Approximation Algorithms for Embedding General Metrics Into Trees

Mihai Badoiu (MIT)

Piotr Indyk (MIT)

Anastasios Sidiropoulos (MIT)

Metric Embeddings

- Given finite **metric space** *M=(X,D), M'=(Y,D')*
- Mapping $f : X \rightarrow Y$
- **Distortion** of *f* is

$$\max_{x_1,x_2} \frac{D'(f(x_1),f(x_2))}{D(x_1,x_2)} \times \max_{y_1,y_2} \frac{D(y_1,y_2)}{D'(f(y_1),f(y_2))}$$

GOAL : minimize distortion

Two kinds of problems:

- **Worst-case** embeddings
- **Relative** embeddings

Relative Embeddings - Known Results

From	Into	Upper	Lower	Citation
General	L_2	c		[Linial,London,Rabinovich 94]
Unweighted graphs	Line	$O(c^2)$	$1.01c$	[Badoiu,Dhamdhere,Gupta, Rabinovich,Raecke,Ravi,S.'05]
Unweighted trees	Line	$O(c^{3/2})$		[Badoiu,Dhamdhere,Gupta, Rabinovich,Raecke,Ravi,S.'05]
General	Line	$O(\Delta^{3/4}c^{11/4})$		[Badoiu,Chuzhoy,Indyk,S.'05]
Trees	Line	$c^{O(1)}$	$\Omega(n^{1/12}c)$	[Badoiu,Chuzhoy,Indyk,S.'05]
Ultrametrics	$\mathbf{R}^d$	$c^{O(d)}$	NP-complete	[Badoiu,Chuzhoy,Indyk,S.'06]
Unweighted graphs	Sub-trees	$O(c \log n)$	$\Omega(c)$	[Emek,Peleg'04]

Motivation

- Computational Biology: **Phylogenetic Reconstruction**

- Computer Networks: **Tree-Spanners** for weighted complete graphs

Our Results

- Embedding **unweighted graphs** into trees, with distortion *O(c)*.
- Embedding **general metrics** into trees, with distortion

  $a^*(\text{poly}(c)^*\log(n))^{\log\Delta/\log(a)}$, for any a>1 (Δ: spread)

  - Setting $a = 2^{\sqrt{\log \Delta}}$, distortion

    $$(c \cdot \log n)^{O(\sqrt{\log \Delta})}$$

  - When $\Delta = n^{O(1)}$, setting $a = n^{\varepsilon}$, distortion

    $$n^{\epsilon}(c \log n)^{O(1/\epsilon)}$$

  By [Matousek'90], implies $O(n^{1-b})$-approximation
- Composing with algorithm of [Badoiu,Chuzhoy,Indyk,S.'05], for embedding trees into the line, we obtain an algorithm for embedding **general metrics into the line**, with the above guarantees.

Our Results (cont.)

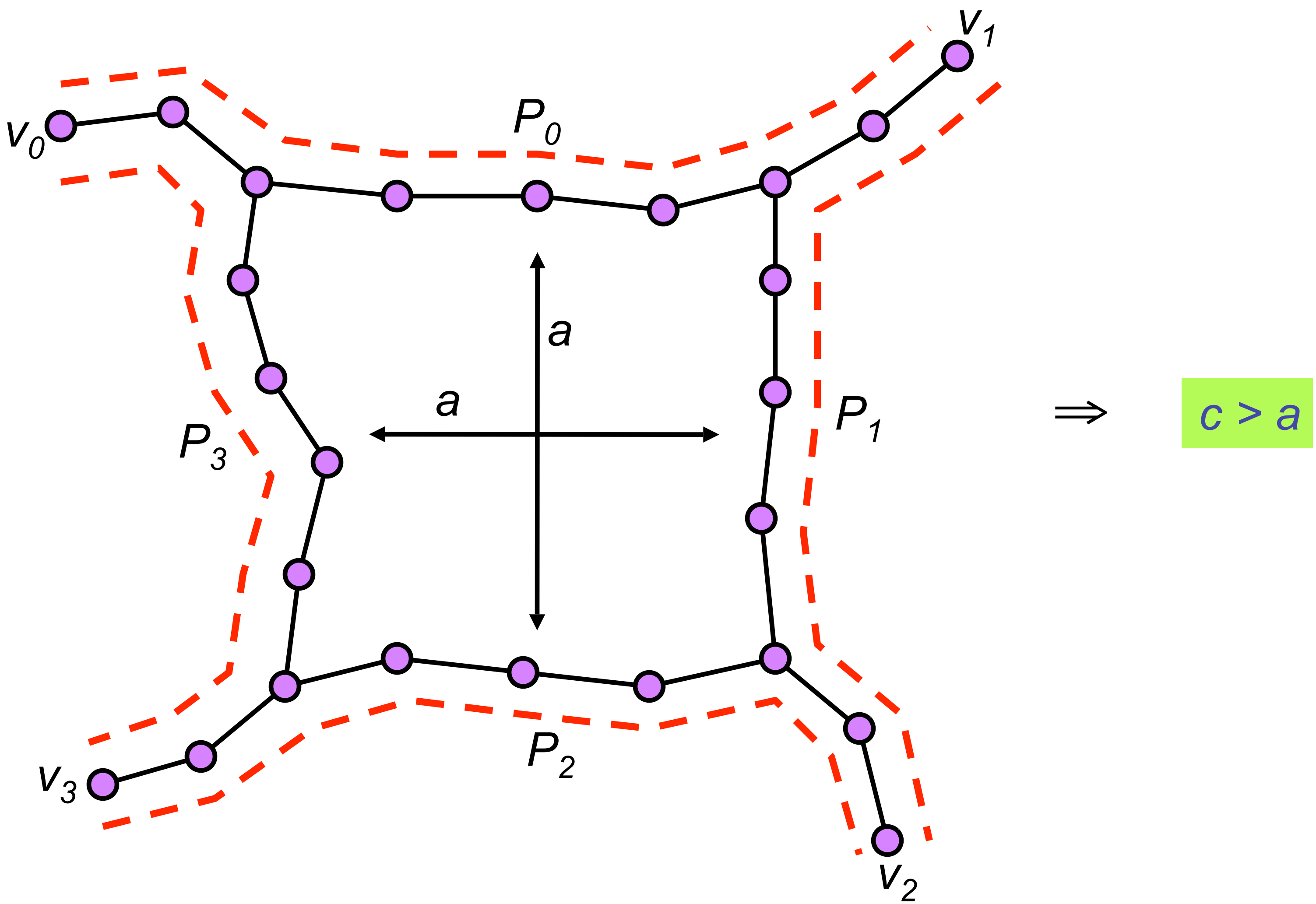
Trees vs. Sub-trees:

- If an unweighted graph embeds into a **tree** with distortion *c*, then it embeds into a **sub-tree** with distortion $O(c \log n)$.
- There exist graphs that embed with distortion *c* into a **tree**, and any embedding into a **sub-tree** requires distortion $\Omega(c \log n / \log\log n)$.

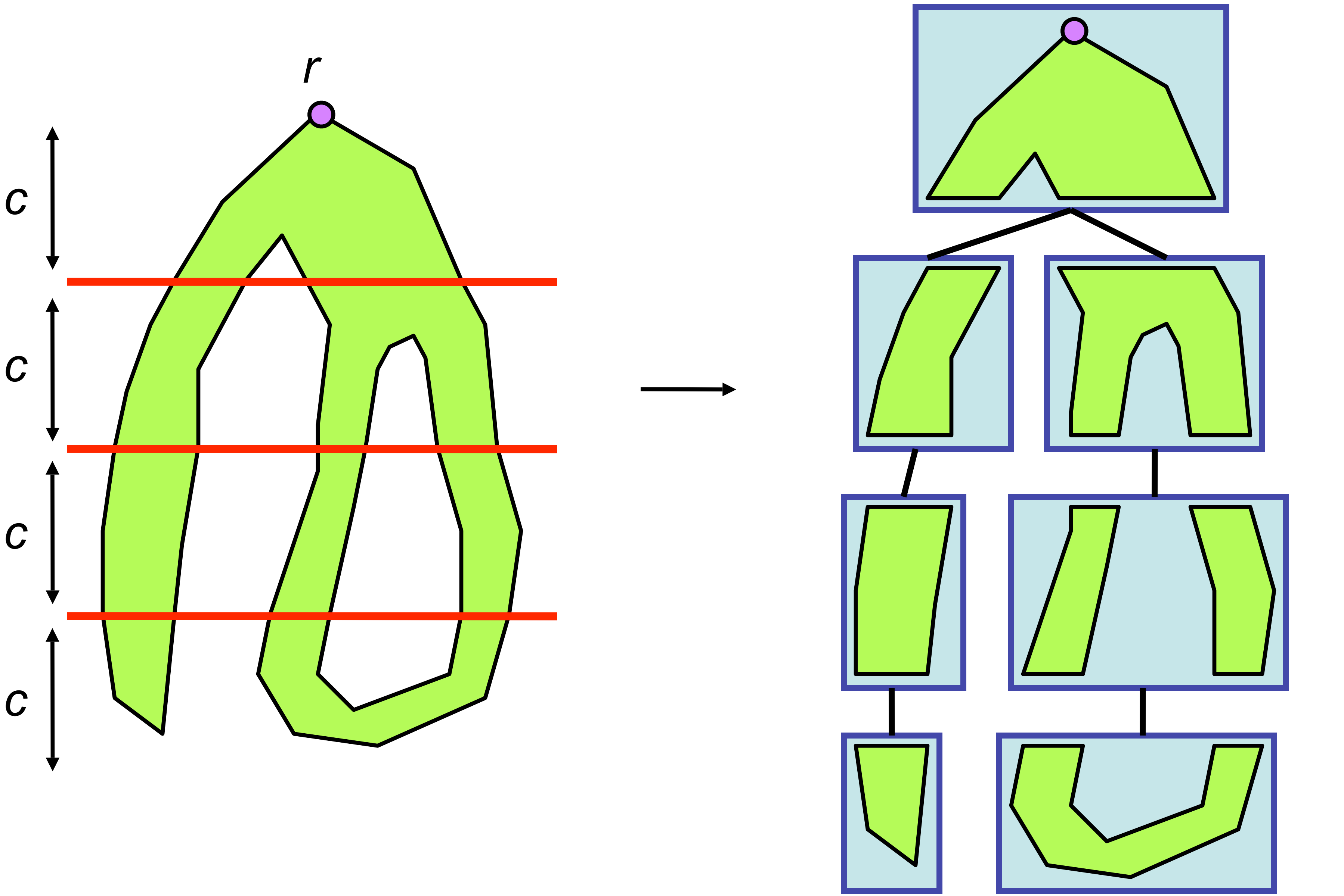
Part 1: Unweighted Graphs

A Forbidden Structure (cont.)

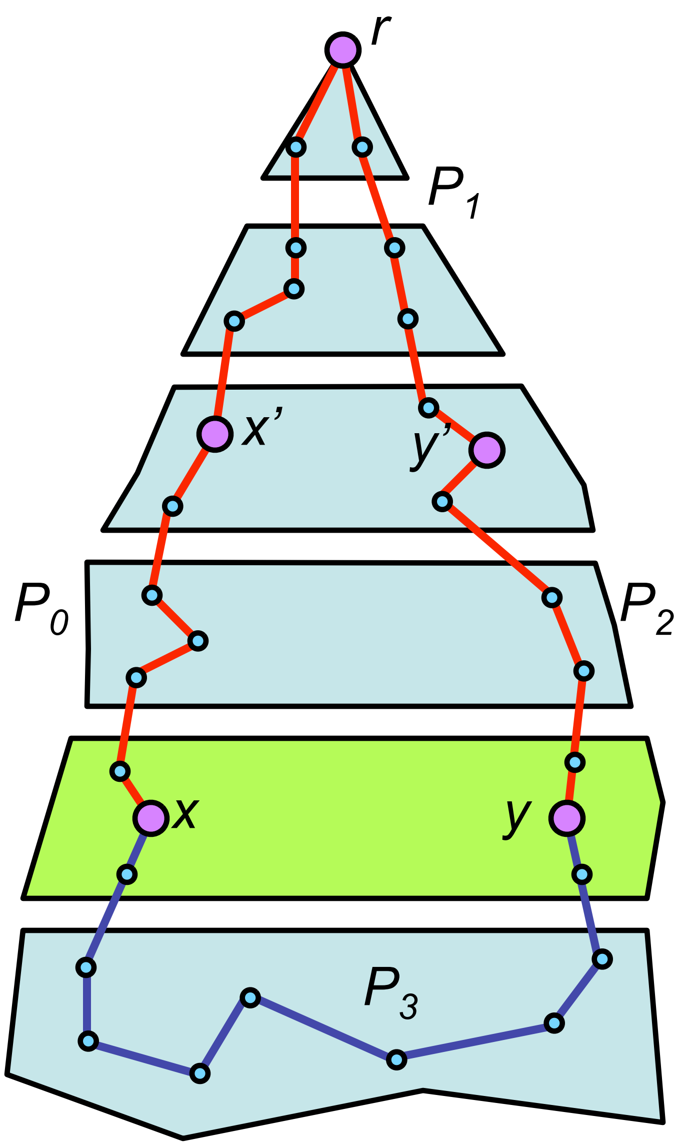
If there exists v_0, v_1, v_2, v_3, and P_0, P_1, P_2, P_3, s.t. $D(P_i,P_j) > a$:



Tree-Like Decompositions



Tree-Like Decompositions: Small Diameter



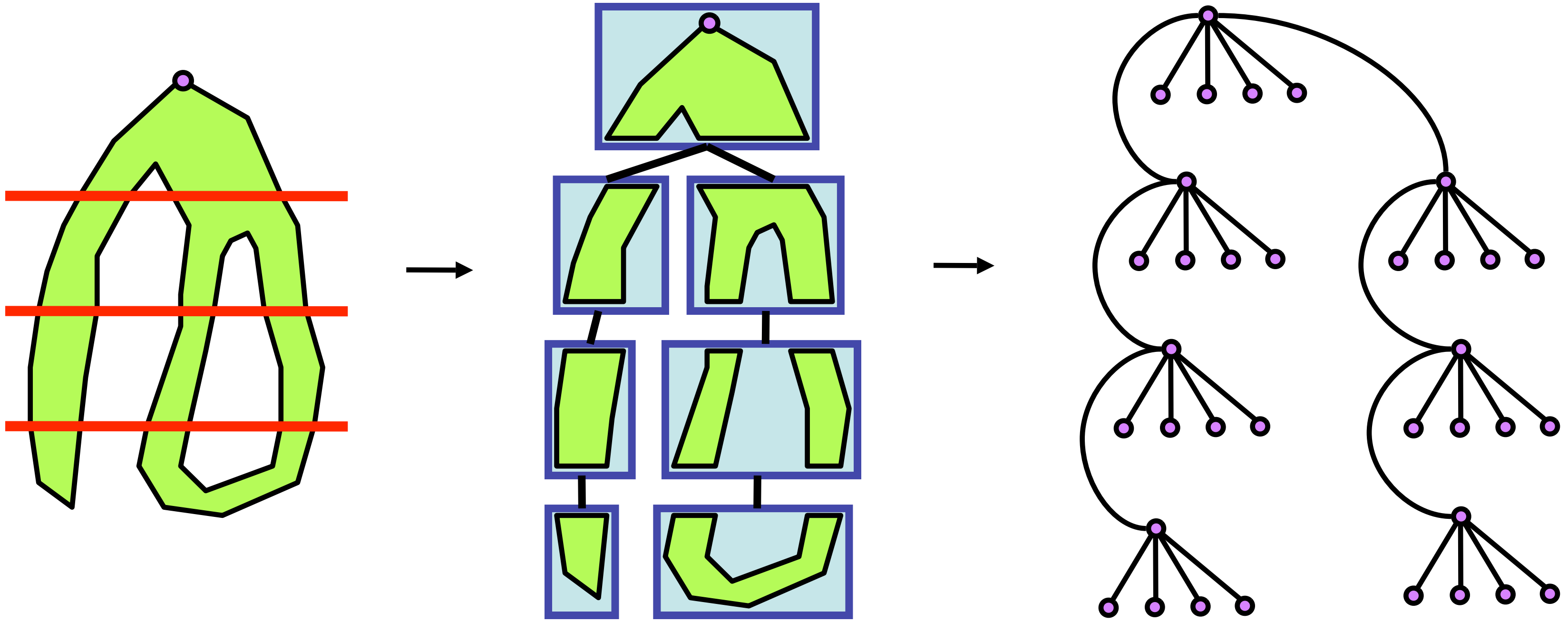
If $D(x,y) > 10\,c$, then P_0, P_1, P_2, P_3, form a forbidden structure.

$\Rightarrow$

The diameter of each cluster is $O(c)$.

The Algorithm

1. Compute tree-like decomposition
2. Replace each cluster with a star
3. Connect the stars in a tree

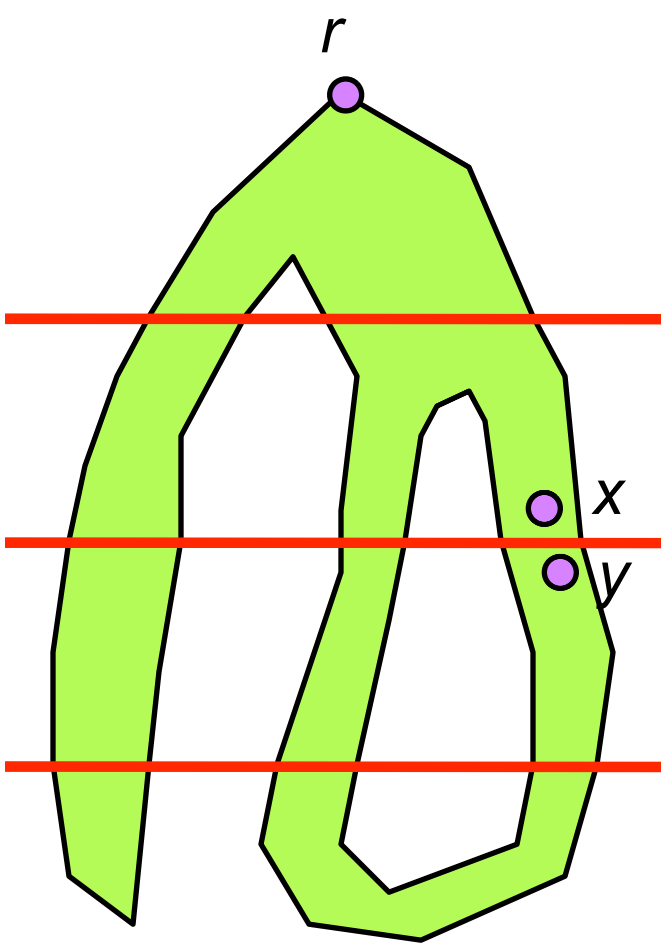


Distortion *O(c)*

Part 2: General Metrics

Tree-Like Decompositions do not work

Obstacle: Clusters small-diameter might be arbitrarily close



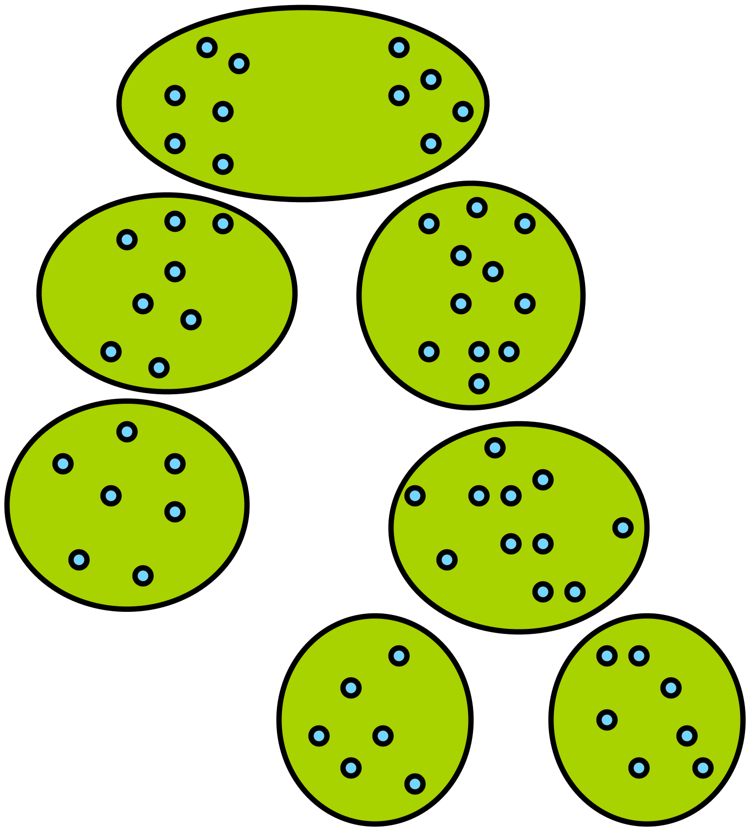
$D(x,y)=\varepsilon$

Question: How to deal with different scales?

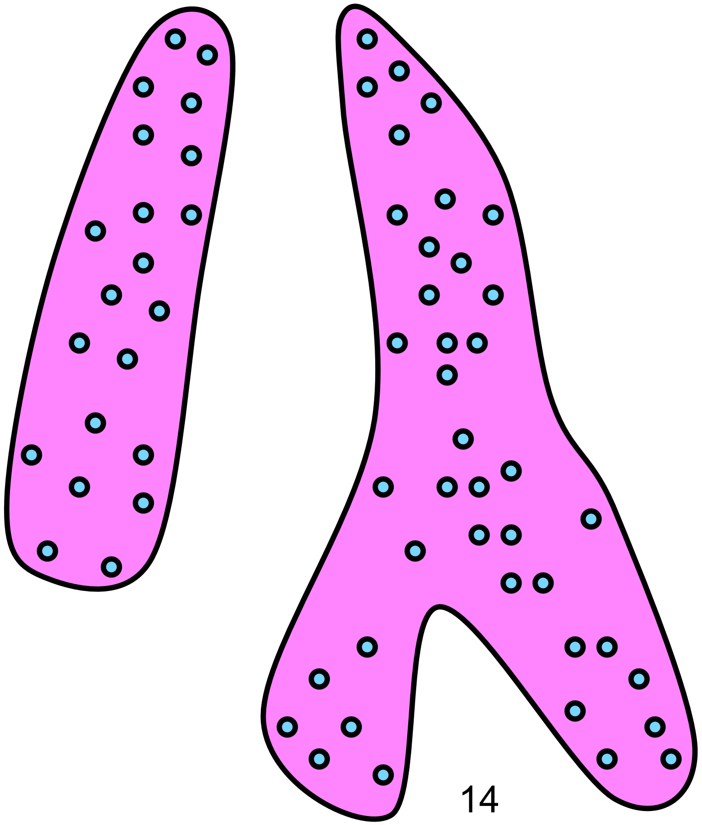
Well-Separated Tree-Like Decompositions

Idea: Use 2 decompositions

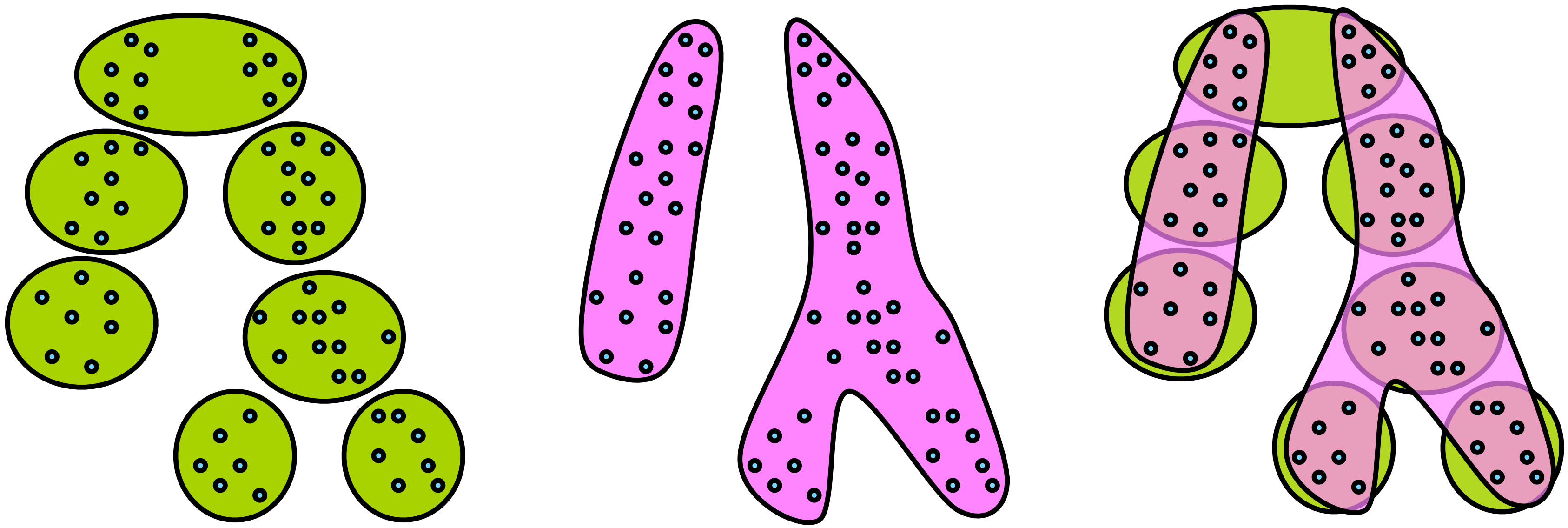
Small-diameter decomposition



Well-separated decomposition



Well-Separated Tree-Like Decompositions



Properties:

- Clusters of well-separated decomposition are **subtrees** of the small-diameter decomposition.
- These subtrees have **small intersection**.

The Algorithm

1. Compute well-separated, tree-like decomposition
2. Recurse inside each well-separated cluster
3. Merge solutions using the tree-like decomposition

Analysis:

- At every recursive iteration, we reduce the maximum edge weight by a.
- At each iteration, we accumulate distortion poly(c)*log(n) in the merging step.
- Resulting distortion a*(poly(c)*log(n))$^{\log\Delta/\log(a)}$

Setting $a = 2^{\sqrt{\log \Delta}}$

Distortion: $(c \cdot \log n)^{O(\sqrt{\log \Delta})}$

The End

- Can we do better? Poly(c) distortion?

- Other forbidden-structure characterizations of tree-embedability?