Phylogenetic Trees

Distance trees

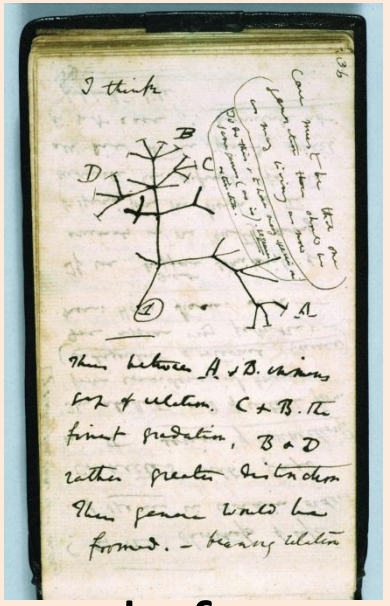
Genome 373

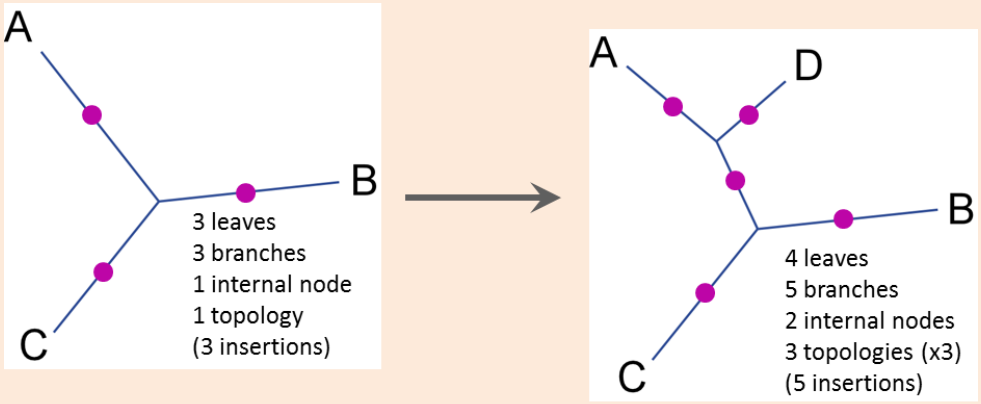
Genomic Informatics

Elhanan Borenstein

A quick review

- Tree definition
  - Leaves, root, and branchs
  - Evolutionary time vs. chronological time
- Tree structure
  - Topology vs. branch lengths
- The number of tree topologies grows extremely fast
- Tree inference methods





Trees and Distances

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)

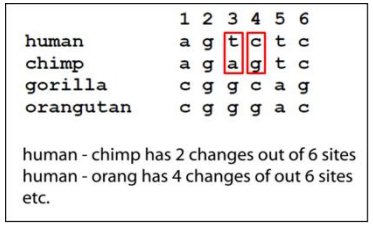
- Try to build the tree whose distances best match the real distances.

Best Match?

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

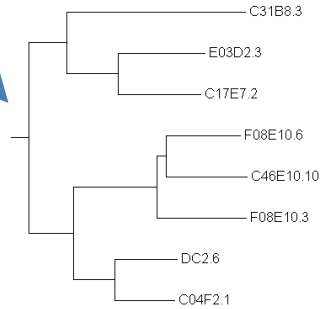
Let D_m be the measured distances.
Let D_t be the tree distances.
Find the tree that minimizes:

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



	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)



Enumerate and score all trees?

- **How about the following algorithm:**
  Enumerate every tree topology, fit least-squares best distances for each topology, keep best.

- Not used for distance trees - there is a much faster way to get very close to correct.

The UPGMA algorithm

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.

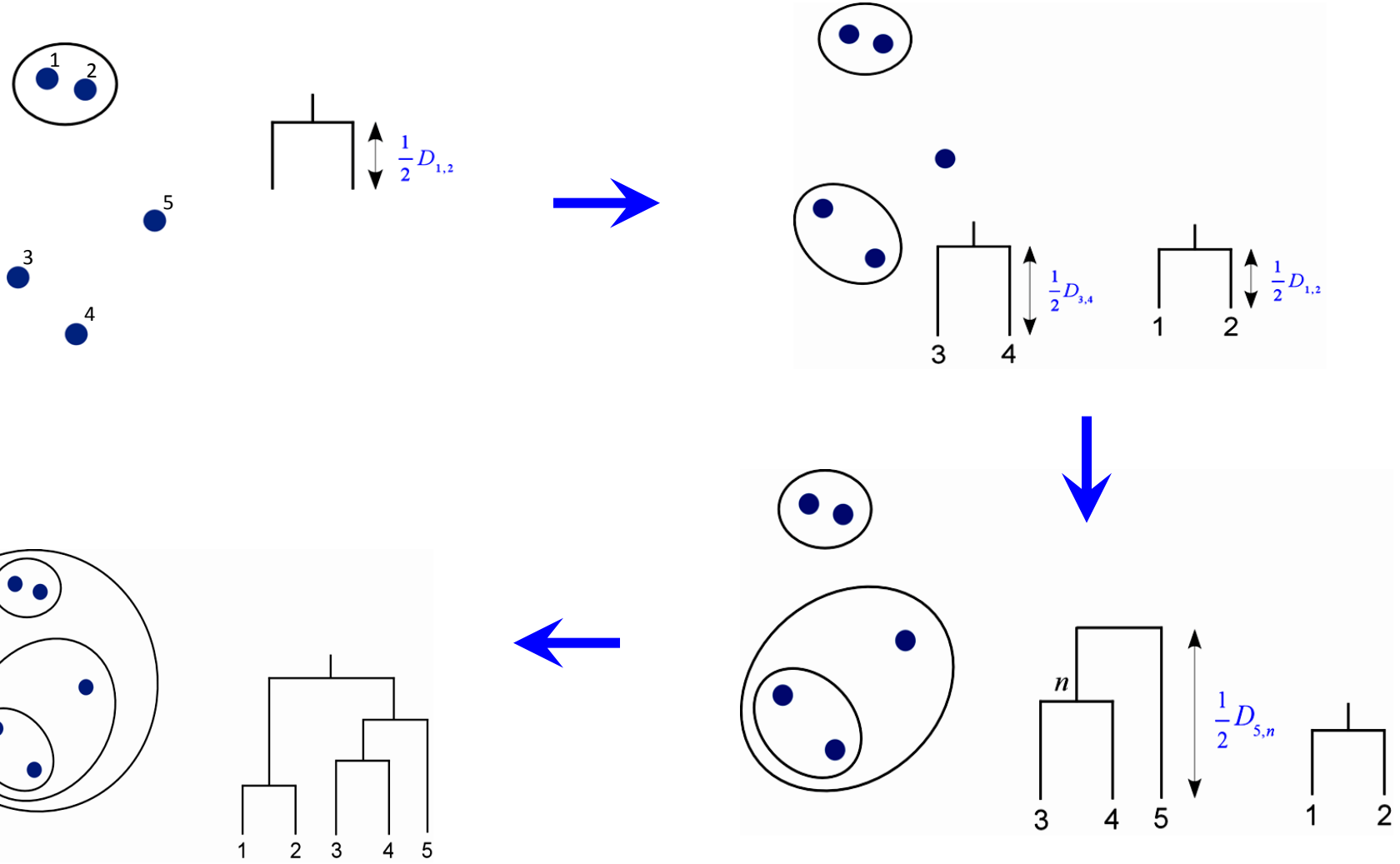
$$D_{n1,n2} = \frac{1}{N} \sum_i \sum_j d_{ij}$$

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

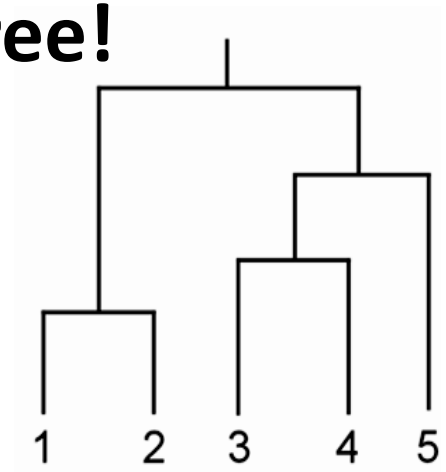
(in words, this is just the arithmetic average of the distances between all the leaves in one node and all the leaves in the other node)

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

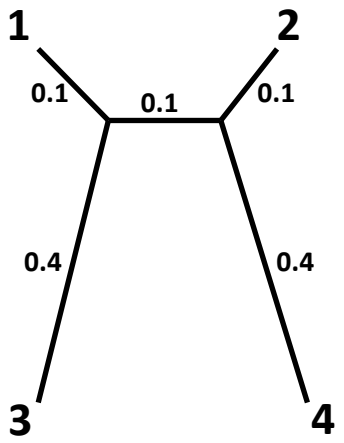


The Molecular Clock

- **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.

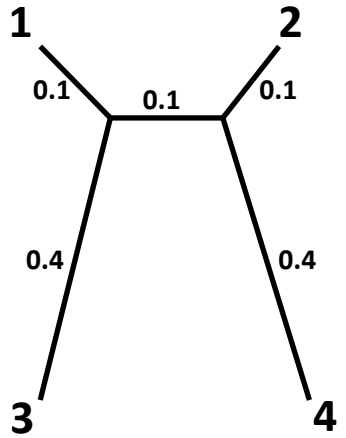
- Specifically, for sets of leaves *i* and *j*, we denote the set of all **other** leaves as *L*, and the size of that set as $|L|$, and we compute the corrected distance D_{ij} as:

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

where

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

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

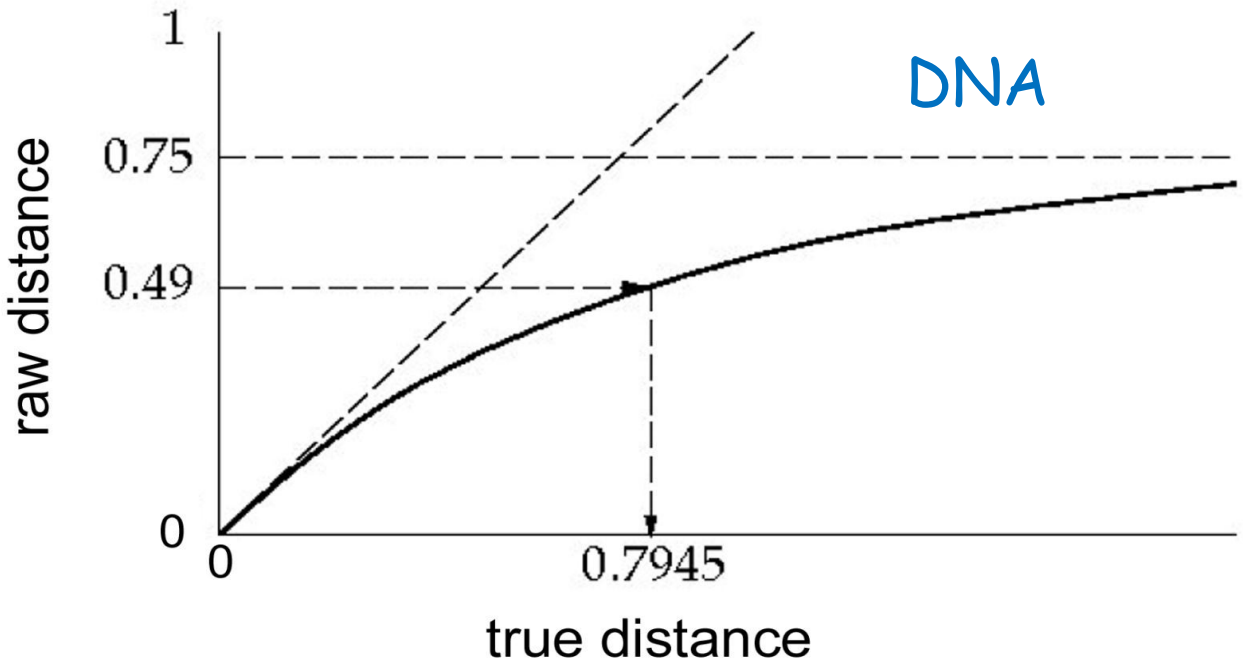


Raw distance correction

- As two DNA sequences diverge, it is easy to see that their maximum raw distance is ~0.75 (assuming equal nt frequencies, $\frac{1}{4}$ 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)

Mutational models for DNA

- Jukes-Cantor (JC) - all mutations equally likely.

- Kimura 2-parameter (K2P) - transitions and transversions have separate rates.

- Generalized Time Reversible (GTR) - all changes have separate rates.

(Models similar to GTR are also available for protein)

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.

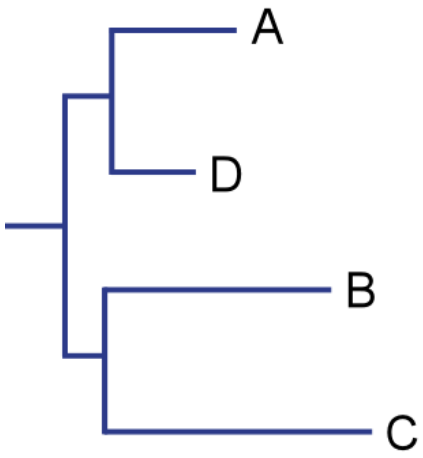
Trees and Phyton

Representing a tree in Python

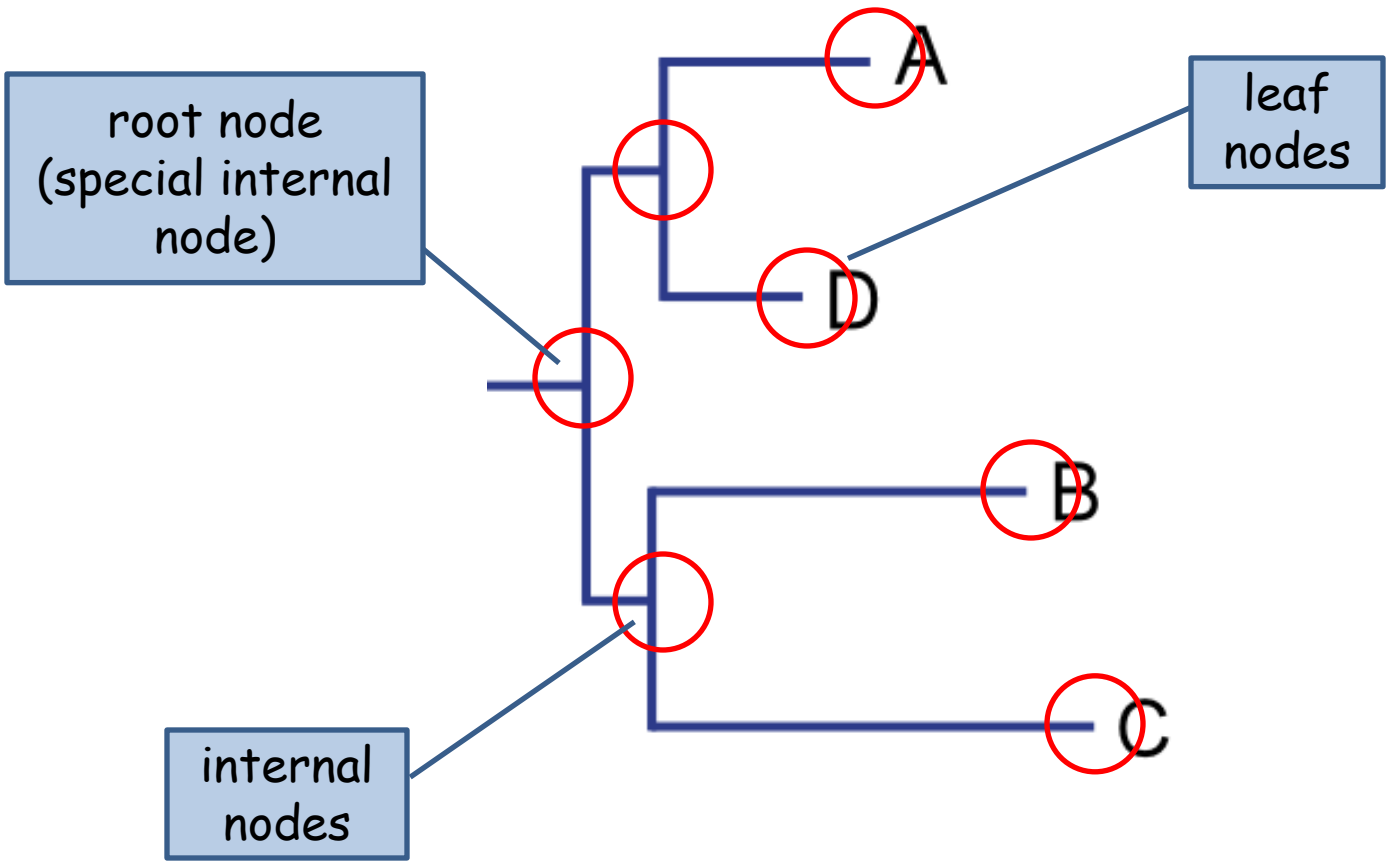
Some bioinformatic entities are easy to represent with standard Python types, e.g. :

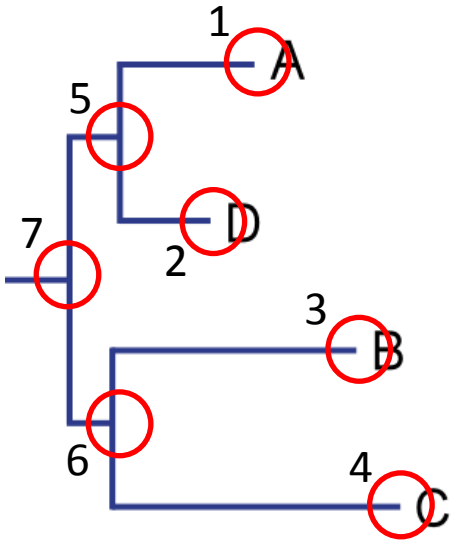
- Protein or DNA sequence
- Alignment score
- Sequence names paired with scores (or other things)

How would you represent a tree??



Natural approach - represent tree nodes





tree nodes numbered for reference

What kinds of information should we associate with nodes?

1) A sequence name (for leaf nodes)
2) A distance to parent (except for the root)
3) Connections to other nodes