CS 364 COMPUTATIONAL BIOLOGY

Sara Mathieson
Haverford College

Outline

- Neighbor Joining algorithm
- Theory of Q-criteria and consistency of NJ
- Go over midterm 1

Neighbor Joining Algorithm

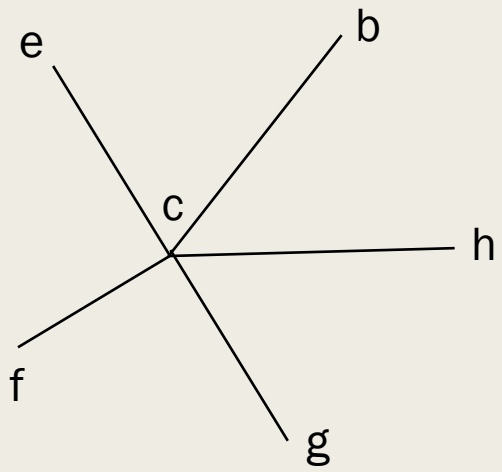
NJ initialization

<u>Input</u>

We are given a set of samples $\mathcal{X}$ and a dissimilarity map δ on $\mathcal{X}$.

<u>Initialization</u>

- Create a star tree with center vertex c and an edge (c, u) between c and all samples $u \in \mathcal{X}$.
- Let N_c be the set of neighbors of c and $n = |N_c|$ (cardinality of N_c). Set d equal to δ.



$N_c = \{b, e, f, g, h\}, \quad |N_c| = 5$

NJ Iterative step (part a)

(a) Find vertices f, g that minimize the Q-criteria. Note that UPGMA would only use the first term in this formula, $d(i, j)$. The remaining terms represent how far i and j are from the other vertices.

$$Q(i,j) = (n-2) \cdot d(i,j) - S_i - S_j, \quad \text{where}$$

$$S_i = \sum_{k \in N_c} d(i,k)$$

NJ Iterative step (part a)

(a) Find vertices f, g that minimize the Q-criteria. Note that UPGMA would only use the first term in this formula, $d(i,j)$. The remaining terms represent how far i and j are from the other vertices.

$$Q(i,j) = (n-2)\,\boxed{d(i,j)} - S_i - S_j, \quad \text{where}$$

$$S_i = \sum_{k \in N_c} d(i,k)$$

UPGMA

NJ Iterative step (part a)

(a) Find vertices f, g that minimize the Q-criteria. Note that UPGMA would only use the first term in this formula, $d(i,j)$. The remaining terms represent how far i and j are from the other vertices.

$$Q(i,j) = (n-2)\,\boxed{d(i,j)}\,\boxed{- S_i - S_j,} \quad \text{where}$$

$$S_i = \sum_{k \in N_c} d(i,k)$$

UPGMA

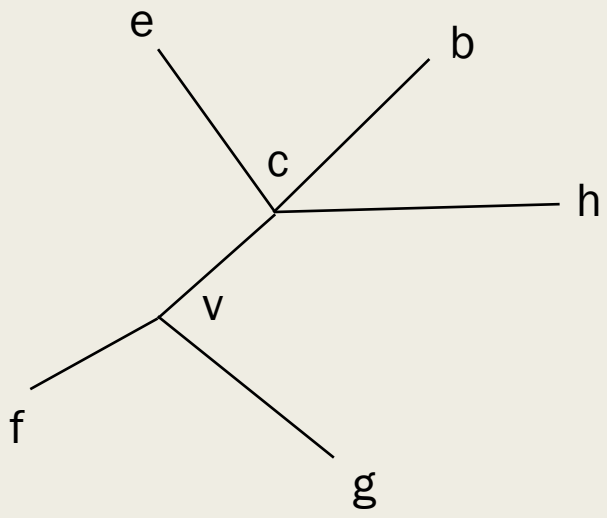
How far away *i* and *j* are from all the other vertices
(further away means we'll join them earlier)

NJ Iterative step (part b)

(b) Join f and g at internal vertex v. Now N_c contains v but not f and g. Compute the new edges weights:

$$d(f,v) = \frac{1}{2}d(f,g) + \frac{1}{2(n-2)}[S_f - S_g]$$

$$d(g,v) = \frac{1}{2}d(f,g) + \frac{1}{2(n-2)}[S_g - S_f]$$

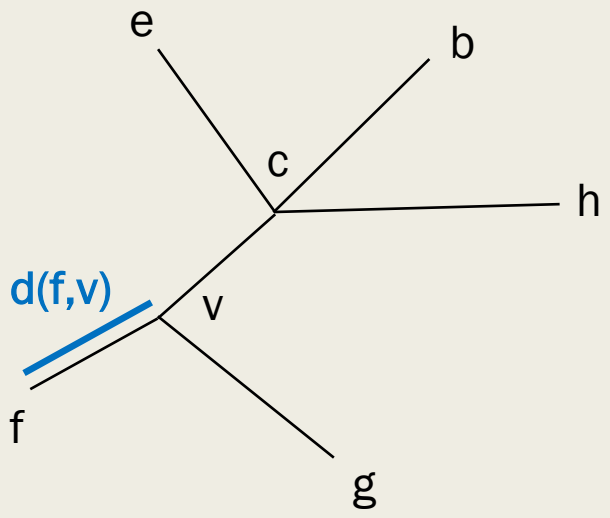


NJ Iterative step (part b)

(b) Join f and g at internal vertex v. Now N_c contains v but not f and g. Compute the new edges weights:

$$\boxed{d(f,v)} = \frac{1}{2}d(f,g) + \frac{1}{2(n-2)}[S_f - S_g]$$

$$d(g,v) = \frac{1}{2}d(f,g) + \frac{1}{2(n-2)}[S_g - S_f]$$

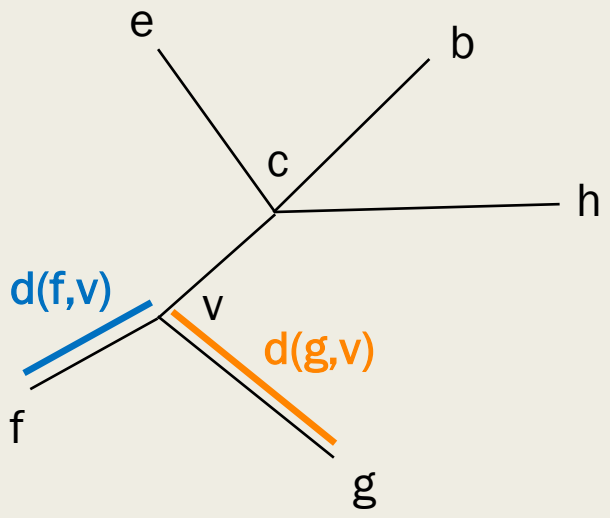


NJ Iterative step (part b)

(b) Join f and g at internal vertex v. Now N_c contains v but not f and g. Compute the new edges weights:

$$\boxed{d(f,v)} = \frac{1}{2}d(f,g) + \frac{1}{2(n-2)}[S_f - S_g]$$

$$\boxed{d(g,v)} = \frac{1}{2}d(f,g) + \frac{1}{2(n-2)}[S_g - S_f]$$



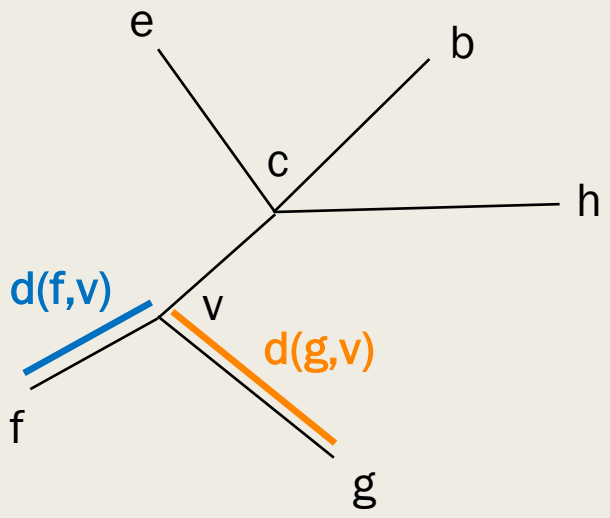
NJ Iterative step (part b)

UPGMA

(b) Join f and g at internal vertex v. Now N_c contains v but not f and g. Compute the new edges weights:

$$\boxed{d(f,v)} = \boxed{\frac{1}{2}d(f,g)} + \frac{1}{2(n-2)}[S_f - S_g]$$

$$\boxed{d(g,v)} = \boxed{\frac{1}{2}d(f,g)} + \frac{1}{2(n-2)}[S_g - S_f]$$



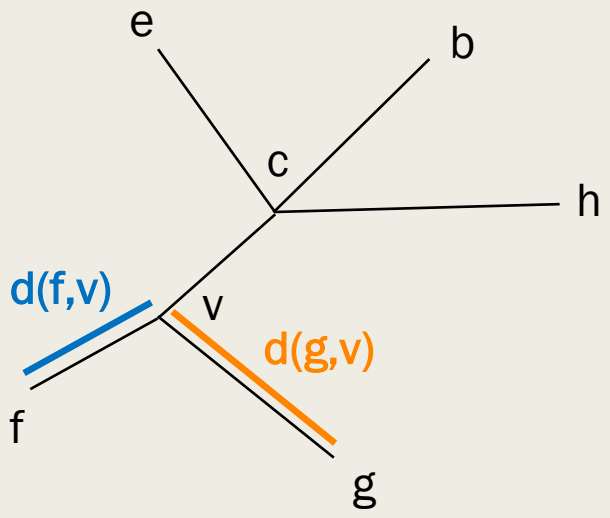
NJ Iterative step (part b)

UPGMA

(b) Join f and g at internal vertex v. Now N_c contains v but not f and g. Compute the new edges weights:

$$\boxed{d(f,v)} = \boxed{\frac{1}{2}d(f,g)} + \frac{1}{2(n-2)}\boxed{[S_f - S_g]}$$

$$\boxed{d(g,v)} = \boxed{\frac{1}{2}d(f,g)} + \frac{1}{2(n-2)}\boxed{[S_g - S_f]}$$



The *difference* between how far *f* and *g* are from other vertices. In this example *g* is on average further from other vertices, so d(g,v) > d(f,v)

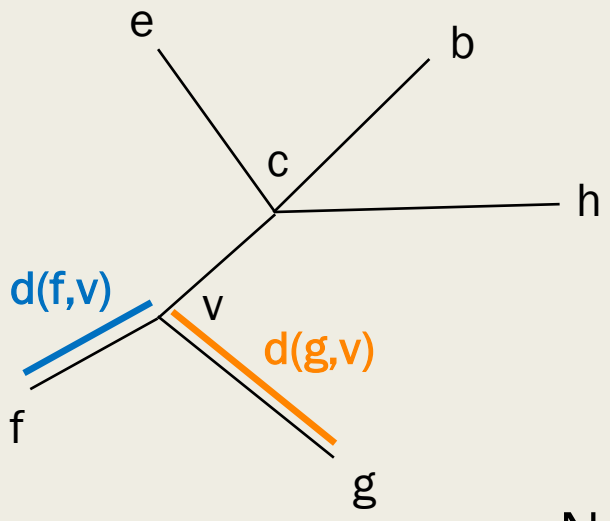
NJ Iterative step (part b)

UPGMA

(b) Join f and g at internal vertex v. Now N_c contains v but not f and g. Compute the new edges weights:

$$\boxed{d(f,v)} = \boxed{\frac{1}{2}d(f,g)} + \frac{1}{2(n-2)}\boxed{[S_f - S_g]}$$

$$\boxed{d(g,v)} = \boxed{\frac{1}{2}d(f,g)} + \frac{1}{2(n-2)}\boxed{[S_g - S_f]}$$



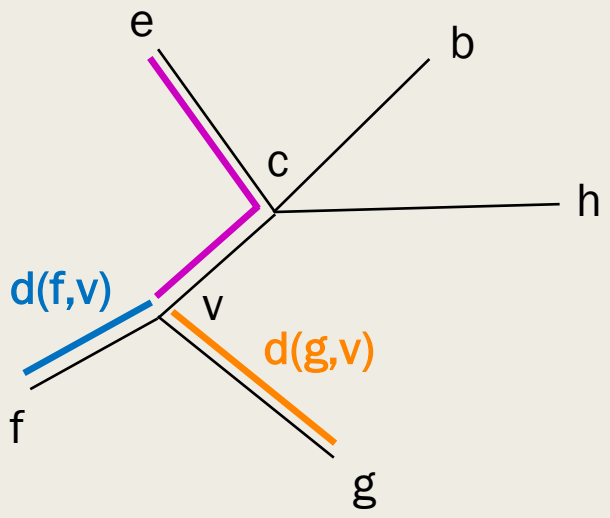
The *difference* between how far *f* and *g* are from other vertices. In this example *g* is on average further from other vertices, so d(g,v) > d(f,v)

$N_c = \{b, e, h, v\}, \quad |N_c| = 4$

NJ Iterative step (part c)

(c) Compute the distances from v to all remaining vertices $i \in N_c$:

$$\boxed{d(i,v)} = \frac{1}{2}[d(f,i) - d(f,v)] + \frac{1}{2}[d(g,i) - d(g,v)]$$



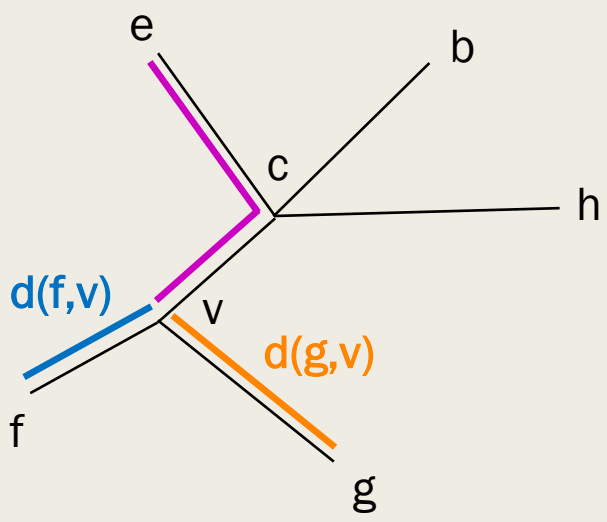
NJ Iterative step (part c)

(c) Compute the distances from v to all remaining vertices $i \in N_c$:

$$\boxed{d(i,v)} = \frac{1}{2}[d(f,i) - d(f,v)] + \frac{1}{2}[d(g,i) - d(g,v)]$$

Another way to write this:

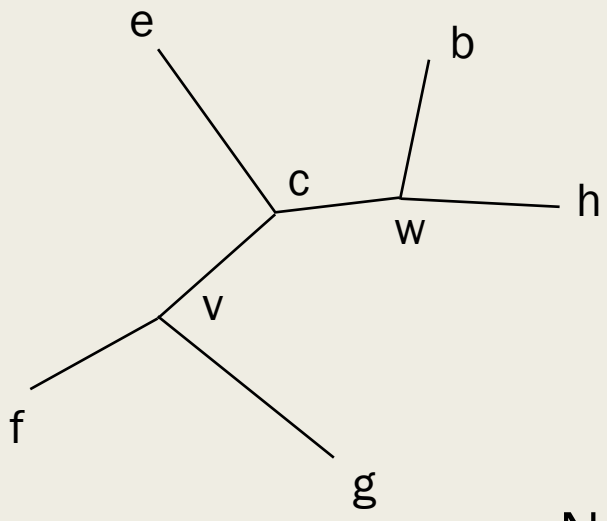
d(i,v) = ½[d(f,i) + d(g,i) – d(f,g)]



NJ Termination

<u>Termination</u>

When $n = 3$, the tree topology does not change since we have obtained a binary tree. We still need to run the last iteration though to determine the 3 remaining edge weights. The output is then the tree topology and all edge weights.

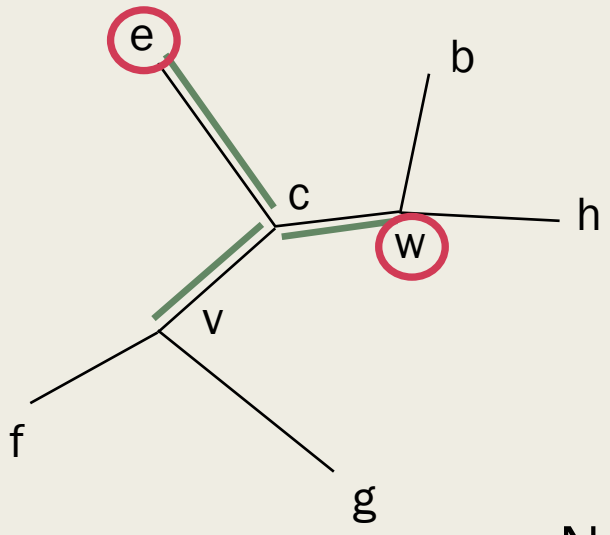


$N_c = \{e, v, w\}, \quad |N_c| = 3$

NJ Termination

<u>Termination</u>

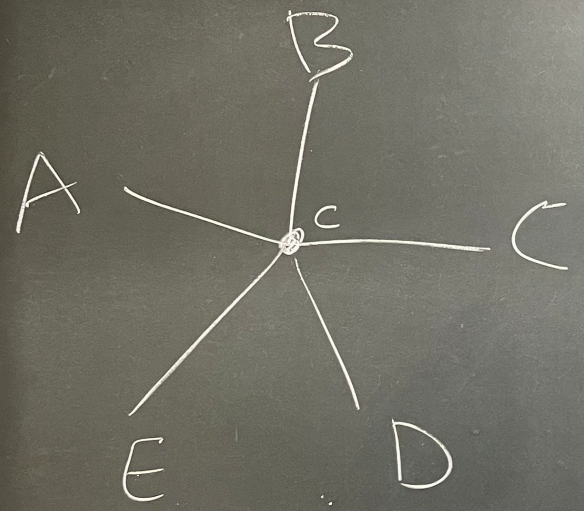
When $n = 3$, the tree topology does not change since we have obtained a binary tree. We still need to run the last iteration though to determine the 3 remaining edge weights. The output is then the tree topology and all edge weights.



We could "merge" *e* and *w* at *c*, then we would find d(e,c) and d(w,c) in step (b) and find d(v,c) in step (c)

$N_c = \{e, v, w\}, \quad |N_c| = 3$

Handout 10



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

$N_c = \{A, B, C, D, E\}$ $\quad n = |N_c| = 5$

① a) $S_c = 5 + 5 + 3 + 2 = 15$

$$Q(D,E) = (n-2)\,\delta(D,E) - S_D - S_E$$
$$= 3 \cdot 2 - 18 - 18$$
$$= \boxed{-30}$$

b) combine D & E

$$d(D,v) = \frac{1}{2} d(D,E) + \cancel{\frac{1}{2(n-2)}\left[\overset{18}{S_D} - \overset{18}{S_E}\right]}$$
$$= 1$$

$$d(E,v) = 1$$

c)

d	A	B	C	v
A	0	1	3	5
B		0	2	
C			0	
v				0

$$d(A,v) = \frac{1}{2}\left[d(A,E) - d(E,v)\right] + \frac{1}{2}\left[d(A,D) - d(D,v)\right]$$

$$= \frac{1}{2}(6-1) + \frac{1}{2}(6-1)$$

$$= 5$$

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

2 a) $S_v = 5+4+4 = 13$ $\qquad N_c = \{A, B, C, v\}$, $n = 4$

$Q(C, v) = (4-2)4 - 9 - 13$

$= 8 - 22$

$= -14$

b) $d(A, w) = \frac{1}{2} \cdot 1 + \frac{1}{2(4-2)} \overset{S_A - S_B}{(9-7)} = 1$

$d(B, w) = \frac{1}{2} \cdot 1 + \frac{1}{2 \cdot 2} \overset{S_B - S_A}{(7-9)} = 0$

c) $d(w, C) = \frac{1}{2}(3-1) + \frac{1}{2}(2-0) = 2$

$d(w, v) = \frac{1}{2}(5-1) + \frac{1}{2}(4-0) = 4$

(3) $N_c = \{w, c, v\}$, $n = 3$

i	w	c	v
s_i	6	6	8

Q	c	v
w	-10	-10
c	0	-10

d	w	c	v
w			4
c			4
v			

(b) $d(w, c) = \frac{1}{2} \cdot 2 + \frac{1}{2 \cdot 1}(6-6) = 1$

$d(c, c) = \frac{1}{2} \cdot 2 + \frac{1}{2+1}(6-6) = 1$

(c)

$d(c, v) = \frac{1}{2}(4-1) + \frac{1}{2}(4-1) = 3.$

little c

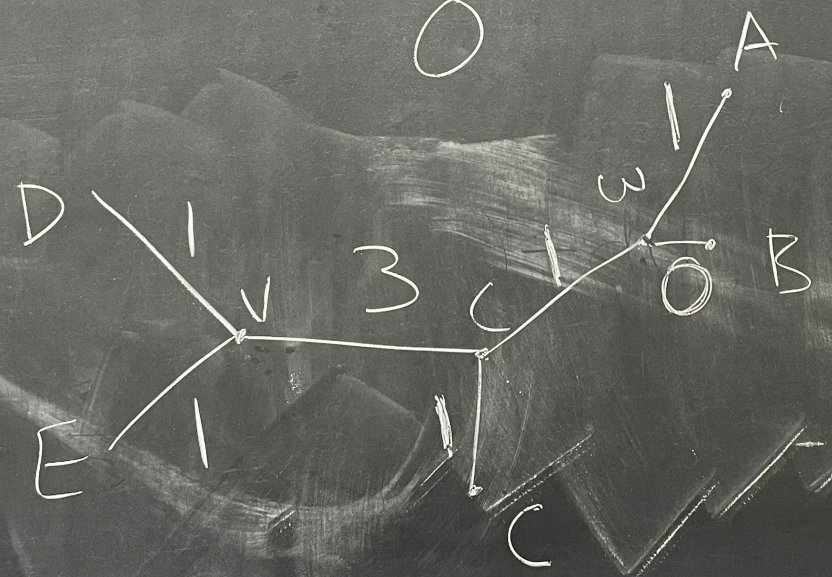
c)

d	A	B	C	v
A	0	1	3	5
B		0	2	4
C			0	4
v				0

$$d(B,v) = \frac{1}{2}(5-1) + \frac{1}{2}(5-1)$$
$$= 4$$

$$d(C,v) = 4$$

$$\delta'(A,E) = 6$$

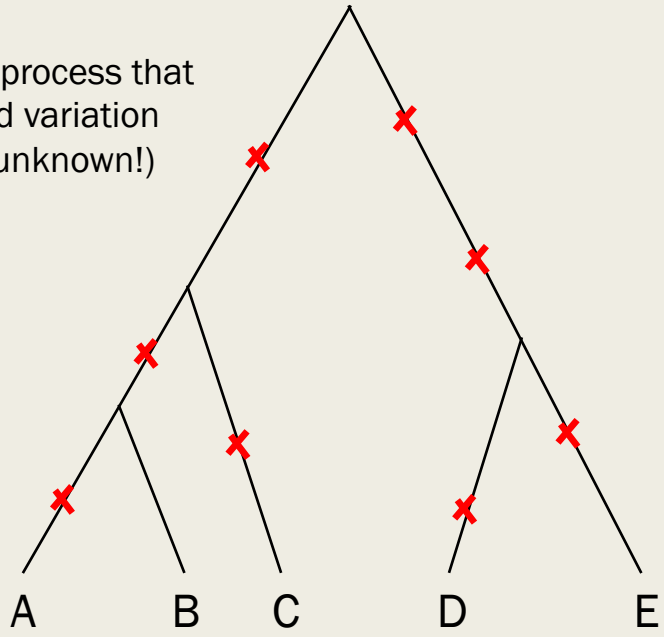


$$\boxed{\text{induced metric} \quad \delta' = \delta}$$

Q-criteria theory and consistency

What have we learned from our example?

Mutational process that generated variation (usually unknown!)

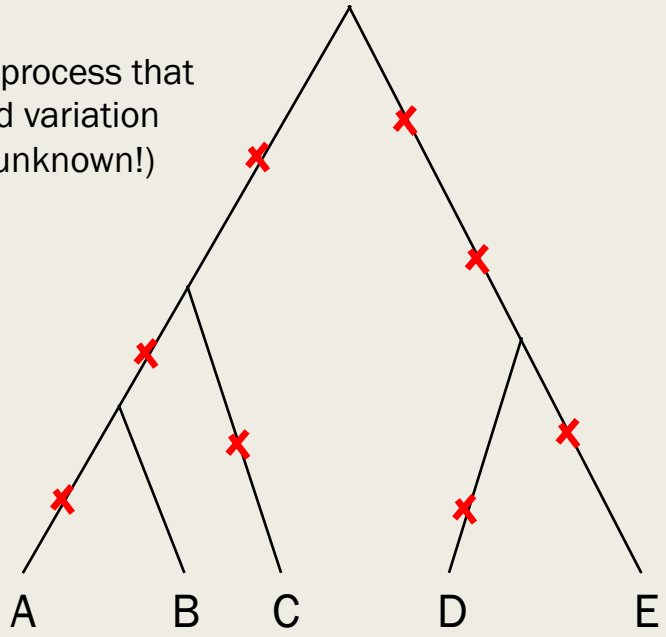


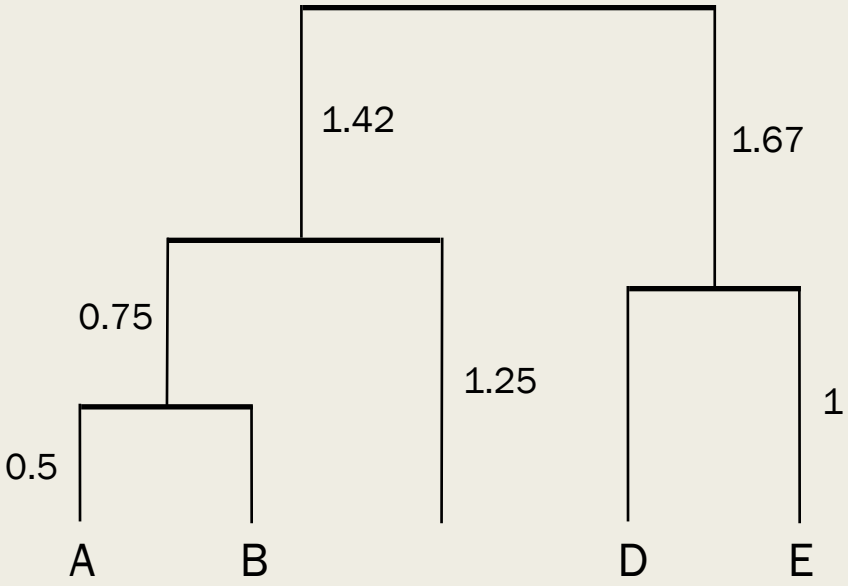
Original input dissimilarity map

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

What have we learned from our example?

Mutational process that generated variation (usually unknown!)





Tree produced by UPGMA (rooted)

Original input dissimilarity map

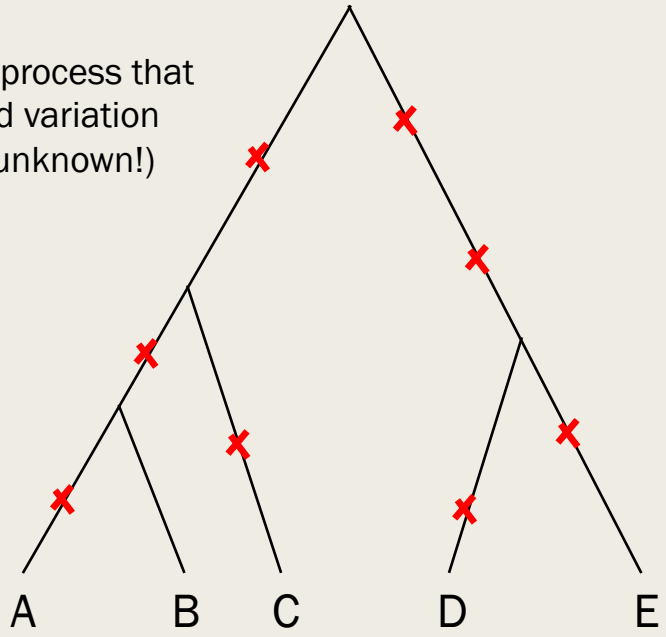
δ	A	B	C	D	E
A	0	1	3	6	6
B		0	2	5	5
C			0	5	5
D				0	2
E					0

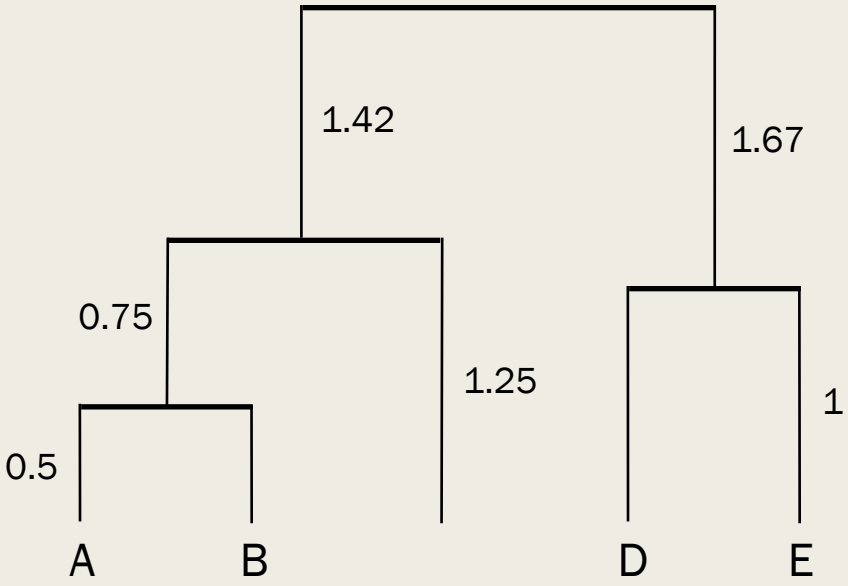
δ_{UPGMA}	A	B	C	D	E
A	0	1	2.5	5.33	5.33
B		0	2.5	5.33	5.33
C			0	5.33	5.33
D				0	2
E					0

Tree metric on *X* induced by UPGMA

What have we learned from our example?

Mutational process that generated variation (usually unknown!)





Tree produced by UPGMA (rooted)

Original input dissimilarity map

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

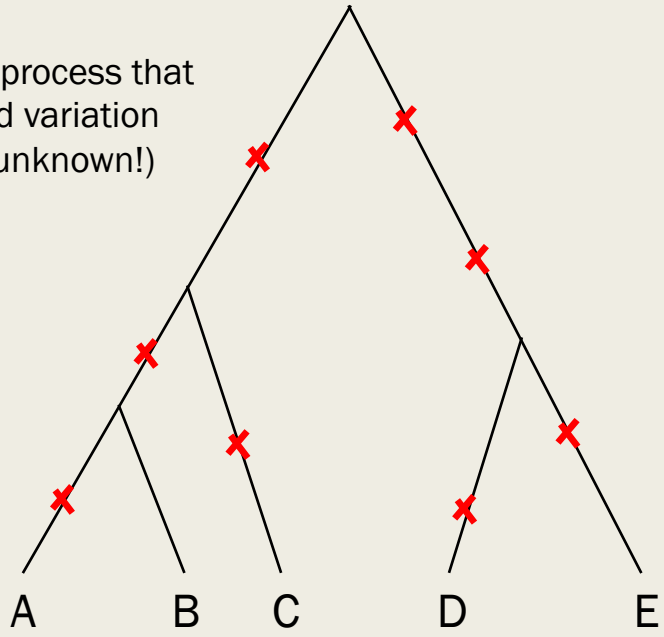
$\neq$

δ_{UPGMA}	A	B	C	D	E
A	0	1	2.5	5.33	5.33
B		0	2.5	5.33	5.33
C			0	5.33	5.33
D				0	2
E					0

Tree metric on *X* induced by UPGMA

What have we learned from our example?

Mutational process that generated variation (usually unknown!)

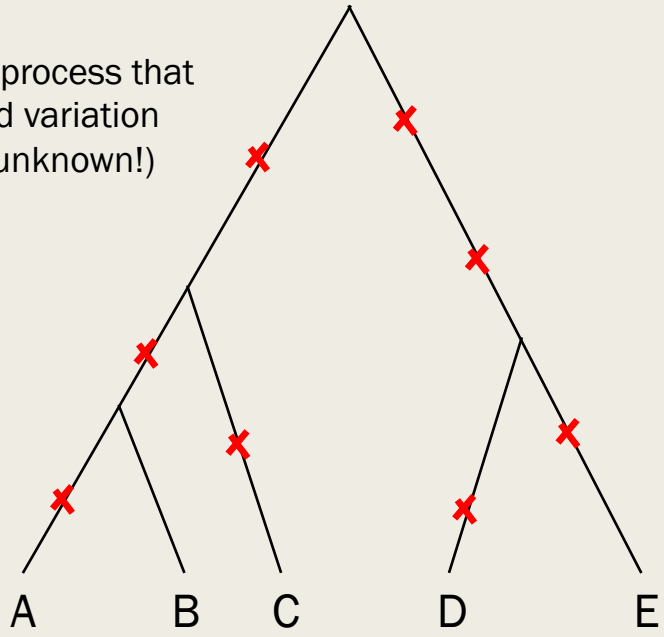


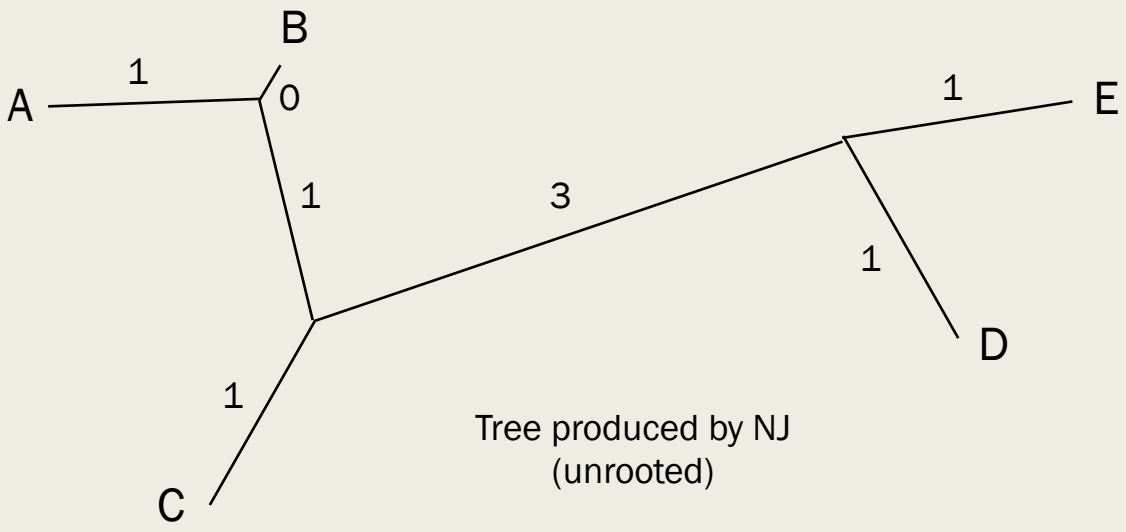
Original input dissimilarity map

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

What have we learned from our example?

Mutational process that generated variation (usually unknown!)





Tree produced by NJ (unrooted)

Original input dissimilarity map

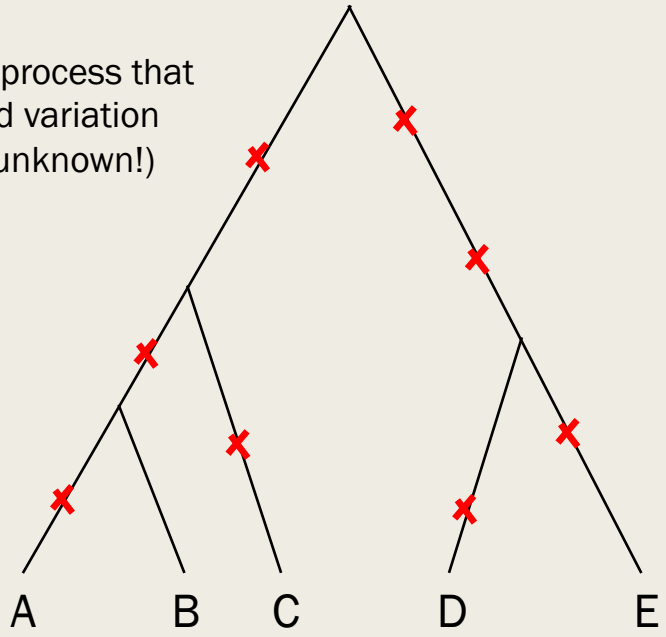
δ	A	B	C	D	E
A	0	1	3	6	6
B		0	2	5	5
C			0	5	5
D				0	2
E					0

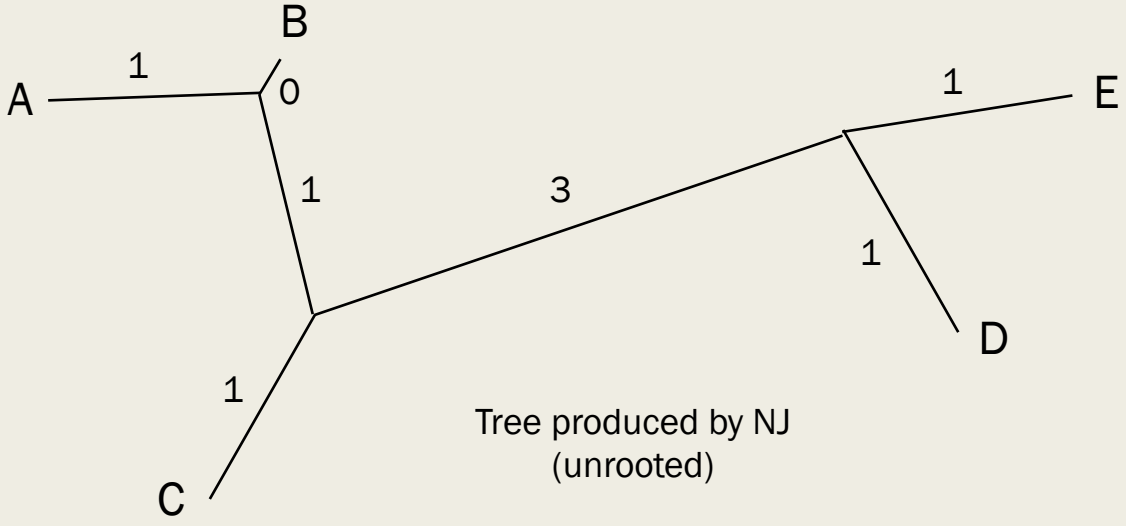
δ_{NJ}	A	B	C	D	E
A	0	1	3	6	6
B		0	2	5	5
C			0	5	5
D				0	2
E					0

Tree metric on *X* induced by NJ

What have we learned from our example?

Mutational process that generated variation (usually unknown!)





Tree produced by NJ (unrooted)

Original input dissimilarity map

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_{NJ}	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

Tree metric on *X* induced by NJ

Neighbor-Joining is consistent

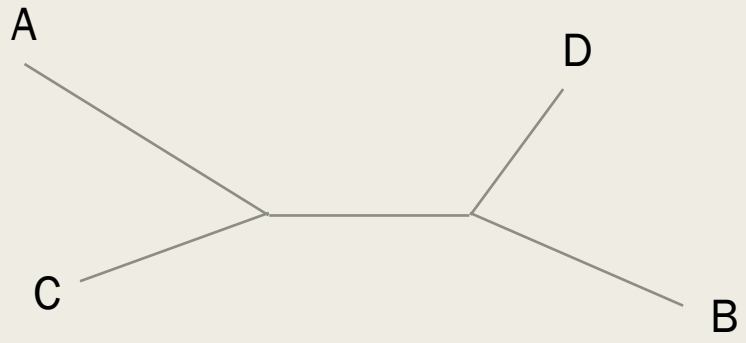
- If the original dissimilarity map is a tree metric, NJ will produce an induced tree metric equal to the original (*consistency*)

- UPGMA is not always consistent

- If the original dissimilarity map is not a tree metric (almost always the case), NJ will get closer, but both UPGMA and NJ are heuristics and not guaranteed to produce the edge-weighted tree that induces the very closest map to the original input (NP-complete)

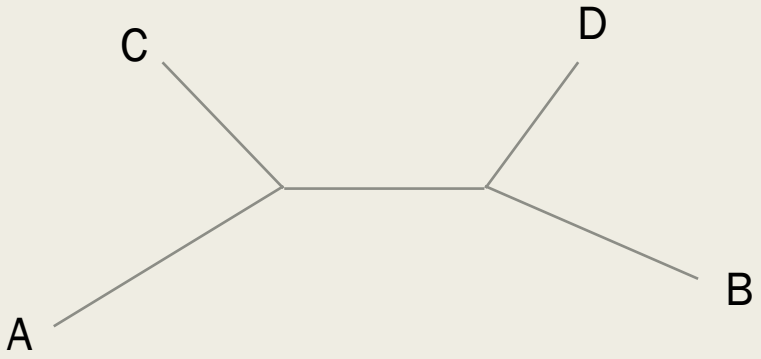
Handout 13

Handout 13 example



d(A,D)+d(D,B)+d(B,C)+d(C,A) = 12+5+9+7 = 33

δ	A	B	C	D
A	0	10	7	12
B		0	9	5
C			0	10
D				0

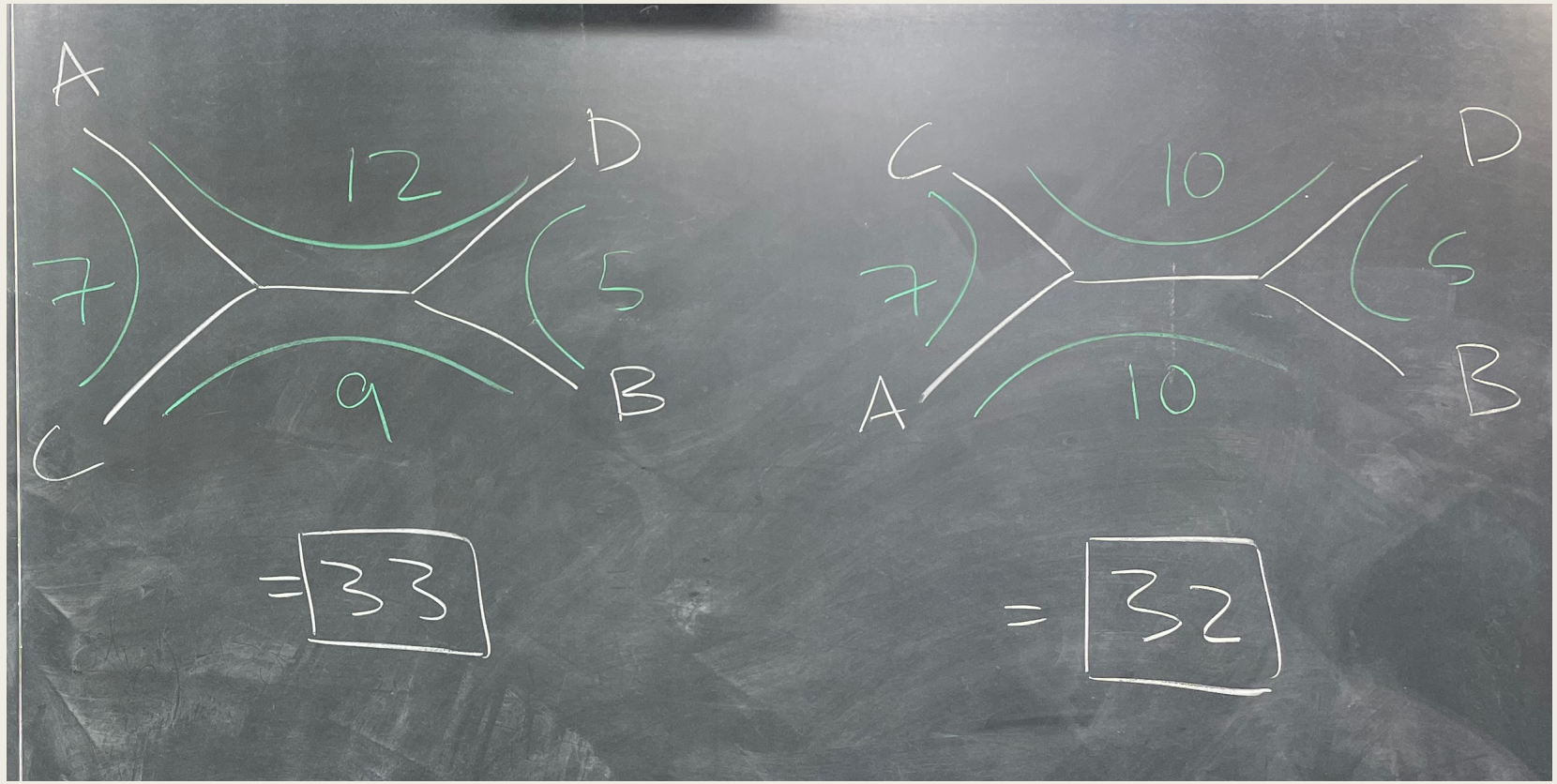


d(A,C)+d(C,D)+d(D,B)+d(B,A) = 7+10+5+10 = 32

Different ways of “walking” around the entire tree produce different lengths

Q-criteria seeks to choose the neighbors that would minimize the average tree length the most

Handout 13 example

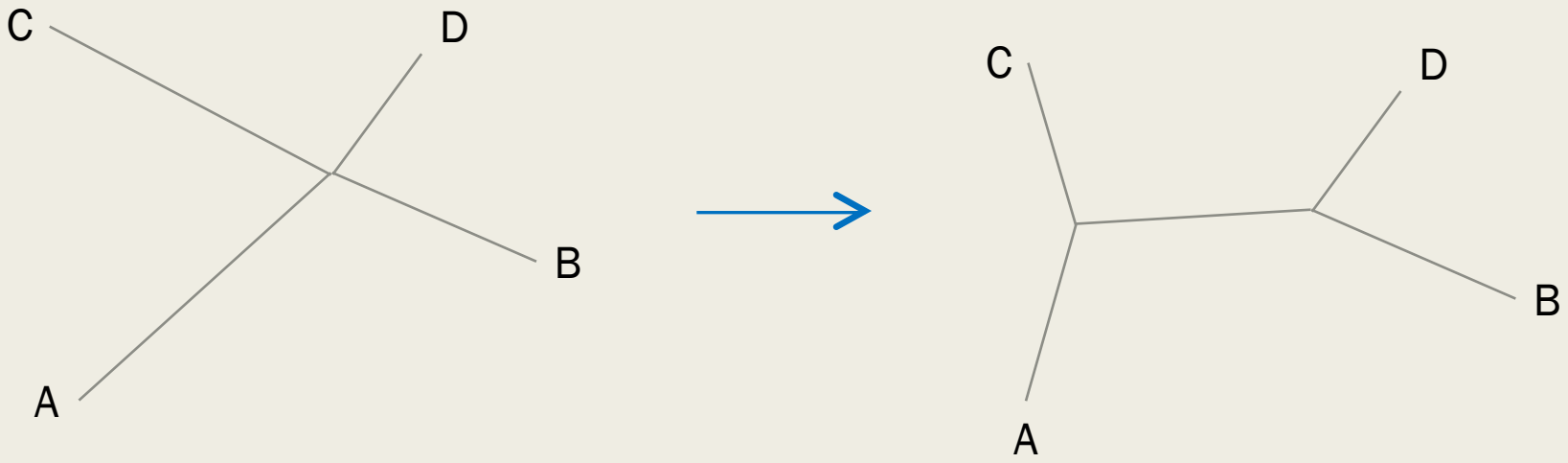


Q-criteria intuition

- Goal: we want the smallest tree that adequately explains the observed patterns of evolution (called BME: Balanced Minimum Evolution)

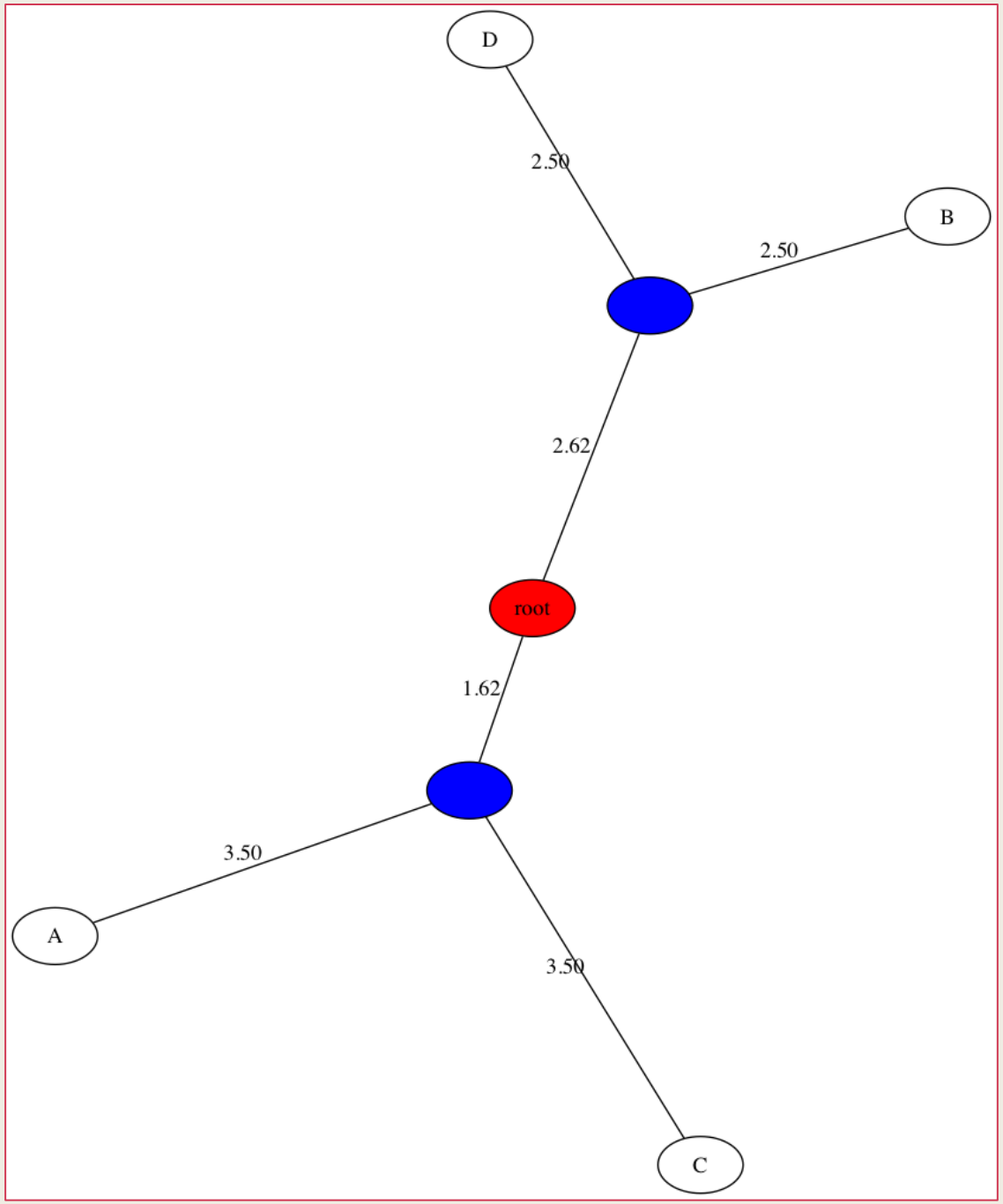
- Q-criteria minimizes the “whole tree length”, which is the average of all the different ways we could walk around the tree

- The idea is that we want to merge nodes that are far away, so we don’t have to “walk” to each of them separately, we can use the path to their merged vertex



UPGMA on Handout 13 example

δ	A	B	C	D
A	0	10	7	12
B		0	9	5
C			0	10
D				0

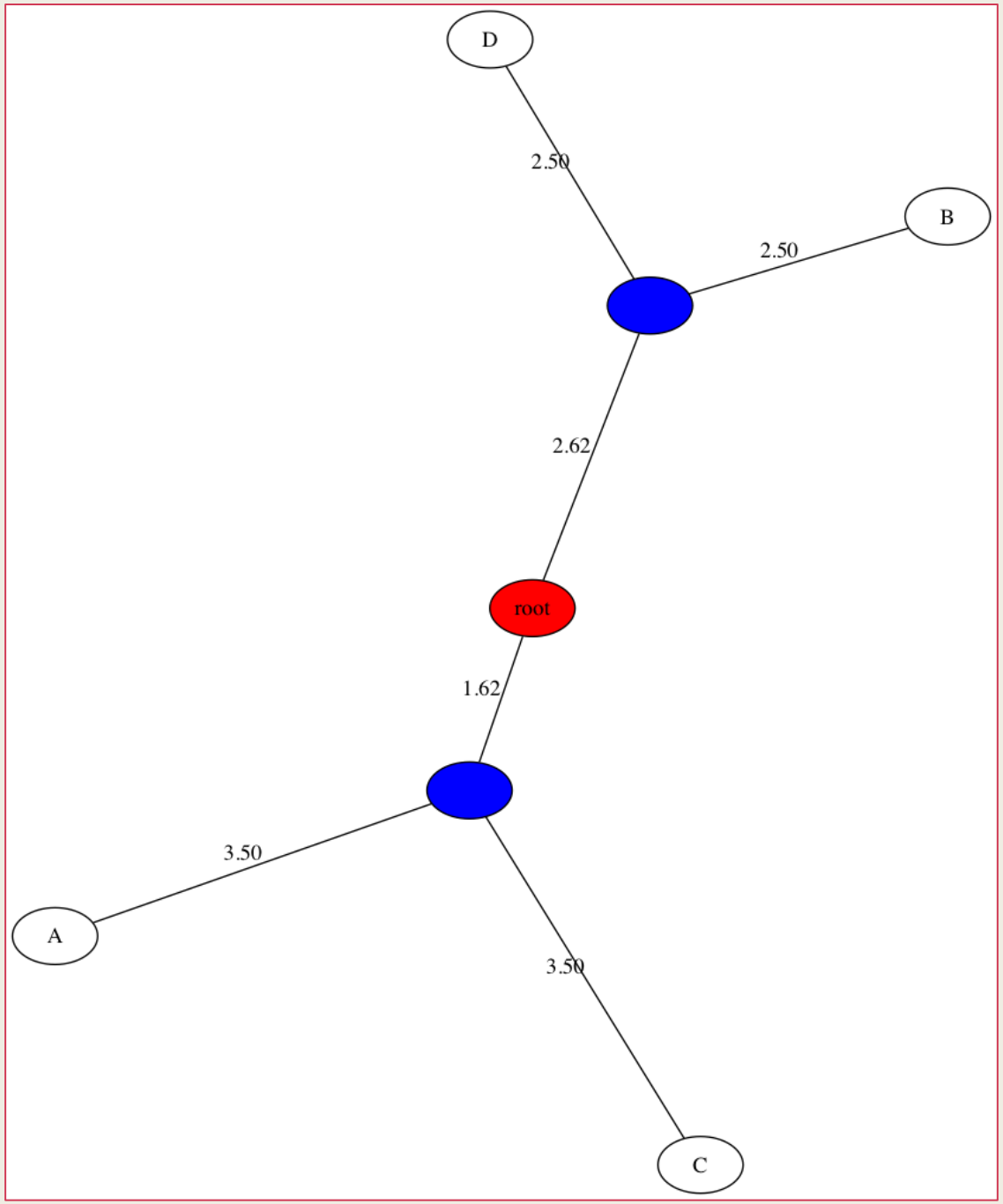


UPGMA on Handout 13 example

δ	A	B	C	D
A	0	10	7	12
B		0	9	5
C			0	10
D				0

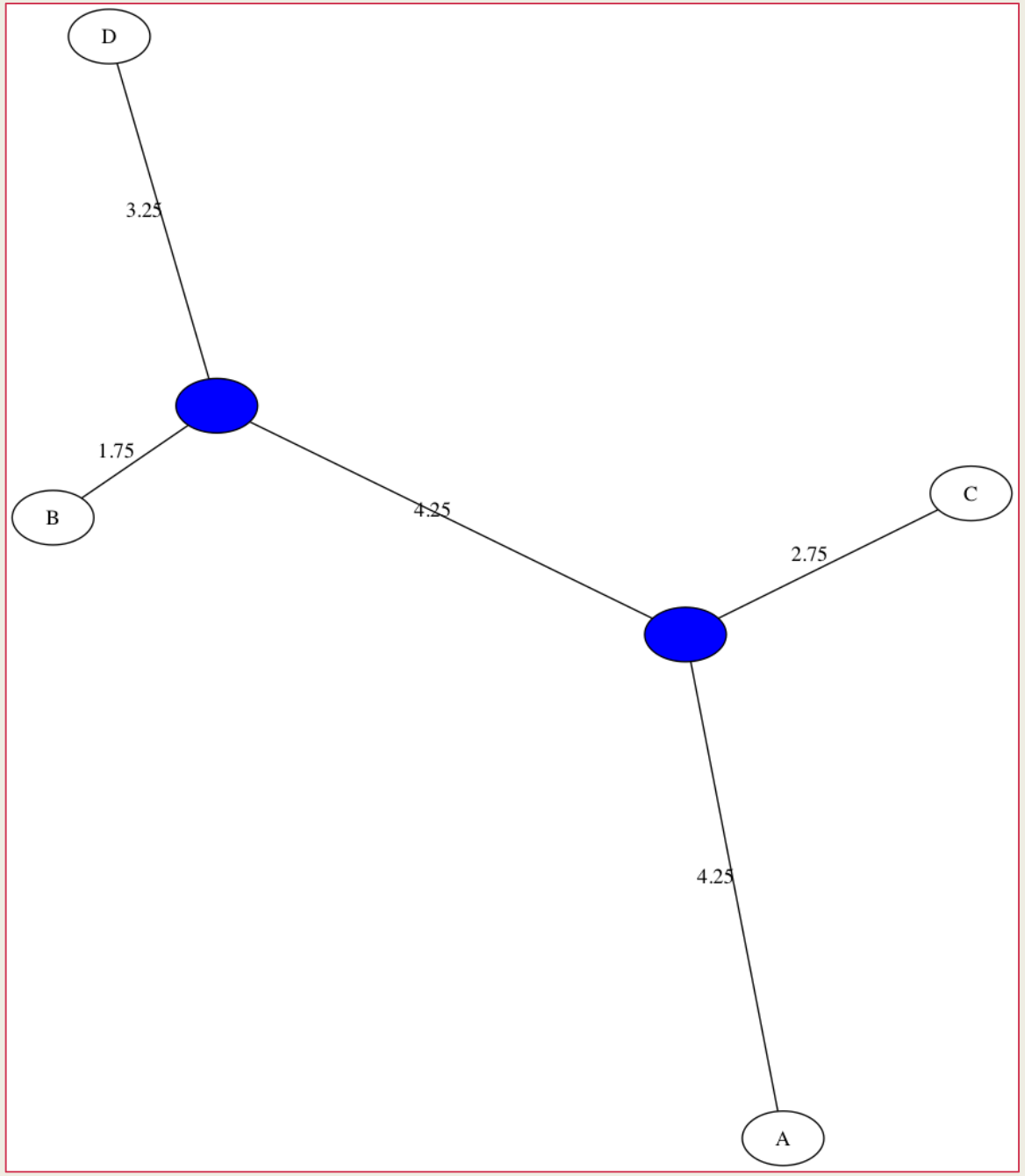
10.25 = (10+12+9+10)/4 (unweighted average)

δ_{UPGMA}	A	B	C	D
A	0	10.25	7	10.25
B		0	10.25	5
C			0	10.25
D				0



NJ on Handout 13 example

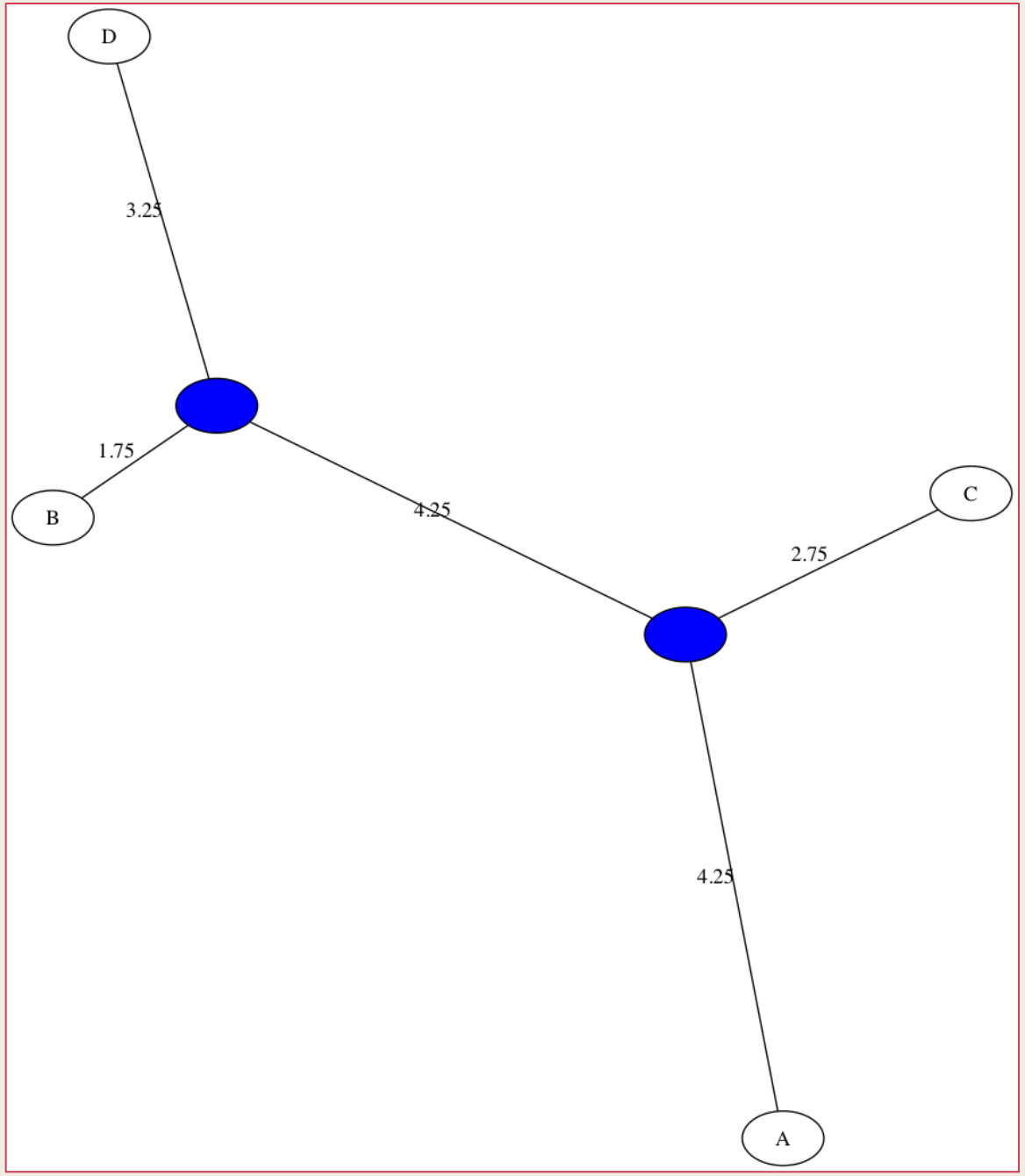
δ	A	B	C	D
A	0	10	7	12
B		0	9	5
C			0	10
D				0



NJ on Handout 13 example

δ	A	B	C	D
A	0	10	7	12
B		0	9	5
C			0	10
D				0

δ_{NJ}	A	B	C	D
A	0	10.25	7	11.75
B		0	8.75	5
C			0	10.25
D				0

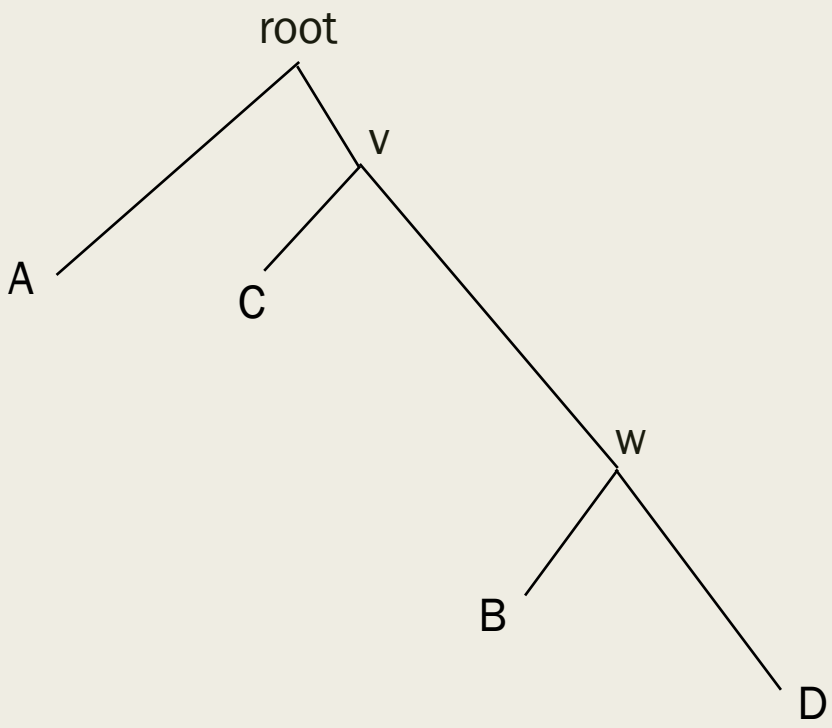


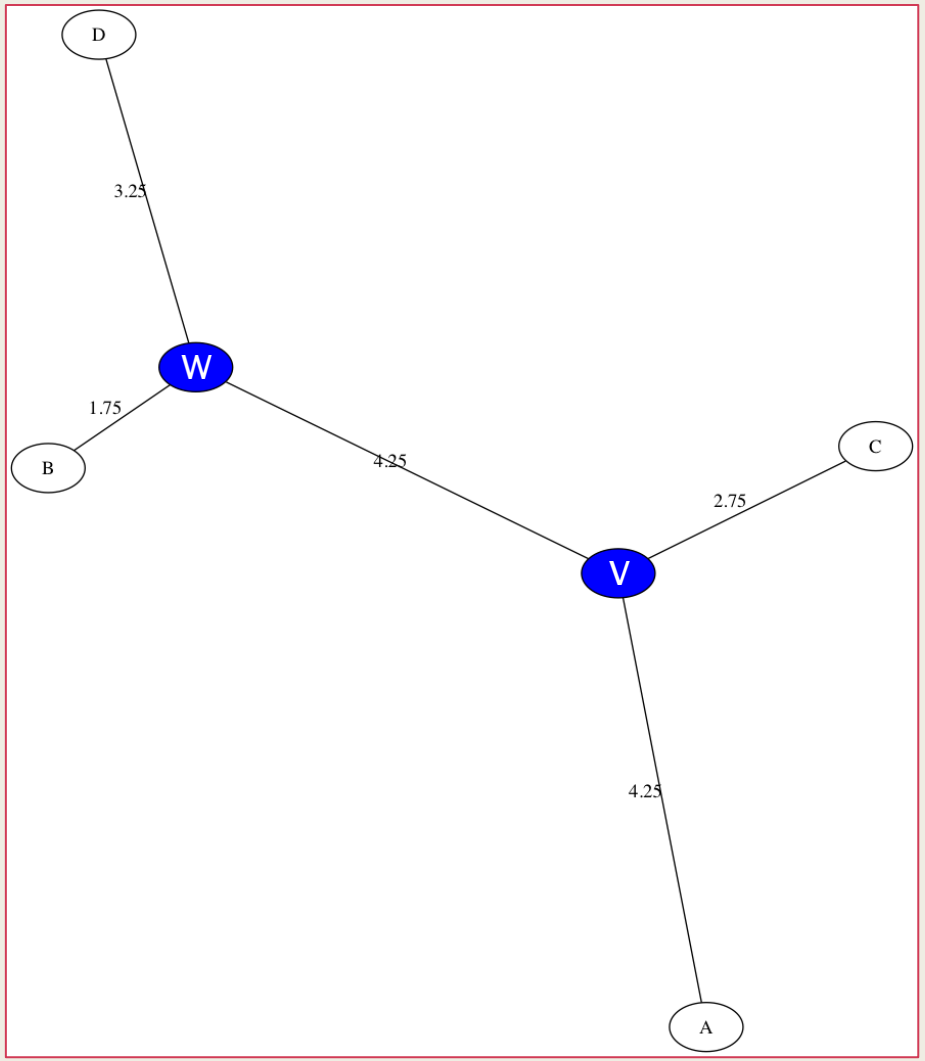
How to root a NJ tree?

- Method 1: use an ***outgroup***
- An outgroup is a species or sample that is more distantly related to all the other samples (“ingroup”) than any pair of ingroup samples

How to root a NJ tree?

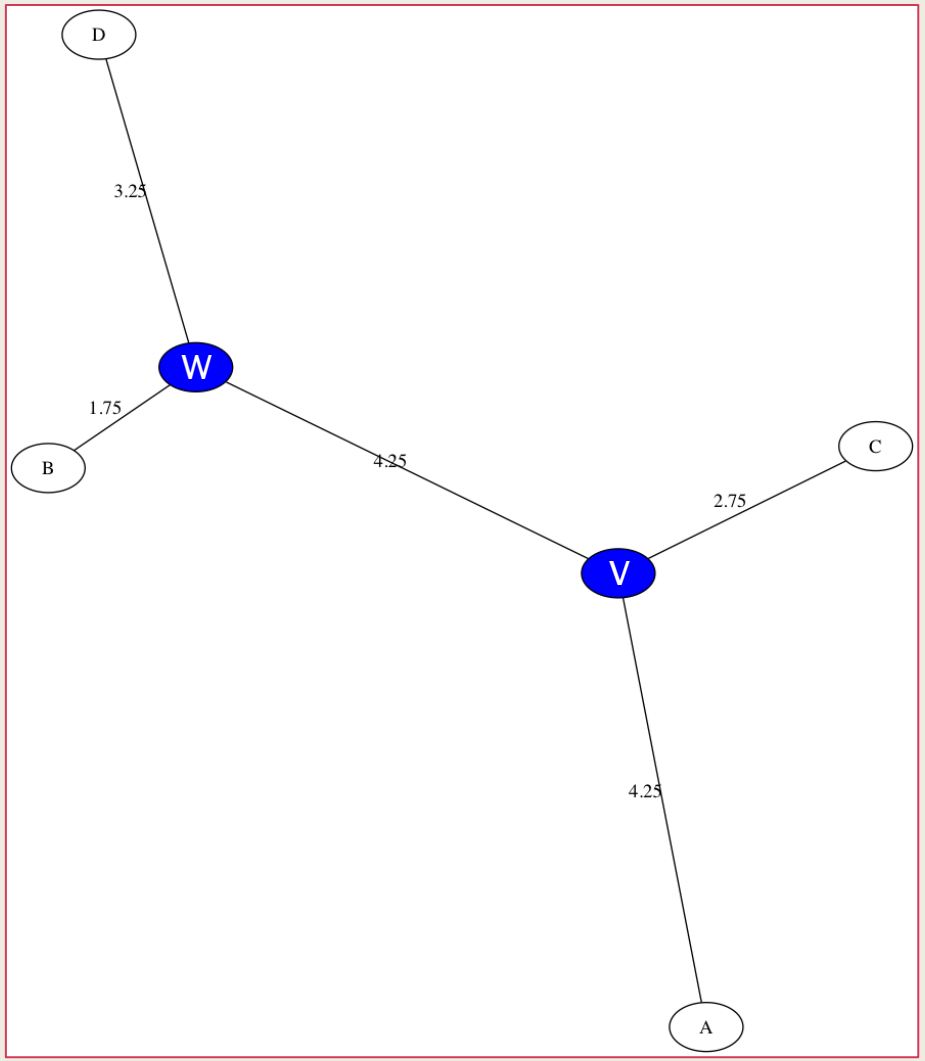
- Method 1: use an ***outgroup***
- An outgroup is a species or sample that is more distantly related to all the other samples ("ingroup") than any pair of ingroup samples
- For example, if we knew that A is an outgroup to ingroup {B,C,D}, we could root the NJ like this:





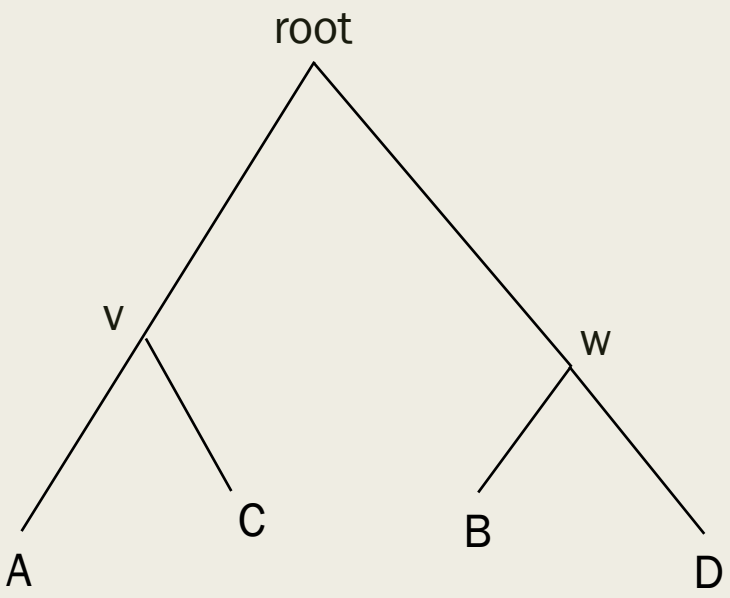
How to root a NJ tree?

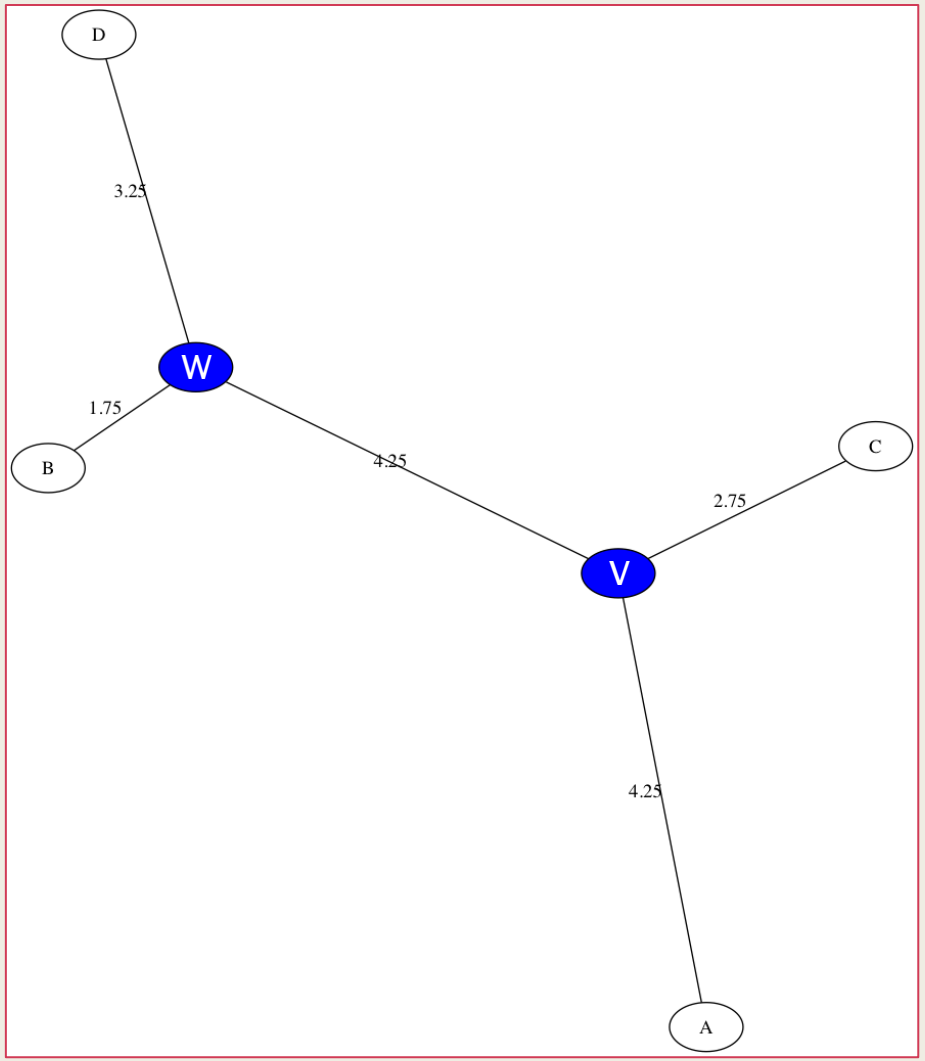
- Method 2: divide the longest path between leaves by 2
- *Assumption: molecular clock more or less valid*



How to root a NJ tree?

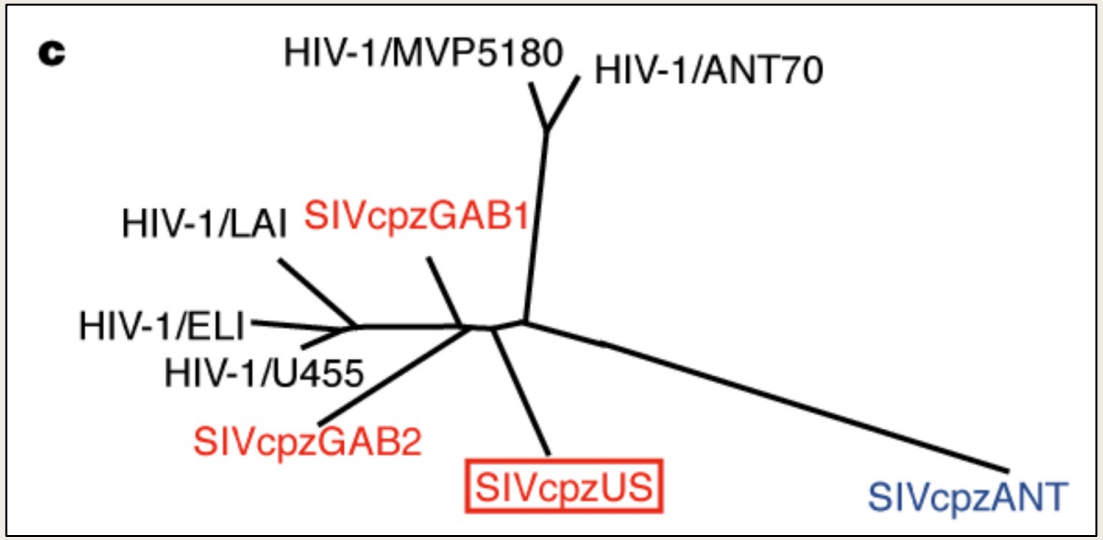
- Method 2: divide the longest path between leaves by 2
- *Assumption: molecular clock more or less valid*
- Longest path:
- A -> v -> w -> D = 4.25+4.25+3.25 = 11.75

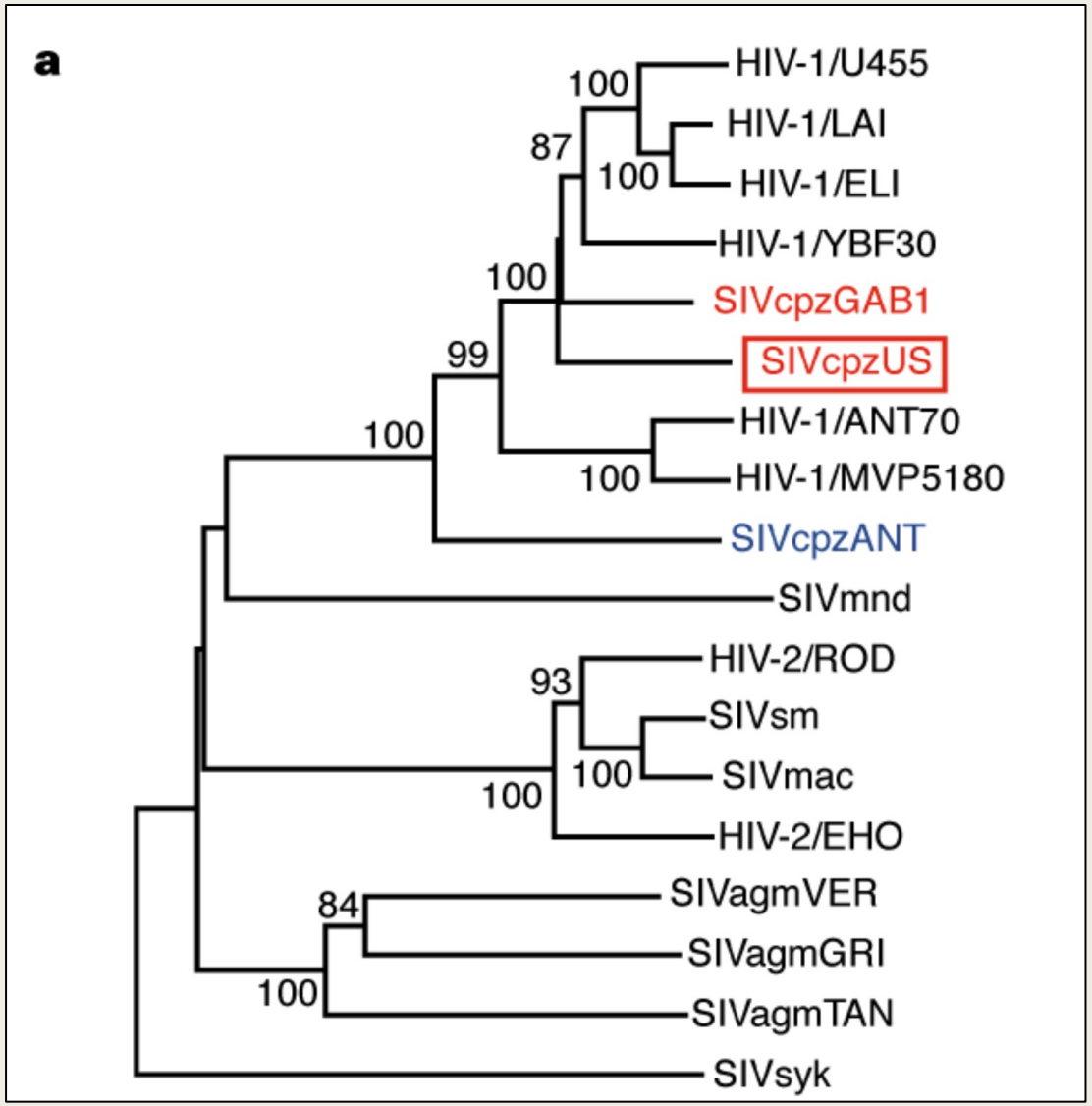




Example of NJ in research

Neighbor Joining trees (unrooted and rooted) for different strains of HIV



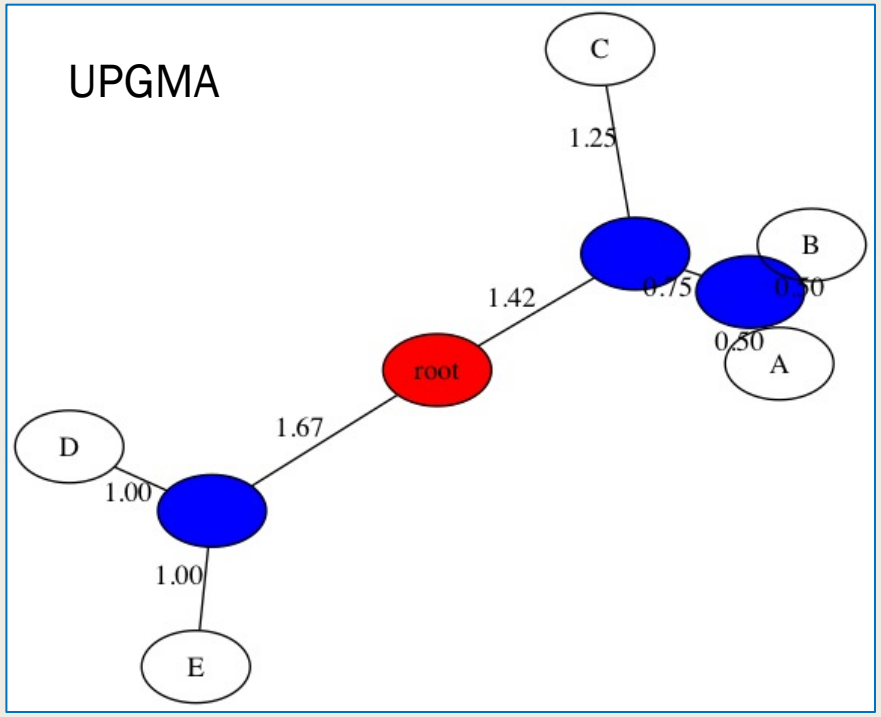


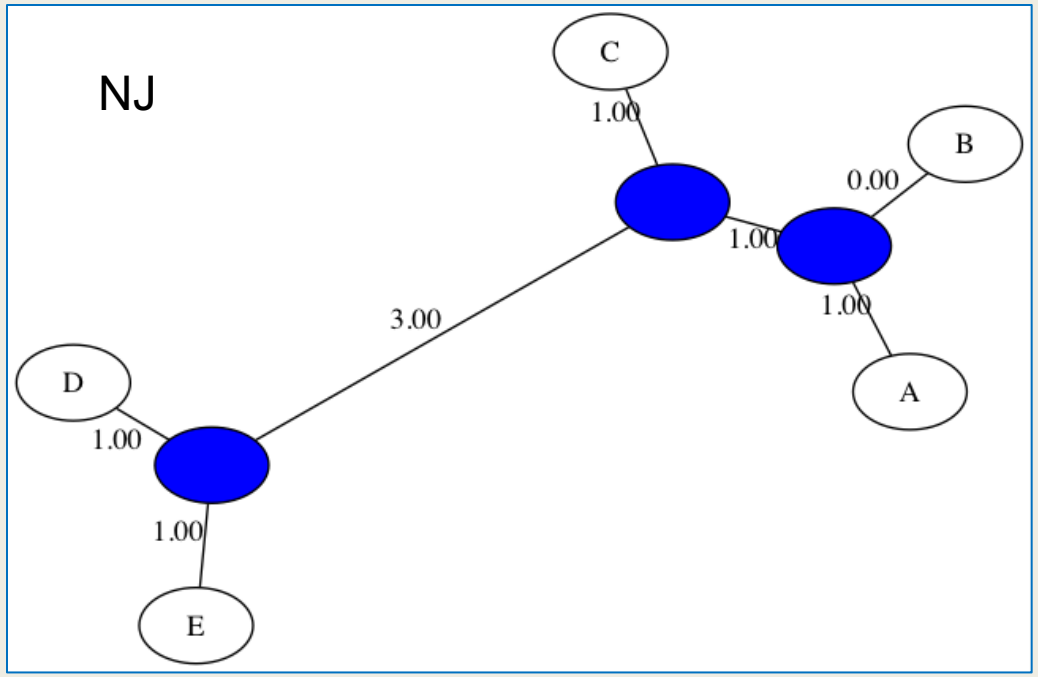
Origin of HIV-1 in the chimpanzee *Pan troglodytes troglodytes* (Nature, 2009)

Lab 5 introduction

Lab 5

- Goals: implement both UPGMA and NJ
- Analyze the trees produced by each one





Example output images

Lab 5

- Using pygraphviz

```
import pygraphviz as gv
tree = gv.AGraph() # constructs a graph object
tree.add_node("A") # the string is both the label and hash key
tree.add_node("B")
tree.add_edge("A", "B", label="1.0", len=1.0) # set string label as length
tree.draw("my_tree.png",prog="neato") # neato does node/edge layout
```

- Extra credit opportunities (rare!)
  - *Figure out how to layout UPGMA trees so the root is at the top and leaves at the bottom*
  - *Analyze the induced metrics produced by UPGMA and NJ*

Midterm 1

(not posted online)