

# Phylogenetics — some terminology

operational taxonomic unit (OTU)

- OTUs are the things being classified (e.g. species, genes)

homologous

- having a common ancestor, and therefore inherited similarity

analogous

- similar, but not due to a common ancestry

clade

- a grouping that includes a common ancestor and all descendents

phenotype

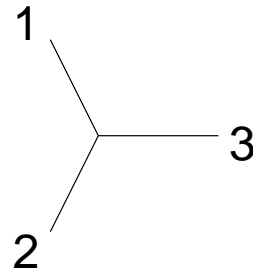
- an organism's observable characteristics

genotype

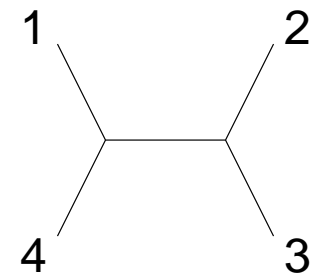
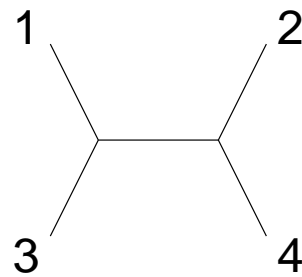
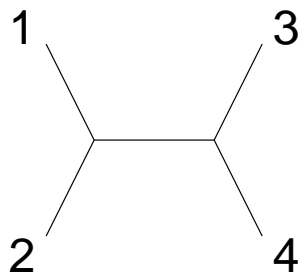
- an organism's genetic constitution

# Unrooted trees

3 nodes

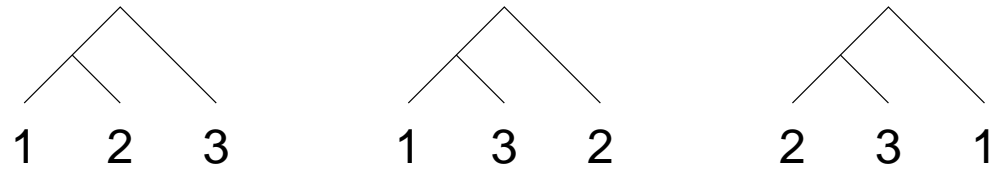


4 nodes

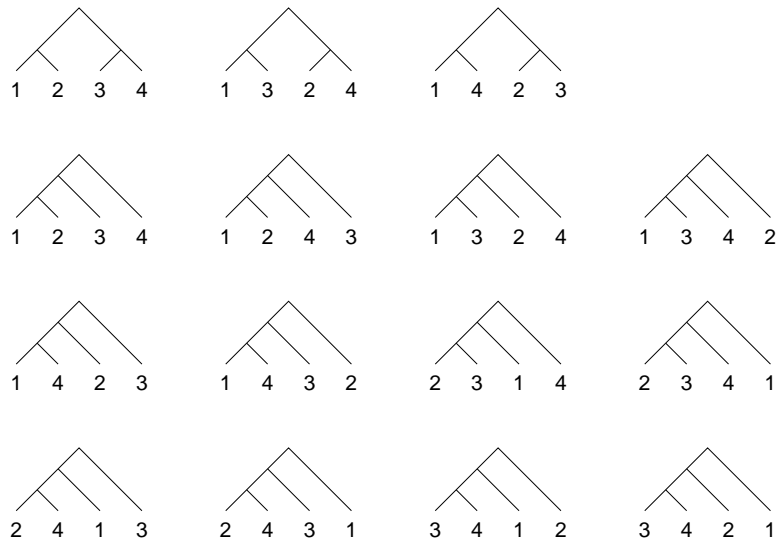


# Rooted trees

3 nodes

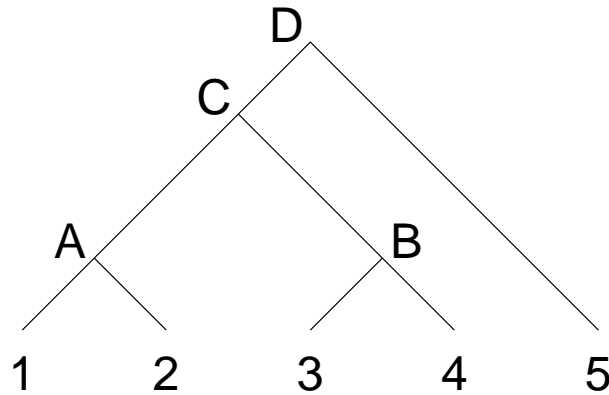


4 nodes



# Phylogenetic tree

A branching diagram that shows inferred evolutionary relationships



- Internal nodes represent “inferred ancestors”.
- Terminal nodes represent genes or organisms or species (OTUs).
- Newick format:  $((1,2),(3,4)),5$

# UPGMA

Unweighted-pair-group method with arithmetic mean

Oldest (early 1960s) and simplest method for tree reconstruction.

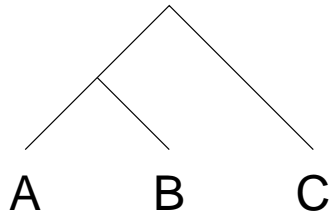
Species	A	B	C	D
B	9			
C	8	11		
D	12	15	10	
E	15	18	13	5

Species	B	AC
AC	10	
DE	16.5	12.5

Species	A	B	C
B	9		
C	8	11	
DE	13.5	16.5	11.5

# Estimating branch length (1)

Assuming rate of evolution to be constant in all lineages.

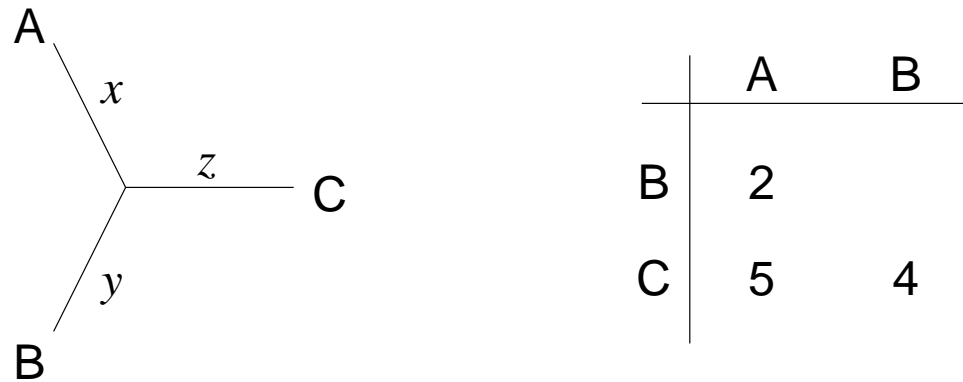


	A	B
B	2	
C	5	4

	AB
C	4.5

## Estimating branch length (2)

Not assuming constant rate of evolution.



$$x = (d_{AB} + d_{AC} - d_{BC})/2$$

$$y = (d_{AB} + d_{BC} - d_{AC})/2$$

$$z = (d_{AC} + d_{BC} - d_{AB})/2$$

## Neighbour-joining

- a) Start with a star network.
- b) A score matrix is computed in which scores are based on the distance between nodes  $i$  and  $j$ , and the distances between  $i$  and  $j$  and all other nodes.  
$$(n - 2) \text{ times the distance between } i \text{ and } j$$
$$\text{minus the sum of distances between } i \text{ and all nodes}$$
$$\text{minus the sum of distances between } j \text{ and all nodes}$$
- c) Find a pair with the lowest score, and join that pair with a new node.
- d) Compute distance from each node in the pair to the new node.
- e) Compute distance from all other OTUs to the new node.
- f) Repeat from step b, with the pair of joined nodes replaced by the new node.



## Neighbour-joining — advantages

- Fast. Practical for 100s or 1000s of OTUs.
- If input distances are correct, then output tree will be correct.
- Doesn't assume same rate of evolution in all lineages.

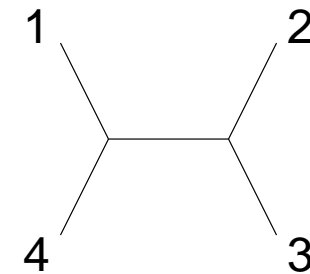
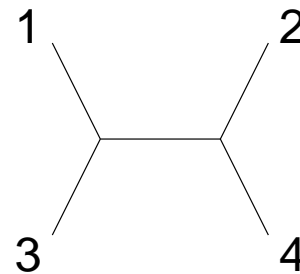
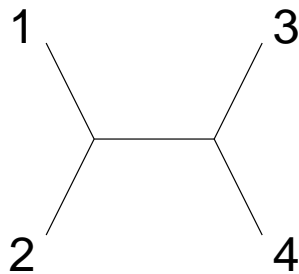
# Parsimony

- The quality of being careful with money or resources.
- In science, prefer the simplest explanation that fits the evidence.
- In phylogenetics, prefer the tree that represents the fewest mutational events.
- Inferred ancestral sequences can be obtained as a by-product of the parsimony approach.

# Informative and uninformative sites

To be informative, a position in a multiple alignment must have at least two different characters, each of which occurs at least twice.

Invariant sites are uninformative.



## Unweighted parsimony

- Consider every possible tree for every informative site in a multiple alignment.
- For each possible tree, add up the minimum number of mutations at each site.
- Tree with the fewest mutations is the most likely tree.

## Weighted parsimony

- e.g. transitions vs. transversions

## Sum of pairs score for a multiple sequence alignment

- could imagine generalising substitution matrix to N-dimensions, but is there good data to determine reliable scores?
- one alternative approach is to use the sum of pairs

Compute the sum of column scores, where each column score is:

$$\sum_{i < j} s(a_i, a_j)$$

where  $a_i$  and  $a_j$  are the residues in that column from sequences  $i$  and  $j$ , and  $s(x, y)$  is a score taken from a substitution matrix (e.g, from the BLOSUM or PAM families).

This score is simple to compute, but a drawback is that it assumes that all sequences in the set are separated by the same evolutionary distance.

# Multiple sequence alignment

## Dynamic programming

- in principle this could be done, using the sum of pairs approach for scoring matches/mismatches
- possible for a few short sequences
- not practical for many long sequences

## Progressive alignment

- perform pairwise alignment between all pairs of sequences
- construct a guide tree based on distances between each pair
- add sequences into the multiple alignment in the order given by the guide tree
- “once a gap, always a gap”

# Progressive alignment

Sequence 1: MGLPKSFVSM  
 Sequence 2: MGVPKTFVSM  
 Sequence 3: MGVPKTFVASM  
 Sequence 4: MGGLPKSYAVSM

1: MGLPKSFVSM  
 || || |||| (2)  
 2: MGVPKTFVSM

1: MGLPKSFV-SM  
 || || || || (3)  
 3: MGVPKTFVASM

1: M-GLPKS-FVSM  
 | |||| | || (3)  
 4: MGGLPKSYAVSM

2: MGVPKTFV-SM  
 ||||| || (1)  
 3: MGVPKTFVASM

2: M-GVPK-TFVSM  
 | | | | (5)  
 4: MGGLPKSYAVSM

3: M-GVPKTFVASM  
 | | | | (5)  
 4: MGGLPKSYAVSM

Sequence 1: M-GLPKSFV-SM  
 Sequence 2: M-GVPKTFV-SM  
 Sequence 3: M-GVPKTFVASM  
 Sequence 4: MGGLPKSYAVSM