Aalto University
School of Science

CS-E5865 Computational genomics

Autumn 2020, Lecture 7: Phylogenetic trees

Lecturer: Pekka Marttinen

Assistants: Alejandro Ponce de León, Zeinab Yousefi, Onur Poyraz

Evolutionary studies

- Big genome sequencing projects produce huge amounts of data
  - How to use these data?
- Evolutionary history relates all organisms and genes, and helps us understand and predict
  - interactions between genes (genetic networks)
  - drug design
  - predicting functions of genes
  - influenza vaccine development
  - origins and spread of disease
  - origins and migrations of humans

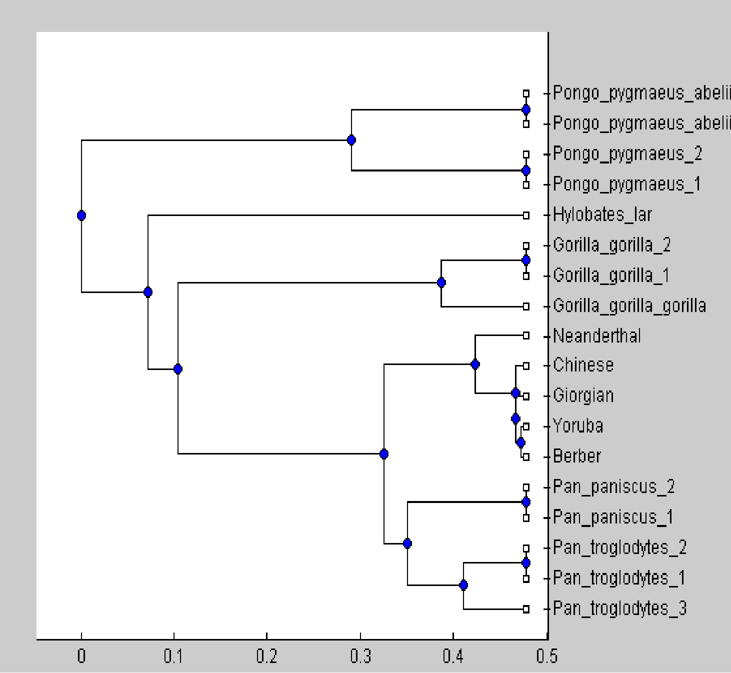
Phylogenetic analysis

- Starting point: a set of homologous, aligned DNA or protein sequences

- Result of the process: a tree describing evolutionary relationships between the sequences, i.e., a phylogenetic tree

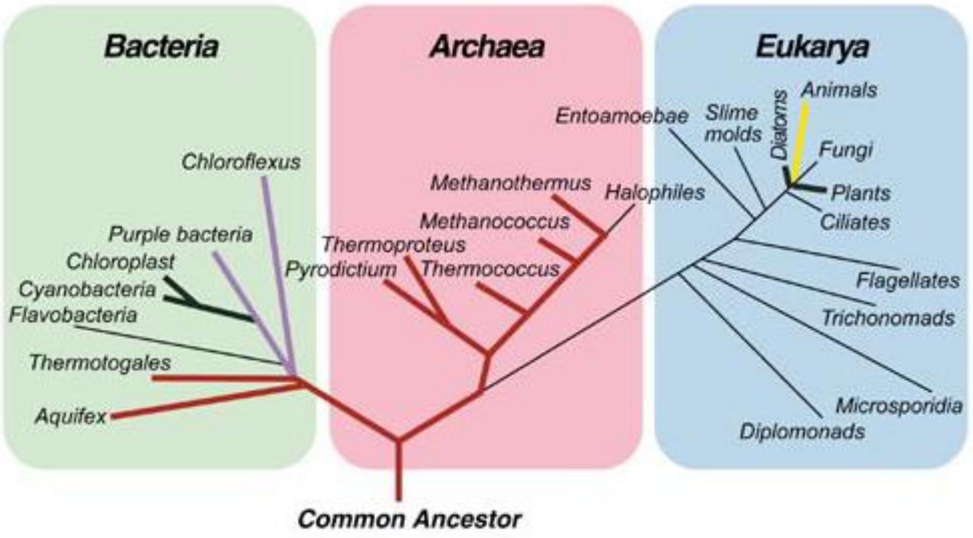
Phylogenetic trees

- A phylogenetic tree shows the evolutionary interrelationships among various species or individuals that have a common ancestor.
- Each node in a phylogenetic tree is called a taxonomic unit or taxon (plural taxa).
  - Internal nodes are generally referred to as Hypothetical Taxonomic Units (HTUs) as they cannot be directly observed.
  - Leaves or external nodes represent present (or extant) species.
- Branches (or edges) between nodes denote ancestor relations, and edge lengths correspond to time estimates.



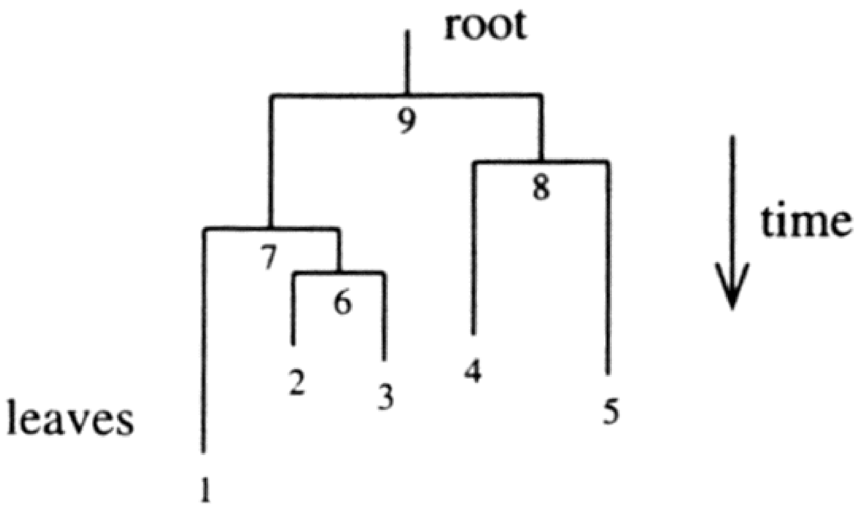
Example: tree of life

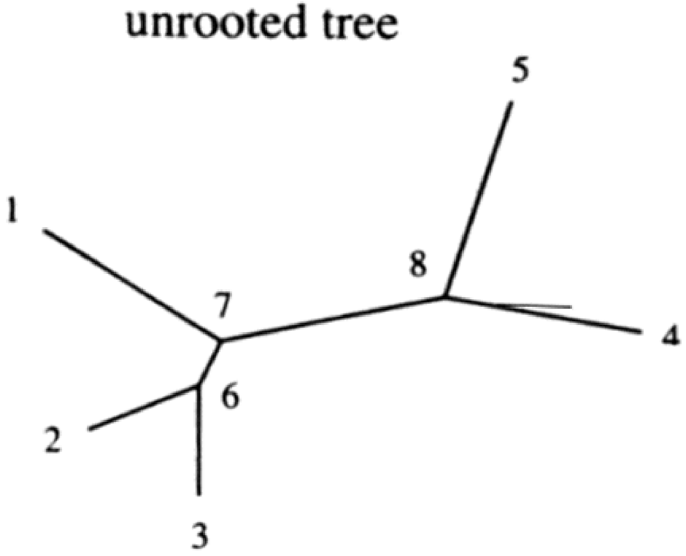
- Phylogenetic tree of living things, based on RNA data, shows the separation of bacteria, archaea, and eukaryotes.
- This tree is referred to as the tree of life or the universal tree.



Rooted and unrooted trees

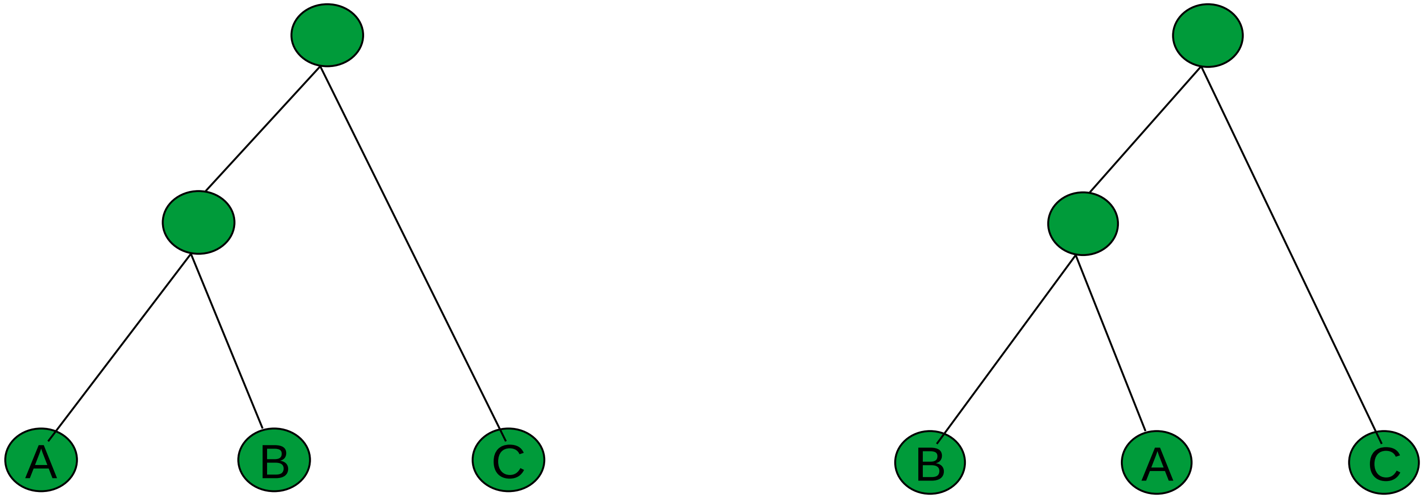
- A root of a tree is a node that does not have parents
  - represents the common ancestors of all taxa in the tree
  - generally requires adding an “outgroup” to the analysis, a species that is known to be outside the taxa under analysis
- An unrooted tree only represents the relationships between species, with no notion of the direction of time





Rotation invariance

- Any rotation of the internal branches of a tree keeps the the phylogenetic relations intact
- In other words: there is no information in the order of the child nodes of any internal node



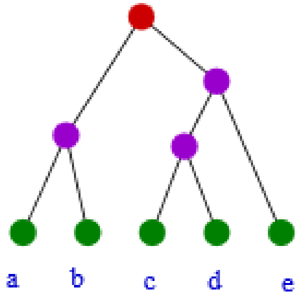
Phylogenetic tree reconstruction

- Input:
  - A set of n species
  - A method for computing a score for a labeled tree with n leaves
- Output:
  - The labeled phylogenetic tree with the optimal score

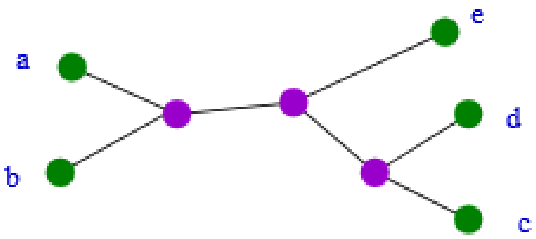
- Question: Should we solve this problem by enumerating and evaluating all trees with n leaves?
- Answer: No! Enumerating all trees with n leaves becomes computationally unfeasible even for n relatively small (e.g., 10-20).

Counting rooted and unrooted trees

- Rooted Trees
  - A rooted binary tree with n leaves has 2n-2 edges and n-1 internal nodes

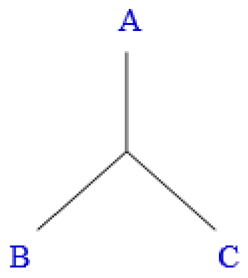


- Unrooted Trees
  - An unrooted binary tree (think of the root and its two edges combining to become a single edge) with n leaves has 2n-3 edges and n-2 internal nodes

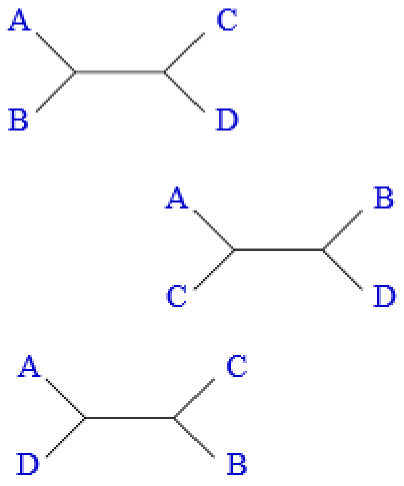


Counting unrooted trees

- If there are 3 labeled leaves then there is just one possible unrooted tree



- If there are 4 labeled leaves there are 3 different unrooted trees



Counting unrooted trees

- Let U(n) be the number of unrooted trees with n leaves
- Given an unrooted tree with n leaves, an extra leaf can be added on any branch to make a tree with (n+1) leaves
- n leaves ⇒ 2n-3 possible branches ⇒
- $U(n+1) = (2n-3)U(n)$
- $U(n) = (2n-5)!!$ (by induction)

n!! = n*(n-2)*...*3*1 is a double factorial multiplying every other (odd) number in the sequence

# Taxa (N)	# Unrooted trees
3	1
4	3
5	15
6	105
7	945
8	10,935
9	135,135
10	2,027,025
.	.
.	.
.	.
.	.
30	$\approx 3.58 \times 10^{36}$

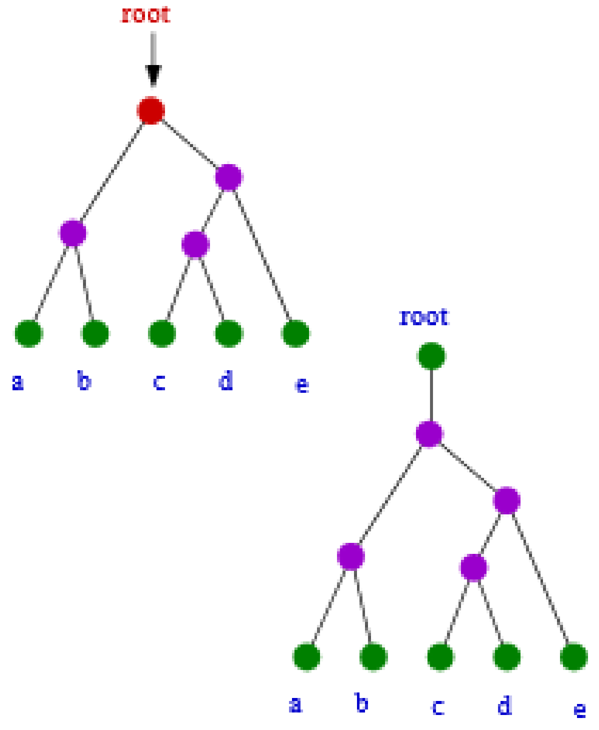
Counting rooted trees

- The root is a special node

- If we want to though, we can look at it as just another leaf (labeled root)

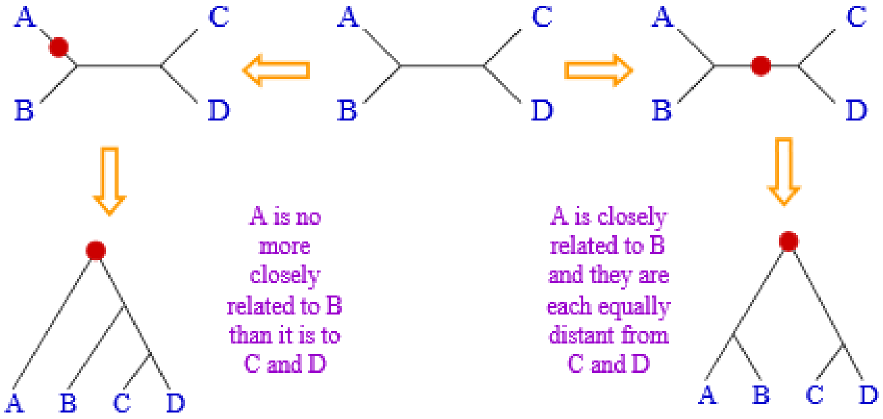
- A rooted tree with n leaves corresponds to an unrooted tree with n+1 leaves

- Thus there are (2n-3)!! rooted trees with n leaves



Rooted vs unrooted trees

- Usually we want rooted trees
- A single unrooted tree can imply different relationships between species depending on the location of the root

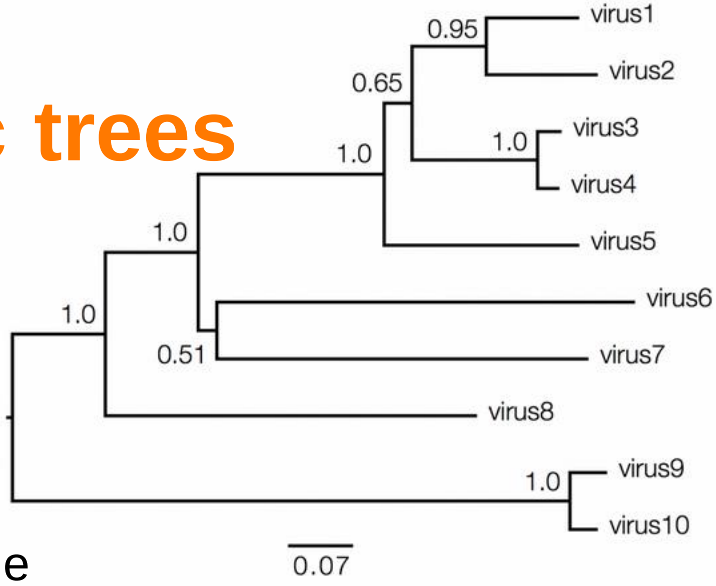


A is no more closely related to B than it is to C and D

A is closely related to B and they are each equally distant from C and D

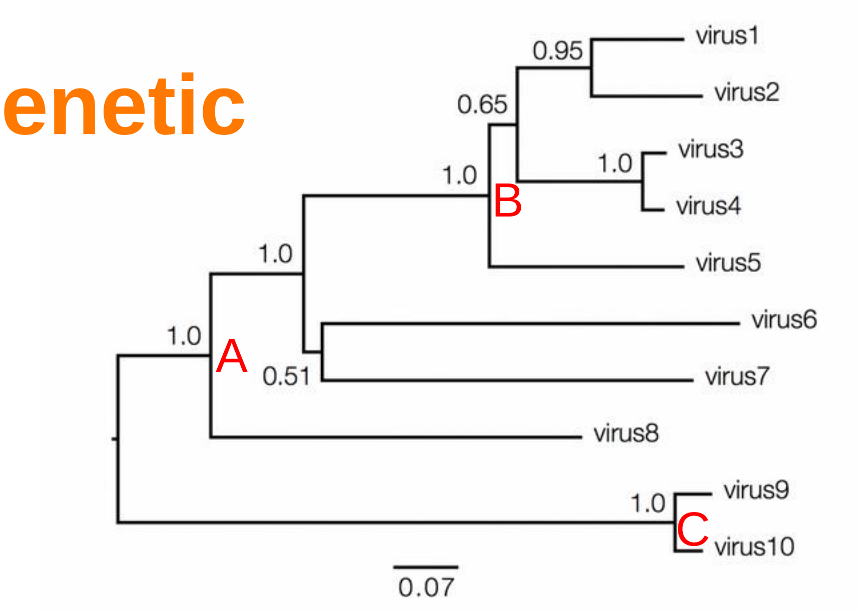
Aalto University
School of Science

Information in phylogenetic trees



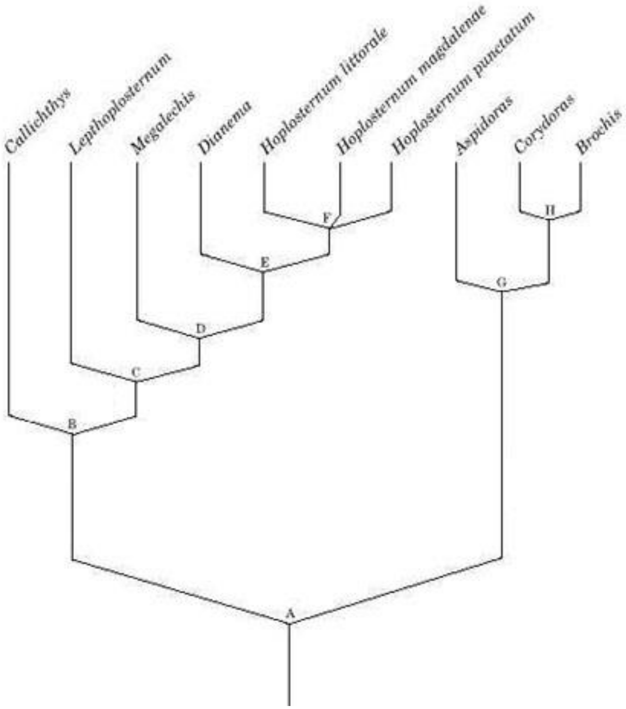
- Branches represent evolutionary lineages changing over time:
  - longer branches→ bigger changes
  - bar at the bottom of the figure provides a scale
  - The unit of branch length is usually either time or nucleotide substitutions per site: in the picture '0.07‘ shows the length of the branch that corresponds to a genetic change of 0.07.
- Nodes
  - external nodes ('leaves‘) represent the species sampled and sequenced
  - internal nodes represent putative ancestors
  - numbers next to each internal node represent a measure of support for the node; between 0 and 1: high values indicate a strong evidence that the sequences to the right of the node cluster together.

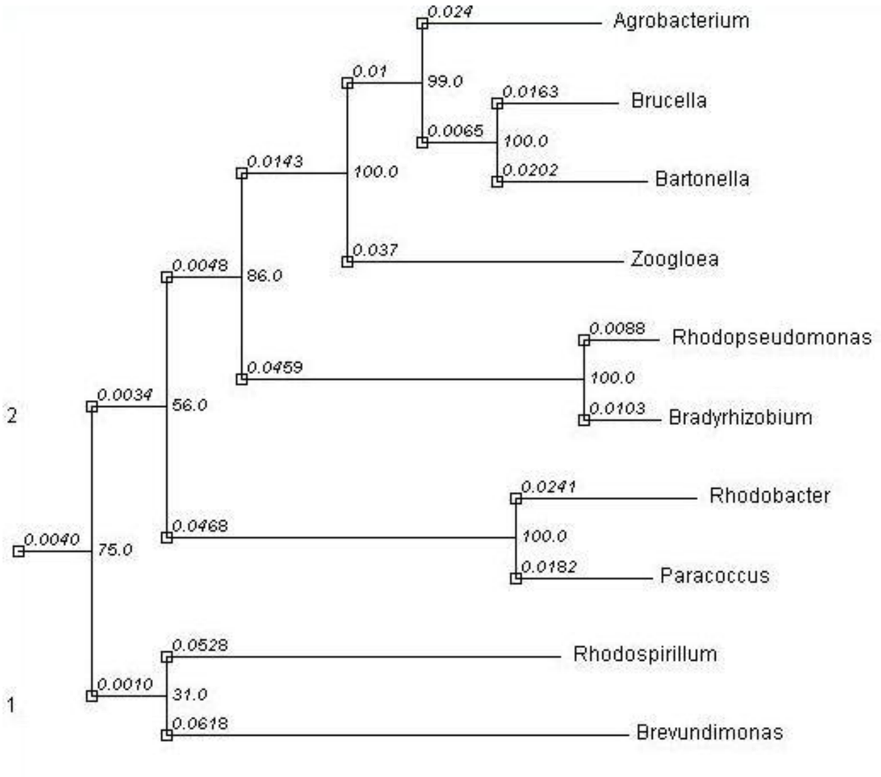
Information in phylogenetic trees



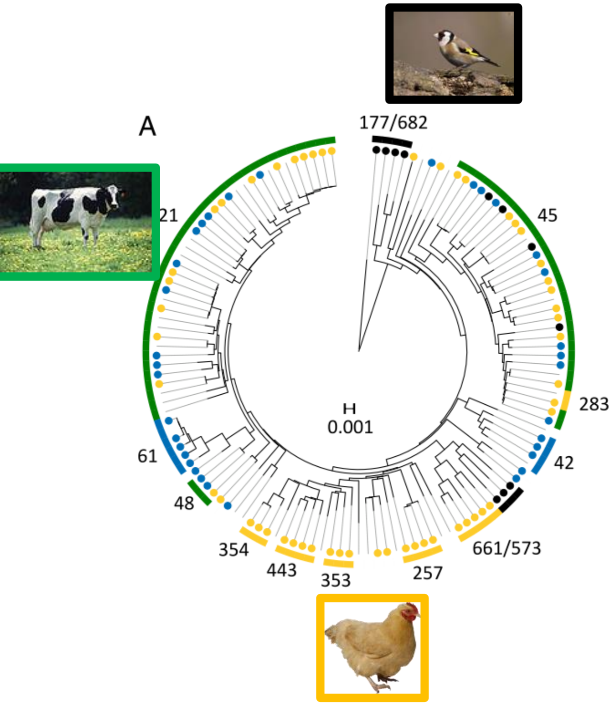
- In this picture:
  - Internal nodes (ancestors) = infected hosts sometime in the past that in turn infected 2 or more new hosts
  - Branches = chains of the epidemic that lead to the sampled viruses
  - Root = the common ancestor of all the viruses
  - The tree shows an ordering of branching events in the horizontal dimension: Ancestor 'A' existed before ancestors 'B' and 'C' (time flows from left to right).

Trees with branch lengths and without

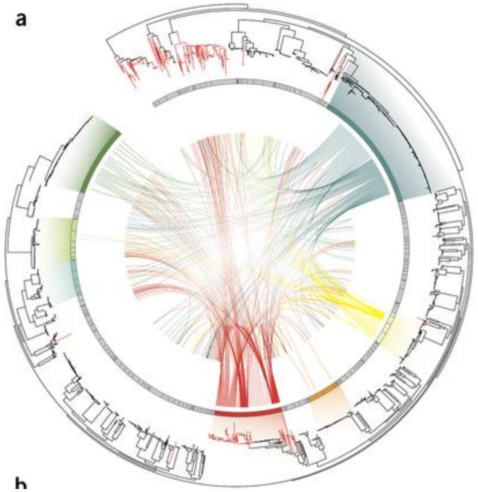




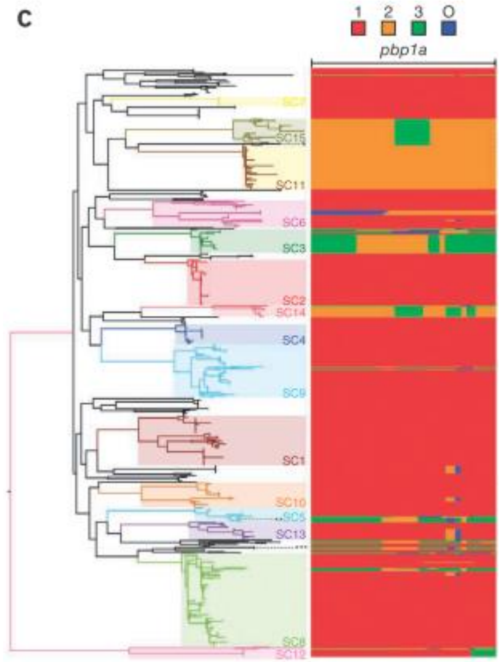
Different visualizations of phylogenies



Sheppard et al. 2013, PNAS
(modified)



Chewapreecha et al. 2014, Nature Genetics



Croucher et al. 2013, Nature Genetics

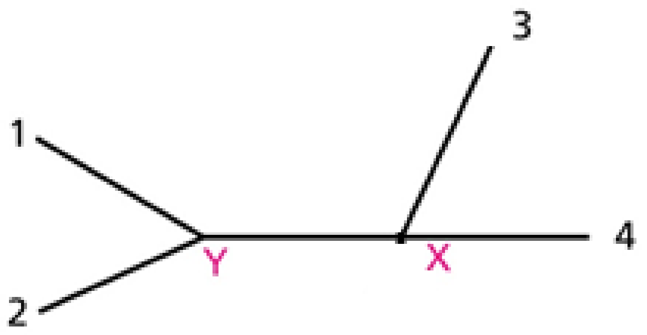
Phylogenetic tree reconstruction methods

- Methods based on the sequences themselves:
  - Parsimony-based methods: find a phylogenetic tree that explains the data with as few evolutionary changes as possible.
  - Probabilistic methods: find a tree that maximizes the probability of the genetic data given the tree.

- Methods based on distances between the sequences:
  - find a tree such that total branch lengths of paths between sequences (species) match the matrix of pairwise distances between sequences.

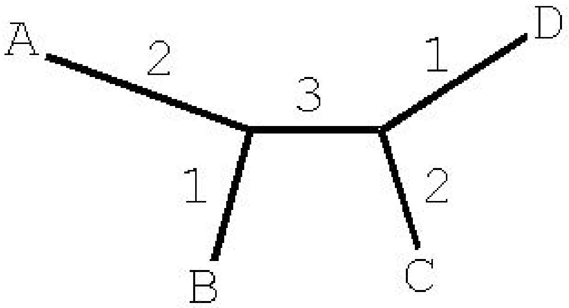
Inferring trees by Neighbor-joining

- A distance-based approach
- Assume we have
  - n taxa $\{t_1,\ldots,t_n\}$
  - matrix D of pairwise genetic distances (pairwise differences + Jukes-Cantor-correction)
- Neigbors are defined as leaves of a tree which are connected to the same node
  - 1 and 2 are neighbors
  - 3 and 4 are neighbors



Inferring trees by Neighbor-joining

- Additive tree of a distance matrix:
  - Tree T is an **additive tree** of D if for every pair of nodes (i,j), D(i,j) is the length of the path connecting i and j in T
  - D may not always have an additive tree (some path lengths may not be exactly correct)
  - Jukes-Cantor correction makes distances “more additive”
- Total branch length of a tree: sum of all branch lengths.

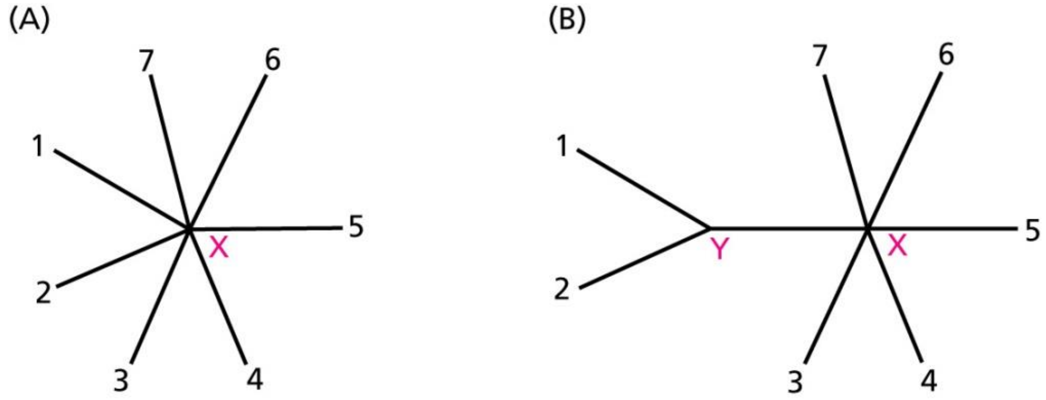


	A	B	C	D
A	0	3	7	6
B		0	6	5
C			0	3
D				0

http://slideplayer.com/slide/5215606/

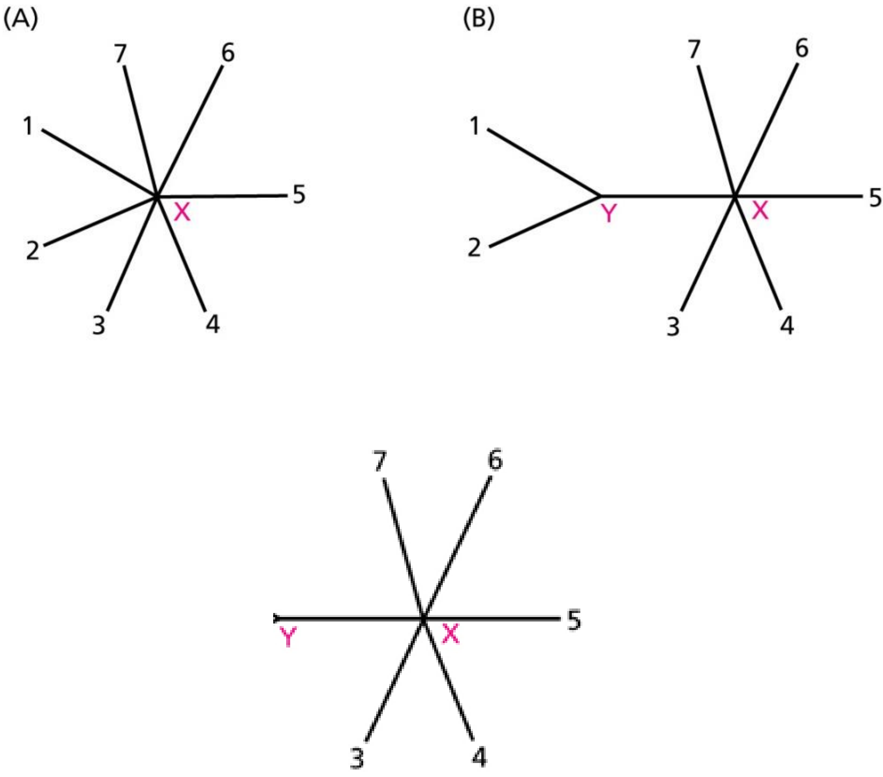
Neighbor-joining: the principle

- Start with an initial star tree (all taxa directly connected to the root X)
- For each pair of nodes, create a new node (Y) connected to both of the original nodes
  - Calculate the total length in this new "tree"
- Identify the pair yielding a tree with the shortest total branch length
  - This pair of sequences will be neighbours in the final tree



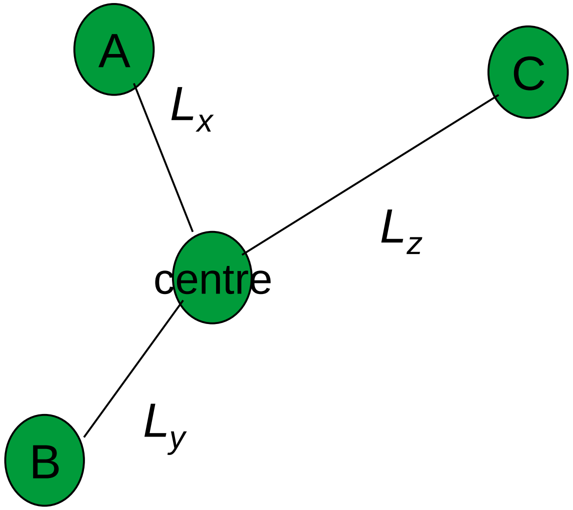
Neighbor-joining: the principle

- For the pair of nodes corresponding to the shortest total length
  - Create a new node Y
  - Connect the original nodes to Y
- Consider a new star tree with one node fewer than the original one
  - The 2 nodes identified earlier are removed.
- Repeat from the beginning, until only two nodes are left. Connect those and you have the final tree.



Finding Branch lengths: three nodes

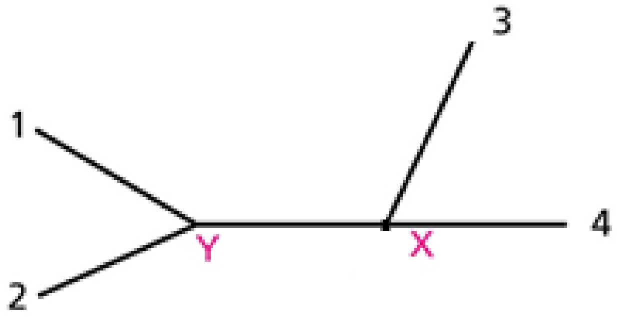
- The branch lengths in an unrooted tree with 3 external nodes can be computed from pairwise distances for additive matrices
- Three-point formula:
  - $L_x + L_y = d_{AB}$
  - $L_x + L_z = d_{AC}$
  - $L_y + L_z = d_{BC}$
- The solution gives the branch lengths:
  - $L_x = (d_{AB}+d_{AC}-d_{BC})/2$
  - $L_y = (d_{AB}+d_{BC}-d_{AC})/2$
  - $L_z = (d_{AC}+d_{BC}-d_{AB})/2$
- This way, we can infer the individual branch lengths in a tree from the pairwise distances



Finding a pair of nodes to merge

- If nodes 1 and 2 are neighbors, their distances satisfy the four-point formula (for any nodes 3 and 4):

  $$d(1,2) + d(3,4) < d(1,4) + d(2,3)$$

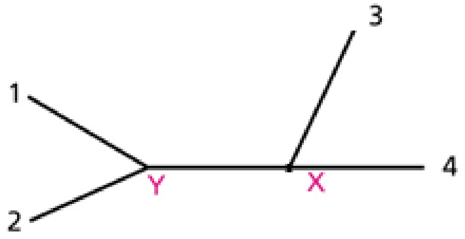


- In other words: the sum of distances between 1, 2, 3, and 4 is minimized when neighbors are paired in the summation
- This can be used to devise a criterion for detecting neighbors

Finding a pair of nodes to merge

- Compute the total distance from a given node i to all other nodes

$$R_i = \sum_j d(i,j)$$



- Define a 'neighborliness' measure

$$M(i,j) = (n-2)d(i,j) - R_i - R_j$$

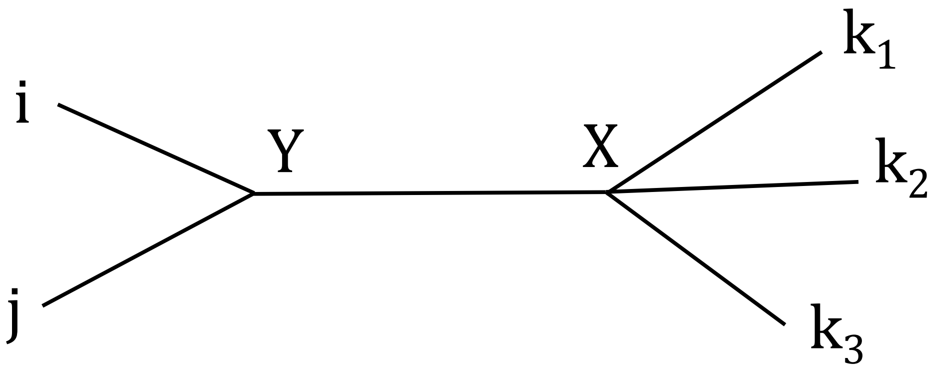
- $M(i,j)$ is small when the distance from other nodes R_i+R_j is large and $d(i,j)$ small
  - Nodes i and j that are close to each other and far from other nodes.
- Merging criterion: choose a pair of nodes (i,j) that minimizes $M(i,j)$
  - It can be shown that this yields a tree with the smallest total length.

Aalto University
School of Science

Joining the nodes in the tree

- Create a new parent node Y for i and j.
- Compute distances $d(Y,k)$ to all remaining nodes k

$$d(Y,k) = \tfrac{1}{2}(d(j,k)+d(i,k)-d(i,j)).$$



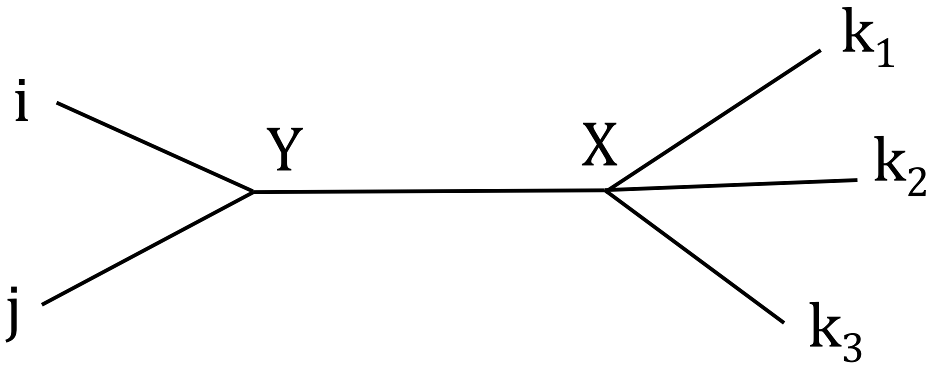
Joining the nodes in the tree

- Compute the lengths of the new branches from Y to i and j using the 3-point formula

$$\mathrm{L(i,Y)} = \frac{\mathrm{d(i,j)}}{2} + \frac{1}{2}\left(\frac{1}{\mathrm{n}-2}\mathrm{R_i} - \frac{1}{\mathrm{n}-2}\mathrm{R_j}\right)$$

$$\mathrm{L(j,Y)} = \frac{\mathrm{d(i,j)}}{2} + \frac{1}{2}\left(\frac{1}{\mathrm{n}-2}\mathrm{R_j} - \frac{1}{\mathrm{n}-2}\mathrm{R_i}\right)$$

- Note: Two latter terms above are average path lengths from i and j respectively



NJ algorithm

- **Input**: $n \times n$ distance matrix D
- **Output**: unrooted phylogenetic tree T, with $T(t,1)$ and $T(t,2)$ denoting the children of t, and table of branch lengths $B(t)$ denoting the length of the branch towards the parent.

- **Step 1:** Calculate neighbor distances (M) between all pairs of nodes i and j using the formula: $M(i,j) = (n-2)d(i,j) - R_i - R_j$
  - Find the smallest value: these nodes are both close to each other and far from all others. Say these are nodes i and j.
- **Step 2:** Join the two nodes i and j to a new node Y
  - $T(Y,1) = i$ and $T(Y,2) = j$ and $B(i) = L(i,Y)$ and $B(j) = L(j,Y)$
  - compute branch lengths from i and j to Y using 3-point formula

$L(i,Y) = \frac{d(i,j)}{2} + \frac{1}{2}(\frac{1}{n-2}R_i - \frac{1}{n-2}R_j)$ and $L(j,Y) = \frac{d(i,j)}{2} + \frac{1}{2}(\frac{1}{n-2}R_j - \frac{1}{n-2}R_i)$

- **Step 3:** calculate the updated distance matrix D' where i and j are replaced by Y: $d(Y,k) = ½(d(j,k)+d(i,k)-d(i,j))$ for all the other nodes k.
- **Step 4:** The distance matrix D' now contains $n - 1$ nodes. If there are more than 2 nodes left, go to step 1. If two nodes are left join them by a branch of length $d(i,j)$.

Example

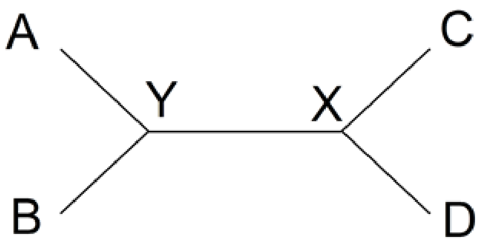
- Suppose we have only 4 taxa: A, B, C, and D.
- Step 1 is to calculate the neighbor distances M, using the equation on the previous slide.
  - $M(i,j) = (n-2)d(i,j) - R_i - R_j$
  - $R_i = \sum_j d(t_i, t_j)$
- -50 is the lowest score, and we could use either A-B or C-D. We arbitrarily choose A-B to join first.

dist	A	B	C	D
A	0	7	13	17
B	7	0	8	12
C	13	8	0	14
D	17	12	14	0

M		score
A-B	(4-2)*7 – (7+13+17) – (7+8+12)	-50
A-C	(4-2)*13 – (7+13+17) – (13+8+12)	-46
A-D	(4-2)*17 – (7+13+17) – (17+12+14)	-46
B-C	(4-2)*8 – (7+8+12) – (13+8+12)	-46
B-D	(4-2)*12– (7+8+12) – (17+12+14)	-46
C-D	(4-2)*14– (13+8+12) – (17+12+14)	-50

Example cont.

- We have created a new node Y, which joins A and B.
- Y is connected to X, which joins to all the other leaves.
- We calculate the distances of A and B to the new node Y with an equation different from the equation used for updating distances from Y to all the other leaf nodes.
- d(Y,C) = ½(d(B,C)+d(A,C)-d(A,B))
- $L(A,Y) = \frac{d(A,B)}{2} + \frac{1}{2}(\frac{1}{n-2}R_A - \frac{1}{n-2}R_B)$
- $L(B,Y) = \frac{d(A,B)}{2} + \frac{1}{2}(\frac{1}{n-2}R_B - \frac{1}{n-2}R_A)$



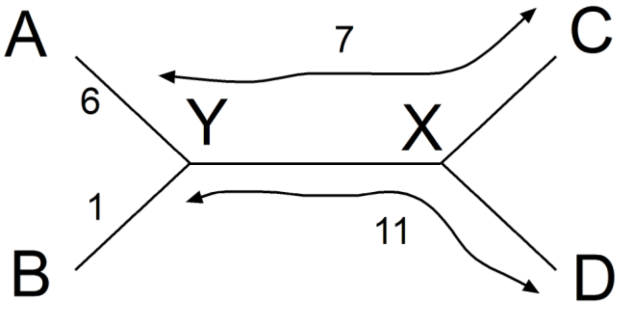
dist	A	B	C	D
A	0	7	13	17
B	7	0	8	12
C	13	8	0	14
D	17	12	14	0

Distances to new node Y		
A-Y	0.5*7 + 1/(2*(4-2))*[(7+13+17) – (7+8+12)]	6
B-Y	0.5*7 + 1/(2*(4-2))*[(7+8+12) – (7+13+17)]	1
C-Y	0.5*(8 +13– 7)	7
D-Y	0.5*(12 +17– 7)	11

Aalto University
School of Science

Example cont.

- Note that we don’t have distances C-X, D-X, or X-Y yet.
- We now have a new distance matrix, and we will repeat the process.

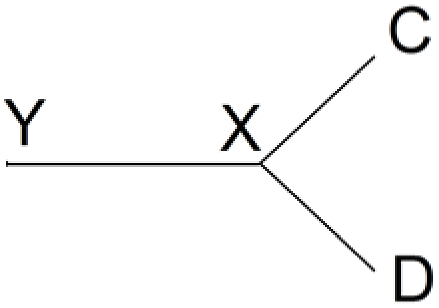


Distances to new node Y		
A-Y	0.5*7 + 1/(2*(4-2))*[(7+13+17) – (7+8+12)]	6
B-Y	0.5*7 + 1/(2*(4-2))*[(7+8+12) – (7+13+17)]	1
C-Y	0.5*(13 – 6) + 0.5*(8 – 1)	7
D-Y	0.5*(17– 6) + 0.5*(12 – 1)	11

Example cont.

- We now have 3 nodes to deal with: leaves C and D, and node Y.
- Now that A and B have been joined into Y, we ignore them.

dist	A	B	C	D
A	0	7	13	17
B	7	0	8	12
C	13	8	0	14
D	17	12	14	0



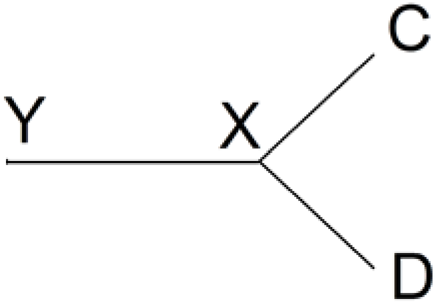
- The new distance matrix is:

Dist	Y	C	D
Y	0	7	11
C	7	0	14
D	11	14	0

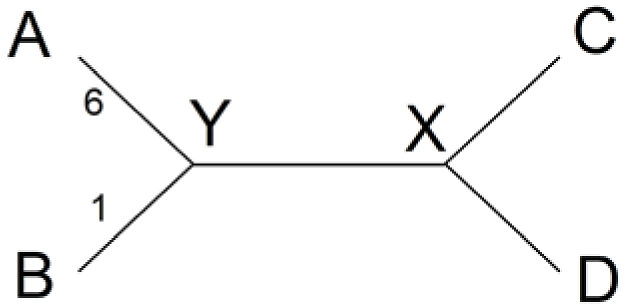
Distances to new node Y		
A-Y	0.5*7 + 1/(2*(4-2))*[(7+13+17) – (7+8+12)]	6
B-Y	0.5*7 + 1/(2*(4-2))*[(7+8+12) – (7+13+17)]	1
C-Y	0.5*(13 – 6) + 0.5*(8 – 1)	7
D-Y	0.5*(17– 6) + 0.5*(12 – 1)	11

Example cont.

- We again calculate M values
  - which all turn out to be the same,
- In the tree, we chose to join the taxa C and D into a new node X



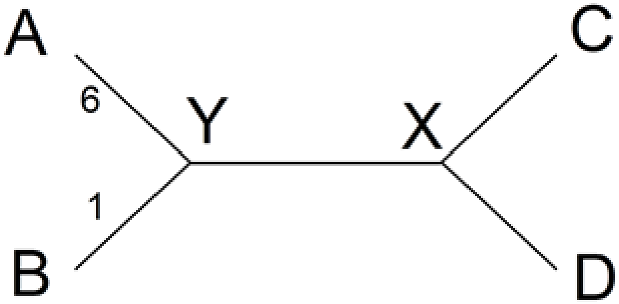
Dist	Y	C	D
Y	0	7	11
C	7	0	14
D	11	14	0



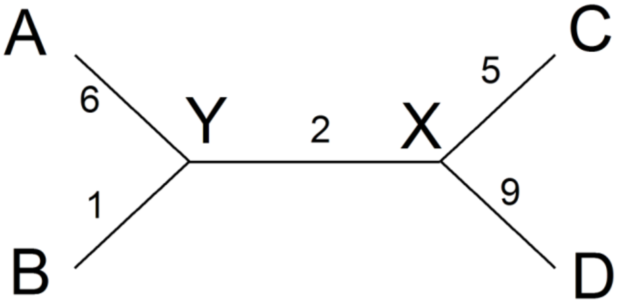
M		
Y-C	= (3-2)*7 – (7+11) – (7+14)	-32
Y-D	= (3-2)*11 – (7+11) – (11+14)	-32
C-D	= (3-2)*14 – (7+14) – (11+14)	-32

Example cont.

- We calculate distances, using different equations for C-X and D-X, and for X-Y
- $d(Y,X) = ½(d(Y,C)+d(Y,D)-d(C,D))$
- $L(X, C) = \frac{d(C,D)}{2} + \frac{1}{2}(\frac{1}{n-2}R_C - \frac{1}{n-2}R_D)$
- $L(X, D) = \frac{d(C,D)}{2} + \frac{1}{2}(\frac{1}{n-2}R_D - \frac{1}{n-2}R_C)$
- All branch lengths are now specified.



Dist	Y	C	D
Y	0	7	11
C	7	0	14
D	11	14	0



Distance to new node X		
C-X	0.5*14 + 0.5*1/(3-2)*[(7+14) – (11+14)]	5
D-X	0.5*14 + 0.5*1/(3-2)*[(11+14) – (7+14)]	9
X-Y	0.5*(7+11-14)	2

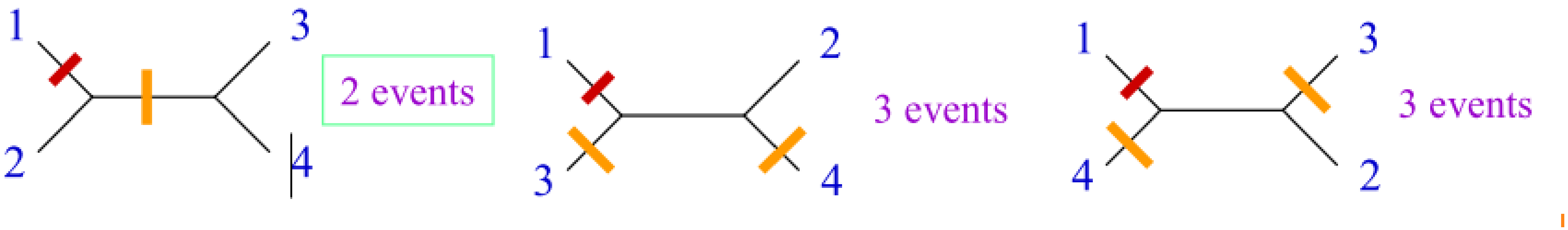
Aalto University
School of Science

+/- of distance methods

- Advantages:
  - easy to perform
  - quick calculation
  - fit for sequences having high similarity scores
- Disadvantages:
  - the sequences are not considered as such (loss of information)
  - all sites are generally equally treated (do not take into account differences of substitution rates)
  - not applicable to distantly divergent sequences.

Inferring trees – Parsimony Methods

- Basic idea: look at the aligned sequences and generate a tree that minimizes the number of mutations it takes to get from the common ancestor to the final sequences.
- Occam's razor principle – the simplest explanation is the best explanation
  - Assumes observed character differences resulted from the fewest possible mutations
- Example: 1: AC; 2: TC; 3: TG; 4: TG



Inferring trees – Parsimony Methods

- A tree is scored by counting the number of mutations that have occurred in it.

- Parsimony methods work directly on the aligned sequences and don't use a distance matrix or evolutionary model.

- One issue here: parsimony methods look specifically at individual sites with variation.
  - It completely ignores the possibility of multiple mutations that cancel each other out.

Inferring trees – Maximum Likelihood method

- Maximum likelihood method supposes a model M of evolution
  - we might use the BLOSUM or PAM matrices to indicate the likelihood of various substitutions
- Idea: Given a tree, we evaluate the probability that this tree is produced under the assumption that evolution operates according to model M
- The tree with the highest probability is assumed to be the correct one

Inferring trees – Maximum Likelihood method

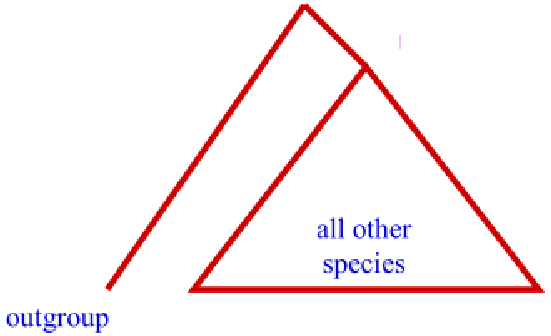
- Advantages:
  - Statistically well-justified
  - Relatively robust to sampling error
- Disadvantages:
  - Computationally expensive
  - Result depend on model of evolution

Pros & Cons

- Sequence based methods
  - Computationally expensive
  - Can create hypotheses about ancestral sequences
- Distance based methods
  - Character data can be converted to distance data, but information is lost
  - Generally faster

Rooting an Unrooted Tree

- Most of these methods produce unrooted rather than rooted trees
- One method for finding the root: include an outgroup
  - An outgroup is species known to have branched off before all the other species (e.g., use a bird as an outgroup for a mammalian tree)



- Another method: Choose midpoint of longest path between leaves