

Visual Analytics

Hunting down unknown unknowns, and
opening the black box of bioinformatics

Prof Jan Aerts
Faculty of Engineering, ESAT - STADIUS
KU Leuven

@jandot - jan.aerts@esat.kuleuven.be - <http://visualanalyticsleuven.be>

Overview of lecture

- A. Why visual analytics?
- B. Data visualization
 - Data foundations
 - Human perception foundations
 - Visualization foundations and techniques
- C. Visualization evaluation
- D. Tools of the trade
- E. Examples

Overview of lecture

A. Why visual analytics?

B. Data visualization

- Data foundations
- Human perception foundations
- Visualization foundations and techniques

C. Visualization evaluation

D. Tools of the trade

E. Examples

F. Exercises

A. What's the problem?

hypothesis-driven -> data-driven

Scientific Research Paradigms (Jim Gray, Microsoft)

1st	1,000s years ago	empirical
2nd	100s years ago	theoretical
3rd	last few decades	computational
4rd	today	data exploration

I have an hypothesis -> need to generate data to (dis)prove it.
I have data -> need to find hypotheses that I can test.

What does this mean?

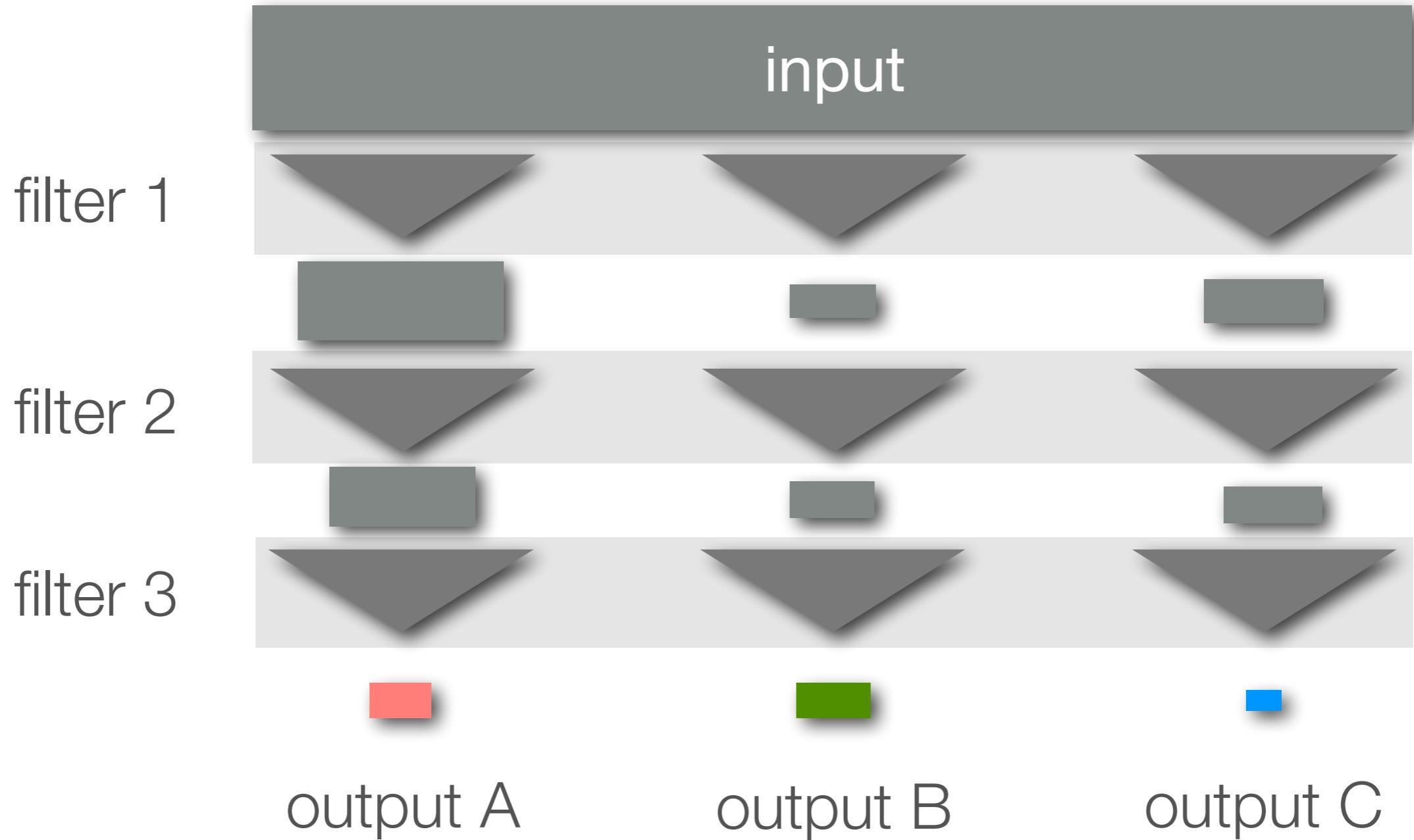
- immense **re-use** of existing datasets
- biologically interesting signals may be **too poorly understood** to be analyzed in automated fashion
- much of initial analysis is **exploratory** in nature => what's my **hypothesis**?
=> searching for **unknown unknowns**
- automated algorithms often act as **black boxes** => biologists must have blind faith in bioinformatician (and bioinformatician in his/her own skills)

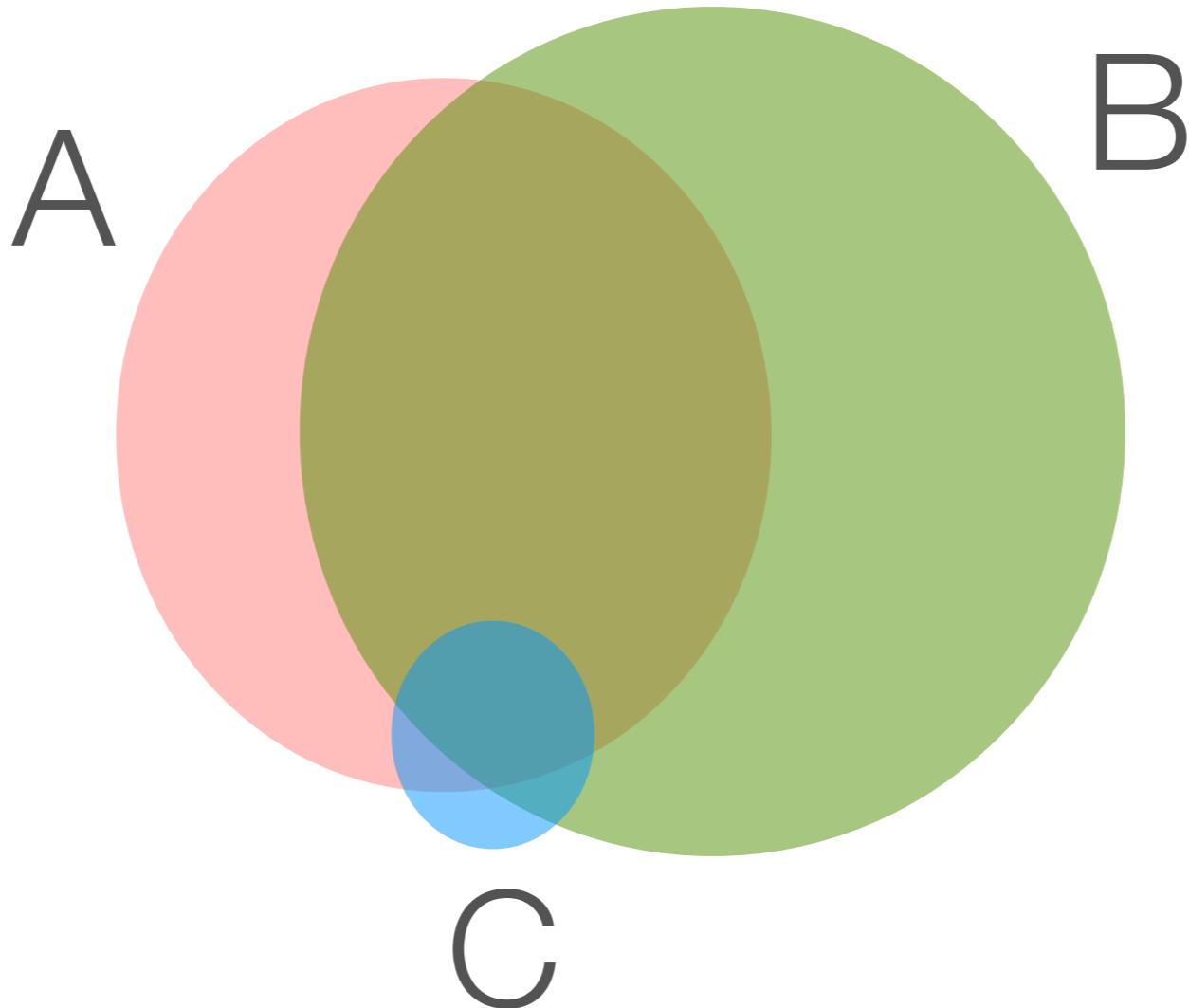
For domain expert: what's my hypothesis?

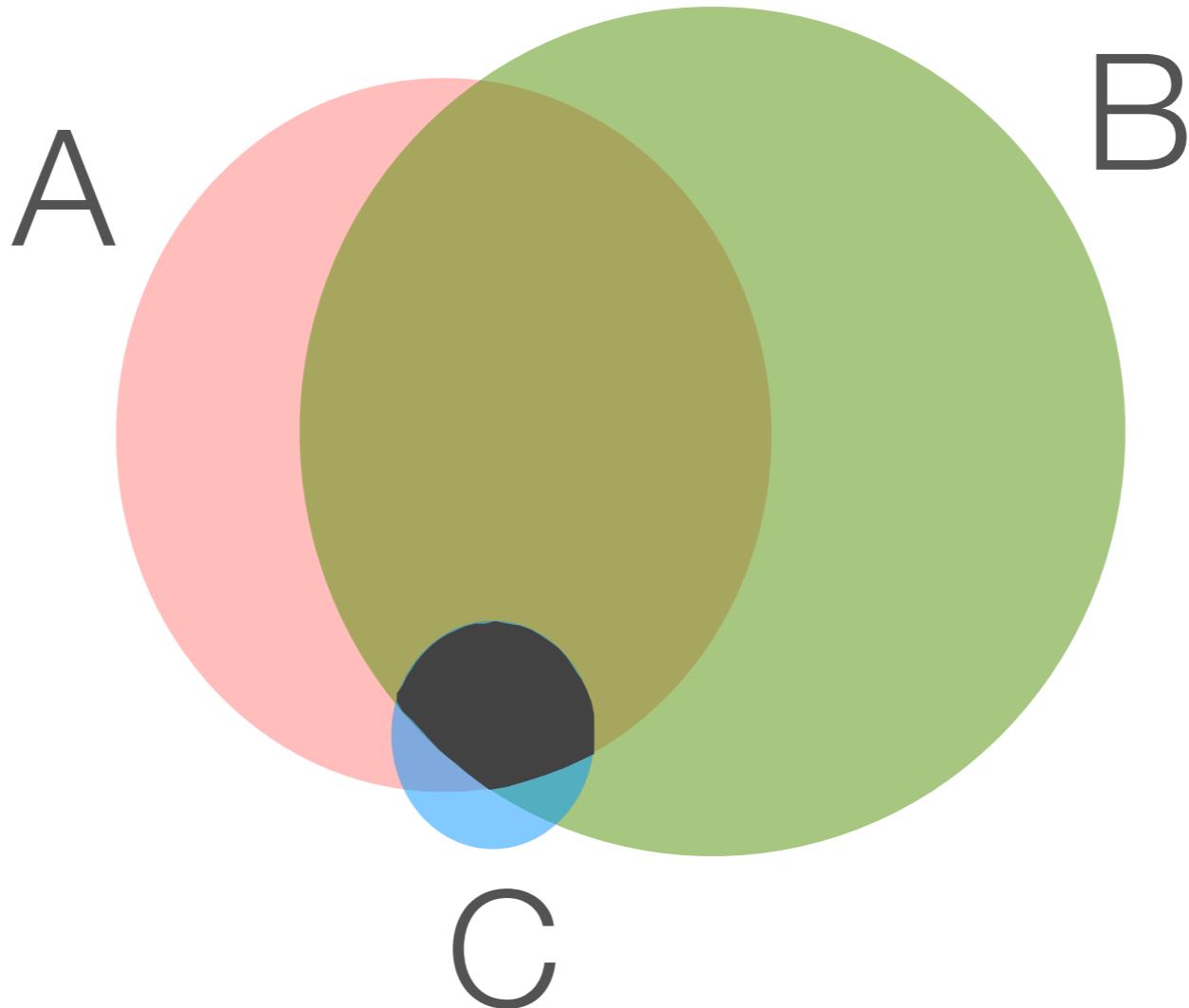


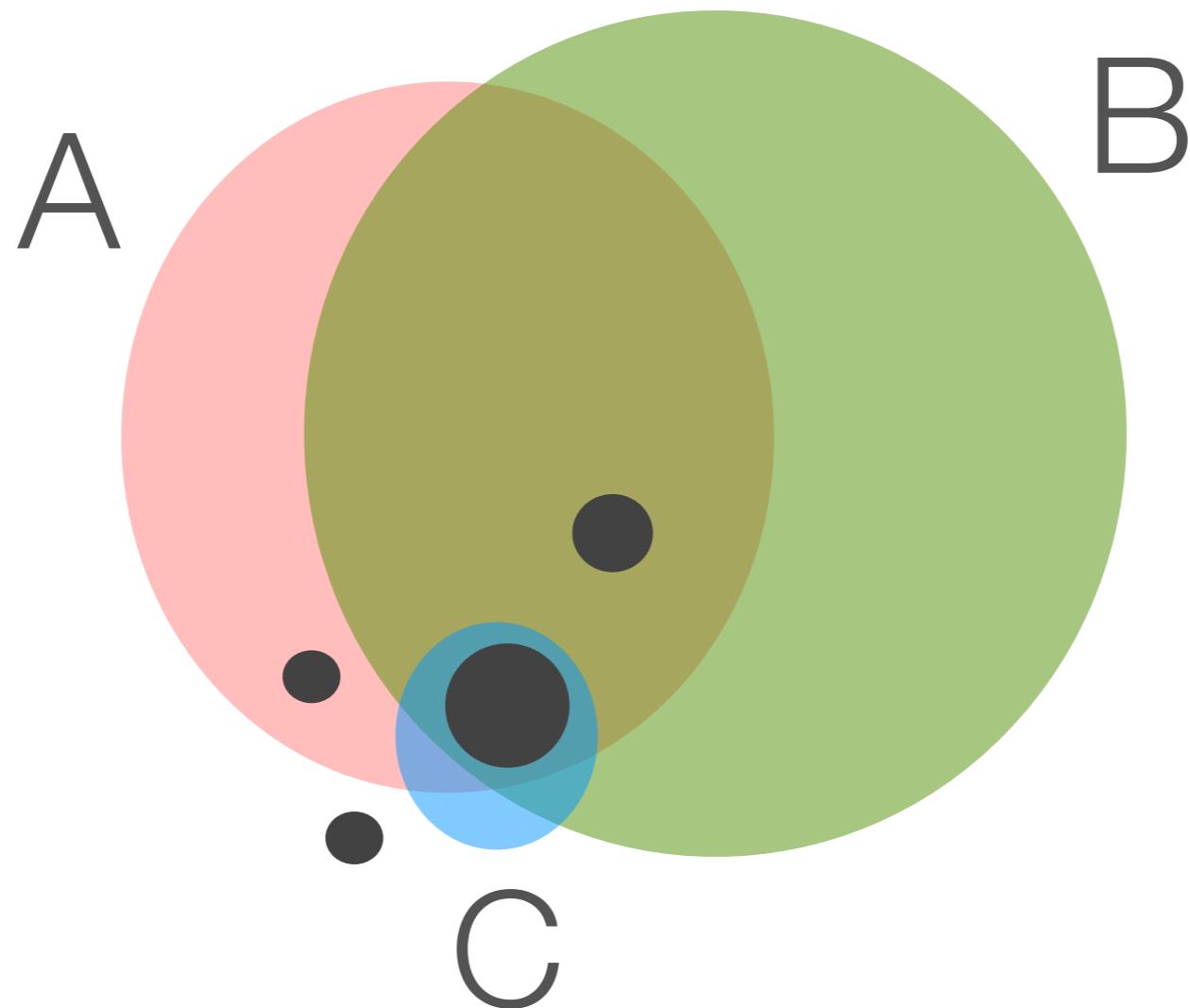
Martin Krzywinski

For developer and domain expert: opening the black box

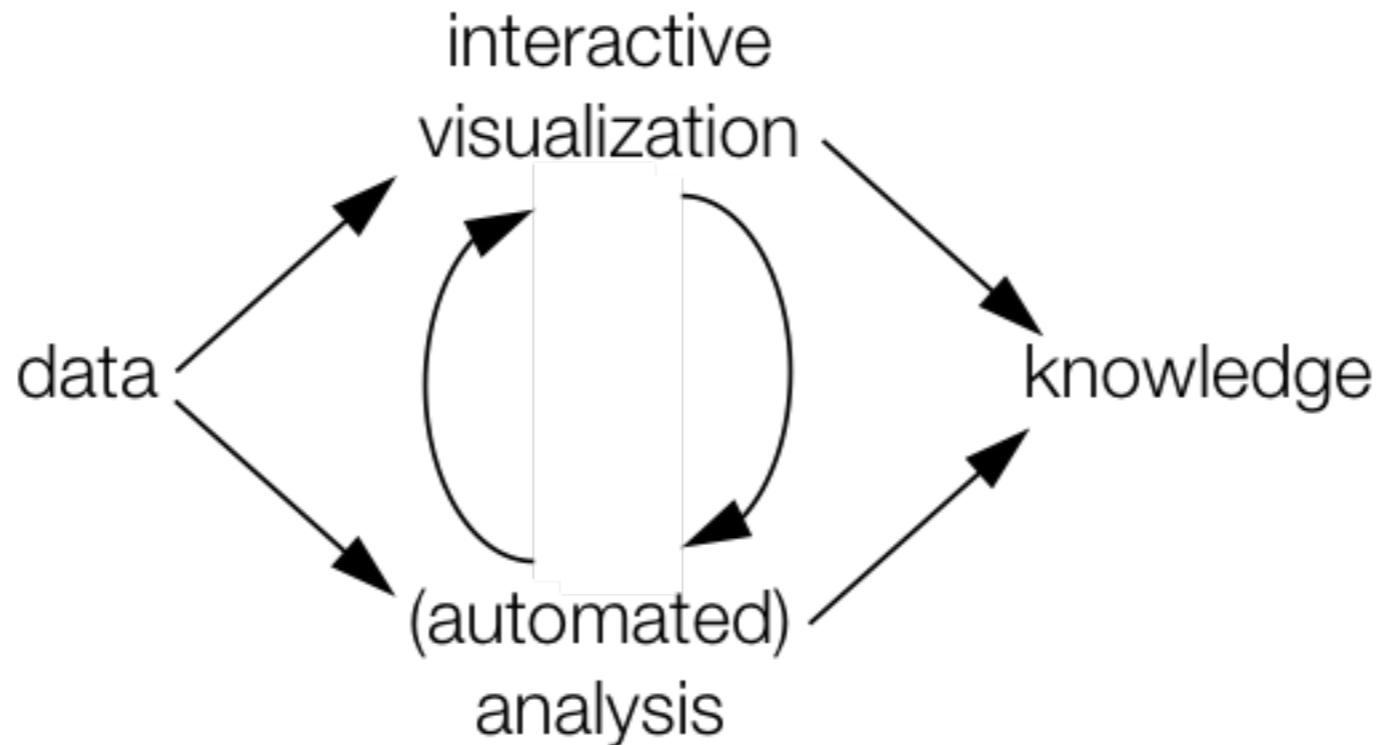








Visual Analytics to the rescue



Our research interest:
visual design + interaction design + backend

Overview of lecture

A. Why visual analytics?

B. Data visualization

- Data foundations
- Human perception foundations
- Visualization foundations and techniques

C. Visualization evaluation

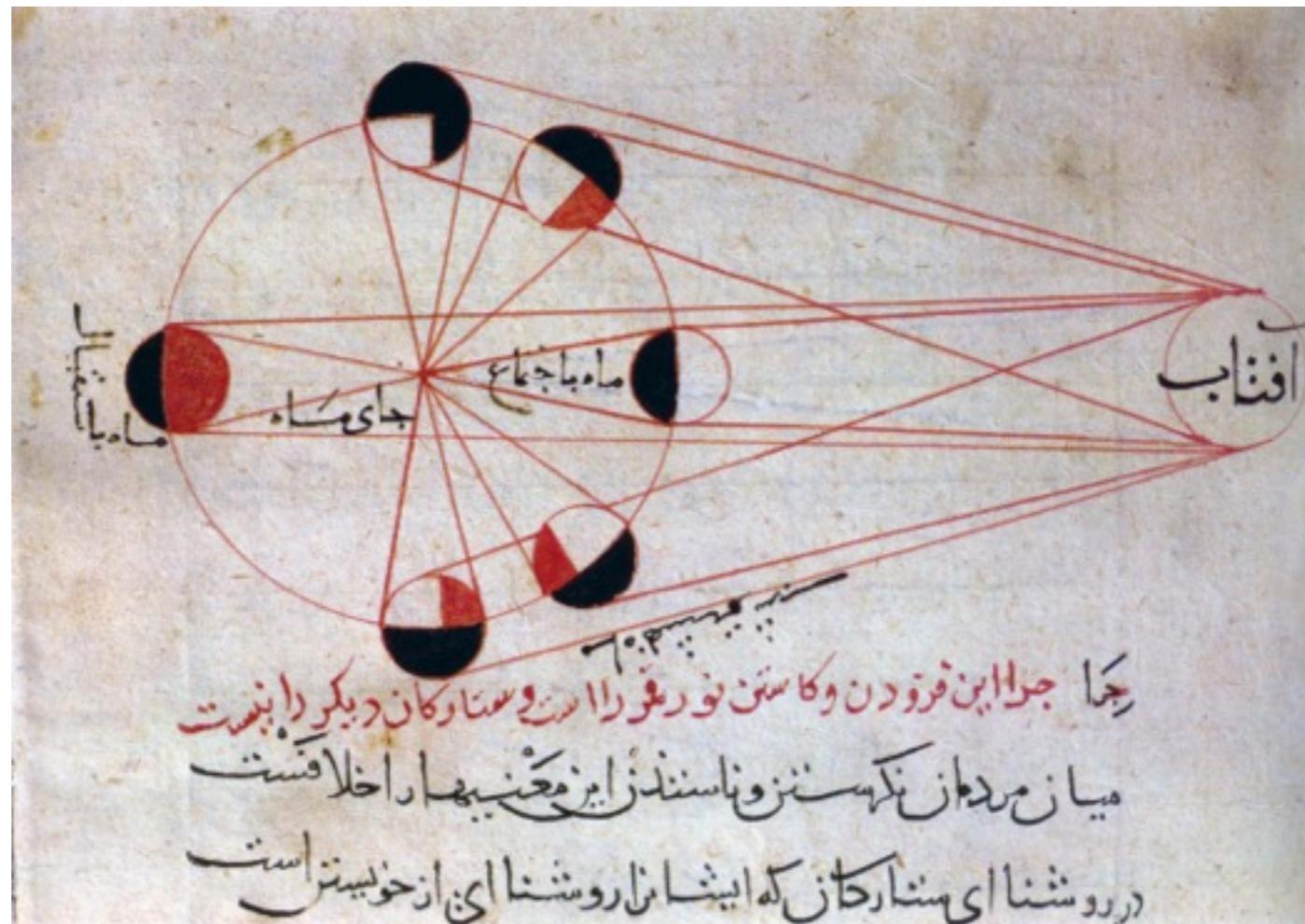
D. Tools of the trade

E. Examples

F. Exercises

B. Data Visualization

Historical perspective



Biruni - time series visualization: phases of the moon in orbit (circa 1030)



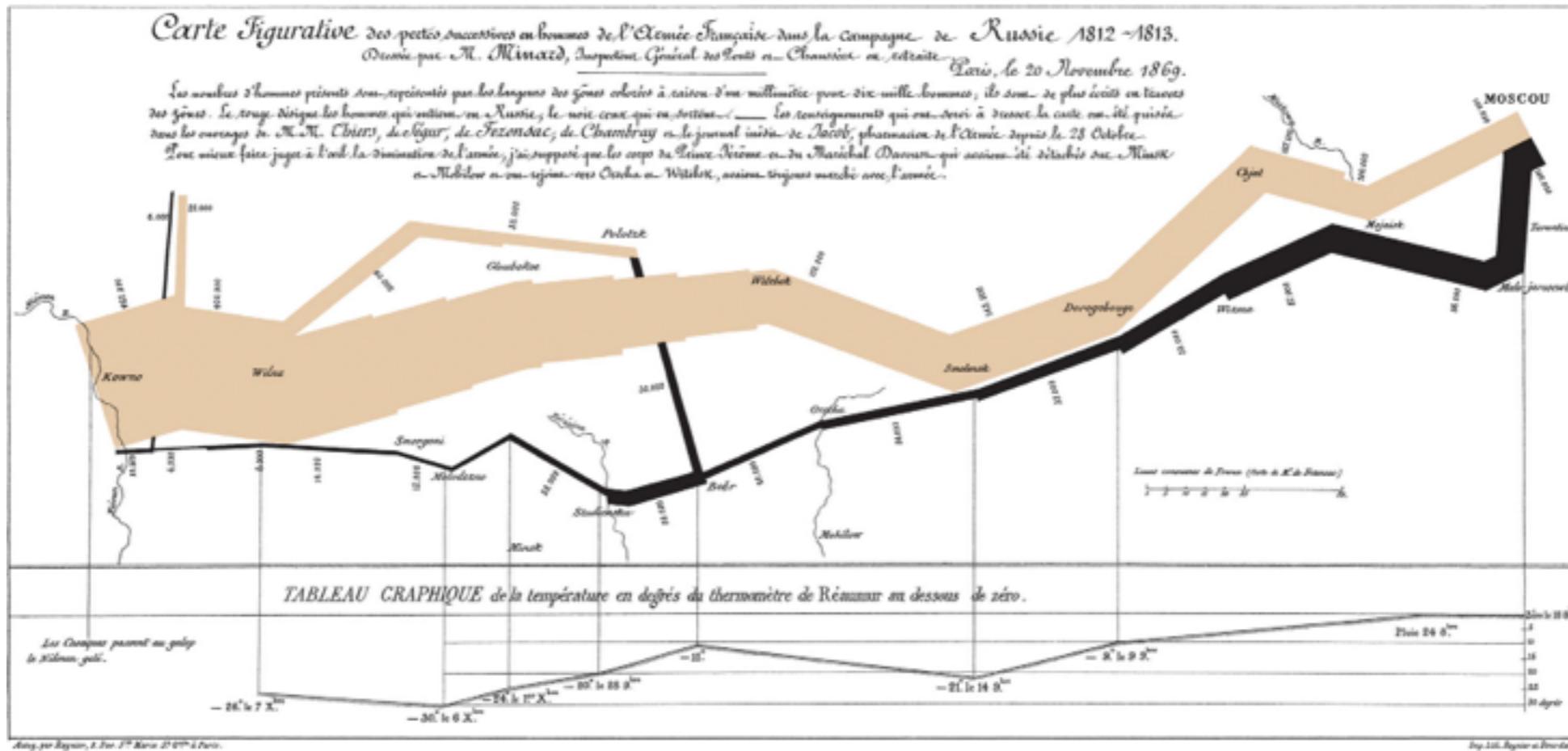
Hereford map - largest surviving map of the Middle Ages (1280s)



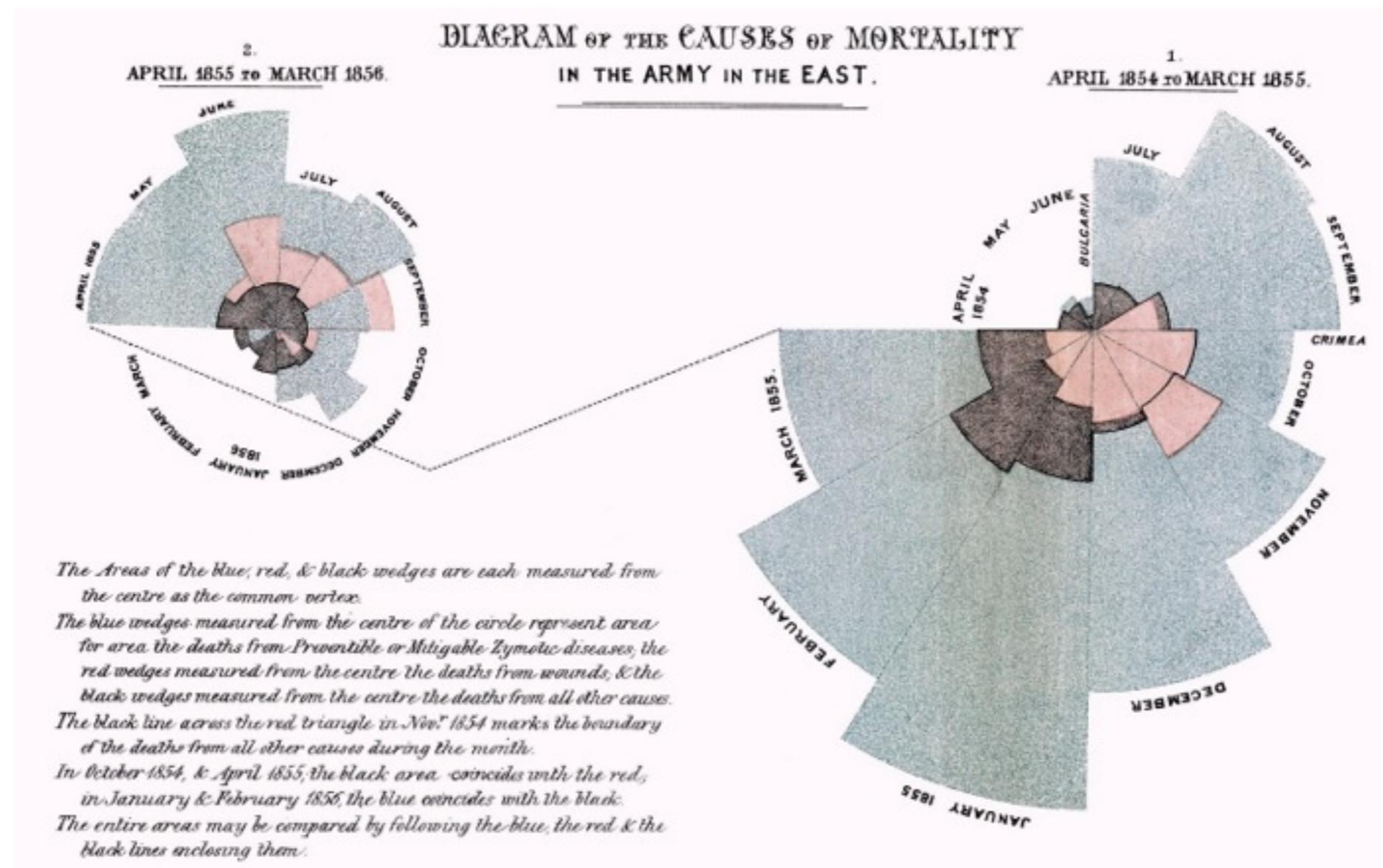
Peutinger Map - roads in Roman Empire, 1570
Abraham Ortelius (Flemish cartographer)



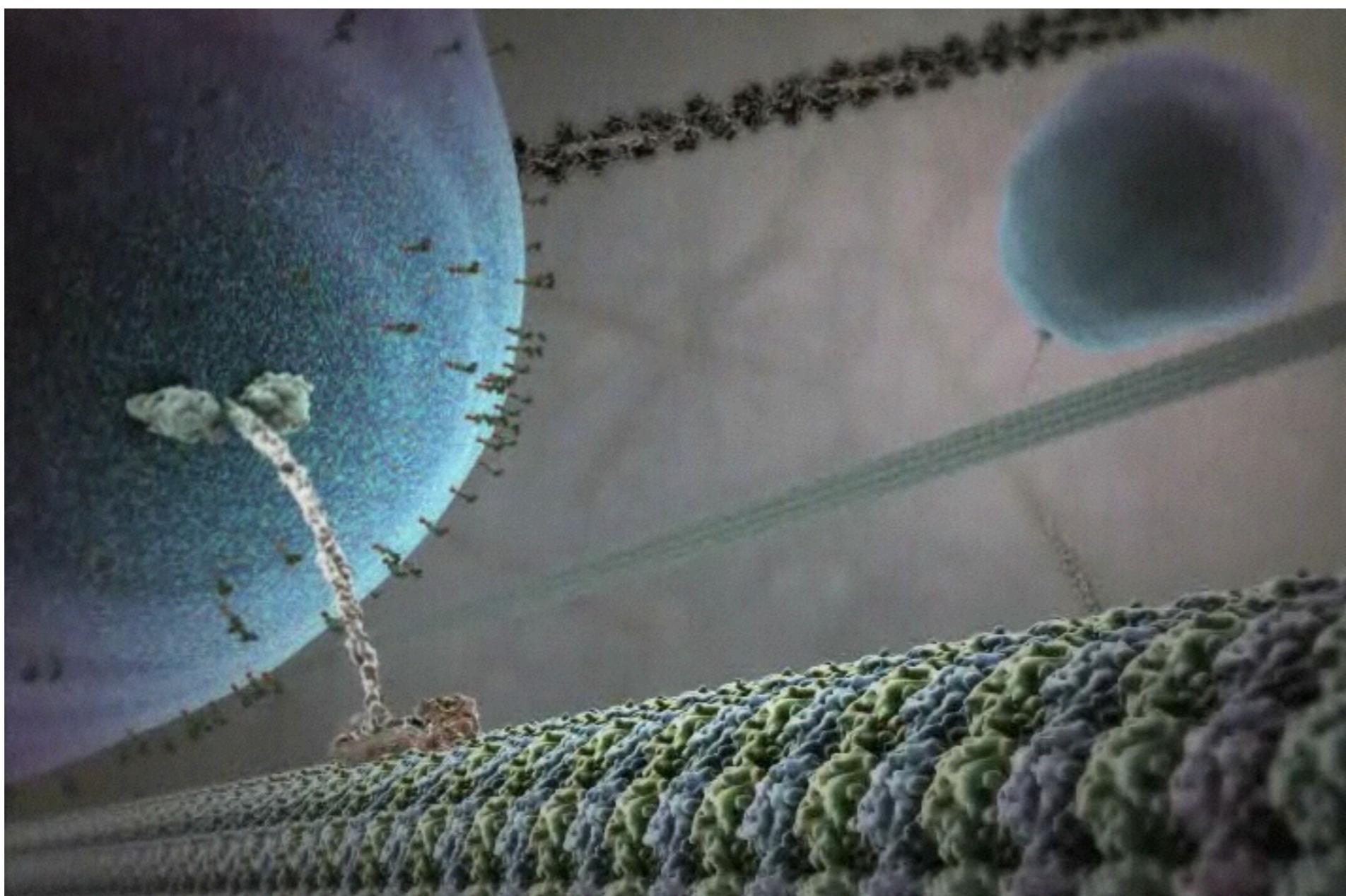
John Snow - cases of cholera in London (1854)



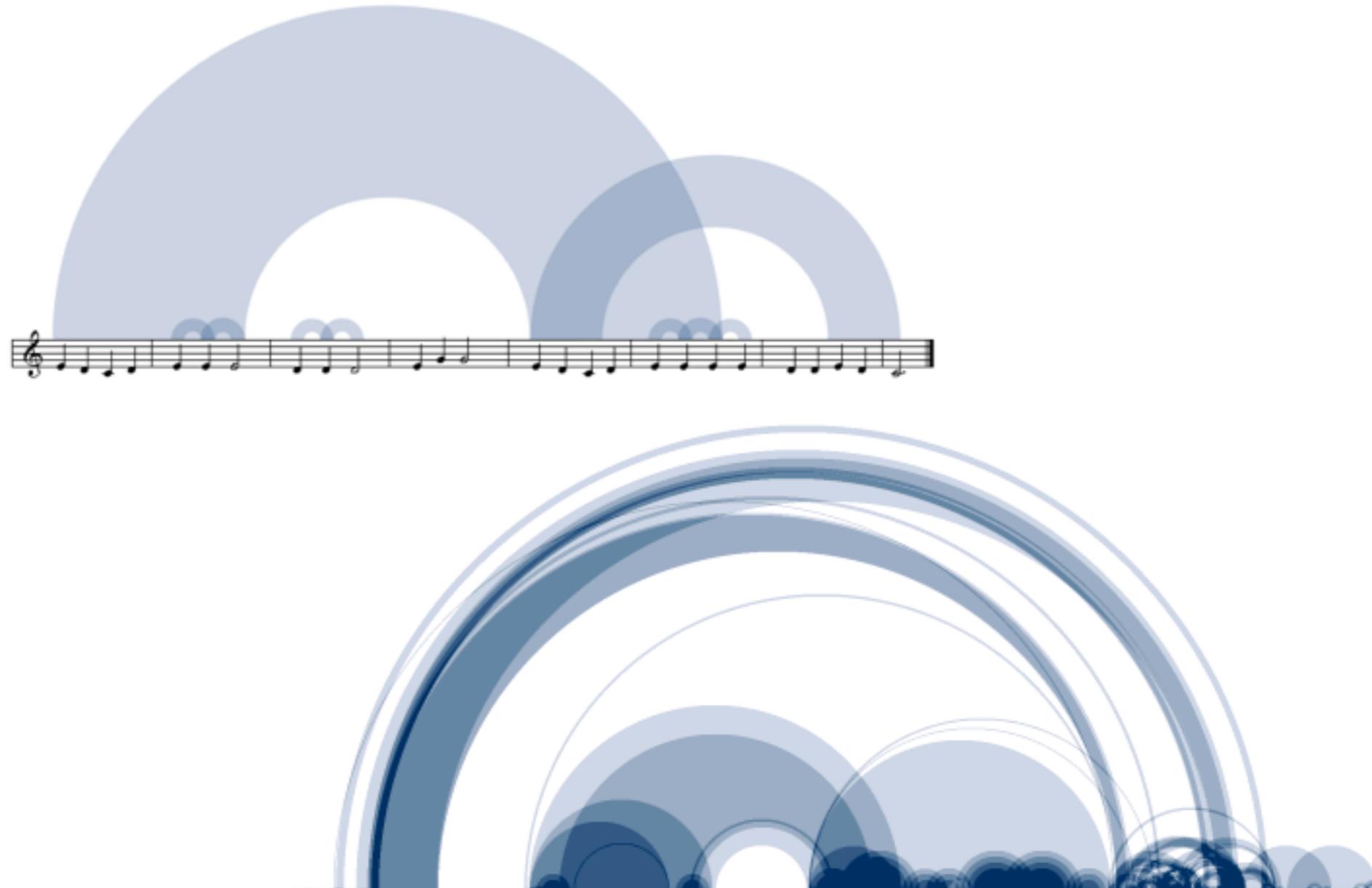
Charles Joseph Minard (1781-1870) - Napoleon's march on Moscow



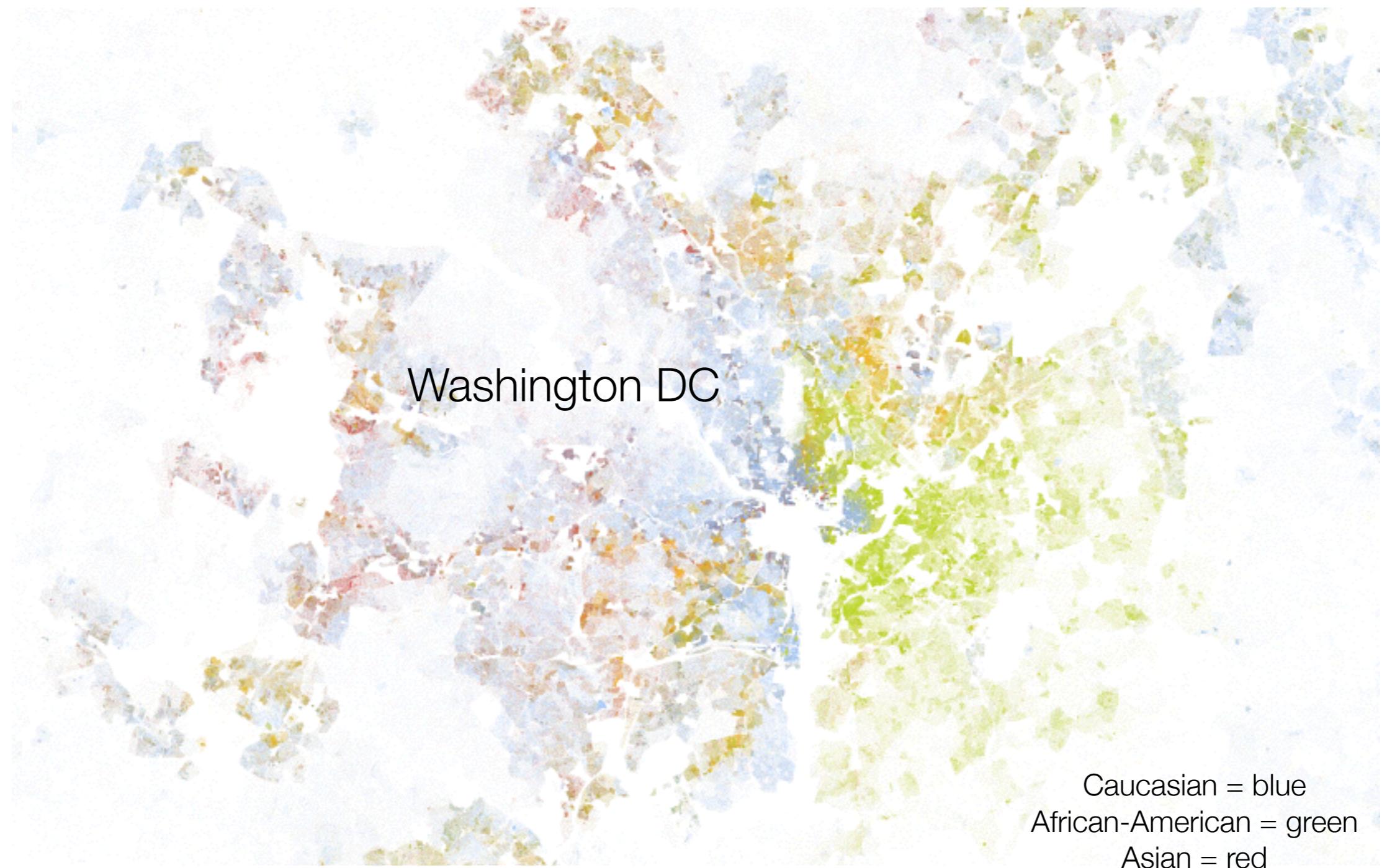
Florence Nightingale (1820-1910)
coxcomb chart monthly deaths from battle and other causes



http://multimedia.mcb.harvard.edu/anim_innerlife.html



Shape of Songs: “Like a Prayer” (Madonna)
Martin Wattenberg



Racial dot map: one dot per person in the US, coloured by ethnicity

source:<http://www.coopercenter.org/demographics/Racial-Dot-Map>



F | © 2013 National Geographic Society

What is data visualization?

perception vs cognition

human in the loop needs the details

computer-based visualization systems providing
visual representations of datasets to help people
carry out some task more effectively

intended task

measurable definitions of effectiveness

T. Munzner

perception vs cognition

human in the loop needs the details

cognition \Leftrightarrow perception
cognitive task \Rightarrow perceptive task

identify anomalies, clusters, trends

T. Munzner

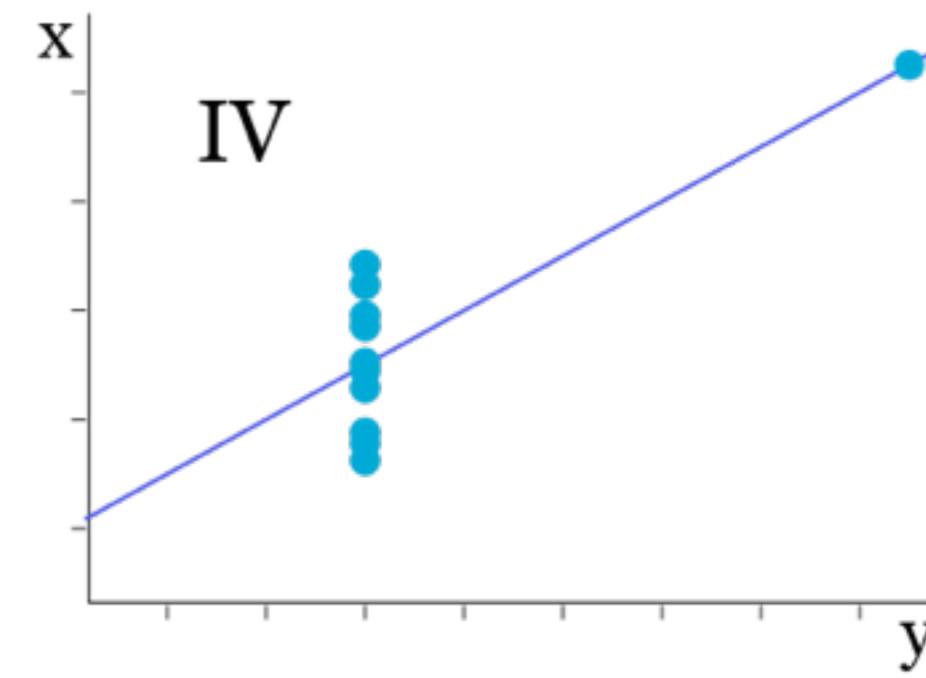
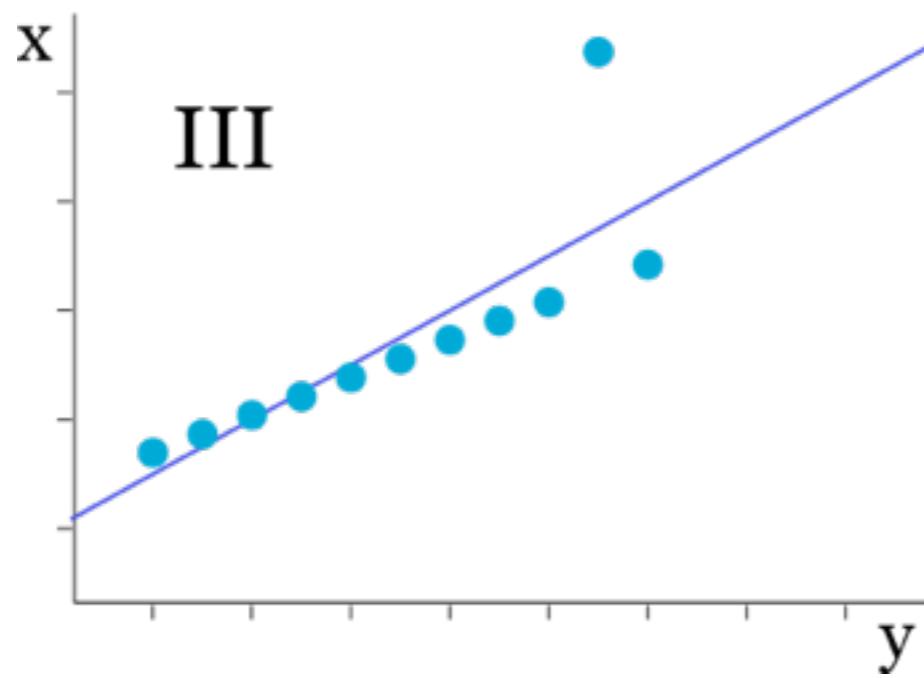
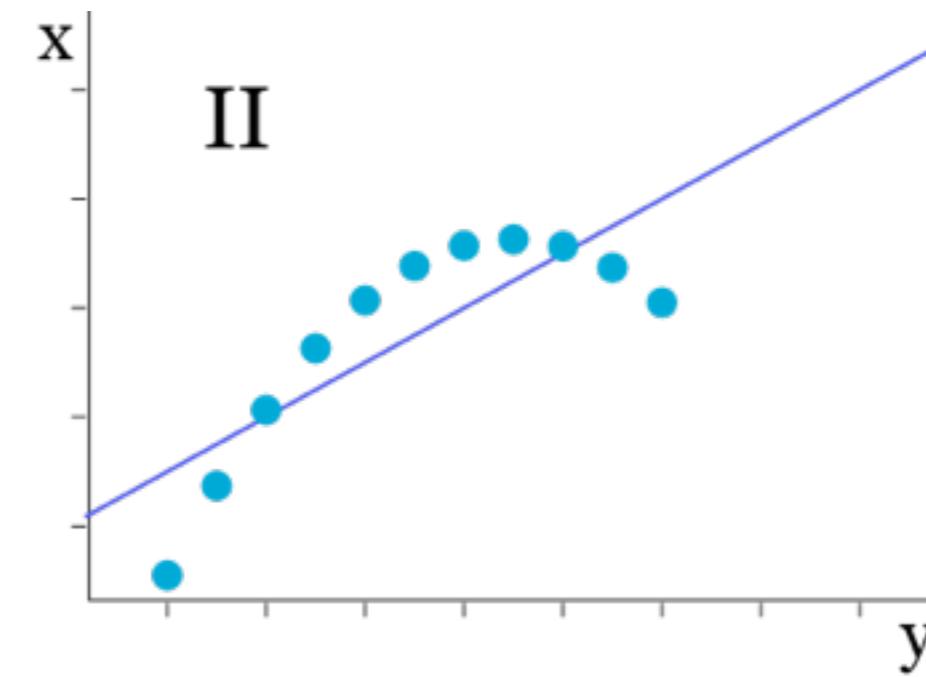
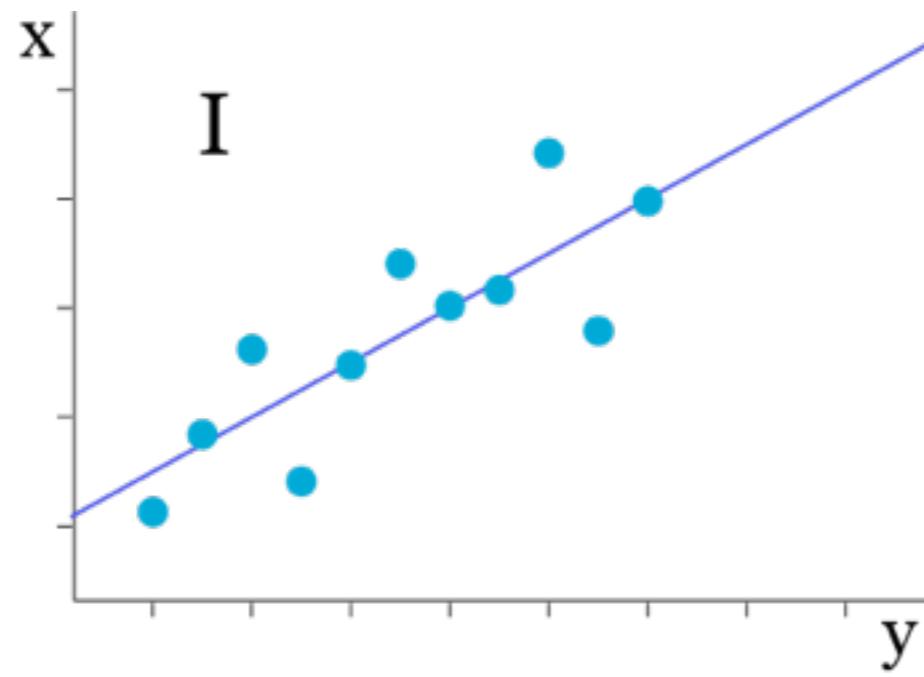
I		II		III		IV	
x	y	x	y	x	y	x	y
10.0	8.04	10.0	9.14	10.0	7.46	8.0	6.58
8.0	6.95	8.0	8.14	8.0	6.77	8.0	5.76
13.0	7.58	13.0	8.74	13.0	12.74	8.0	7.71
9.0	8.81	9.0	8.77	9.0	7.11	8.0	8.84
11.0	8.33	11.0	9.26	11.0	7.81	8.0	8.47
14.0	9.96	14.0	8.10	14.0	8.84	8.0	7.04
6.0	7.24	6.0	6.13	6.0	6.08	8.0	5.25
4.0	4.26	4.0	3.10	4.0	5.39	19.0	12.50
12.0	10.84	12.0	9.13	12.0	8.15	8.0	5.56
7.0	4.82	7.0	7.26	7.0	6.42	8.0	7.91
5.0	5.68	5.0	4.74	5.0	5.73	8.0	6.80

n = 11

mean x = 9.0
mean y = 7.5

variance x = 11.0
variance y = 4.12

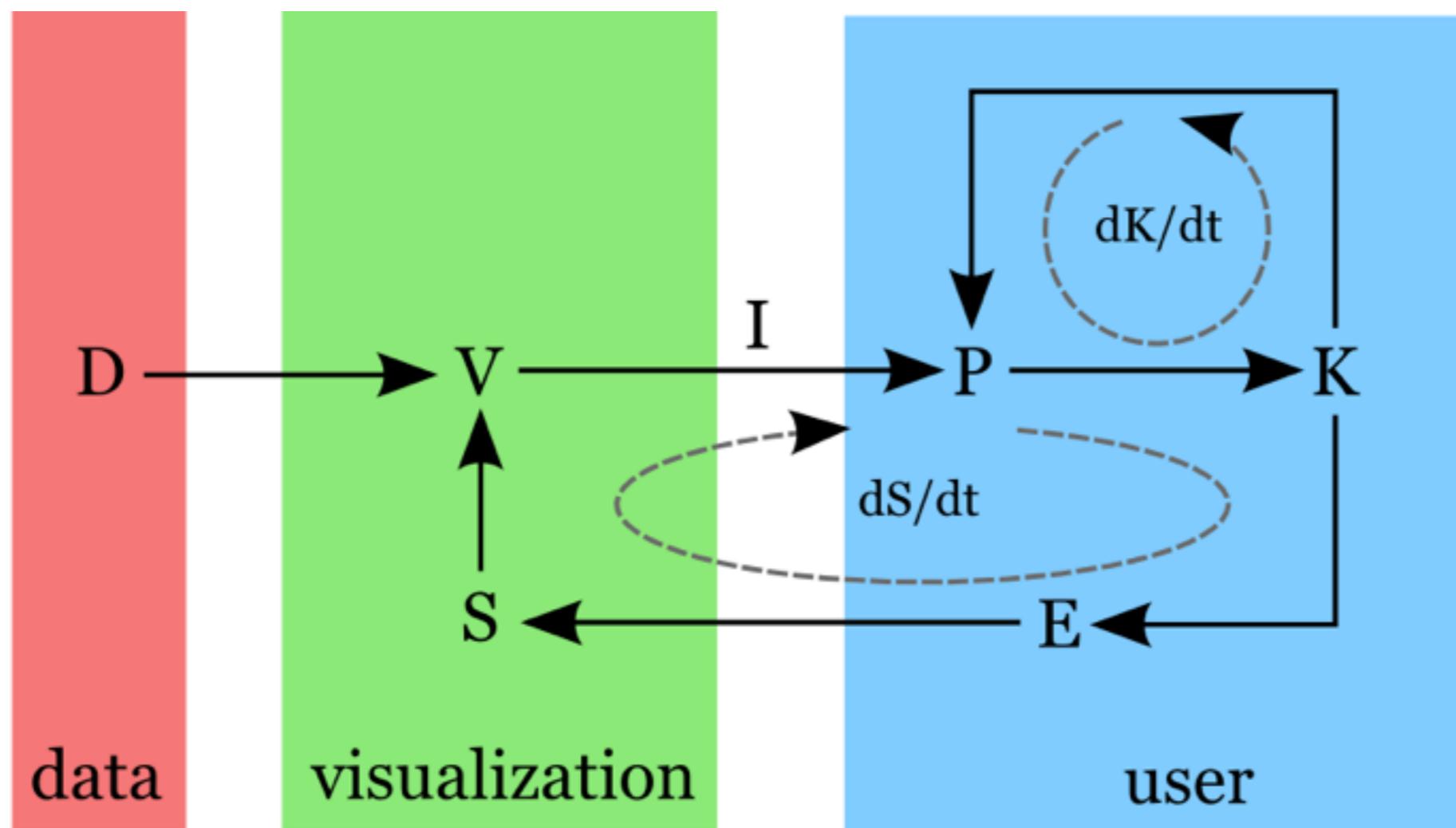
correlation x & y = 0.816
regression line: y = 3+0.5x



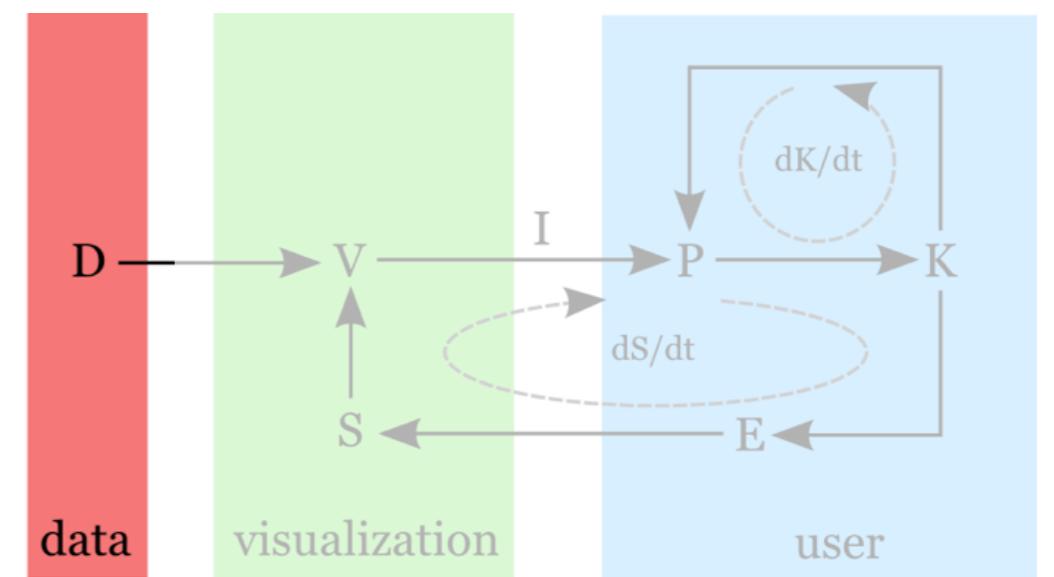
Misunderstandings visual design

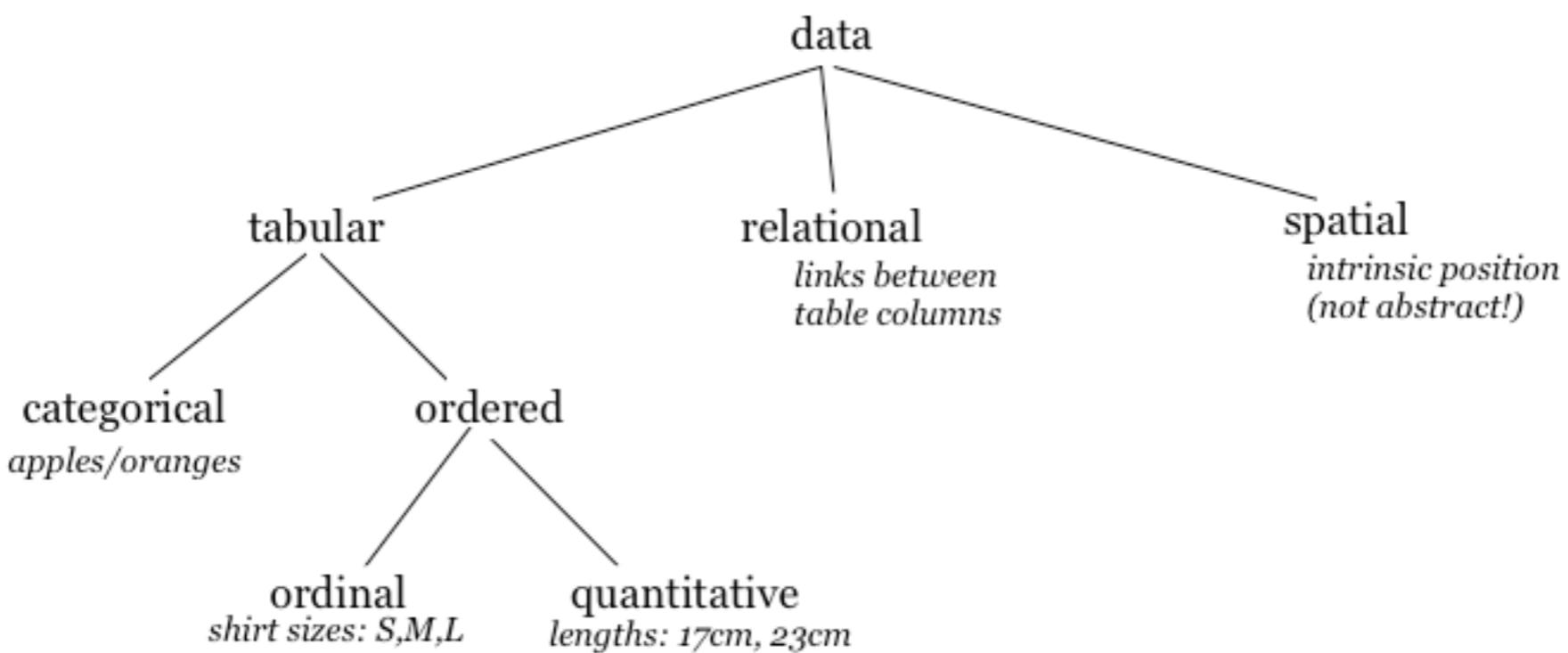
- Is it a matter of talent? creativity? inspiration? aesthetics? taste?
- practice => anyone can learn (to some extent)
- be rational
- aim at usability => beauty/aesthetics often follows

Components



Data foundations



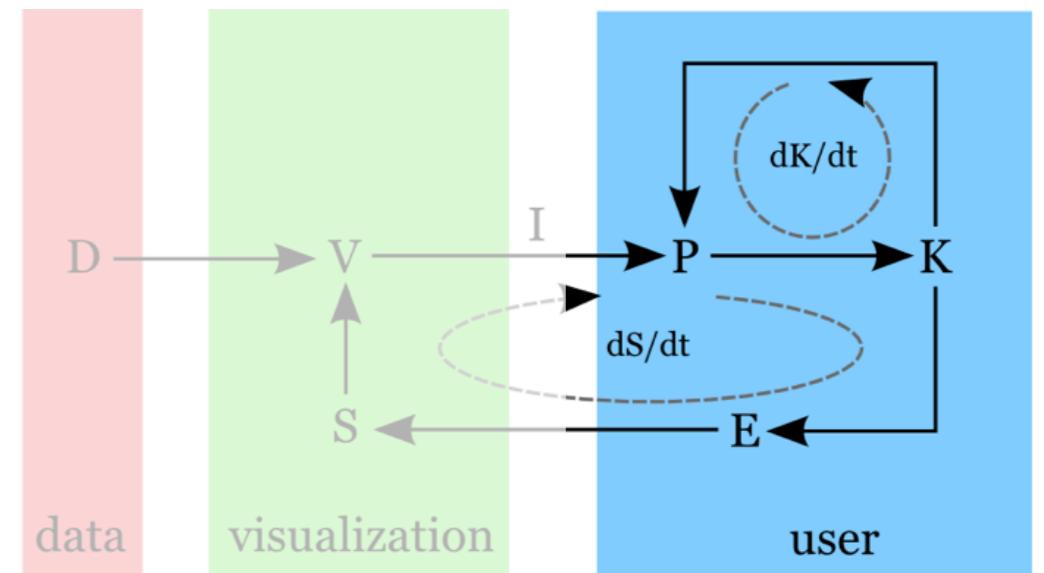


S Stevens “On the theory of scales and measurements” (1946)

Data type taxonomy (?) (Schneiderman, 2006):

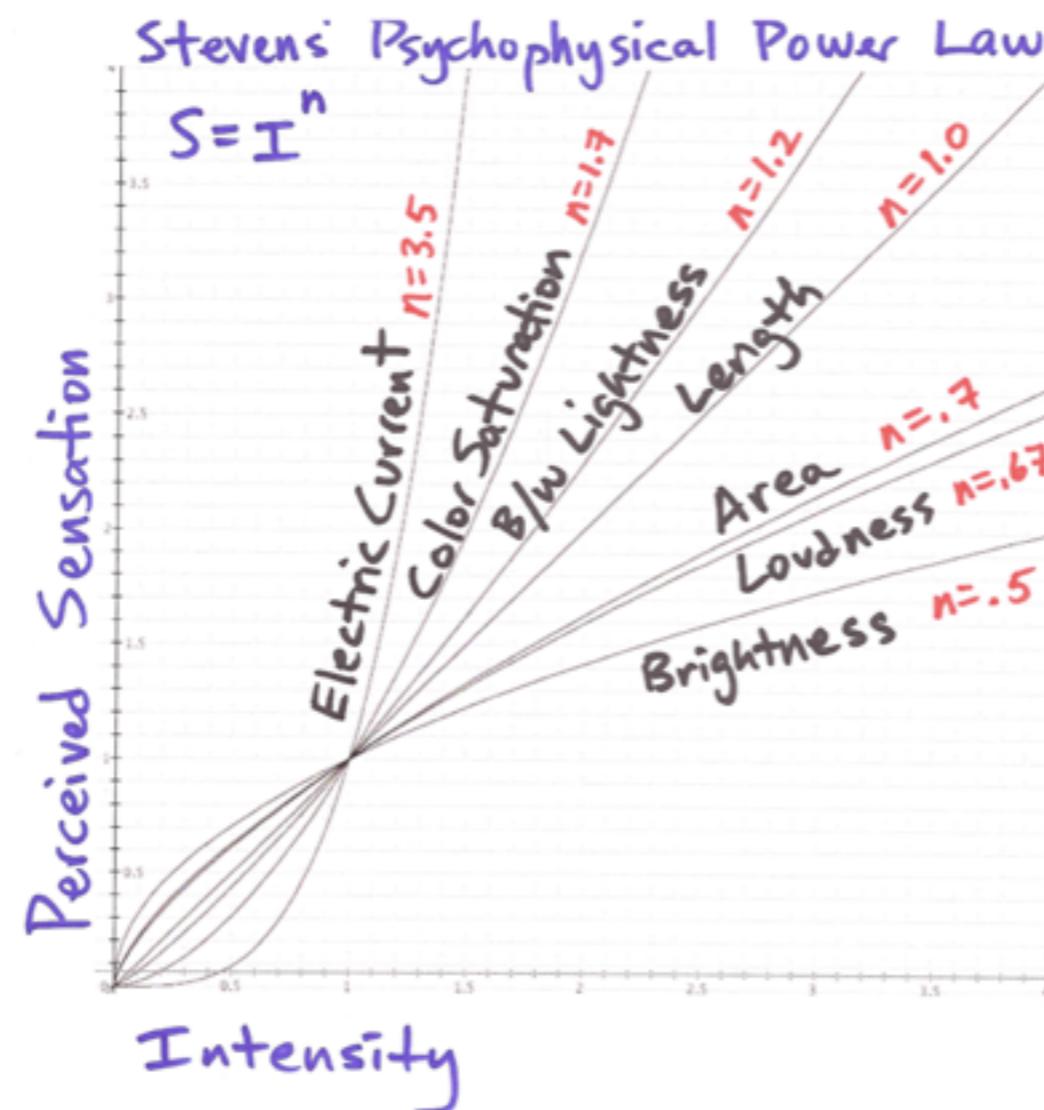
- 1D (e.g. DNA sequences)
- temporal (e.g. time series gene expression)
- 2D (e.g. distribution maps)
- 3D (e.g. anatomical structures)
- nD (e.g. Fisher's iris dataset; what if 1,000s of dimensions?)
- trees (e.g. phylogenies)
- networks/graphs (e.g. metabolic pathways)
- text & documents (e.g. publications)

Human perception foundations



Steven's psychophysical law

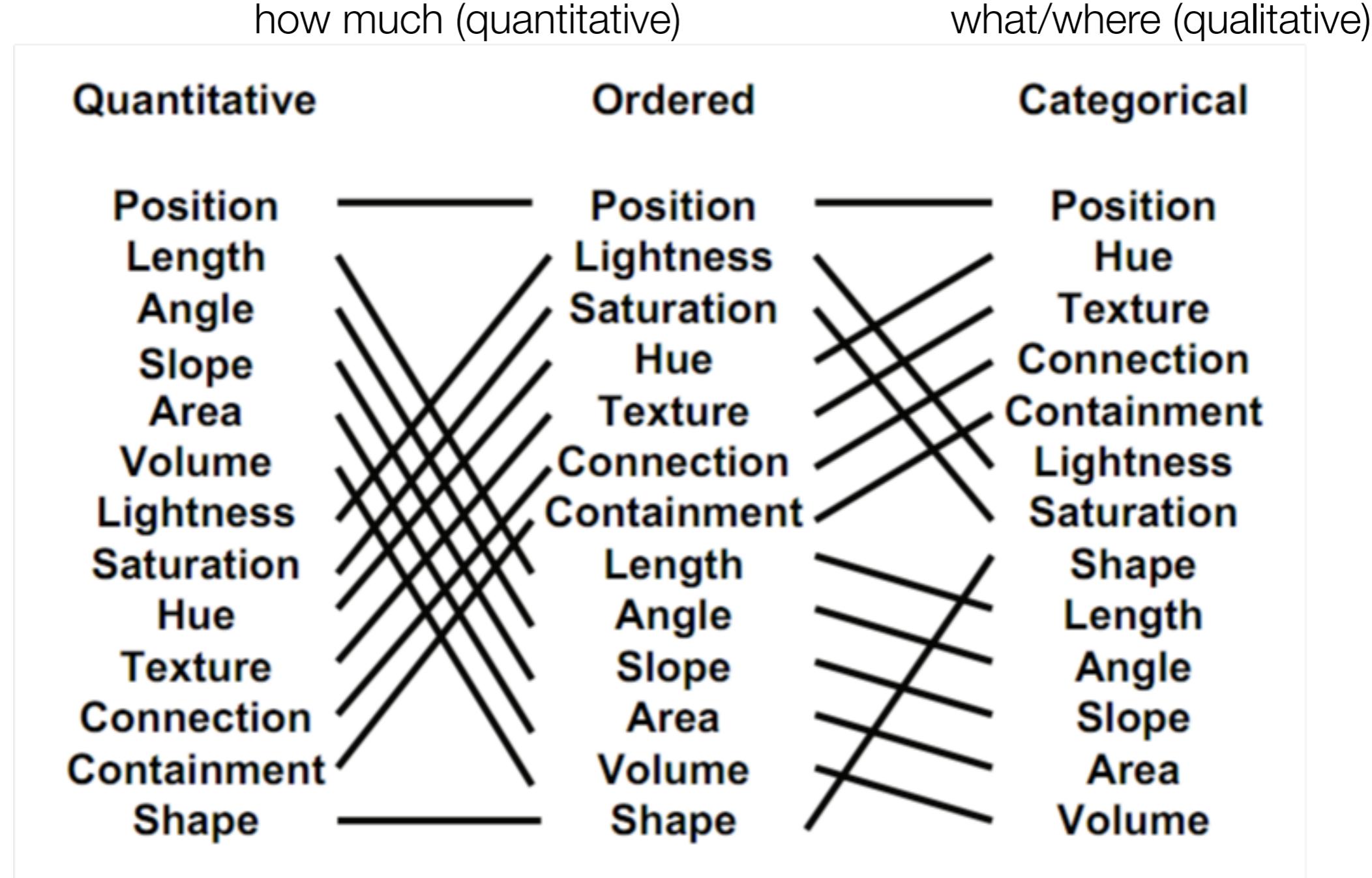
= proposed relationship between the magnitude of a physical stimulus and its perceived intensity or strength



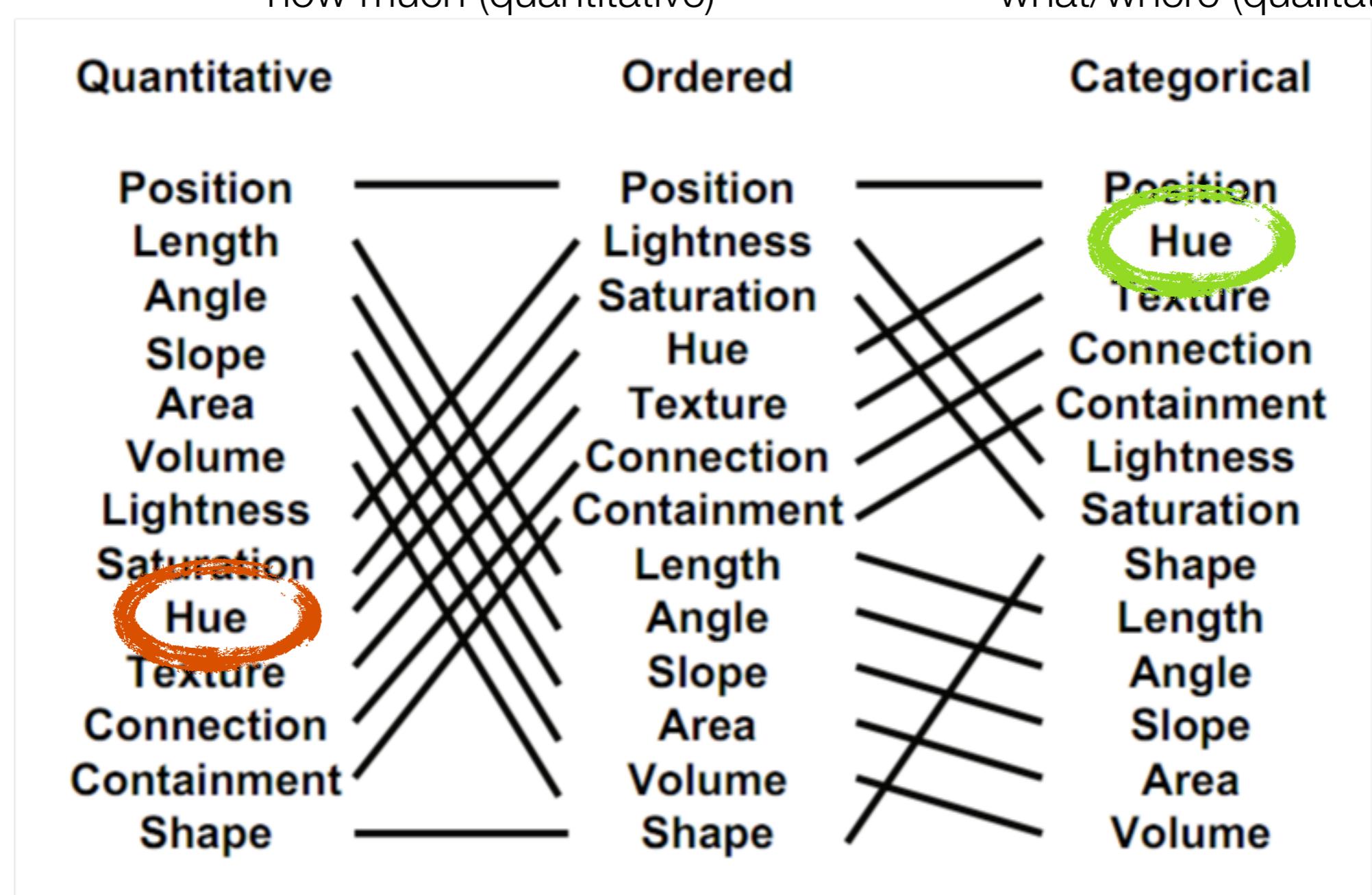
Semiology of graphics

	Points	Lines	Areas	Best to show
Shape		possible, but too weird to show	cartogram	qualitative differences
Size			cartogram	quantitative differences
Color Hue				qualitative differences
Color Value				quantitative differences
Color Intensity				qualitative differences
Texture				qualitative & quantitative differences

Accuracy of quantitative perceptual tasks

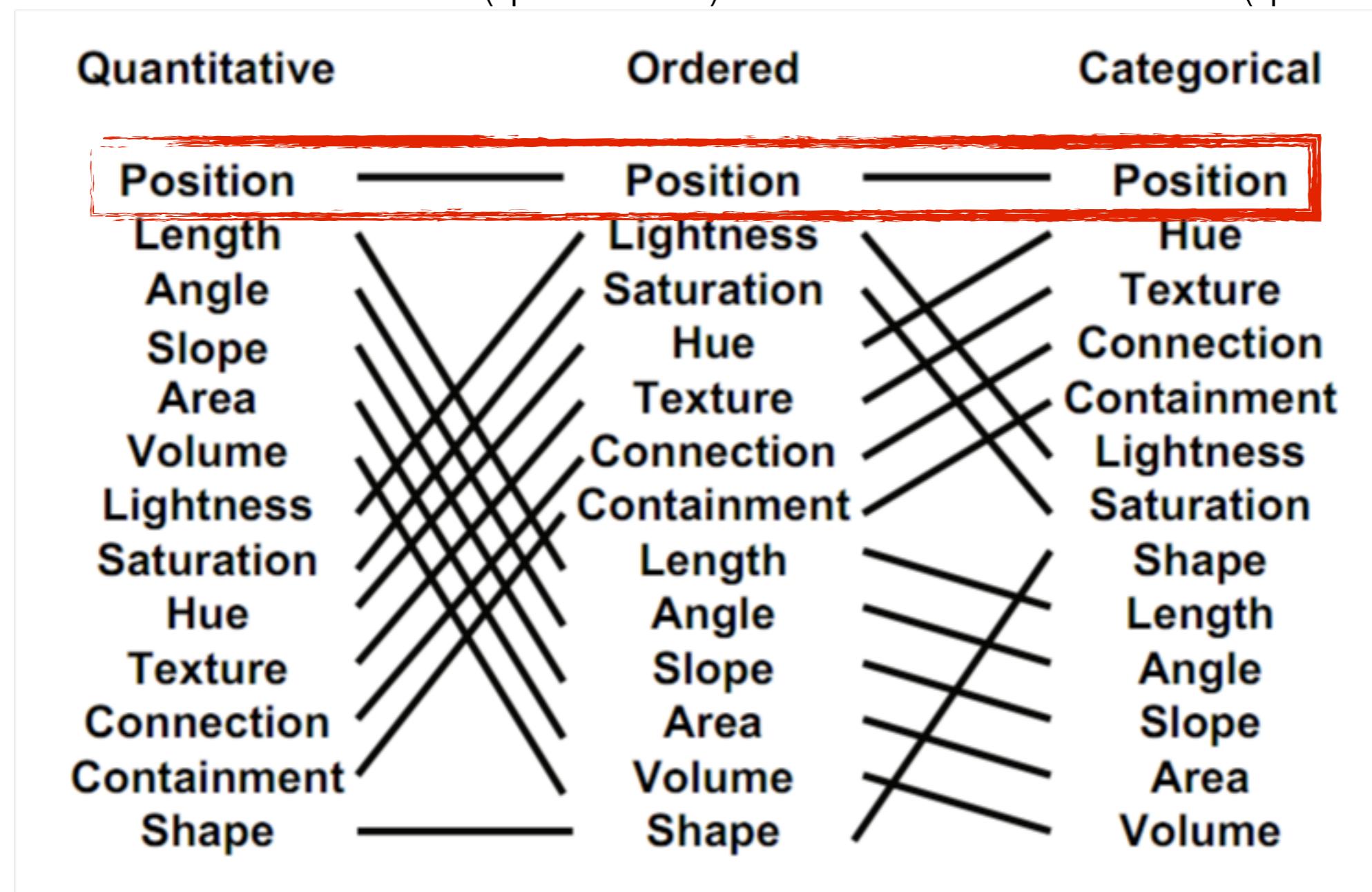


Accuracy of quantitative perceptual tasks



McKinlay

Accuracy of quantitative perceptual tasks

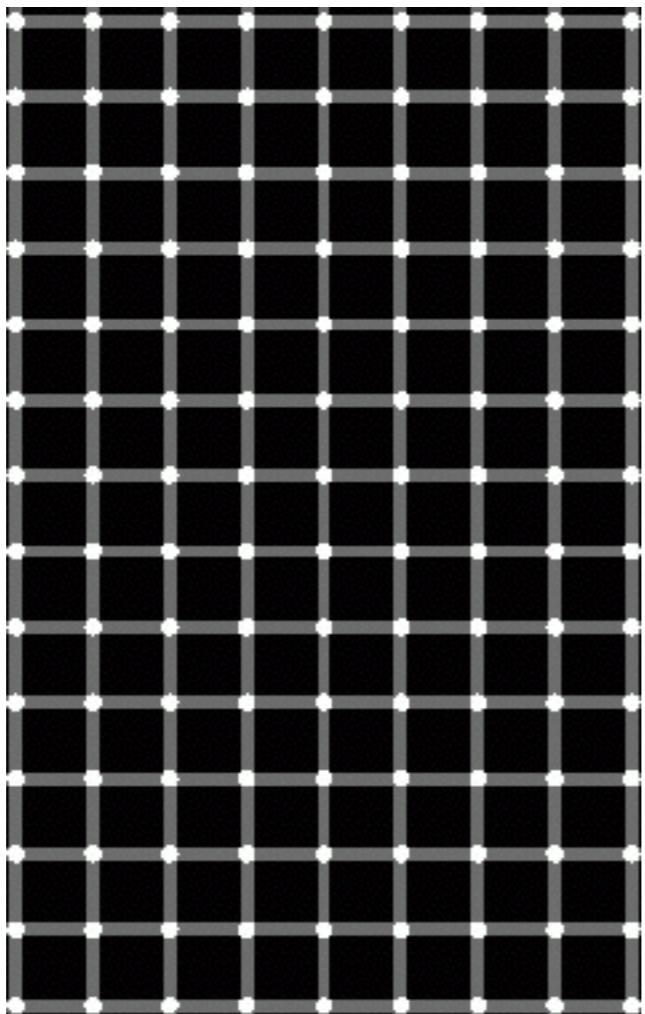


“power of the plane”

McKinlay

Gestalt laws - interplay between parts and the whole (Kurt Koffka)

series of principles



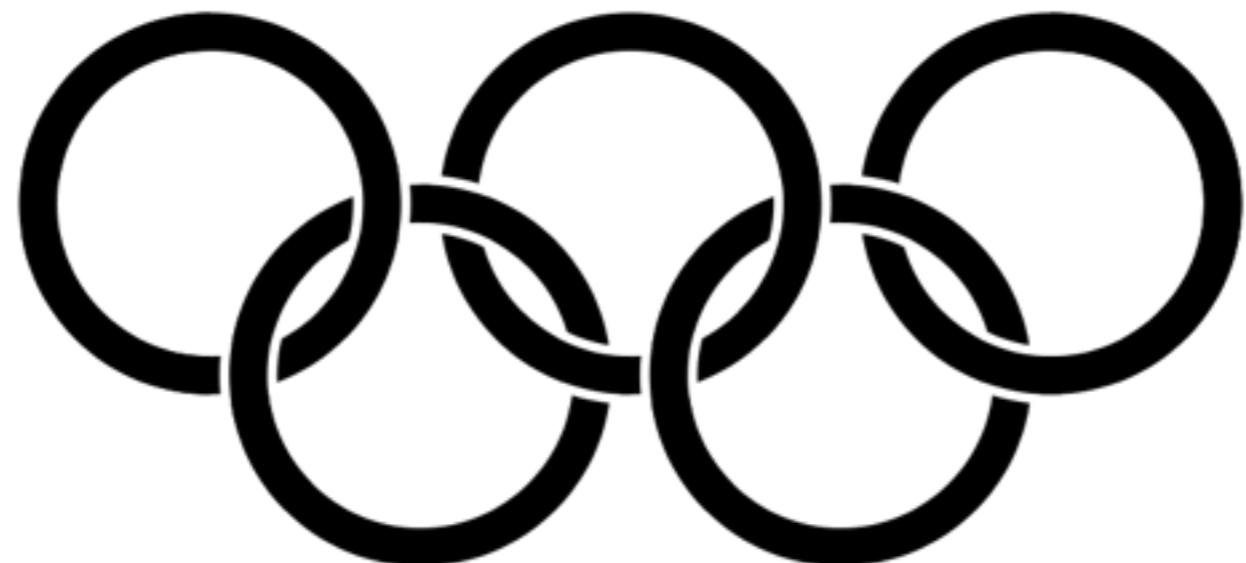
Election results Florida:

- black = Bush
- white = Gore

Hermann grid illusion

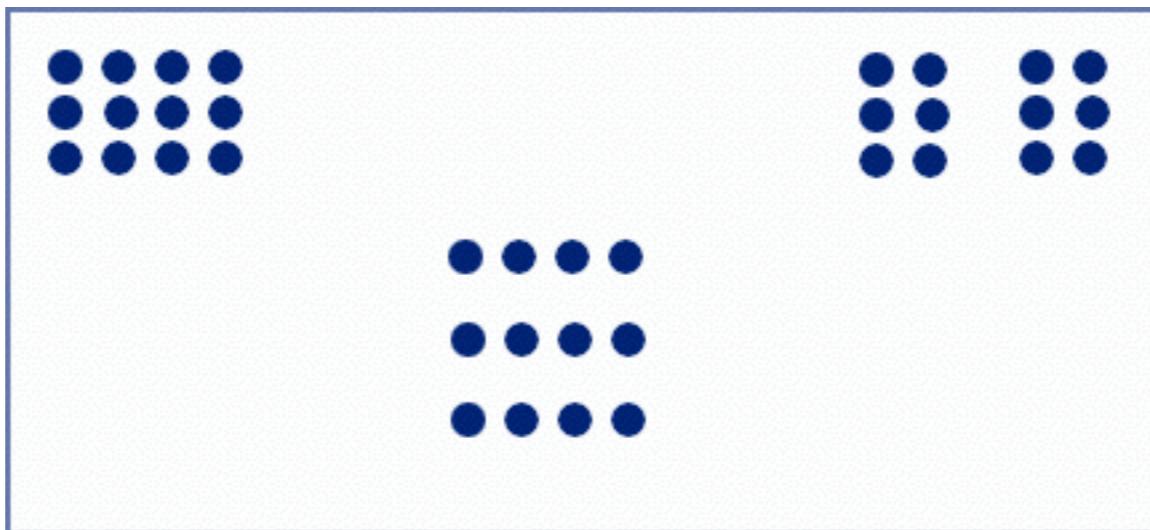
Gestalt - Principle of Simplicity

Every pattern we see is seen such that we see a structure that is as simple as possible.



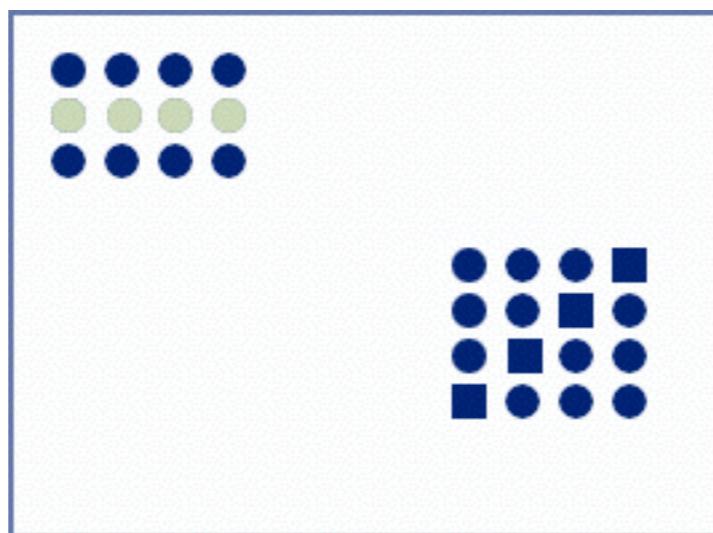
Gestalt - Principle of Proximity

Things that are close to each other are seen as belonging together (=> clusters)



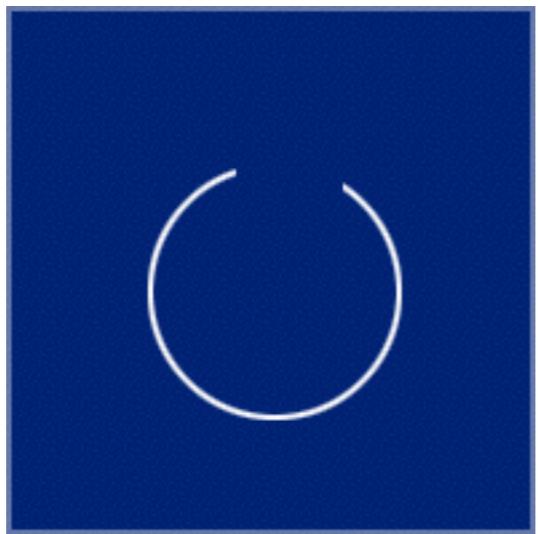
Gestalt - Principle of Similarity

Things that are similar in some way are perceived as belonging together.



Gestalt - Principle of Closure

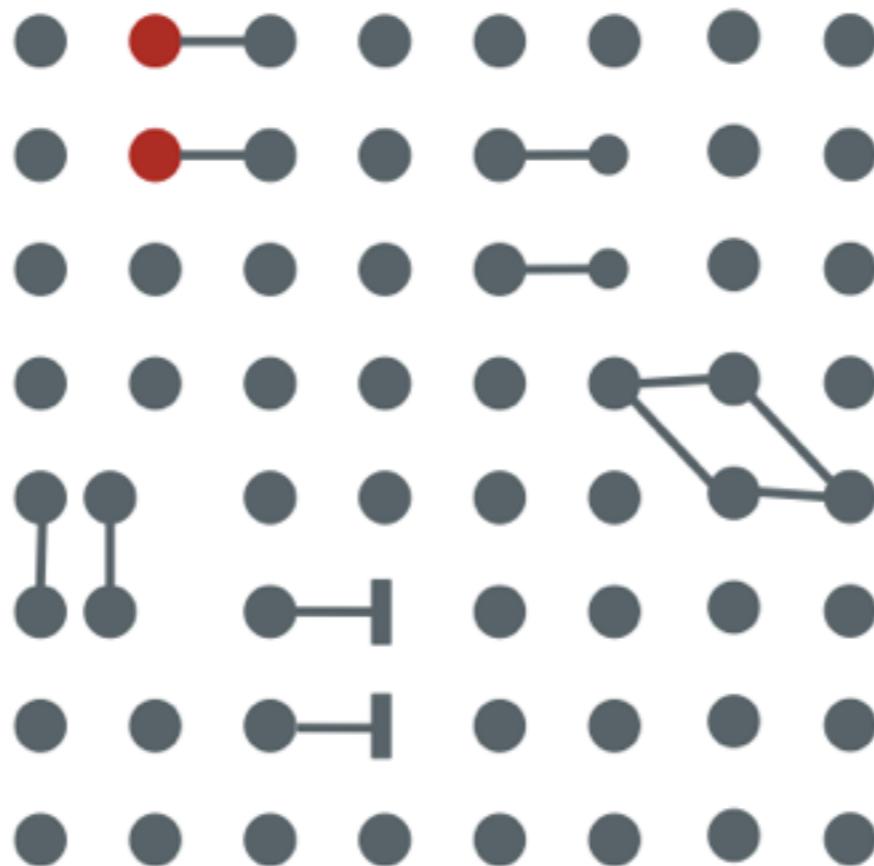
You will try to complete a pattern.



Kanizsa illusion

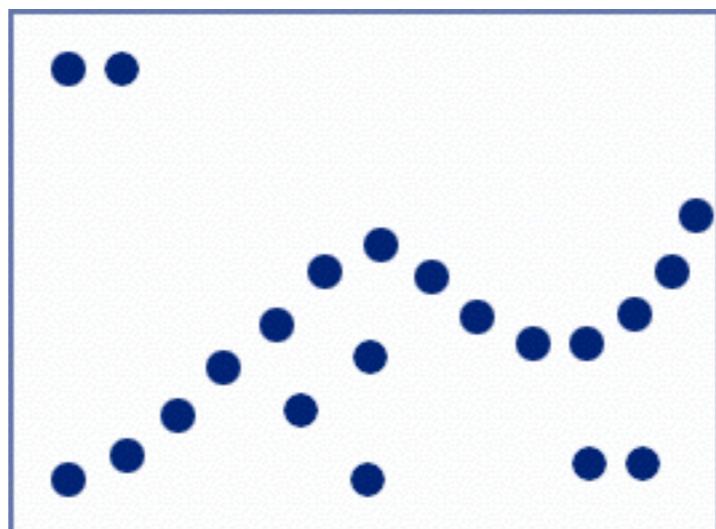
Gestalt - Principle of Connectedness

Things that are connected are perceived as belonging together. This encoding is stronger than similarity, shape, colour, and size.



Gestalt - Principle of Good Continuation

Objects that are arranged in a straight or smooth line tend to be seen as a unit.

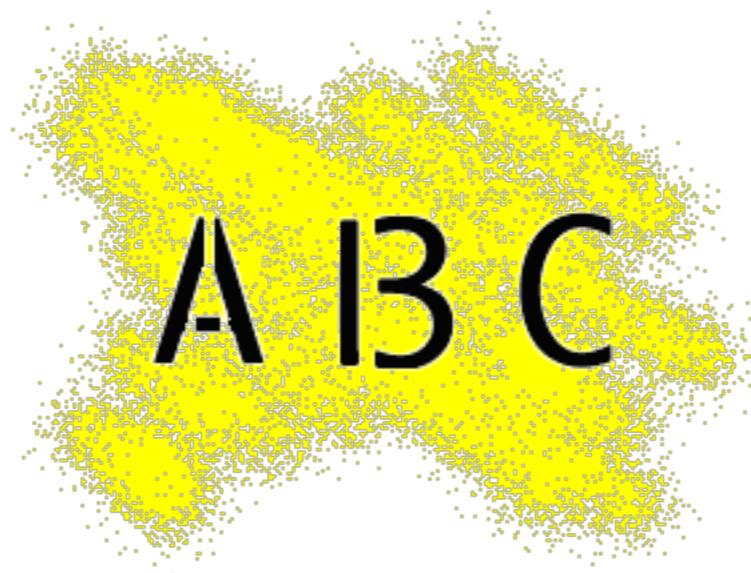


Gestalt - Principle of Common Fate

Objects that move in the same direction tend to be seen as a unit.



Gestalt - Principle of Familiarity



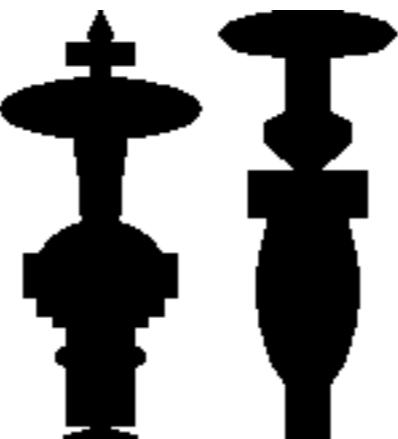
13

I2 I3 I4



Gestalt - Principle of Symmetry

Symmetrical areas tend to be seen as figures against asymmetrical backgrounds.



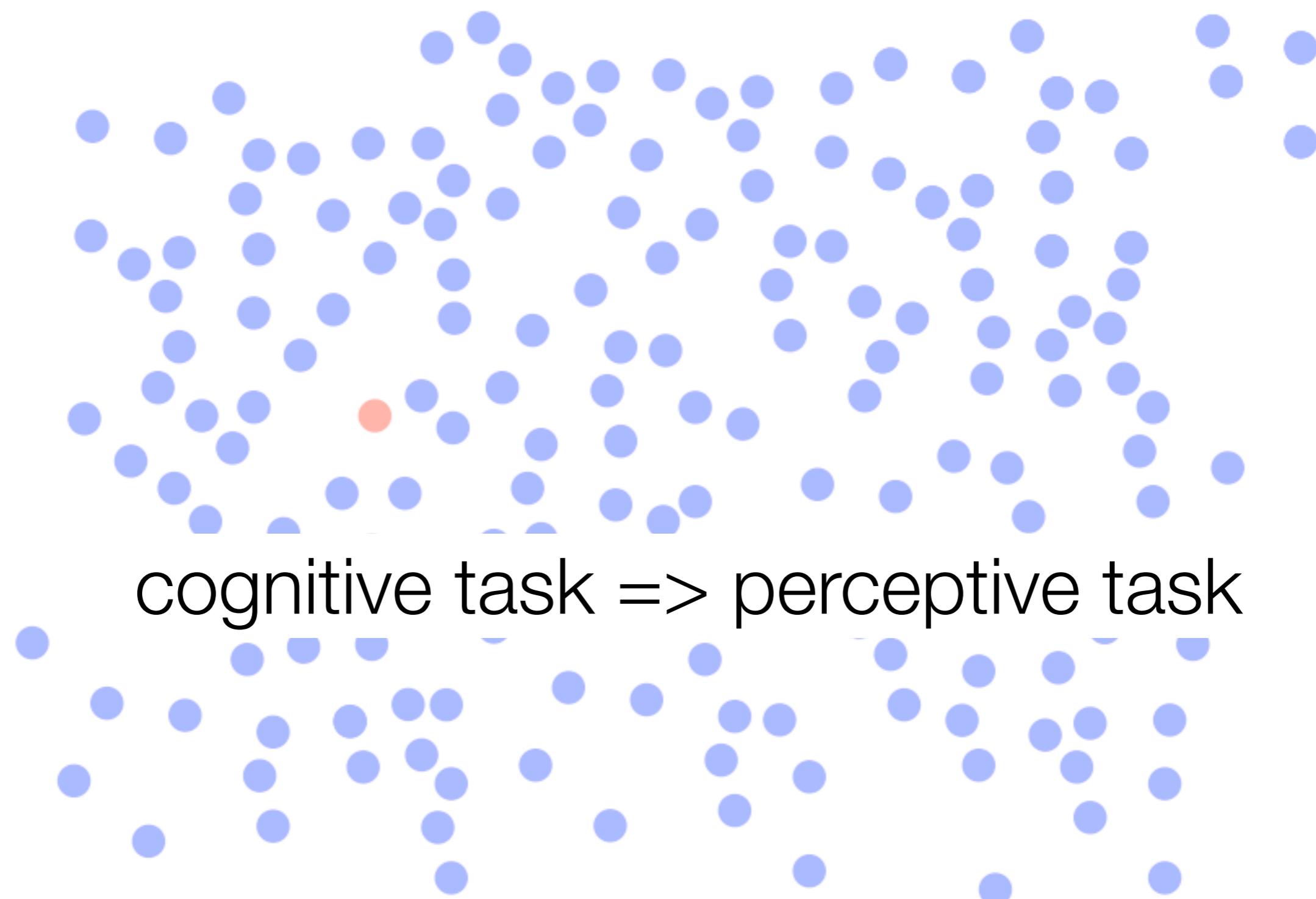
Pre-attentive vision

= ability of low-level human visual system to rapidly identify certain basic visual properties

- some features “pop out”
- used for:
 - target detection
 - boundary detection
 - counting/estimation
 - ...
- visual system takes over => all cognitive power available for interpreting the figure, rather than needing part of it for processing the figure

5

5

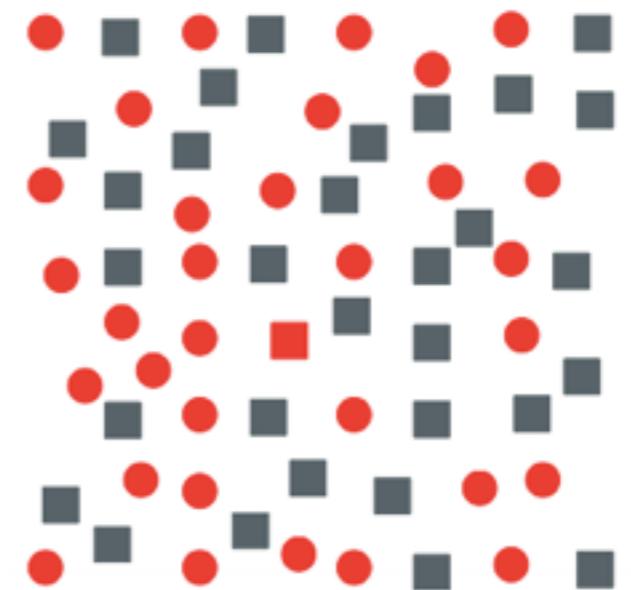


cognitive task => perceptive task

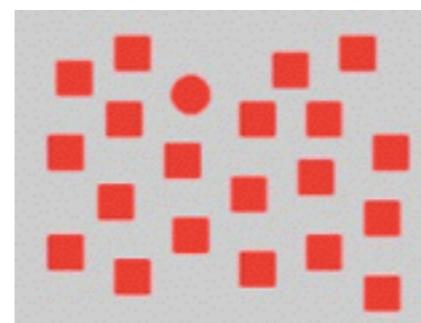
Limitations of preattentive vision

1. **Combining** pre-attentive features does *not* always work => would need to resort to “**serial search**” (most channel pairs; all channel triplets)

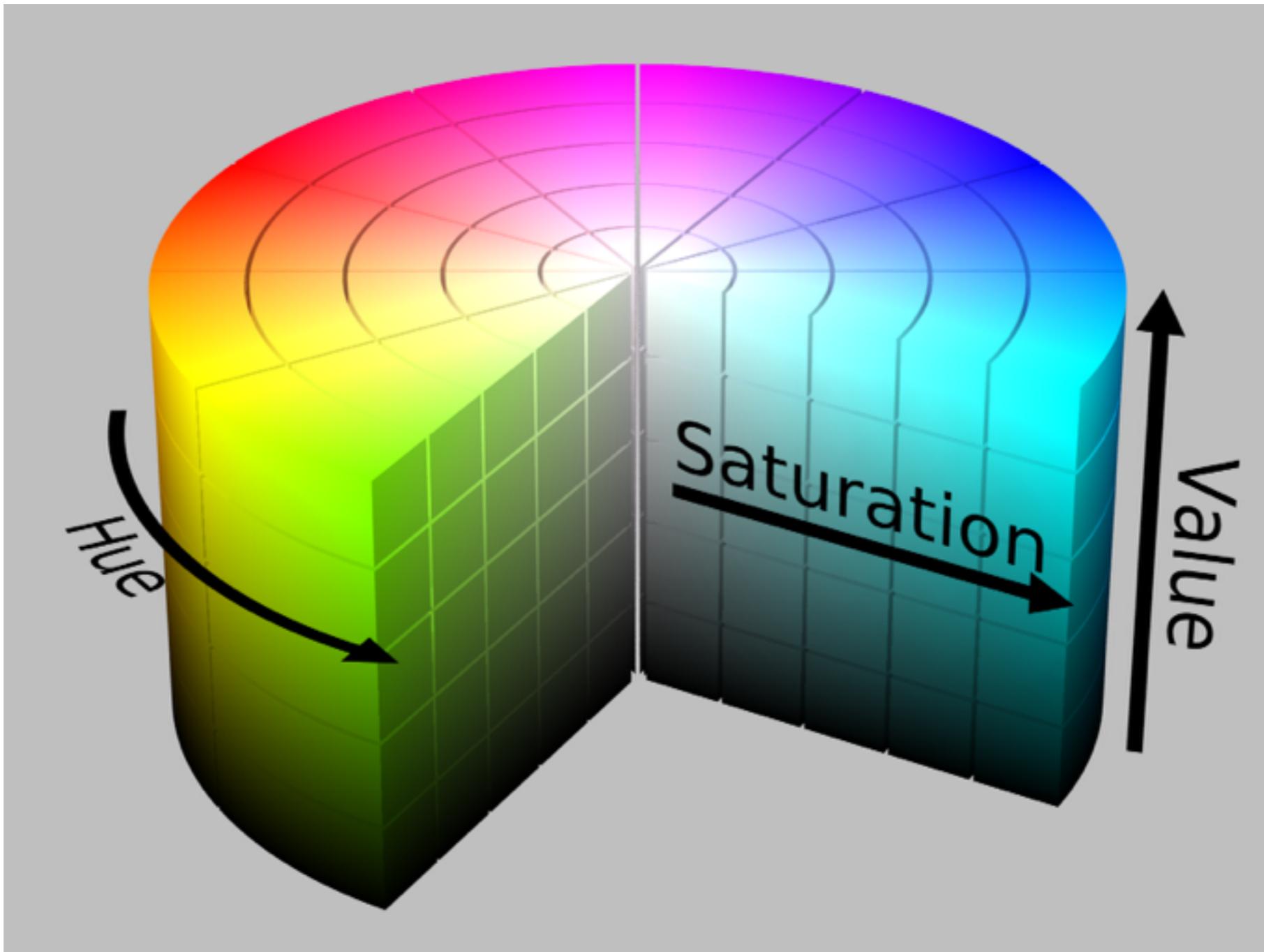
e.g. is there a red square in this picture



2. Speed depends on **which channel** (use one that is good for categorical)



About colour



colorbrewer2.org

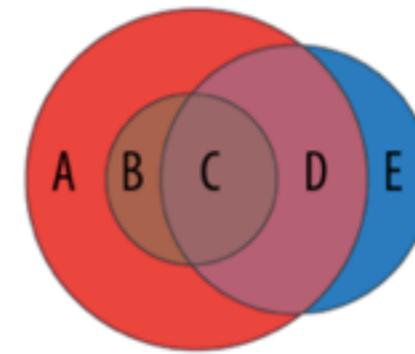
*one color
dominates*



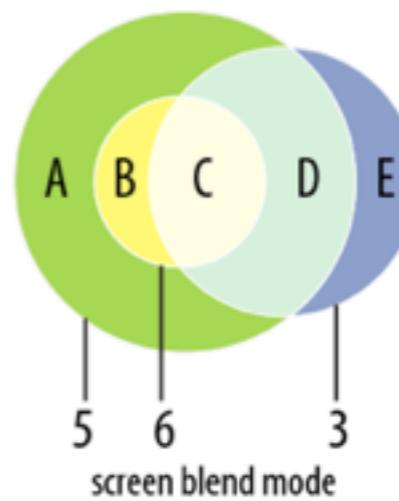
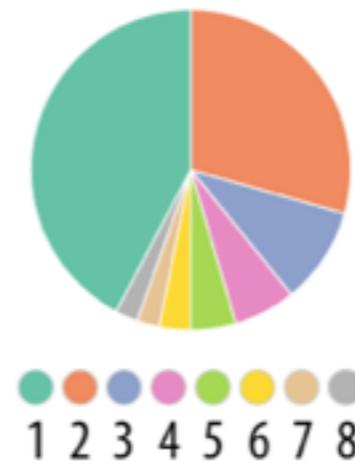
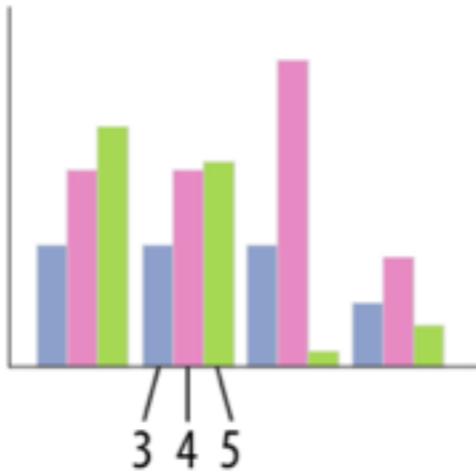
*difficult to
distinguish*



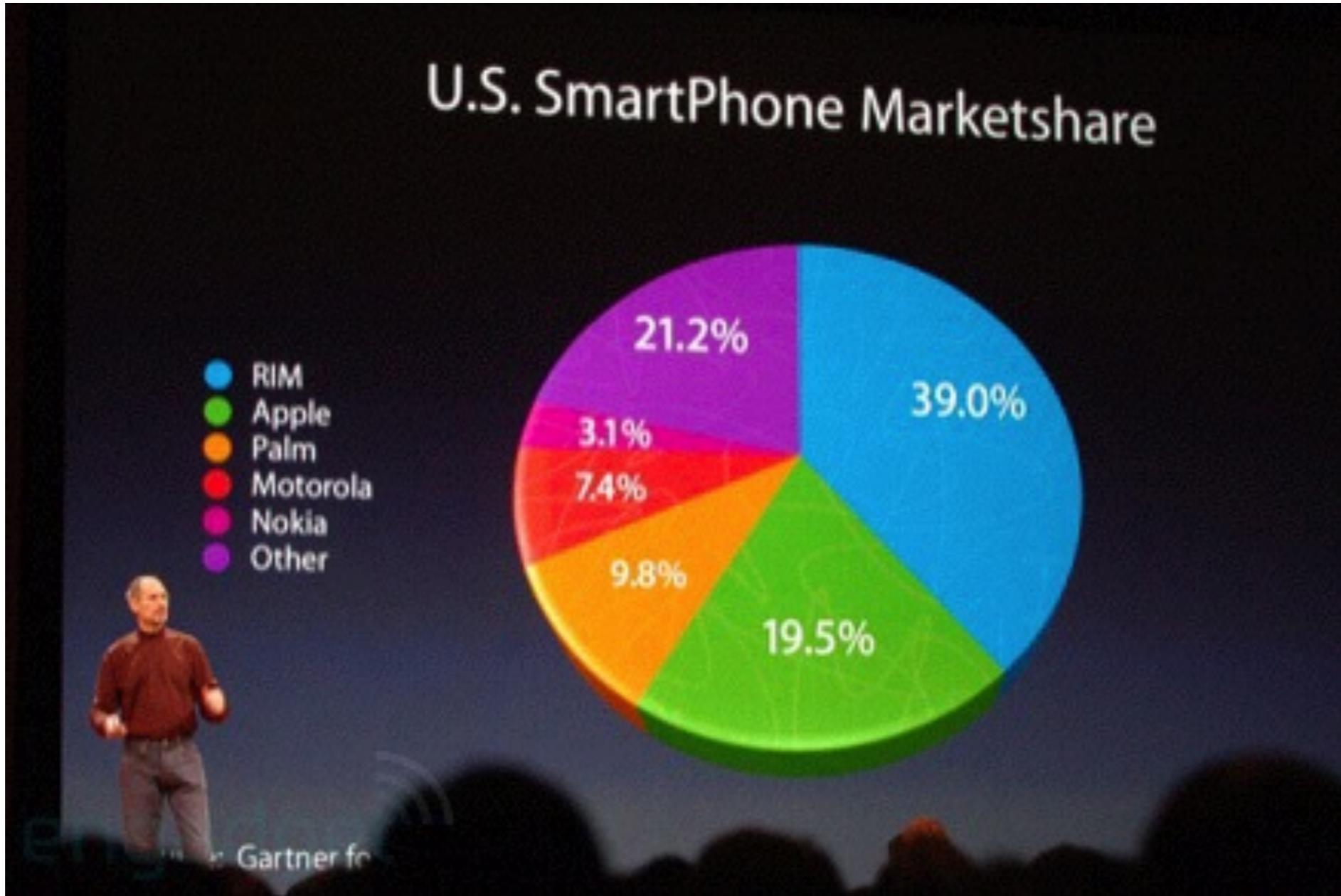
murky



recolored with Brewer palettes

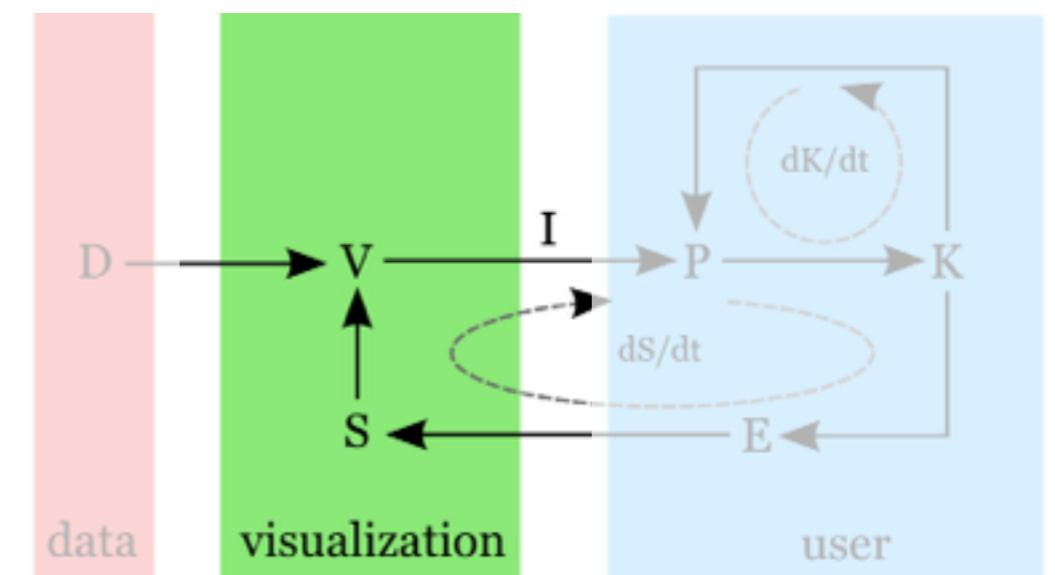


in R: please use RColorBrewer!



"lie factor" = $\frac{\text{size of effect shown in graphic}}{\text{size of effect in data}}$

Visualization foundations and techniques

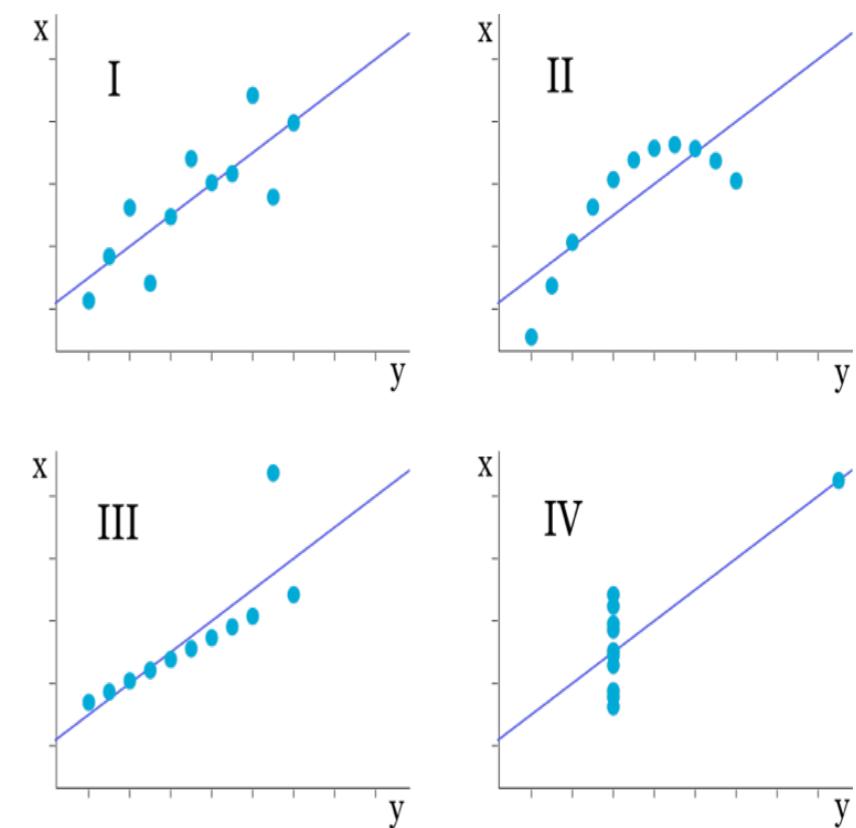
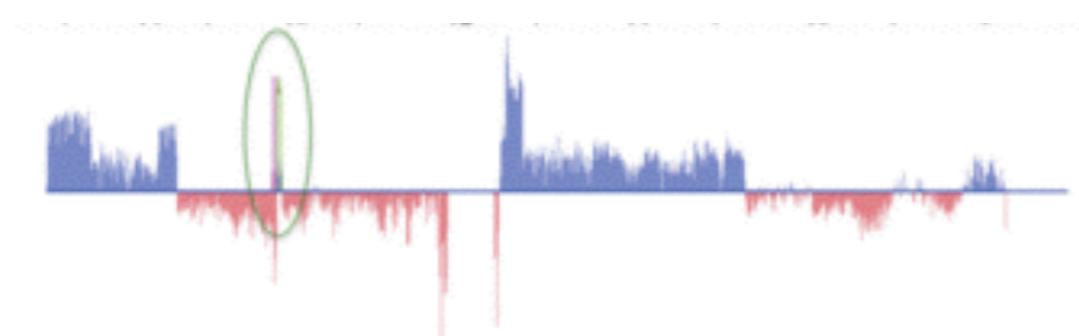
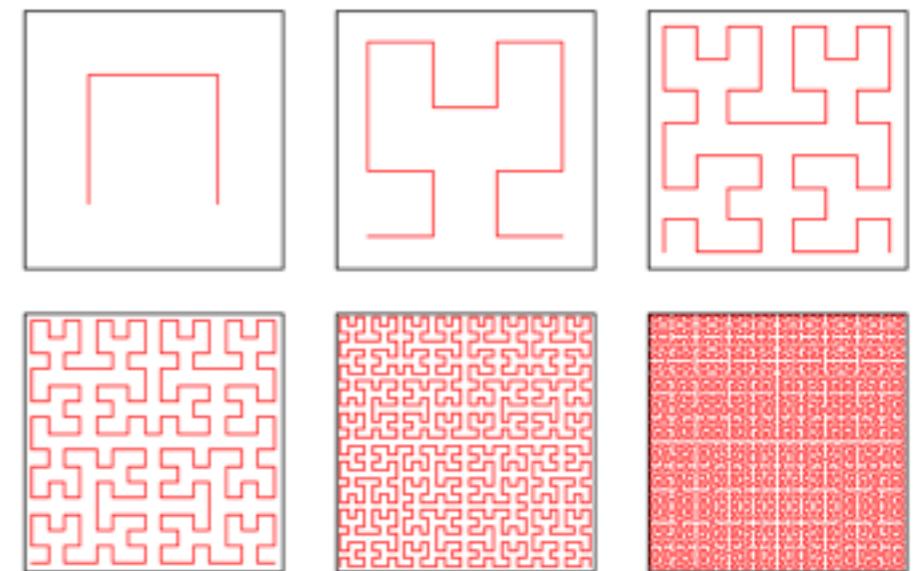


8 visual variables

- See **semiology of graphics**
 - position
 - mark (circle, cross, rectangle, diamond, ...)
 - size (length, area, volume)
 - brightness
 - colour (hue)
- orientation
- texture
- motion

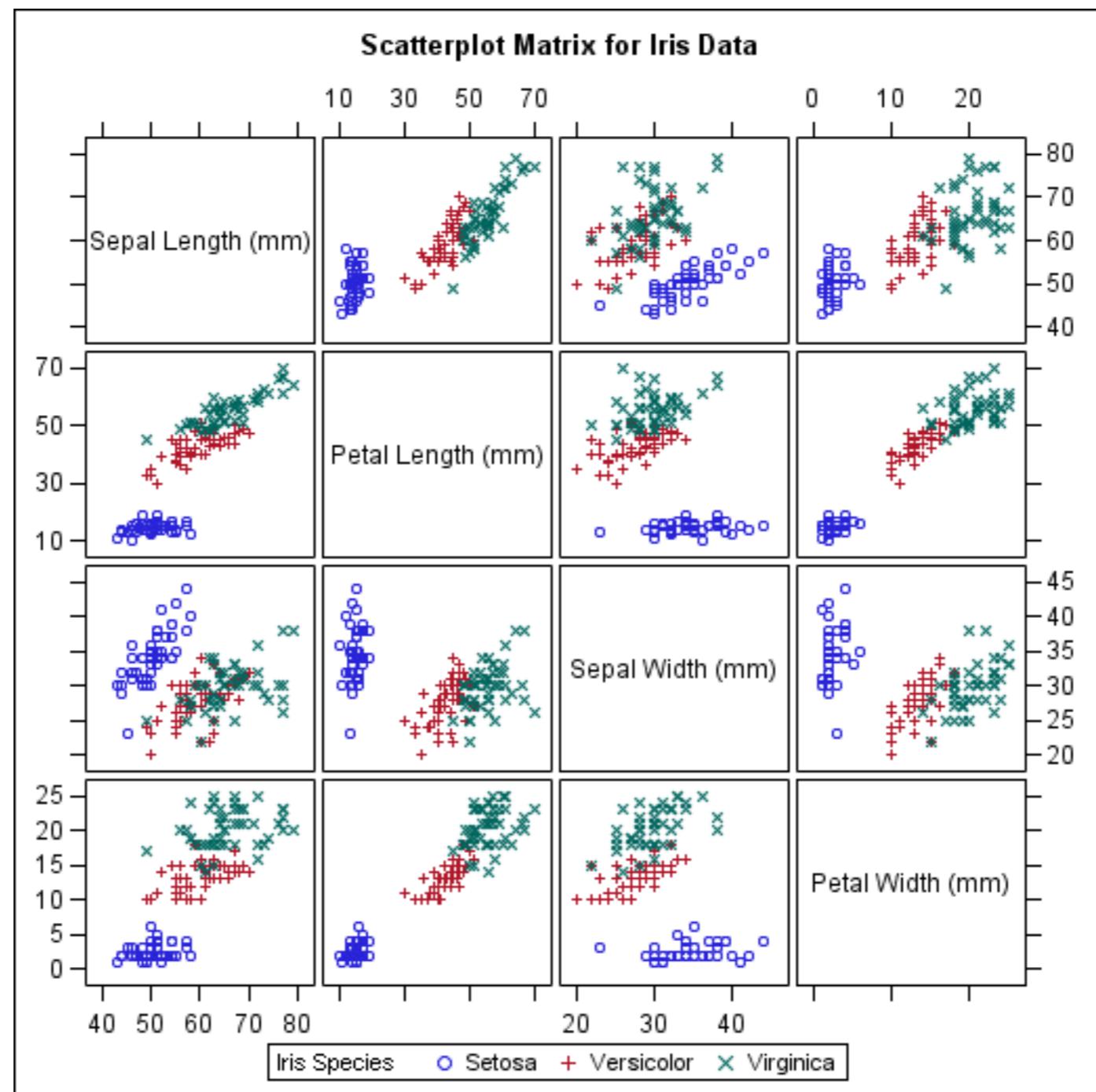
Visualization techniques for (geo)spatial data

- 1D, 2D
- space-filling curves: 1D \rightarrow 2D
 - e.g. Hilbert curve



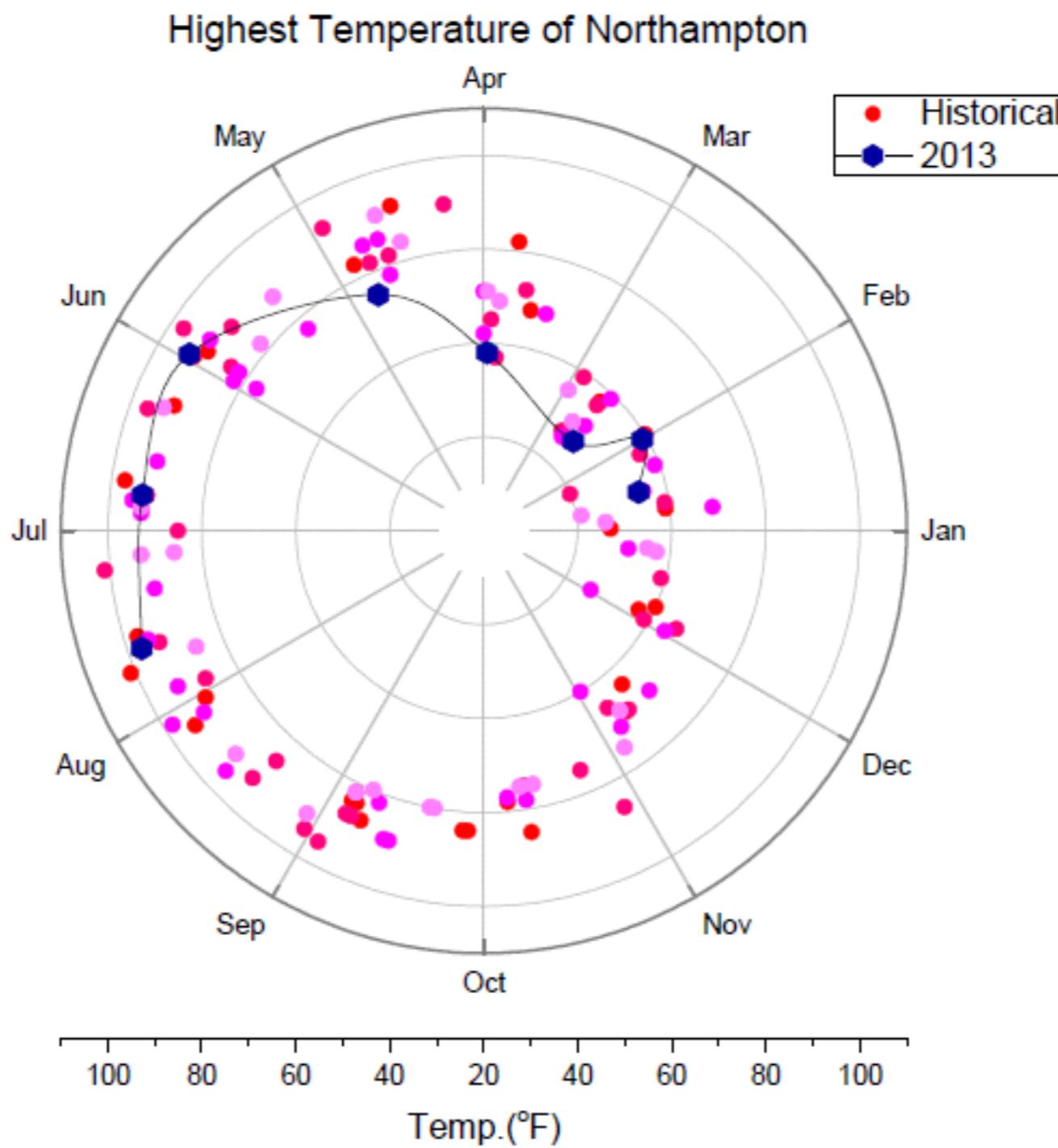
Visualization techniques for multivariate data

- Really important in scientific research: often looking at **multiple dimensions** of the same data (e.g. gene expression values + SNPs + protein structures + phylogeny + geographical location of sample)
- **Point-based techniques**
 - scatterplots and scatterplot matrices
 - radial axis plots (polar coordinates)
 - dimensionality reduction
 - MDS, PCA: calculate most important combinations of dimensions
 - radviz: put all dimensions on circle and act as springs

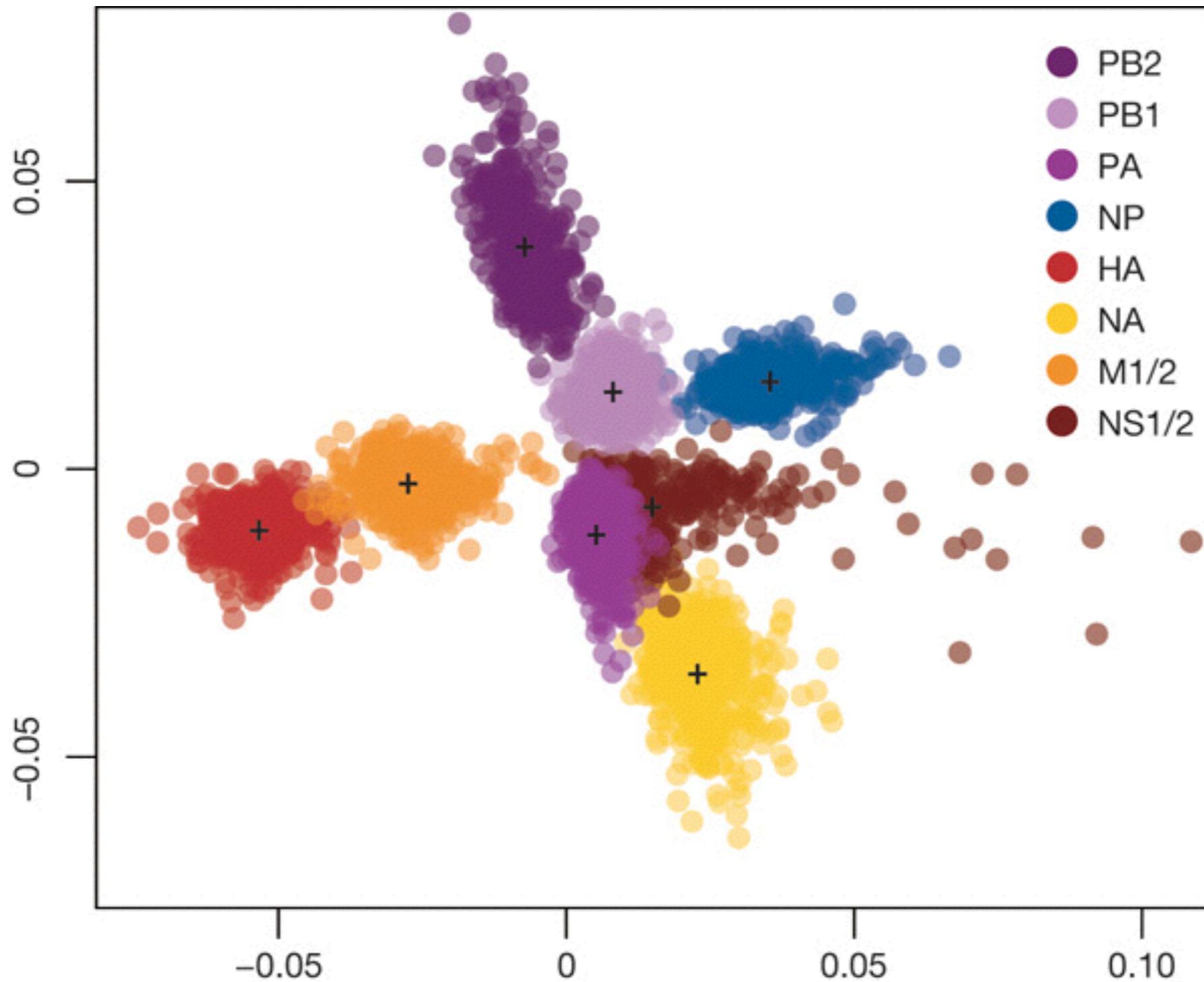


scatterplot matrix

(source: support.sas.com)

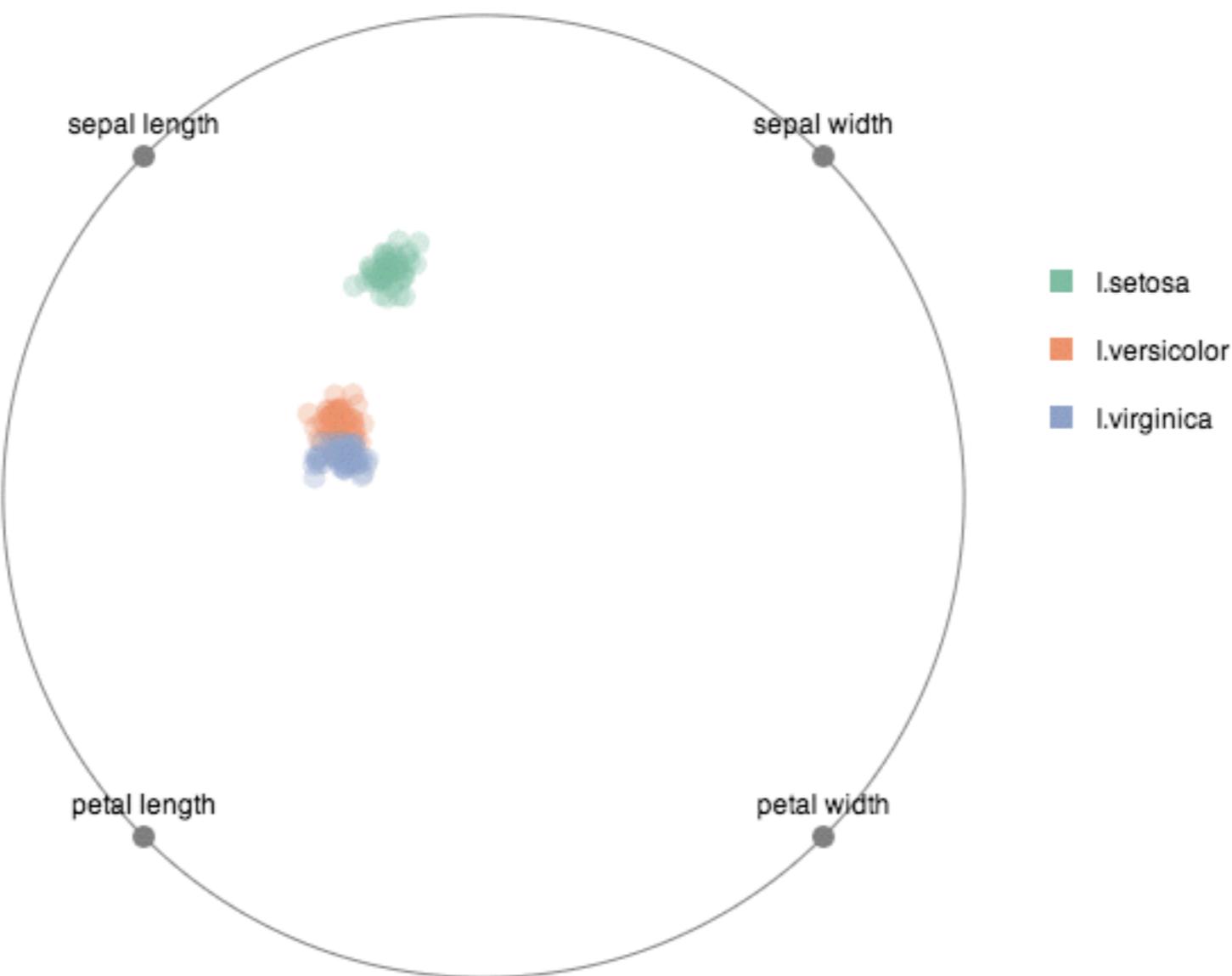


(source:originlab.com)



multi-dimensional scaling plot of distances between samples trees

(source:doi:10.1038/nature06945)

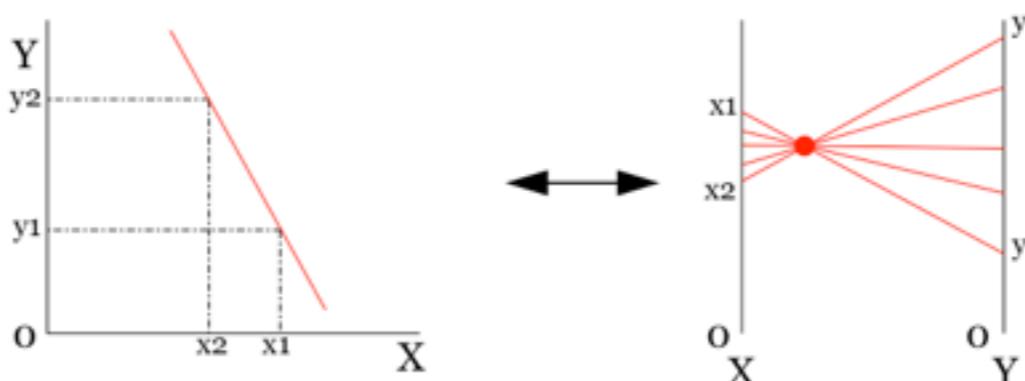
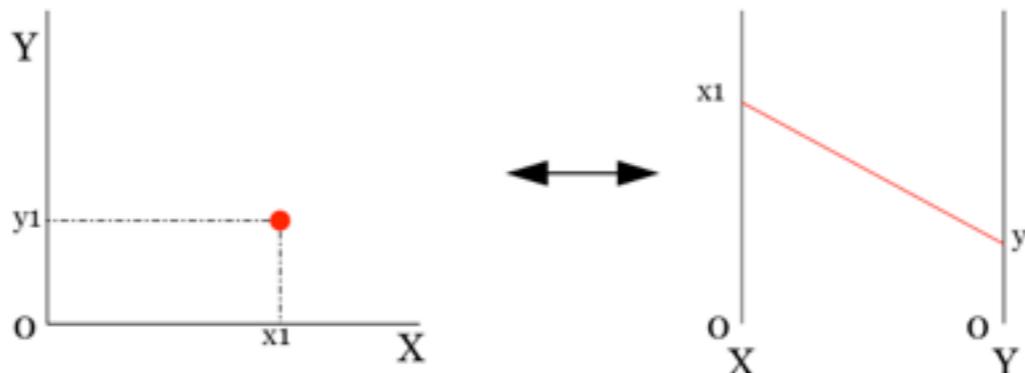


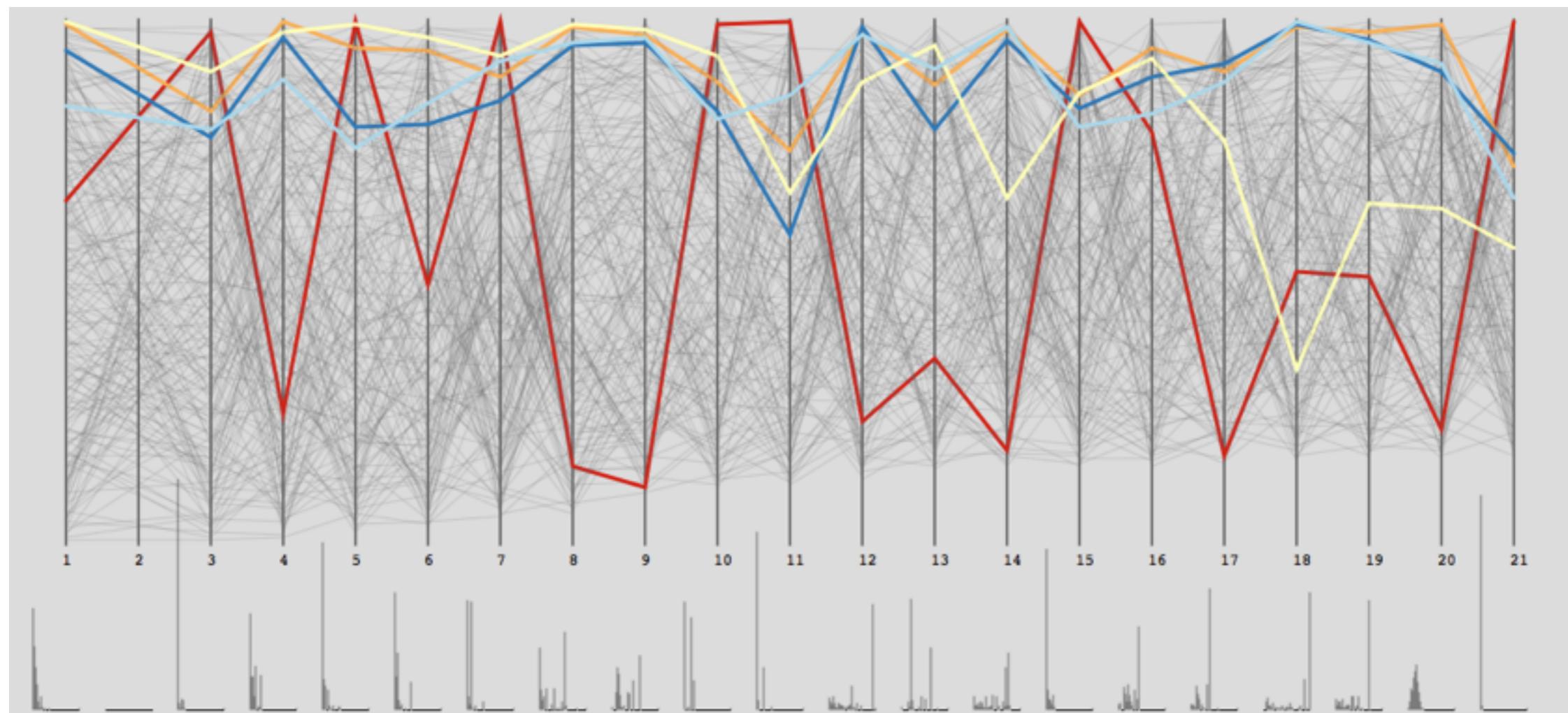
radviz

- **Line-based techniques**

- line graphs

- “parallel coordinates”: point in Euclidean space = line in parallel coordinate space and vice versa





parallel coordinates (Tour de France, 2005)

- **Region-based techniques**
 - barcharts & histograms
 - tabular displays (e.g. heatmap)
 - dimensional stacking

Scale-stack bar chart

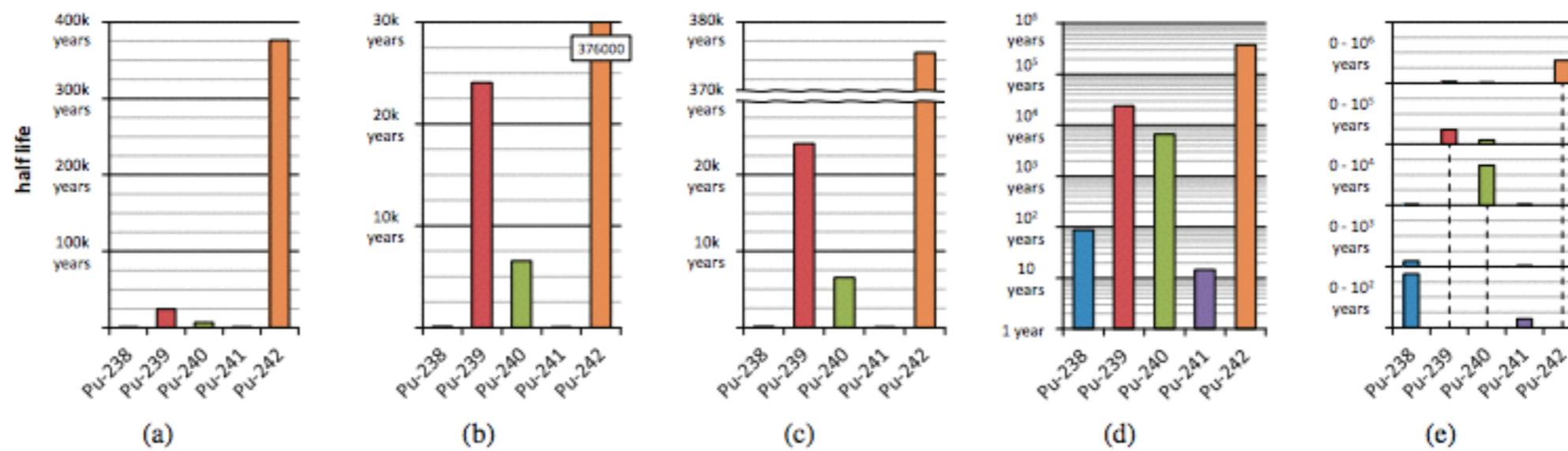
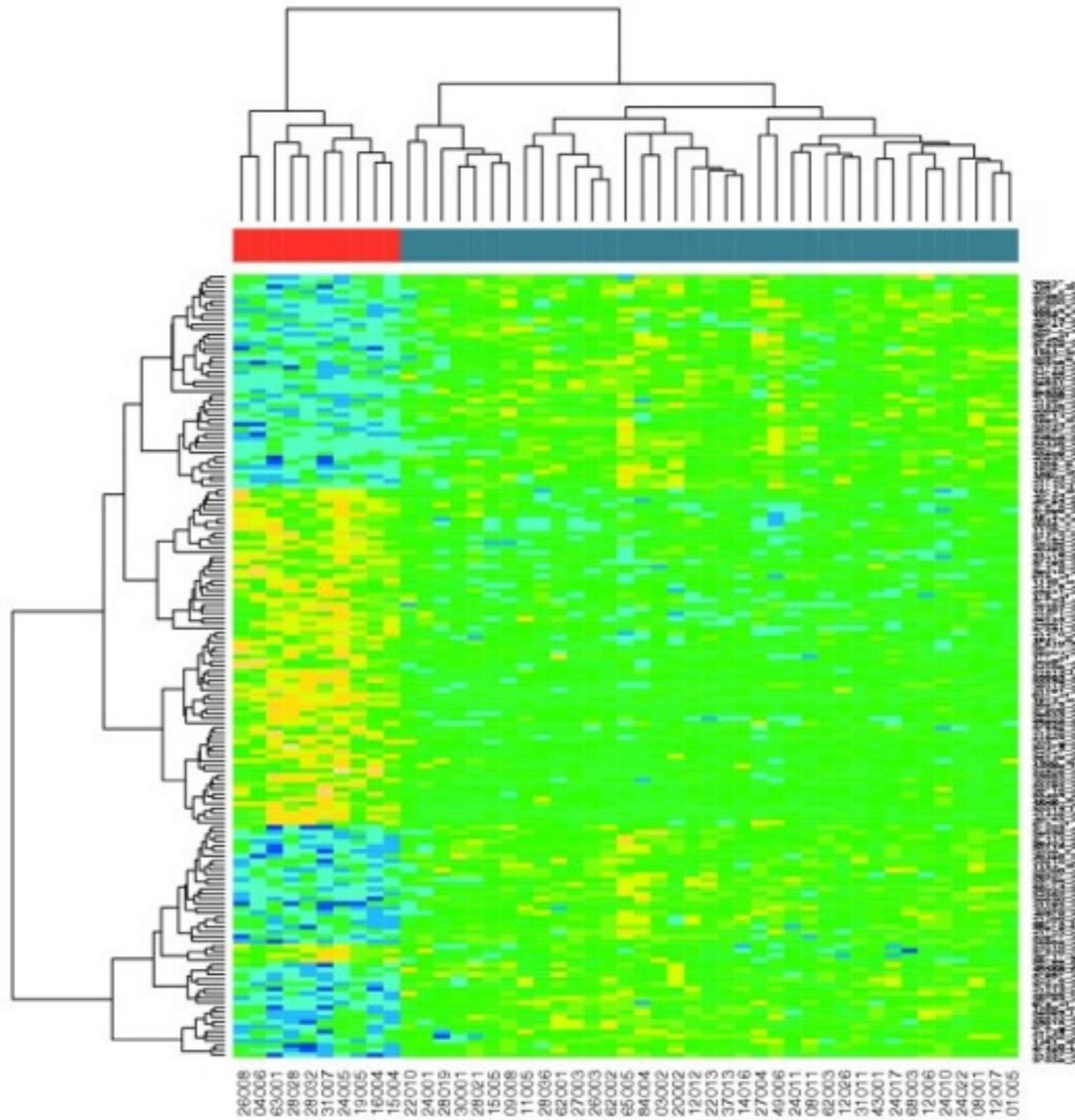


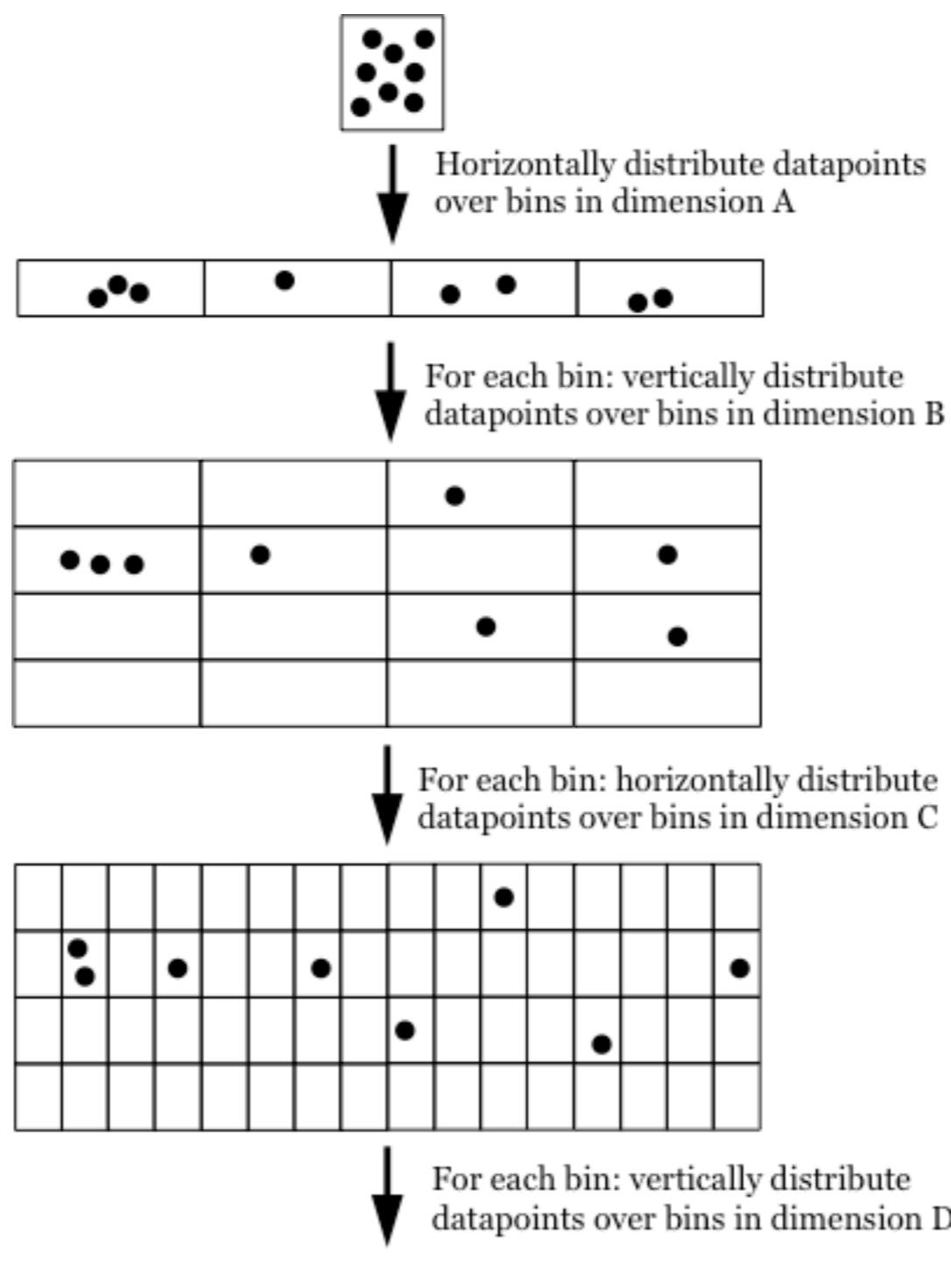
Figure 1: Different bar charts for visualizing data with large value range. Half lives of different plutonium isotopes (see Sect. 6.1) are visualized with: (a) classic linear bar chart, (b) linear bar chart with cut-off bars, (c) linear bar chart with scale break, (d) logarithmic bar chart, and (e) our novel scale-stack bar chart.

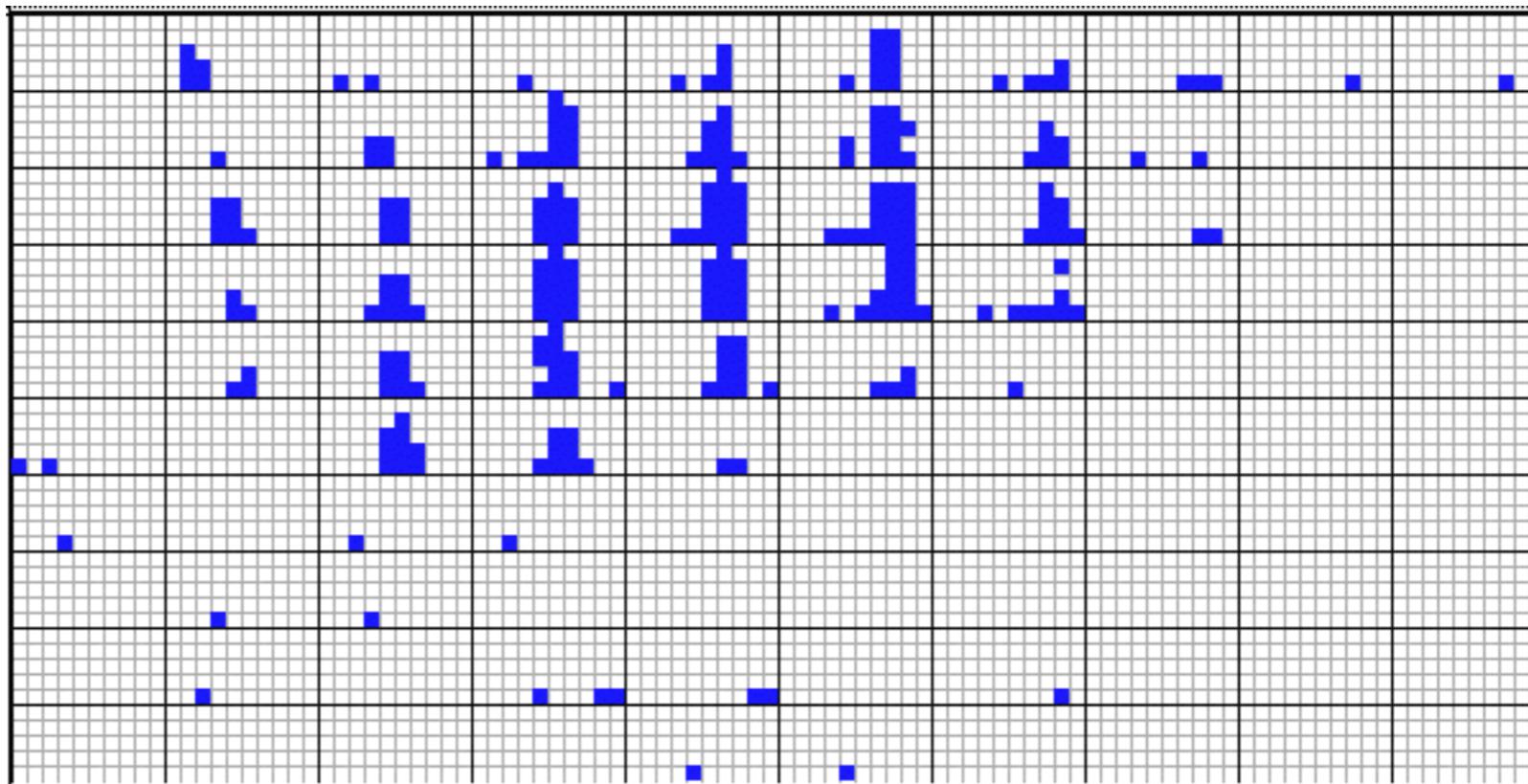
<http://www.vis.uni-stuttgart.de/~sadlo/download/hlawatsch2013scaleStack.pdf>



Heat map

(source:doi:10.1186/gb-2004-5-10-r80)



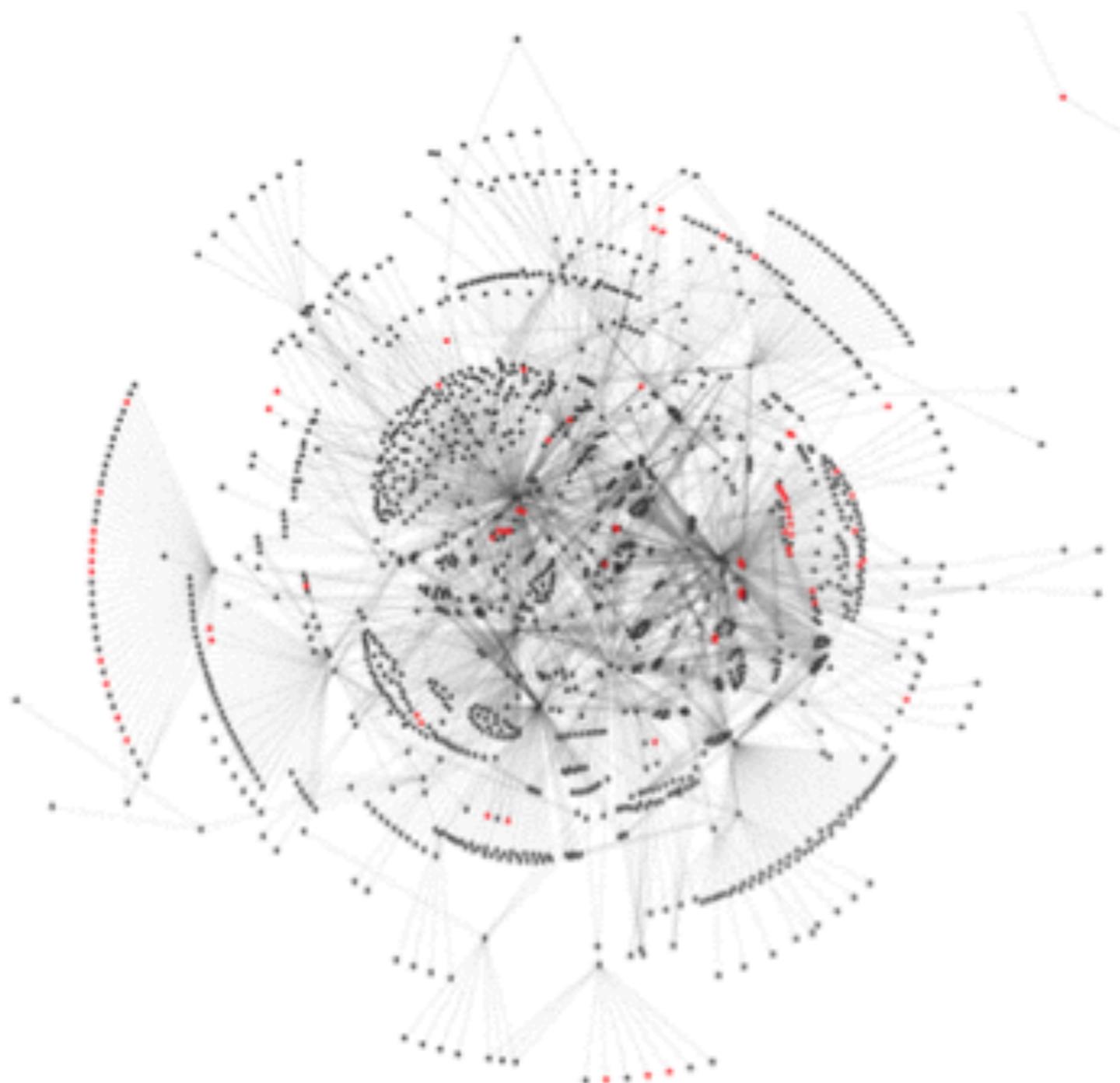


Dimensional stacking example of 4D drill-hole data: 3 spatial dimensions + ore grade as the fourth dimension

(source: <http://web.cs.wpi.edu/>)

Visualization techniques for graphs/networks

- node-link diagrams
 - max 20 nodes...



- adjacency matrix

- compressed adjacency matrix
(Dinkla *et al*, 2012)

Directed Graph (a)

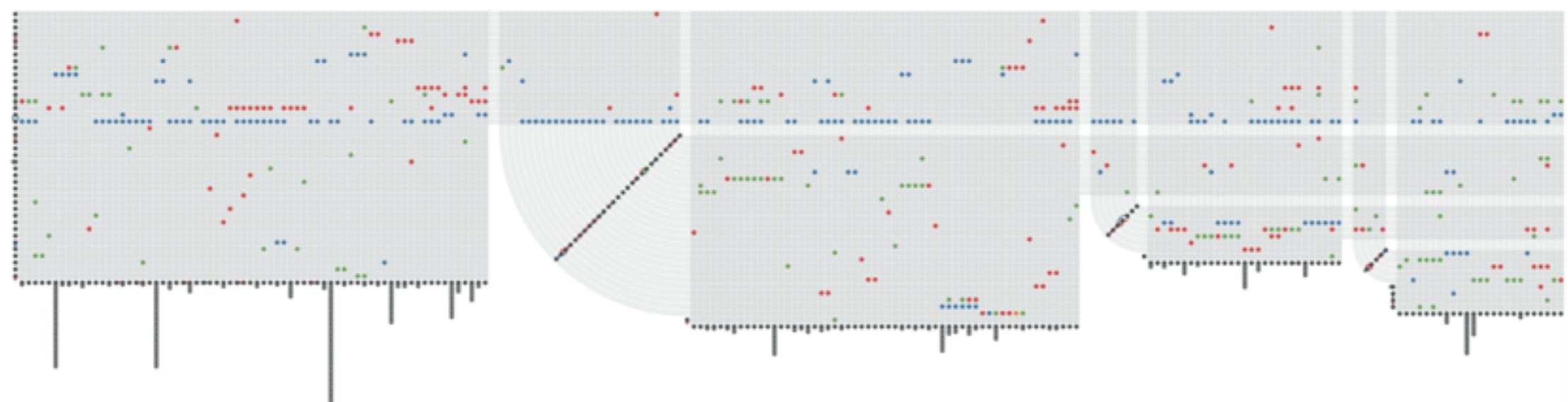
	a	b	c	d	e	f
a		✓				
b				✓	✓	
c	✓					✓
d				✓		✓
e	✓				✓	
f						✓

Adjacency Matrix Representation (a)

Undirected Graph (b)

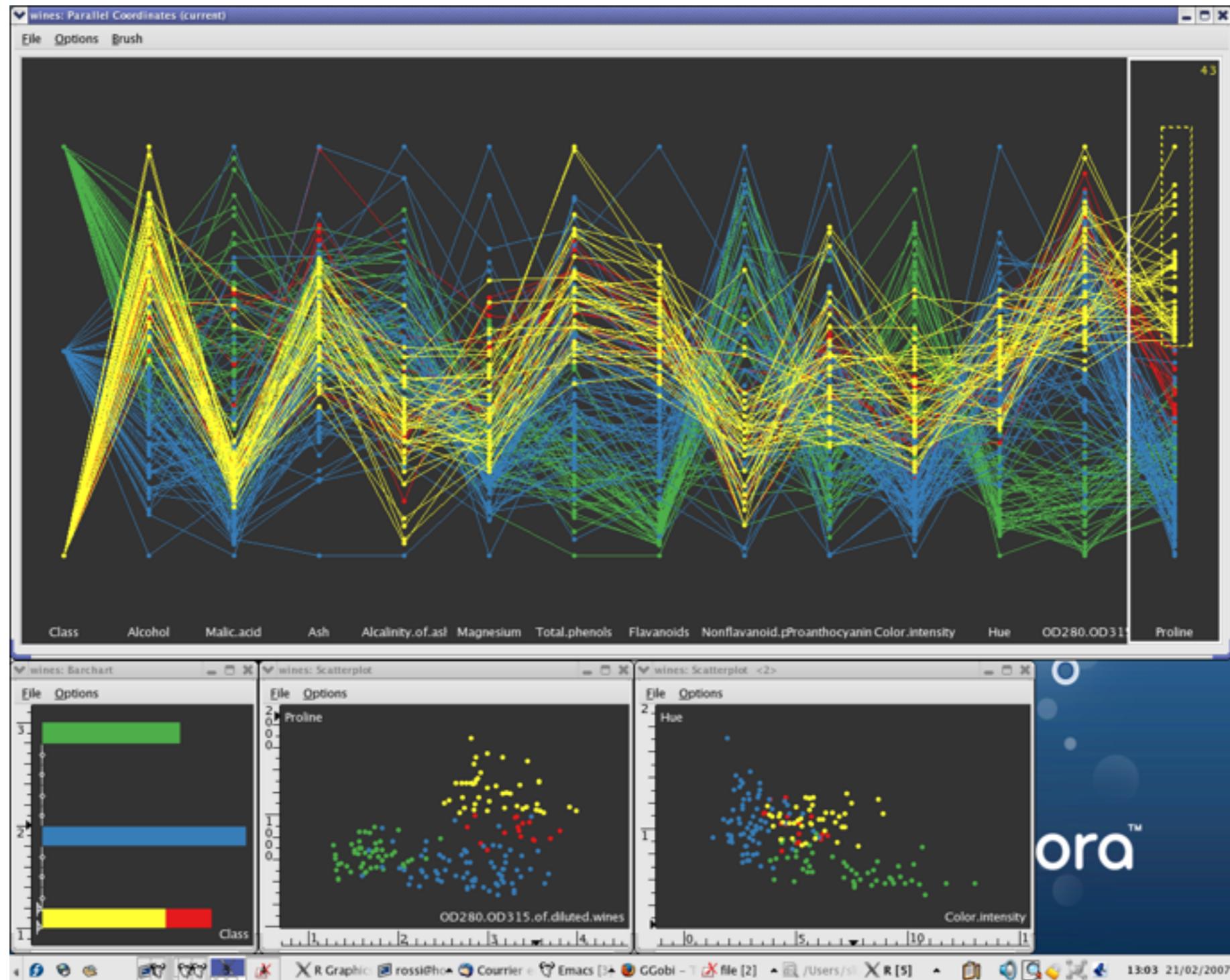
	a	b	c	d	e	f
a		✓				
b	✓		✓	✓		
c				✓		
d			✓	✓		✓
e					✓	
f						✓

Adjacency Matrix Representation (b)



Interaction concepts

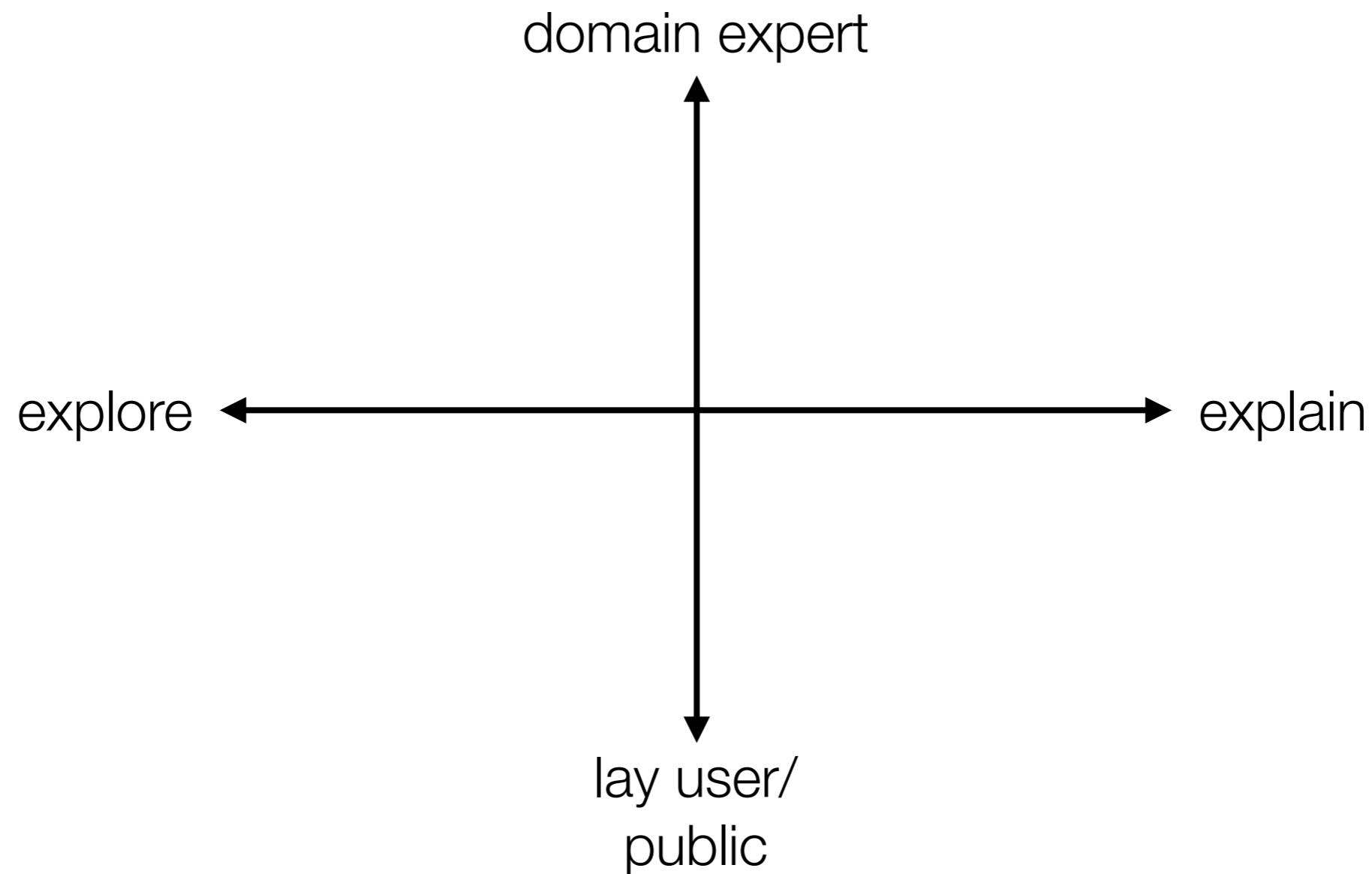
- operators:
 - navigation
 - selection
 - filtering
 - connecting (“brushing and linking”)



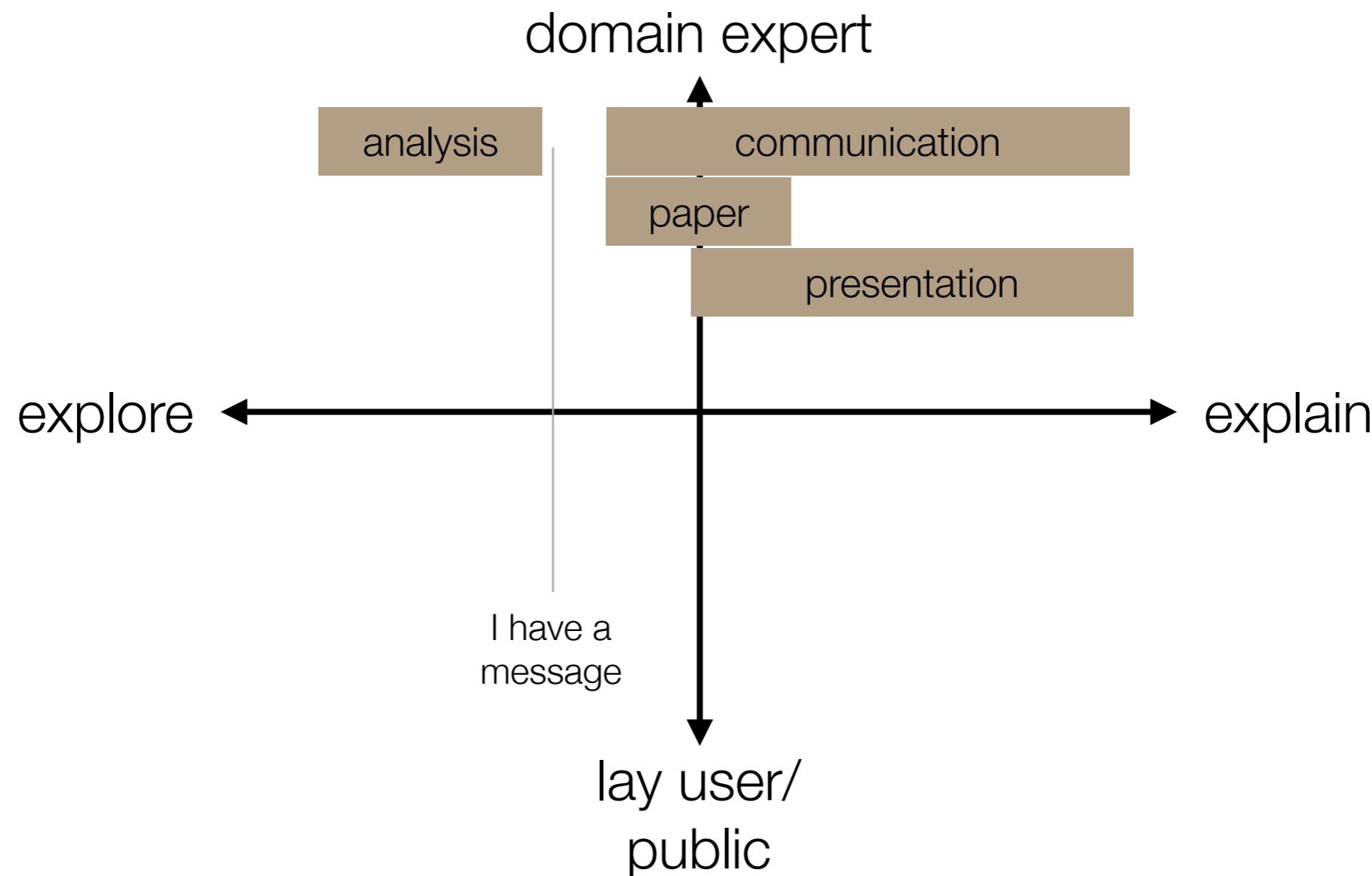
brushing and linking

(source: apiacoa.org)

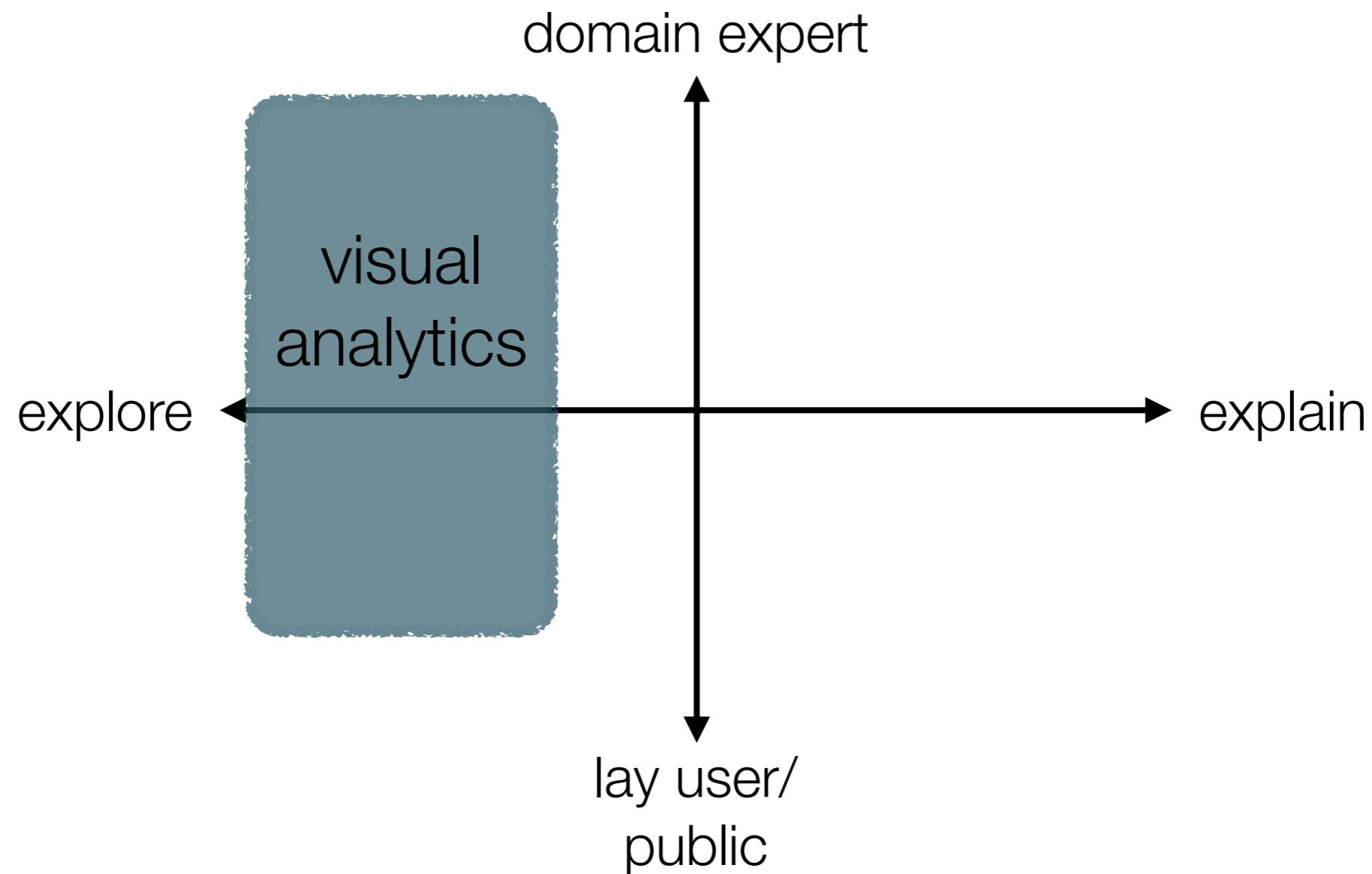
Data visualization framework



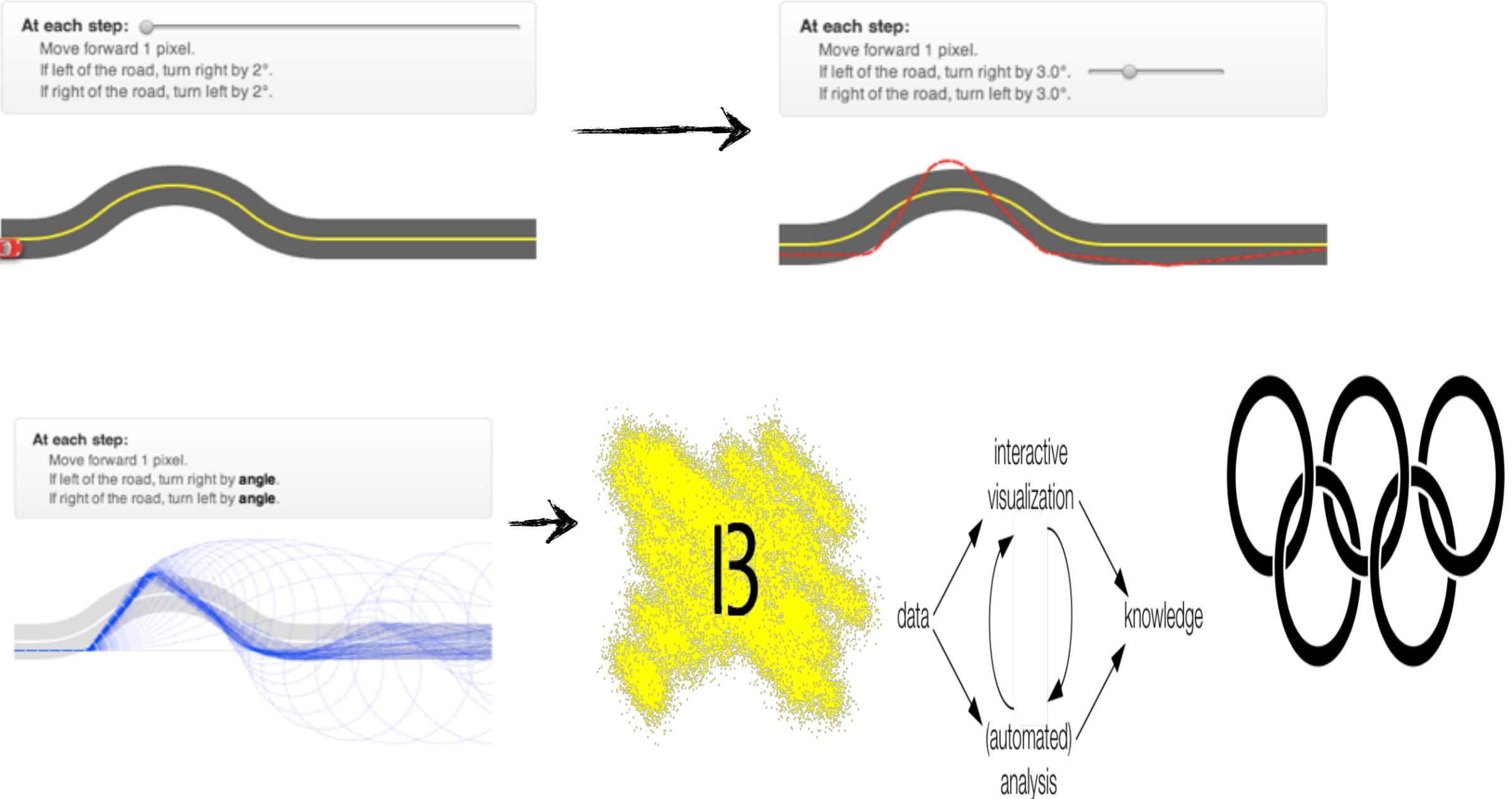
Data visualization framework



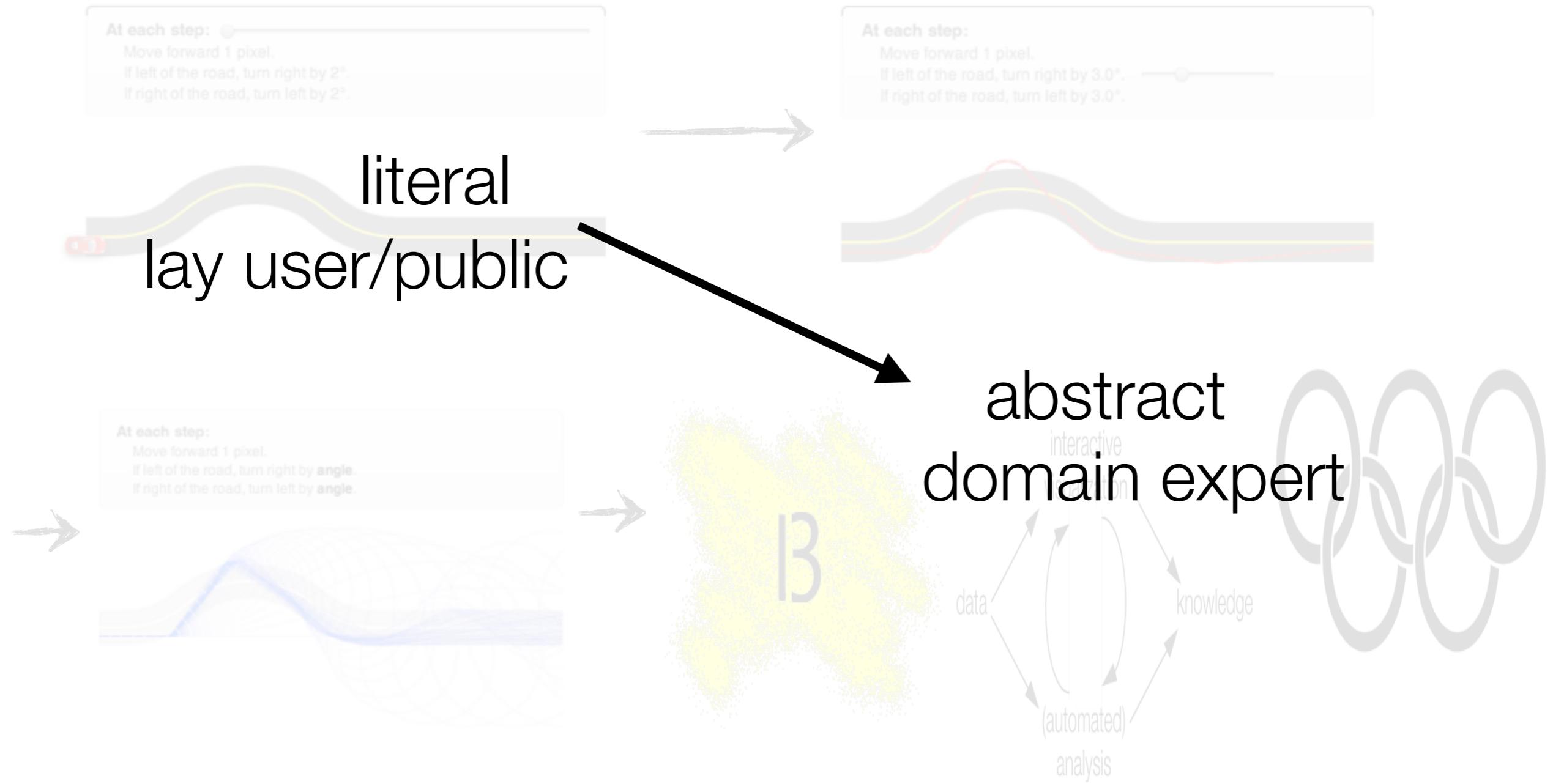
Data visualization framework



Bret Victor - Ladder of abstraction



Bret Victor - Ladder of abstraction



Design decision styles

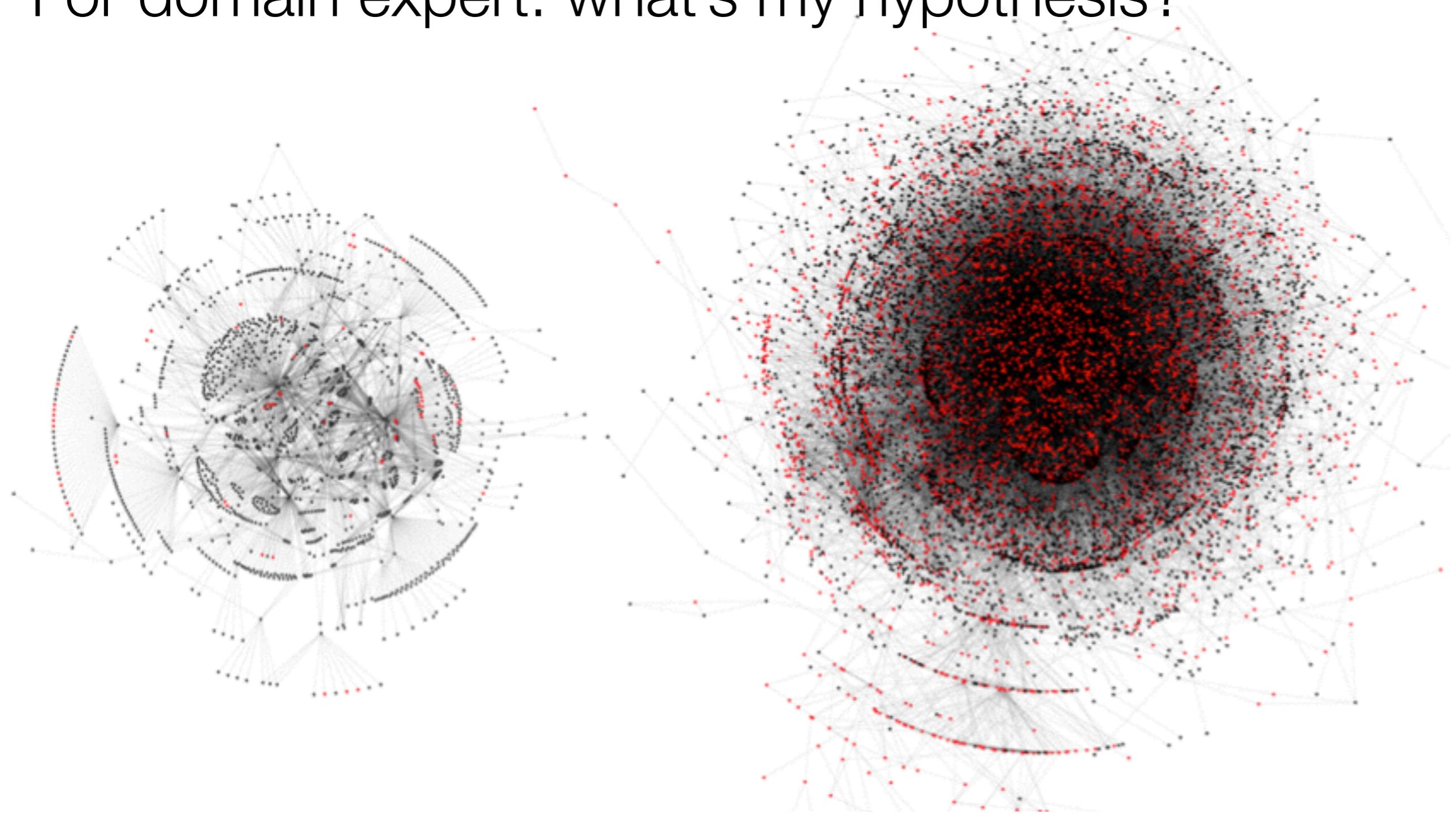
1. **Unintended design** - Design decisions based on what's easiest to implement. Developer focuses on development and deployment without any consideration of what will happen when people use the tool.
2. **Self design** - Design decisions based on by developer's own use.
3. **Genius design** - Developer still does not look beyond own experience, but that experience is extensive.
4. **Task-focused design** - Developer investigates which actions the user wants/needs to perform.
5. **Goal-focused design** - Developer goes further than activities and investigates goals, needs and contexts of the user.

Design decision styles

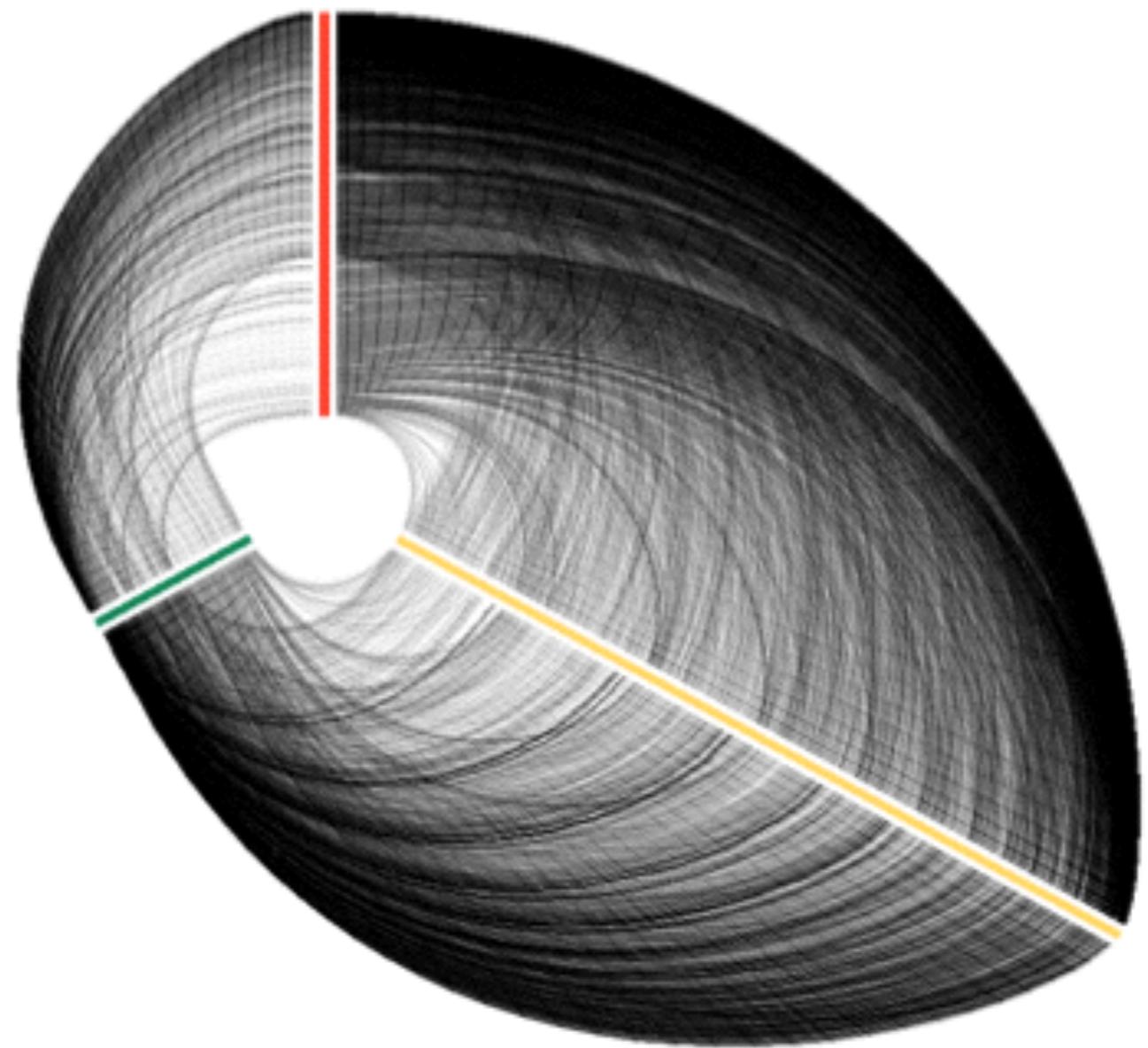
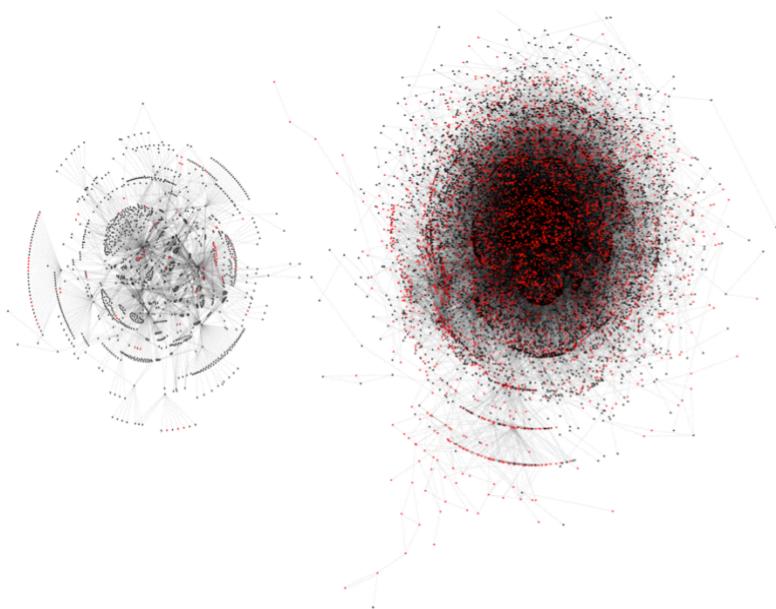
1. **Unintended design** - Design decisions based on what's easiest to implement. Developer focuses on development and deployment without any consideration of what will happen when people use the tool.
2. **Self design** - Design decisions based on by developer's own use.
combine
3. **Genius design** - Developer still does not look beyond own experience, but that experience is extensive.
4. **Activity-focused design** - Developer investigates which actions the user wants/needs to perform.
5. **User-focused design** - Developer goes further than activities and investigates goals, needs and contexts of the user.

Examples hypothesis generation

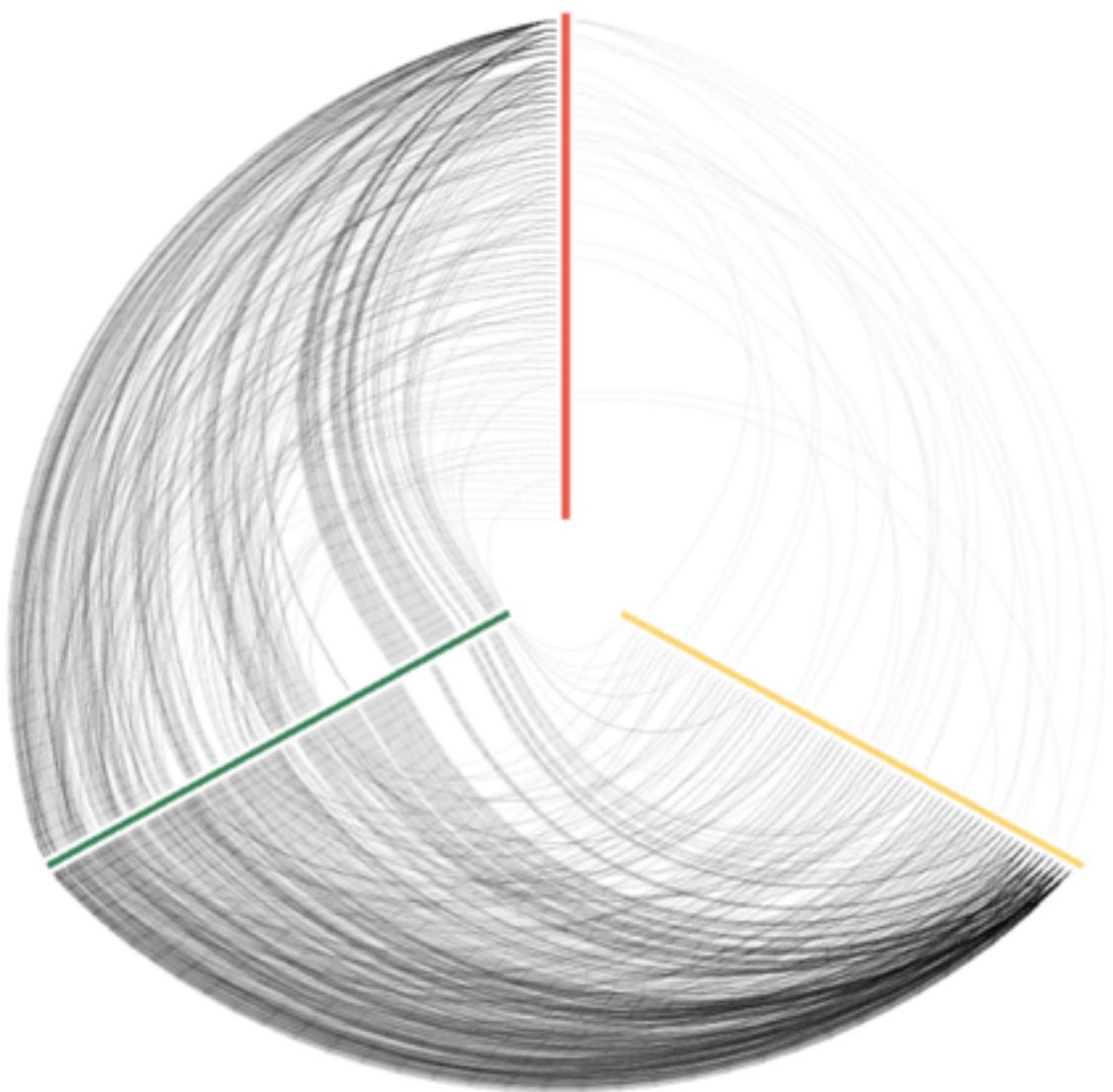
For domain expert: what's my hypothesis?



Martin Krzywinski

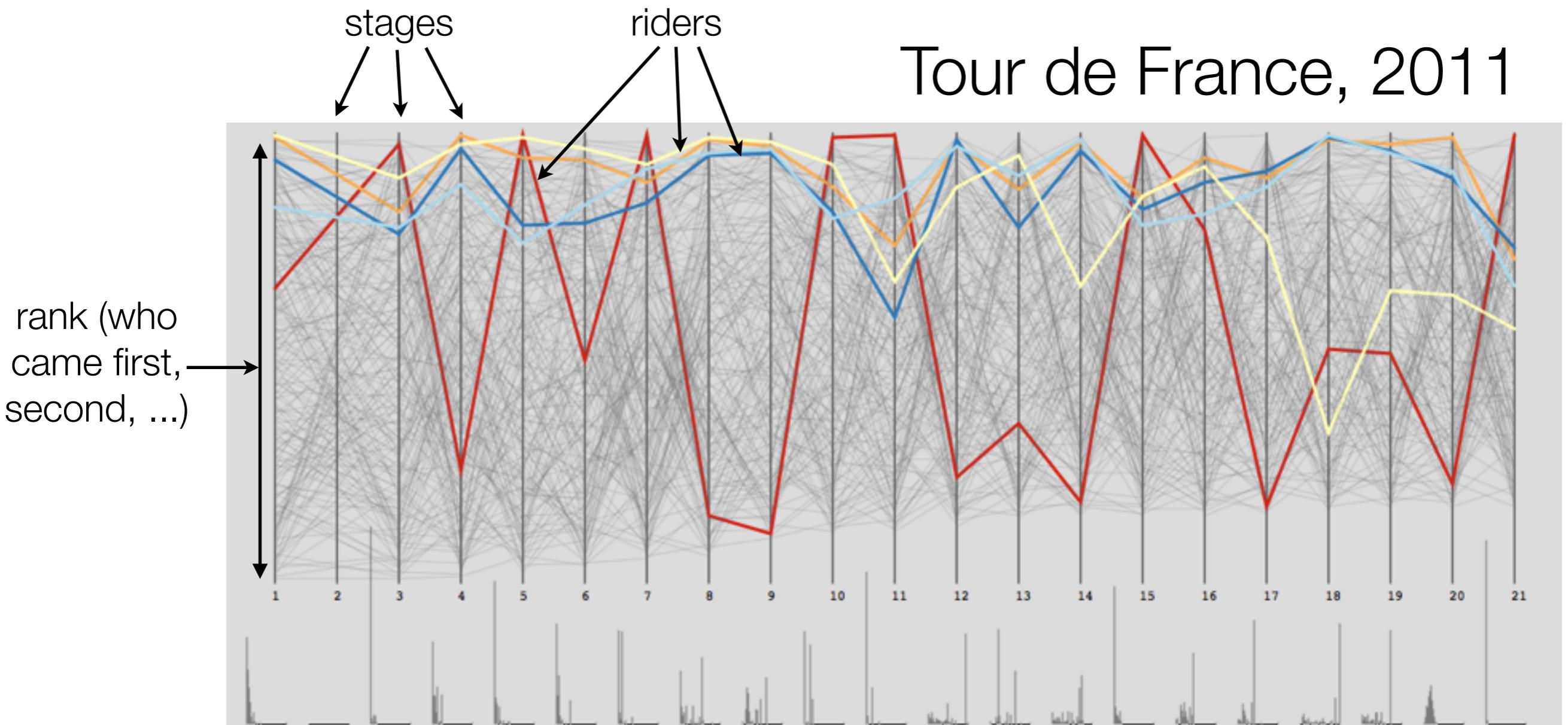


Martin Krzywinski



Martin Krzywinski

Tour de France, 2011



Cavendish: red

Evans: orange

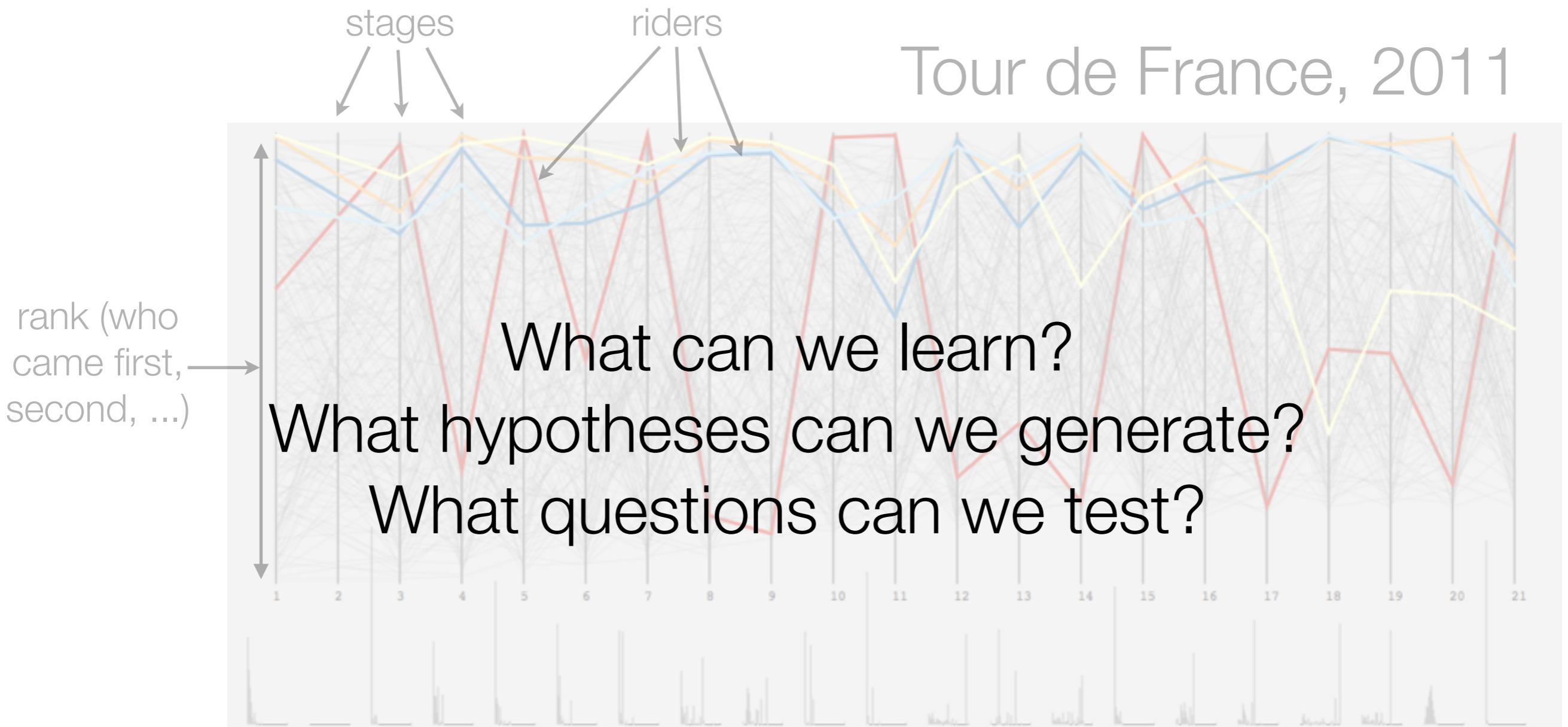
Gilbert: yellow

Andy Schleck: light blue

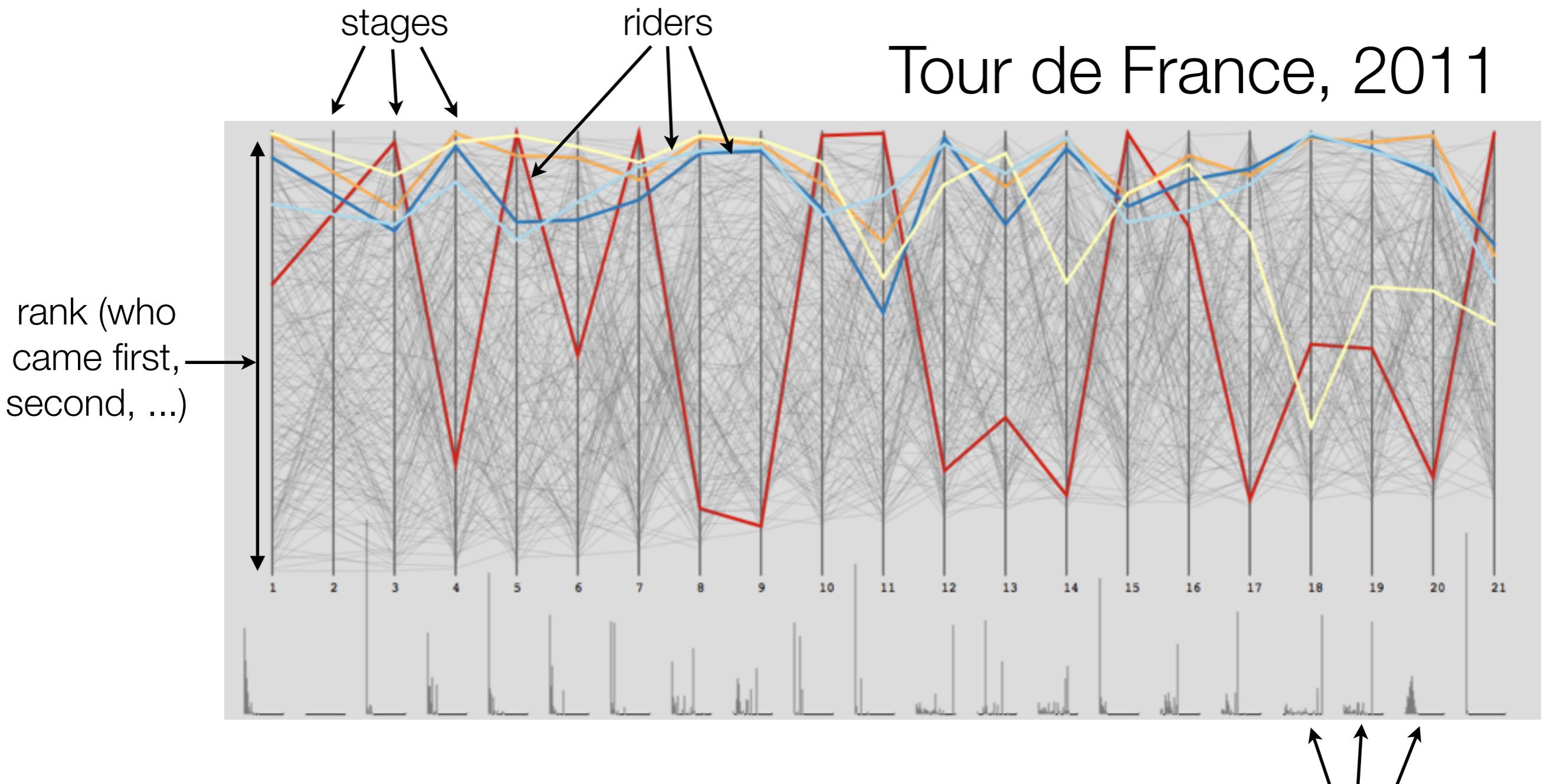
Frank Schleck: dark blue

↑ ↑ ↑
distribution of arrival times

Tour de France, 2011



Tour de France, 2011



- Which trips are in mountains?
 - When do riders start giving up and leaving the Tour?
 - Is Cavendish an expert for the mountains or for flats?
 - How does Philippe Gilbert perform across the Tour?
 - What's with the group of riders performing bad in stages 18 and 20, but good in stage 19?
- distribution of arrival times

Overview of lecture

- A. Why visual analytics?
- B. Data visualization
 - Data foundations
 - Human perception foundations
 - Visualization foundations and techniques
- C. **Visualization evaluation**
- D. Tools of the trade
- E. Examples
- F. Exercises

C. Visualization evaluation

Quantitative evaluation

- spike data with known signal, and record time that it takes for user to find that signal => measure => run statistics “this visualization is better than that one”
- user tasks:
 - identify
 - locate
 - distinguish
- categorize
- cluster
- rank
- compare
- associate
- correlate

Qualitative evaluation

- very close interaction with domain expert
- let expert use the interactive visualization and try to find out what insights he/she gained from the visualization
 - experimenter observation
 - think-aloud protocol
 - collecting participant opinions

Make sure you measure the right thing

problem: you misunderstood their needs

abstraction: you're showing them the wrong thing

encoding: the way you show it doesn't work

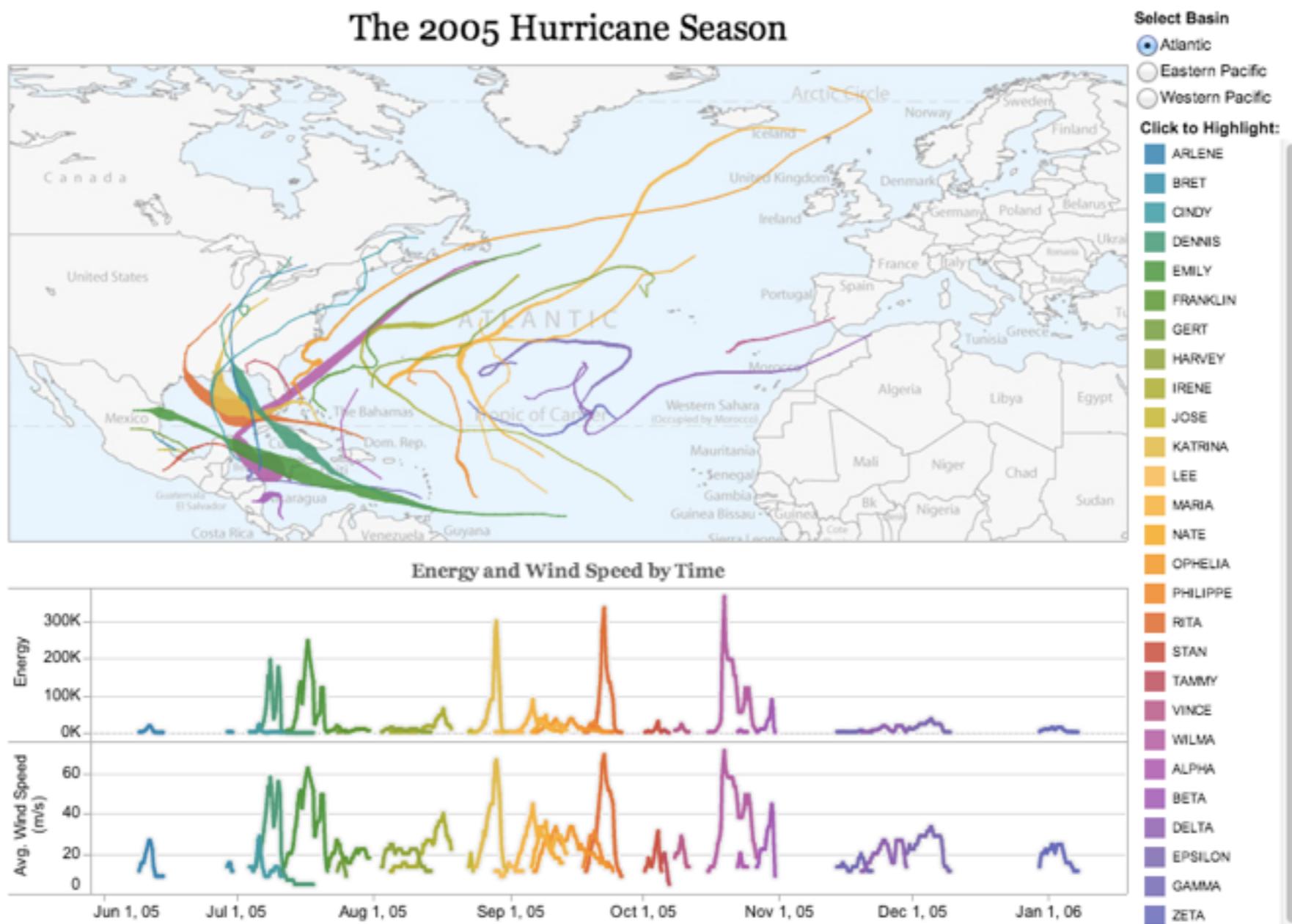
algorithm: your code is too slow

Overview of lecture

- A. Why visual analytics?
- B. Data visualization
 - Data foundations
 - Human perception foundations
 - Visualization foundations and techniques
- C. Visualization evaluation
- D. Tools of the trade
- E. Examples
- F. Exercises

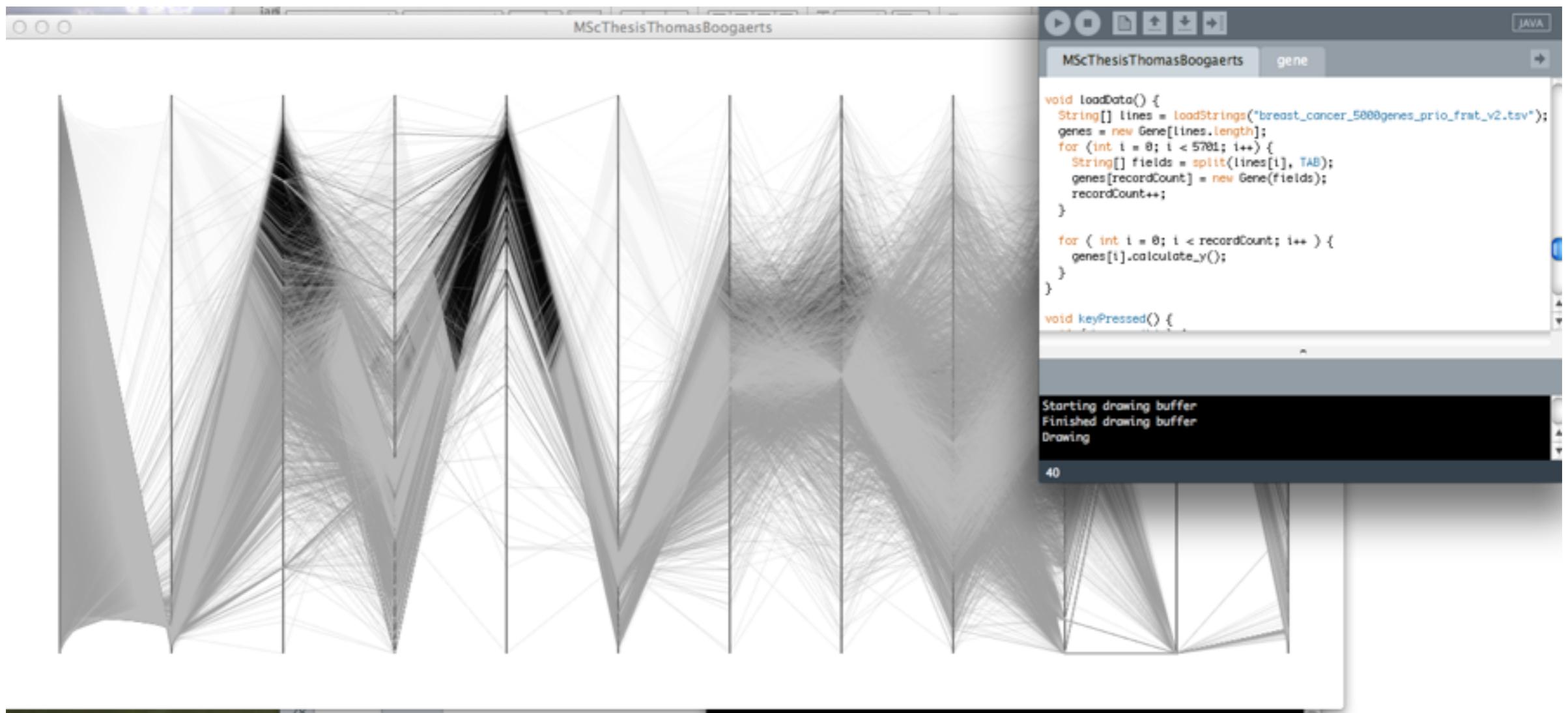
D. Tools of the trade

Tableau



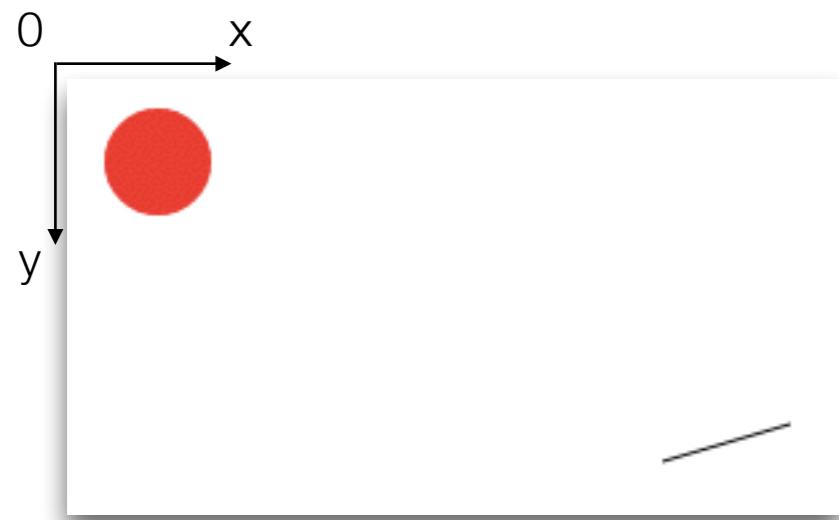
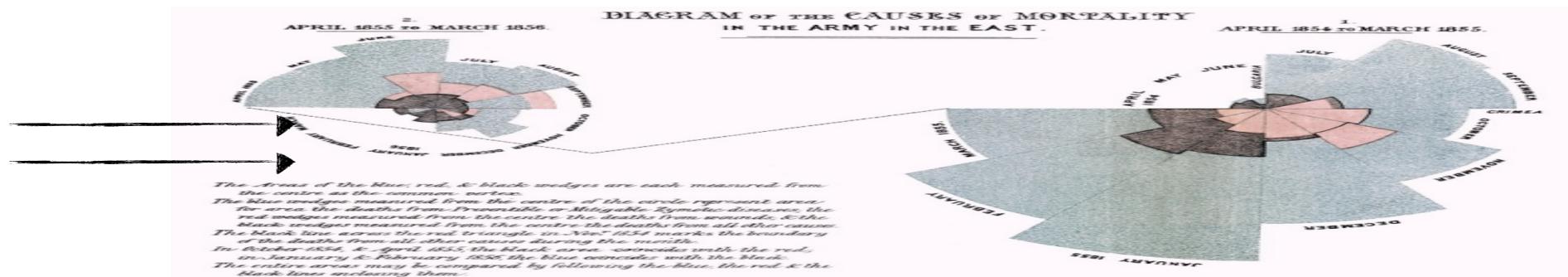
Processing - <http://processing.org>

- java



SVG

my_first_graphic.html

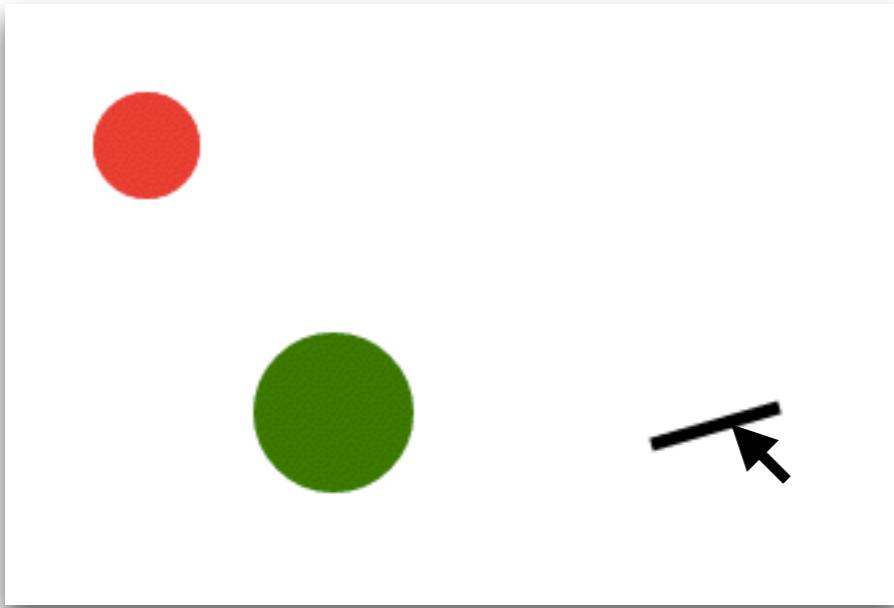


<http://www.w3.org/TR/SVG/>

<http://www.w3schools.com/SVG/default.asp>

<https://developer.mozilla.org/en-US/docs/Web/SVG/Element>

Adding some interactivity



```
1  <html>
2    <head>
3      <style>
4        circle:hover {
5          opacity: 0.5;
6        }
7        line:hover {
8          stroke-width: 5;
9        }
10       </style>
11     </head>
12     <body>
13       <svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink">
14         <line x1="239" y1="212" x2="287" y2="198" stroke="black"></line>
15         <circle cx="50" cy="100" r="20" fill="red"></circle>
16       <a xlink:href="http://www.standaard.be">
17         <circle cx="120" cy="200" r="30" fill="green"/>
18       </a>
19     </svg>
20   </body>
21 </html>
```

Perl::SVG

```
#!/usr/bin/perl
use strict;
use warnings;

use SVG;

# create an SVG object with a size of 40x40 pixels
my $svg = SVG->new(
    width  => 40,
    height => 40,
);

# add a circle
$svg->circle(
    cx  => 20,
    cy  => 20,
    r   => 18,
);

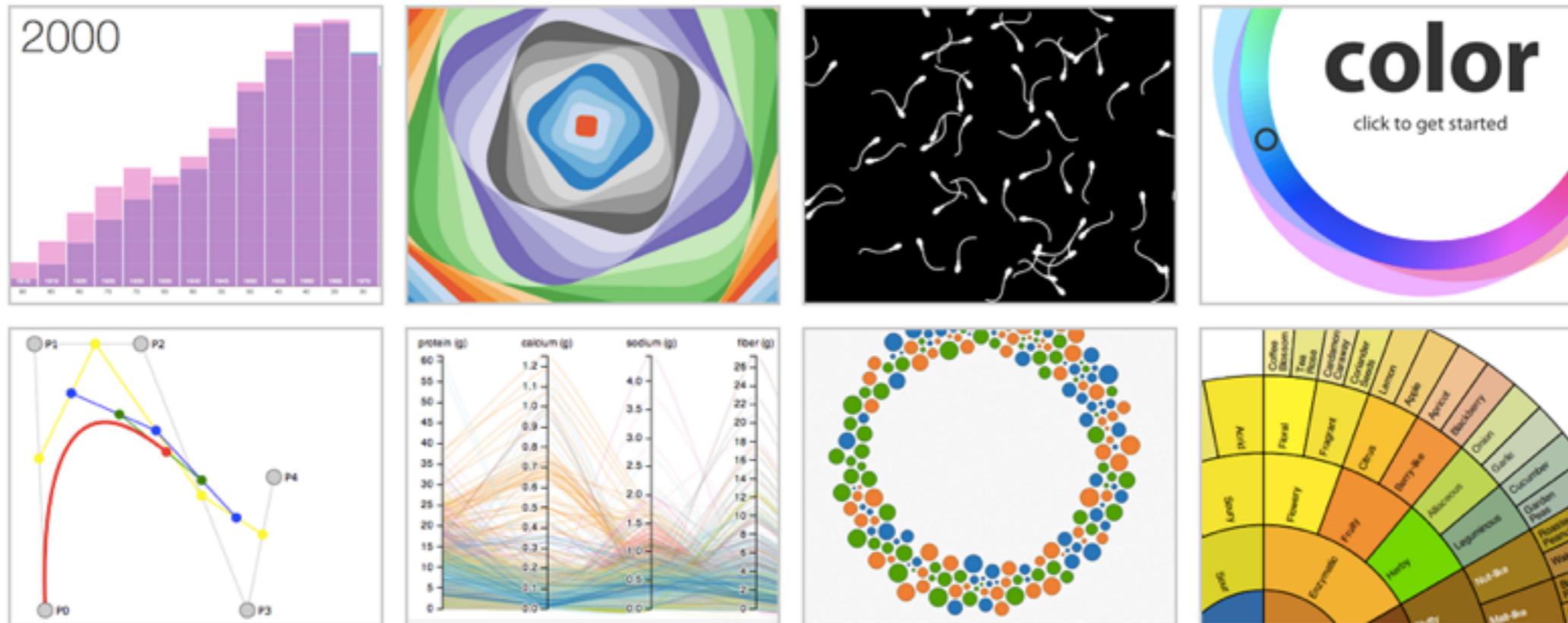
# now render the SVG object, implicitly use svg namespace
print $svg->xmlify;
```

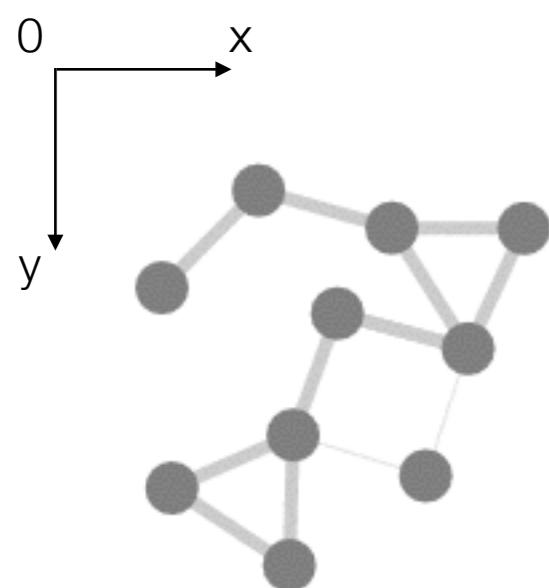
<http://www.w3schools.com/SVG/default.asp>

D3 - <http://d3js.org/>

- javascript

Data-Driven Documents

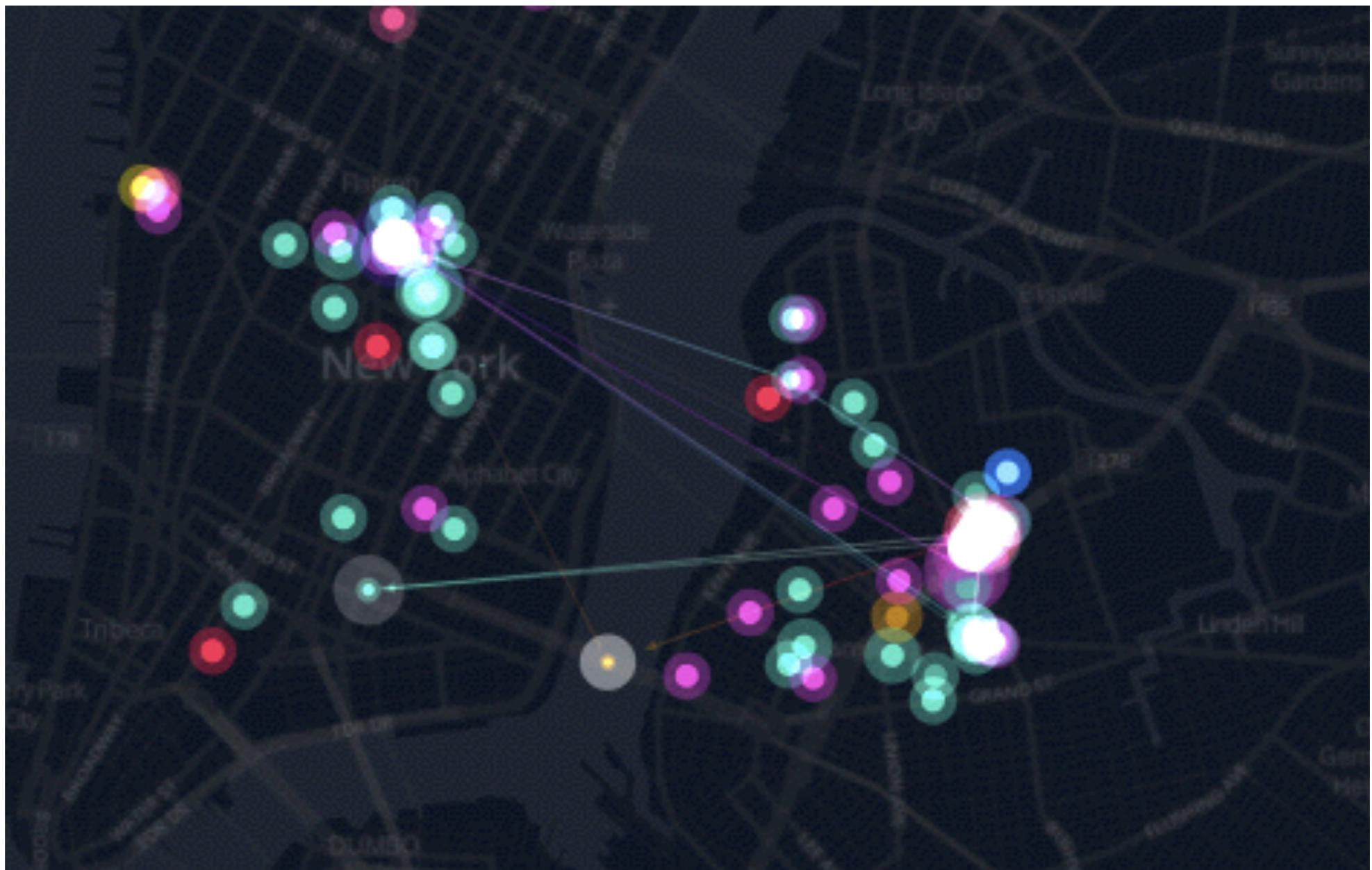




```
<!DOCTYPE html>
<html data-brackets-id="1" lang="en">
  <head data-brackets-id="2">
    <meta data-brackets-id="3" charset="utf-8">
    <title data-brackets-id="4">Playing with D3</title>
    <script data-brackets-id="5" type="text/javascript" src="http://mbostock.github.com/d3/d3.js"></script>
    <script src="http://d3js.org/d3.v2.js"></script>
  </head>
  <body data-brackets-id="6">
    <script data-brackets-id="7" type="text/javascript" src="graph.js"></script>
    <svg>
      <line x1="283" y1="216" x2="322" y2="248" style="stroke: #cccccc; stroke-width: 0.5px;"></line>
      <line x1="283" y1="216" x2="243" y2="248" style="stroke: #cccccc; stroke-width: 5px;"></line>
      <line x1="283" y1="216" x2="272" y2="168" style="stroke: #cccccc; stroke-width: 5px;"></line>
      <line x1="283" y1="216" x2="233" y2="199" style="stroke: #cccccc; stroke-width: 5px;"></line>
      <line x1="322" y1="248" x2="283" y2="279" style="stroke: #cccccc; stroke-width: 0.5px;"></line>
      <line x1="243" y1="248" x2="283" y2="279" style="stroke: #cccccc; stroke-width: 5px;"></line>
      <line x1="243" y1="248" x2="283" y2="279" style="stroke: #cccccc; stroke-width: 5px;"></line>
      <line x1="272" y1="168" x2="233" y2="199" style="stroke: #cccccc; stroke-width: 5px;"></line>
      <line x1="283" y1="279" x2="235" y2="300" style="stroke: #cccccc; stroke-width: 5px;"></line>
      <line x1="283" y1="279" x2="276" y2="329" style="stroke: #cccccc; stroke-width: 5px;"></line>
      <line x1="180" y1="234" x2="187" y2="285" style="stroke: #cccccc; stroke-width: 5px;"></line>
      <line x1="187" y1="285" x2="235" y2="300" style="stroke: #cccccc; stroke-width: 5px;"></line>
      <line x1="235" y1="300" x2="276" y2="329" style="stroke: #cccccc; stroke-width: 5px;"></line>
      <circle r="10" cx="283" cy="216" style="fill: #808080;"></circle>
      <circle r="10" cx="322" cy="248" style="fill: #808080;"></circle>
      <circle r="10" cx="243" cy="248" style="fill: #808080;"></circle>
      <circle r="10" cx="272" cy="168" style="fill: #808080;"></circle>
      <circle r="10" cx="233" cy="199" style="fill: #808080;"></circle>
      <circle r="10" cx="283" cy="279" style="fill: #808080;"></circle>
      <circle r="10" cx="180" cy="234" style="fill: #808080;"></circle>
      <circle r="10" cx="187" cy="285" style="fill: #808080;"></circle>
      <circle r="10" cx="235" cy="300" style="fill: #808080;"></circle>
      <circle r="10" cx="276" cy="329" style="fill: #808080;"></circle>
    </svg>
  </body>
</html>
```

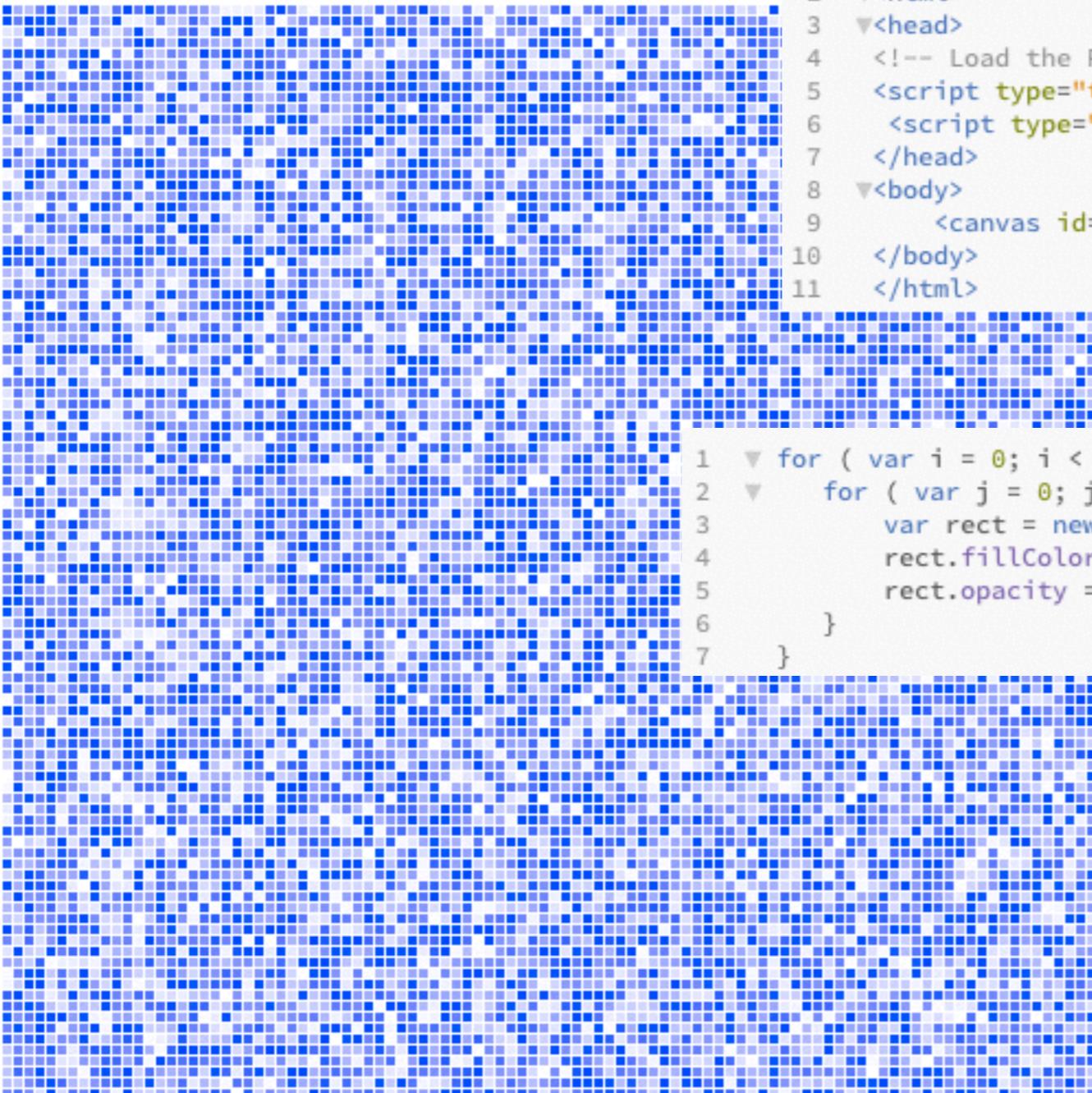
paper.js - <http://paperjs.org>

- javascript



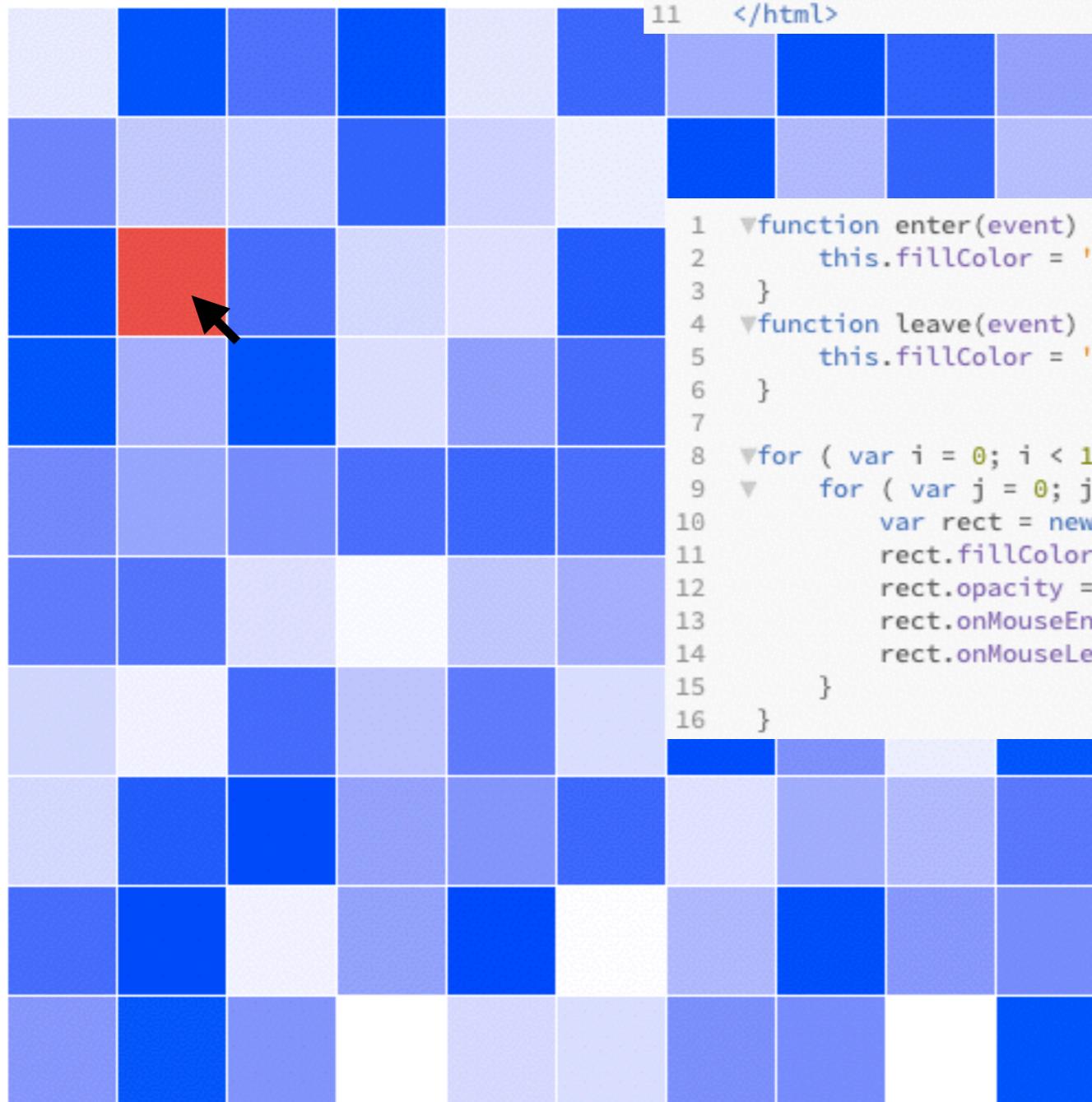
index.html

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <!-- Load the Paper.js library -->
5  <script type="text/javascript" src="paperjs/dist/paper.js"></script>
6  <script type="text/paperscript" src="heatmap.js" canvas="myCanvas"></script>
7  </head>
8  <body>
9   <canvas id="myCanvas" resize></canvas>
10 </body>
11 </html>
```



heatmap.js

```
1  < for ( var i = 0; i < 100; i++ ) {
2    < for ( var j = 0; j < 100; j++ ) {
3      var rect = new Path.Rectangle(new Point(i*5,j*5), new Point(i*5+4,j*5+4))
4      rect.fillColor = 'blue'
5      rect.opacity = Math.random()
6    }
7  }
```



```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <!-- Load the Paper.js library -->
5  <script type="text/javascript" src="paperjs/dist/paper.js"></script>
6  <script type="text/paperscript" src="heatmap_hover.js" canvas="myCanvas"></script>
7  </head>
8  <body>
9    <canvas id="myCanvas" resize></canvas>
10 </body>
11 </html>
```

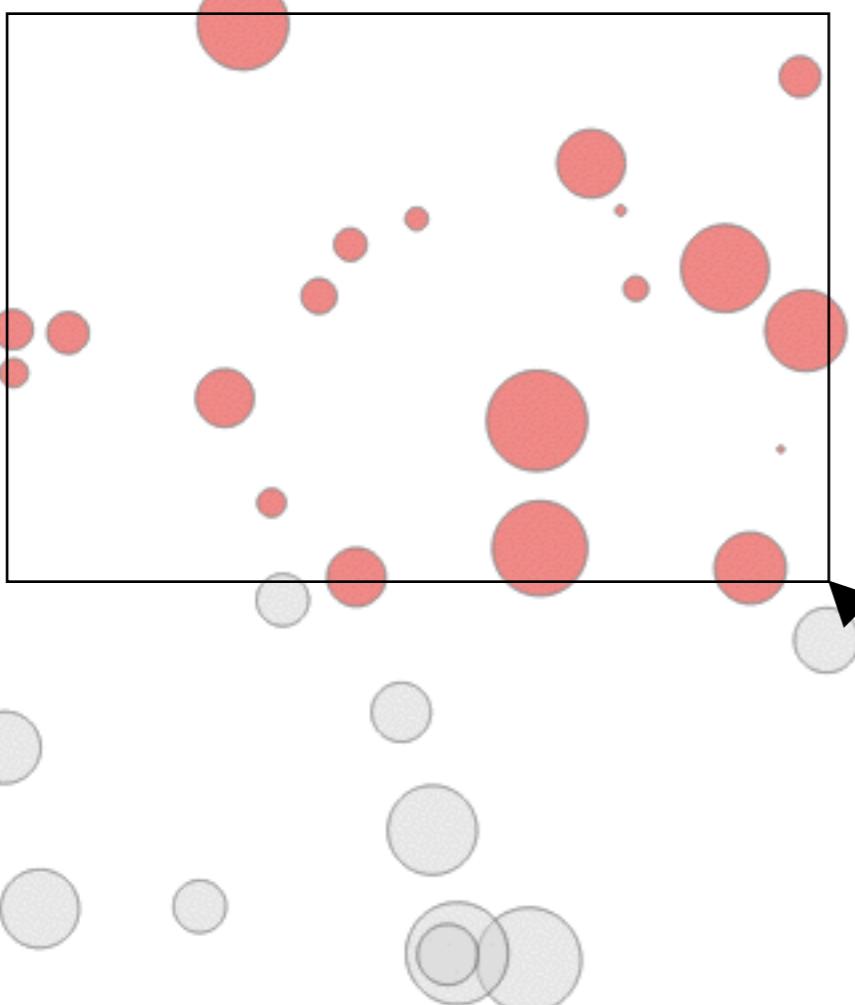
index.html

```
1  <function enter(event) {
2      this.fillColor = 'red'
3  }
4  <function leave(event) {
5      this.fillColor = 'blue'
6  }
7
8  <for ( var i = 0; i < 10; i++ ) {
9      <for ( var j = 0; j < 10; j++ ) {
10          var rect = new Path.Rectangle(new Point(i*50,j*50), new Point(i*50+49,j*50+49))
11          rect.fillColor = 'blue'
12          rect.opacity = Math.random()
13          rect.onMouseEnter = enter
14          rect.onMouseLeave = leave
15      }
16  }
```

heatmap.js

index.html

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <!-- Load the Paper.js library -->
5  <script type="text/javascript" src="paperjs/dist/paper.js"></script>
6  <script type="text/paperscript" src="drag_select.js" canvas="myCanvas"></script>
7  </head>
8  <body>
9    <canvas id="myCanvas" resize></canvas>
10 </body>
11 </html>
```



drag_select.js

```
1  var objects = new Array();
2  for (i = 0; i < 50; i++) {
3    var circle = new Path.Circle(new Point(Math.random()*500, Math.random()*500), Math.random()*20);
4    circle.fillColor = 'lightgrey';
5    circle.strokeColor = 'black';
6    circle.opacity = 0.5;
7    objects.push(circle);
8  }
9
10 var mouseDownPosition;
11 var mouseDragPosition;
12 var selectionRectangle;
13 function onMouseDown(event) {
14   mouseDownPosition = event.point;
15 }
16 function onMouseDrag(event) {
17   mouseDragPosition = event.point;
18   if (selectionRectangle) { selectionRectangle.remove(); }
19   selectionRectangle = new Path.Rectangle(mouseDownPosition, mouseDragPosition);
20   selectionRectangle.strokeColor = 'grey';
21
22   // This is slow because has to go over all objects with every pixel dragged
23   for (var i = 0; i < objects.length; i++ ) {
24     if ( selectionRectangle.bounds.intersects(objects[i].bounds) ) {
25       objects[i].fillColor = 'red';
26     } else {
27       objects[i].fillColor = 'lightgrey';
28     }
29   }
30 }
31 function onMouseUp(event) {
32   selectionRectangle.remove();
33 }
```

Overview of lecture

- A. Why visual analytics?
- B. Data visualization
 - Data foundations
 - Human perception foundations
 - Visualization foundations and techniques
- C. Visualization evaluation
- D. Tools of the trade
- E. Examples
- F. Exercises

E. Examples

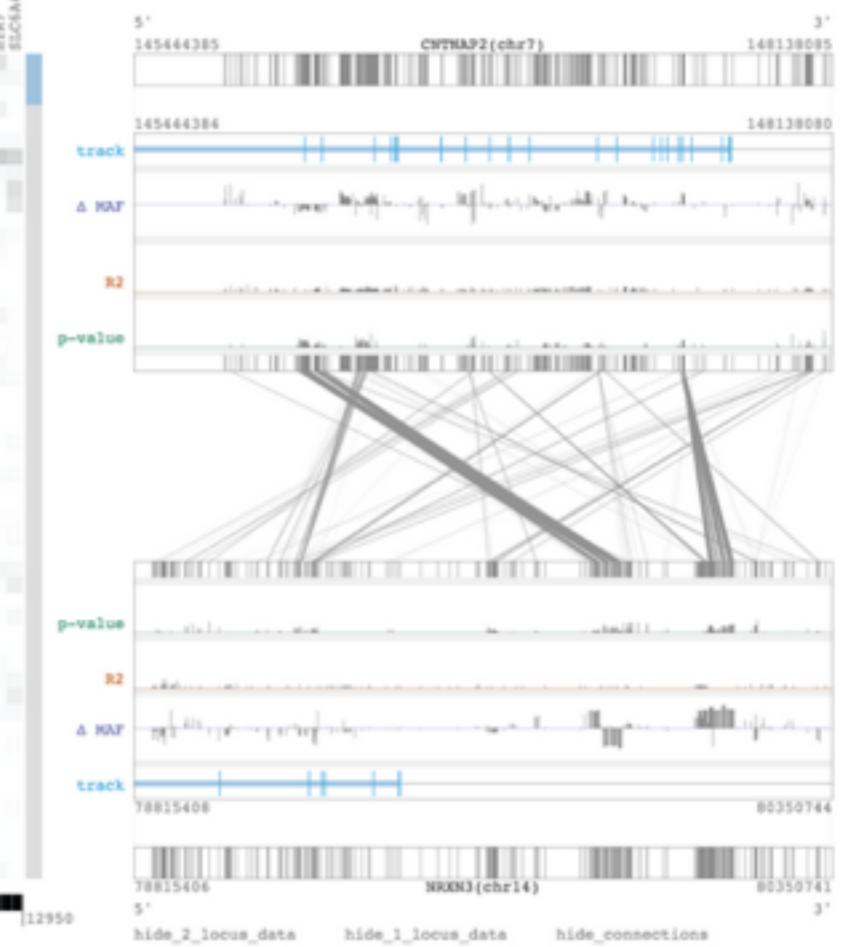
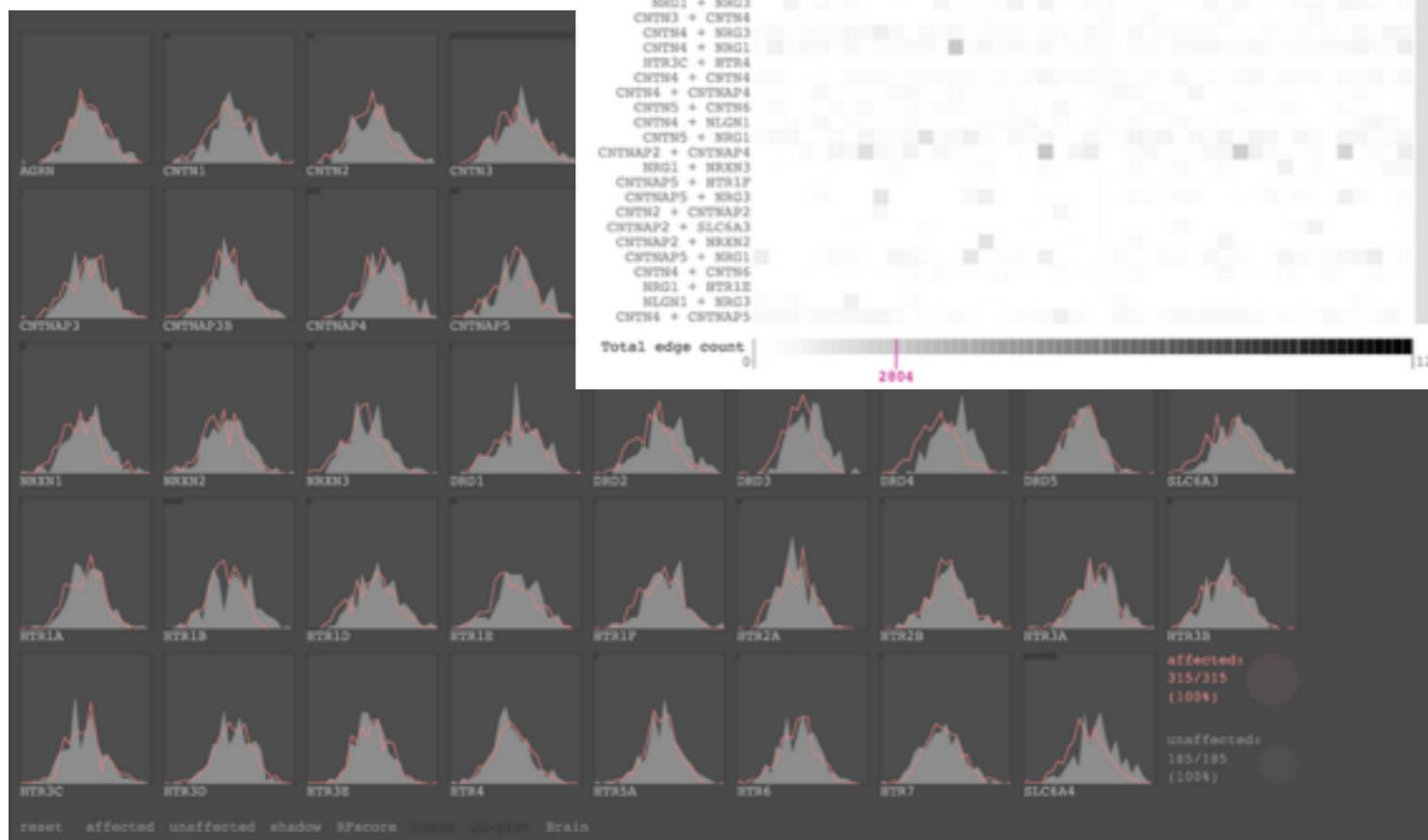
Data exploration
Data filtering
User-guided analysis

Data exploration

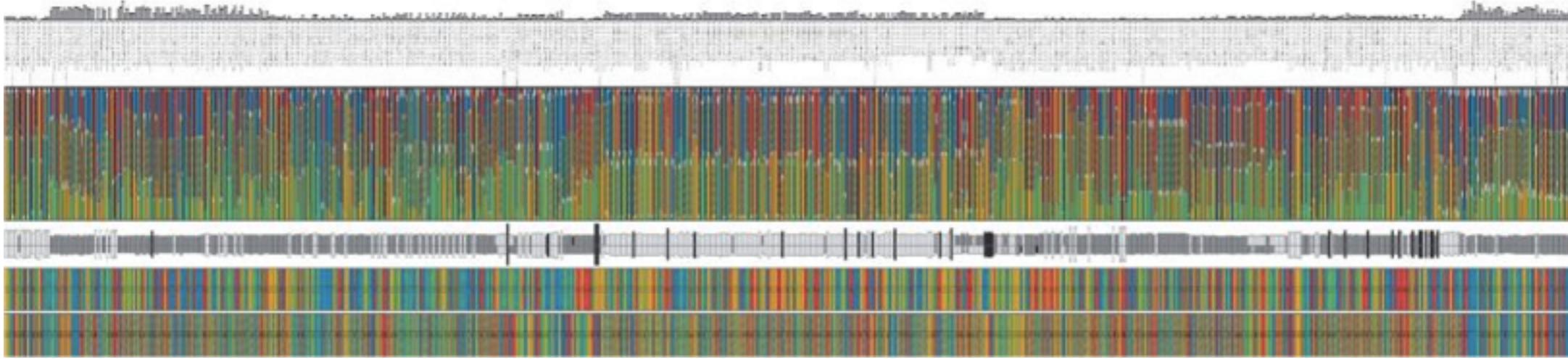


Aracari

Bartlett C et al. BMC Bioinformatics (2012)

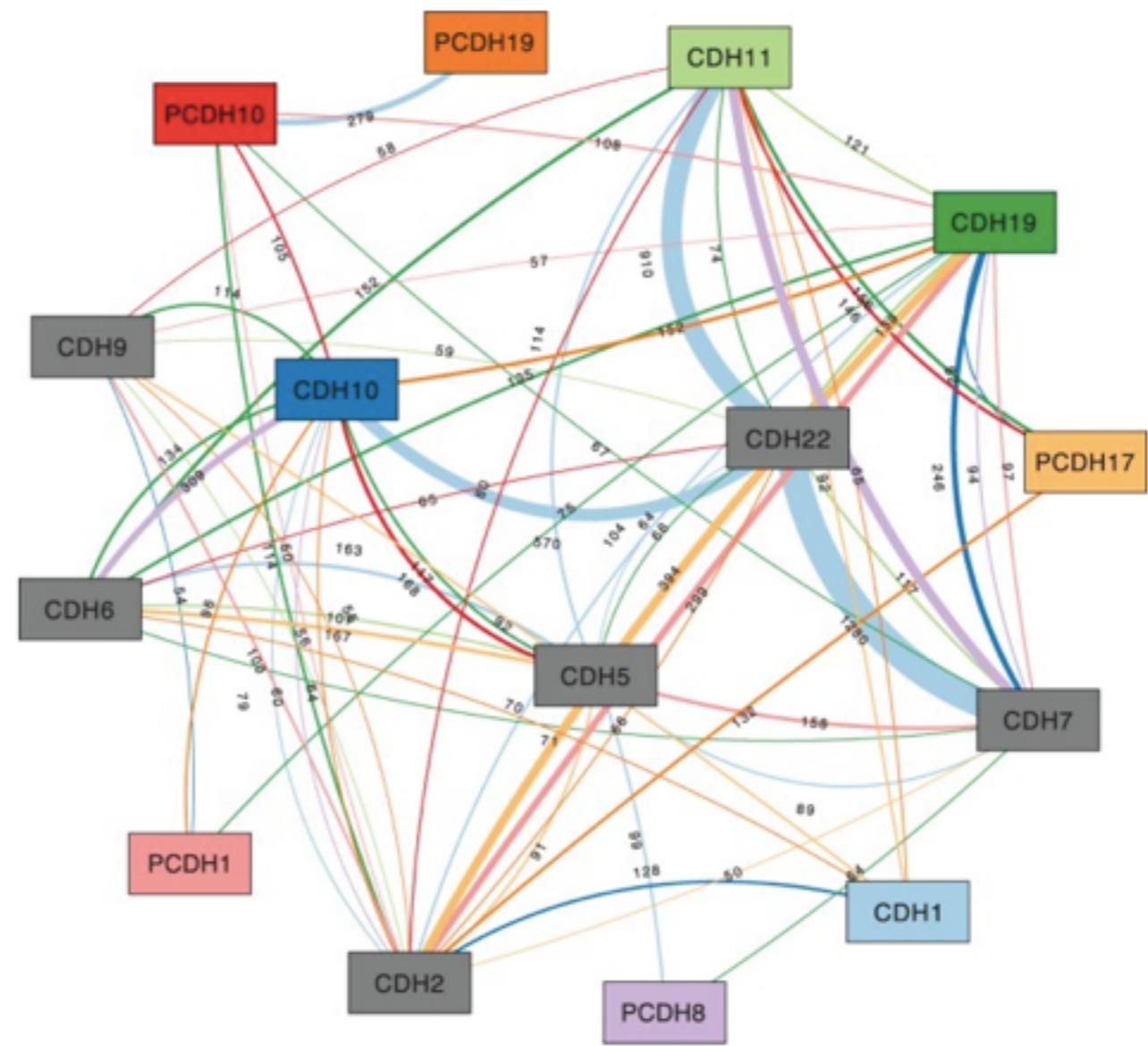


Ryo Sakai



Reveal

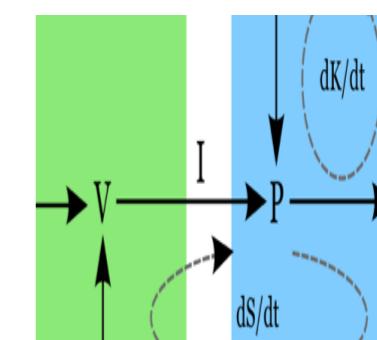
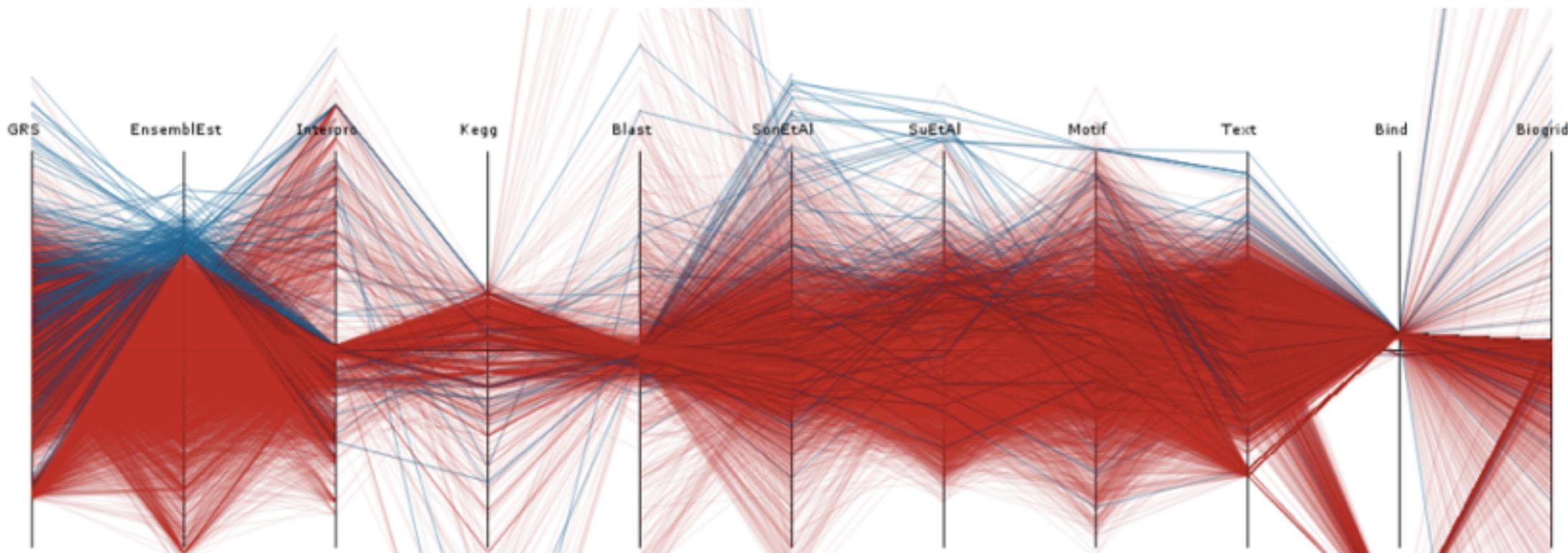
Jäger, G et al. Bioinformatics (2012)



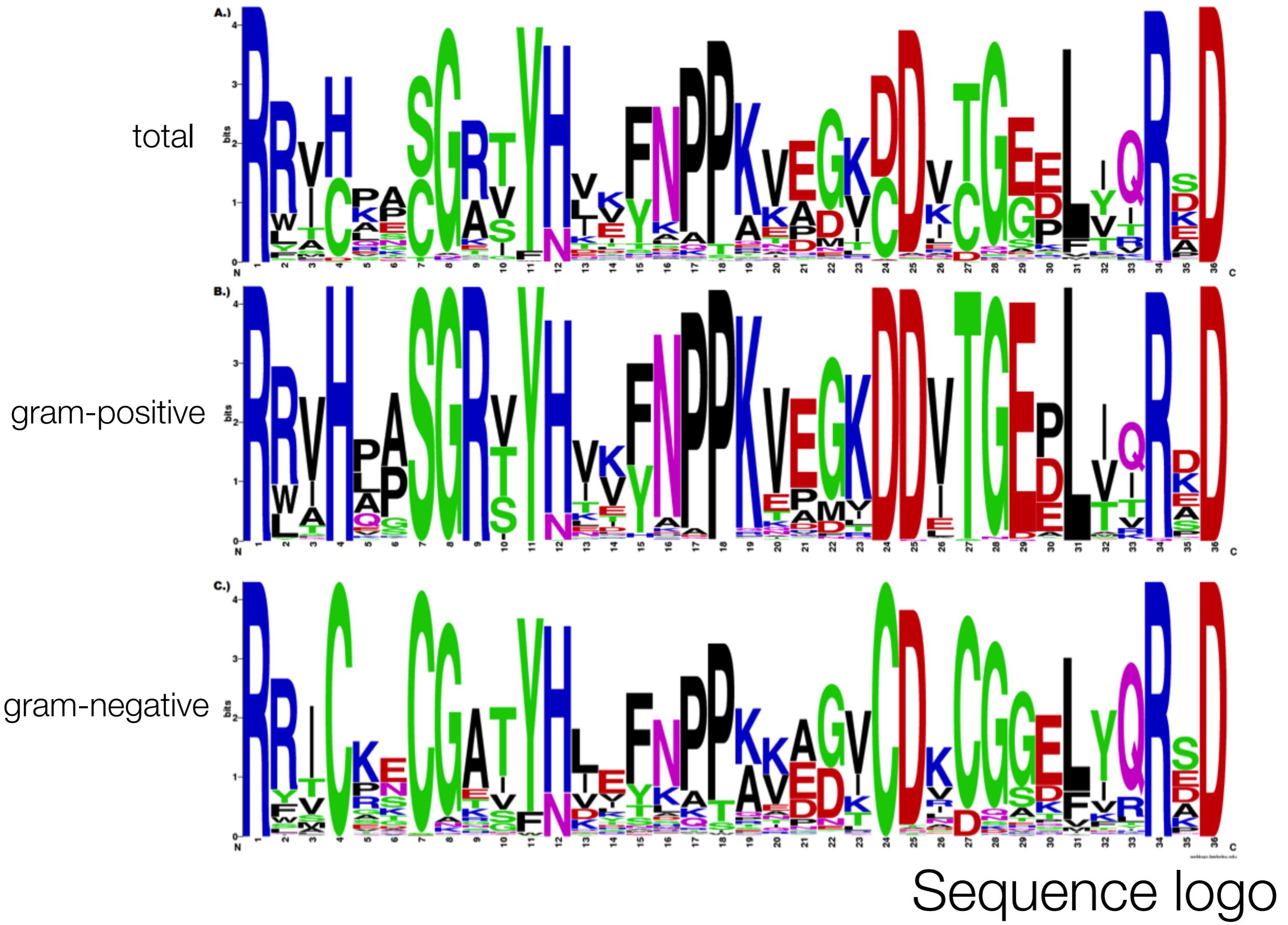
ParCoord

Boogaerts T et al. IEEE International Conference on
Bioinformatics & Bioengineering (2012)

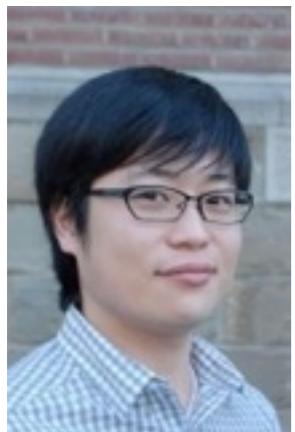
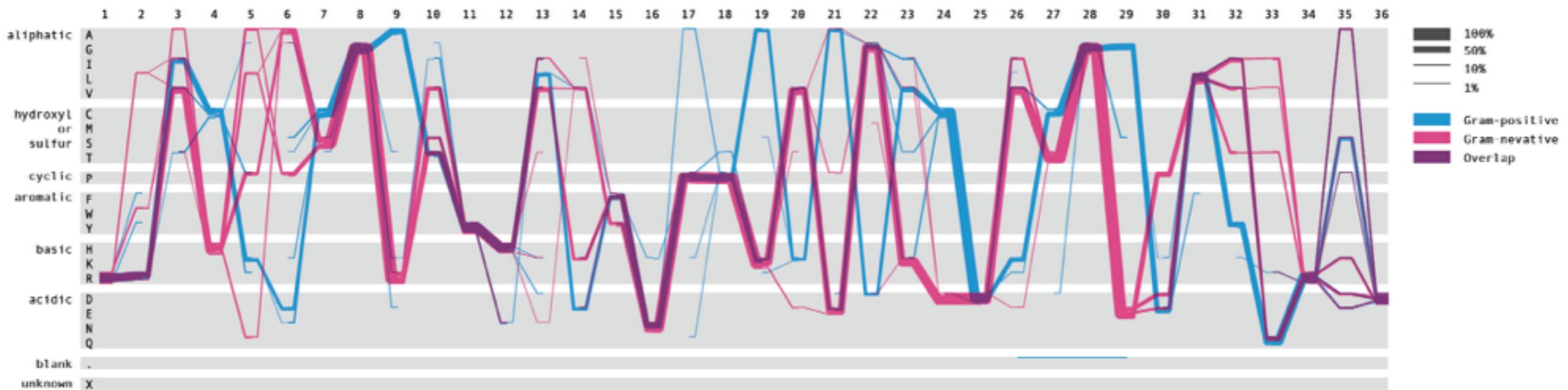
Endeavour gene prioritization



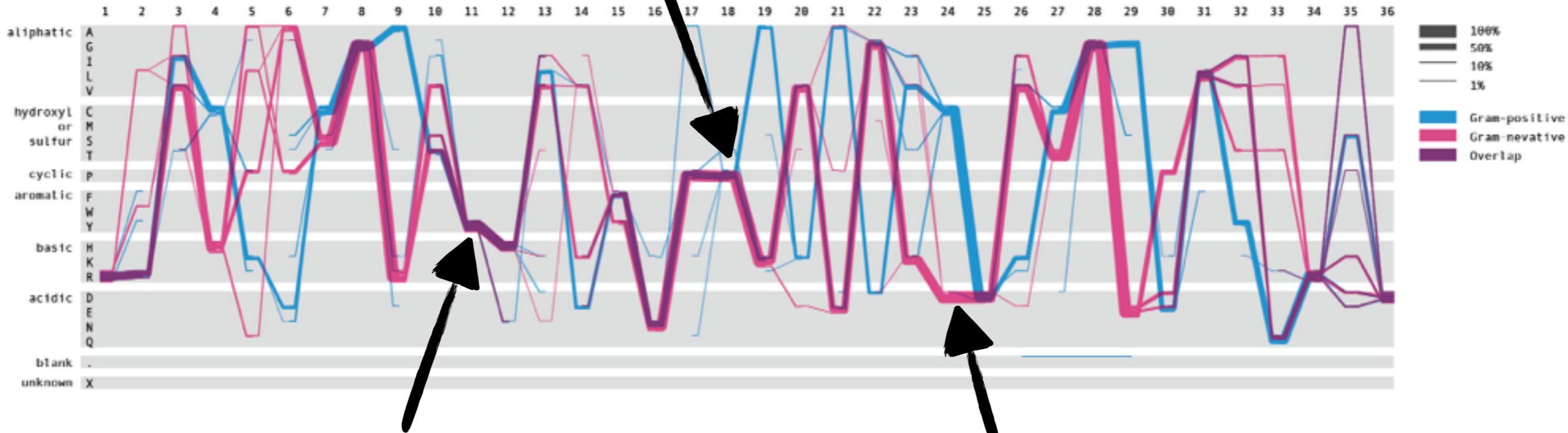
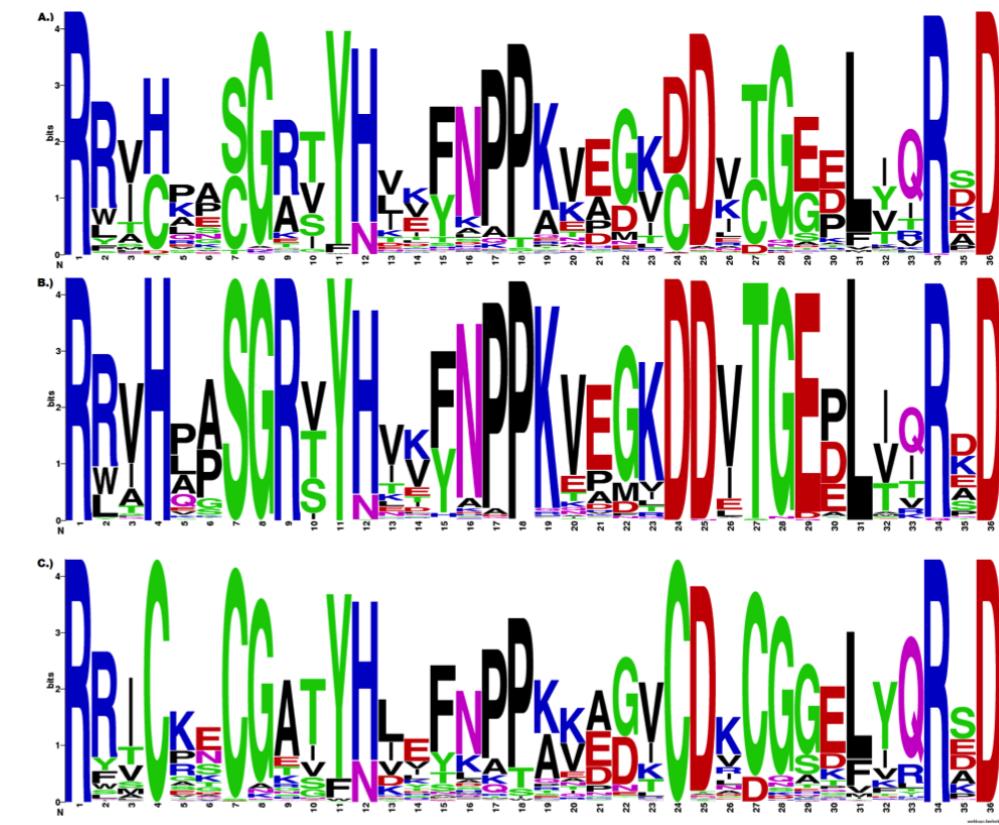
Thomas Boogaerts



Sequence Diversity Diagram



subgroup

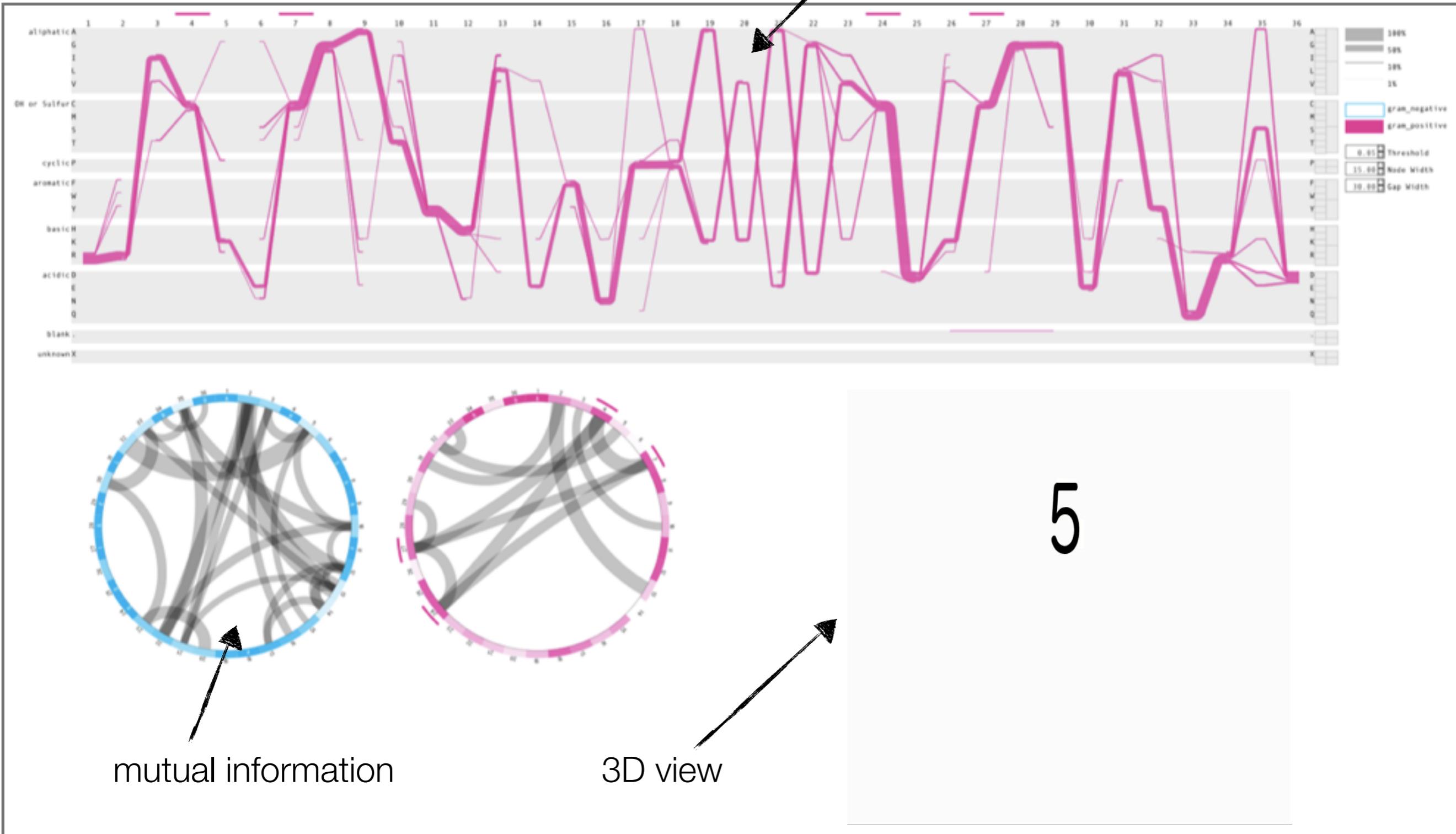


similarity

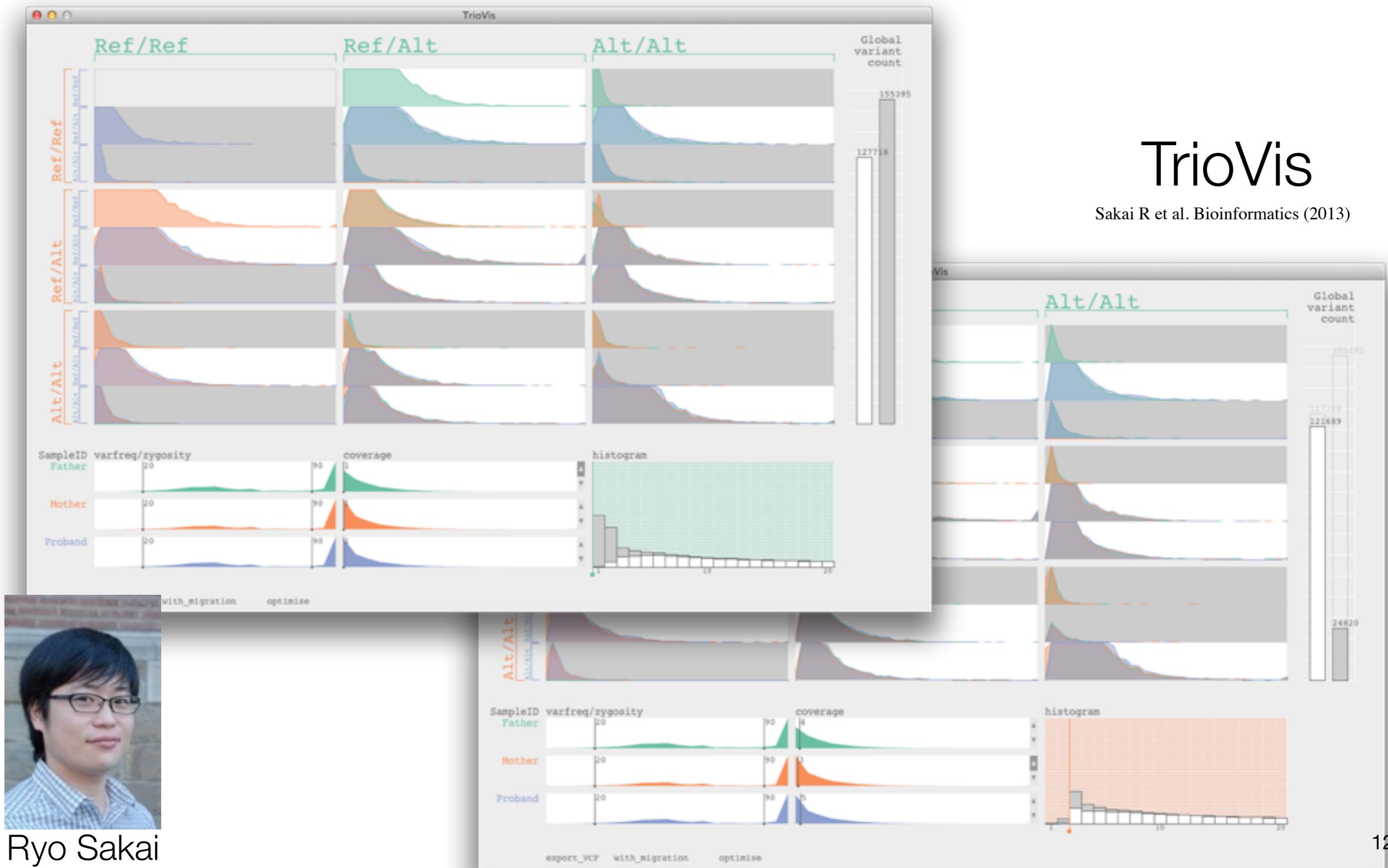
difference

Seagull

sequence diversity diagram

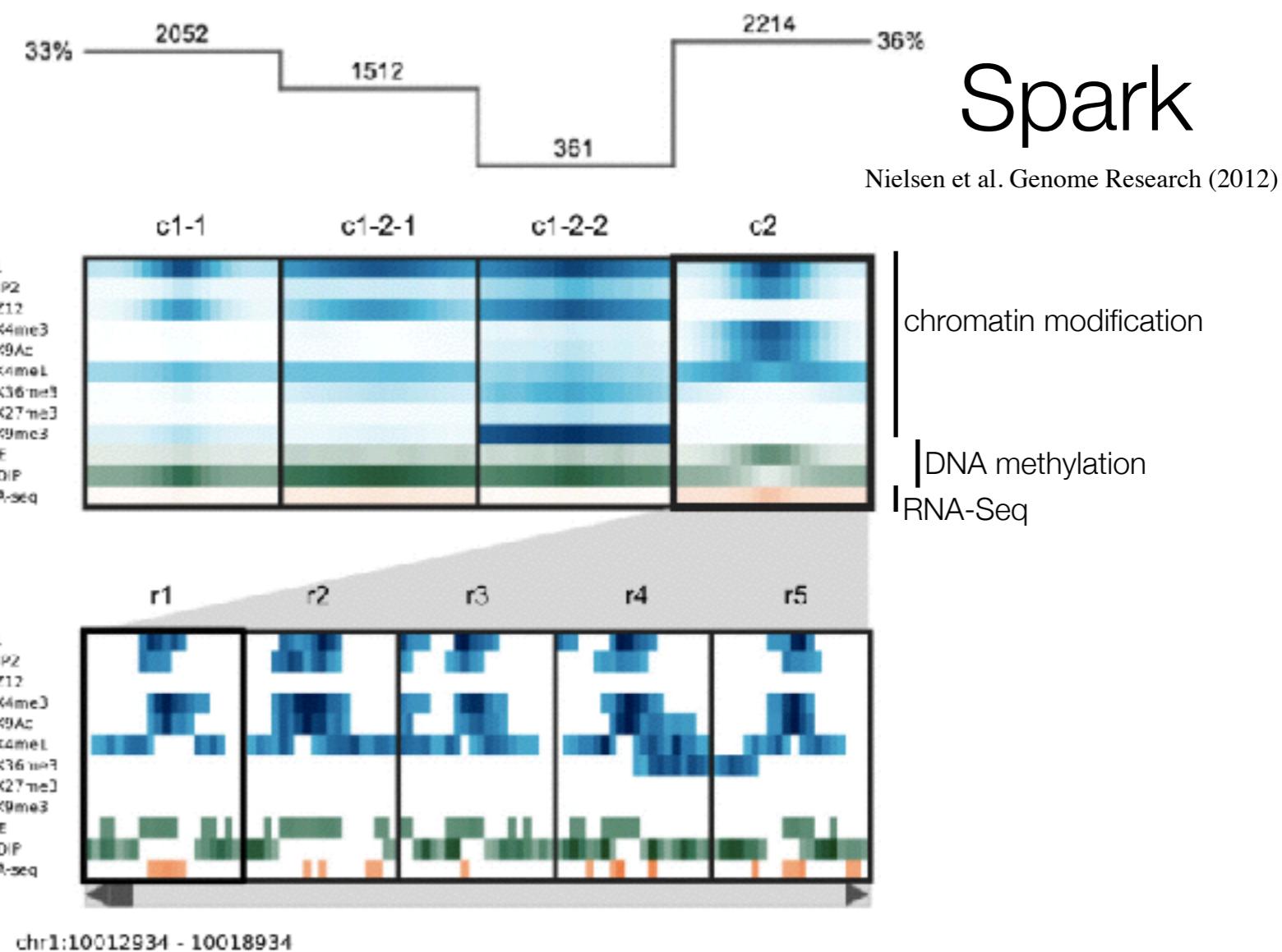
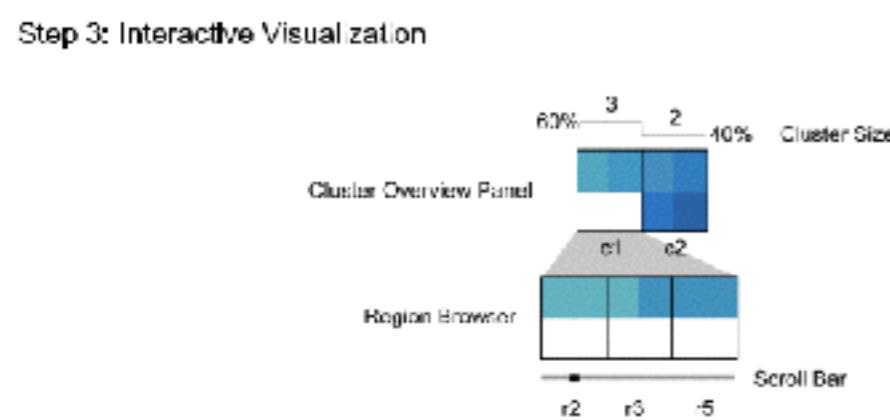
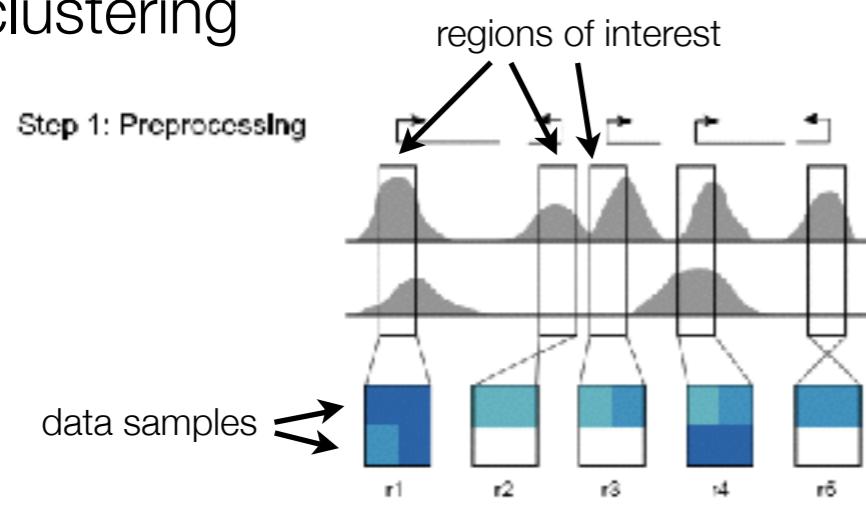


Data filtering (visual parameter setting)



User-guided analysis

clustering



Overview of lecture

- A. Why visual analytics?
- B. Data visualization
 - Data foundations
 - Human perception foundations
 - Visualization foundations and techniques
- C. Visualization evaluation
- D. Tools of the trade
- E. Examples
- F. Exercises

F. Exercises

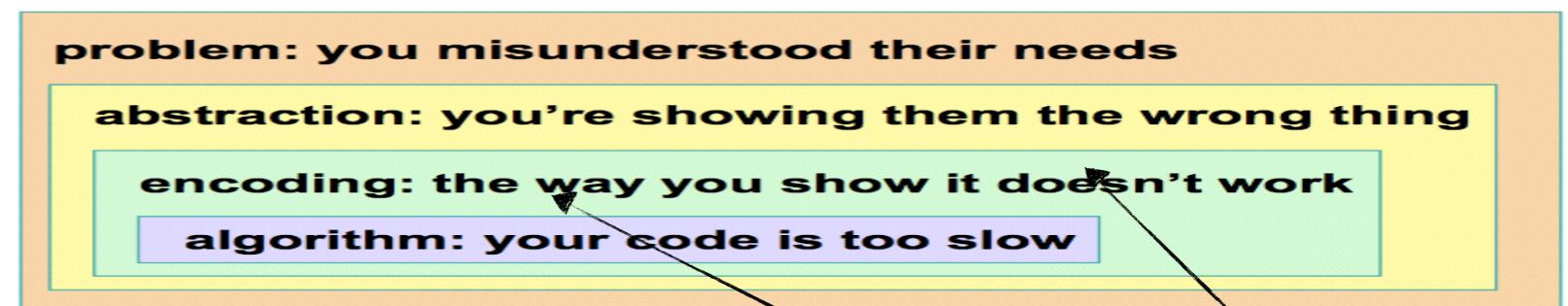
Redesign exercise

(Thanks Martin! Vizbi 2012 practical session - <http://bit.ly/1OGOTt2>)

- Every row will be given a different re-design exercise.
- Sit together as a group and discuss what's wrong with the picture.
- Present this to the rest of the class, covering:
 - A brief description of the figure
 - What are its strong and weak points
 - What requires improvement?
 - *Why* does it require improvement?
 - Do you have an idea on how to make it better? (maybe make sketch)

Programming SVG

- Preamble: to use perl and SVG for visualization:
- Write a perl script that spits out a text file, e.g. named something.svg
- That file should look like this:
- For help, see
 - <https://developer.mozilla.org/en-US/docs/Web/SVG>
 - <https://developer.mozilla.org/en-US/docs/Web/SVG/Element>
 - <https://developer.mozilla.org/en-US/docs/Web/SVG/Attribute>
- Open the SVG file in your favourite web-browser (except IE...)



change these bits...

- Given the GFF file /homes/evopserver/lectures/Visualization/structural_variation.gvf:

- Create a graphic like this one
- Or can you think of an alternative (or even better) visual encoding for this data?

```

1 ▼function enter(event) {
2      this.fillColor = 'red'
3  }
4 ▼function leave(event) {
5      this.fillColor = 'blue'
6  }
7
8 ▼for ( var i = 0; i < 10; i++ ) {
9   ▼ for ( var j = 0; j < 10; j++ ) {
10       var rect = new Path.Rectangle(new Point(i*50,j*50), new Point(i*50+49,j*50+49))
11       rect.fillColor = 'blue'
12       rect.opacity = Math.random()
13       rect.onMouseEnter = enter
14       rect.onMouseLeave = leave
15   }
16 }
```

- How would you visualize the data from /homes/evopserver/lectures/R-Intro/Features.txt?
 - Sketch, then implement

