CS 68: BIOINFORMATICS

Prof. Sara Mathieson
Swarthmore College
Spring 2018

Outline: Feb 28

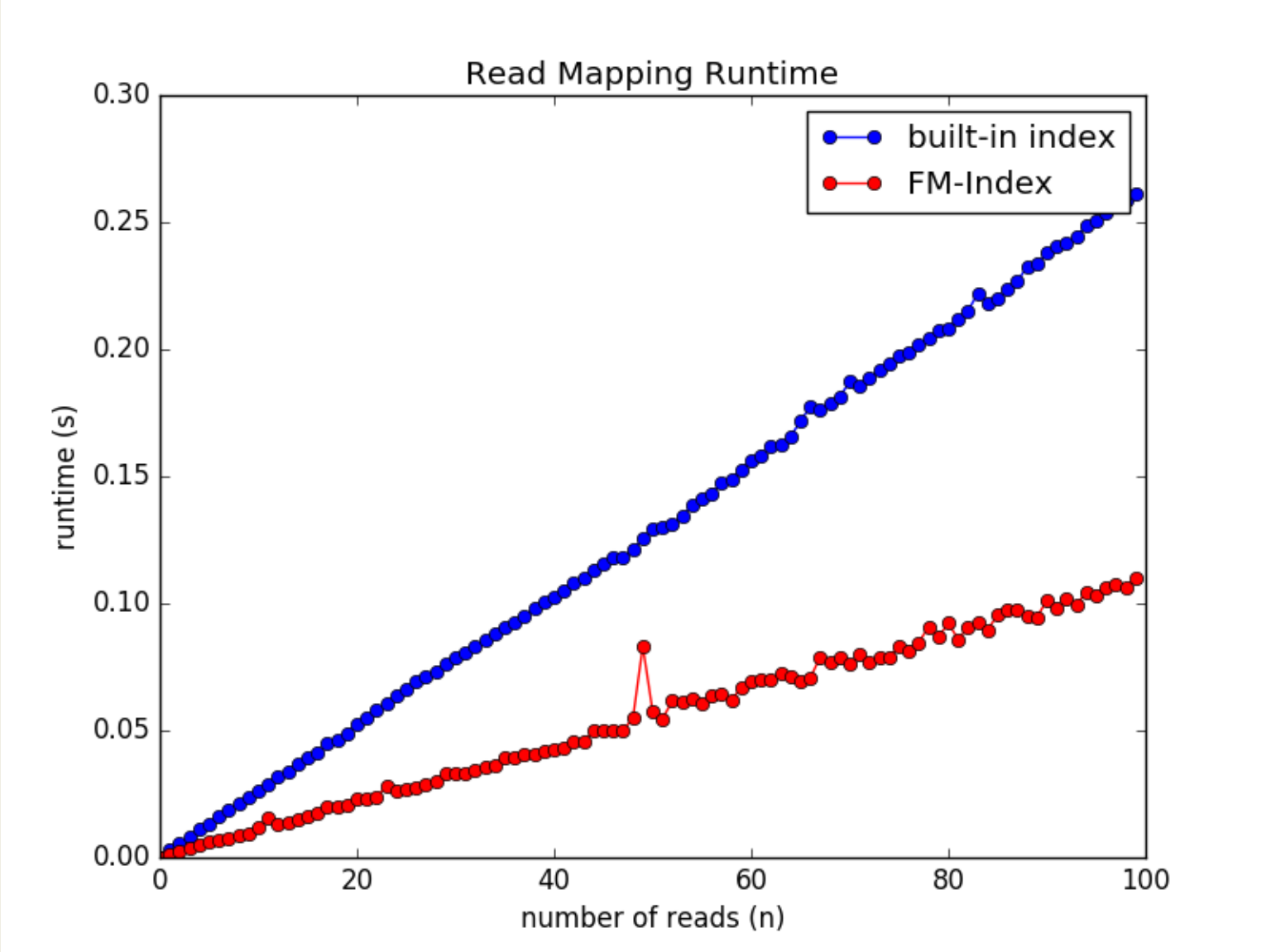
- Continue Neighbor-Joining (NJ)
- Theory of the Q-criteria
- Consistency of NJ

Notes:

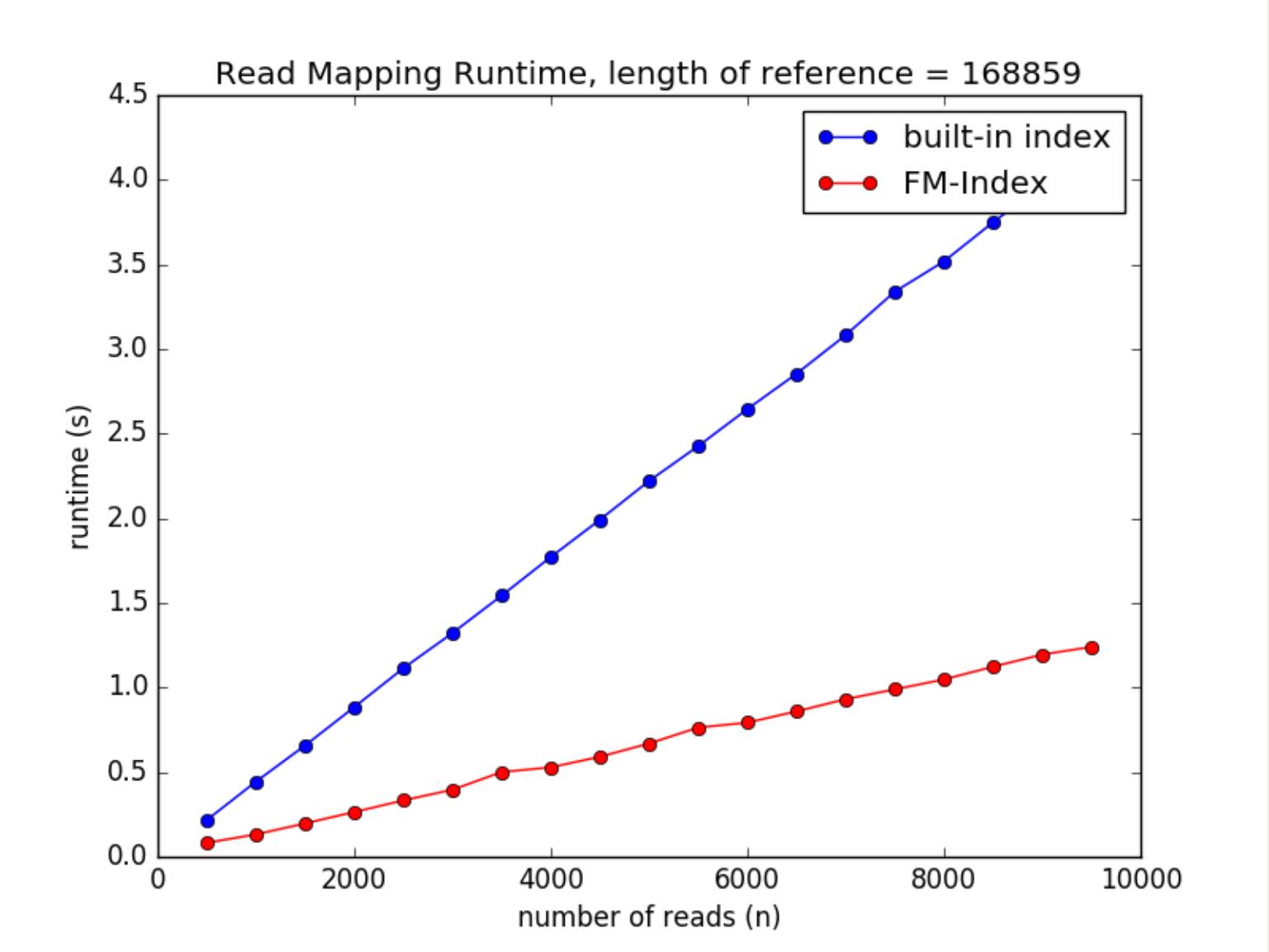
- Office hours TODAY 1-3pm
- Create “cheat-sheet” for midterm
- Choose partners for Lab 5

Lab 4 Runtime plot examples

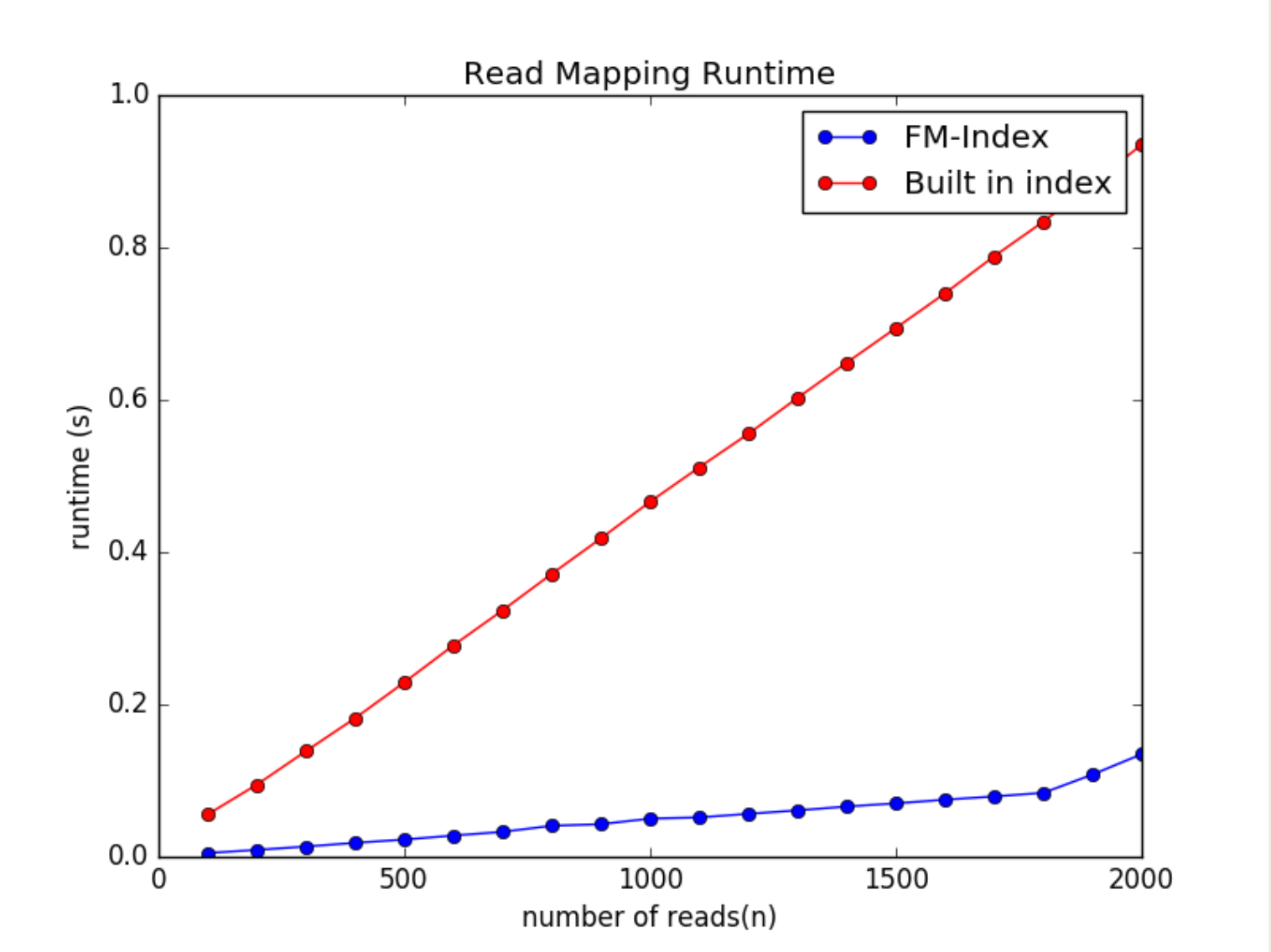
Hannah and Melissa



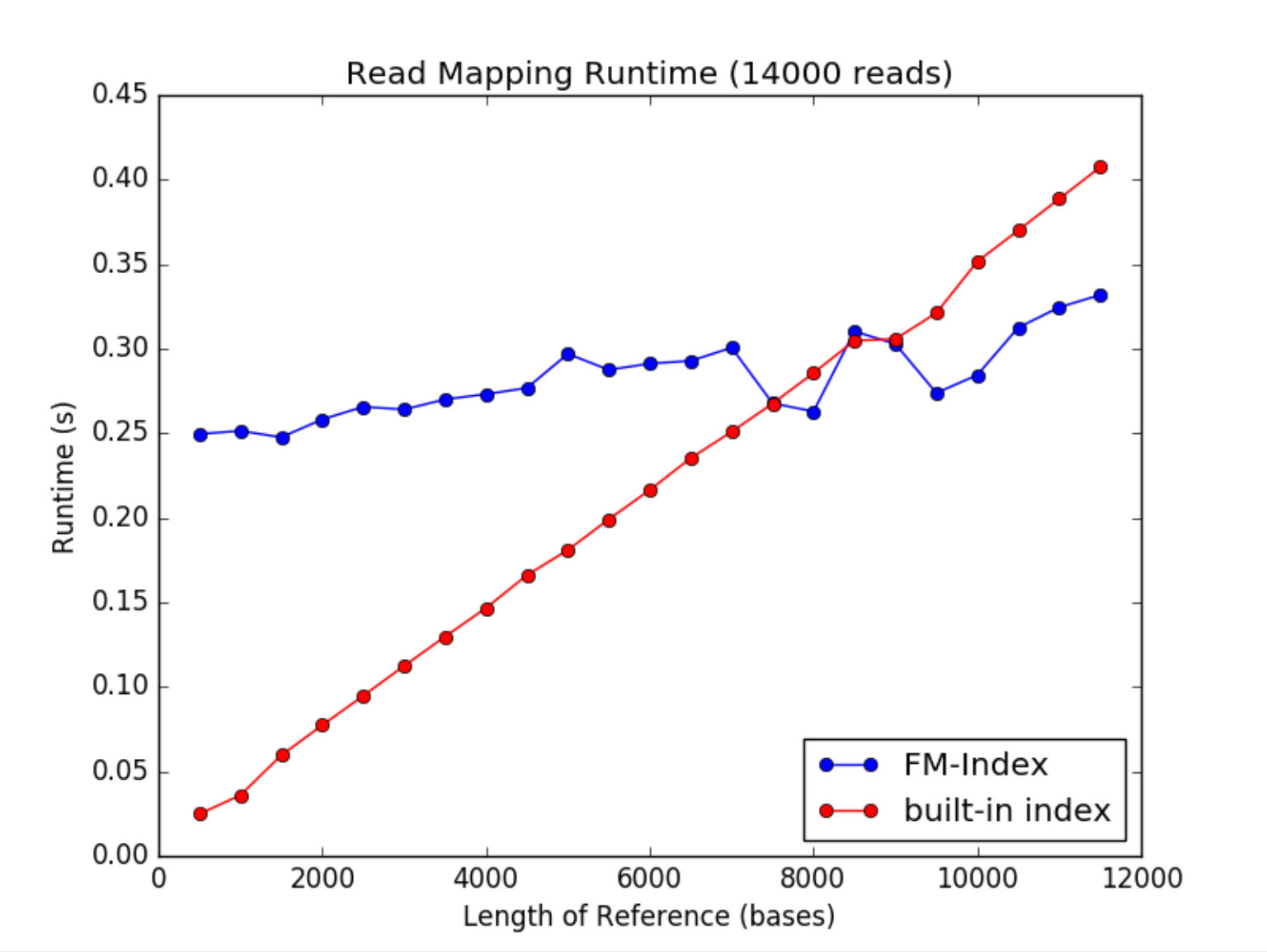
Angelina and Rye



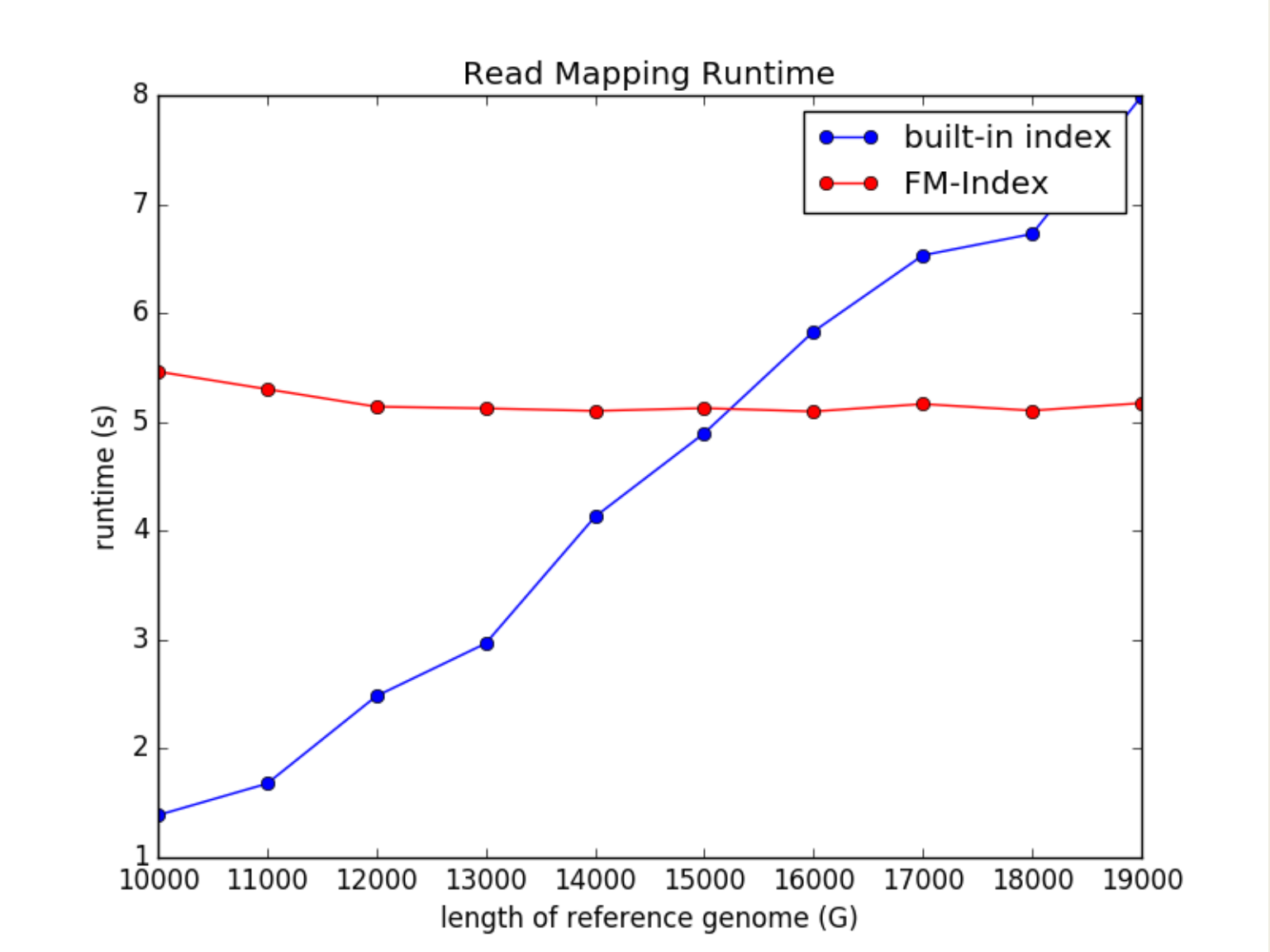
Charlotte and Emily



Kelly and Quinn



Lesia and Linda



Continue Neighbor-Joining (NJ)

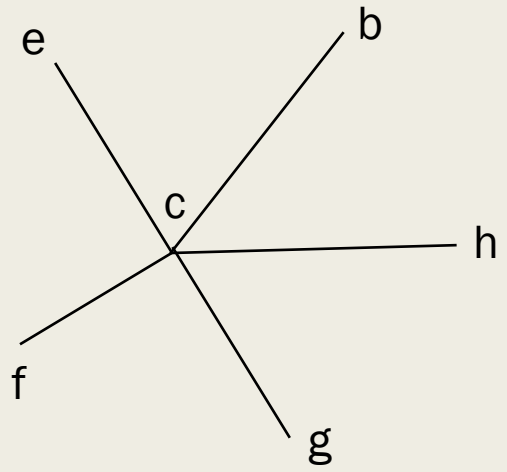
NJ initialization

Input

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

Initialization

- 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.

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

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

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.

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

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

UPGMA

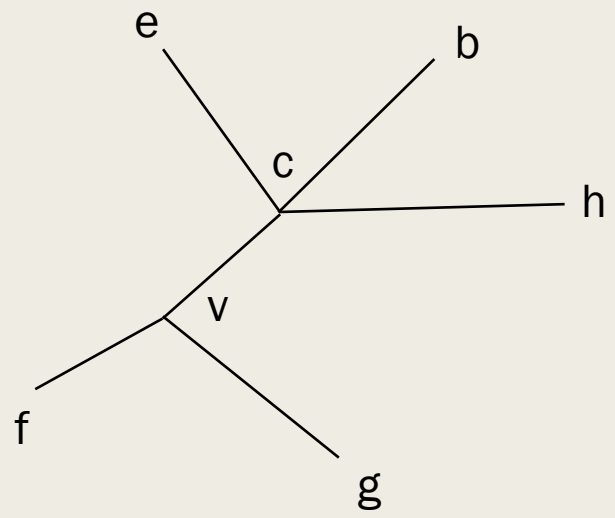
How far away *i* and *j* are from all the other vertices (further away means we'll join them earlier)

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:

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

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

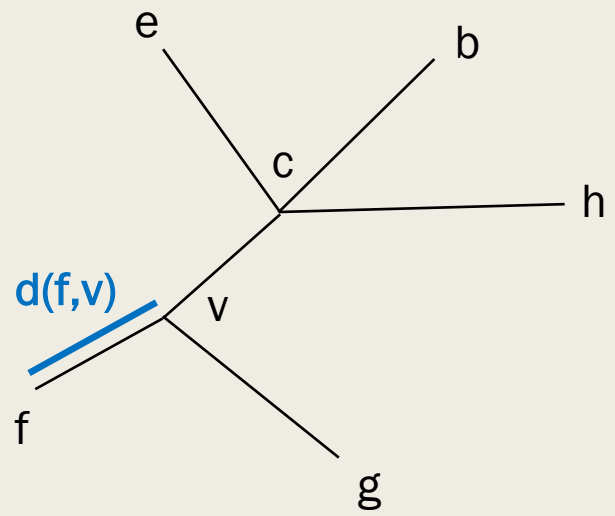


NJ Iterative step (part b)

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

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

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

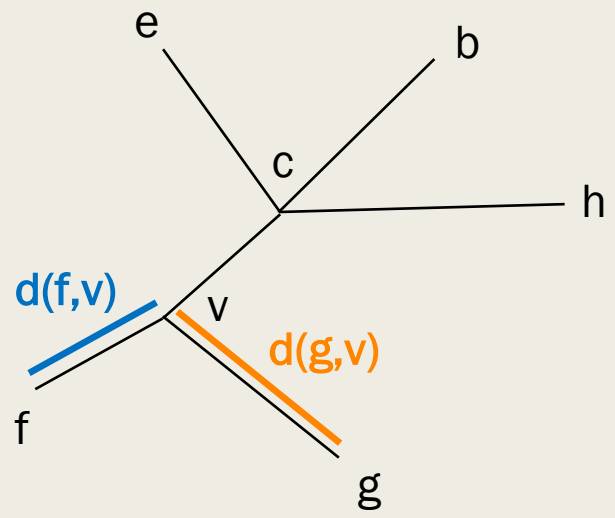


NJ Iterative step (part b)

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

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

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



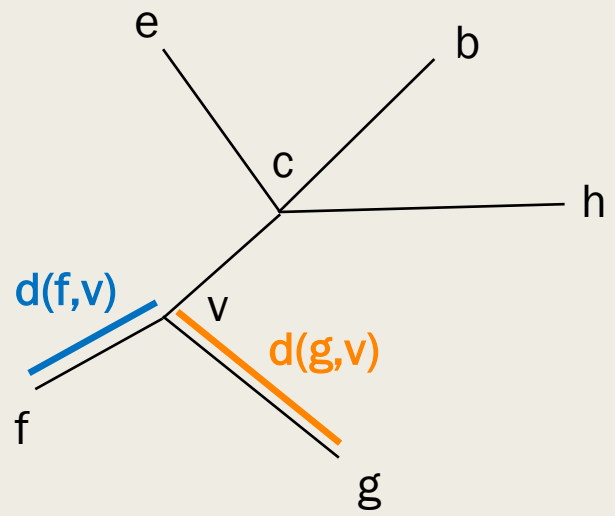
NJ Iterative step (part b)

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:

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

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



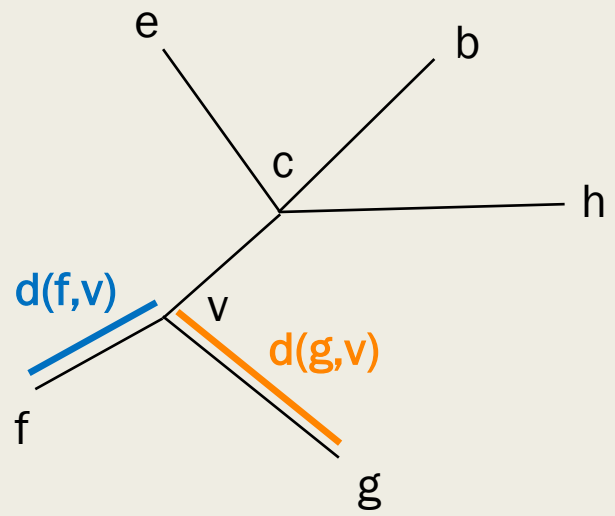
NJ Iterative step (part b)

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:

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

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



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)

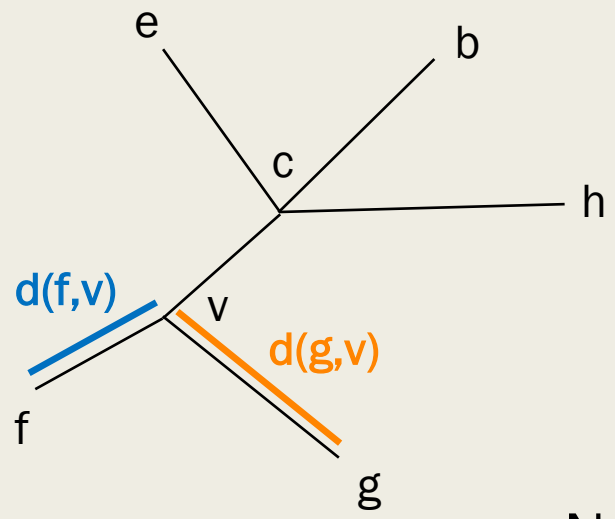
NJ Iterative step (part b)

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:

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

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



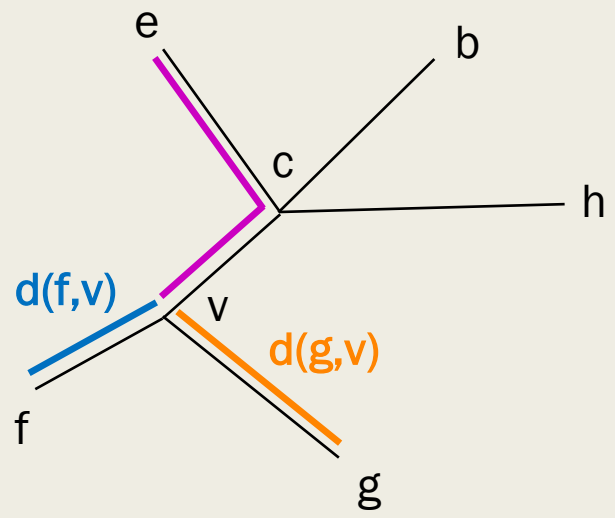
The *difference* between how far *f* and *g* are from other vertices. In this example *g* is on average further from other vertices, so d(g,v) > d(f,v)

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

NJ Iterative step (part c)

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

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



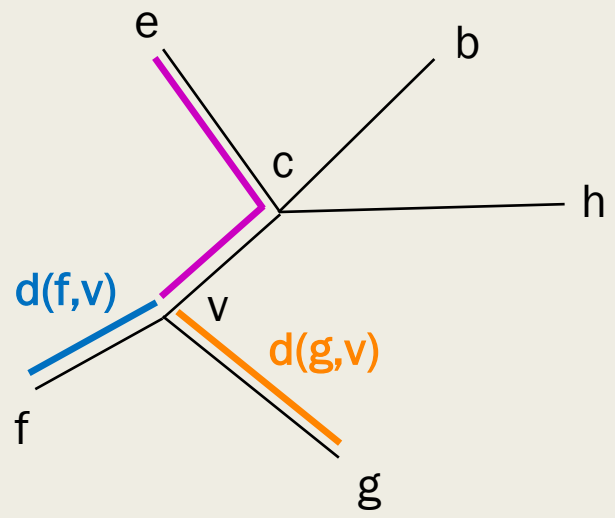
NJ Iterative step (part c)

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

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

Another way to write this:

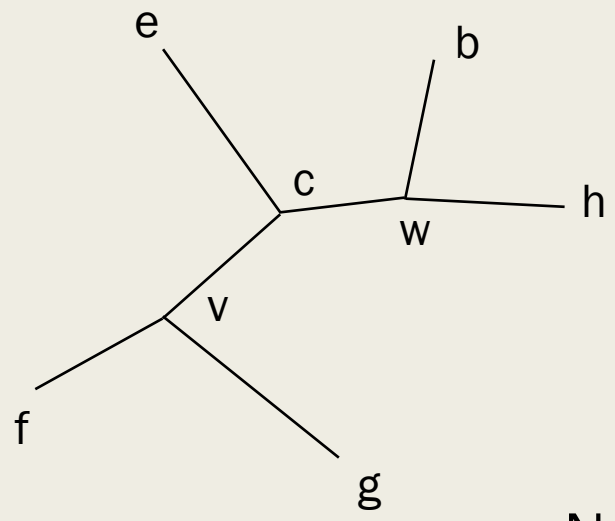
$$d(i,v) = \tfrac{1}{2}[d(f,i) + d(g,i) - d(f,g)]$$



NJ Termination

Termination

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.

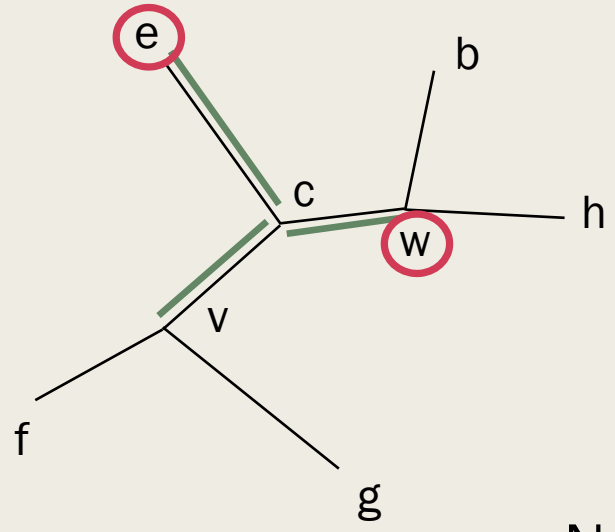


$N_c = \{e, v, w\}, \quad |N_c| = 3$

NJ Termination

Termination

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 13 Solution

① (a) $n=5$

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

i	A	B	C	D	E
S_i	16	13	15	18	18

Q	B	C	D	E
A	-26	-22	-16	-16
B		-22	-16	-16
C			-18	-18
D				-30

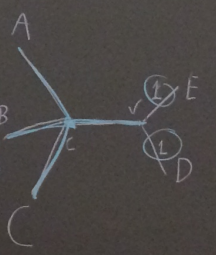
$d(D,v) = 1$

$= \frac{1}{2} d(D,E) + \frac{1}{2(n-2)}(S_D - S_E)$

$d(E,v) = 1$ $\quad 2(5-2)$

Tree:

$N_C = \{A, B, C, v\}$



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

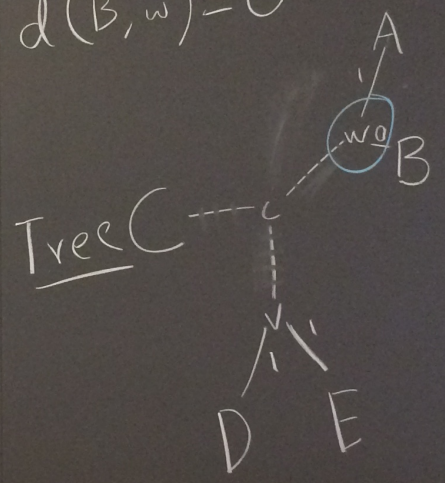
(2)

i	A	B	C	v
S_i	9	7	9	B

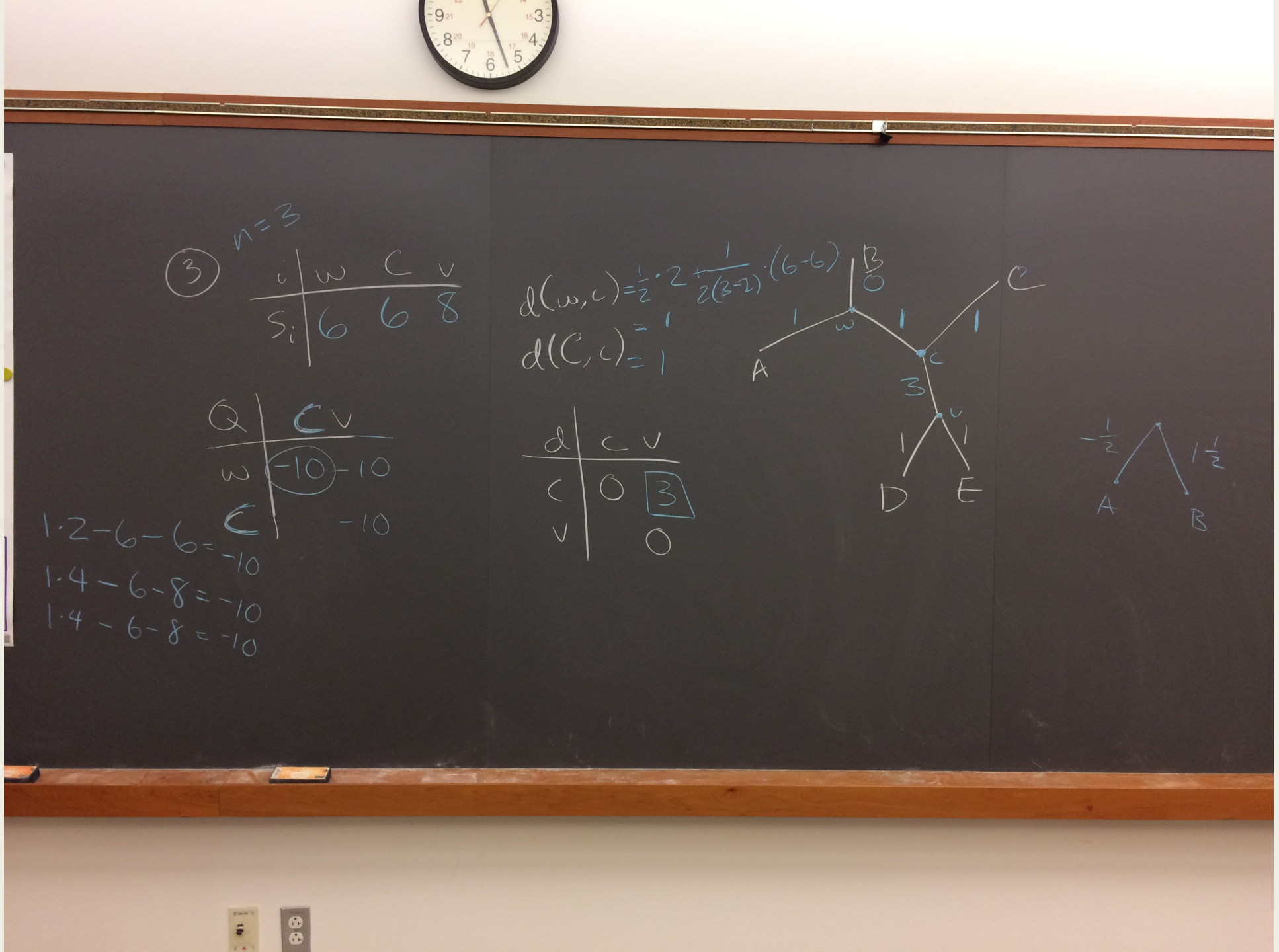
Q	B	C	v
A	-14	-12	-12
B		-12	-12
C			-14

$d(A,w)=1$

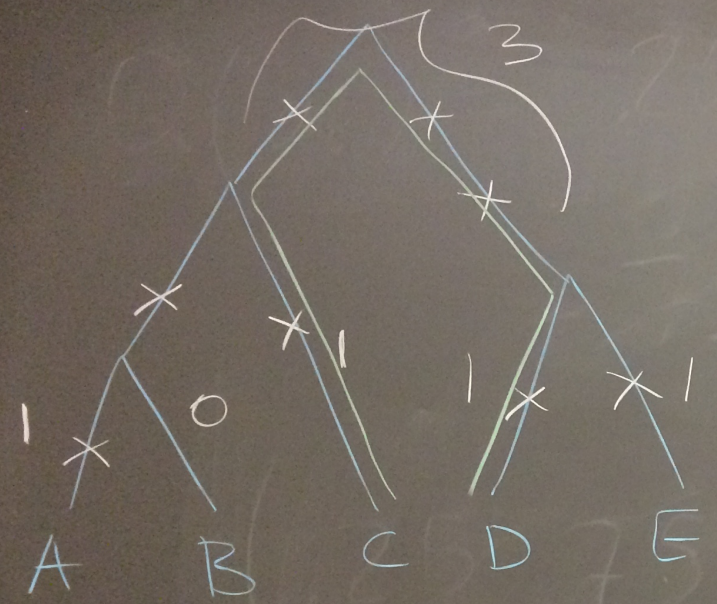
$d(B,w)=0$



d	w	C	v
w	0	2	4
C		0	4
v			0


③ n=3
i | w C v
s_i | 6 6 8
$d(w,c) = \frac{1}{2} \cdot 2 + \frac{1}{2(3-2)} \cdot (6-6)$
$d(C,c) = 1$
Q | C v
w | -10 -10
C | -10
1·2-6-6=-10
1·4-6-8=-10
1·4-6-8=-10
d | c v
c | 0 3
v | 0
A
B
C
D
E
w
c
v
3
1
$-\frac{1}{2}$
$1\frac{1}{2}$

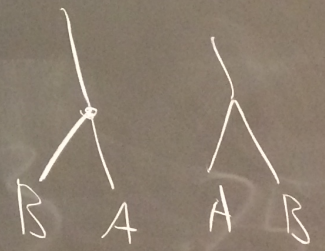
Q-criteria theory and consistency



NJ: is consistent:

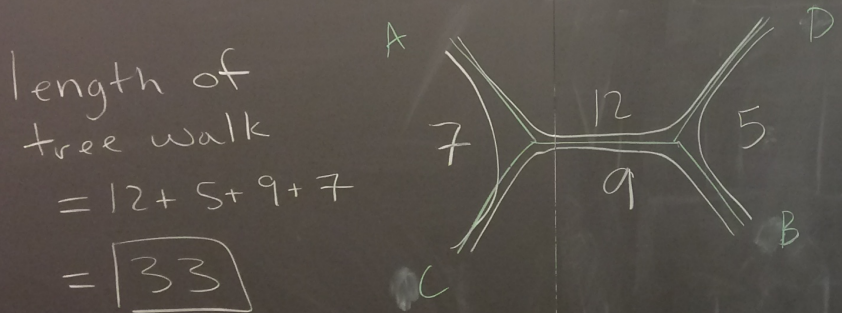
if δ is a tree metric,

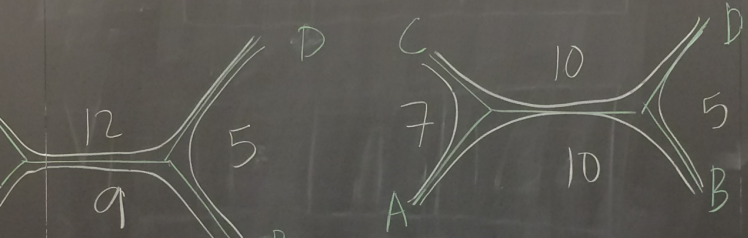
then δ' (induced tree metric from NJ) is equal to δ



Tree walk (Handout 15)

length of tree walk
$= 12 + 5 + 9 + 7$
$= \boxed{33}$

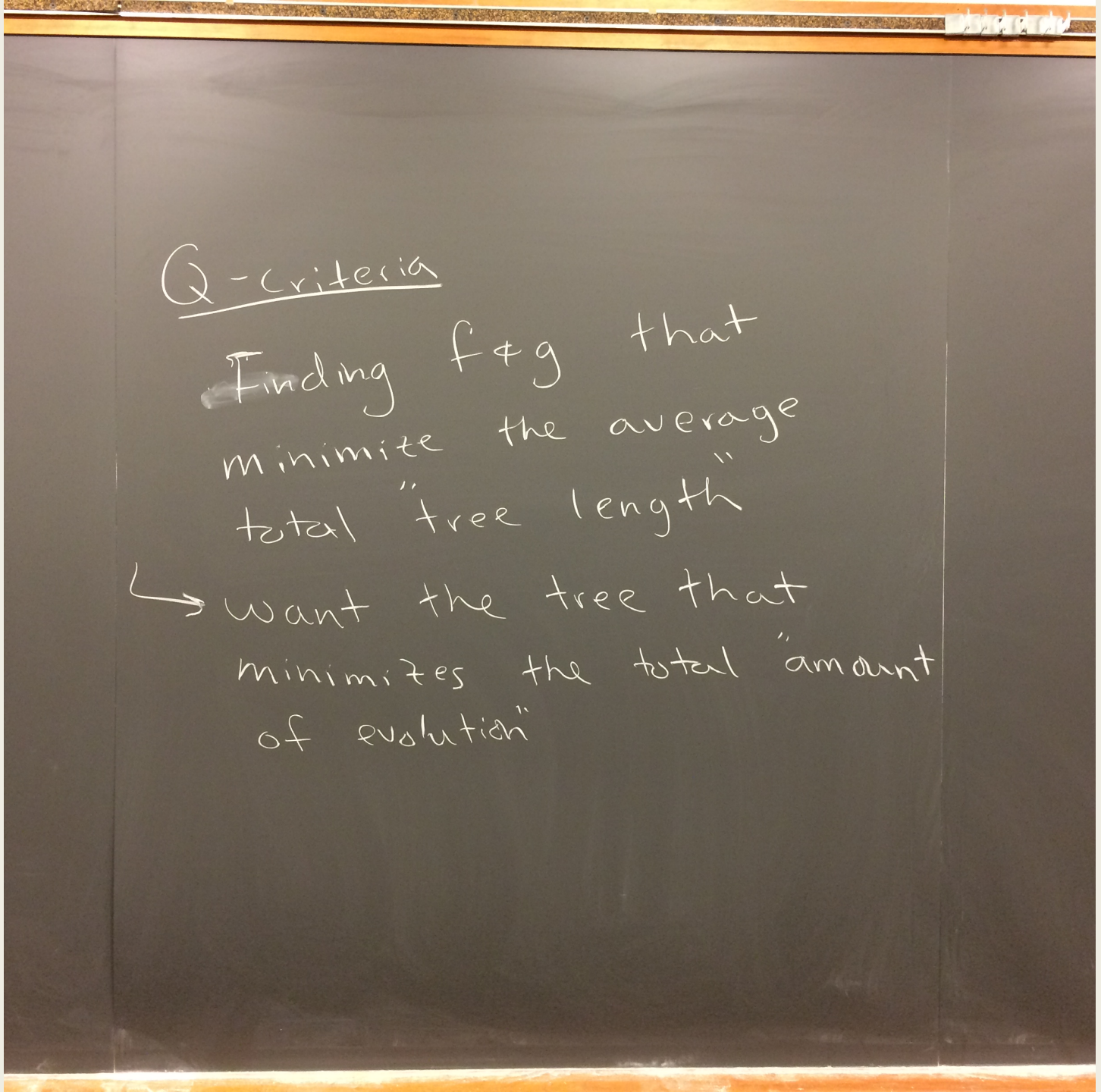




tree walk =
$10 + 5 + 10 + 7$
$= \boxed{32}$

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

not a tree metric


Q-criteria
Finding f+g that
minimize the average
total "tree length"
↳ want the tree that
minimizes the total "amount
of evolution"