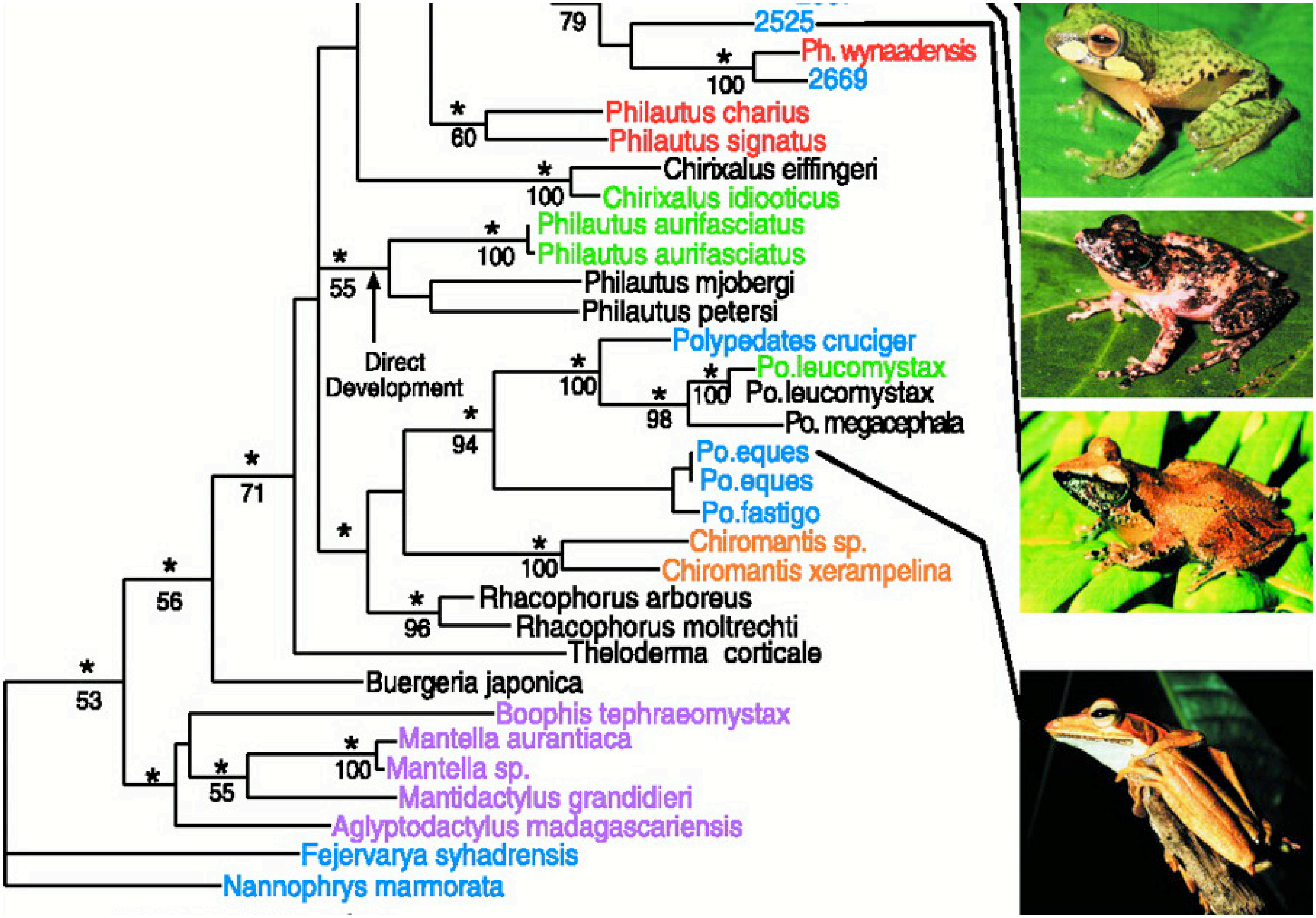
Scalable Visualization with Accordion Drawing

Tamara Munzner
University of British Columbia
Department of Computer Science

joint work with James Slack, Kristian Hildebrand, Katherine St. John

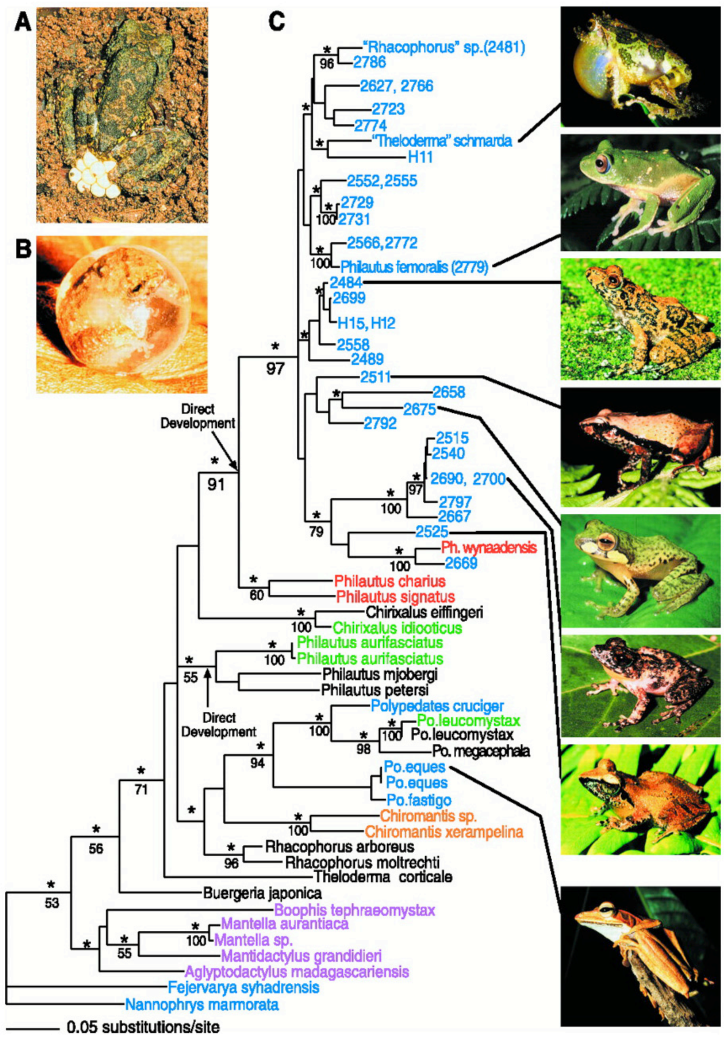
Imager

Problem: Comparing Evolutionary Trees



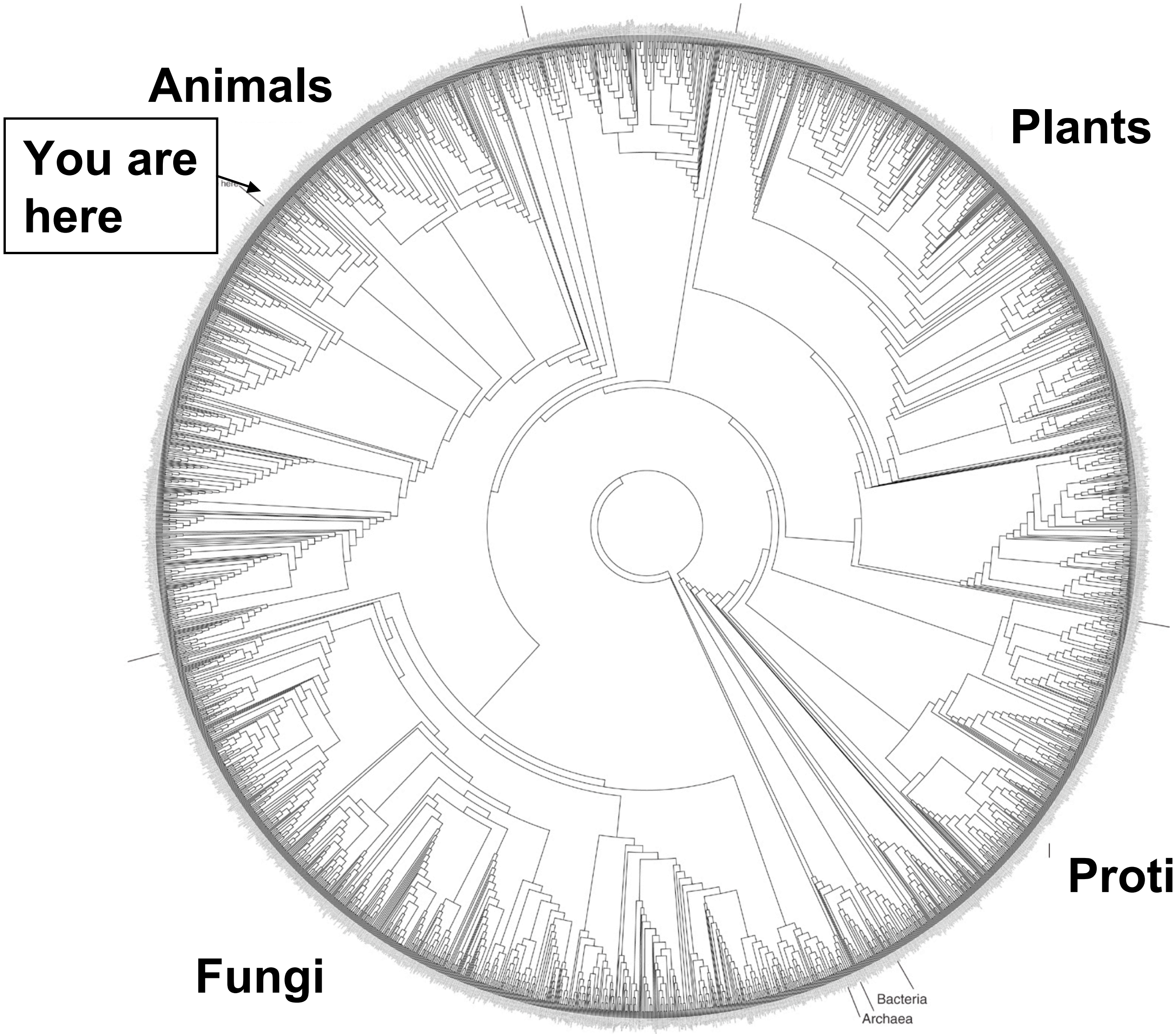
M Meegaskumbura et al., Science 298:379 (2002)

Common Dataset Size Today



M Meegaskumbura et al., Science 298:379 (2002)

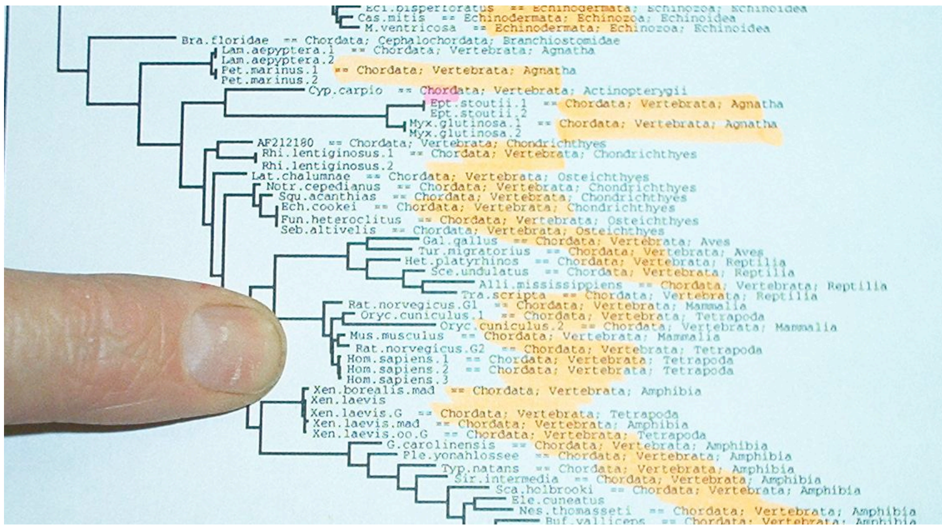
Future Goal: 10M Node Tree of Life



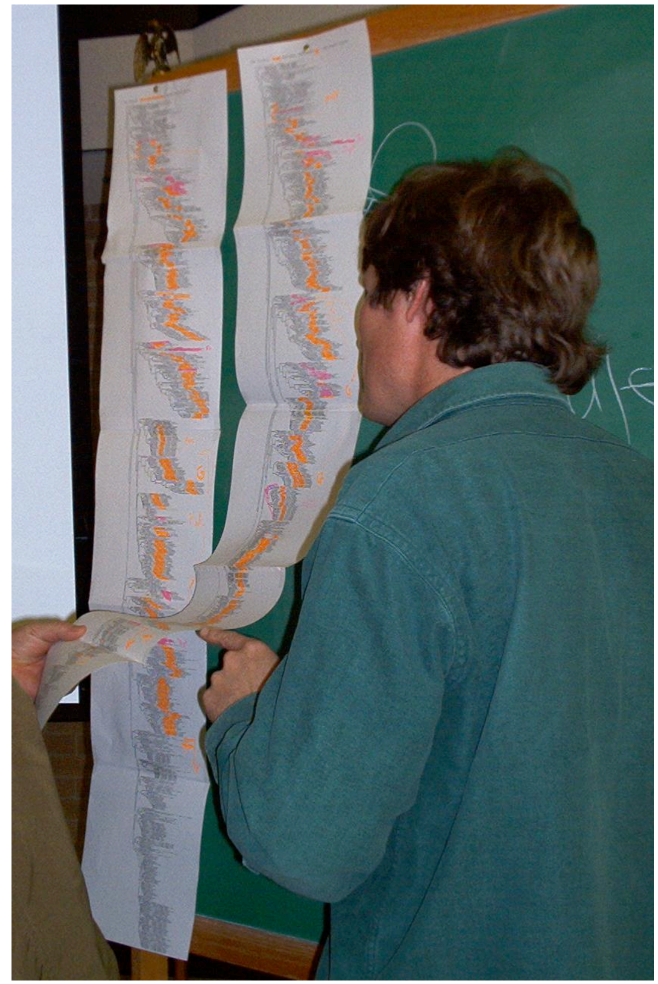
David Hillis, Science 300:1687 (2003)

Paper Comparison: Multiple Trees

focus

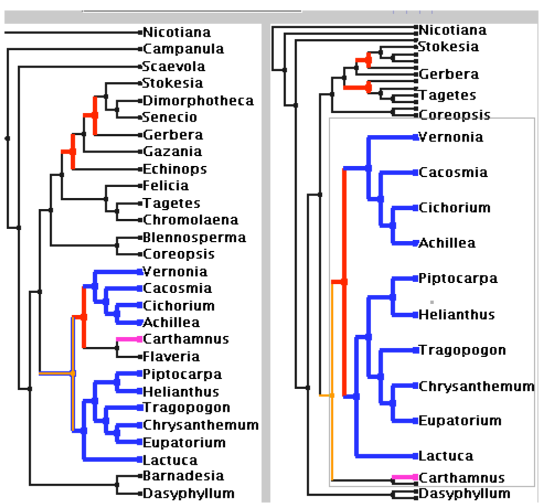


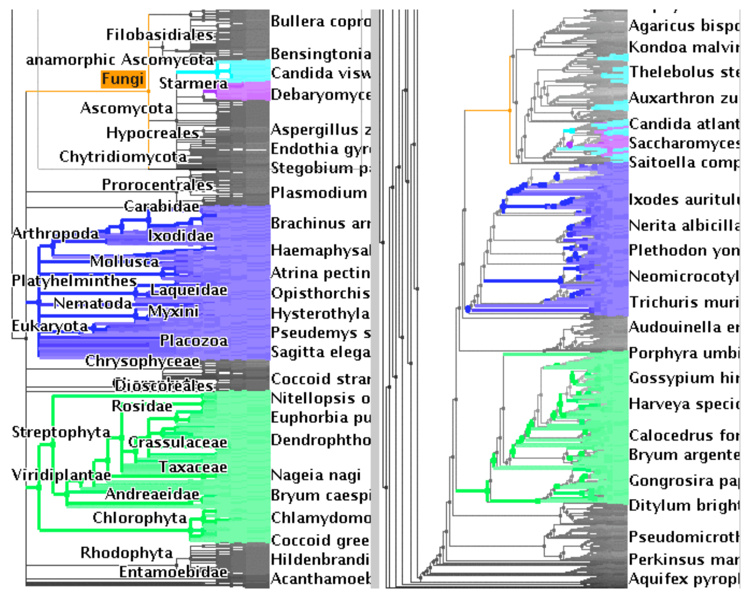
context



TreeJuxtaposer

- side by side comparison of evolutionary trees
  - video, software downloadable from http://olduvai.sf.net/tj





[TreeJuxtaposer: Scalable Tree Comparison using Focus+Context with Guaranteed Visibility. Tamara Munzner, François Guimbretière, Serdar Tasiran, Li Zhang, Yunhong Zhou. Proc SIGGRAPH 2003]

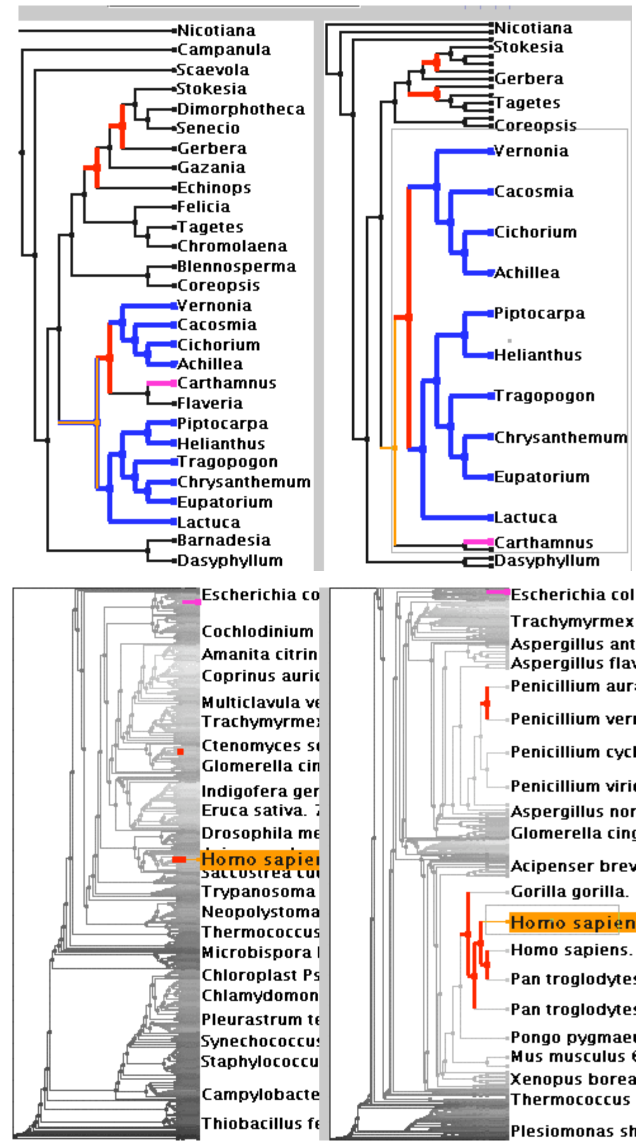
TJ Contributions

- first interactive tree comparison system
  - automatic structural difference computation

- scalable to large datasets
  - 250,000 to 500,000 total nodes: original
  - up to 4,000,000 nodes: later, with PRISAD
  - all preprocessing subquadratic
  - all realtime rendering sublinear
    - items to render >> number of available pixels
- scalable to large displays (4000 x 2000)
- introduced accordion drawing

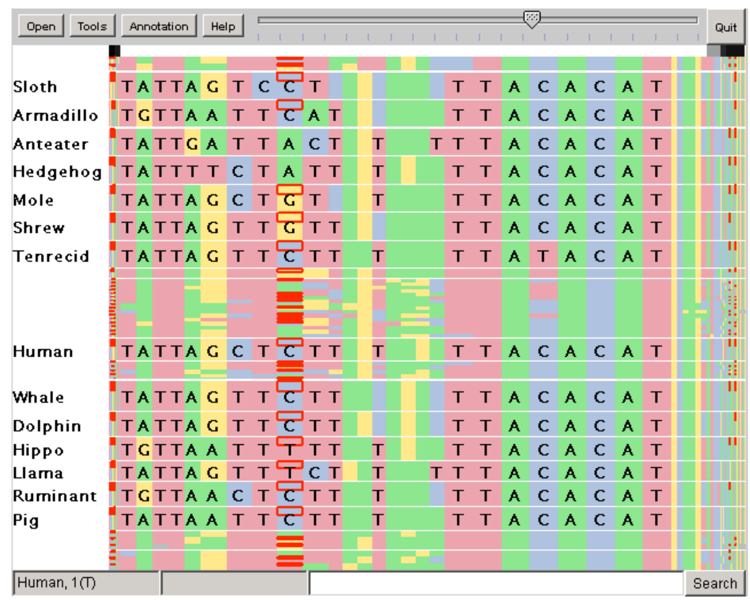
Accordion Drawing

- rubber-sheet navigation
  - stretch out part of surface, the rest squishes
  - borders nailed down
  - Focus+Context technique
    - integrated overview, details
  - old idea
    - [Sarkar et al 93], [Robertson et al 91]
- guaranteed visibility
  - marks always visible
  - important for scalability
  - new idea
    - [Munzner et al 03]



SequenceJuxtaposer

- side by side comparison of multiple aligned gene sequences
- would accordion drawing help?
  - multiple focus areas, smooth transitions, guaranteed visible landmarks
- now commonly browsed with web apps: zoom/pan with jumps, just one region
- video/ software downloadable from http://olduvai.sf.net/sj



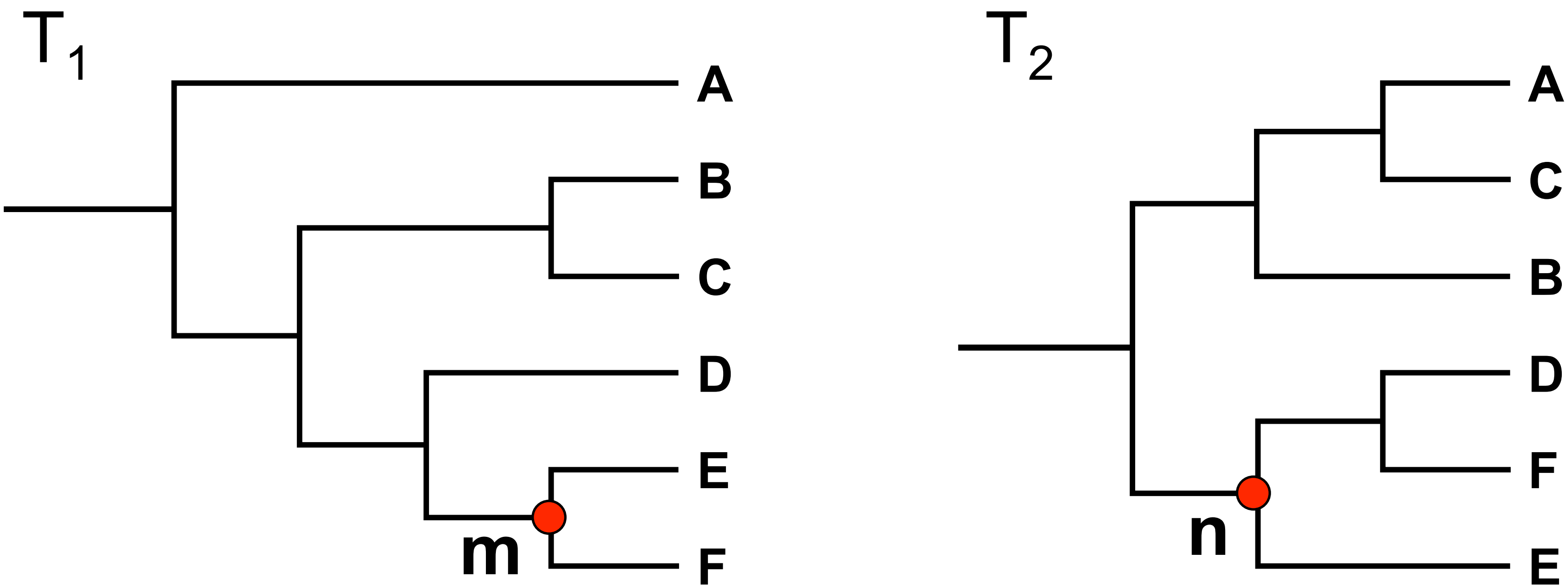
- scalability (later, with PRISAD)
  - 44 species * 17K nucleotides = 770K items
  - 6400 species * 6400 nucleotides = 40M items

[SequenceJuxtaposer: Fluid Navigation For Large-Scale Sequence Comparison In Context. James Slack, Kristian Hildebrand, Tamara Munzner, and Katherine St. John. Proc. German Conference on Bioinformatics 2004]

What's Hard?

- Tree Diff
  - Find best corresponding nodes between trees
  - Algorithm complexity - preprocessing: $O(n \log^2 n)$. Per-frame: constant
- Guaranteed Visibility
  - Landmarks don't vanish
- Rendering
  - For each frame, partition into visible regions, draw something useful
  - Provide guaranteed visibility of landmarks
  - Algorithm complexity depends on screen size, not dataset size
- Navigation
  - Have: (Objects drawn each frame) << (Total dataset objects)
  - Want: (Updates for navigation) << (Total dataset objects)
  - Algorithm complexity logarithmic in dataset size

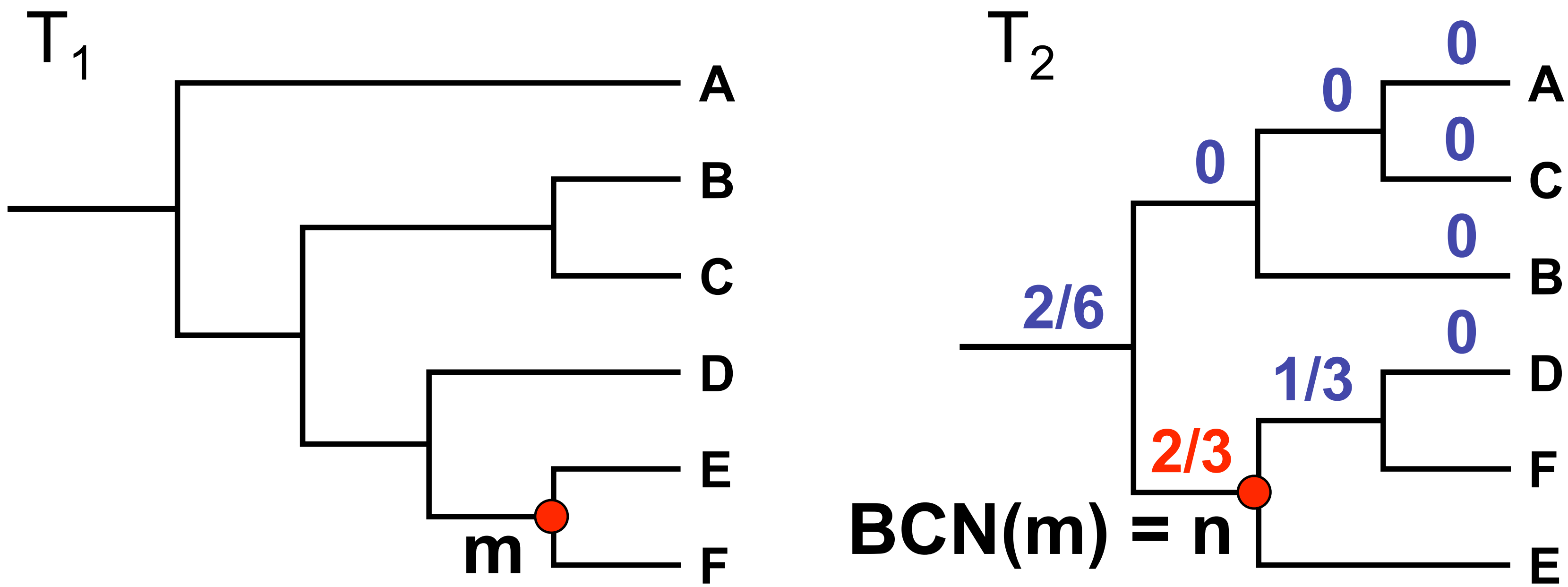
Tree Diff



$$L(\mathrm{m}) = \{\mathrm{E,F}\} \qquad L(\mathrm{n}) = \{\mathrm{D,E,F}\}$$

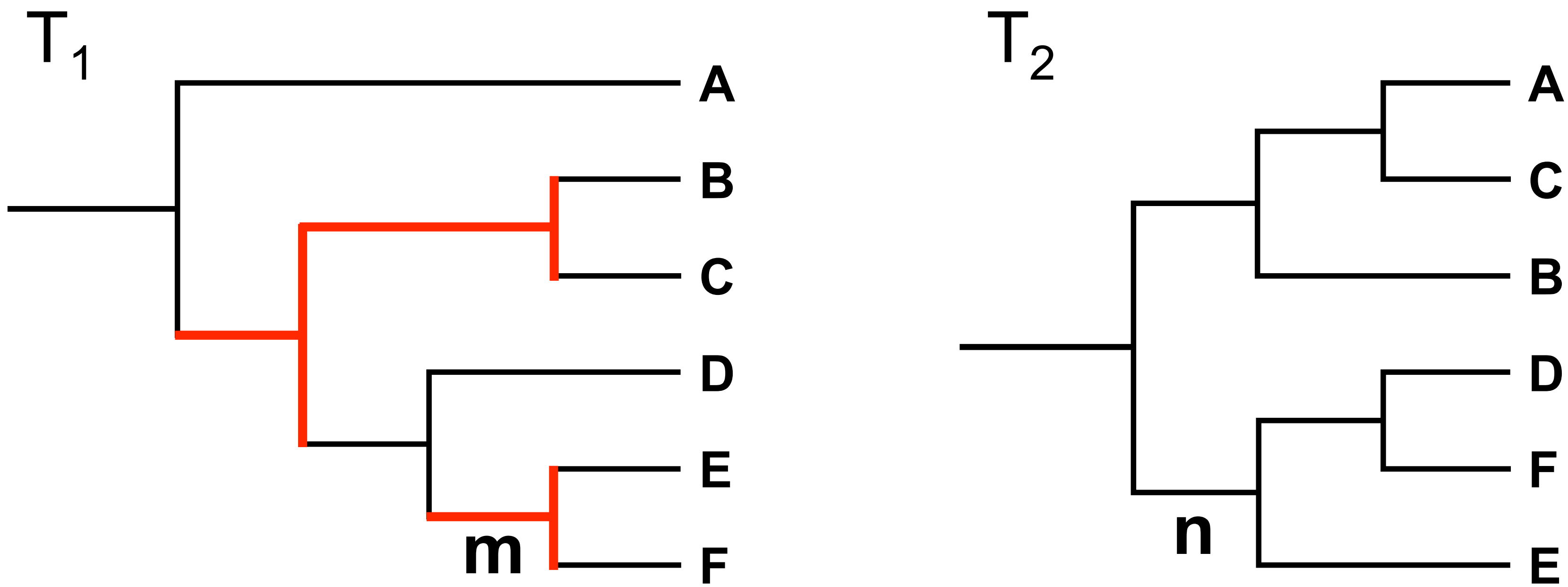
$$S(\mathrm{m,n}) = \frac{|L(\mathrm{m}) \cap L(\mathrm{n})|}{|L(\mathrm{m}) \cup L(\mathrm{n})|} = \frac{|\{\mathrm{E,F}\}|}{|\{\mathrm{D,E,F}\}|} = \frac{2}{3}$$

Best Corresponding Node



- $\text{BCN(m)} = \text{argmax}_{v \in \text{T}_2}(S(\text{m}, v))$
  - computable in $O(n \log^2 n)$
  - linked highlighting

Marking Structural Differences

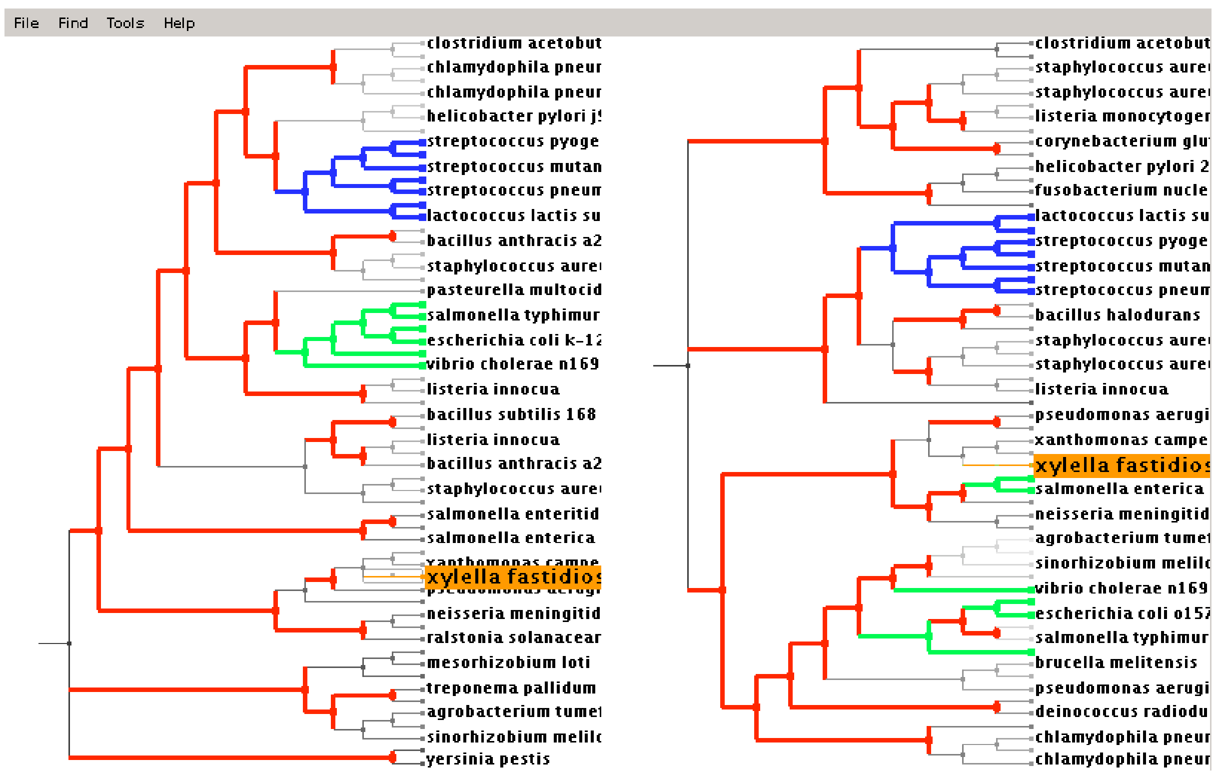


- Nodes for which $S(v,\mathrm{BCN}(v)) \neq 1$
  - Matches intuition

[TreeJuxtaposer: Scalable Tree Comparison using Focus+Context with Guaranteed Visibility. Tamara Munzner, François Guimbretière, Serdar Tasiran, Li Zhang, Yunhong Zhou. Proc SIGGRAPH 2003]

Guaranteed Visibility

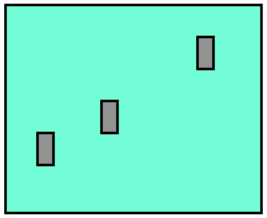
- marks are always visible
  - regions of interest shown with color highlights
  - search results, structural differences, user specified
- easy with small datasets



14

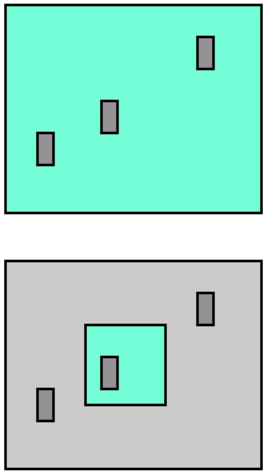
Guaranteed Visibility Challenges

- hard with larger datasets
- reasons a mark could be invisible

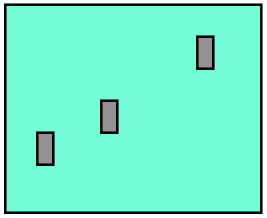


Guaranteed Visibility Challenges

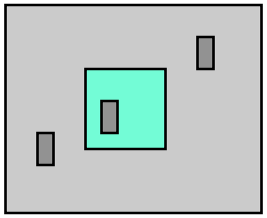
- hard with larger datasets
- reasons a mark could be invisible
  - outside the window
    - AD solution: constrained navigation



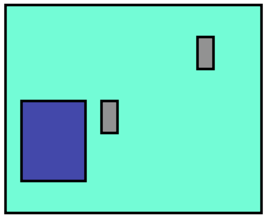
Guaranteed Visibility Challenges



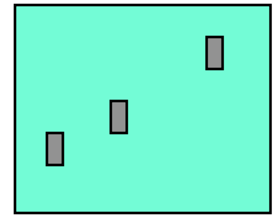
- hard with larger datasets
- reasons a mark could be invisible
  - outside the window
    - AD solution: constrained navigation



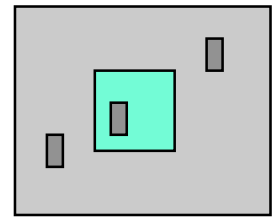
  - underneath other marks
    - AD solution: avoid 3D



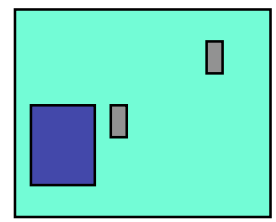
Guaranteed Visibility Challenges



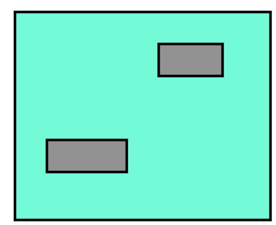
- hard with larger datasets
- reasons a mark could be invisible
  - outside the window
    - AD solution: constrained navigation



  - underneath other marks
    - AD solution: avoid 3D

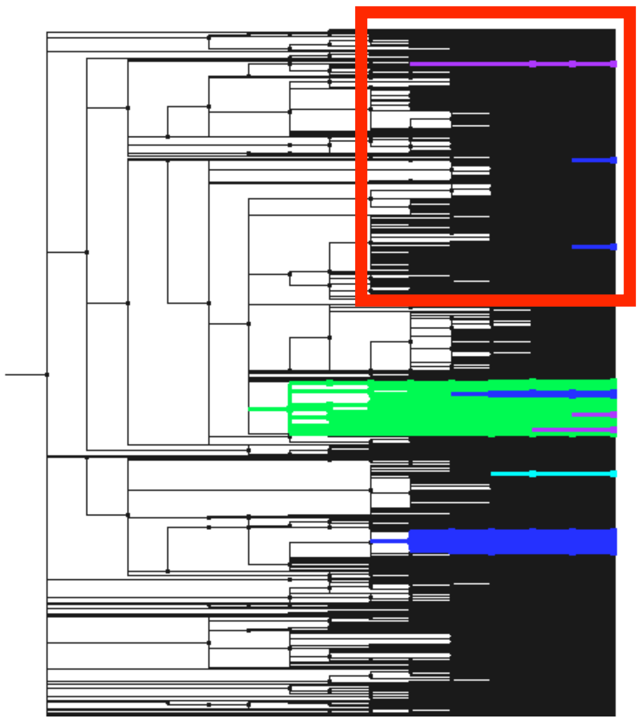


  - smaller than a pixel
    - AD solution: smart culling

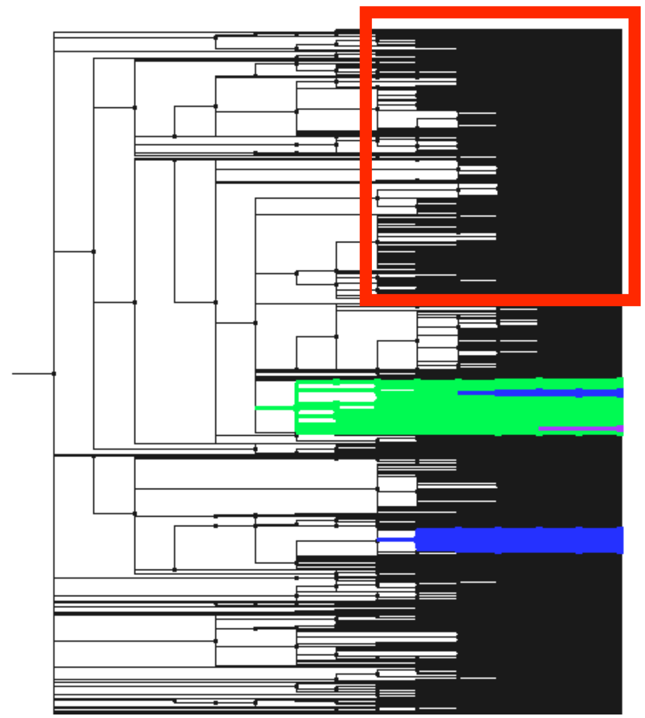


Guaranteed Visibility: Small Items

- Naïve culling may not draw all marked items



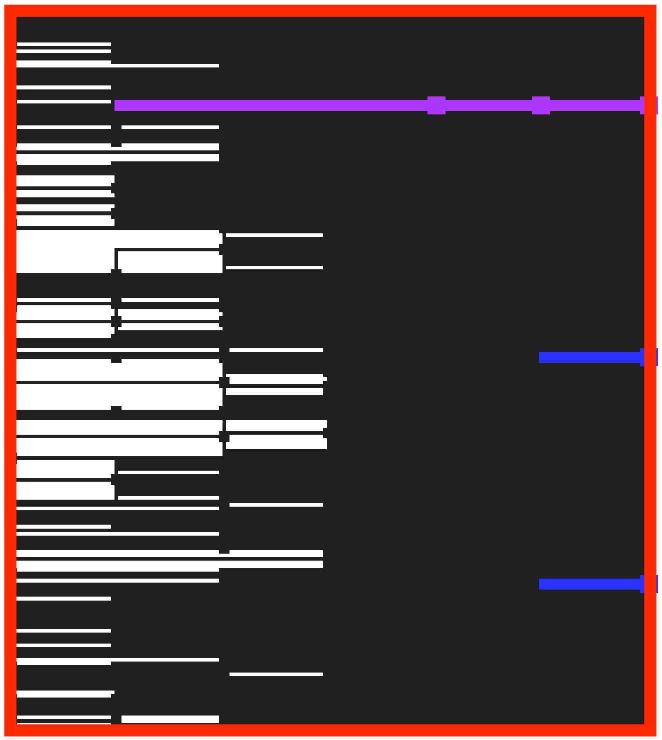
Guaranteed visibility of marks



No guaranteed visibility

Guaranteed Visibility: Small Items

- Naïve culling may not draw all marked items



Guaranteed visibility of marks

No guaranteed visibility

Guaranteed Visibility Rationale

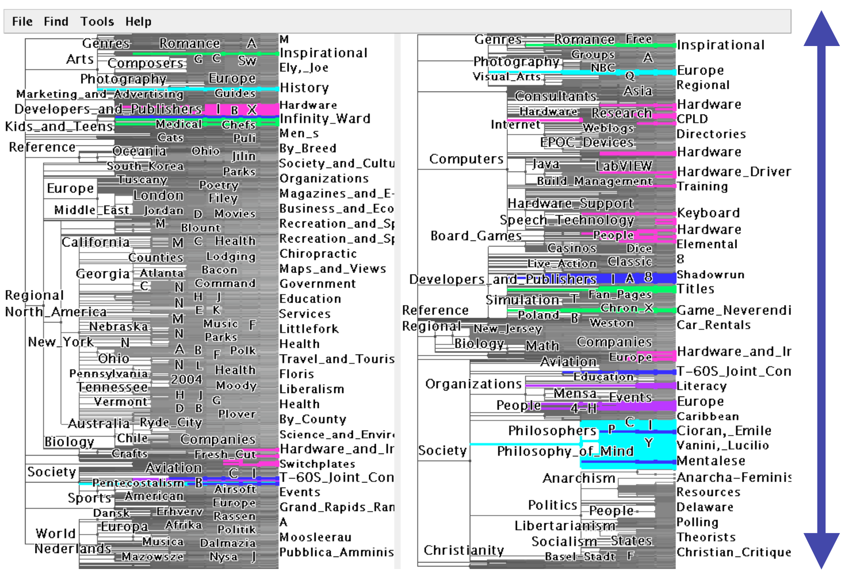
- relief from exhaustive exploration
  - missed marks lead to false conclusions
  - hard to determine completion
  - tedious, error-prone

- compelling reason for Focus+Context
  - controversy: does distortion help or hurt?
  - strong rationale for comparison

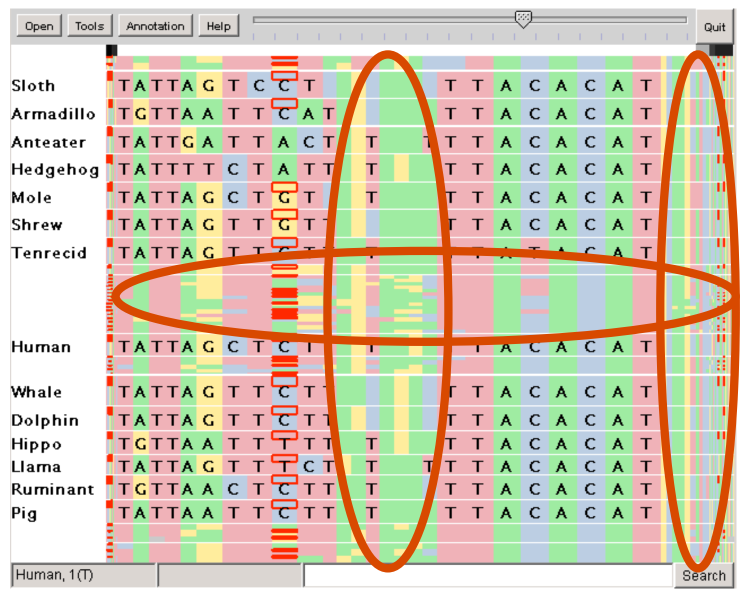
- infrastructure needed for efficient computation

Rending Complexity

- Reduce drawing complexity with sneaky culling
  - For each frame: draw representative visible subset, not entire dataset
  - (Total number of drawn objects per frame) << (Total dataset items)
    - In tree dataset with 600,000 leaves, draw only 1000 leaves
    - In sequence datasets, aggregate dense regions in software



1000 leaves visible



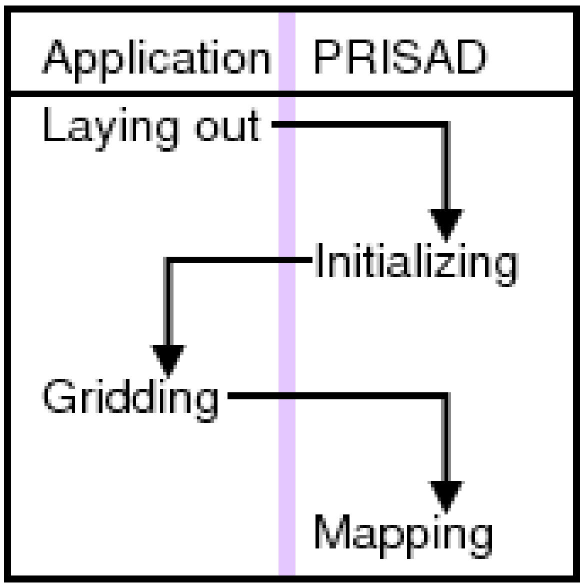
Dense, culled regions

[Partitioned Rendering Infrastructure for Scalable Accordion Drawing (Extended Version). James Slack, Kristian Hildebrand, and Tamara Munzner. Information Visualization, 5(2), p. 137-151, 2006]

PRISAD Architecture

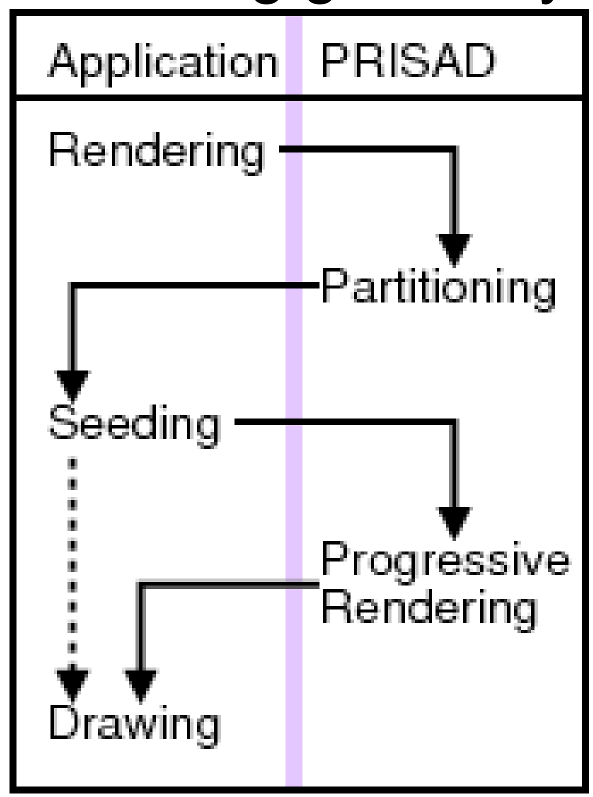
world-space discretization

- preprocessing
  - initializing data structures
  - placing geometry



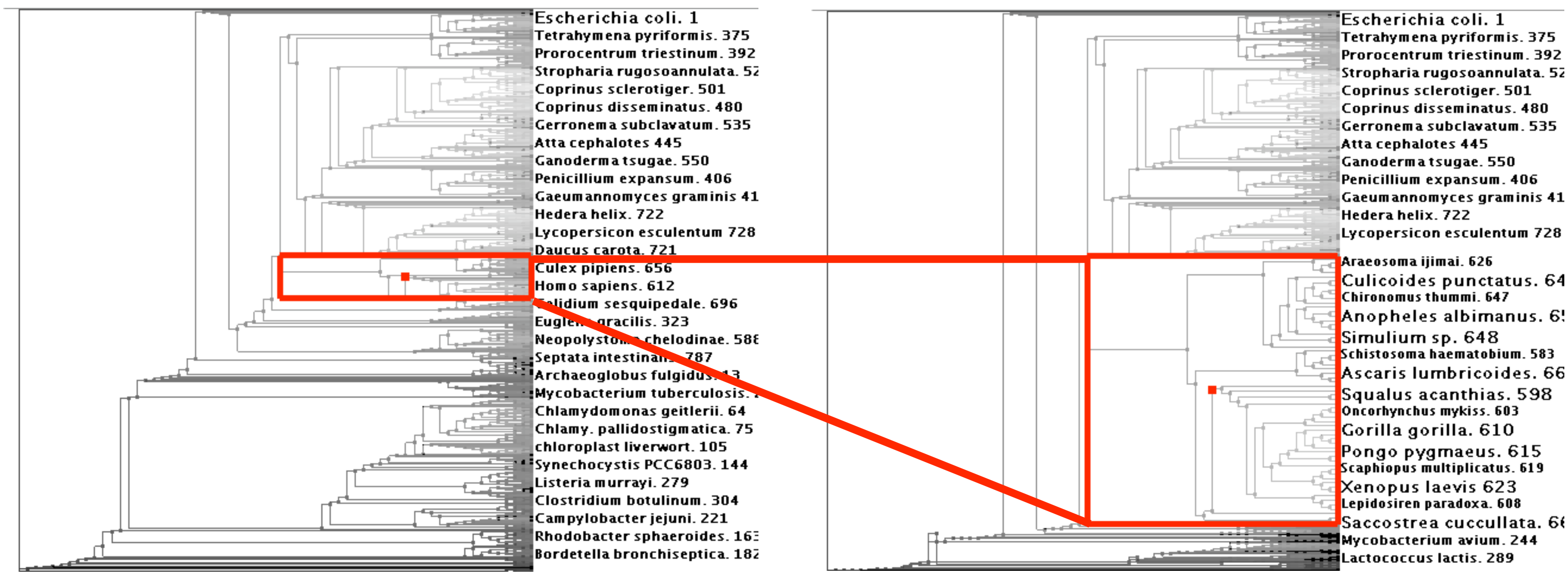
screen-space rendering

- frame updating
  - analyzing navigation state
  - drawing geometry



Stretch and Squish Navigation

- User selects any region to grow or shrink
  - Everything else shrinks or grows, accordingly
- Goal: handle millions of items, landmarks always stay visible



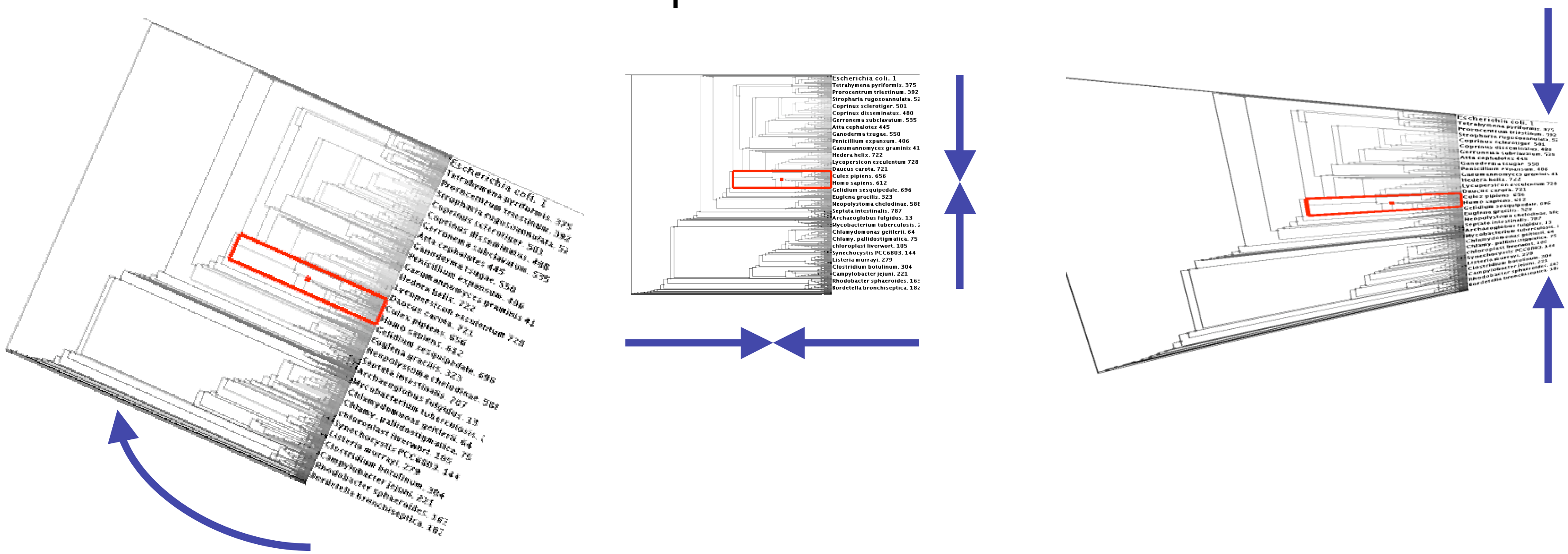
Growing a region

Composite Rectilinear Deformation for Stretch and Squish Navigation. James Slack and Tamara Munzner. Proc. Visualization 2006, published as Transactions on Visualization and Computer Graphics 12(5), September 2006

Successive Navigations Preserve Visual History

Implementing Stretch and Squish Navigation

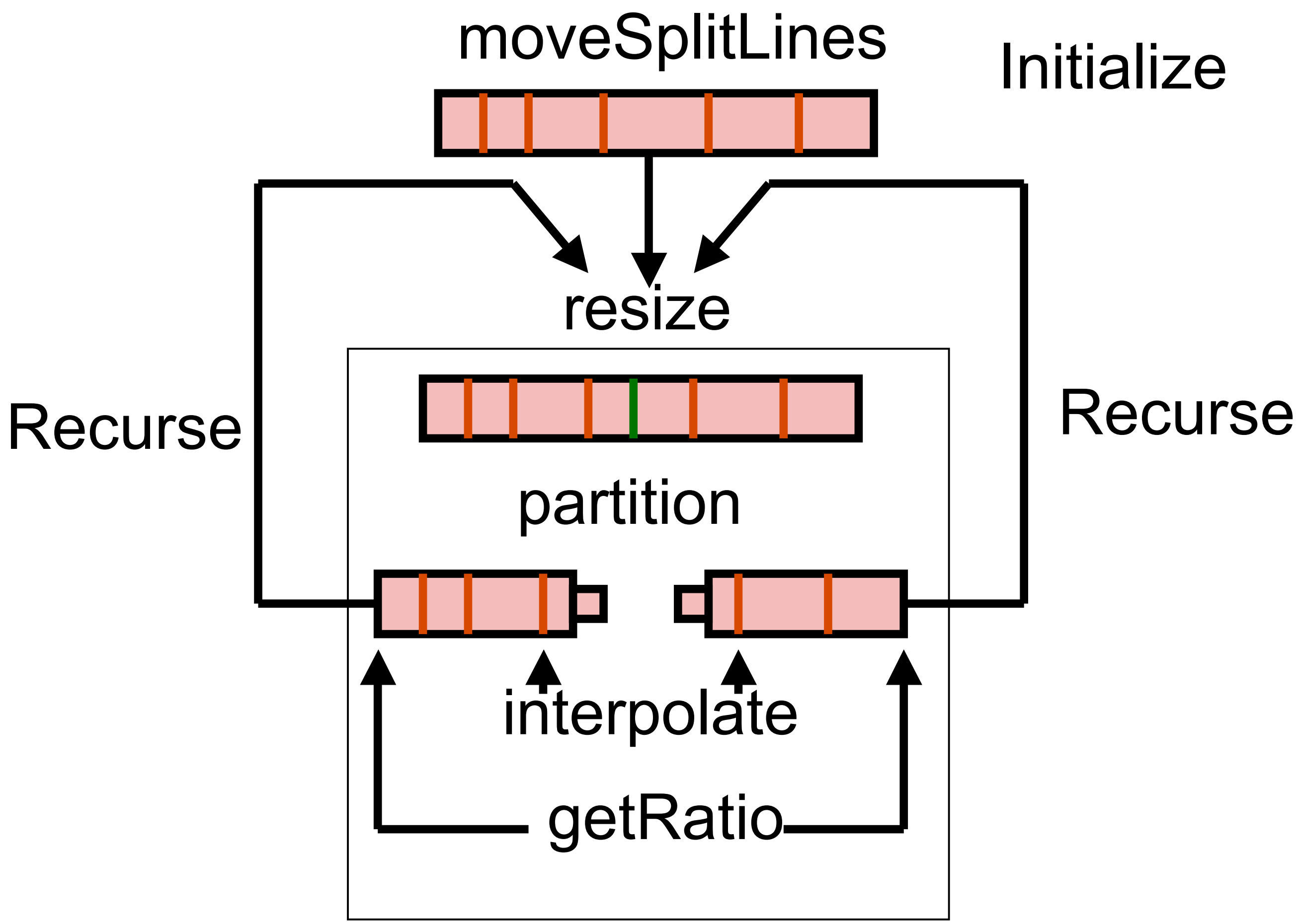
- Simple to use
- Underlying infrastructure is complex to implement
  - Standard graphics pipeline has a single, monolithic transformation
    - Fast 4x4 matrix multiplication



  - **Stretch and squish cannot be implemented using this pipeline**

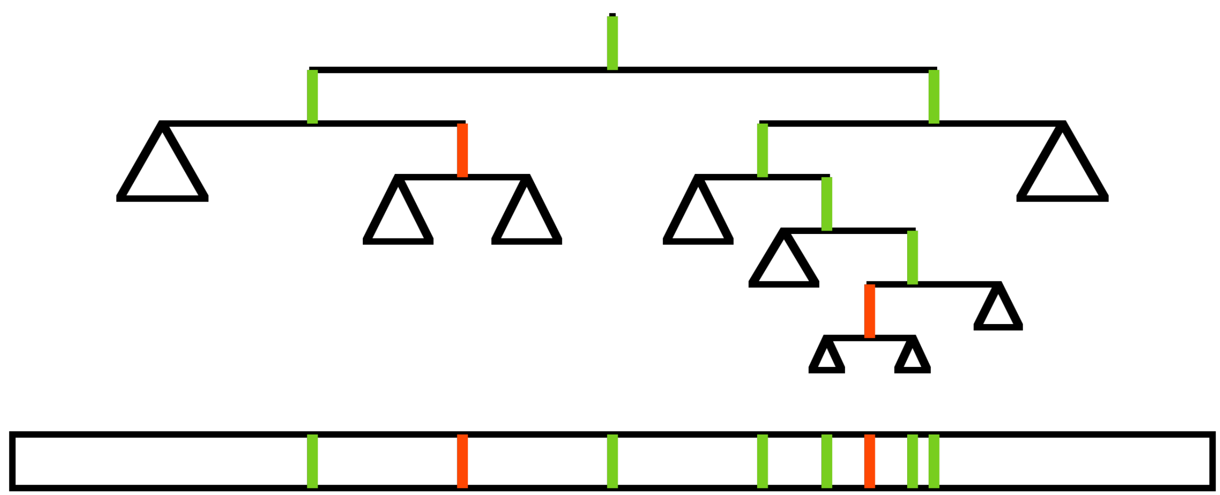
Navigation Algorithm

- Flow of our navigation algorithm:



Navigation Algorithm Complexity

- Logarithmic complexity: |Q| ≈ |K| log |N| << |N|
  - Q: Lines needing ratio updates
  - K: Lines to move
  - N: All lines
- Many positions change, but few ratios require updates
  - Moving 2 grid lines only requires changing ratios for 8 split lines
  - Subtrees not affected will conserve their internal ratios



- Speed: under 1 millisecond for |N| = 2,000,000 lines

Lots More Information

- download software: http://olduvai.sf.net
  - TreeJuxtaposer, SequenceJuxtaposer

- many papers, talks, videos: http://www.cs.ubc.ca/~tmm
  - Composite Rectilinear Deformation for Stretch and Squish Navigation. James Slack and Tamara Munzner. Proc. Visualization 2006, published as Transactions on Visualization and Computer Graphics 12(5), September 2006.
  - Partitioned Rendering Infrastructure for Scalable Accordion Drawing (Extended Version). James Slack, Kristian Hildebrand, and Tamara Munzner. Information Visualization, 5(2), p. 137-151, 2006
  - SequenceJuxtaposer: Fluid Navigation For Large-Scale Sequence Comparison In Context. James Slack, Kristian Hildebrand, Tamara Munzner, and Katherine St. John. German Conference on Bioinformatics 2004, pp 37-42
  - TreeJuxtaposer: Scalable Tree Comparison using Focus+Context with Guaranteed Visibility. Tamara Munzner, François Guimbretière, Serdar Tasiran, Li Zhang, and Yunhong Zhou. SIGGRAPH 2003, pp 453--462