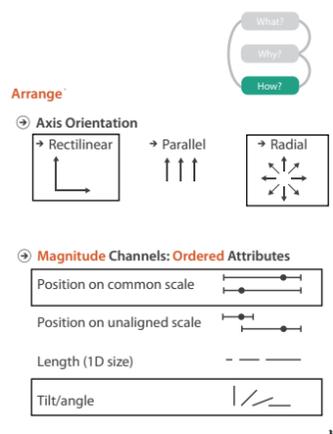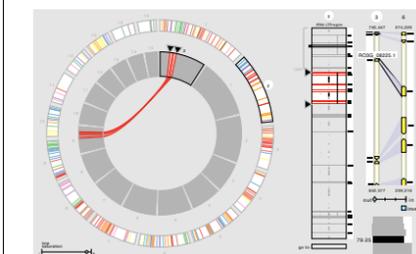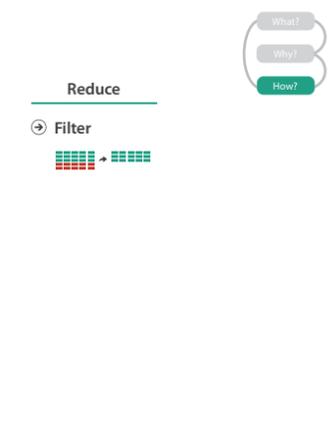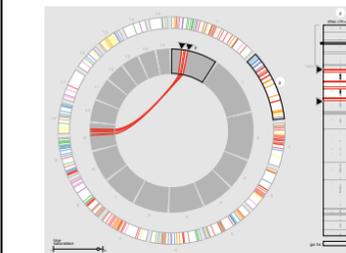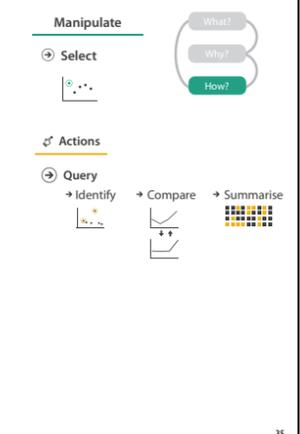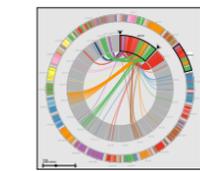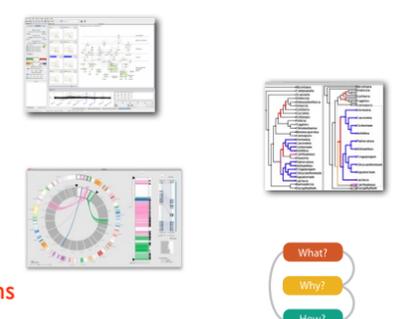
- Cerebral

- TreeJuxtaposer

- MizBee

- summary and conclusions

What? / Why? / How?

37

Visualization Analysis & Design

Tamara Munzner

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

What? / Why? / How?

Visualization Analysis and Design.
Munzner. AK Peters / Taylor and Francis, to appear Oct 2014.

38

What?

Datasets | **Attributes**

→ Data Types: → Items → Attributes → Links → Positions → Grids

→ Attribute Types: → Categorical

→ Data and Dataset Types

Tables | Networks & Trees | Fields | Geometry | Clusters, sets, lists

→ Ordered → Ordinal → Quantitative

→ Dataset Types: → Tables → Networks → Fields (Continuous)

→ Ordering Direction: → Sequential → Diverging → Cyclic

→ Geometry (Spatial) → Dataset Availability: → Static → Dynamic

What? / Why? / How?

39

Why?

♂ **Actions** | ⊙ **Targets**

→ Analyze → Consume → Discover → Present → Enjoy
→ Produce → Annotate → Record → Derive

→ All Data → Trends → Outliers → Features
→ Attributes → One → Many → Distribution → Dependency → Correlation → Similarity → Extremes

→ Search
	Target known	Target unknown
Location known	Lookup	Browse
Location unknown	Locate	Explore

→ Network Data → Topology → Paths

→ Query → Identify → Compare → Summarise

→ Spatial Data → Shape

What? / Why? / How?

40

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? / How?

41

Channels: Expressiveness types and effectiveness rankings

→ **Magnitude** Channels: **Ordered** Attributes

Position on common scale
Position on unaligned scale
Length (1D size)
Tilt/angle
Area (2D size)
Depth (3D position)
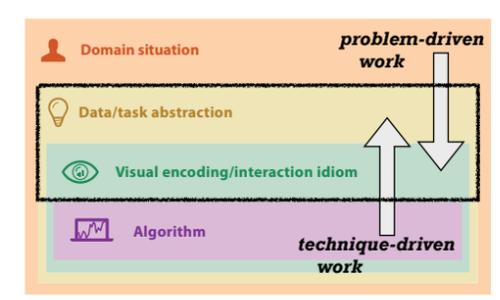Color luminance
Color saturation
Curvature
Volume (3D size)

Best / Least (Effectiveness)

→ **Identity** Channels: **Categorical** Attributes

Spatial region
Color hue
Motion
Shape

42

Four levels of design and validation

- inverse cases: problem-driven vs. technique-driven work
  - call for action: more problem-driven work in bioinformatics
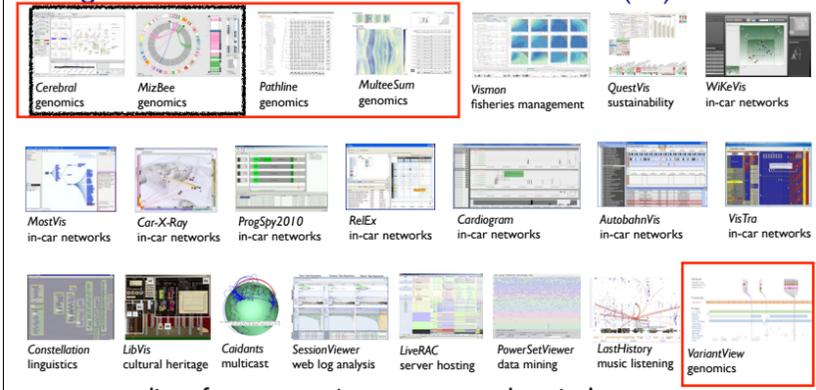   - could be beneficial in broader contexts, beyond biovis!

👤 **Domain situation**

📋 **Data/task abstraction**

👁 **Visual encoding/interaction idiom**

📈 **Algorithm**

problem-driven work

technique-driven work

43

Design Study Methodology

Reflections from the Trenches and from the Stacks

joint work with:
Michael Sedlmair, Miriah Meyer
http://www.cs.ubc.ca/labs/imager/tr/2012/dsm/

ALGORITHM AUTOMATION POSSIBLE

DESIGN STUDY METHODOLOGY SUITABLE

TASK CLARITY / NOT ENOUGH DATA

INFORMATION LOCATION

Design Study Methodology: Reflections from the Trenches and from the Stacks.
Sedlmair, Meyer, Munzner. IEEE Trans. Visualization and Computer Graphics 18(12): 2431-2440, 2012 (Proc. InfoVis 2012).

44

Design Studies: Lessons learned after 21 of them (+1)

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 | VariantView genomics |

- commonality of representations cross-cuts domains!

45

Methodology for Problem-Driven Work

- definitions

- 9-stage framework

- 32 pitfalls and how to avoid them

DESIGN STUDY METHODOLOGY SUITABLE

PRECONDITION | CORE | ANALYSIS

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 change for change: existing tools are good enough	winnow

46

Conclusions

- three biovis systems analyzed
  - Cerebral
  - TreeJuxtaposer
  - MizBee

- analysis framework big ideas
  - what: data abstraction
    - including transformation through deriving data
  - why: task abstraction
    - translate from domain-specific
  - how: visual encoding and interaction idioms
  - scaffolding for thinking systematically about full design space
    - describe existing, generate new

47

More Information

- this talk
  http://www.cs.ubc.ca/~tmm/talks.html#biovis14
- papers, videos, software, talks, courses
  http://www.cs.ubc.ca/~tmm
- book *(to appear Oct 2014)*
  http://www.cs.ubc.ca/~tmm/vadbook

- acknowledgements
  - funding: Agilent, NSERC, NSF
  - talk feedback: Matt Brehmer

Visualization Analysis & Design
Tamara Munzner