Outline

• **Session 1** 8:30-10:10am
  – Analysis: What, Why, How
  – Marks and Channels
  – Arrange Tables
  – Arrange Spatial Data
  – Arrange Networks and Trees

• **Session 2** 10:30am-12:10pm
  – Map Color and Other Channels
  – Manipulate: Change, Select, Navigate
  – Facet: Juxtapose, Partition, Superimpose
  – Reduce: Filter, Aggregate
  – Embed: Focus+Context

http://www.cs.ubc.ca/~tmm/talks.html#halfdaycourse14
Defining visualization (vis)

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: don’t know exactly what to ask in advance

**Anscombe’s Quartet**

<table>
<thead>
<tr>
<th>Identical statistics</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>x mean</td>
<td>9</td>
</tr>
<tr>
<td>x variance</td>
<td>10</td>
</tr>
<tr>
<td>y mean</td>
<td>8</td>
</tr>
<tr>
<td>y variance</td>
<td>4</td>
</tr>
<tr>
<td>x/y correlation</td>
<td>1</td>
</tr>
</tbody>
</table>
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**
    - **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


Analysis example: Compare idioms

SpaceTree

TreeJuxtaposer


What?

Why?

How?

**Datasets**

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

- **Data and Dataset Types**
  - **Tables**
    - Items
    - Attributes
  - **Networks & Trees**
    - Items (nodes)
    - Grids
    - Positions
    - Attributes
  - **Geometry**
    - Items
    - Positions
    - Attributes
  - **Clusters, Sets, Lists**
    - Items

**Attributes**

- **Attribute Types**
  - Categorical
    - +
    - 
    - ◆
    - ▲
  - Ordered
    - Ordinal
    - 📈
  - Quantitative
  - 🏁

**Ordering Direction**

- Sequential
- Diverging
- Cyclic

**Dataset Availability**

- Static
- Dynamic

---

**Why?**

**How?**

**What?**
Dataset and data types

Dataset Types

- Tables
- Networks
- Fields (Continuous)
- Geometry (Spatial)

Data Types

- Items
- Attributes
- Links
- Positions
- Grids
Attribute types

- Attribute Types
  - Categorical
  - Ordered
  - Ordinal
  - Quantitative

- Ordering Direction
  - Sequential
  - Diverging
  - Cyclic
• \{action, target\} pairs
  – *discover distribution*
  – *compare trends*
  – *locate outliers*
  – *browse topology*
High-level actions: Analyze

- consume
  - discover vs present
    - classic split
    - aka explore vs explain
  - enjoy
    - newcomer
    - aka casual, social

- produce
  - annotate, record
  - derive
    - crucial design choice
Actions: Mid-level search, low-level query

• what does user know?
  – target, location

• how much of the data matters?
  – one, some, all

Search

<table>
<thead>
<tr>
<th>Location known</th>
<th>Target known</th>
<th>Target unknown</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Lookup</td>
<td>Browse</td>
</tr>
</tbody>
</table>

Query

Identify

Compare

Summarize
Why: 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?
Further reading

  - Chap 1: What’s Vis, and Why Do It?
  - Chap 2: What: Data Abstraction
  - Chap 3: Why: Task Abstraction


- **A taxonomy of tools that support the fluent and flexible use of visualizations. Heer and Shneiderman. Communications of the ACM 55:4 (2012), 45–54.**


- **Visualization of Time-Oriented Data. Aigner, Miksch, Schumann, and Tominski. Springer, 2011.**
Outline

• **Session 1** 8:30-10:10am
  – Analysis: What, Why, How
  – Marks and Channels
  – Arrange Tables
  – Arrange Spatial Data
  – Arrange Networks and Trees

• **Session 2** 10:30am-12:10pm
  – Map Color and Other Channels
  – Manipulate: Change, Select, Navigate
  – Facet: Juxtapose, Partition, Superimpose
  – Reduce: Filter, Aggregate
  – Embed: Focus+Context

[http://www.cs.ubc.ca/~tmm/talks.html#halfdaycourse14](http://www.cs.ubc.ca/~tmm/talks.html#halfdaycourse14)
Visual encoding

• analyze idiom structure
Definitions: Marks and channels

- **marks**
  - geometric primitives

- **channels**
  - control appearance of marks
  - can redundantly code with multiple channels

- **interactions**
  - point marks only convey position; no area constraints
    - can be size and shape coded
  - line marks convey position and length
    - can only be size coded in 1D (width)
  - area marks fully constrained
    - cannot be size or shape coded
Visual encoding

• 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: 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)

Identity Channels: Categorical Attributes
- Spatial region
- Color hue
- Motion
- Shape

Effectiveness rankings:
- Best
- Least
Effectiveness and expressiveness principles

• effectiveness principle
  – encode most important attributes with highest ranked channels

• expressiveness principle
  – match channel and data characteristics


• rankings: where do they come from?
  – accuracy
  – discriminability
  – separability
  – popout
Accuracy: Fundamental Theory

Steven’s Psychophysical Power Law: \( S = I^N \)
Accuracy: Vis experiments

Discriminability: How many usable steps?

• must be sufficient for number of attribute levels to show
  – linewidth: few bins

[mappa.mundi.net/maps/maps 014/telegeography.html]
Separability vs. Integrality

Position
+ Hue (Color)

Size
+ Hue (Color)

Width
+ Height

Red
+ Green

2 groups each

3 groups total: integral area

4 groups total: integral hue

Fully separable

Some interference

Some/significant interference

Major interference

2 groups each
Popout

• find the red dot
  – how long does it take?

• parallel processing on many individual channels
  – speed independent of distractor count
  – speed depends on channel and amount of difference from distractors

• serial search for (almost all) combinations
  – speed depends on number of distractors

25
many channels: tilt, size, shape, proximity, shadow direction, ...

but not all! parallel line pairs do not pop out from tilted pairs
Grouping

- containment
- connection

Marks as Links

- Containment
- Connection

Identity Channels: Categorical Attributes

- Spatial region
- Color hue
- Motion
- Shape

proximity
- same spatial region

similarity
- same values as other categorical channels
Relative vs. absolute judgements

• perceptual system mostly operates with relative judgements, not absolute
  – that’s why accuracy increases with common frame/scale and alignment
  – Weber’s Law: ratio of increment to background is constant
    • filled rectangles differ in length by 1:9, difficult judgement
    • white rectangles differ in length by 1:2, easy judgement

Further reading

  – Chap 5: Marks and Channels


Outline

• **Session 1 8:30-10:10am**
  – Analysis: What, Why, How
  – Marks and Channels
  – Arrange Tables
  – Arrange Spatial Data
  – Arrange Networks and Trees

• **Session 2 10:30am-12:10pm**
  – Map Color and Other Channels
  – Manipulate: Change, Select, Navigate
  – Facet: Juxtapose, Partition, Superimpose
  – Reduce: Filter, Aggregate
  – Embed: Focus+Context

http://www.cs.ubc.ca/~tmm/talks.html#halfdaycourse14
### How?

<table>
<thead>
<tr>
<th>Encode</th>
<th>Manipulate</th>
<th>Facet</th>
<th>Reduce</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Arrange</strong></td>
<td><strong>Map</strong> from <em>categorical</em> and <em>ordered</em> attributes</td>
<td><strong>Change</strong></td>
<td><strong>Filter</strong></td>
</tr>
<tr>
<td>➜ Express</td>
<td>➜ Color: <strong>Hue</strong> ➜ <strong>Saturation</strong> ➜ <strong>Luminance</strong></td>
<td><strong>Select</strong></td>
<td><strong>Facet</strong></td>
</tr>
<tr>
<td>➜ Order</td>
<td>➜ <strong>Size, Angle, Curvature, ...</strong></td>
<td><strong>Partition</strong></td>
<td><strong>Aggregate</strong></td>
</tr>
<tr>
<td>➜ Use</td>
<td>➜ <strong>Shape</strong></td>
<td>➜ <strong>Superimpose</strong></td>
<td><strong>Embed</strong></td>
</tr>
<tr>
<td>➜ Align</td>
<td>➜ <strong>Navigate</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>➜ Separate</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### What? Why? How?
Arrange space

Encode

- Arrange
  - Express
  - Order
- Separate
- Align
- Use

[Diagram of space arrangement]

[Map of the United States]
Arrange tables

Express Values

Separate, Order, Align Regions

Separate
Order
Align

Axis Orientation

Rectilinear
Parallel
Radial

Layout Density

Dense
Space-Filling

1 Key
List

2 Keys
Matrix

3 Keys
Volume

Many Keys
Recursive Subdivision
Keys and values

- **key**
  - independent attribute
  - used as unique index to look up items
  - simple tables: 1 key
  - multidimensional tables: multiple keys
- **value**
  - dependent attribute, value of cell
- **classify arrangements by key count**
  - 0, 1, 2, many...

Express Values

- 1 Key
- 2 Keys
- 3 Keys
- Many Keys

Tables

- Attributes (columns)
- Items (rows)
- Cell containing value

Multidimensional Table

- Key 1
- Key 2
- Attributes
- Value in cell

Recursive Subdivision
Idiom: scatterplot

- **express** values
  - quantitative attributes
- no keys, only values
  - data
    - 2 quant attribs
  - mark: points
  - channels
    - horiz + vert position
- tasks
  - find trends, outliers, distribution, correlation, clusters
- scalability
  - hundreds of items

Some keys: Categorical regions

- **regions**: contiguous bounded areas distinct from each other
  - using space to *separate* (proximity)
  - following expressiveness principle for categorical attributes

- use ordered attribute to *order* and *align* regions

- 1 Key: List
- 2 Keys: Matrix
- 3 Keys: Volume
- Many Keys: Recursive Subdivision
Idiom: **bar chart**

- **one key, one value**
  - data
    - 1 categ attrib, 1 quant attrib
  - mark: lines
  - channels
    - length to express quant value
    - spatial regions: one per mark
      - separated horizontally, aligned vertically
      - ordered by quant attrib
        » by label (alphabetical), by length attrib (data-driven)
- task
  - compare, lookup values
- scalability
  - dozens to hundreds of levels for key attrib
Idiom: **stacked bar chart**

- one more key
  - data
    - 2 categ attrib, 1 quant attrib
  - mark: vertical stack of line marks
    - **glyph**: composite object, internal structure from multiple marks
  - channels
    - length and color hue
    - spatial regions: one per glyph
      - aligned: full glyph, lowest bar component
      - unaligned: other bar components
  - task
    - part-to-whole relationship
  - scalability
    - several to one dozen levels for stacked attrib

Idiom: **streamgraph**

- generalized stacked graph
  - emphasizing horizontal continuity
    - vs vertical items
  - data
    - 1 categ key attrib (artist)
    - 1 ordered key attrib (time)
    - 1 quant value attrib (counts)
  - derived data
    - geometry: layers, where height encodes counts
    - 1 quant attrib (layer ordering)
- scalability
  - hundreds of time keys
  - dozens to hundreds of artist keys
    - more than stacked bars, since most layers don’t extend across whole chart

Idiom: **line chart**

- one key, one value
  - data
    - 2 quant attribs
  - mark: points
    - line connection marks between them
  - channels
    - aligned lengths to express quant value
    - separated and ordered by key attrib into horizontal regions
- task
  - find trend
    - connection marks emphasize ordering of items along key axis by explicitly showing relationship between one item and the next
Choosing bar vs line charts

• depends on type of key attrib
  – bar charts if categorical
  – line charts if ordered

• do not use line charts for categorical key attrs
  – violates expressiveness principle
  • implication of trend so strong that it overrides semantics!
    – “The more male a person is, the taller he/she is”

Idiom: heatmap

• two keys, one value
  – data
    • 2 categ attrs (gene, experimental condition)
    • 1 quant attr (expression levels)
  – marks: area
    • separate and align in 2D matrix
      – indexed by 2 categorical attributes
  – channels
    • color by quant attr
      – (ordered diverging colormap)
  – task
    • find clusters, outliers
  – scalability
    • 1M items, 100s of categ levels, ~10 quant attr levels
Idiom: **cluster heatmap**

- **in addition**
  - derived data
    - 2 cluster hierarchies
  - dendrogram
    - parent-child relationships in tree with connection line marks
    - leaves aligned so interior branch heights easy to compare
  - heatmap
    - marks (re-)ordered by cluster hierarchy traversal
Axis Orientation

- Rectilinear
- Parallel
- Radial
Idioms: scatterplot matrix, parallel coordinates

- scatterplot matrix (SPLOM)
  - rectilinear axes, point mark
  - all possible pairs of axes
  - scalability
    - one dozen attribs
    - dozens to hundreds of items

- parallel coordinates
  - parallel axes, jagged line representing item
  - rectilinear axes, item as point
    - axis ordering is major challenge
  - scalability
    - dozens of attribs
    - hundreds of items

Task: Correlation

- scatterplot matrix
  - positive correlation
    - diagonal low-to-high
  - negative correlation
    - diagonal high-to-low
  - uncorrelated
- parallel coordinates
  - positive correlation
    - parallel line segments
  - negative correlation
    - all segments cross at halfway point
  - uncorrelated
    - scattered crossings


Idioms: **pie chart, polar area chart**

- **pie chart**
  - area marks with angle channel
  - accuracy: angle/area much less accurate than line length

- **polar area chart**
  - area marks with length channel
  - more direct analog to bar charts

- **data**
  - 1 categ key attrib, 1 quant value attrib

- **task**
  - part-to-whole judgements
Idioms: normalized stacked bar chart

• task
  – part-to-whole judgements

• normalized stacked bar chart
  – stacked bar chart, normalized to full vert height
  – single stacked bar equivalent to full pie
    • high information density: requires narrow rectangle

• pie chart
  – information density: requires large circle

http://bl.ocks.org/mbostock/3887235
http://bl.ocks.org/mbostock/3886208
http://bl.ocks.org/mbostock/3886394.
Idiom: **glyphmaps**

- rectilinear good for linear vs nonlinear trends
- radial good for cyclic patterns

Orientation limitations

- rectilinear: scalability wrt #axes
  - 2 axes best
  - 3 problematic
    - more in afternoon
  - 4+ impossible
- parallel: unfamiliarity, training time
- radial: perceptual limits
  - angles lower precision than lengths
  - asymmetry between angle and length
    - can be exploited!

Further reading

  – Chap 7: Arrange Tables


Outline

- **Session 1 8:30-10:10am**
  - Analysis: What, Why, How
  - Marks and Channels
  - Arrange Tables
  - Arrange Spatial Data
  - Arrange Networks and Trees

- **Session 2 10:30am-12:10pm**
  - Map Color and Other Channels
  - Manipulate: Change, Select, Navigate
  - Facet: Juxtapose, Partition, Superimpose
  - Reduce: Filter, Aggregate
  - Embed: Focus+Context

http://www.cs.ubc.ca/~tm/talks.html#/halfdaycourse14
Arrange spatial data

Use Given

Geometry
  Geographic
  Other Derived

Spatial Fields

Scalar Fields (one value per cell)
  Isocontours
  Direct Volume Rendering

Vector and Tensor Fields (many values per cell)
  Flow Glyphs (local)
  Geometric (sparse seeds)
  Textures (dense seeds)
  Features (globally derived)
Idiom: choropleth map

• **use** given spatial data
  – when central task is understanding spatial relationships

• data
  – geographic geometry
  – table with 1 quant attribute per region

• encoding
  – use given geometry for area mark boundaries
  – sequential segmented colormap [*more later*]

http://bl.ocks.org/mbostock/4060606
Idiom: **topographic map**

- **data**
  - geographic geometry
  - scalar spatial field
    - 1 quant attribute per grid cell

- **derived data**
  - isoline geometry
    - isocontours computed for specific levels of scalar values

*Land Information New Zealand Data Service*
Idiom: isosurfaces

• data
  – scalar spatial field
    • 1 quant attribute per grid cell

• derived data
  – isosurface geometry
    • isocontours computed for specific levels of scalar values

• task
  – shape understanding
  – spatial relationships
Idioms: **DVR, multidimensional transfer functions**

- **Direct volume rendering**
  - Transfer function maps scalar values to color, opacity
    - No derived geometry

- **Multidimensional transfer functions**
  - Derived data in joint 2D histogram
    - Horiz axis: data values of scalar function
    - Vert axis: gradient magnitude
      - Direction of fastest change
    - [more later: cutting planes and histograms]

---

Vector and tensor fields

• data
  – many attributes per cell

• idiom families
  – flow glyphs
    • purely local
  – geometric flow
    • derived data from tracing particle trajectories
    • sparse set of seed points
  – texture flow
    • derived data, dense seeds
  – feature flow
    • global computation to detect features
      – encoded with one of methods above


Vector fields

- empirical study tasks
  - finding critical points, identifying their types
  - identifying what type of critical point is at a specific location
  - predicting where a particle starting at a specified point will end up (advection)


Idiom: similarity-clustered streamlines

• data
  – 3D vector field

• derived data (from field)
  – streamlines: trajectory particle will follow

• derived data (per streamline)
  – curvature, torsion, tortuosity
  – signature: complex weighted combination
  – compute cluster hierarchy across all signatures
  – encode: color and opacity by cluster

• tasks
  – find features, query shape

• scalability
  – millions of samples, hundreds of streamlines

Further reading

  – *Chap 8: Arrange Spatial Data*


Outline

• **Session 1 8:30-10:10am**
  – Analysis: What, Why, How
  – Marks and Channels
  – Arrange Tables
  – Arrange Spatial Data
  – Arrange Networks and Trees

• **Session 2 10:30am-12:10pm**
  – Map Color and Other Channels
  – Manipulate: Change, Select, Navigate
  – Facet: Juxtapose, Partition, Superimpose
  – Reduce: Filter, Aggregate
  – Embed: Focus+Context

http://www.cs.ubc.ca/~tmm/talks.html#halfdaycourse14
Arrange networks and trees

- **Node–Link Diagrams**
  - Connection Marks
    - NETWORKS
    - TREES

- **Adjacency Matrix**
  - Derived Table
    - NETWORKS
    - TREES

- **Enclosure**
  - Containment Marks
    - NETWORKS
    - TREES
Idiom: **force-directed placement**

- **visual encoding**
  - link connection marks, node point marks

- **considerations**
  - spatial position: no meaning directly encoded
    - left free to minimize crossings
  - proximity semantics?
    - sometimes meaningful
    - sometimes arbitrary, artifact of layout algorithm
    - tension with length
      - long edges more visually salient than short

- **tasks**
  - explore topology; locate paths, clusters

- **scalability**
  - node/edge density $E < 4N$

Idiom: **adjacency matrix view**

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

---

**Figure 7.5: Comparing matrix and node-link views of a five-node network.**

a) Matrix view. b) Node-link view. From [Henry et al. 07], Figure 3b and 3a. (Permission needed.)

Matrix views of networks can achieve very high information density, up to a limit of one thousand nodes and one million edges, just like cluster heatmaps and all other matrix views that uses small area marks.

Technique: **network matrix view**

Data Types: **network**

Derived Data: **table: network nodes as keys, link status between two nodes as values**

View Comp.: **space: area marks in 2D matrix alignment**

Scalability: **nodes: 1K**, **edges: 1M**

---

7.1.3.3 Multiple Keys: Partition and Subdivide

When a dataset has only one key, then it is straightforward to use that key to separate into one region.

---


Connection vs. adjacency comparison

• adjacency matrix strengths
  – predictability, scalability, supports reordering
  – some topology tasks trainable

• node-link diagram strengths
  – topology understanding, path tracing
  – intuitive, no training needed

• empirical study
  – node-link best for small networks
  – matrix best for large networks
    • if tasks don’t involve topological structure!

Idiom: radial node-link tree

• data
  – tree

• encoding
  – link connection marks
  – point node marks
  – radial axis orientation
    • angular proximity: siblings
    • distance from center: depth in tree

• tasks
  – understanding topology, following paths

• scalability
  – 1K - 10K nodes

Idiom: **treemap**

- **data**
  - tree
  - 1 quant attrib at leaf nodes

- **encoding**
  - area containment marks for hierarchical structure
  - rectilinear orientation
  - size encodes quant attrib

- **tasks**
  - query attribute at leaf nodes

- **scalability**
  - 1M leaf nodes

Link marks: Connection and containment

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

Further reading

  – Chap 9: Arrange Networks and Trees


Outline

• **Session 1** 8:30-10:10am
  – Analysis: What, Why, How
  – Marks and Channels
  – Arrange Tables
  – Arrange Spatial Data
  – Arrange Networks and Trees

• **Session 2** 10:30am-12:10pm
  – Map Color and Other Channels
  – Manipulate: Change, Select, Navigate
  – Facet: Juxtapose, Partition, Superimpose
  – Reduce: Filter, Aggregate
  – Embed: Focus+Context

http://www.cs.ubc.ca/~tmn/talks.html#halfdaycourse14
Idiom design choices: First half

Encode

Why?
How?
What?

Arrange
- Express
- Order
- Use

Separate
Align

Map from categorical and ordered attributes

- Color
  - Hue
  - Saturation
  - Luminance

- Size, Angle, Curvature, ...

- Shape
- Motion
  Direction, Rate, Frequency, ...

What?
Why?
How?
Color: Luminance, saturation, hue

• 3 channels
  – what/where for categorical
    • hue
  – how-much for ordered
    • luminance
    • saturation

• other common color spaces
  – RGB: poor choice for visual encoding
  – HSL: better, but beware
    • lightness ≠ luminance

• transparency
  – useful for creating visual layers
    • but cannot combine with luminance or saturation
## Colormaps

### Categorical

- **Ordered**
  - **Sequential**
  - **Diverging**

### Bivariate

- **Diverging**

### Categorical limits: noncontiguous
- 6-12 bins hue/color
  - far fewer if colorblind
- 3-4 bins luminance, saturation
- size heavily affects salience
  - use high saturation for small regions, low saturation for large

---

Categorical color: Discriminability constraints

• noncontiguous small regions of color: only 6-12 bins

[Cinteny: flexible analysis and visualization of synteny and genome rearrangements in multiple organisms. Sinha and Meller. BMC Bioinformatics, 8:82, 2007.]
Ordered color: Rainbow is poor default

• problems
  – perceptually unordered
  – perceptually nonlinear

• benefits
  – fine-grained structure visible and nameable

• alternatives
  – fewer hues for large-scale structure
  – multiple hues with monotonically increasing luminance for fine-grained
  – segmented rainbows good for categorical, ok for binned


Map other channels

• size
  – length accurate, 2D area ok, 3D volume poor

• angle
  – nonlinear accuracy
    • horizontal, vertical, exact diagonal

• shape
  – complex combination of lower-level primitives
  – many bins

• motion
  – highly separable against static
    • binary: great for highlighting
  – use with care to avoid irritation

Shape

Motion
Further reading

  – Chap 10: Map Color and Other Channels

• ColorBrewer, Brewer.
  – http://www.colorbrewer2.org

  – http://www.stonesc.com/Vis06


Outline

• **Session 1 8:30-10:10am**
  – Analysis: What, Why, How
  – Marks and Channels
  – Arrange Tables
  – Arrange Spatial Data
  – Arrange Networks and Trees

• **Session 2 10:30am-12:10pm**
  – Map Color and Other Channels
  – Manipulate: Change, Select, Navigate
  – Facet: Juxtapose, Partition, Superimpose
  – Reduce: Filter, Aggregate
  – Embed: Focus+Context

http://www.cs.ubc.ca/~tmm/talks.html#halfdaycourse14
# Idiom design choices: Second half

<table>
<thead>
<tr>
<th>Manipulate</th>
<th>Facet</th>
<th>Reduce</th>
</tr>
</thead>
<tbody>
<tr>
<td>✗ Change</td>
<td>✗ Juxtapose</td>
<td>✗ Filter</td>
</tr>
<tr>
<td>✗ Select</td>
<td>✗ Partition</td>
<td>✗ Aggregate</td>
</tr>
<tr>
<td>✗ Navigate</td>
<td>✗ Superimpose</td>
<td>✗ Embed</td>
</tr>
</tbody>
</table>

- **Manipulate**:
  - Change
  - Select
  - Navigate

- **Facet**:
  - Juxtapose
  - Partition
  - Superimpose

- **Reduce**:
  - Filter
  - Aggregate
  - Embed
Manipulate

- **Change over Time**
- **Select**
- **Navigate**
  - Item Reduction
    - Zoom
    - Geometric or Semantic
  - Pan/Translate
    - Constrained
- **Attribute Reduction**
  - Slice
  - Cut
  - Project
Change over time

• change any of the other choices
  – encoding itself
  – parameters
  – arrange: rearrange, reorder
  – aggregation level, what is filtered...

• why change?
  – one of four major strategies
    • change over time
    • facet data by partitioning into multiple views
    • reduce amount of data shown within view
      – embedding focus + context together
  – most obvious, powerful, flexible
  – interaction entails change
Idiom: **Re-encode**  
System: **Tableau**  

*made using Tableau, [http://tableausoftware.com](http://tableausoftware.com)*
Idiom: **Reorder**

- data: tables with many attributes
- task: compare rankings

**System: LineUp**

Idiom: **Realign**

- stacked bars
  - easy to compare
    - first segment
    - total bar
- align to different segment
  - supports flexible comparison

**System: LineUp**

Idiom: **Animated transitions**

- smooth transition from one state to another
  - alternative to jump cuts
  - support for item tracking when amount of change is limited
- example: multilevel matrix views
  - scope of what is shown narrows down
    - middle block stretches to fill space, additional structure appears within
    - other blocks squish down to increasingly aggregated representations

Select and highlight

• selection: basic operation for most interaction
• design choices
  – how many selection types?
    • click vs hover: heavyweight, lightweight
    • primary vs secondary: semantics (eg source/target)
• highlight: change visual encoding for selection targets
  – color
    • limitation: existing color coding hidden
  – other channels (eg motion)
  – add explicit connection marks between items
Navigate: Changing item visibility

• change viewpoint
  – changes which items are visible within view
  – camera metaphor
    • zoom
      – geometric zoom: familiar semantics
      – semantic zoom: adapt object representation based on available pixels
        » dramatic change, or more subtle one
    • pan/translate
    • rotate
      – especially in 3D
  – constrained navigation
    • often with animated transitions
    • often based on selection set
Idiom: **Semantic zooming**

- visual encoding change
  - colored box
  - sparkline
  - simple line chart
  - full chart: axes and tickmarks

Navigate: Reducing attributes

• continuation of camera metaphor
  – slice
    • show only items matching specific value for given attribute: slicing plane
    • axis aligned, or arbitrary alignment
  – cut
    • show only items on far slide of plane from camera
  – project
    • change mathematics of image creation
      – orthographic
      – perspective
      – many others: Mercator, cabinet, ...
Further reading

  – Chap 11: Manipulate View


Outline

• **Session 1** 8:30-10:10am
  – Analysis: What, Why, How
  – Marks and Channels
  – Arrange Tables
  – Arrange Spatial Data
  – Arrange Networks and Trees

• **Session 2** 10:30am-12:10pm
  – Map Color and Other Channels
  – Manipulate: Change, Select, Navigate
  – Facet: Juxtapose, Partition, Superimpose
  – Reduce: Filter, Aggregate
  – Embed: Focus+Context

[http://www.cs.ubc.ca/~tmm/talks.html#halfdaycourse14](http://www.cs.ubc.ca/~tmm/talks.html#halfdaycourse14)
Facet

- Juxtapose

- Partition

- Superimpose
Juxtapose and coordinate views

- Share Encoding: Same/Different
  - Linked Highlighting

- Share Data: All/Subset/None

- Share Navigation
Idiom: **Linked highlighting**

- see how regions contiguous in one view are distributed within another
  - powerful and pervasive interaction idiom

- encoding: different
  - *multiform*

- data: all shared

Idiom: **bird’s-eye maps**

- encoding: same
- data: subset shared
- navigation: shared
  - bidirectional linking

- differences
  - viewpoint
  - (size)

- overview-detail

---

Idiom: **Small multiples**

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

---

Coordinate views: Design choice interaction

<table>
<thead>
<tr>
<th>Encoding</th>
<th>All</th>
<th>Subset</th>
<th>None</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Same</td>
<td>Redundant</td>
<td>Overview/Detail</td>
<td>Small Multiples</td>
<td></td>
</tr>
<tr>
<td>Different</td>
<td>Multiform</td>
<td>Multiform, Overview/Detail</td>
<td>No Linkage</td>
<td></td>
</tr>
</tbody>
</table>

- **Encoding**: Determines whether the views are the same or different.
- **All Subset None**: Represents the view options available.
- **Data**: Shows the possible data representations for each encoding option.
Juxtapose design choices

• design choices
  – view count
    • few vs many
      – how many is too many? open research question
  – view visibility
    • always side by side vs temporary popups
  – view arrangement
    • user managed vs system arranges/aligns

• why juxtapose views?
  – benefits: eyes vs memory
    • lower cognitive load to move eyes between 2 views than remembering previous state with 1
  – costs: display area
    • 2 views side by side each have only half the area of 1 view
System: **Improvise**

- investigate power of multiple views
  - pushing limits on view count, interaction complexity
  - reorderable lists
    - easy lookup
    - useful when linked to other encodings

Partition into views

• how to divide data between views
  – encodes association between items using spatial proximity
  – major implications for what patterns are visible
  – split according to attributes

• design choices
  – how many splits
    • all the way down: one mark per region?
    • stop earlier, for more complex structure within region?
  – order in which attribs used to split
  – how many views
Views and glyphs

• **view**
  – contiguous region in which visually encoded data is shown on the display

• **glyph**
  – object with internal structure that arises from multiple marks

• no strict dividing line
  – view: big/detailed
  – glyph: small/iconic
Partitioning: List alignment

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

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

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

System: HIVE

Partitioning: Recursive subdivision

• switch order of splits
  – neighborhood then type
• very different patterns

Partitioning: Recursive subdivision

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

Partitioning: Recursive subdivision

- different encoding for second-level regions
  - choropleth maps

System: HIVE

Superimpose layers

- **layer**: set of objects spread out over region
  - each set is visually distinguishable group
  - extent: whole view

- design choices
  - how many layers?
  - how are layers distinguished?
  - small static set or dynamic from many possible?
  - how partitioned?
    - heavyweight with attribs vs lightweight with selection

- distinguishable layers
  - encode with different, nonoverlapping channels
    - two layers achieveable, three with careful design
Static visual layering

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

Superimposing limits

- few layers, but many lines
  - up to a few dozen
  - but not hundreds
- superimpose vs juxtapose: empirical study
  - superimposed for local visual, multiple for global
  - same screen space for all multiples, single superimposed
  - tasks
    - local: maximum, global: slope, discrimination

Dynamic visual layering

- interactive, from selection
  - lightweight: click
  - very lightweight: hover

- ex: 1-hop neighbors

Further reading

  – Chap 12: Facet Into Multiple Views
Outline

• **Session 1 8:30-10:10am**
  – Analysis: What, Why, How
  – Marks and Channels
  – Arrange Tables
  – Arrange Spatial Data
  – Arrange Networks and Trees

• **Session 2 10:30am-12:10pm**
  – Map Color and Other Channels
  – Manipulate: Change, Select, Navigate
  – Facet: Juxtapose, Partition, Superimpose
  – Reduce: Filter, Aggregate
  – Embed: Focus+Context

http://www.cs.ubc.ca/~tmm/talks.html#halfdaycourse14
## 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, change, facet

### Reducing Items and Attributes

<table>
<thead>
<tr>
<th>Reduce</th>
<th>Filter</th>
<th>Aggregate</th>
<th>Embed</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Reduce</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Filter</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Items</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Attributes</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Aggregate</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Items</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Attributes</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Filter

- **Items**
- **Attributes**

#### Aggregate

- **Items**
- **Attributes**

#### Embed

- **Reduce**
Idiom: **dynamic filtering**

- item filtering
- browse through tightly coupled interaction
  - alternative to queries that might return far too many or too few

Idiom: scented widgets

• augment widgets for filtering to show information scent
  – cues to show whether value in drilling down further vs looking elsewhere
• concise, in part of screen normally considered control panel

Idiom: **DOSFA**

- attribute filtering
- encoding: star glyphs

Idiom: histogram

- static item aggregation
- task: find distribution
- data: table
- derived data
  - new table: keys are bins, values are counts
- bin size crucial
  - pattern can change dramatically depending on discretization
  - opportunity for interaction: control bin size on the fly
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: **Hierarchical parallel coordinates**

- dynamic item aggregation
- derived data: *hierarchical clustering*
- encoding:
  - cluster band with variable transparency, line at mean, width by min/max values
  - color by proximity in hierarchy

Dimensionality reduction

- attribute aggregation
  - derive low-dimensional target space from high-dimensional measured space
  - use when you can’t directly measure what you care about
    - true dimensionality of dataset conjectured to be smaller than dimensionality of measurements
    - latent factors, hidden variables

Tumor Measurement Data

data: 9D measured space

\[ \text{DR} \]

derived data: 2D target space

Malignant

Benign
**Idiom:** Dimensionality reduction for documents

**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**

In Scatterplot Clusters & points → Out Labels for clusters

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

**Why?**
- Produce
- Annotate
Further reading

  – Chap 13: Reduce Items and Attributes


Outline

• Session 1 8:30-10:10am
  – Analysis: What, Why, How
  – Marks and Channels
  – Arrange Tables
  – Arrange Spatial Data
  – Arrange Networks and Trees

• Session 2 10:30am-12:10pm
  – Map Color and Other Channels
  – Manipulate: Change, Select, Navigate
  – Facet: Juxtapose, Partition, Superimpose
  – Reduce: Filter, Aggregate
  – Embed: Focus+Context

http://www.cs.ubc.ca/~tmm/talks.html#halfdaycourse14
Embed: Focus+Context

- combine information within single view
- elide
  - selectively filter and aggregate
- superimpose layer
  - local lens
- distortion design choices
  - region shape: radial, rectilinear, complex
  - how many regions: one, many
  - region extent: local, global
  - interaction metaphor
Idiom: **DOITrees Revisited**

- **elide**
  - some items dynamically filtered out
  - some items dynamically aggregated together
  - some items shown in detail

Idiom: **Fisheye Lens**

- distort geometry
  - shape: radial
  - focus: single extent
  - extent: local
  - metaphor: draggable lens

http://tulip.labri.fr/TulipDrupal/?q=node/351  
http://tulip.labri.fr/TulipDrupal/?q=node/371
Distortion costs and benefits

• benefits
  – combine focus and context information in single view

• costs
  – length comparisons impaired
    • network/tree topology comparisons unaffected: connection, containment
  – effects of distortion unclear if original structure unfamiliar
  – object constancy/tracking maybe impaired

Further reading

  – Chap 14: Embed: Focus+Context


Sneak preview: Not covered today

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

• **Validation**

  **Domain situation**
  Observe target users using existing tools

  **Data/task abstraction**

  **Visual encoding/interaction idiom**
  Justify design with respect to alternatives

  **Algorithm**
  Measure system time/memory
  Analyze computational complexity

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

  Observe target users after deployment (field study)

  Measure adoption
More Information

• this tutorial
  http://www.cs.ubc.ca/~tmm/talks.html#halfdaycourse14

• papers, videos, software, talks, full courses
  http://www.cs.ubc.ca/group/infovis
  http://www.cs.ubc.ca/~tmm

• book (Nov 2014)
  http://www.cs.ubc.ca/~tmm/vadbook

• acknowledgements
  – illustrations: Eamonn Maguire