Genome 559: Introduction to Statistical and Computational Genomics

Elhanan Borenstein

Who am I?

- Faculty at *Genome Sciences*, *Computer Science*
  (taught 559 for the past 8 years)
- **Training:** CS, physics, hi-tech, biology
- **Research :** Metagenomics; Microbiome; Networks; systems-bio;

What will change?

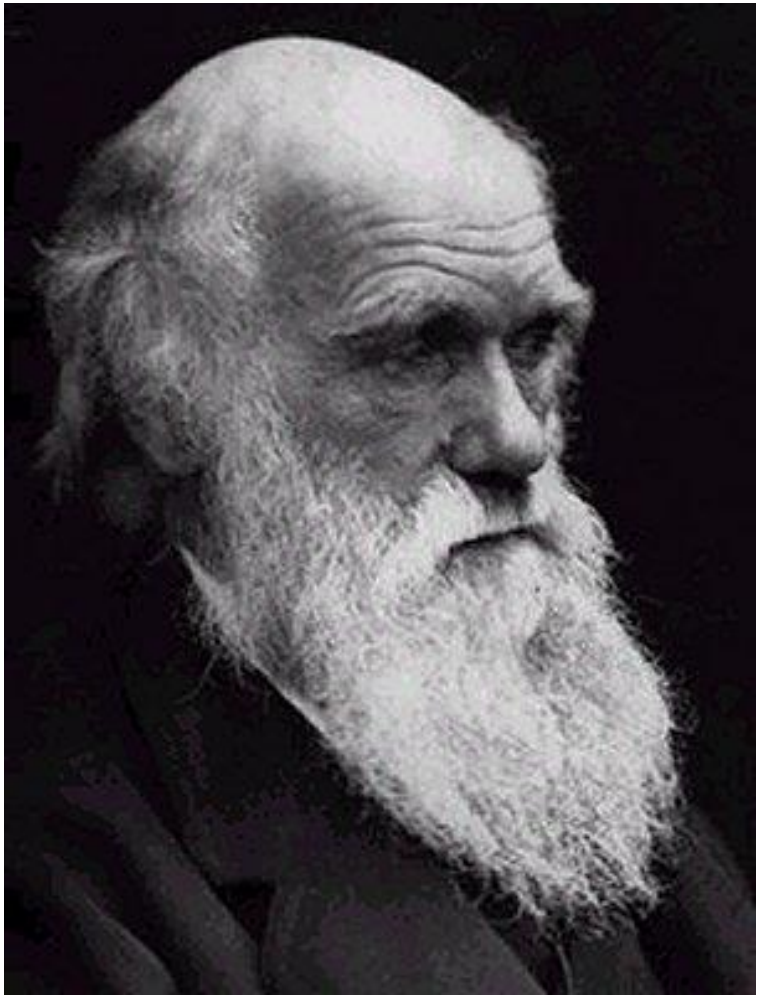
- **Not much!**
- **Scope**: From sequence (to genes) to systems;
- **Bioinformatics:** more emphasis on generic approaches; concepts; algorithm complexity; method development
- **Programming:** more emphasis on design, architecture, coding practices & style; tip of the day;
- More discussions;

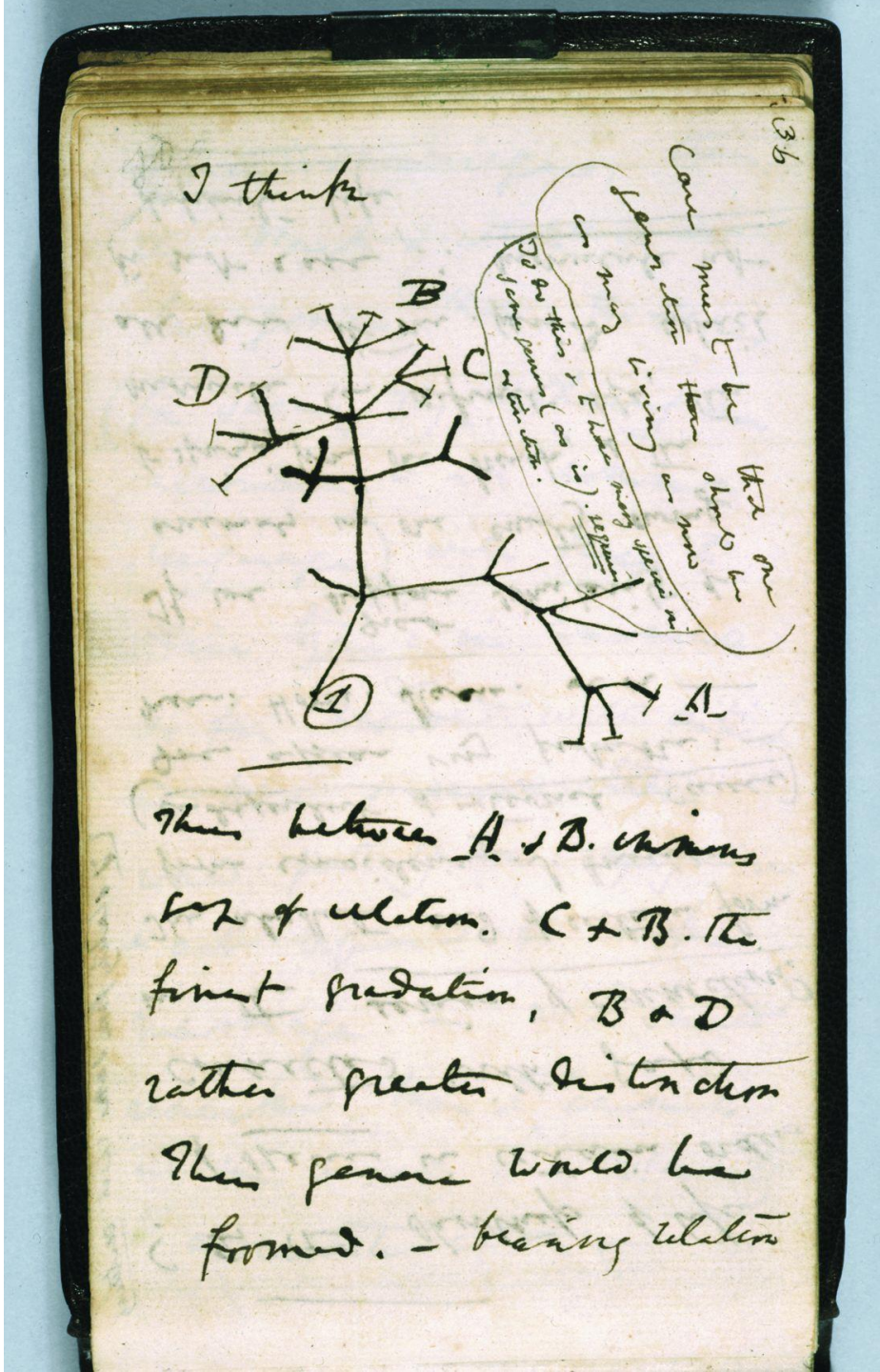
Website: http://elbo.gs.washington.edu/courses/GS_559_18_wi/

Phylogenetic Trees

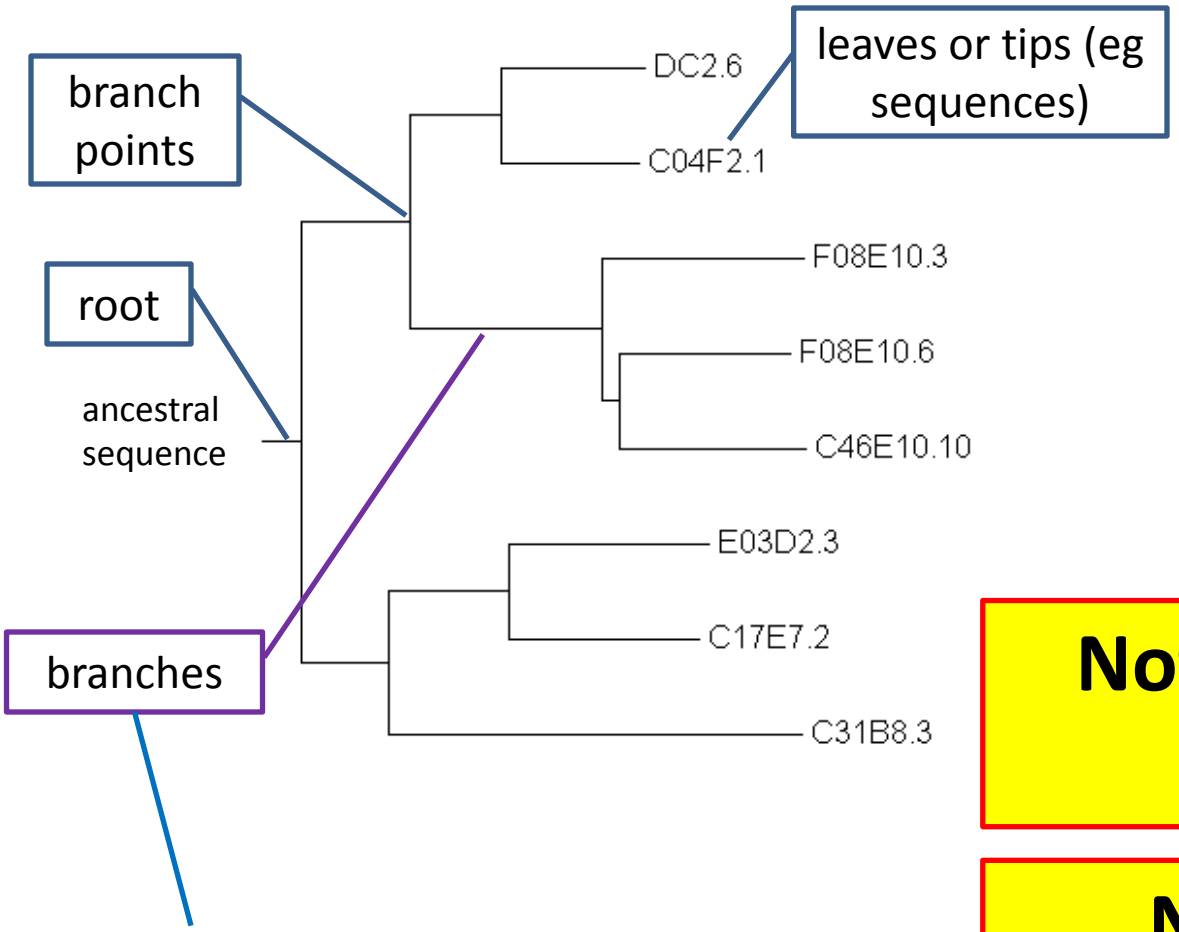
Genome 559: Introduction to Statistical and Computational Genomics

Elhanan Borenstein




36
I think
B
C
D
A
I

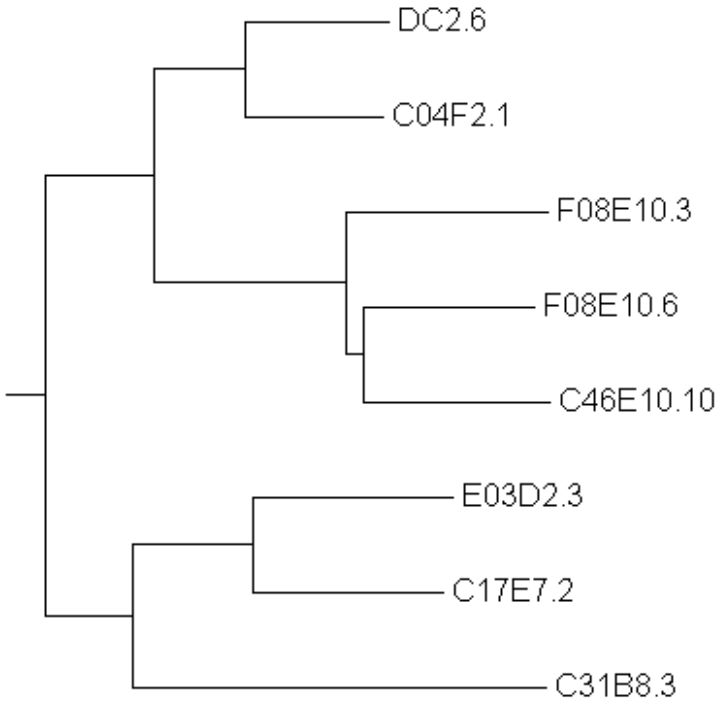
Defining what a “tree” means

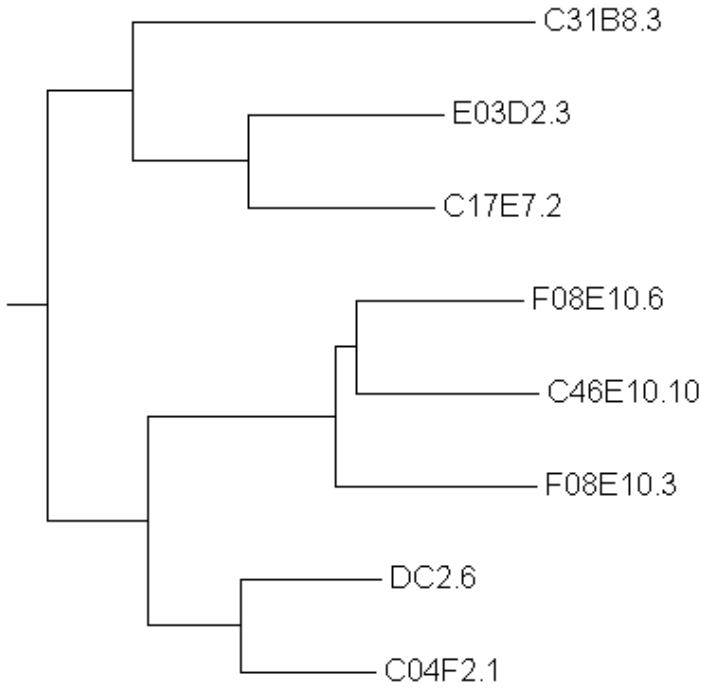


Note: Many drawing practices exist

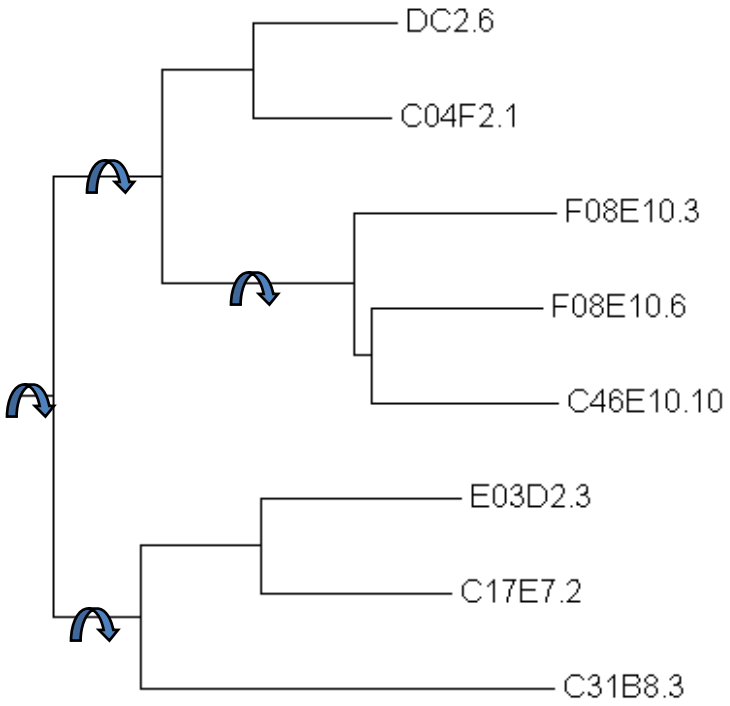
Note: A tree has topology and distances

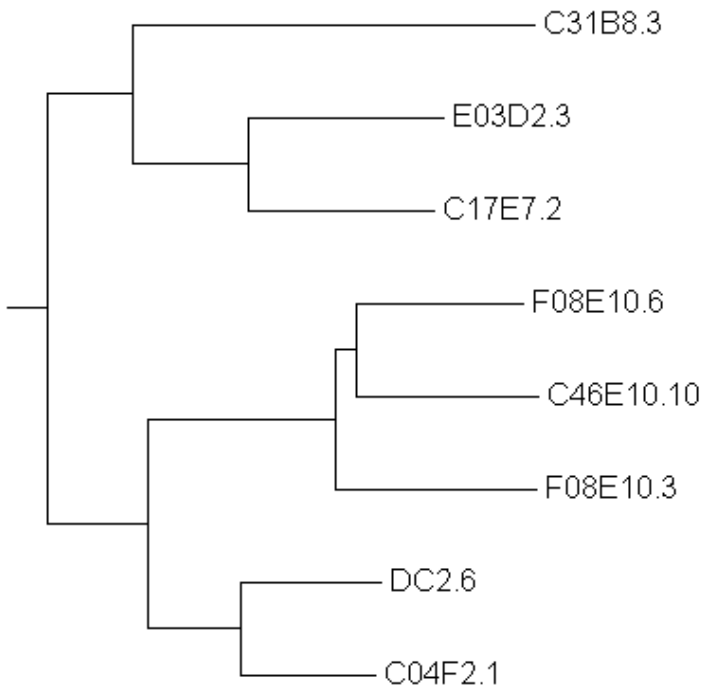
Are these topologically different trees?





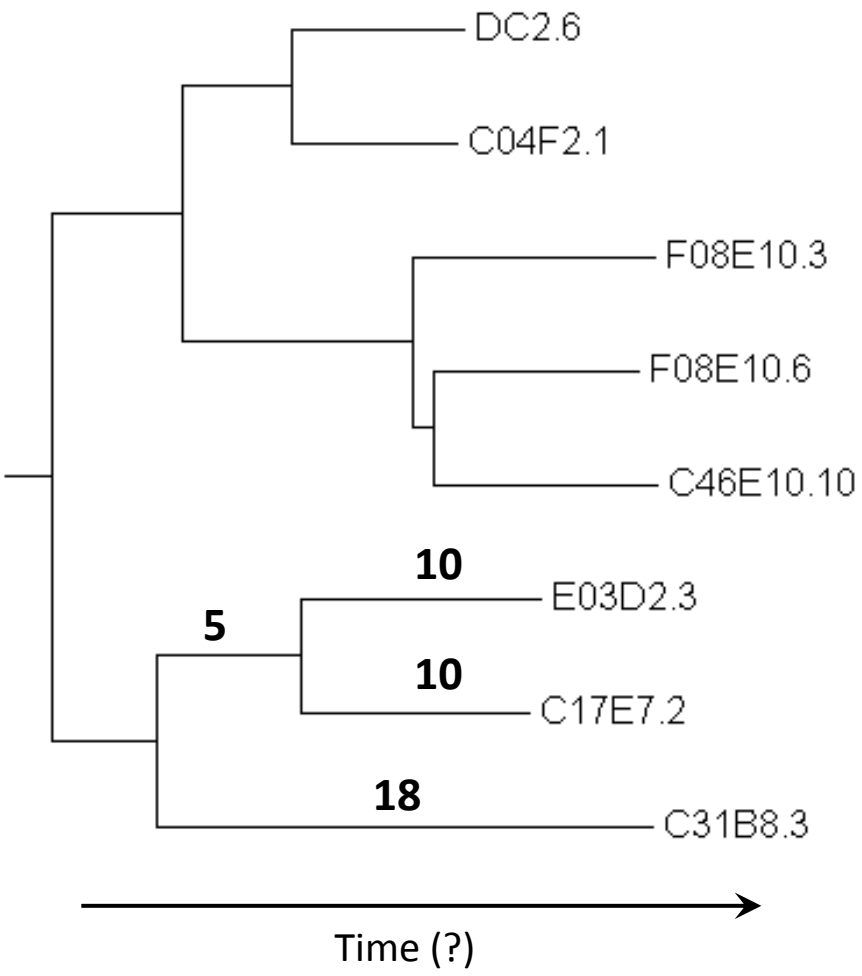
Are these topologically different trees?





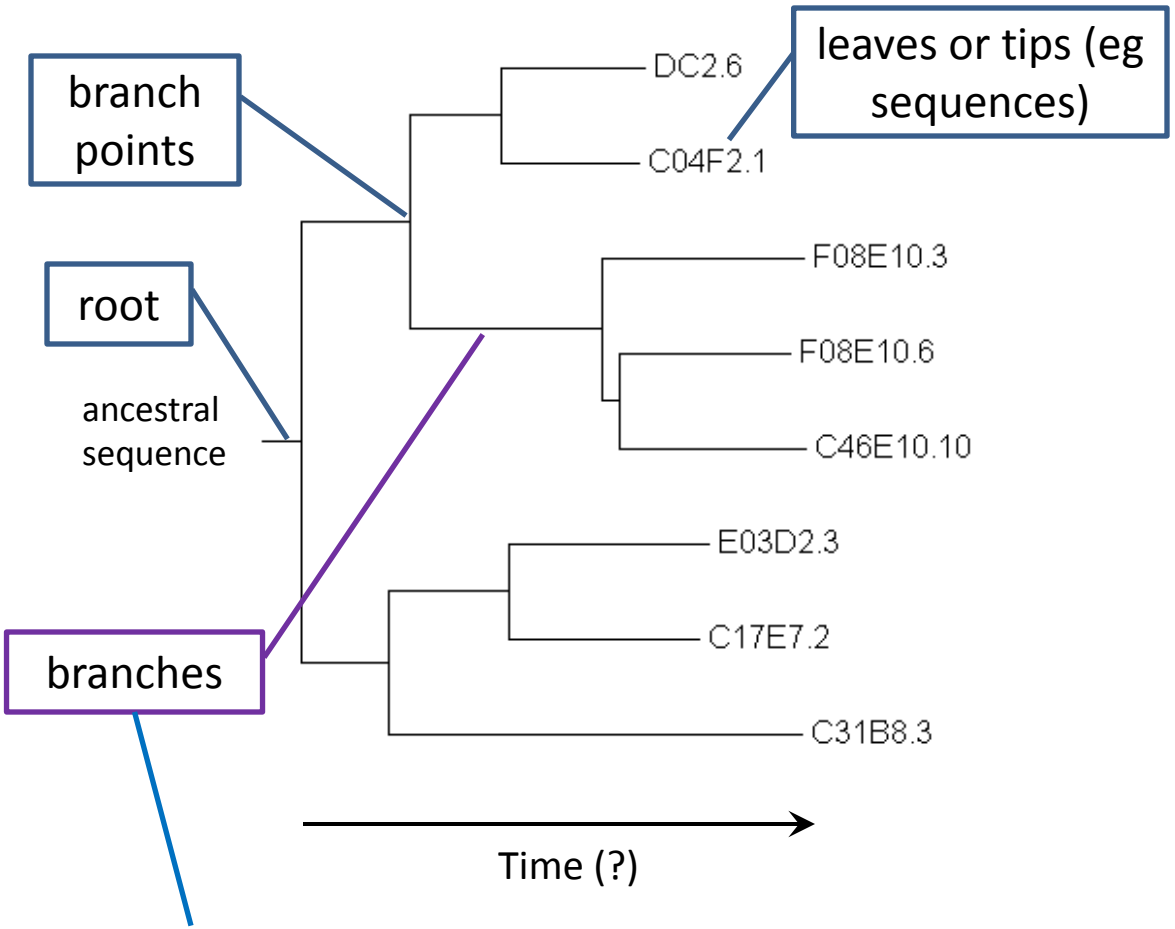
Topologically, these are the SAME tree. In general, two trees are the same if they can be inter-converted by branch rotations.

Branch lengths and evolutionary divergence time



Rooted and unrooted trees

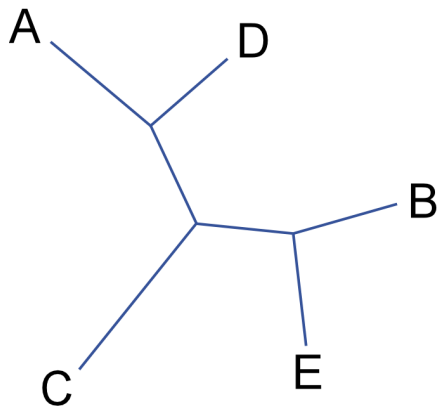
Rooted tree (all real trees are rooted):



… (horizontal) branch lengths sequence is proportional to divergence

Unrooted tree

(used when the root isn't known):

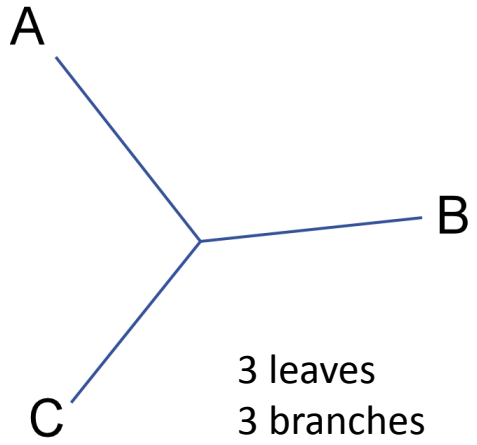


time radiates out from somewhere (probably near the center)

Why is inferring phylogeny a hard problem?

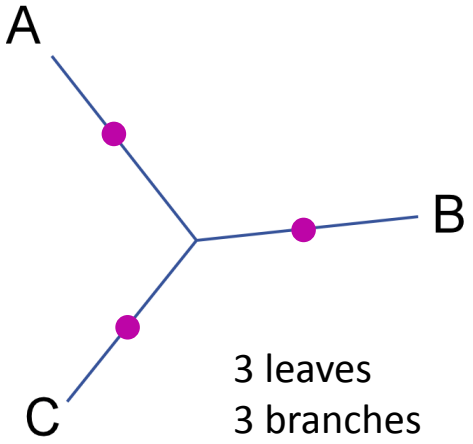
(assume, for example, we are trying to infer the phylogenetic tree for 20 primate species)

The number of tree topologies grows extremely fast

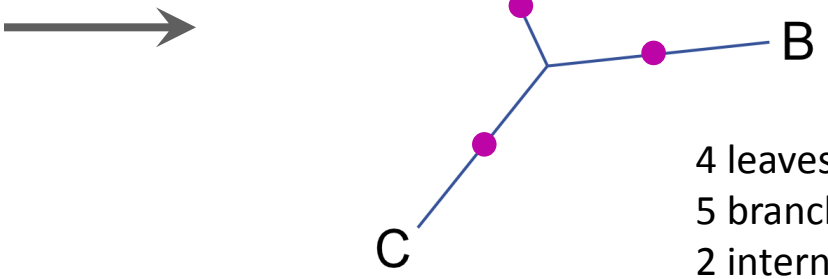


3 leaves
3 branches
1 internal node
1 topology
(3 insertions)

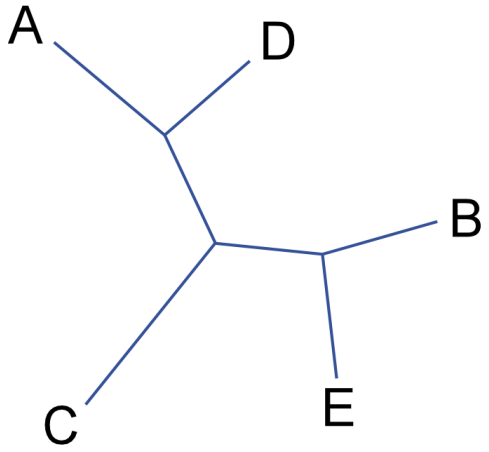
The number of tree topologies grows extremely fast



3 leaves
3 branches
1 internal node
1 topology
(3 insertions)

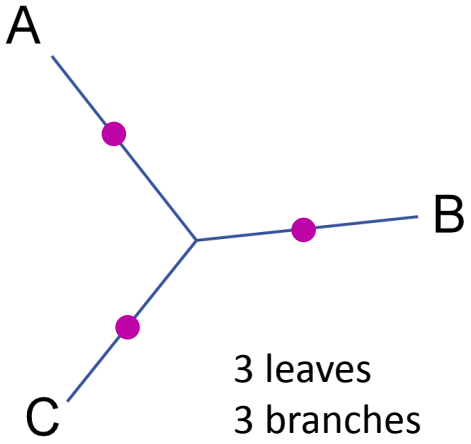


4 leaves
5 branches
2 internal nodes
3 topologies (x3)
(5 insertions)

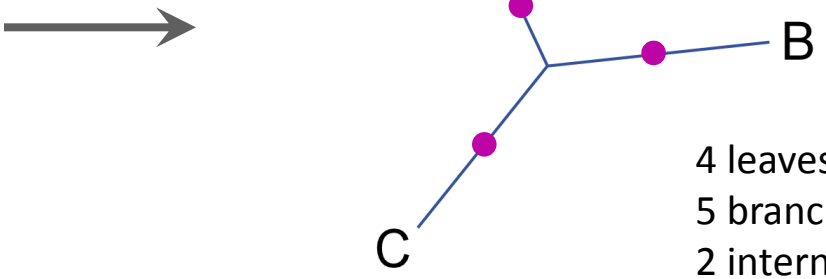


5 leaves
7 branches
3 internal nodes
15 topologies (x5)
(7 insertions)

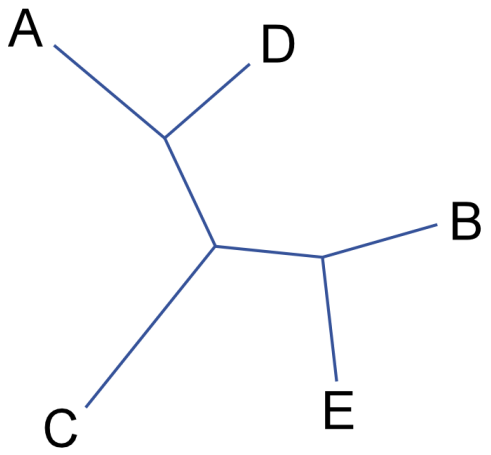
The number of tree topologies grows extremely fast



3 leaves
3 branches
1 internal node
1 topology
(3 insertions)



4 leaves
5 branches
2 internal nodes
3 topologies (x3)
(5 insertions)

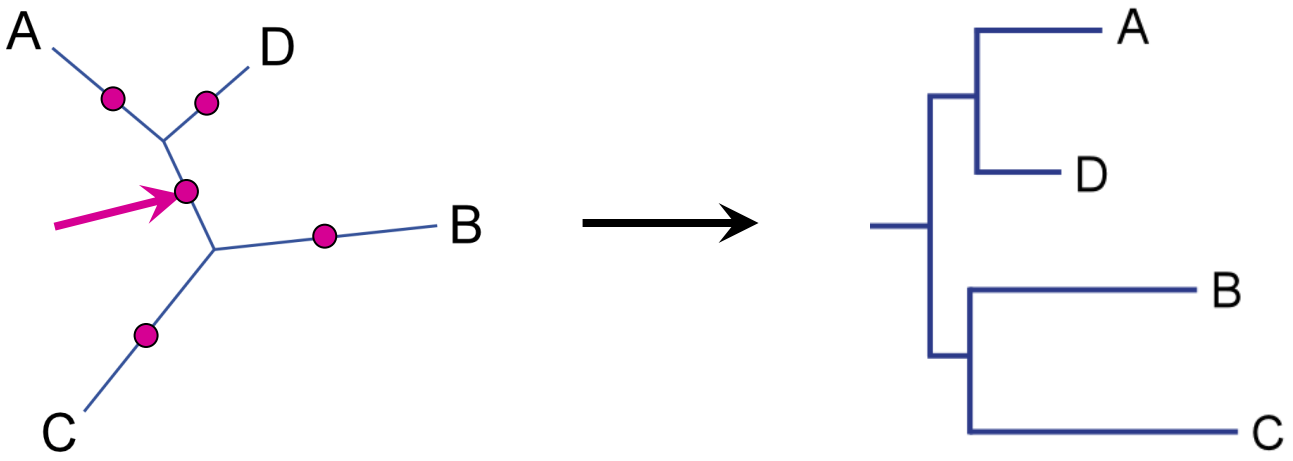


5 leaves
7 branches
3 internal nodes
15 topologies (x5)
(7 insertions)

In general, an unrooted tree with **N** leaves has:
2N - 3 total branches
N leaf branches
N - 3 internal branches
N - 2 internal nodes
3*5*7*…*(2N-5) ~O(N!) topologies

There are many rooted trees for each unrooted tree

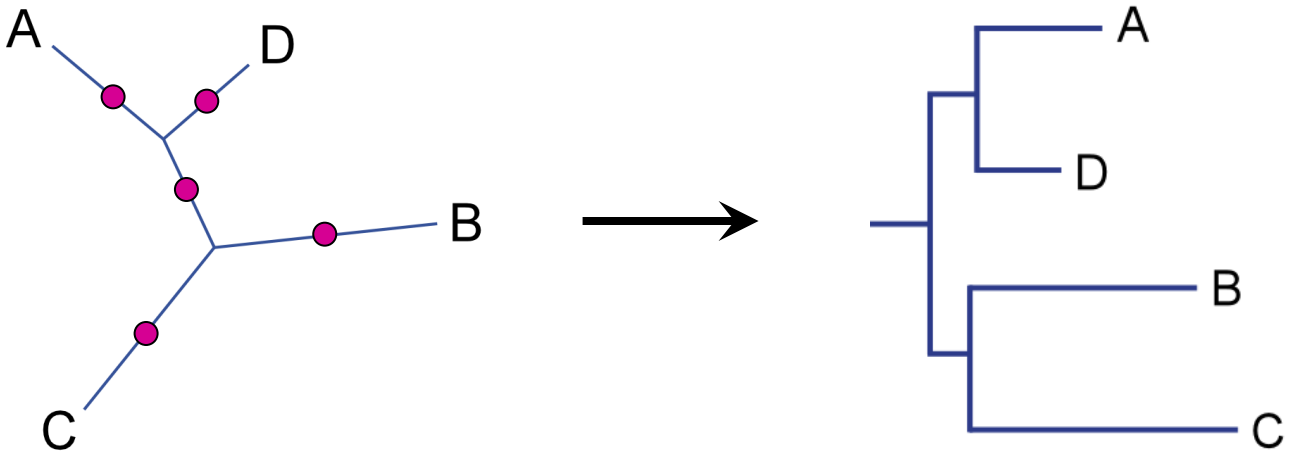
For each unrooted tree, there are 2N - 3 times as many rooted trees, where N is the number of leaves (# branches = 2N – 3).



The number of tree topologies grows extremely fast

There are many rooted trees for each unrooted tree

For each unrooted tree, there are 2N - 3 times as many rooted trees, where N is the number of leaves (# branches = 2N – 3).



The number of tree topologies grows extremely fast

20 leaves - 564,480,989,588,730,591,336,960,000,000 topologies

How can you compute a tree?

- Many methods available, we will talk about:
  - Distance trees
  - Parsimony trees
- Others include:
  - Maximum-likelihood trees
  - Bayesian trees

Distance matrix methods

- Methods based on a set of pairwise distances typically from a multiple alignment.

```
                1 2 3 4 5 6
human           a g t c t c
chimp           a g a g t c
gorilla         c g g c a g
orangutan       c g g g a c
```

human - chimp has 2 changes out of 6 sites
human - orang has 4 changes of out 6 sites
etc.

	human	chimp	gorilla	orang
human	0	2/6	4/6	4/6
chimp		0	5/6	3/6
gorilla			0	2/6
orang				0

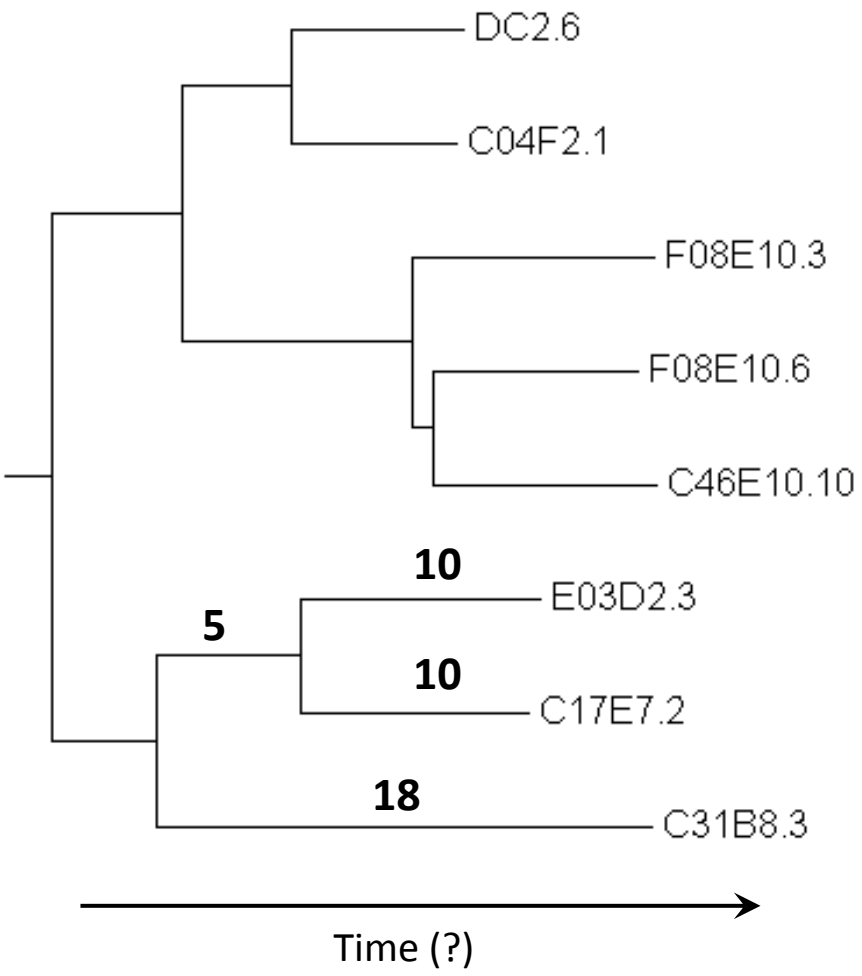
(symmetrical, lower left not filled in)

- **Many different metrics can be used !!**

Approach:

Try to build the tree whose **distances best match** the real distances

Trees and distances



	E03D2.3	C17E7.2	C31B8.3	…
E03D2.3	0	20	33	.
C17E7.2		0	33	.
C31B8.3			0	.
…				0

Best Match?

- "Best match" based on **least squares** of real pairwise distances compared to the tree distances:

Let D_m be the measured distances.

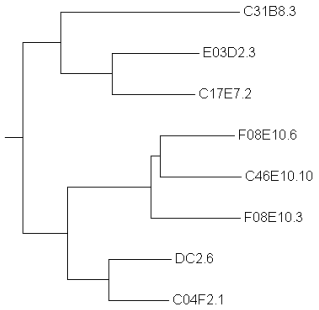
	1	2	3	4	5	6
human	a	g	t	c	t	c
chimp	a	g	a	g	t	c
gorilla	c	g	g	c	a	g
orangutan	c	g	g	g	a	c

human - chimp has 2 changes out of 6 sites
human - orang has 4 changes of out 6 sites
etc.

	human	chimp	gorilla	orang
human	0	2/6	4/6	4/6
chimp		0	5/6	3/6
gorilla			0	2/6
orang				0

(symmetrical, lower left not filled in)

Let D_t be the tree distances.



Find the tree that minimizes:

$$\sum_{i=1}^{N} \left(D_t - D_m \right)^2$$

Why not enumerate and score all trees?

The UPGMA algorithm

(**U**nweighted **P**air **G**roup **M**ethod with **A**rithmetic Mean)

1) generate a table of pairwise sequence distances and assign each sequence to a list of N tree nodes.

2) look through current list of nodes (initially these are all leaf nodes) for the pair with the smallest distance.

3) merge the closest pair, remove the pair of nodes from the list and add the merged node to the list.

4) repeat until only one node left in list - it is the root.

The UPGMA algorithm

(**U**nweighted **P**air **G**roup **M**ethod with **A**rithmetic Mean)

1) generate a table of pairwise sequence distances and assign each sequence to a list of N tree nodes.

2) look through current list of nodes (initially these are all leaf nodes) for the pair with the smallest distance.

3) merge the closest pair, remove the pair of nodes from the list and add the merged node to the list.

4) repeat until only one node left in list - it is the root.

$$D_{n1,n2} = \frac{1}{N} \sum_i \sum_j d_{ij}$$

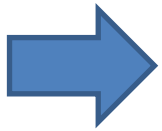
where i is each leaf of $n1$ (node1), j is each leaf of $n2$ (node2), and N is the number of distances summed

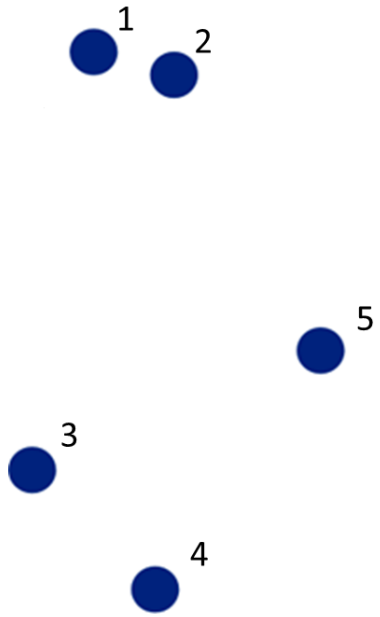
definition of distance

(in words, this is just the arithmetic average of the distances between all the leaves in one node and all the leaves in the other node)

The UPGMA algorithm

	1	2	3	4	5
1	0	5	18	22	17
2		0	20	24	15
3			0	10	12
4				0	12
5					0





distances

N tree

tially these

ne smallest

air of nodes

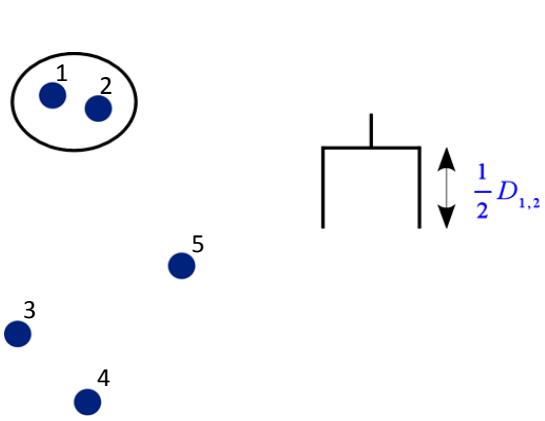
e to the list.

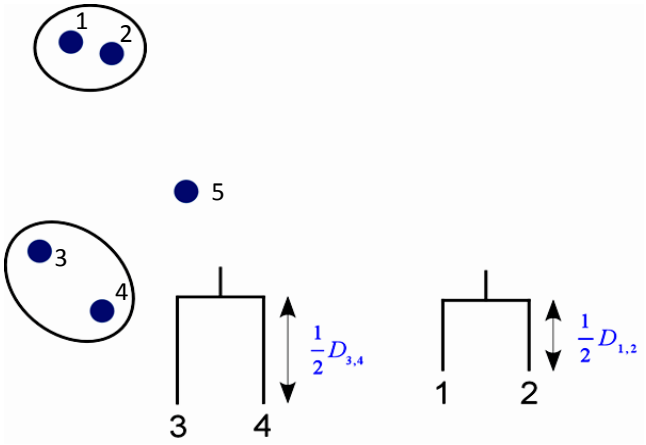
- it is the

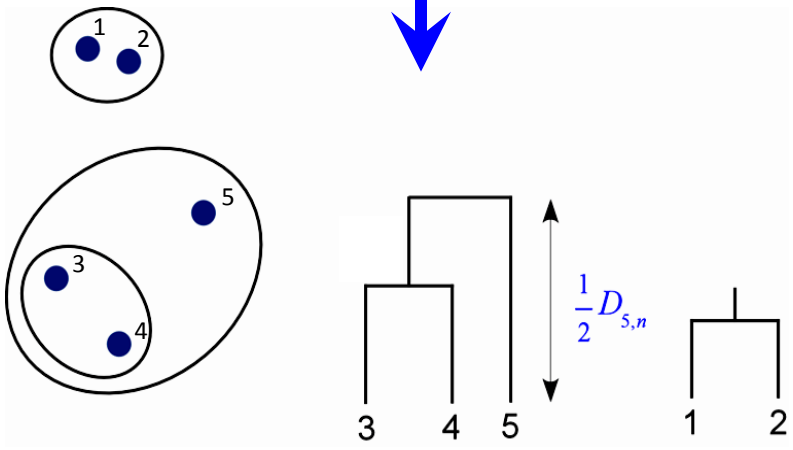
UPGMA

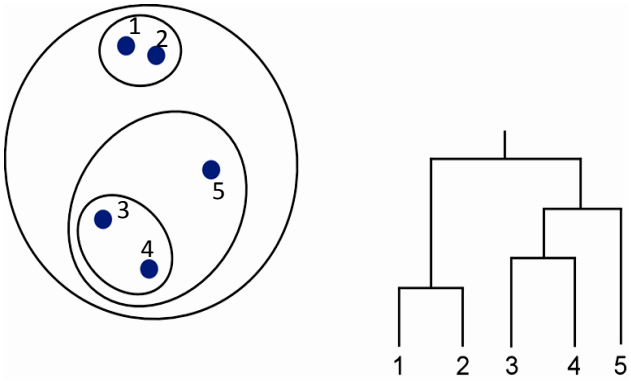
(**U**nweighted **P**air **G**roup **M**ethod with **A**rithmetic Mean)

	1	2	3	4	5
1	0	5	18	22	17
2		0	20	24	15
3			0	10	12
4				0	12
5					0



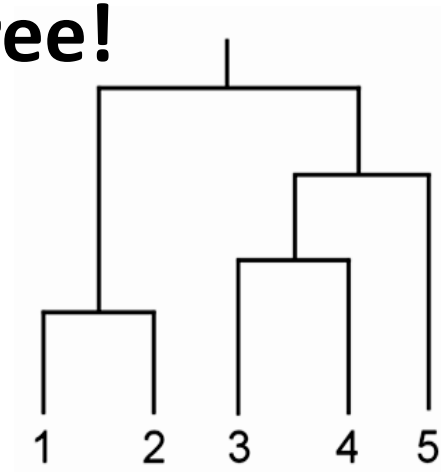




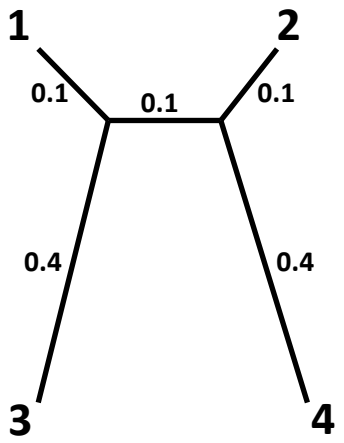


The Molecular Clock

- **UPGMA assumes a constant rate of the molecular clock across the entire tree!**
  - The sum of times down a path to any leaf is the same



- This assumption may not be correct … and will lead to incorrect tree reconstruction.



Neighbor-Joining (NJ) Algorithm

- Essentially similar to UPGMA, but correction for distance to other leaves is made.

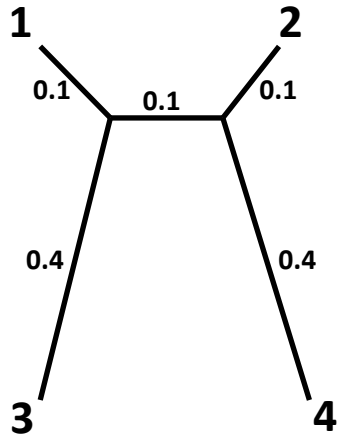
- Specifically, for sets of leaves *i* and *j*, we denote the set of all **other** leaves as *L*, and the size of that set as $|L|$, and we compute the corrected distance D_{ij} as:

$$D_{ij} = d_{ij} - (r_i + r_j)$$

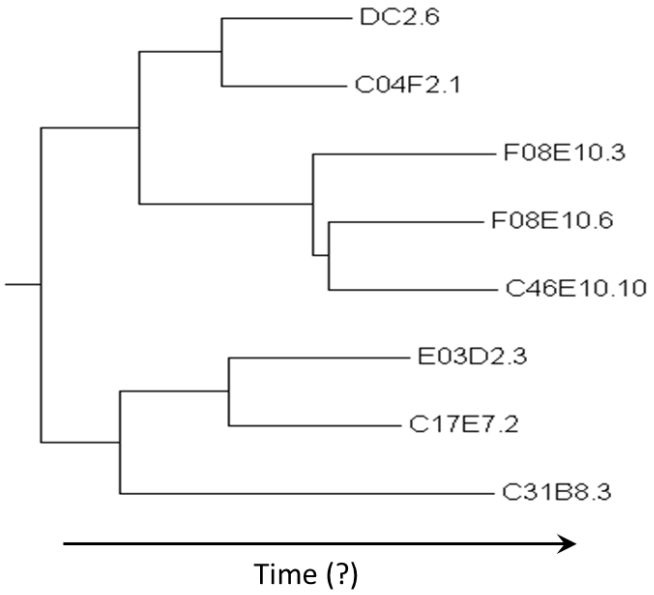
where

$$r_i = \frac{1}{|L|} \sum_{k \in L} d_{ik}$$

(the mean distance from i to all 'other' leaves)



But wait, there's one more problem



VS

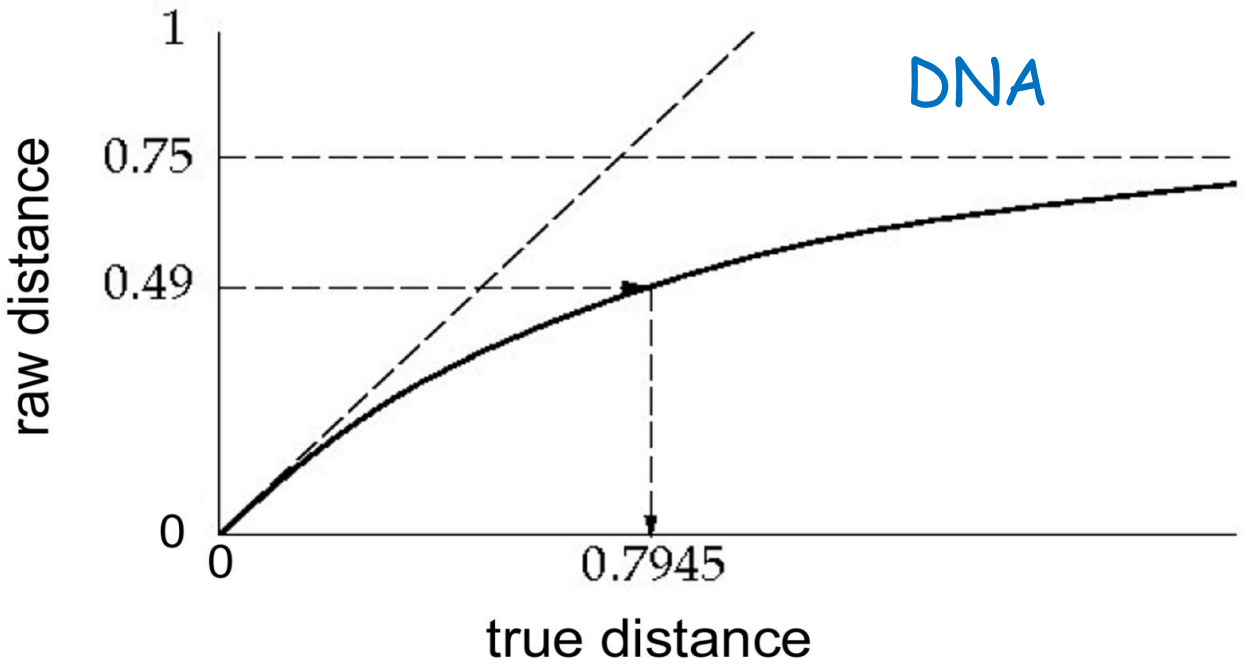
	E03D2.3	C17E7.2	C31B8.3	…
E03D2.3	0	20	33	.
C17E7.2		0	33	.
C31B8.3			0	.
…				0

Raw distance correction

- As two DNA sequences diverge, it is easy to see that their maximum raw distance is ~0.75 (assuming equal nt frequencies, ¼ of residues will be identical even if unrelated sequences).

- We would like to use the "true" distance, rather than raw distance.

- This graph shows evolutionary distance related to raw distance:



Jukes-Cantor model

Jukes-Cantor model:

$$D = -\frac{3}{4}\ln\left(1 - \frac{4}{3}D_{raw}\right)$$

D_{raw} is the raw distance (what we directly measure)

D is the corrected distance (what we want)

Distance trees - summary

- Convert each pairwise raw distance to a corrected distance.

- Build tree as before (UPGMA algorithm).

- Notice that these methods don't need to consider all tree topologies - they are very fast, even for large trees.

Enumerate and score all trees?

- **How about the following algorithm:**

1) *Construct all possible trees*
2) Fit least-squares best distances for each topology
3) *Pick the tree with the best score*

Will take too much time !!

There is a much faster way to get very close to correct.