Bioinformatics II
Winter Term 2016/17

universitätbonn

Rheinische
Friedrich-Wilhelms-
Universität Bonn

Chapter 6:
Visualization Design

Jun.-Prof. Dr.-Ing. Thomas Schultz

URL: http://cg.cs.uni-bonn.de
E-Mail: schultz@cs.uni-bonn.de
Office: Friedrich-Ebert-Allee 144, 53113 Bonn

December 6/20, 2016

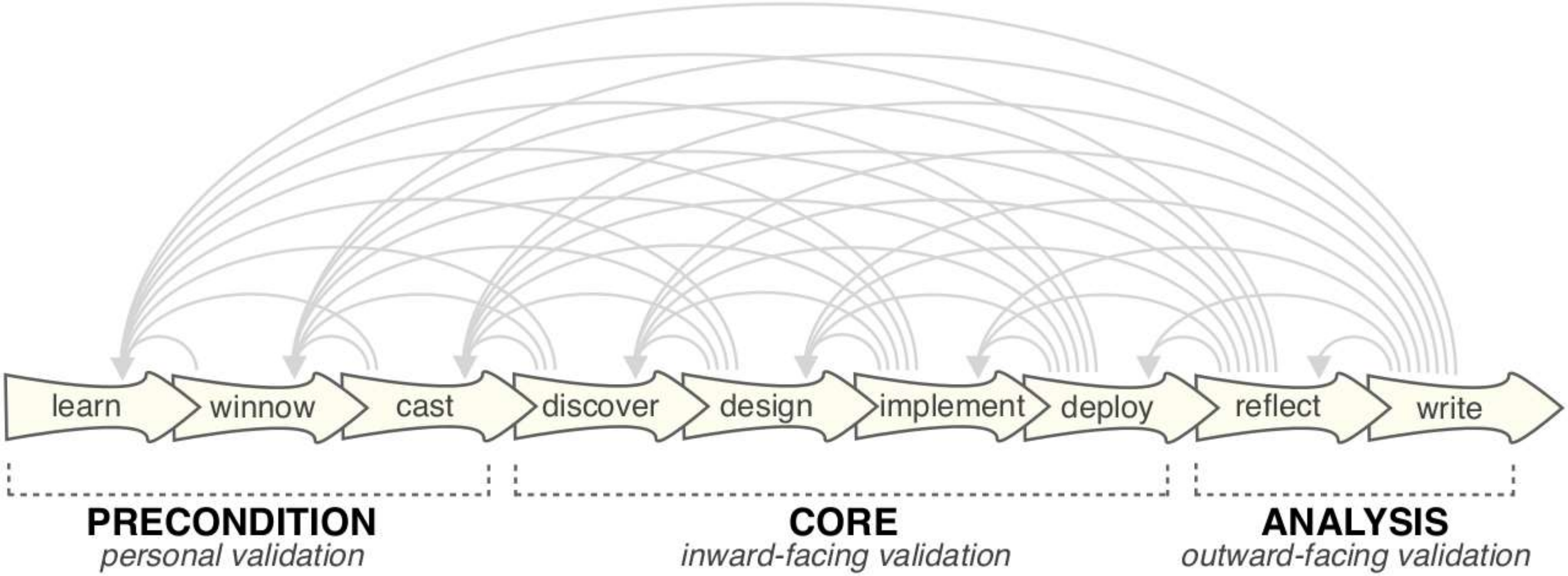
Motivation

- **Lecture so far:**
  - Human visual perception
  - Visualization of multidimensional data
  - Dimensionality Reduction
  - Visualization of graphs
- How to use this knowledge to **produce software that is useful for collaborators** (e.g., biologists):
  - Visualization Design Process
  - Classification of Marks and Channels
  - Visualization Pipelines
  - Example Systems

Section 6.1:
Visualization Design Process

Methodology for Visualization Design

- Nine-stage design methodology [Sedlmair et al. 2012]
  - – Guidelines for developing visualization software prototypes for scientific collaborators

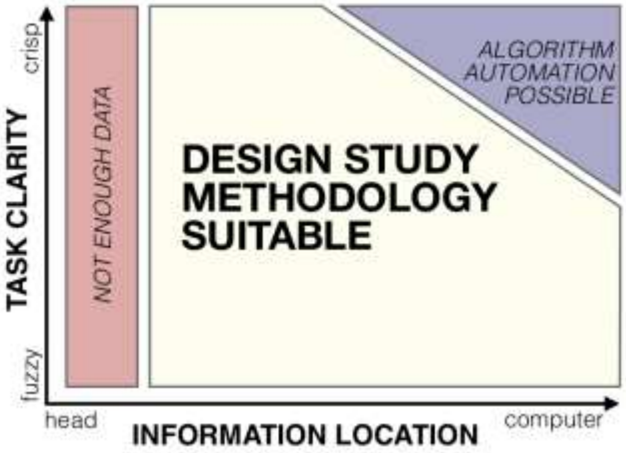


Learn

- **Perception:** Be aware of limitations of the human visual system
- **Techniques:** Have a broad perspective of available tools
  - Explains “superficial” treatment of alternative techniques within this lecture
  - Assignments provide opportunity to become familiar with widely applicable software packages
- **Methodology:** Know how to build collaborations and create systems
  - Today’s lecture!
- **Read up** on literature relevant to your project!

Winnow

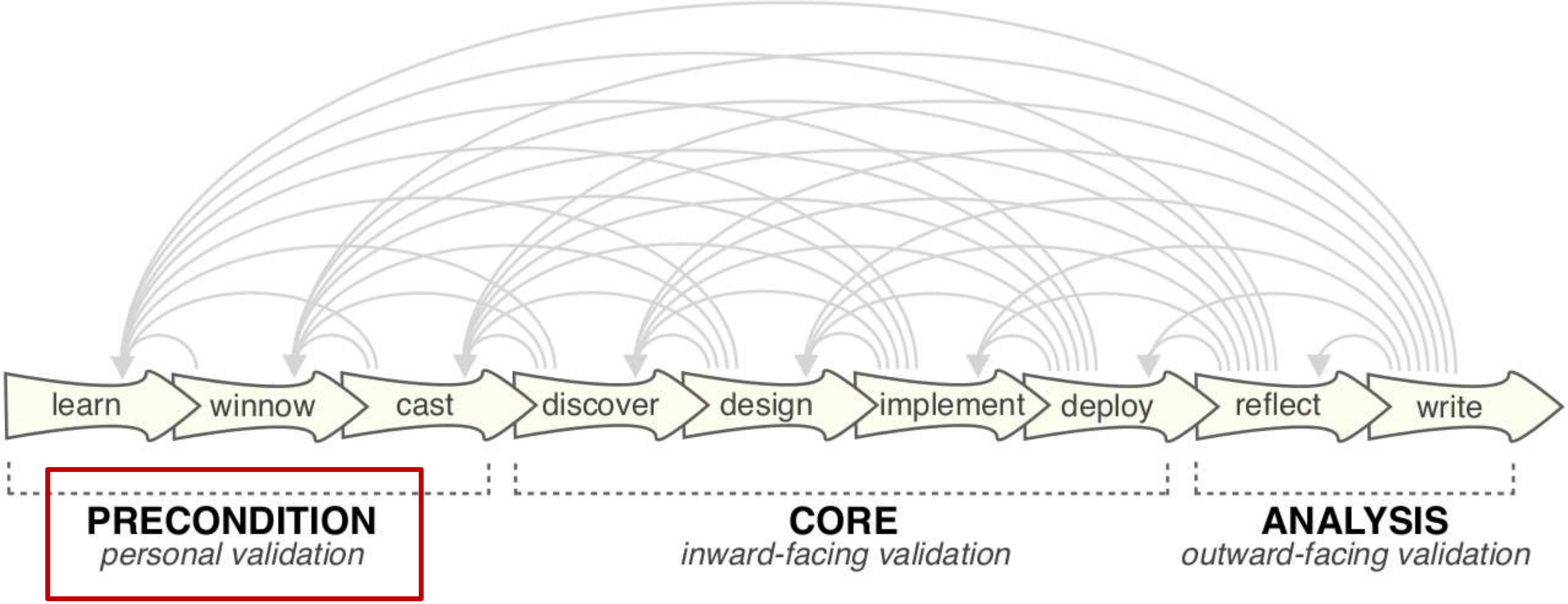
- Selection of collaborators / projects
  - "Talk with many, stay with few"
- **Practical Criteria:**
  - Does data exist, is it enough, can I have it?
  - Can I and can they devote enough time?
- **Intellectual Criteria:**
  - Is the problem interesting and relevant?
  - Do existing solutions suffice?
  - Is visualization suitable?
- **Interpersonal Criteria**



Cast

- Be aware of **important roles** in the project
  - **Front line analyst** (e.g., PhD student)
    - User of your tool
    - Performs day-to-day analysis of the data
    - Often generates the data
  - **Gatekeeper** (e.g., Professor)
    - Grants and denies access to data (and permission to publish about it)
    - Decides on how resources are spent
  - **Fellow tool builders**
    - Avoid relying on their characterization of the problem alone

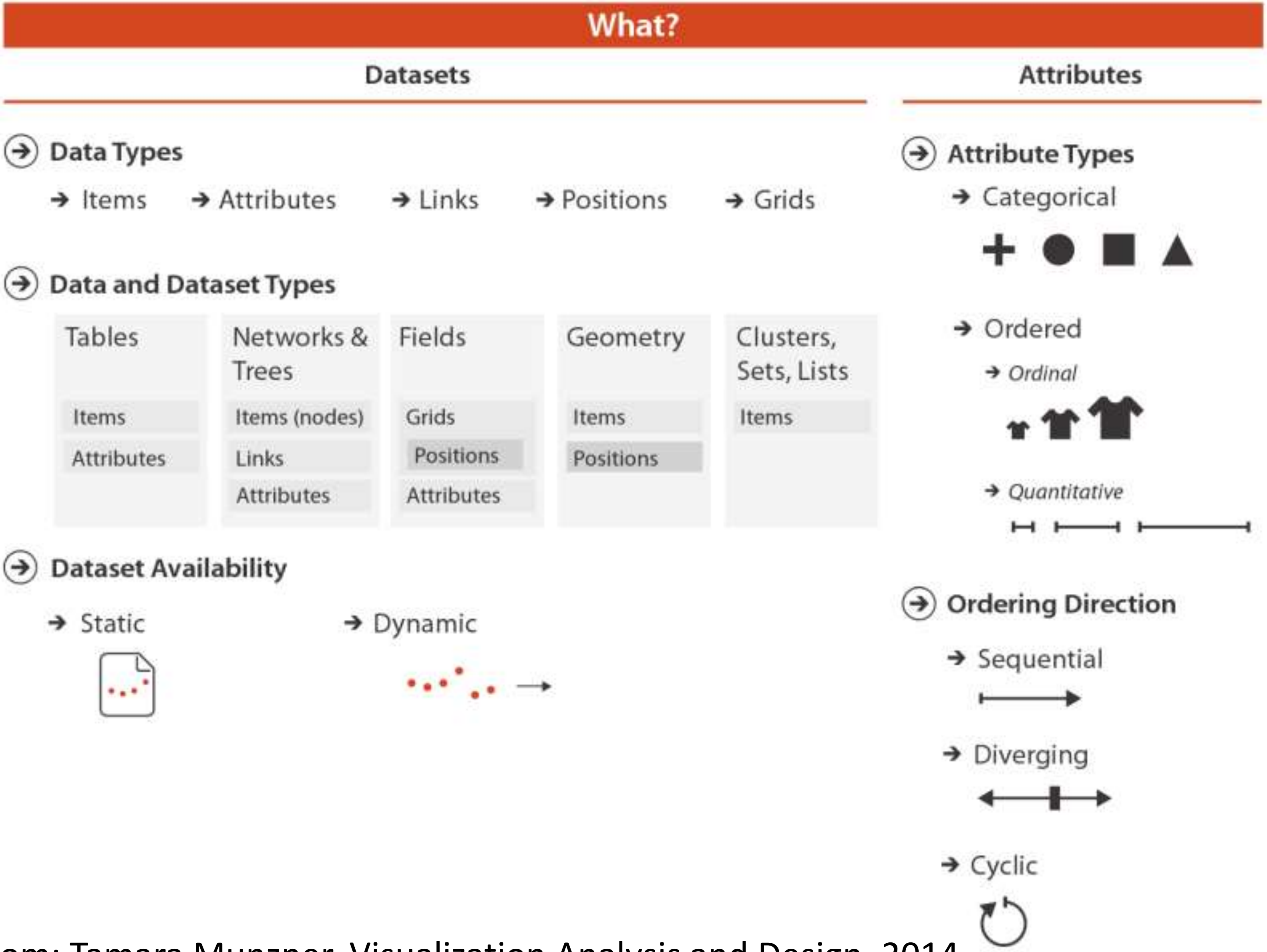
Core Stage


learn
winnow
cast
discover
design
implement
deploy
reflect
write
PRECONDITION
personal validation
CORE
inward-facing validation
ANALYSIS
outward-facing validation

Discover

- Understand the domain user’s **research questions, data analysis problems, and tasks**
- **Methods:**
  – *Talking:* Iterative (get feedback on your abstraction)
  – *Fly-on-the-wall:* Silently observing
  – *Contextual inquiries:* Interrupt and ask
  – *Reading* domain literature

Data Abstraction



What?

Datasets

Data Types

- Items
- Attributes
- Links
- Positions
- Grids

Data and Dataset Types

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

Dataset Availability

- Static
- Dynamic

Attributes

Attribute Types

- Categorical

  + ● ■ ▲

- Ordered
  - *Ordinal*
  - *Quantitative*

Ordering Direction

- Sequential
- Diverging
- Cyclic

Figure from: Tamara Munzner, Visualization Analysis and Design, 2014

Data Abstraction: Dataset Types

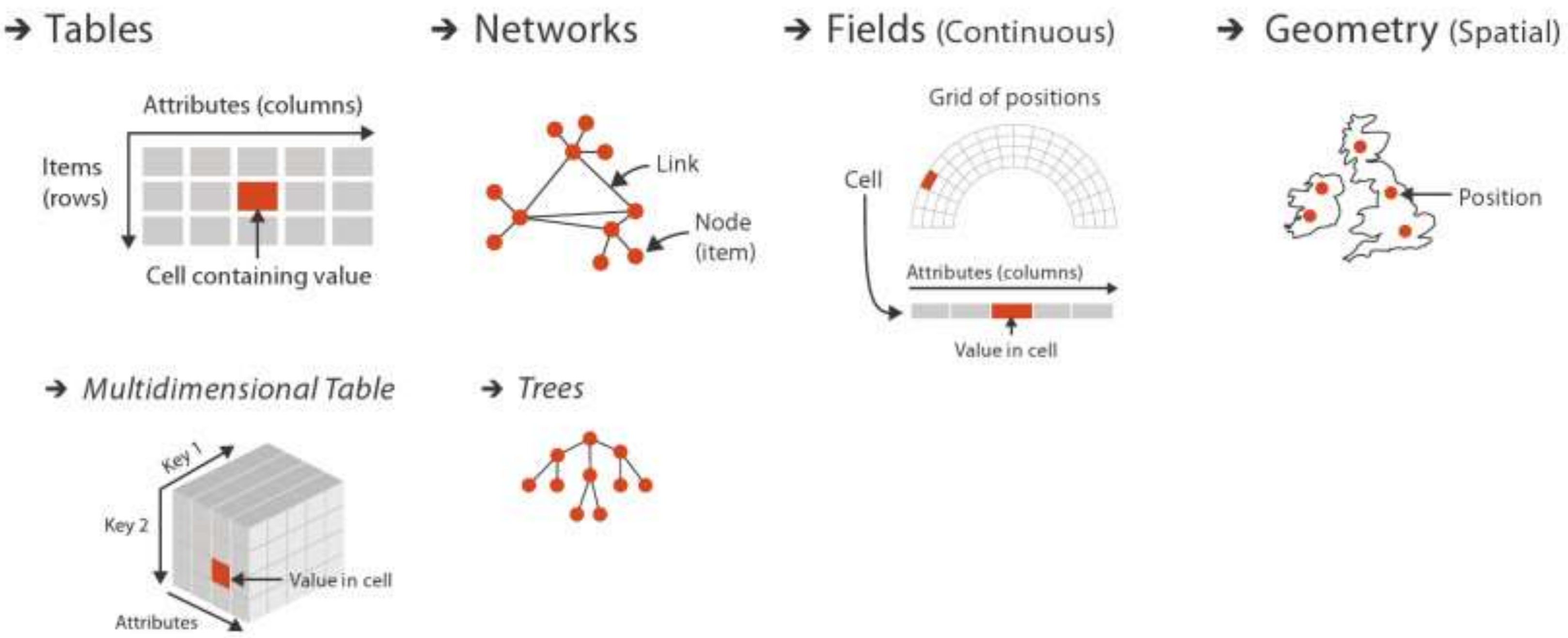
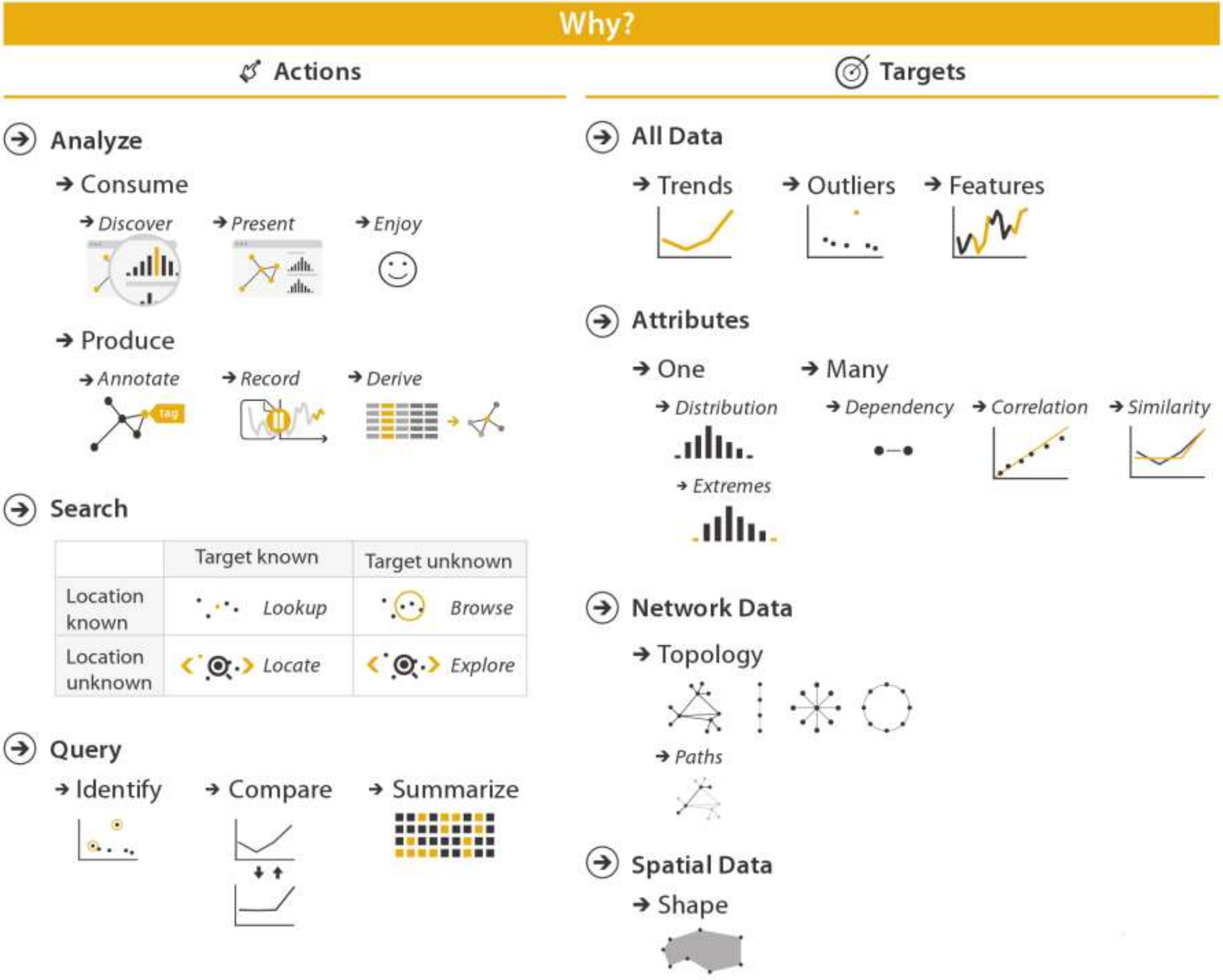


Figure from: Tamara Munzner, Visualization Analysis and Design, 2014

Task Abstraction



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

Images from Tamara Munzner, Visualization Analysis and Design, 2014

Design

- **Goal** of visualization is to create a solution
  - for a specific task
  - performed by a specific group of users
  - on specific data
- **Satisfy** users, do not attempt to **optimize**
  - contradictory objectives are the norm

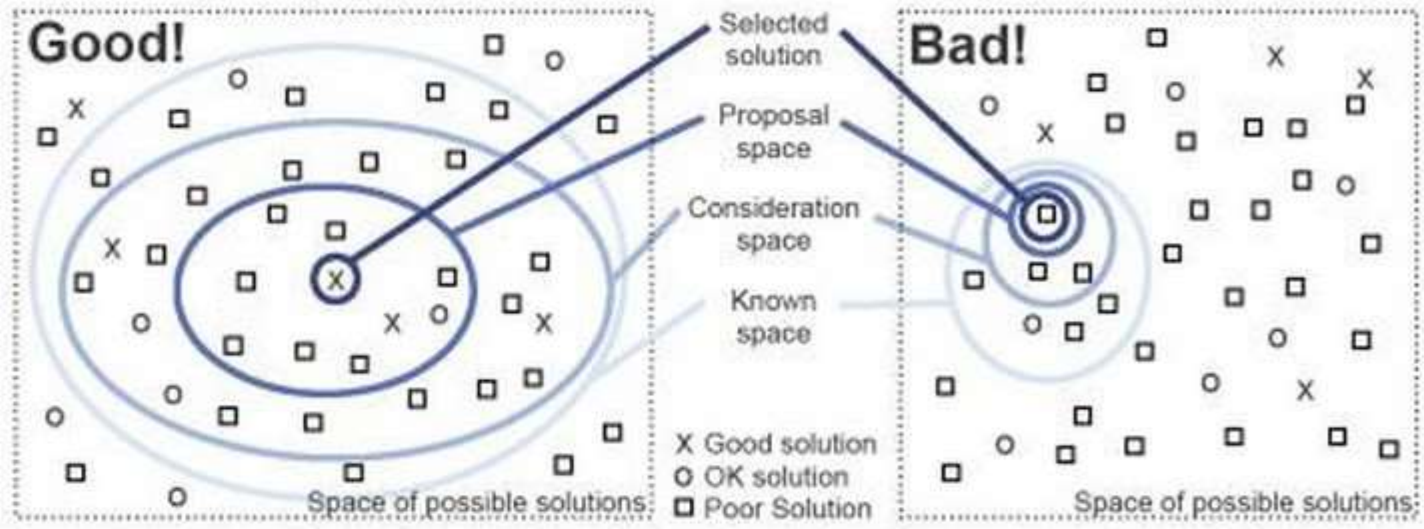
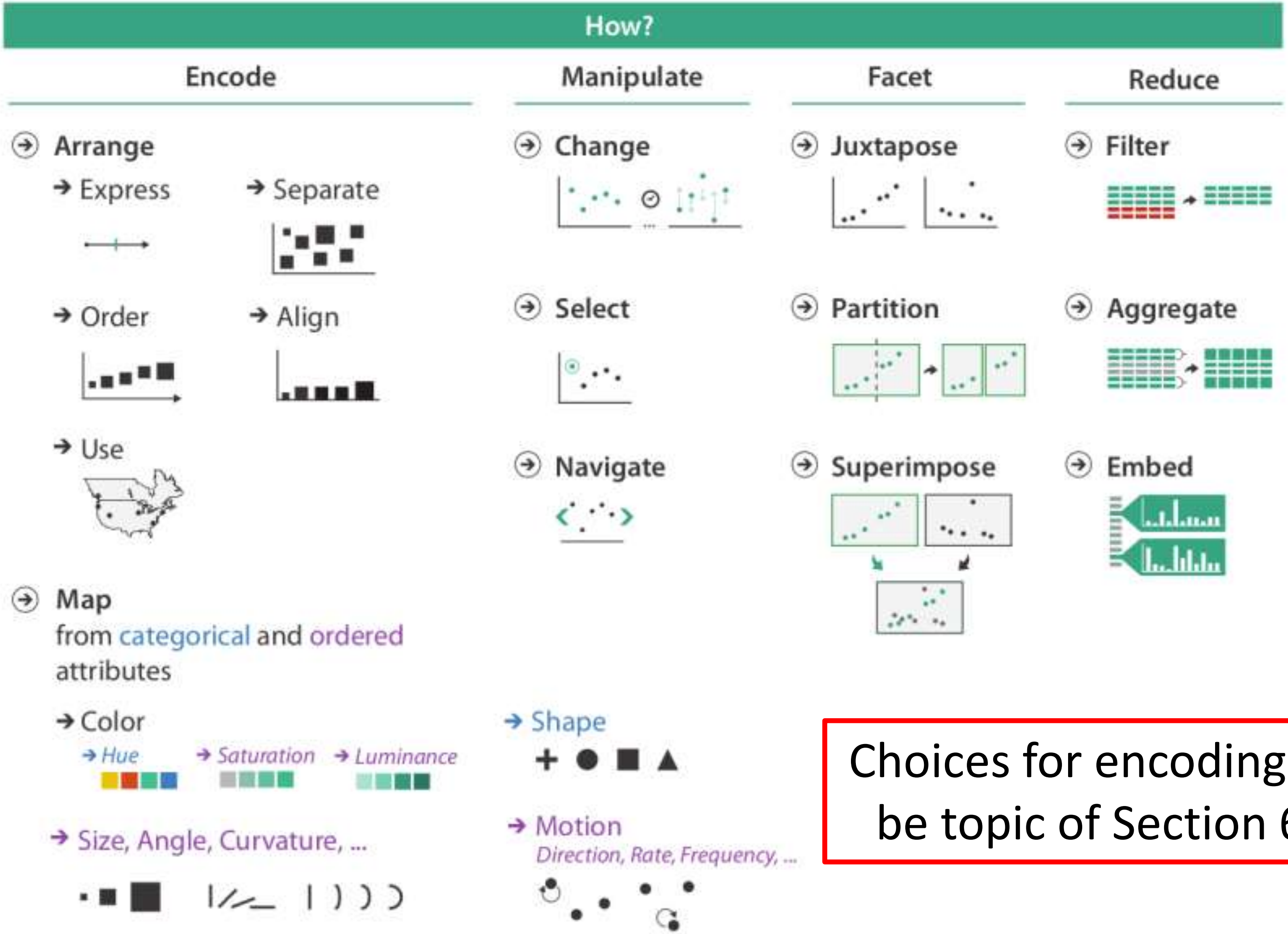


Figure from: Tamara Munzner, Visualization Analysis and Design, 2014

How to Construct a Vis Idiom



Choices for encoding will be topic of Section 6.2

Figure from: Tamara Munzner, Visualization Analysis and Design, 2014

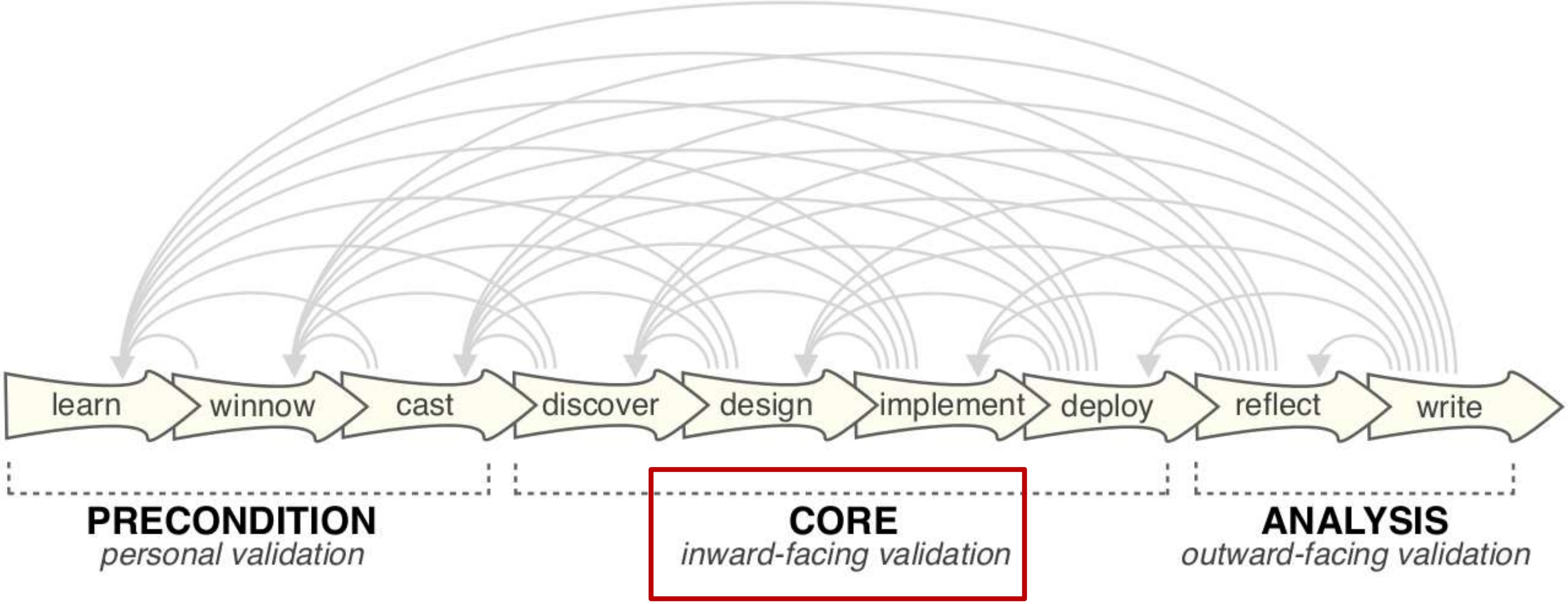
Implement

- **Prototypes** help confirm that you've correctly analyzed the users' needs
  - Avoid investing lots of time in creating a software that ends up not being used
  - Start with paper prototypes / "mock-ups"
  - Write "throw-away" code simply and rapidly
  - Make use of existing infrastructure
- Find middle ground on **usability**
  - With too little usability, tool will not be used
  - Usability should not push utility out of focus

Deploy

- Release your software "into the wild"
- **Validate** benefit from tool:
  - Faster or more accurate analysis
  - New insights (which might enable a higher degree of automation)
- **Case study:** Solve specific real problem with real users and real data
  - **Usage scenario:** Real data, analysis done by developer of tool
  - **Pair analytics:** Insight gained in tandem by computer / data and domain scientist

Analysis Stage

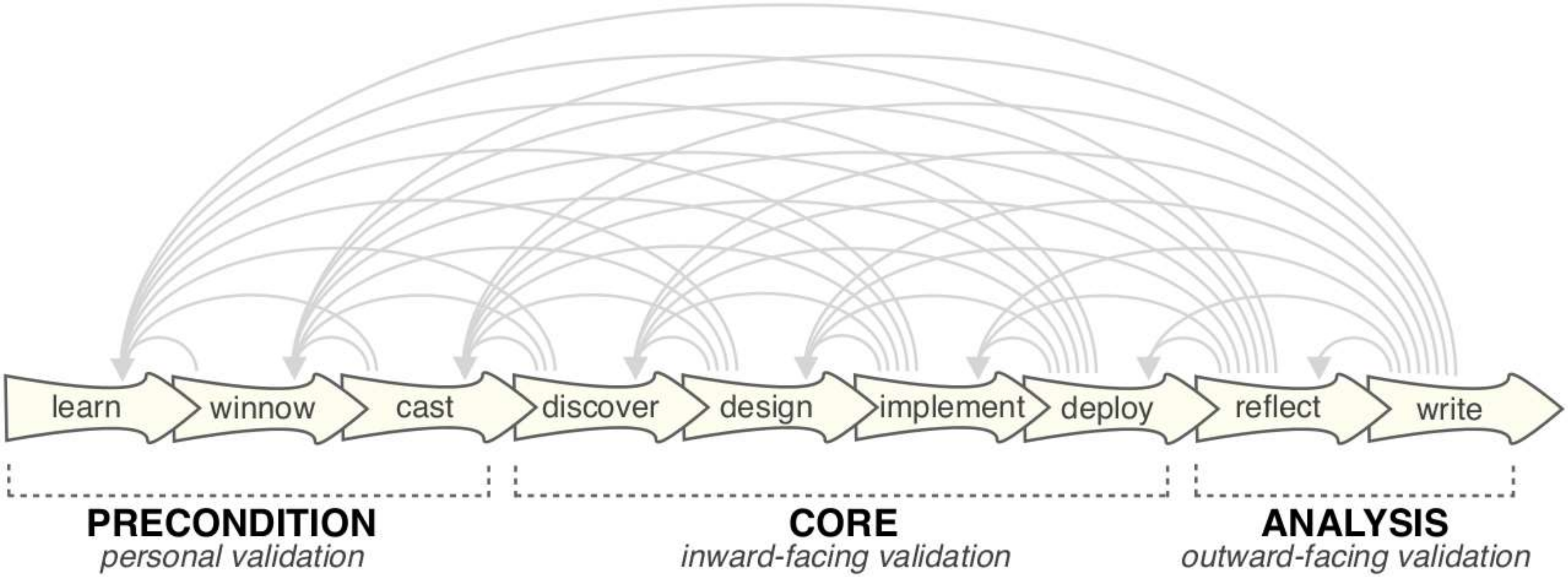


Reflect / Write

- **Reflect** on what you’ve learned from building the tool
  - How to do better next time?
  - Confirm, refine, reject, propose guidelines
- In an academic context, **report** on your tool in a thesis / scientific paper
  - Describe the tool and evaluation
  - Justify design choices
  - Discuss lessons learned

Summary: Nine-Stage Process

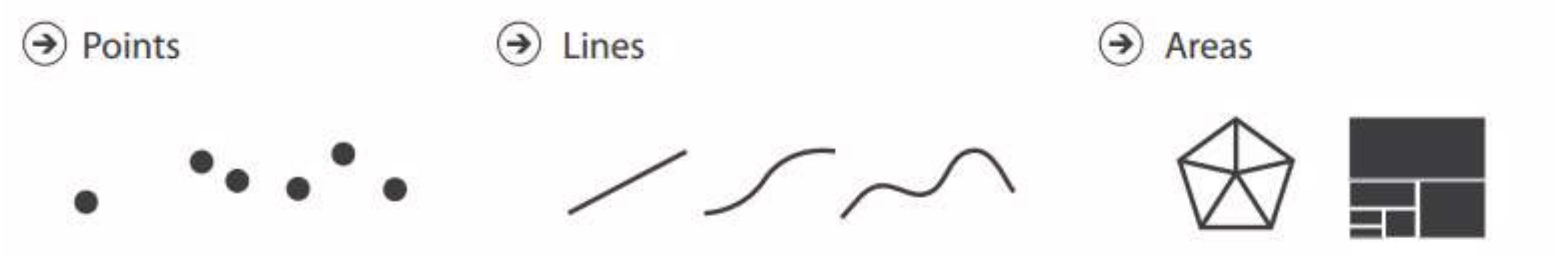
- Nine-stage design methodology provides useful guidance / structure for interdisciplinary collaborations
- Helps avoid missing important steps



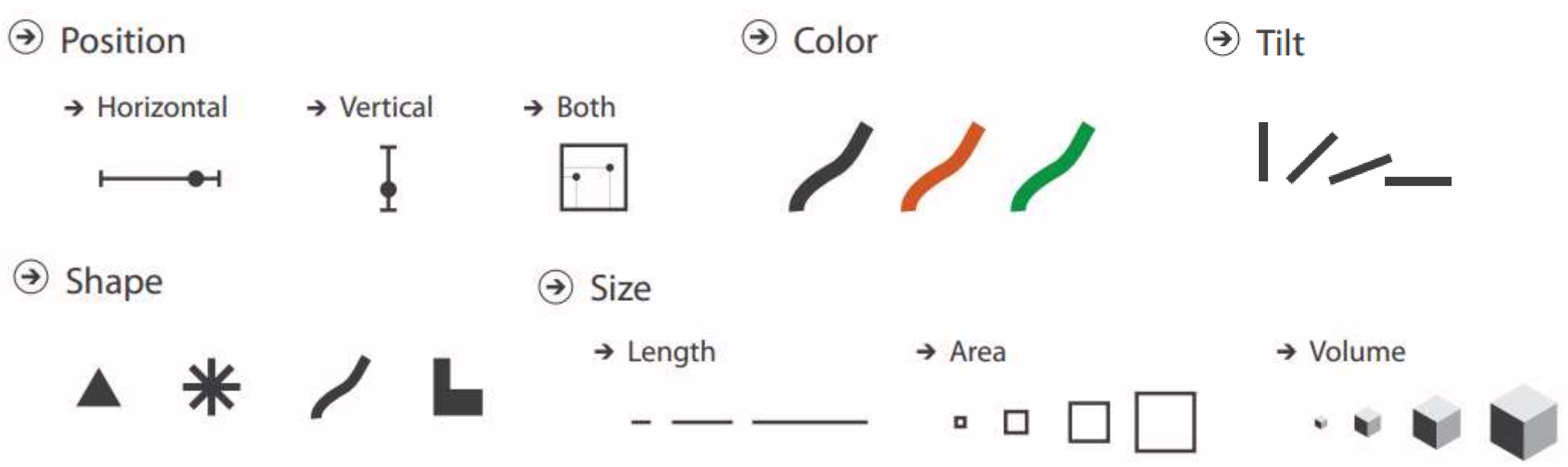
Section 6.2:
Marks and Channels

What are Marks and Channels?

- **Marks** are geometric primitives



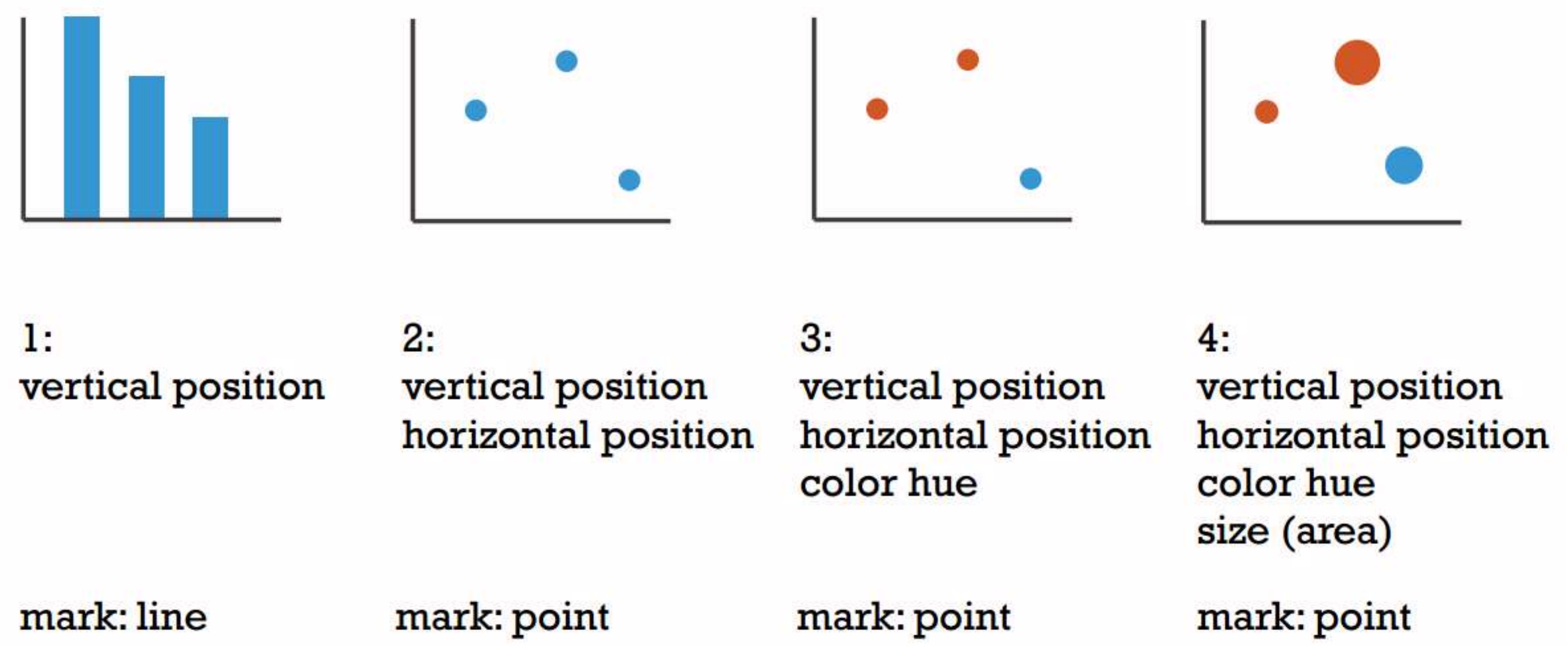
- **Channels** control visual appearance of marks



Images from Tamara Munzner, Visualization Analysis and Design, 2014

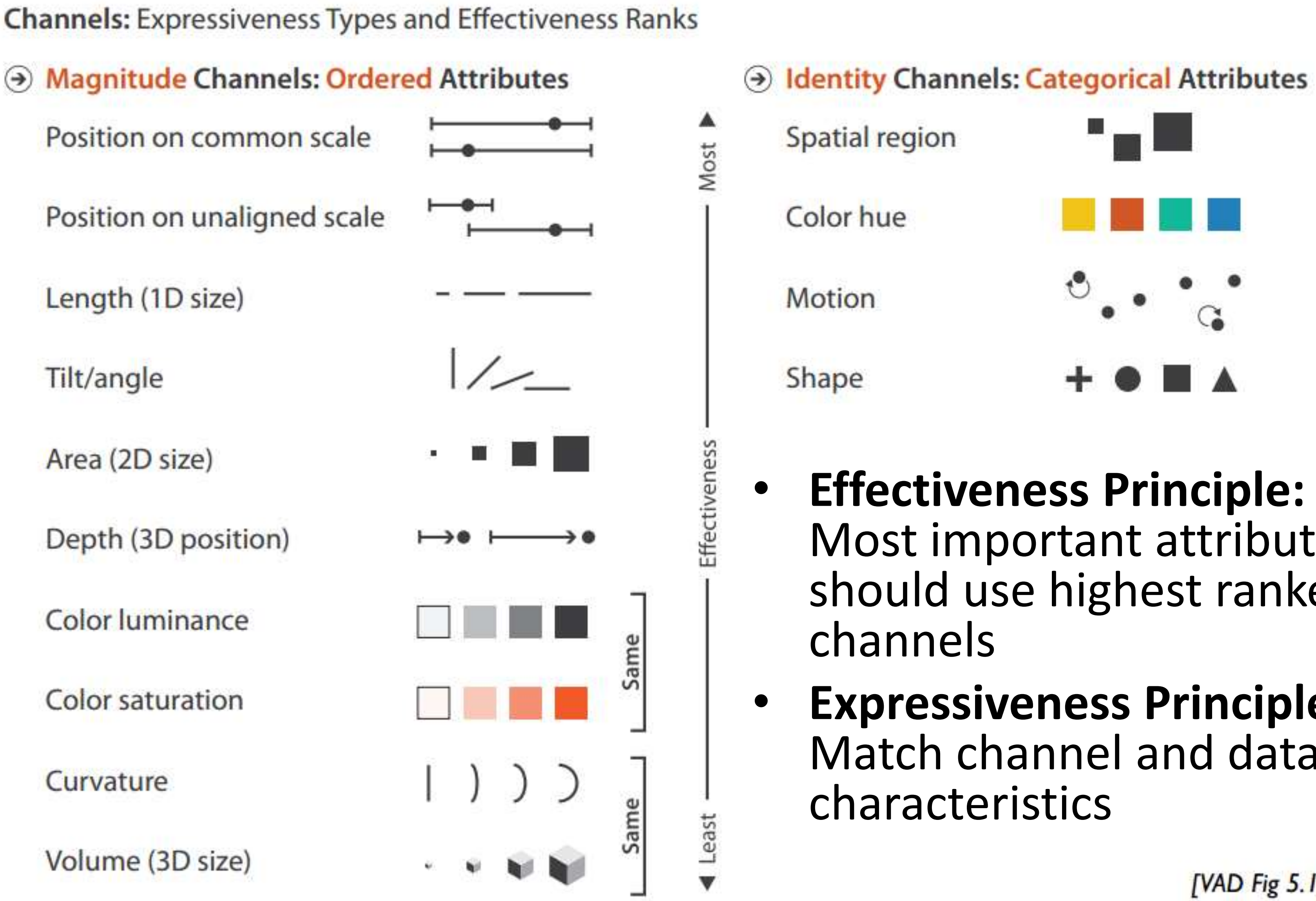
Combining Marks and Channels

- You can analyze common visualization idioms as combinations of marks and channels:



Images from Tamara Munzner, Visualization Analysis and Design, 2014

Effectiveness of Channels



Channels: Expressiveness Types and Effectiveness Ranks

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

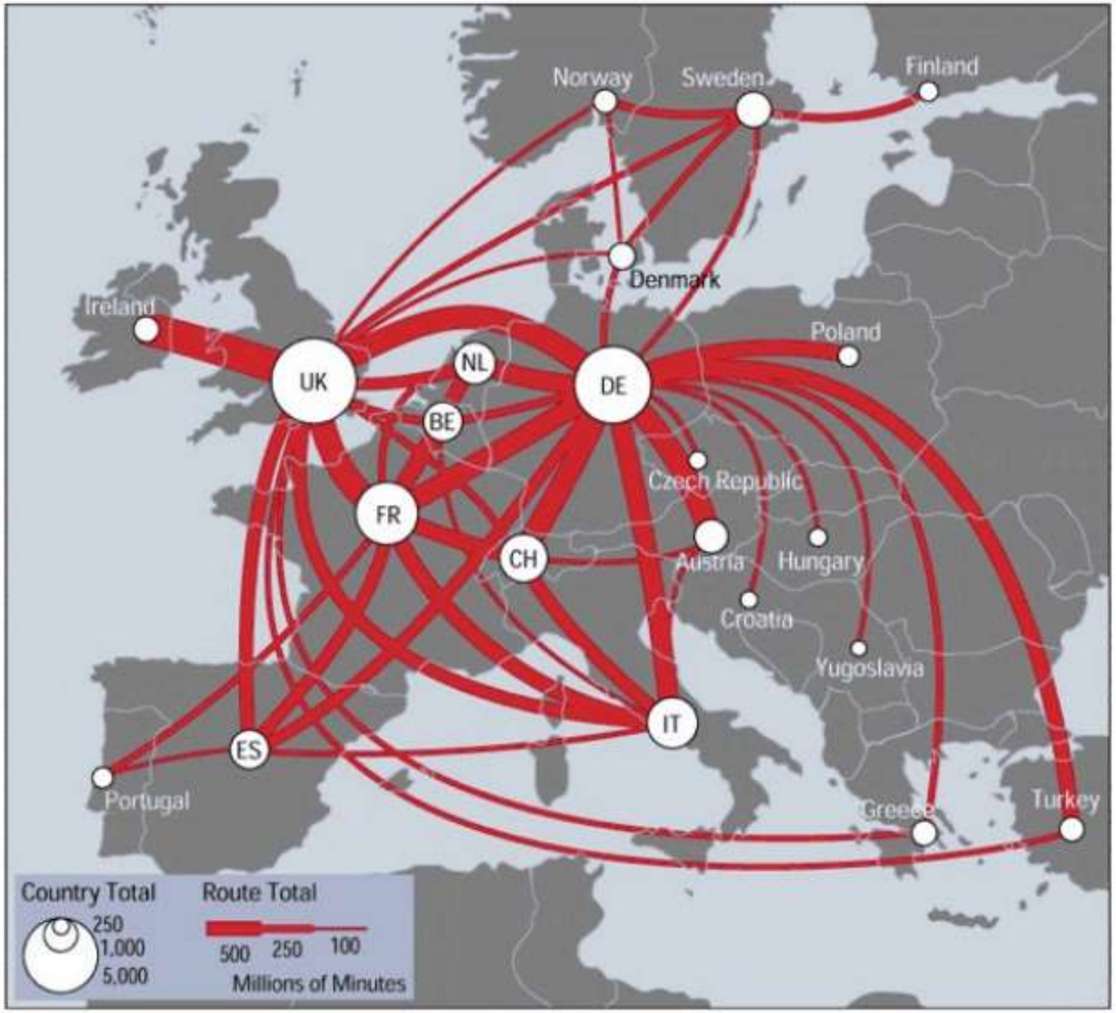
- **Effectiveness Principle:** Most important attributes should use highest ranked channels
- **Expressiveness Principle:** Match channel and data characteristics

[VAD Fig 5.1]

Images from Tamara Munzner, Visualization Analysis and Design, 2014

Expressiveness: How Many Levels?

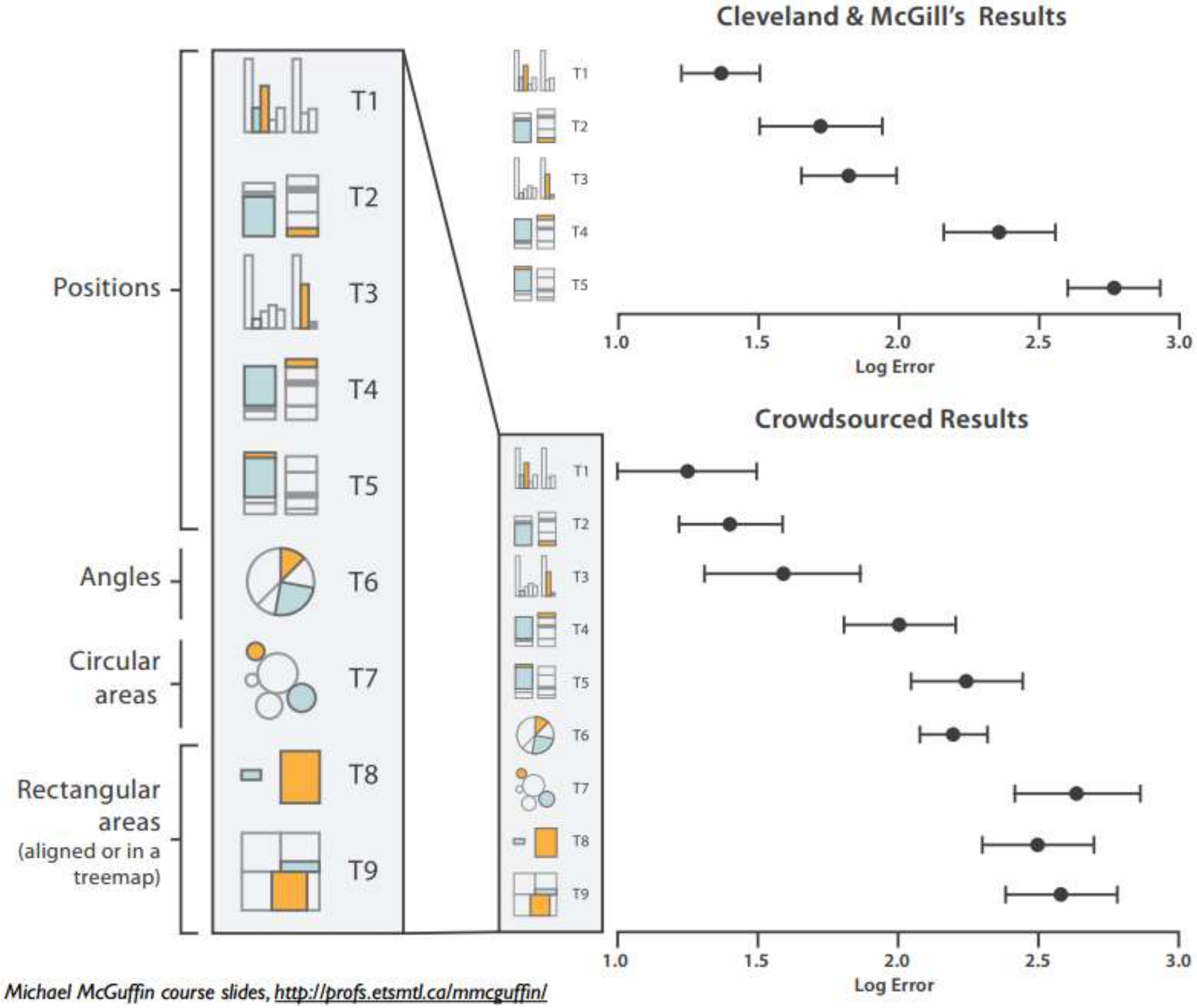
- Some channels only allow us to reliably distinguish between few levels
- Verify that this matches your number of attributes
- Bin if needed



[mappa.mundi.net/maps/maps 014/telegeography.html]

Effectiveness: How do we know?

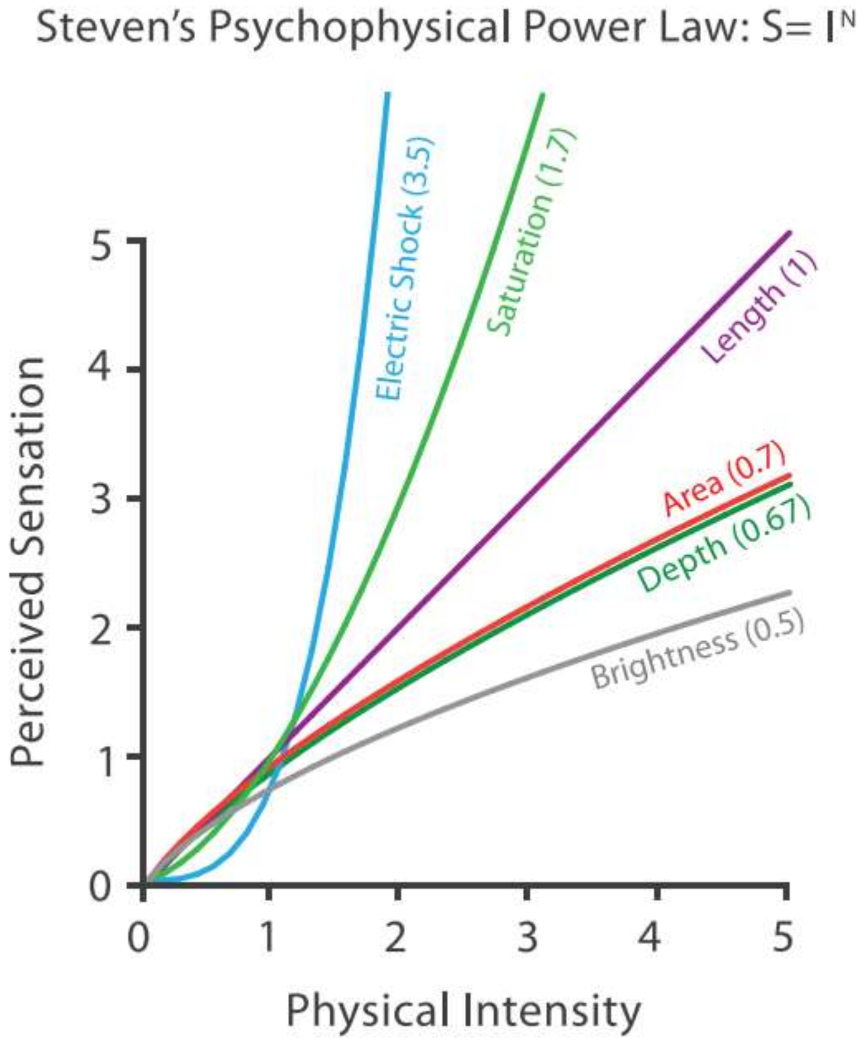
- Compare accuracy (or speed) on clearly defined tasks with different encodings



Michael McGuffin course slides, http://profs.etsmtl.ca/mmcguffin/

Effectiveness: Why?

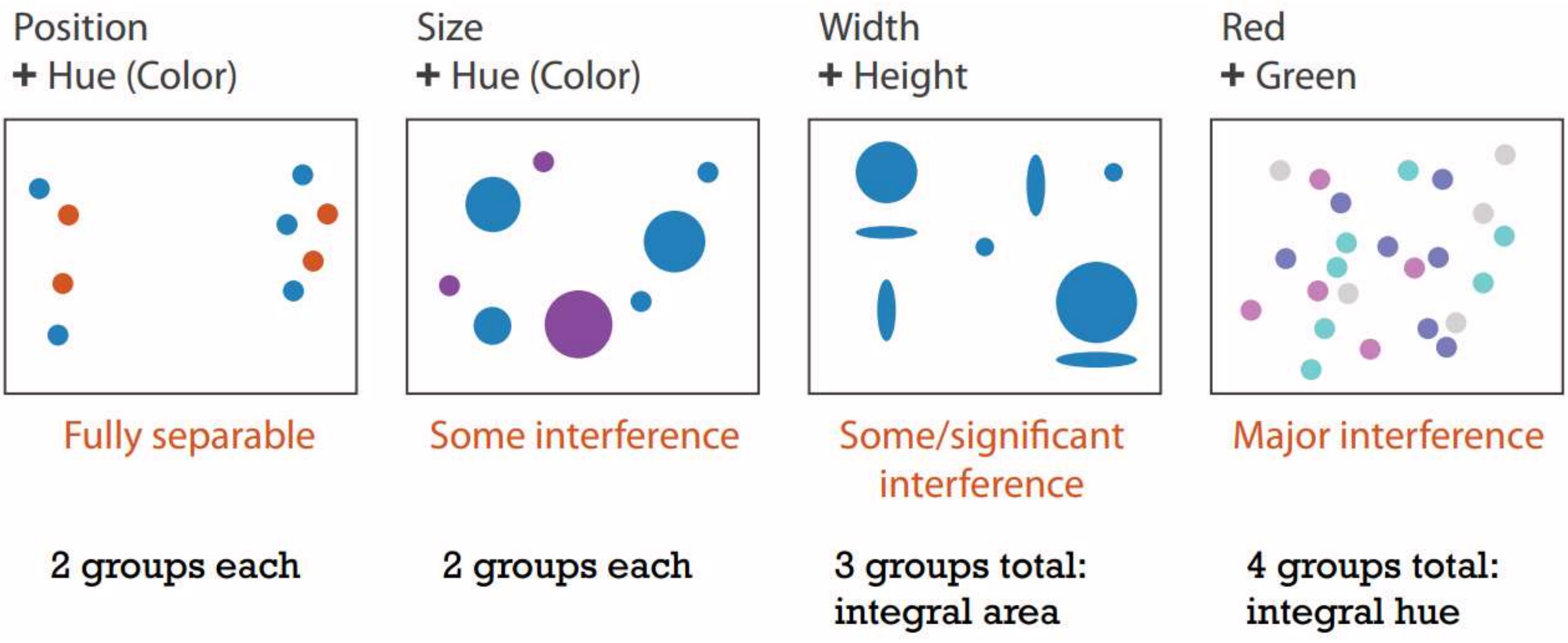
- Differences in accuracy explained by Steven‘s Psychophysical Power



Images from Tamara Munzner, Visualization Analysis and Design, 2014

Combinations of Channels

- Keep in mind separability vs. integrability:



Images from Tamara Munzner, Visualization Analysis and Design, 2014

Summary: Marks and Channels

- **Marks** are the geometrical primitives used for visualization
- **Channels** control their appearance
- Channels should be matched to data in order to be **expressive**
- Some channels are more **effective** than others – use those for the most important attributes!

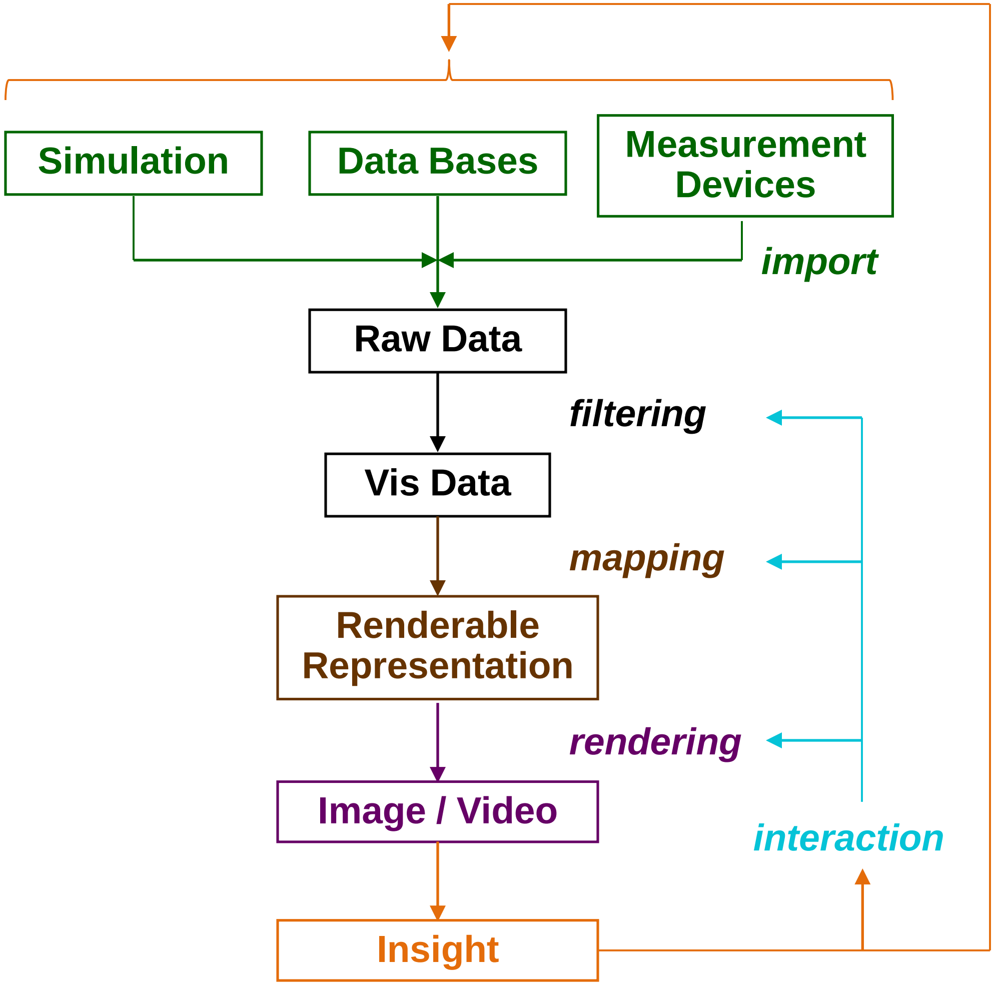
Section 6.3:
Visualization Pipelines

Visualization Pipelines: Motivation

- **Visualization Pipelines**
  - provide a scheme of how **data** is read in, filtered, mapped, and rendered during visualization
  - are a useful **abstraction** and common reference for many individually tailored solutions commonly found in visualization
  - guide organization of **class libraries**

Visualization Pipeline

- Most visualizations follow a common scheme:

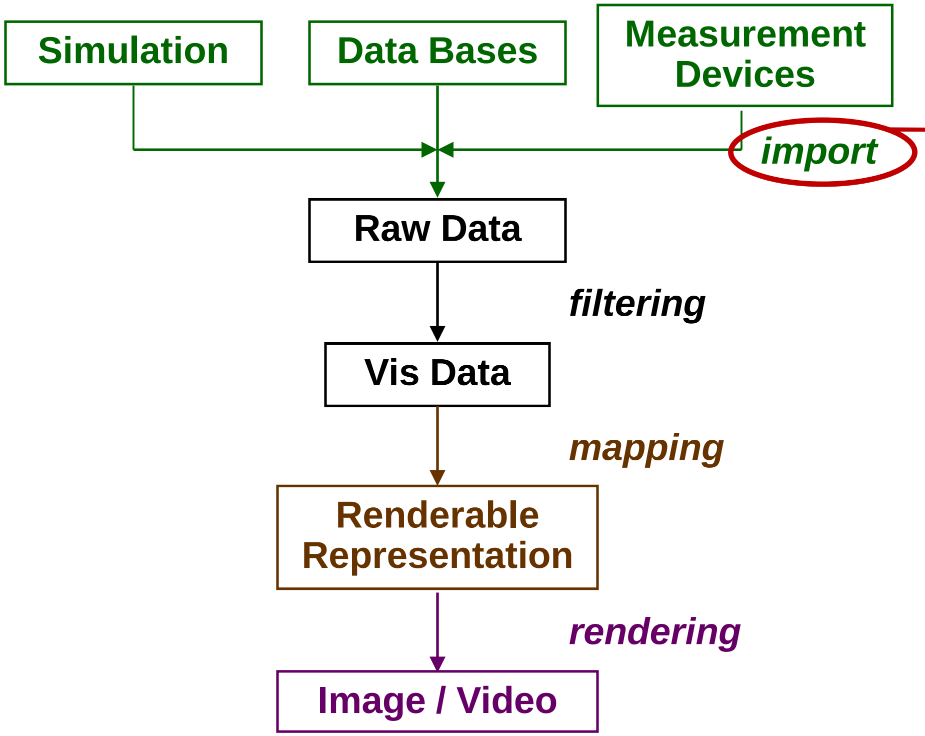


Examples of interaction:

- Filter out parts of the data
- Change level of detail
- Reorder PCP axes
- Switch between visualization techniques

A **crucial benefit** of computer-based visualization

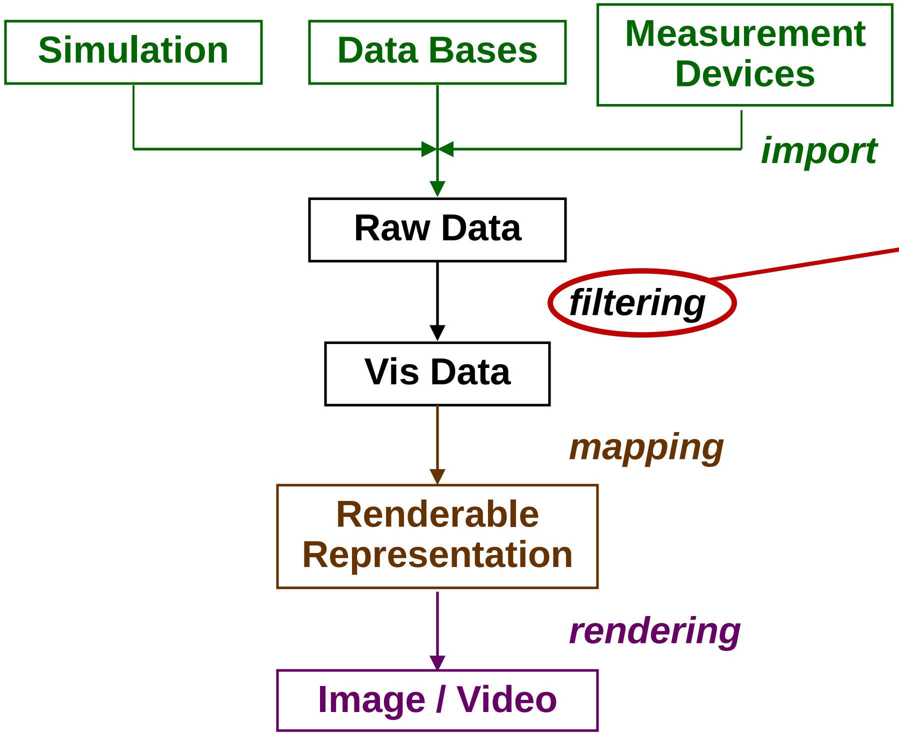
Import



Import:
data → data

- data format conversion
- resampling
- interpolation
- imputation

Filtering



Simulation

Data Bases

Measurement Devices

import

Raw Data

filtering

Vis Data

mapping

Renderable Representation

rendering

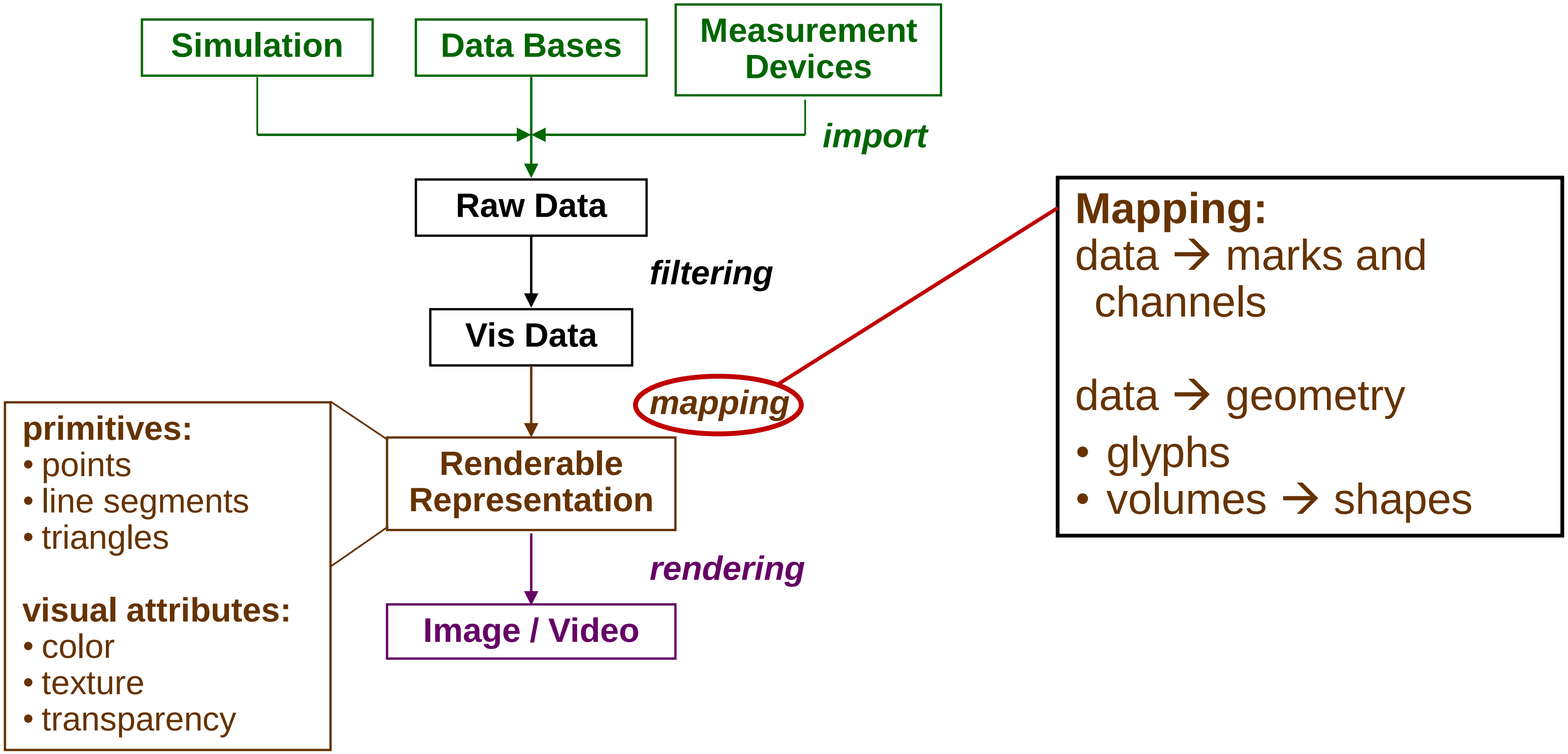
Image / Video

Filtering:

data → data

- aggregation
- denoising
- outlier removal

Mapping



Simulation

Data Bases

Measurement Devices

import

Raw Data

filtering

Vis Data

mapping

Renderable Representation

rendering

Image / Video

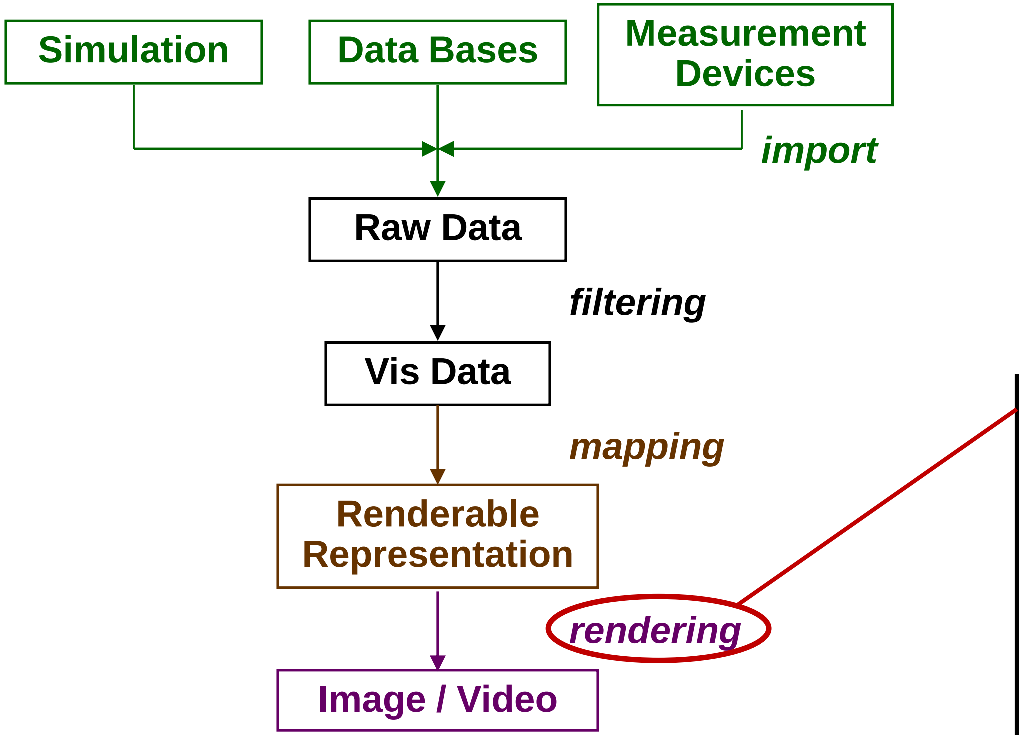
primitives:
- points
- line segments
- triangles

visual attributes:
- color
- texture
- transparency

Mapping:
data → marks and channels

data → geometry
- glyphs
- volumes → shapes

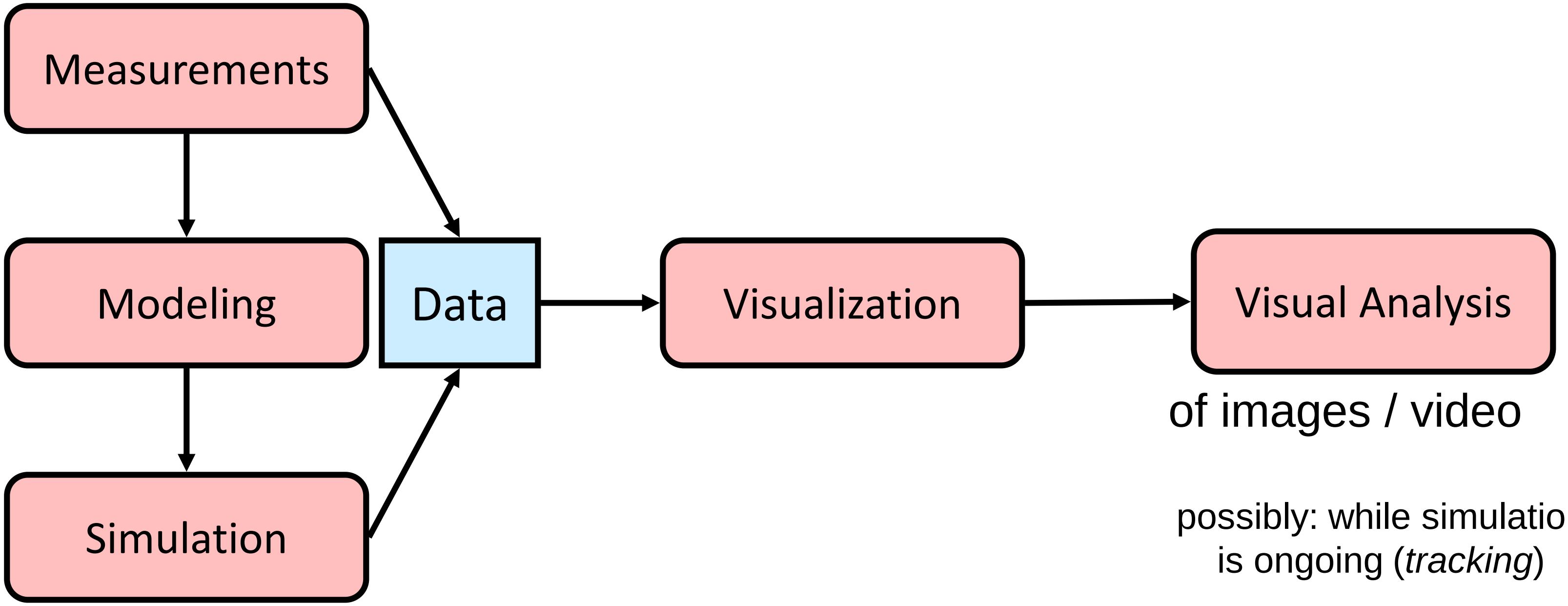
Rendering



Rendering:

- Rasterization
- 3D Volume Rendering
- “realistic“ rendering, e.g., shadows, lighting, shading
- “illustrative” rendering

Non-Interactive Visualization



Measurements

Modeling

Simulation

Data

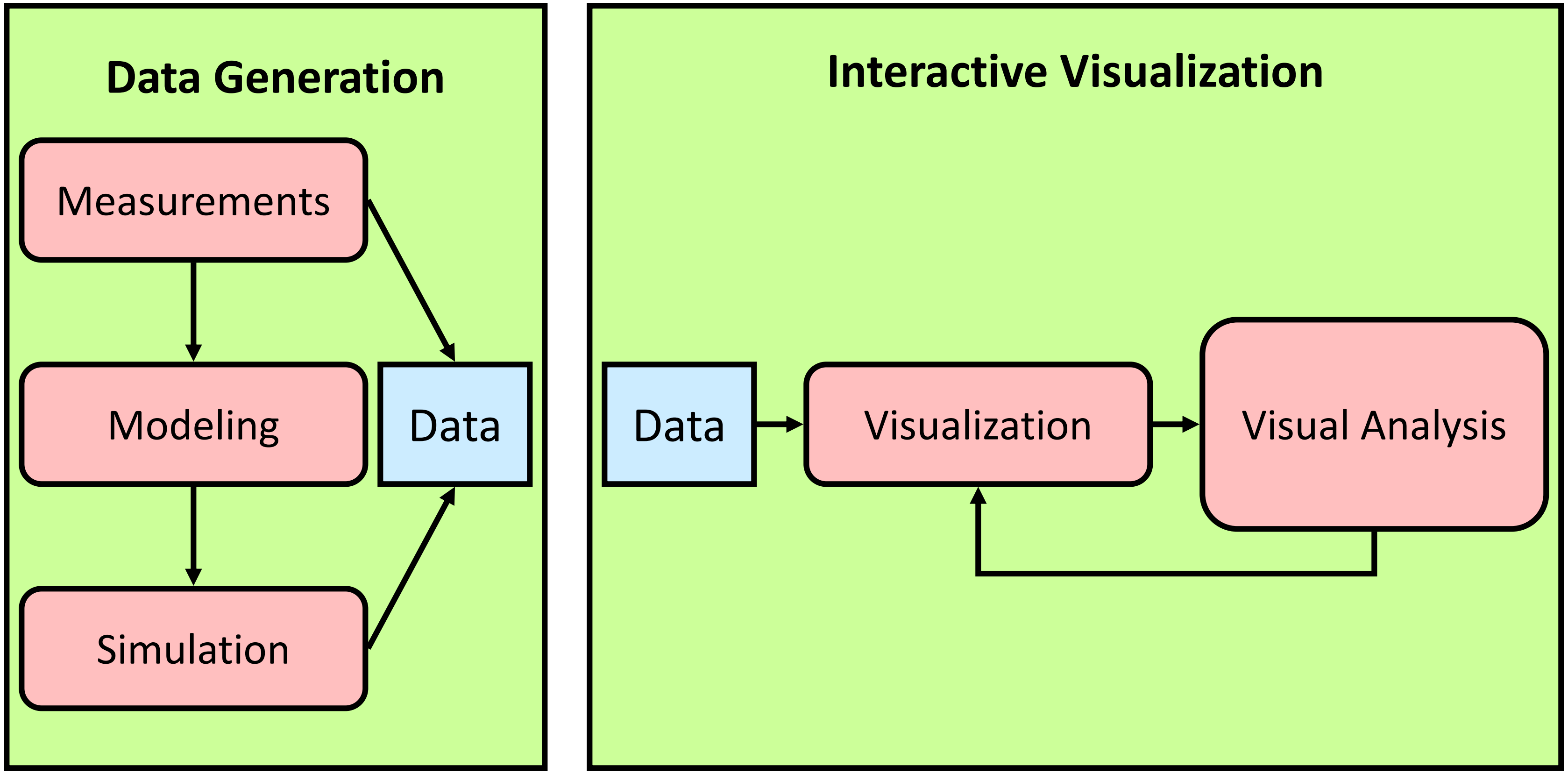
Visualization

Visual Analysis

of images / video

possibly: while simulation
is ongoing (*tracking*)

Interactive Visualization as Post-Process



Data Generation

Measurements

Modeling

Simulation

Data

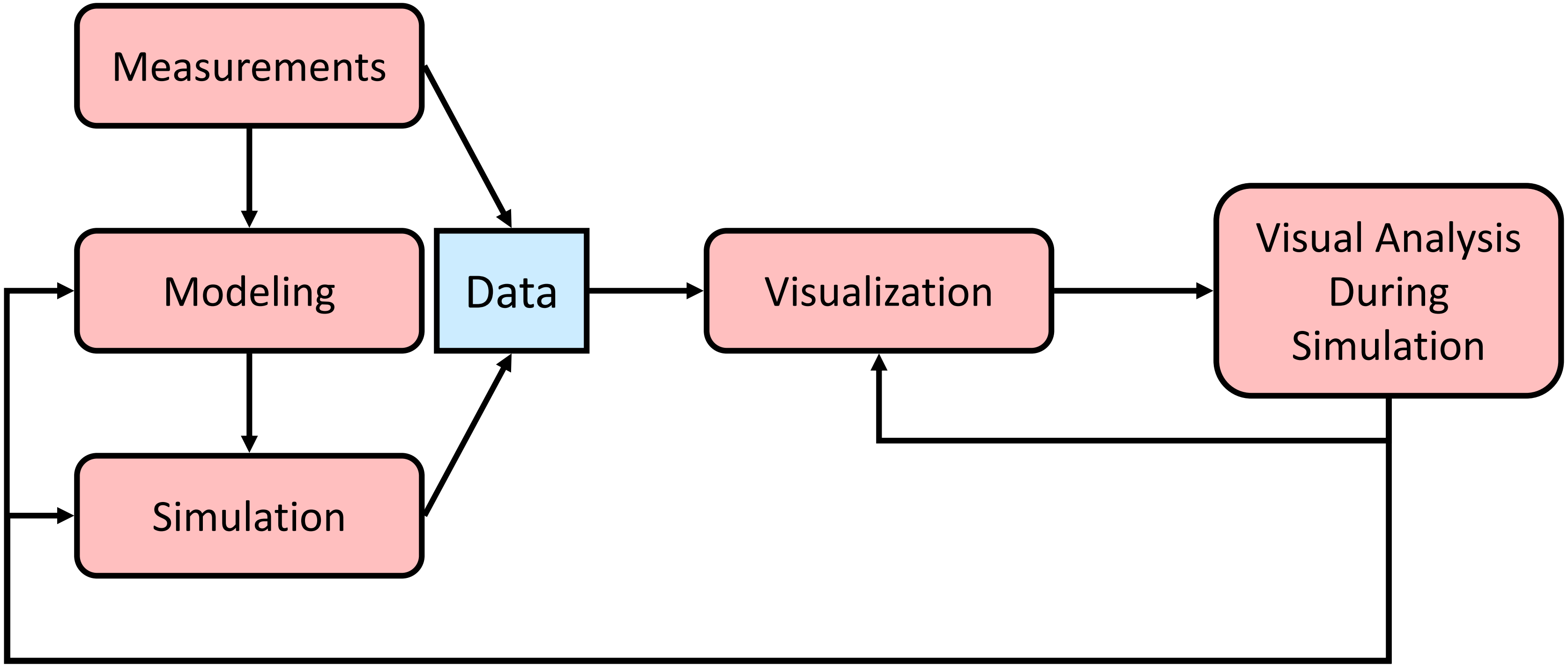
Interactive Visualization

Data

Visualization

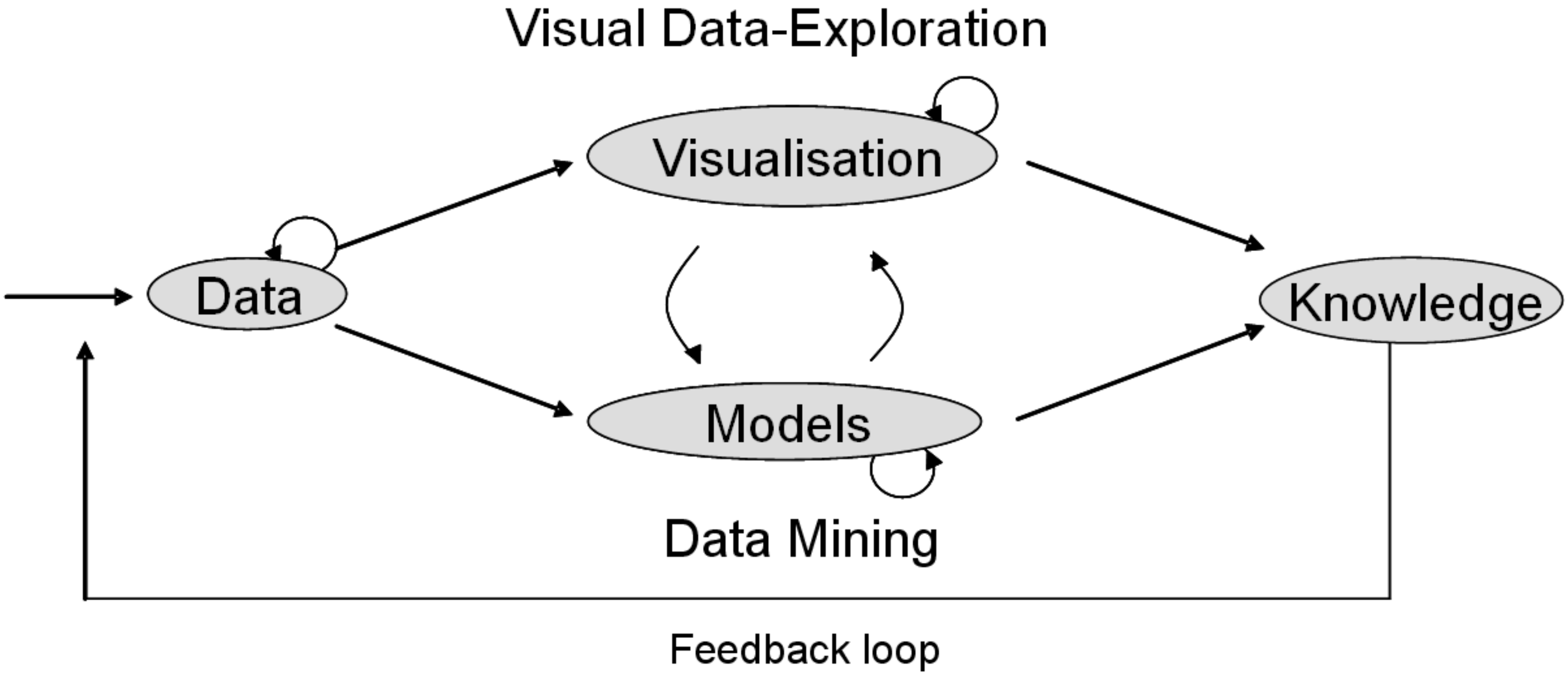
Visual Analysis

Computational Steering



Visual Analytics

- Source: Keim et al. 2008



Section 6.4:
Example System 1: Pathline

Pathline

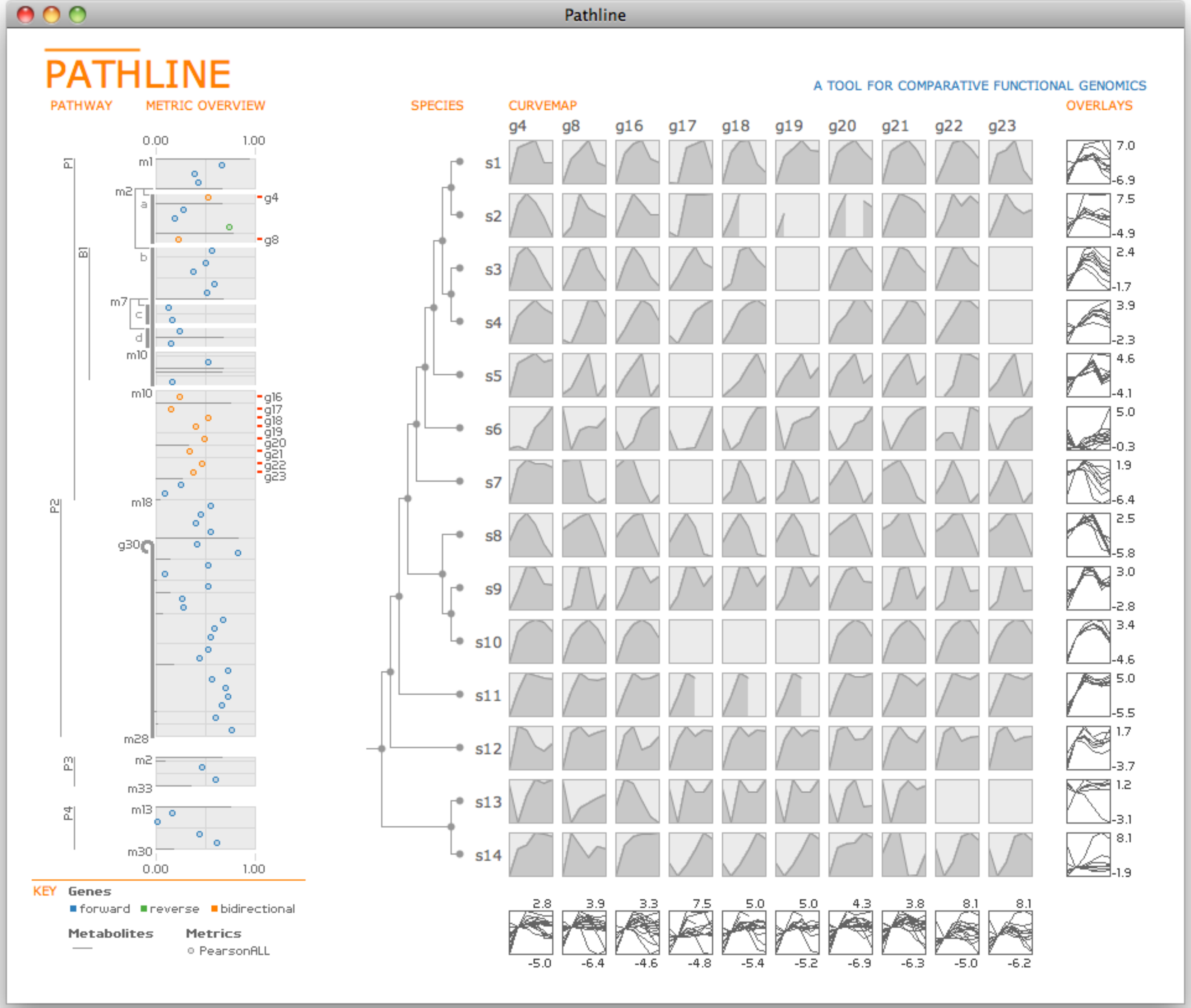
- Design study by [Meyer et al. 2010]
- **Comparative functional genomics**
  - How do gene regulation and metabolite levels in pathways evolve over time and across species?
- **Specific data:** 14 species of yeast
  - **Activity** of 6000 genes and **levels** of 140 metabolites at six points in time
  - **Metabolic pathways:** Directed graph (nodes: metabolites, edges annotated with genes)
  - **Similarity scores:** Similarities between time series measured by Pearson / Spearman correlation
  - **Phylogenetic relationship:** Tree (leaves: species) that shows differentiation of the species

Discover: Task Abstraction

- **High-level task:** Find commonalities and differences between species in the activation patterns of genes and metabolite levels
  - At which point in evolution were specific cellular processes and regulatory mechanisms introduced?
- **More specific tasks:**
  - Look for trends in a set of time series for a gene / metabolite across species
  - Look for trends across genes / metabolites in a set of time series within a species
  - Compare time series to find specific features (e.g., timing of peaks, clusters of time series etc.)

Design: Overview of System

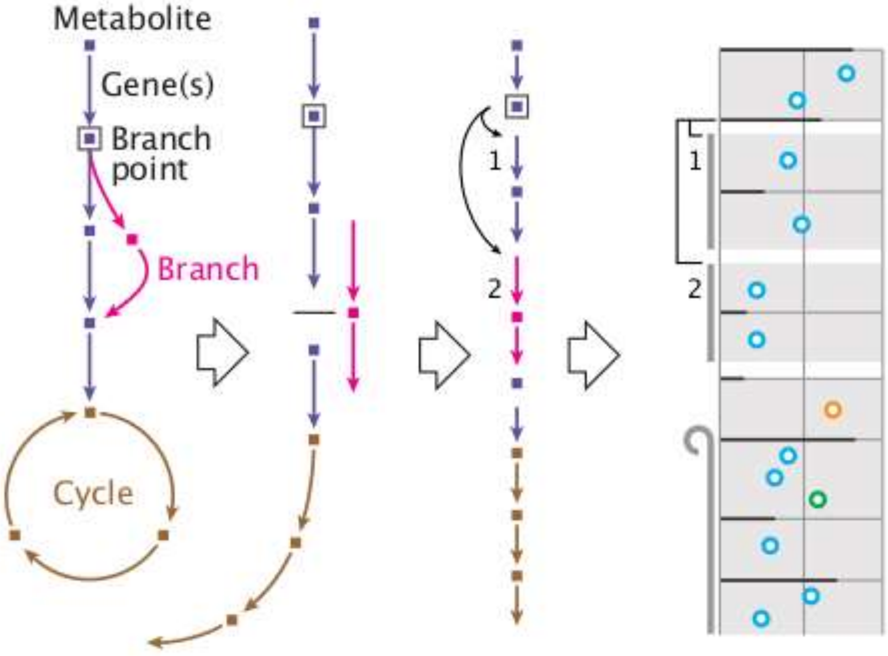
- **Two main views:** Linearized Pathways vs. Curvemap
- Can select genes for analysis in curvemap



Images from [Meyer et al. 2010]

Design: Linearized Pathways

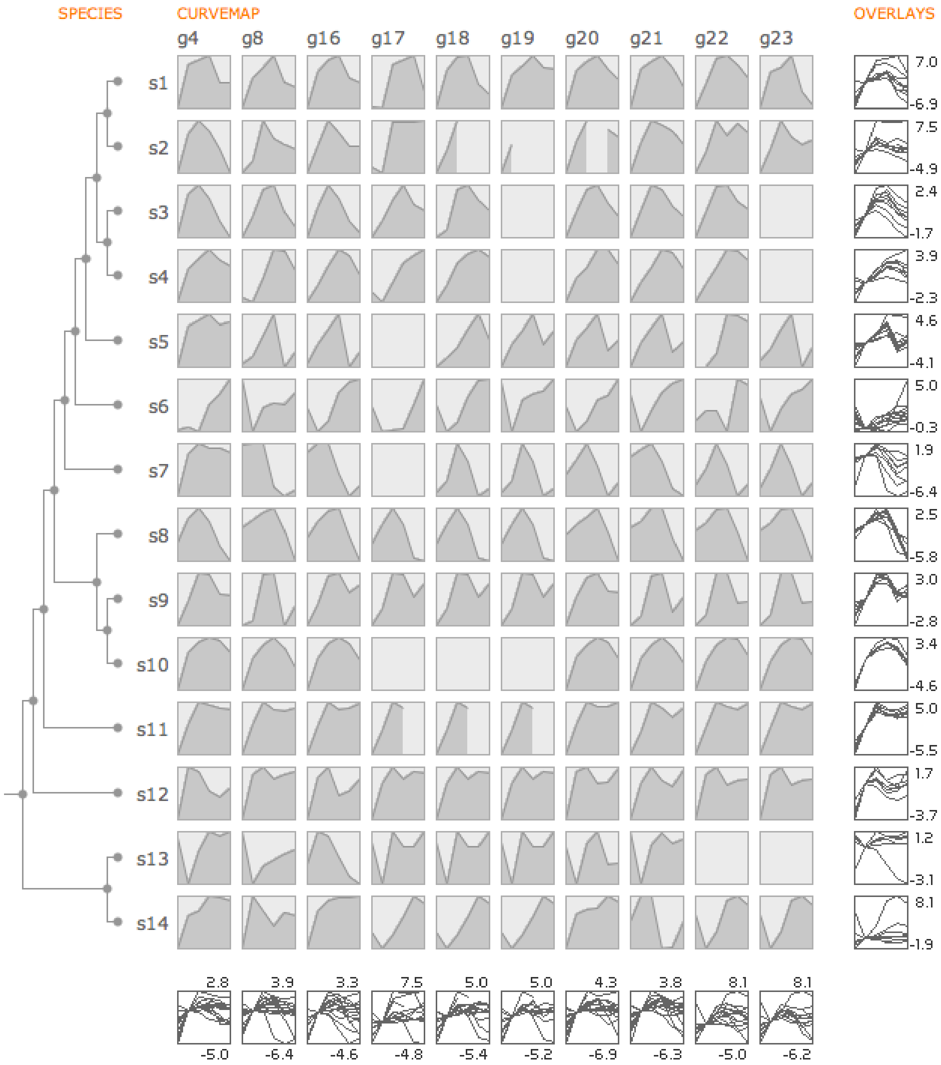
- **Goal:** Emphasize quantitative values along pathways, pathway topology is secondary
  - Unroll loops, disconnect and reinsert branches
  - Gaps indicate branches
  - Only include relevant genes / metabolites
  - Query user in case of ambiguities
  - Lines encode similarity scores of metabolites
  - Circles encode genes
  - Color indicates direction
  - Different pathways shown below each other



Images from [Meyer et al. 2010]

Design: Curvemap View

- **Matrix view,** curves in cells show protein expression / metabolite levels over time
  - Better than heat maps
- **Min/max normalization** per cell to make shape obvious
- **Aggregate views** for each row / column for more detailed comparison (incl. absolute levels)
- **Phylogenetic tree** shows how species relate



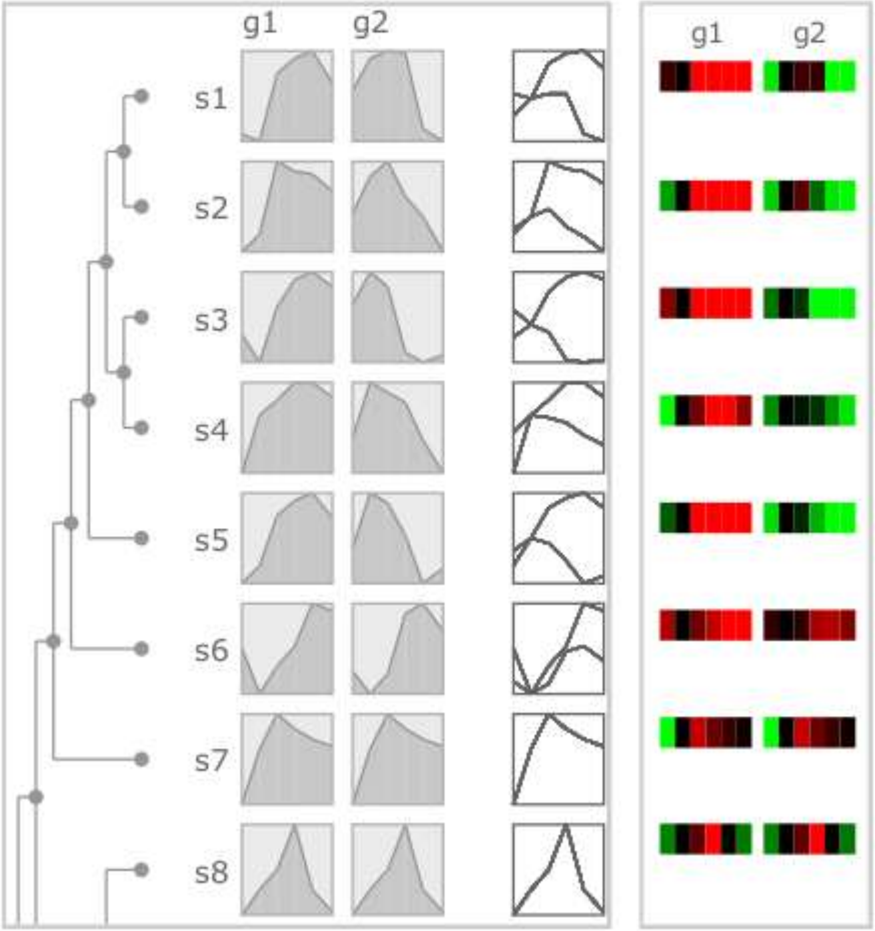
Images from [Meyer et al. 2010]

Implement

- **Interaction:**
  - Mousing over genes / metabolites in pathview shows names and numerical values
  - Clicking on them adds them to curvemap
  - Mousing over species / gene / metabolite highlights its curve in overlay plots

Deploy: Case Study 1

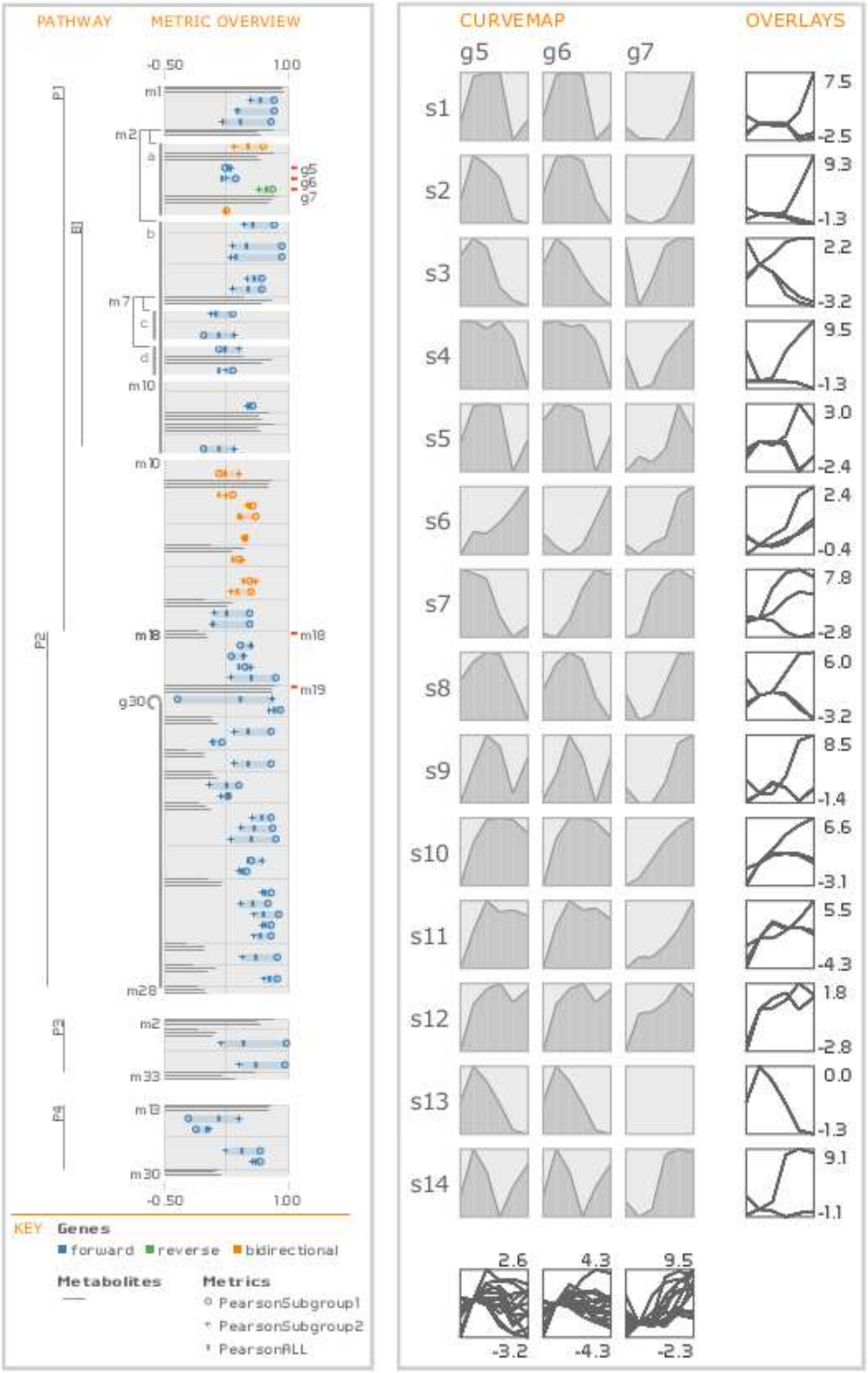
- Investigate **whole-genome duplication**
  - g1/g2 are duplicates in s1-s7
  - Curvemap clearly shows how different regulation mechanisms have evolved
  - Heat maps make it much more difficult to see this



Images from [Meyer et al. 2010]

Deploy: Case Study 2

- **Pathway analysis** (left)
  - Clear trend of reduced similarity in metabolites along the pathway
  - Trend suddenly reversed between m18 and m19
- **Gene-level analysis**
  - g5 and g6 strongly coupled, but time course varies across species
  - Discovered previously unknown gene duplication event in s7

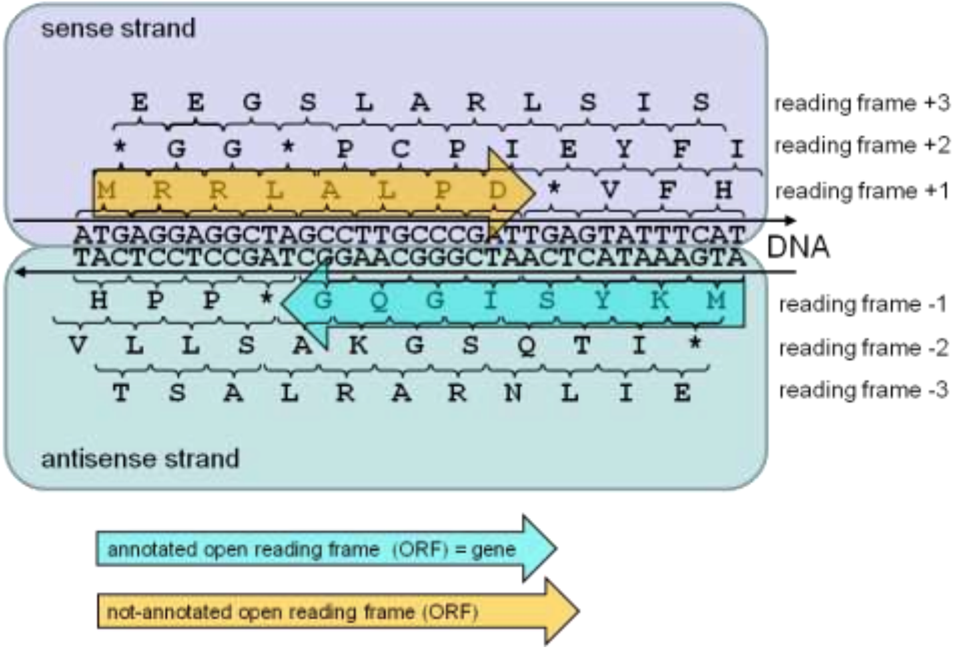


Images from [Meyer et al. 2010]

Section 6.5:
Example System 2:
Overlapping Gene Detection

Task: Discover Overlapping Genes

- Framework by [Simon et al. 2011]
- DNA double strands give rise to six reading frames:

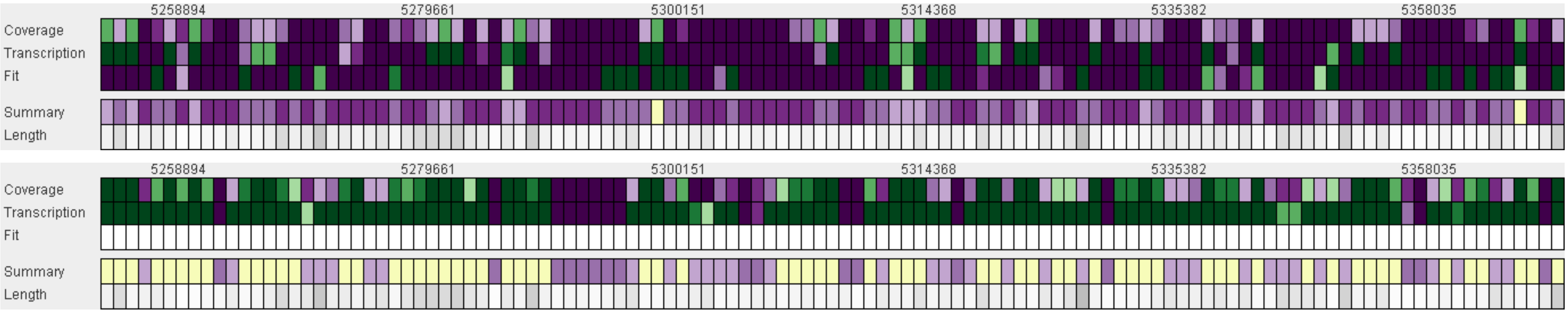


Images from [Simon et al. 2011]

- **Task:** Based on DNA & RNA sequencing, discover locations where genes overlap (e.g., sense and antisense strand encode different proteins)

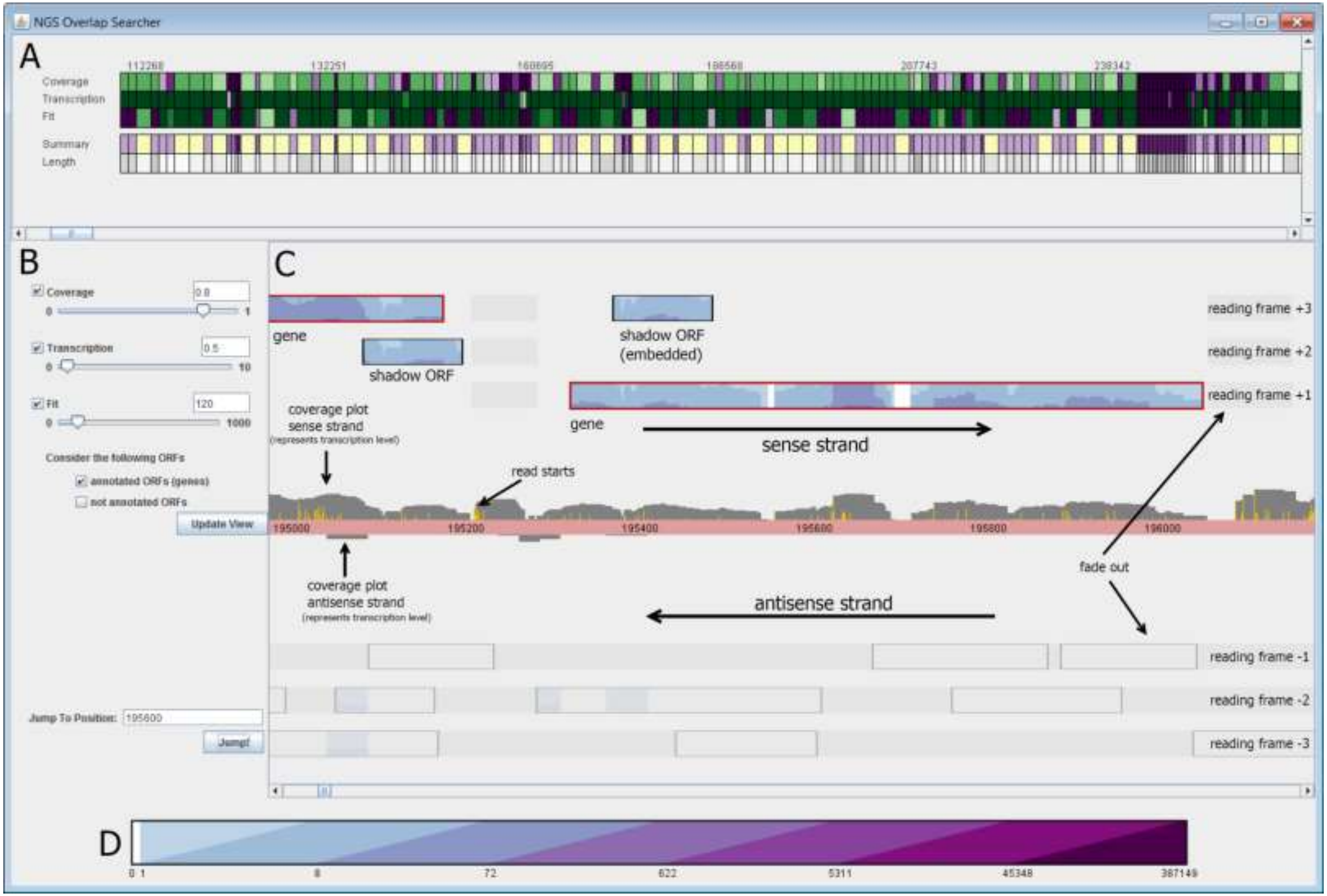
User-Defined Filters

- **Challenge:** Huge amount of data (2-200 Gbp) and open reading frames (ORFs), noisy data
  - – Fully automated analysis impossible
  - – Fully manual analysis impossible
- **Filter** ORFs based on three criteria:
  1. **Coverage:** Percentage of bases that has been counted at least once
  2. **Transcription:** Average counts per basis
  3. **Fit:** Difference between ORF and transcript lengths



Images from [Simon et al. 2011]

Visual Analytics Framework



NGS Overlap Searcher

☑ Coverage 0.8

☑ Transcription 0.5

☑ Fit 120

Consider the following ORFs

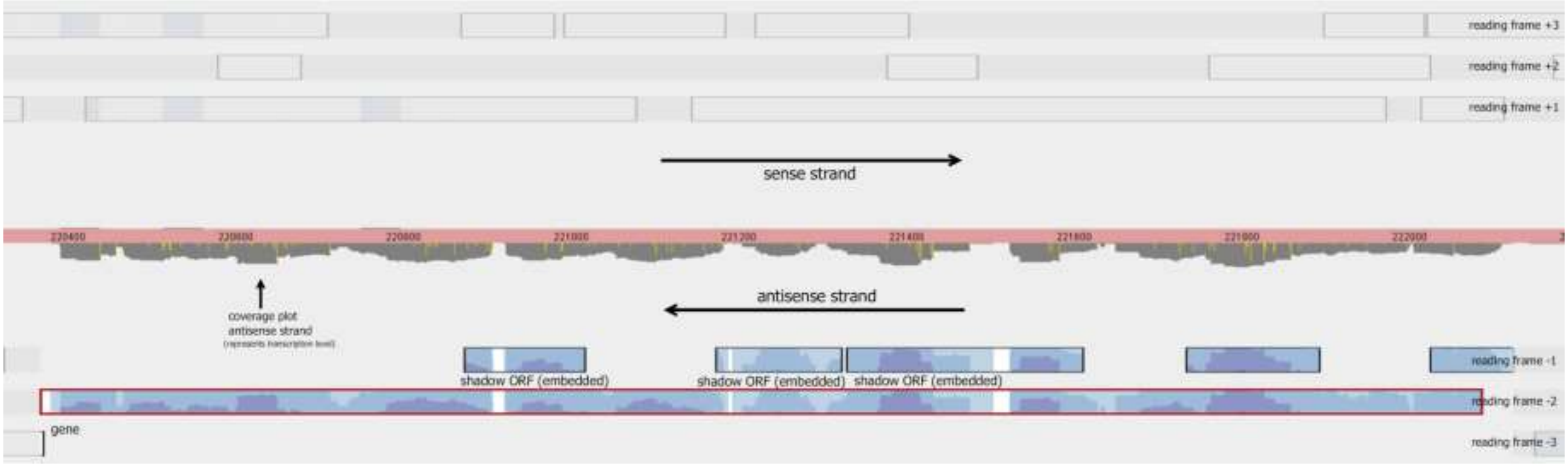
☑ annotated ORFs (genes)

□ not annotated ORFs

Jump To Position: 195600

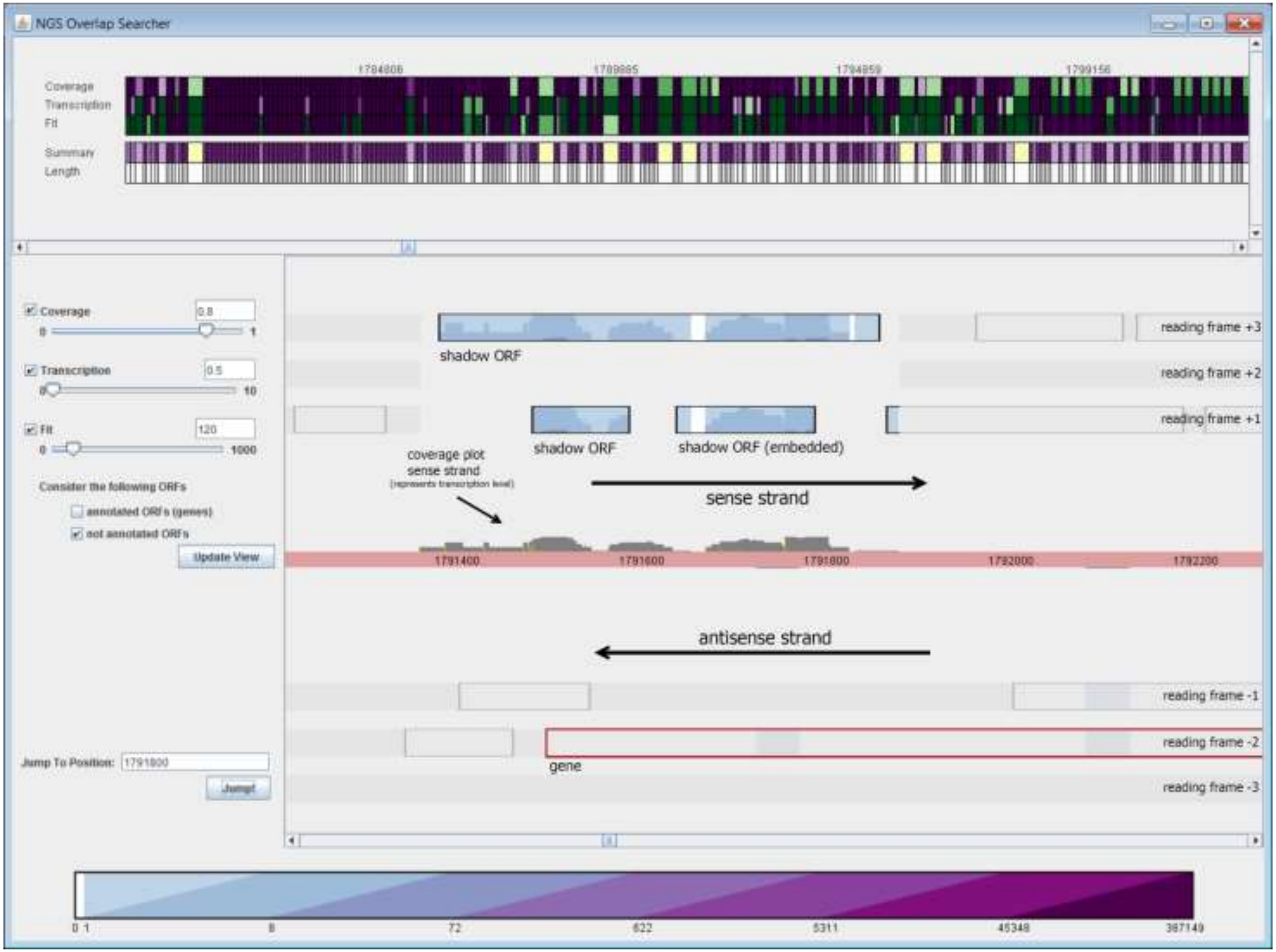
Images from [Simon et al. 2011]

Example Result 1: Known Gene



Images from [Simon et al. 2011]

Example Result 2: Pot. Overlapping Gene



NGS Overlap Searcher

☑ Coverage 0.8

☑ Transcription 0.5

☑ Fit 120

Consider the following ORFs

□ annotated ORFs (genes)

☑ not annotated ORFs

Jump To Position: 1791800

Images from [Simon et al. 2011]

Summary of Chapter 6

- Topics of **Chapter 6** were:
  - Visualization Design Process
  - Classification of Marks and Channels
  - Visualization Pipelines
  - Example Systems: Pathline / OLG Detection
- **Concludes first part of lecture:** How to use visualization for data analysis

References

- Tamara Munzner, **Visualization Analysis and Design,** A K Peters, 2014
- Sedlmair et al., **Design Study Methodology: Reflections from the Trenches and the Stacks**, IEEE Trans. On Visualization and Computer Graphics 18(12):2431-2440, 2012
- Meyer et al., **Pathline: A Tool For Comparative Functional Genomics,** Computer Graphics Forum 29(3), 2010
- Simon et al., **Visual Analysis of Next-Generation Sequencing Data to Detect Overlapping Genes in Bacterial Genomes,** BioVis 2011