CS 68: BIOINFORMATICS

Prof. Sara Mathieson
Swarthmore College
Spring 2018

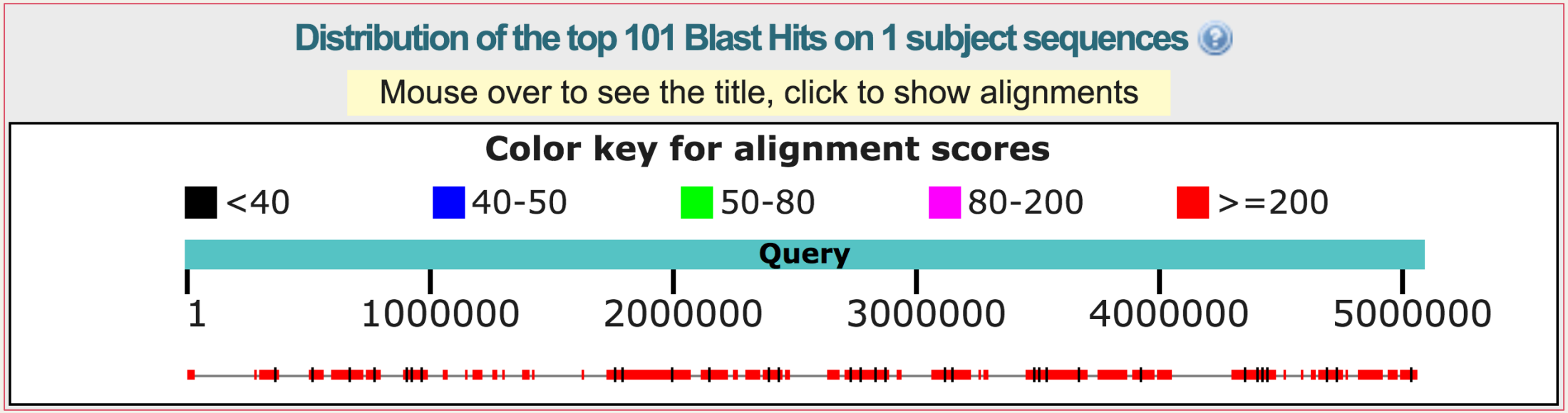
Outline: Feb 21

- Brief review for Midterm 1
- Continue UPGMA
- Tree metrics and ultrametrics

Notes:

- No office hours today
- Lab this week: practice midterm
- No class on Friday
- Reading posted (Durbin Chap. 7)

Best E.coli challenge assembly: Hunter and Sam



Sequences producing significant alignments:

Select: All None Selected:0

Alignments Download Graphics

	Description	Max score	Total score	Query cover	E value	Ident	Accession
☐	contig1	3.672e+05	7.019e+06	58%	0.0	99%	Query_84173

Recap for Midterm 1

Topics for Midterm 1

1) Introduction and Central Dogma
2) Genome Assembly
3) Pairwise Sequence Alignment
4) BWT and Read Mapping

(1) Introduction and Central Dogma

- DNA, RNA, Protein
- How do we transition between these molecules?
- What does the central dogma say about which transitions are possible?

(2) Genome Assembly

- Often the first step in studying the genetics of a new species
- Input: millions-billions of reads (used to be “long” reads, now are “short”)
- Output: contigs (ideally long and accurate, making up as much of the original genome as possible)
- Overlap graph assembly (Overlap Layout Consensus: OLC). Accurate but very slow
- De Bruijn graph (DBG) assembly. Fast but sometimes not as accurate
- What are the runtimes of these assembly algorithms in terms of n, L, G?

(3) Pairwise Sequence Alignment

- Used for studying the relationship between homologous sequences (often genes or regions from different species)
- Could be run after assembling two very different species
- Could be run on repetitive but diverged regions from the same individual
- We are giving up runtime by allowing gaps and mismatches
- Input: two sequences *x* and *y*, typically of similar length but not always. We also need a substitution matrix and gap penalty
- Output: optimal alignment(s) between *x* and *y*, AND an alignment score (higher is more similar, negative is usually not biologically meaningful)
- Two dynamic programming variations: global sequence alignment (align entire *x* with entire *y*) and local alignment (align highly similar regions in *x* and *y*)

(4) BWT and Read Mapping

- Input: previously assembled reference sequence and millions-billions of reads from a new individual of the same species
- Output: the location(s) where each read maps (+ where the mismatches are)
- Pairwise sequence alignment is too slow
- What is the runtime of constructing the BWT and FM-Index? After that, what is the runtime of pattern matching? (see Lab 4)

Back to UPGMA

Dissimilarity maps

- Record pairwise differences (which could be obtained from a pairwise sequence alignment)
- We will use a dissimilarity map as input to our phylogenetic tree algorithms

A *dissimilarity map* δ is a function mapping pairs of samples from a set $\mathcal{X}$ to distances. It has the following two properties, but not necessarily the triangle inequality.

1. $\delta(x, x) = 0$
2. $\delta(x, y) = \delta(y, x)$

Example:

δ	A	B	C	D	E
A	0	1	3	6	6
B		0	2	5	5
C			0	5	5
D				0	2
E					0

UPGMA

UPGMA initialization:

1. Each sample $x \in \mathcal{X}$ starts in it's own cluster $C_x = \{x\}$
2. Set cluster distances $\Delta(C_i, C_j) = \delta(i, j)$ for all i, j

UPGMA

UPGMA initialization:

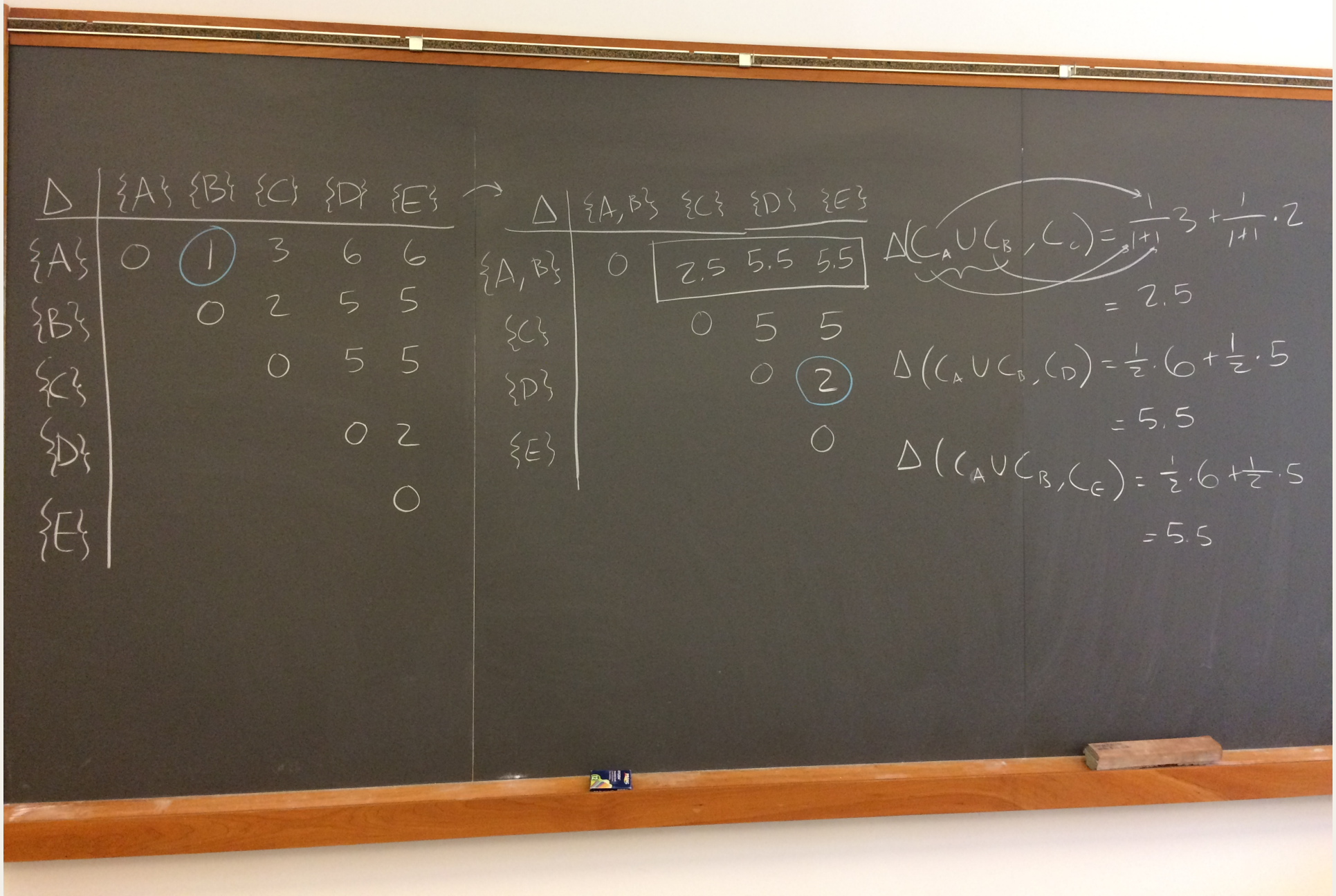
1. Each sample $x \in \mathcal{X}$ starts in it's own cluster $C_x = \{x\}$
2. Set cluster distances $\Delta(C_i, C_j) = \delta(i, j)$ for all i, j

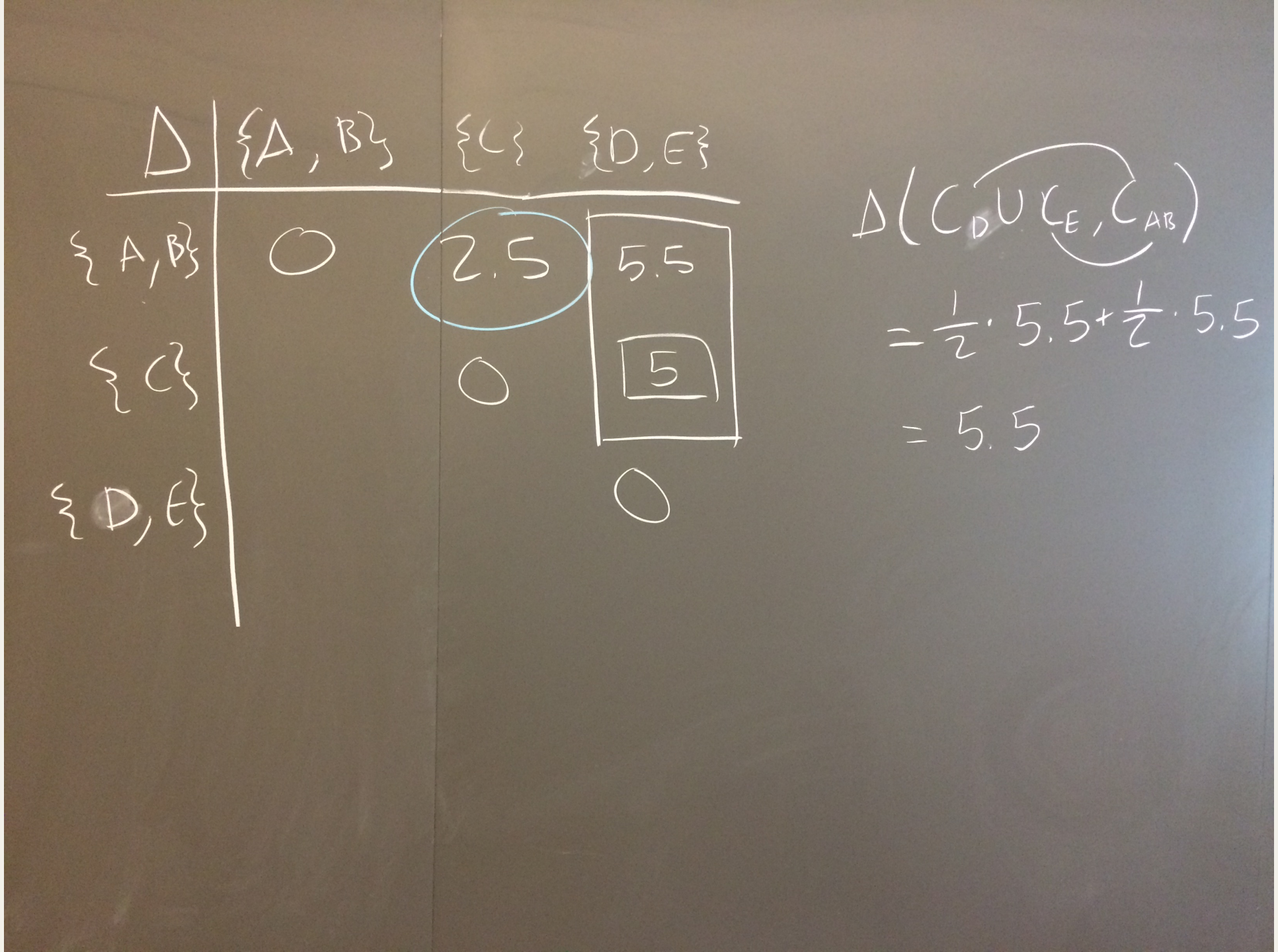
UPGMA update:

1. Find C_i and C_j (where $i \neq j$) that minimize $\Delta(C_i, C_j)$, and merge to create $C_{ij} = C_i \cup C_j$
2. Set the distances from C_{ij} to every other cluster C_k using the update rule:

$$\Delta(C_i \cup C_j, C_k) = \frac{|C_i|}{|C_i| + |C_j|}\Delta(C_i, C_k) + \frac{|C_j|}{|C_i| + |C_j|}\Delta(C_j, C_k)$$

3. Join C_i and C_j with interior vertex v; set the height of v equal to $\Delta(C_i, C_j)/2$


Δ {A} {B} {C} {D} {E}
{A} 0 1 3 6 6
{B} 0 2 5 5
{C} 0 5 5
{D} 0 2
{E} 0
Δ {A,B} {C} {D} {E}
{A,B} 0 2.5 5.5 5.5
{C} 0 5 5
{D} 0 2
{E} 0
Δ(C_A ∪ C_B, C_C) = 1/(1+1) 3 + 1/(1+1) · 2
= 2.5
Δ(C_A ∪ C_B, C_D) = 1/2 · 6 + 1/2 · 5
= 5.5
Δ(C_A ∪ C_B, C_E) = 1/2 · 6 + 1/2 · 5
= 5.5


Δ | {A, B} | {C} | {D, E}
{A, B} | 0 | 2.5 | 5.5
{C} | | 0 | 5
{D, E} | | | 0
Δ(C_D ∪ (C_E, C_AB))
= 1/2 · 5.5 + 1/2 · 5.5
= 5.5

Δ	$\{A,B,C\}$	$\{D,E\}$
$\{A,B,C\}$	0	$5\frac{1}{3}$
$\{D,E\}$		0

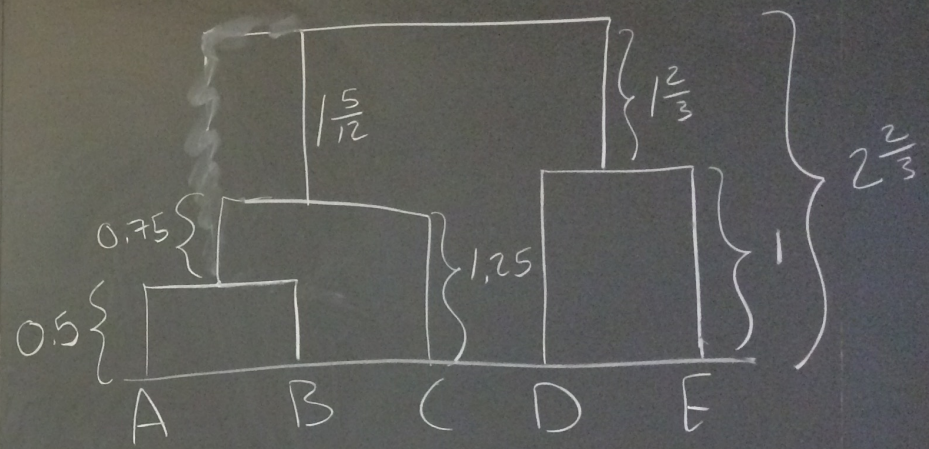
$$\Delta(C_{(AB)} \cup C_C, C_{DE})$$

$$= \frac{2}{3} \cdot 5.5 + \frac{1}{3} \cdot 5$$

$$= \frac{11+5}{3} = \frac{16}{3} = \boxed{5\frac{1}{3}}$$

induced tree metric

δ'	A	B	C	D	E
A	0	1	2.5	$5\frac{1}{3}$	$5\frac{1}{3}$
B		0	2.5	$5\frac{1}{3}$	$5\frac{1}{3}$
C			0	$5\frac{1}{3}$	$5\frac{1}{3}$
D				0	2
E					0



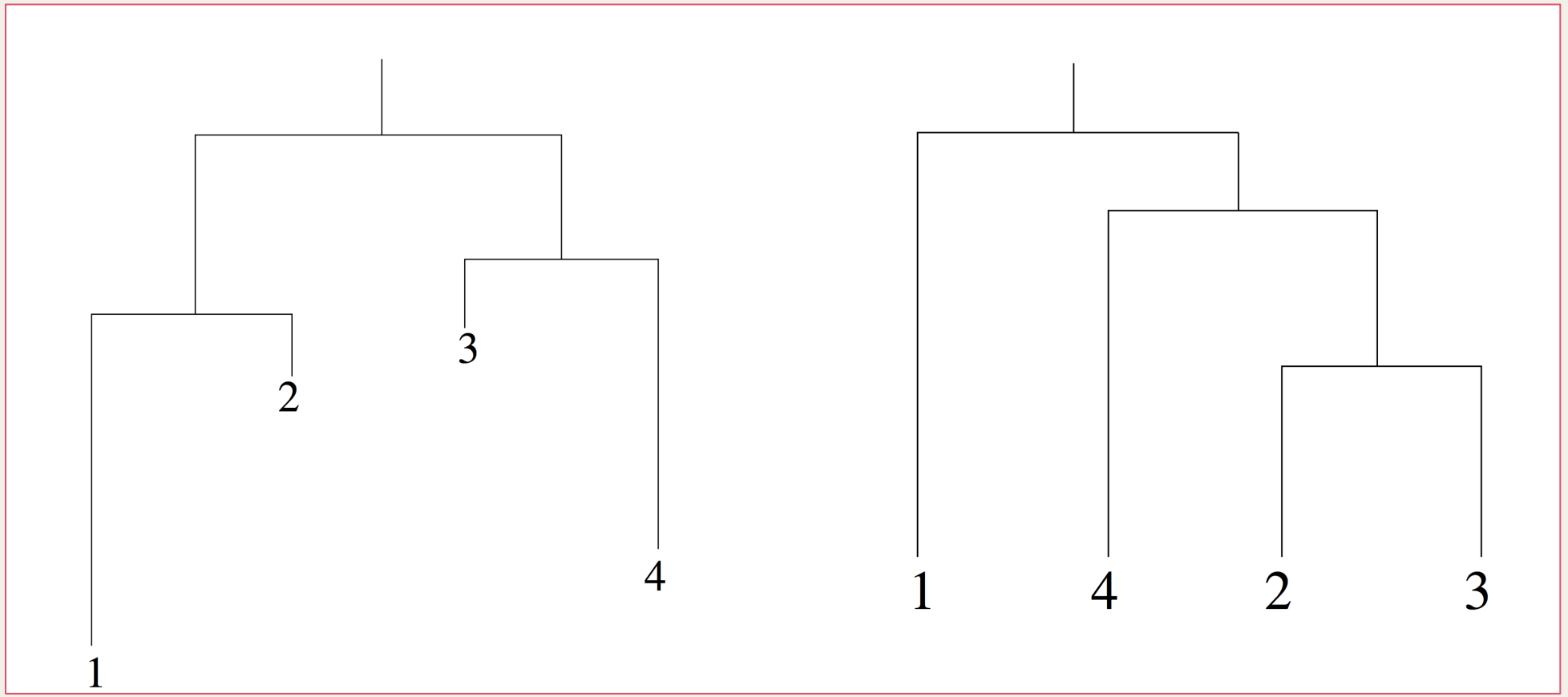
UPGMA

- UPGMA induces a *tree metric* on the samples in ***X***

A dissimilarity map δ is a *tree metric* if $\exists$ a tree topology and edges weights such that $\forall\ x, y \in \mathcal{X}$,

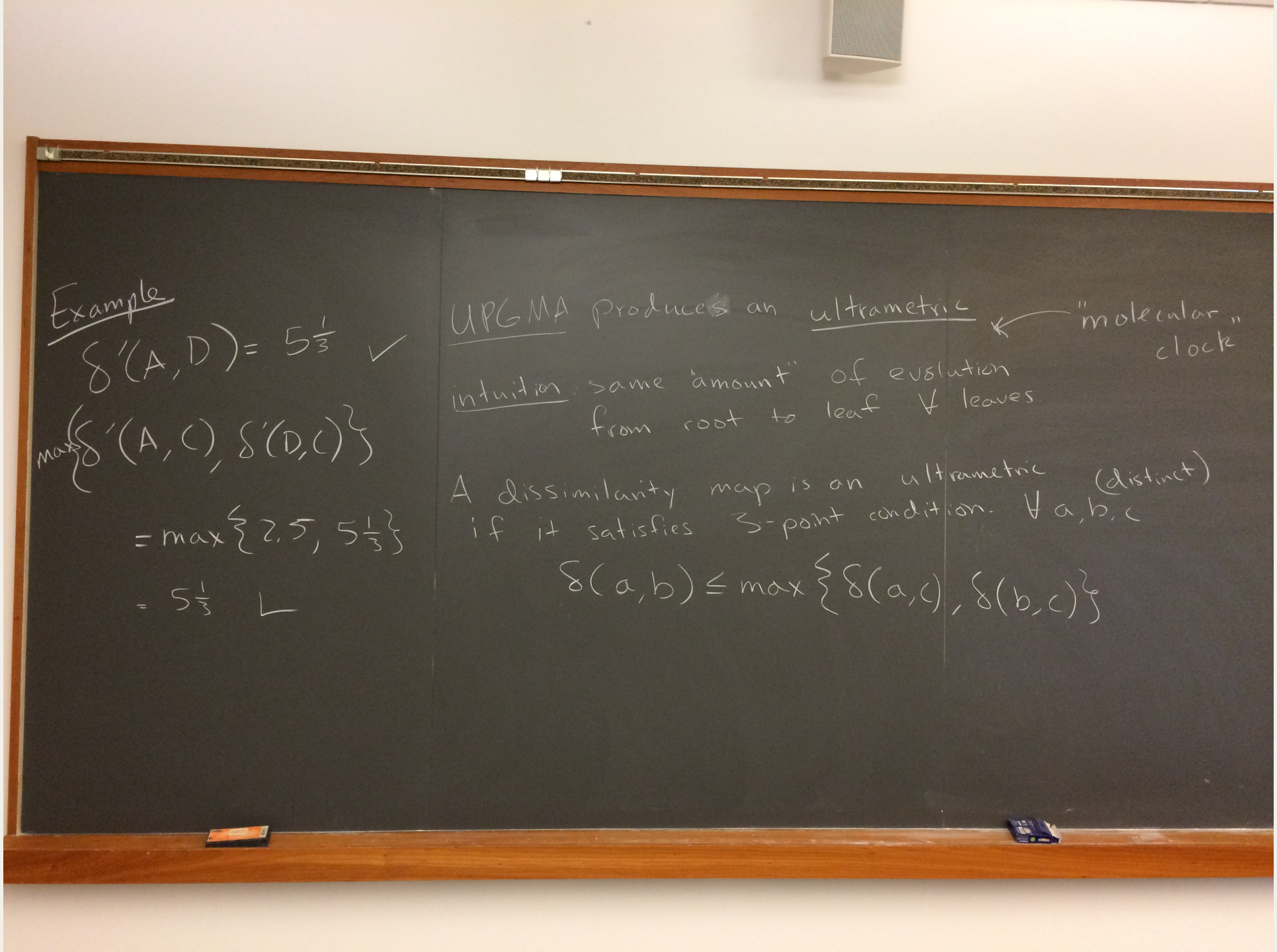
$$\delta(x, y) = \sum \text{all edge weights in the path from } x \text{ to } y.$$

UPGMA can produce unrealistic trees



Closest tree to input data

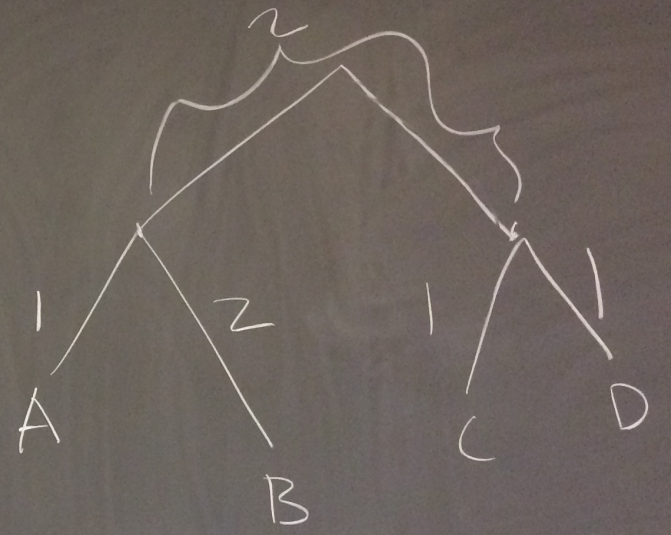
UPGMA tree

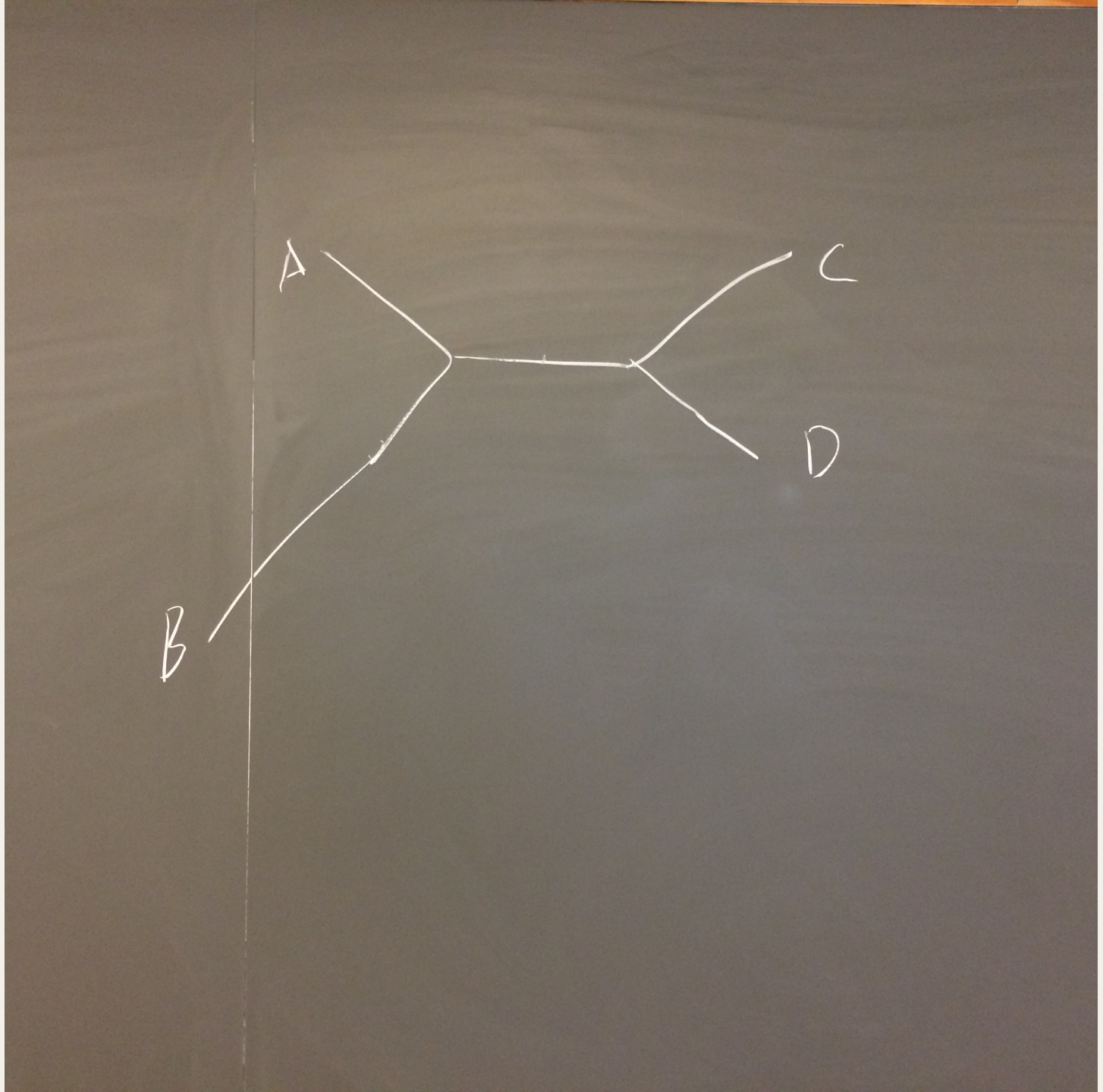

Example
$\delta'(A,D) = 5\frac{1}{3}$ ✓
$\max\{\delta'(A,C), \delta'(D,C)\}$
$= \max\{2,5, 5\frac{1}{3}\}$
$= 5\frac{1}{3}$ ✓
UPGMA produces an ultrametric
"molecular clock"
intuition: same "amount" of evolution from root to leaf ∀ leaves
A dissimilarity map is an ultrametric (distinct) if it satisfies 3-point condition. ∀ a, b, c
$\delta(a,b) \leq \max\{\delta(a,c), \delta(b,c)\}$

$\delta(B,C) \square \{\delta(A,B), \delta(A,C)\}$

$5 \neq \max\{3,4\}$

not ultrametric!




A
C
D
B