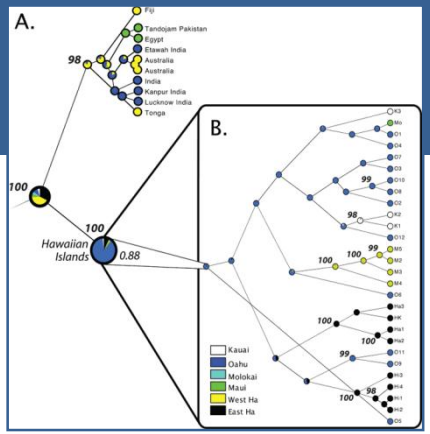
PHYLOGENY INFERENCE METHODS



Contents

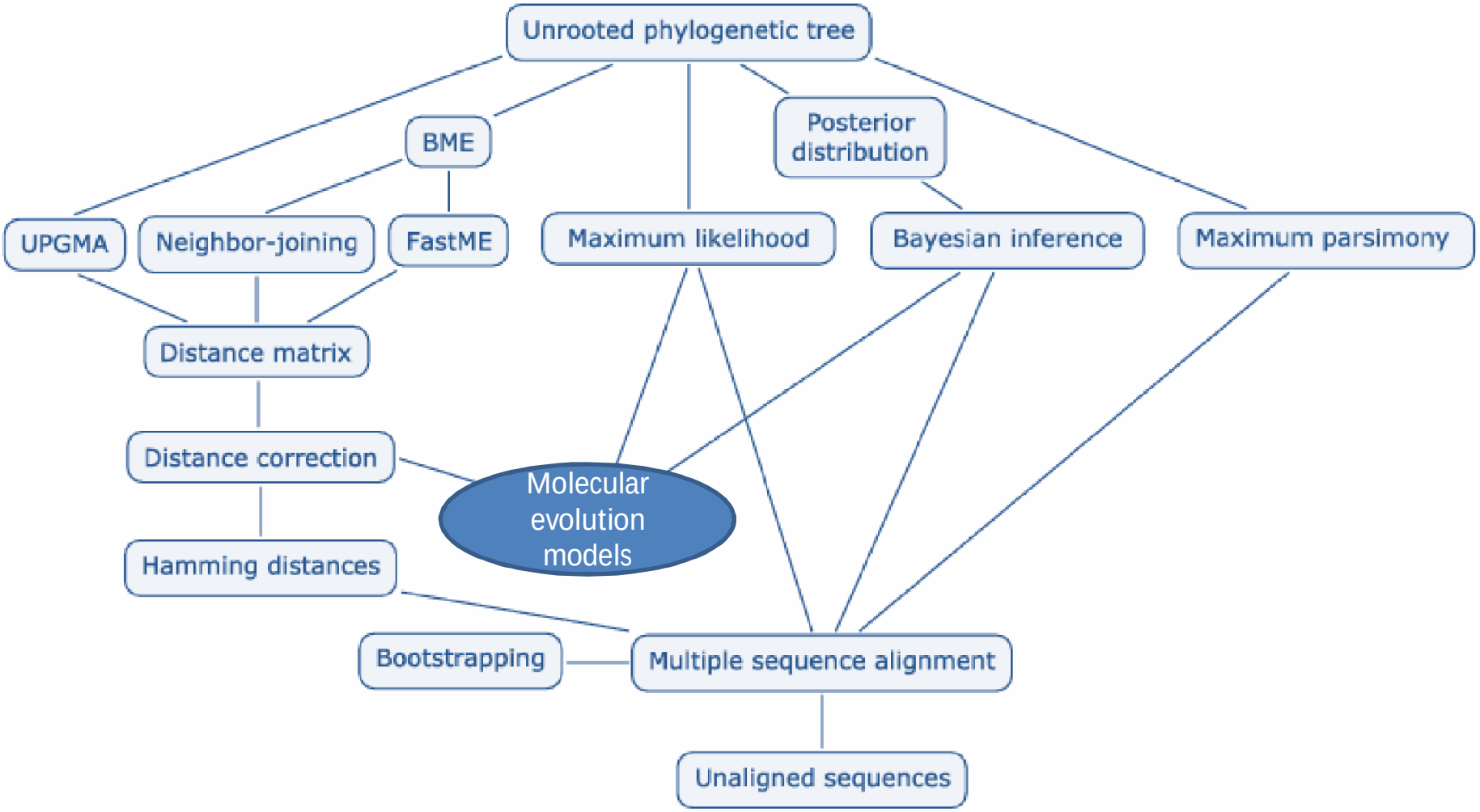
This is not a finalized version.

Pages 4-11 and 19-27 were in the contents of the course Biometry and bioinformatics I

http://evolution.genetics.washington.edu/phylip/software.html

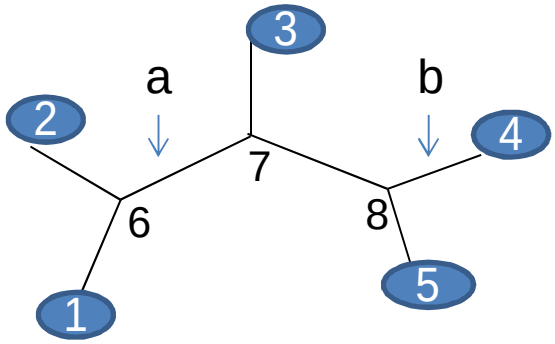
Collection of 392 phylogeny software-packages.
Maintained by Joe Felsenstein who is the author of the first package, PHYLIP, in 1970´s

ROLE OF MODELS IN PHYLOGENY RECONSTRUCTIONS



BASIC TERMINOLOGY

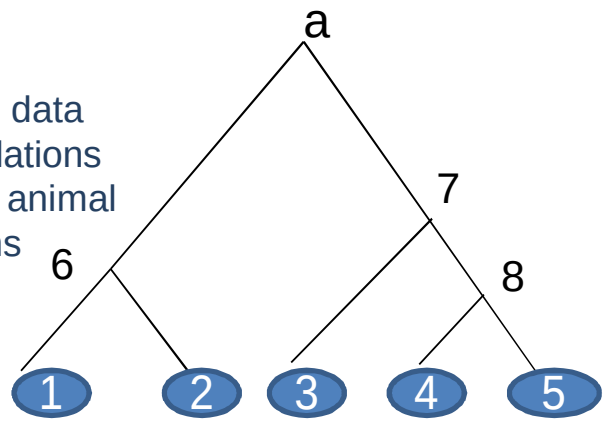
- **Leaves, external nodes** 1,2,3,4,5 are observations which may be, depending on the situation, sequences from different species, populations etc. They are often called **OTUs = Operational Taxonomic Units**. Internal nodes 6,7,8 are hypothetical sequences in ancestral units

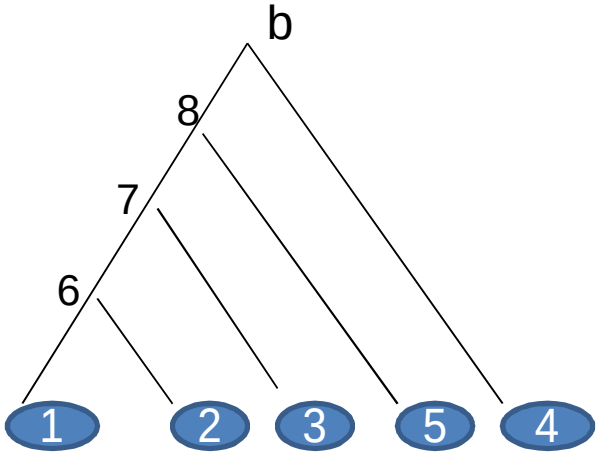


- The tree is **unrooted**.

- In case evidence exists for depicting the root (for example, a or b), a **rooted tree** can be constructed.

- For example, is there is data from different human populations and from chimpanzee, this animal is an outgroup and a means for rooting a tree





- Rooting requires external evidence and cannot be done on the basis of the data which is under a given study.

NUMBER OF POSSIBLE TOPOLOGIES

The number of unrooted trees

$B_n = (2(n-1)-3)b_{n-1} = (2n-5)b_{n-1} = (2n-5) * (2n-7) * \ldots * 3 * 1 =$

$(2n-5)! / ((n-3)!2^{n-3}),\ n > 2$

Number of rooted trees

$b'_n = (2n-3)b_n = (2n-3)! / ((n-2)!2^{n-2}),\quad n > 2$

that is, the number of unrooted trees times the number of branches in the trees

n	B_n	b'_n
3	1	3
4	3	15
5	15	105
6	105	945
7	954	10395
8	10395	135135
9	135135	2027025
10	2027025	34459425
20	2.22E+020	8.20E+021
30	8.69E+036	4.95E+038

MAXIMUM PARSIMONY IN PHYLOGENY INFERENCE

- Parsimony, **Occams razor**, a philosophical concept.

  Monk William of Ockham (1280-1350):

  "*Entitia non sunt multiplicanda praeter necessitate", entities should not be multiplied more than necessary,*

  "The best hypothesis is the one requiring the smallest number of assumptions"

- The principle of *maximum parsimony* (MP) in phylogeny inference involves the identification of a tree topology that requires the *smallest number of changes* to explain the observed differences. The shortest pathway leading to these is chosen as the best tree.

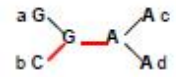
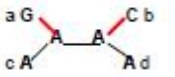
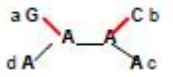
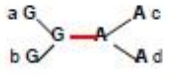
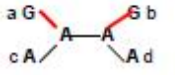
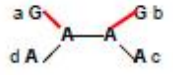
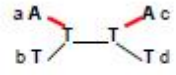
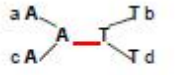
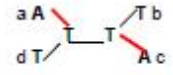
- Two subproblems:
  - Determining the amount of character change, or tree length, required by any given tree.
  - Searching over all possible tree topologies to find the tree that minimize this length.

INFORMATIVE AND UNINFORMATIVE SITES FOR PARSIMONY ANALYSIS

● An example, four OTUs (operational taxonomic units), nine sites

	1	2	3	4	5	6	7	8	9
OTU a	**A**	**A**	**G**	**A**	**G**	**T**	**T**	**C**	**A**
OTU b	**A**	**G**	**C**	**C**	**G**	**T**	**T**	**C**	**T**
OTU c	**A**	**G**	**A**	**T**	**A**	**T**	**C**	**C**	**A**
OTU d	**A**	**G**	**A**	**G**	**A**	**T**	**C**	**C**	**T**

Four OTUs can form three possible unrooted trees, I, II, III

	tree I ((a,b),(c,d))	tree II ((a,c),(b,d))	tree III ((a,d), (b,c))	**NEWICK-formats**
site 3				
site 5				
site 9				

A nucleotide site is informative only if it favors a subset of trees over the other possible trees. *Invariant* (1, 6, 8 in the example) and *uninformative* sites are not considered.

Variable sites:

Site 2 is uninformative because all three possible trees require 1 evolutionary change, G ->A.

Site 3 is uninformative because all trees require 2 changes.

Site 4 is uninformative because all trees require 3 changes.

Site 5 is informative because tree I requires one change, trees II and III require two changes

Site 7 is informative, like site 5

Site 9 is informative because tree II requires one change, trees I and III require two.

INFERRING THE MAXIMUM PARSIMONY TREE

● A site is informative only when there are at least two different kinds of nucleotides at the site (among the OTUs), each of which is represented in at least two OTUs.

● Identification of all informative sites and for each possible tree the minimum number of substitutions at each informative site is calculated:

- In the example for sites 5, 7 and 9:
  tree I requires 1, 1, and 2 changes
  tree II requires 2, 2, and 1 changes
  tree III requires 2, 2, and 2 changes.

● Summing the number of changes over all the informative sites for each possible tree and choosing the tree associated with the smallest number of changes: *Tree I is chosen because it requires 4 changes, II and III require 5 and 6 changes.*

● In the case of 4 OTUs an informative site can favor only one of the three possible alternative trees. For example, site 5 favors tree I over trees II and III, and is thus said to **support tree** I. **The tree supported by the largest number of informative sites is the most parsimonious tree**. In the cases where more than 4 OTUs are involved, an informative site may favor more than one tree and the maximum parsimony tree may not necessarily be the one supported by the largest number of informative sites.

FITCH´S PARSIMONY

- *The rule*:
  - The set at an interior node is the intersection of its two immediately descendant sets if the intersection is not empty.
  - Otherwise it is the union of the descendant sets.
  - For every occasion that a union is required to form the nodal set, a nucleotide substitution at this position must have occurred at some point during the evolution for this position. Thus, counting the number of unions gives the minimum number of substitutions required to account for descendant nucleotides from a common ancestor, given the phylogeny assumed at the outset.

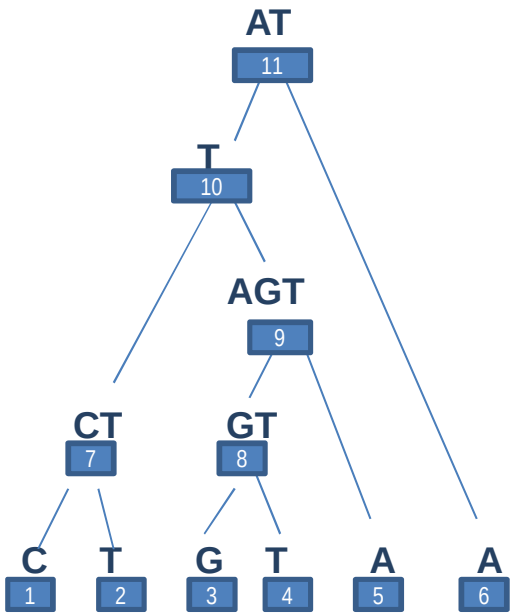
- The example next page (taken from textbook W-H Li, *Molecular evolution,* 1997) considers the case of six OTUs, and one particular *site*, at which the nucleotides are

....*site*.....

OTU 1	C
OTU 2	T
OTU 3	G
OTU 4	T
OTU 5	A
OTU 6	A

- The six OTU´s have five (unknown, to be inferred) ancestors: 7, 8, 9, 10, 11.

FITCH´S PARSIMONY, EXAMPLE



● One possible tree topology for the example site (previous page). The nucleotide at nodes 7, 8 and 9 cannot be determined uniquely under the parsimony rule. At node 10 T is chosen as it is shared by the sets at the two descendant nodes, 7 and 9. The nucleotide at node 11 cannot be determined uniquely. Parsimony requires it to be either A or T.

● At nodes 7, 8 and 10 nucleotide A could be included as a possible ancestral nucleotide because A is a possible common ancestral nucleotide (node 11) of all the six OTUs.

● **NEWICK-format,** the commonly agreed format for phylogeny topologies (not only parsimony), of the tree is **(((1,2) ((3,4) 5)) 6)**

● Consider other possible topologies for the example site. For example:

(((2,4) 1) (3 (5,6)))

Inferred nucleotides at nodes 7, 8, 9, 10 and 11 ?

FITCH´S PARSIMONY

- In the example tree (previous page), the nucleotide at node 10 is the intersection of the sets at nodes 7 and 9. The set at node 9 is the union of the sets at nodes 8 and 5.

- Counting the number of unions gives the minimum number of substitutions required to account for descendant nucleotides from a common ancestor, given the phylogeny assumed at the outset. In the example this number is 4.

    - There are many other alternative trees, each of which requires 3 substitutions. Thus, unlike the case of four OTUs, an informative site may favour many alternative trees.

PARSIMONY ANALYSES

● The total number of substitutions at both informative and uninformative sites in a particular tree is called the tree length. When the number of OTUs is small, it is possible to look at *all possible trees*, determine their length, and choose among them the shortest one(s) = *exhaustive search*. Large number of sequences (more than about 12) makes exhaustive searches impossible.

● Short-cut algorithms, for example ´**branch-and-bound**´: First an arbitrary tree is considered (or a tree obtained by another methods, for example some distance method), and compute the minimum number of substitutions for the this tree, which is considered as the "upper bound" to which the length of any other tree is compared. The rationale is that the maximum parsimony tree must be either equal in length to this tree *or shorter.*

● Above 20 sequences => heuristic searches are needed: only a manageable subset of all the possible trees is examined**. Branch swapping** (rearrangement) is used to generate topologically similar trees from a initial one**. Subtree pruning and regrafting** is one method.

THE LENGTH OF A GIVEN TREE

■ Inferring optimal trees under the parsimony criterion involves

(1) determining the amount of character change, or tree length, required by any given tree, and

(2) searching over all possible tree topologies for the trees that minimize this length.

■ For *n* OTUs, an unrooted binary tree (a fully bifurcating tree) contains *n* terminal nodes, *n* – 2 internal nodes, and 2*n* – 3 branches (edges) that join pairs of nodes.

■ The length of a particular tree topology (one tree chosen from the space of all possible trees) is the sum of sites in the sequence, a single site having a length on the basis of the amount of character change. *N* is the number of sites (characters) and l_j is the amount of character change implied by a most parsimonious reconstruction that assigns a character state x_{ij} to each node *i* for each site *j*. For terminal nodes the character state assignment is fixed by the input data.

■ In Fitch parsimony the *cost* associated with the change from state *x* to state *y* is simply 1 if *x* and *y* are different, 0 is they are identical.

THE LENGTH OF A TREE

■ A commonly used cost scheme is to assign a *greater cost to transversions than to transitions*. This means that the latter are accorded less weight.

■ The cost scheme is represented as a *cost matrix,* or *step matrix,* that assigns a cost for the change between each pair of character states. The cost matrix is usually symmetric ($c_{AG} = c_{GA}$) with the consequence that the length of the tree is the same regardless of the position of the root. If the cost matrix contains elements for which $c_{xy} \neq c_{yx}$, then different rootings of the tree may imply different lengths, and the search among trees must be done over rooted trees rather than unrooted trees.

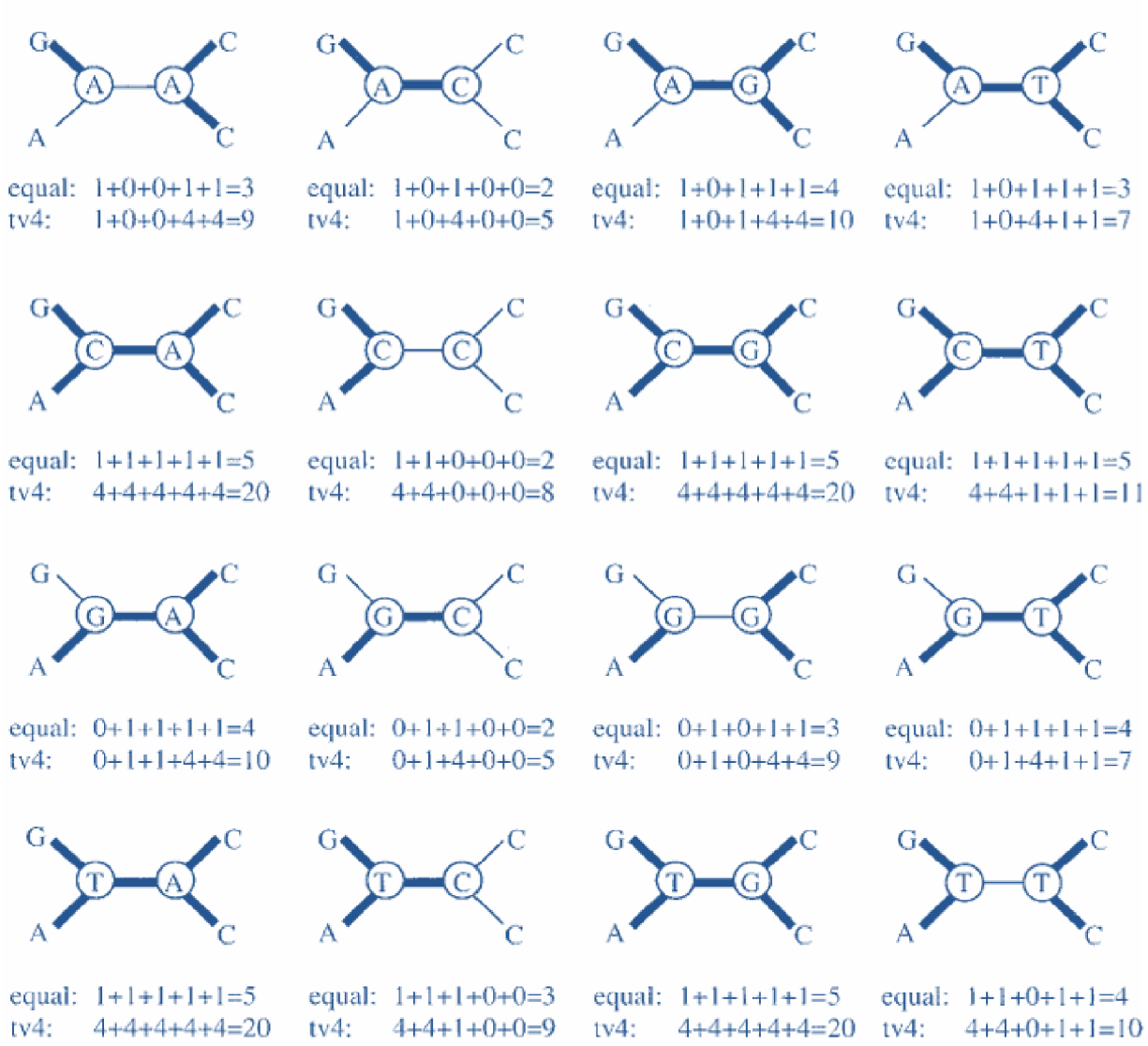
■ Next example (taken from Lemey *et al., The phylogenetic handbook, 2009*), calculation of tree length using "brute-force" approach of evaluating all possible character-state reconstructions. Four OTUs, W, X, Y and Z,

site
j

■ The tree ((W,Y) ,(X,Z)) is shown (next page).

```
W...ACAGGAT....
X...ACACGCT....
Y...GTAAGGT....
Z...GCACGAC....
```

INFERRING THE MAXIMUM PARSIMONY TREE - COST SCHEMES INCLUDED



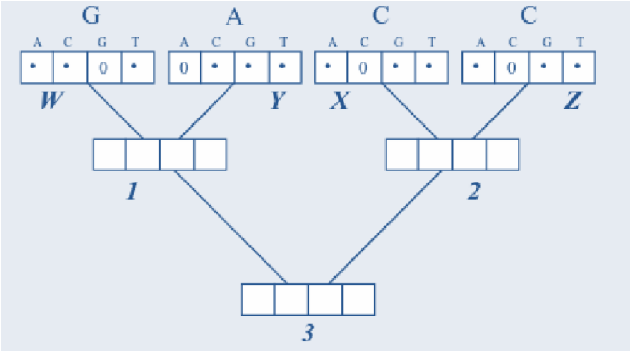
■ Two cost schemes, equal and transversions 4x weighted.

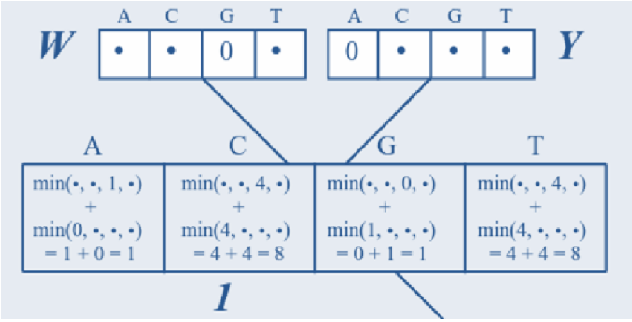
■ With equal costs, the minimum length is two steps and this length is achievable in three different ways, internal nodes assignment A-C, C-C and G-C. If a similar analysis for the other two possible trees, ((W,X),(Y,Z)) and ((W,Z),(Y,X)) is conducted, they are also found to have lengths of two steps. *Thus, this character (state) does not discriminate among three tree topologies and is parsimony-uninformative under this cost scheme.*

■ With 4:1 transversion:transition weighting the minimum length is five steps, achieved by two reconstructions, internal node assignments A-C and G-C. *Similar evaluation of the other two trees finds a minimum of eight steps on both trees (i.e. two transversions are required rather than one transition plus one transversion). The character thus becomes informative as some trees have lower lengths than others.*

SANKOFF´s ALGORITHM FOR CALCULATION OF MINIMUM TREE LENGTH

■ For symmetric cost matrixes an unrooted tree can be rooted arbitrarily to determine the minimum tree length. Then, for each node *i* , a conditional-length vector $\mathbf{S}_{ij}$, containing the minimum possible length above *i* is computed, given each of the possible state assignments to this node for character *j*.
■ Thus, s_{ik} is the minimum possible length of the subtree descending from node *i* if it is assigned state *k.*
■ For the tip sequences, this length is initialized to 0 for the state(s) actually observed in the data, or to infinity otherwise.
■ The algorithm proceeds by working from the tips toward the root, filling in the vector at each node based on the values assigned to the node´s children (i.e. immediate descendants).





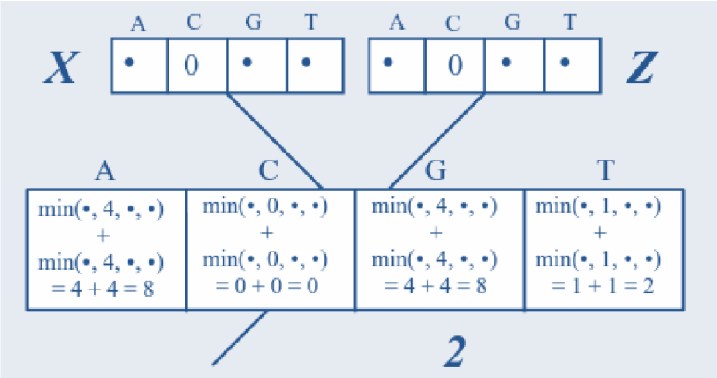
Node 1

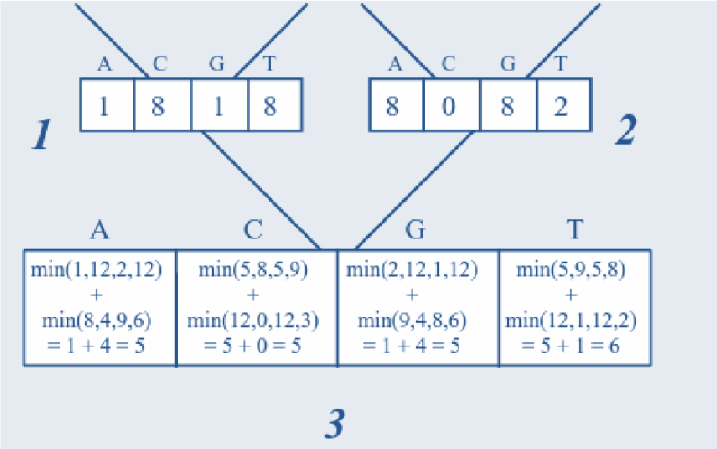
■ For each element *k* of this vector, consider the costs associated with each of the four possible assignments to each of the child nodes W and Y, and the cost needed to reach these states from state *k*, which is obtained from the cost matrix (4:1 transversion:transition is assumed here).

■ Calculation is trivial for nodes ancestral to two terminal nodes because only one state needs to be considered for each child. Thus, if state A is assigned to node 1, the minimum length of the subtree above node 1, given this assignment, is the cost of a change from A to G in the left branch, plus the cost of a (non-) change from A to A in the right branch: $s_{1A} = c_{AG} + c_{AA} = 1 + 0 = 1$. Similarly, s_{1C} is the sum of c_{CG} (left branch) and c_{CA} (right branch) = 8.

■ Continuing like this, the configuration for the subtree of node 1 is obtained.

SANKOFF´s ALGORITHM FOR CALCULATION OF MINIMUM TREE LENGTH





Nodes 2 and 3

■ Node 2 analogously (see node 1, previous page), but calculation for the root node 3 is a bit more complicated:
■ For each state *k* at this node, each of the four state assignments to each of the child nodes 1 and 2 must be considered.
■ For example, when calculating the length, conditional on the assignment of state A to node 3, for the left branch we consider in turn all four of the assignments to node 1.
■ If node 1 is assigned state A as well, the length would be the sum of 1 (for the length above node 1) plus 0 (for the non-change from state A to state A).
■ If instead state C is chosen for node 1, the length contributed by the left branch would be 8 (for the length above node 1) plus 4 (for the change from A to C).
■ The same procedure is used to determine the conditional lengths for the right branch.
■ By summing up these two values for each state *k*, the entire conditional-length vector for node 3 is obtained.
■ Since the root of the tree is now considered, the conditional-length vector $\mathbf{s}_3$ provides the minimum possible lengths for the full tree, given each of the four possible state assignments to the root. **The minimum of these values is the tree that is seeked. This length is 5** (cf. above the example using brute-force enumeration).

SANKOFF´s ALGORITHM - GENERATION OF ALL POSSIBLE TREES

■ Sankoff´s algorithm provides a means of calculating the length required by any character on any tree under any cost scheme. The length of a given tree is obtained by repeating the procedure for each character and summing up all characters. In principle, the most parsimonious tree is found by generating and evaluating all possible trees. However, this **exhaustive-search strategy is feasible only for a relatively small number of OTUs** (in practice, 11 is the maximum number for most (?) phylogeny programs.

■ Below (next page, taken from Lemey *et al., The phylogenetic handbook, 2009*) one procedure: The algorithm recursively adds the *t*th OTU in a stepwise fashion to all possible trees containing the first *t* – 1 OTUs until all *n* OTUs have been joined. For rooted trees the algorithm is modified by including one additional artificial OTU that locates the root of each tree. In this case, the first three trees generated represent each of the three possible rootings of an unrooted three-OTU tree, and the algorithm proceeds as in the unrooted case. Thus, the number of rooted trees for *n* OTUs is equal to the number of unrooted trees for *n* + 1 OTUs.

■ Six OTUs (A,B,C,D,E,F), start with A,B,C, the fourth, D, connected to each of the three branches, fifth, E, connected to each three trees → all 15 possible trees generated....→ all 105 possible trees generated and their lengths evaluated.

PHYLOGENY METHODS BASED ON DISTANCE MATRICES

● Distances are computed for all pairs of OTUs and a phylogenetic tree is constructed by considering the relationships among these distance values. Distances are numbers of nucleotide substitutions between sequences. Distances are simple p-distances or based on some nucleotide substitution model.

● The **unweighted pair-group method with arithmetic mean, UPGMA,** is the simplest method for tree construction. It was originally developed in 1950´s for constructing taxonomic **phenograms**, i.e. trees that reflect the phenotypic similarities between OTUs.

● The most widely used method is the **neighbor-joining, NJ,** algorithm, developed in 1987. NJ tree is usually the first tree constructed for a given research problem, followed by other methods, parsimony, maximum likelihood and bayesian trees.

DISTANCES

- Distance matrix D = (d_{ij}) gives pairwise distances for *leaves* of the phylogenetic tree

- In addition, the phylogenetic tree will now specify distances between leaves and internal nodes

- Distances d_{ij} in evolutionary context satisfy the following conditions:

  Symmetry: $d_{ij} = d_{ji}$ for each i, j
  Distinguishability: $d_{ij} \neq 0$ if and only if $i \neq j$
  Triangle inequality: $d_{ij} \leq d_{ik} + d_{kj}$ for each i, j, k
  Distances satisfying these conditions are called metric
  In addition, evolutionary mechanisms may impose additional constraints on the distances: *additive* and *ultrametric* distances

- A tree is called *additive*, if the distance between any pair of leaves (i, j) is the sum of the distances between the leaves and a node k on the shortest path from i to j in the tree

  $$d_{ij} = d_{ik} + d_{jk}$$

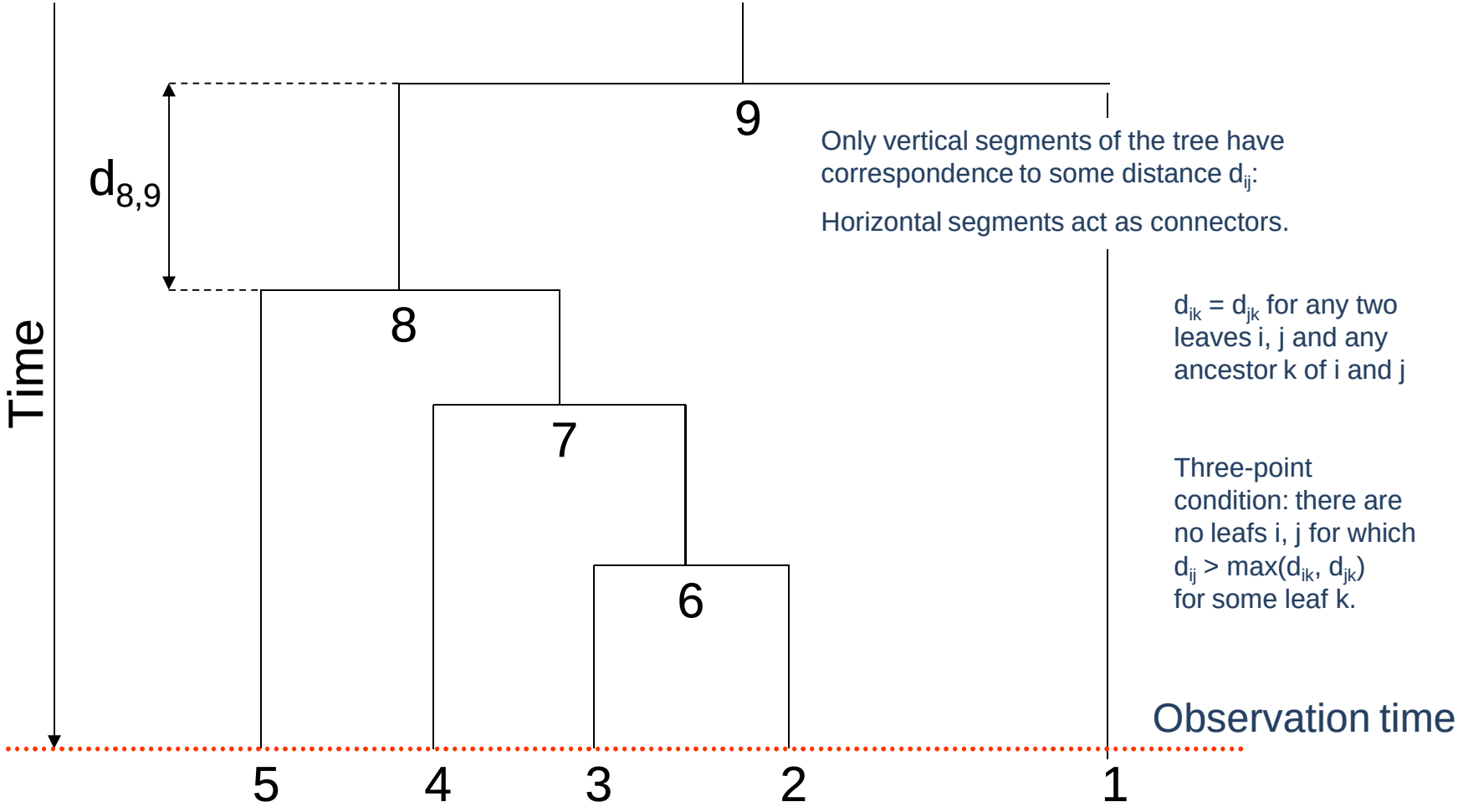
- A rooted additive tree is called an *ultrametric tree*, if the distances between any two leaves i and j, and their common ancestor k are equal

  $$d_{ik} = d_{jk}$$

- Edge length d_{ij} corresponds to the time elapsed since divergence of i and j from the common parent ,i.e. edge lengths are measured by a ”*molecular clock*” with a constant rate

ULTRAMETRIC TREE

Distances to be ultrametric can be found by the three-point condition:
D corresponds to an ultrametric tree if and only if for any three species (OTUs) i, j and k, the distances satisfy $d_{ij} \leq \max(d_{ik}, d_{kj})$



Only vertical segments of the tree have correspondence to some distance d_{ij}:

Horizontal segments act as connectors.

$d_{ik} = d_{jk}$ for any two leaves i, j and any ancestor k of i and j

Three-point condition: there are no leafs i, j for which $d_{ij} > \max(d_{ik}, d_{jk})$ for some leaf k.

UPGMA -method

- The UPGMA method employs a sequential clustering algorithm, in which local topological relationships are inferred in order of decreasing similarity and a phylogenetic tree is built in a stepwise manner.
  - The two OTUs that are most similar to each other, i.e. have the shortest distance, are first identified.
  - The two OTUs are treated as a new single OUT, a **composite OTU**
  - Then, from among the new group of OTUs, the pair with highest similarity is identified, and so on, until only two OTUs are left.

- Consider a case of four OTUs, A, B, C and D. The pairwise distances are given by the following matrix

	A	B	C
B	d_{AB}		
C	d_{AC}	d_{BC}	
D	d_{AD}	d_{BD}	d_{CD}

- Let us assume that d_{AB} has the smallest value. Then, A and B are the first to be clustered, and the branching point is positioned at a distance of $d_{AB}/2$.

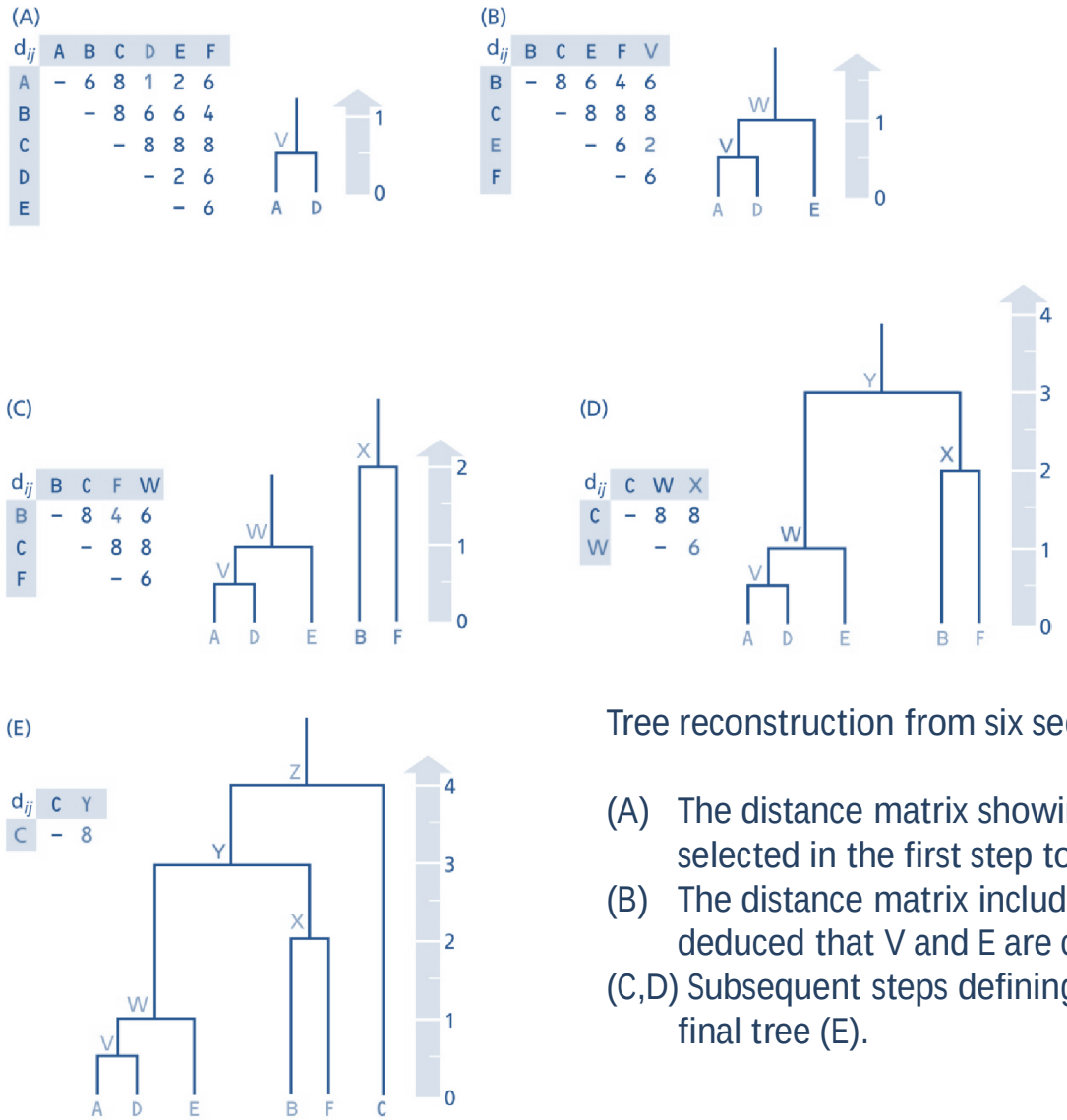
	(AB)	C
C	$d_{(AB)C}$	
D	$d_{(AB)D}$	d_{CD}

- A new distance matrix is computed by using AB composite OTU.

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

$d_{(AB)D} = (d_{AD} + d_{BD}) / 2$

UPGMA –method - a worked example

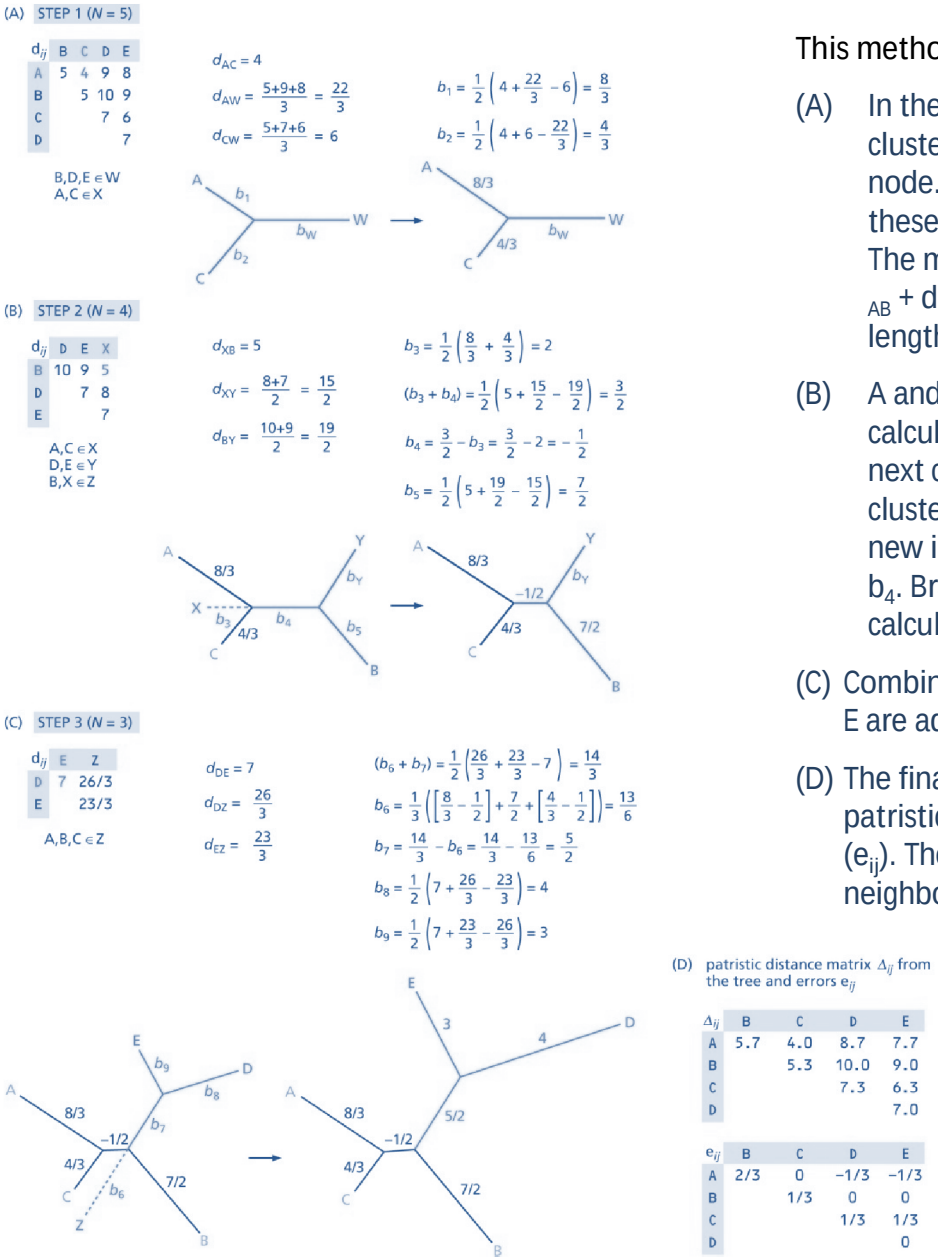


Tree reconstruction from six sequences, A-F.

(A) The distance matrix showing that A and D are closest. They are selected in the first step to produce internal node V (in (B)).
(B) The distance matrix including node V from which it can be deduced that V and E are closest, resulting in internal node W.
(C,D) Subsequent steps defining nodes X, Y and Z and resulting in the final tree (E).

This picture is from : Zvelebid&Baum, Understanding Bioinformatics, 2008, Garland Science, Page 279.

FITCH-MARGOLIASH METHOD - a worked example



This method is not in MEGA5-software

(A) In the first step the shortest distance is used to identify the two clusters (A,C) which are combined to create the next internal node. A temporary cluster (W) is defined as all clusters except these two, and the distances calculated from W to both A and C. The method then uses equations $b_1 = ½(d_{AB} + d_{AC} – d_{BC}$, $b_2 = ½(d_{AB} + d_{BC} – d_{AC})$, $b_3 = ½(d_{AC} + d_{BC} – D_{AB})$ to calculate the branch lengths from A and C to the internal node that connects them.

(B) A and C are combined into the cluster X and the distances calculated from the other clusters. After identifying B and X as the next clusters to be combined to create cluster Z, the temporary cluster Y contains all other sequences. X is the distance b_3 from the new internal node, and the distance between the internal nodes is b_4. Branch length b_4 is negative (not realistic); in future calculations this branch is treated like all others.

(C) Combining sequences A,B and C into cluster Z, the sequences D and E are added to the tree in the final step.

(D) The final tree has a negative branch length. The tables give the patristic distances (those measurer on the tree itself) and the errors (e_{ij}). The tree has a wrong topology, as becomes clear with the neighbor-joining tree from the same data.

(D) patristic distance matrix Δ_{ij} from the tree and errors e_{ij}

Δ_{ij}	B	C	D	E
A	5.7	4.0	8.7	7.7
B		5.3	10.0	9.0
C			7.3	6.3
D				7.0

e_{ij}	B	C	D	E
A	2/3	0	−1/3	−1/3
B		1/3	0	0
C			1/3	1/3
D				0

This picture is from : 277. Zvelebid&Baum, Understanding Bioinformatics, 2008, Garland Science, Page 281.

NEIGHBOR JOINING, NJ, ALGORITHM

- Neighbor joining has similarities to UPGMA, Differences in the choice of function f(C_1, C_2) and how to assign the distances

Find clusters C_1 and C_2 that minimise a function f(C_1, C_2)
Join the two clusters C_1 and C_2 into a new cluster C
Add a node to the tree corresponding to C
Assign distances to the new branch

- The distance d_{ij} for clusters C_i and C_j is $$d_{ij} = \frac{1}{|C_i||C_j|} \sum_{p \in C_i, q \in C_j} d_{pq}$$

- Let u(C_i) be the separation of cluster C_i from other clusters defined by $u(C_i) = \frac{1}{n-2} \sum_{C_j} d_{ij}$ where n is the number of clusters.

- Instead of trying to choose the clusters C_i and C_j closest to each other, neighbor joining at the same time
  - Minimises the distance between clusters C_i and C_j and
  - Maximises the separation of both C_i and C_j from other clusters

- *NJ is easy to use, and understand the results. However, the algorithm is not easy stuff.*

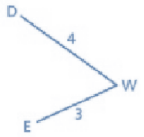
In case you want to learn more about NJ, see the attached original paper.

NJ-METHOD - a worked example

(A) STEP 1 ($N = 5$)

	d_{ij} B	C	D	E	U_i	$3\delta_{ij}$ B	C	D	E	
A	5	4	9	8	26	−40	−36	−32	−32	A
B		5	10	9	29		−36	−32	−32	B
C			7	6	22			−34	−34	C
D				7	33				−42	D
E					30					E

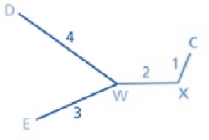
D and E are neighbors through internal node W with $d_{DW} = \frac{1}{2}\left(7 + \frac{33-30}{3}\right) = 4$ and $d_{EW} = 7 - 4 = 3$.



(B) STEP 2 ($N = 4$)

	d_{ij} B	C	W	U_i	$2\delta_{ij}$ B	C	W	
A	5	4	5	14	−20	−18	−18	A
B		5	6	16		−18	−18	B
C			3	12			−20	C
W				14				W

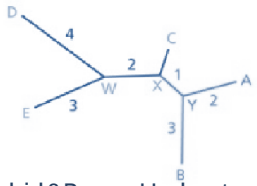
C and W are neighbors through internal node X with $d_{CX} = \frac{1}{2}\left(3 + \frac{12-14}{2}\right) = 1$ and $d_{WX} = 3 - 1 = 2$.



(C) STEP 3 ($N = 3$)

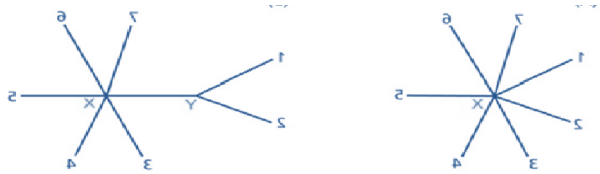
	d_{ij} B	X	U_i	δ_{ij} B	X	
A	5	3	8	−12	−12	A
B		4	9		−12	B
X			7			X

Three alternatives (of which here we choose one of the two with an internal node):
A and X are neighbors through internal node Y with $d_{AY} = 2$ and $d_{XY} = 1$ or
B and X are neighbors through internal node Y with $d_{BY} = 3$ and $d_{XY} = 1$.
Whichever is chosen, the remaining distance d_{AY} or d_{BY} will be found in the next d_{ij} matrix.



The distance matrix is the same as in the Fitch-Margoliash example.
At each step the distances are converted by using the algorithm which minimizes the total tree distance (the minimum evolution principle).

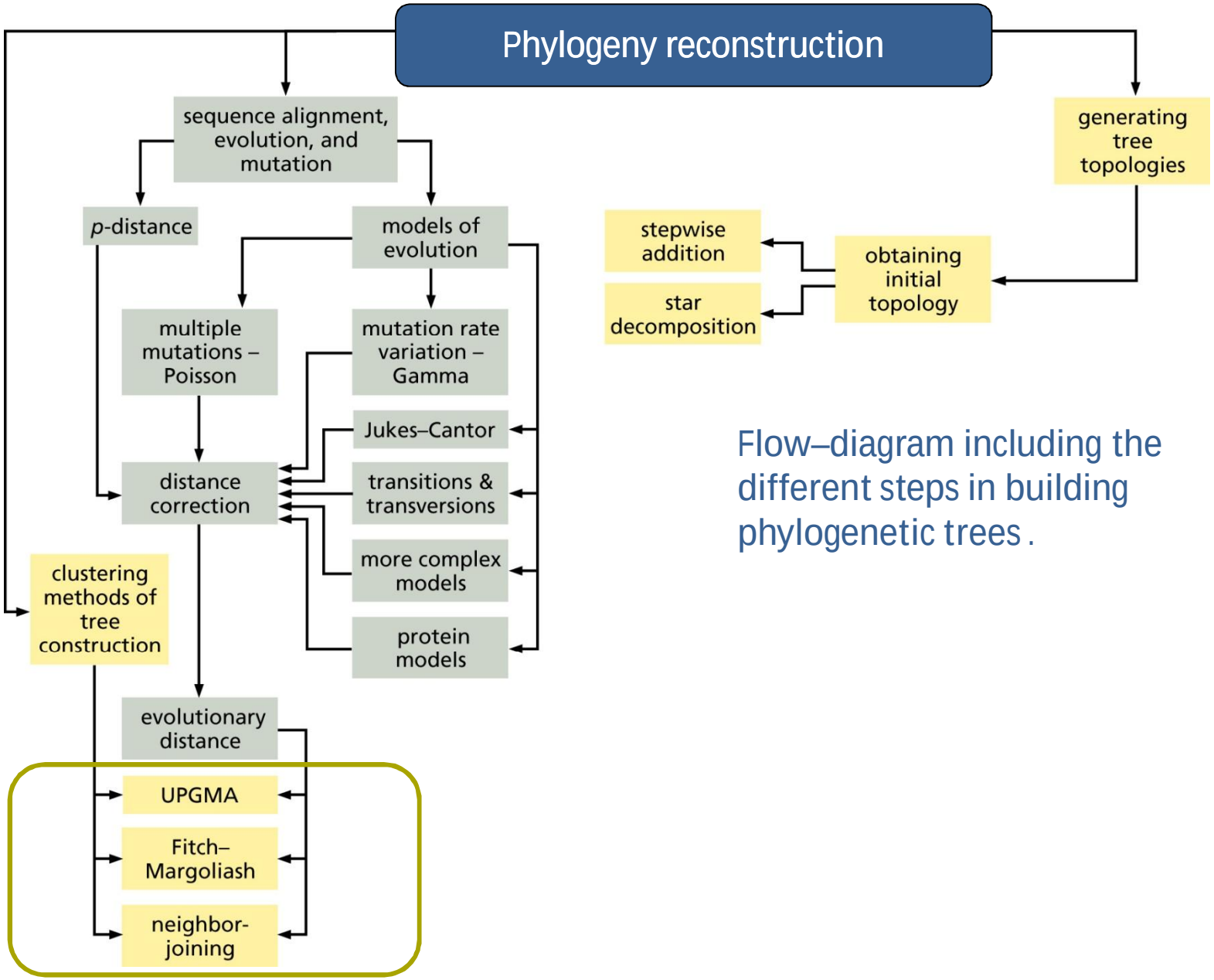
The first step:



(A) Star-tree in which all sequences are joined directly to a single internal node X with no internal branches.

(B) After sequences 1 and 2 have been identified as the first pair of nearest-neighbors, they are separated from node X by and internal node Y. The method calculates the branch lengths from sequences 1 and 2 to node Y to complete the step.

This picture is from : Zvelebid&Baum, Understanding Bioinformatics, 2008, Garland Science, Page 284



Flow–diagram including the different steps in building phylogenetic trees .

This picture is modified from : Zvelebid&Baum, Understanding Bioinformatics, 2008, Garland Science, Page 277.

MAXIMUM LIKELIHOOD PHYLOGENY INFERENCE

- The statistical framework maximum likelihood, was developed in 1922 by **R.A. Fisher.**

He showed that ML estimates have a variety of good properties:

- consistency (converging to the correct value of the parameter)
- efficiency (having the smallest possible variance around the true parameter value.

- The concept *likelihood* refers to a situation in which given some data **D**, a decision must be made about an adequate explanation of the data. A specific model and hypothesis are formulated in which the model as such is generally not in question.

- Two uses of likelihood in phylogenetic analysis:

  - to estimate parameters in the evolutionary model and to test hypotheses concerning the evolutionary process when the tree topology is known or fixed.

  - to estimate the tree topology. The log likelihood for each tree is maximized by estimating branch lengths and other substitution parameters, and the optimized log likelihood is used as a tree score for comparing different trees.

THE STATISTICAL CONCEPT *LIKELIHOOD*

● In phylogeny framework, one part of the model is that sequences actually evolve according to a tree. The possible hypotheses include the different tree topologies, the branch lengths, and the parameters of the model of sequence evolution.

● By assessing values to these elements, it is possible to compute the probability of the data under these parameters and to make statements about their plausibility.

● Some hypotheses produce the data with higher probability than others.

● Using the laws of conditional probability (P), and considering two hypotheses, H_1 and H_2 about a set of data (*D),* it can be shown, that

since $P(H \mid D) = P(H \text{ and } D) / P(D) = P(D \mid H) P(H) / P(D)$,

then $P(H_1 \mid D) / P(H_2 \mid D) = P(D \mid H_1) P(H_1) / P(D \mid H_2) P(H_2)$ **(1)**

● This expresses the *odds ratio* in favor of hypothesis 1 over hypothesis 2 as a product of two terms. The first is the ratio of the probabilities of the data given the two hypothesis. The second is the ratio of the prior probabilities of the two hypotheses (the odds ratio favoring H_1 over H_2) before looking at the data.

●Considering the odds favoring H_1 over H_2, the equation shows how to take into account the evidence provided by the data, and come up with a valid posterior odds ratio.

● The formula is the *odds ratio form of Bayes´ theorem*.

● The quantity $P(D \mid H)$ is called *the likelihood of the hypothesis H* . Note that this does not mean that it is the probability of the hypothesis, that would be $P(H \mid D)$.

It is the probability of data, given the hypothesis.

LIKELIHOOD OF A TREE

- If there are independent observations, then

$$P(D \mid H_i) = P(D^{(1)} \mid H_i) \times P(D^{(2)} \mid H_i) \times \ldots\ldots \times P(D^{(n)} \mid H_i) \quad \textbf{(2)}$$

- It follows that

$$P(D \mid H_1) / P(D \mid H_2) = (\Pi P(D^{(i)} \mid H_1 / P(D^{(i)} \mid H_2))(P(H_1) / P(H_2)) \quad \textbf{(3)}$$

- From equations (1) and (3) it can be seen that if there is large amount of data, the right side of the equation will be dominated by its first term which is the likelihood ratio of the two hypotheses.

- Consider a set of aligned DNA sequences with *n* sites and one possible phylogeny with branch lengths.

- An evolutionary model that allows computing probabilities of changes of states along this tree, the transition probabilities, $P_{ij}(t)$, the probability that state *j* will exist at the end of a branch of length *t*, if the state at the start of the branch is *i.* Note that *t* measures branch length, not time.

- Assumptions: (1) Evolution in different sites (on a given tree) is independent.

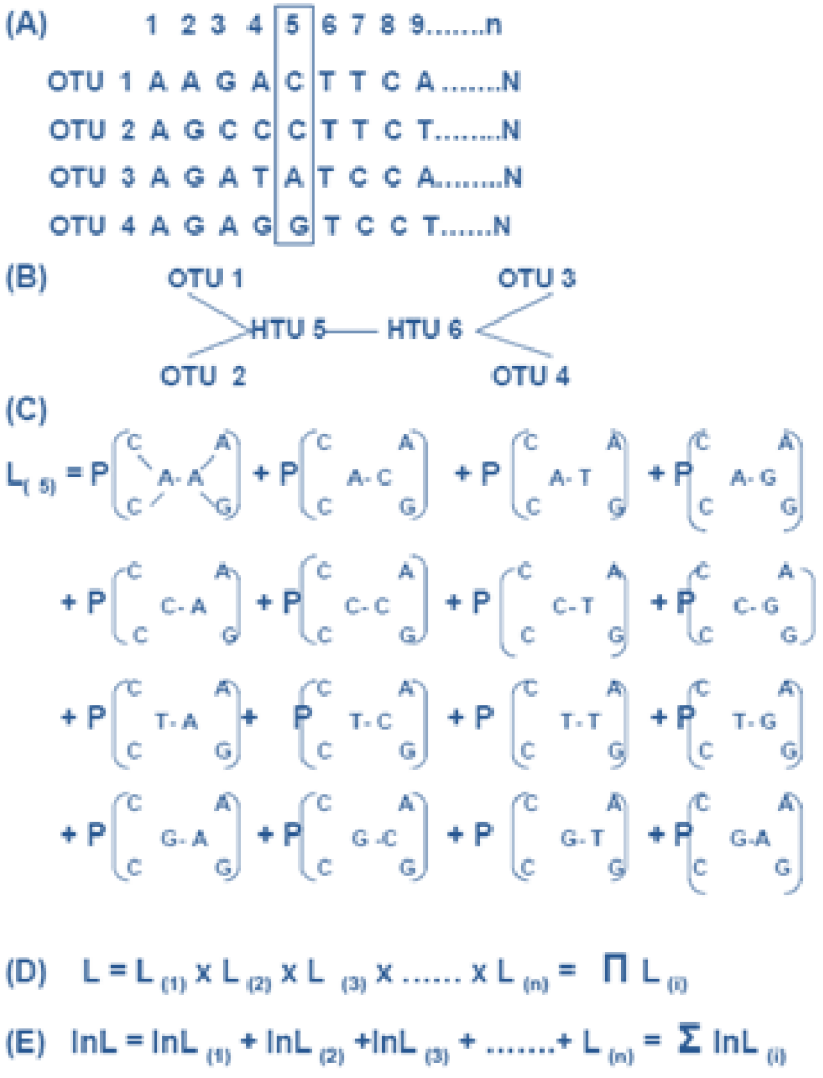
  (2) Evolution in different lineages is independent.

- The first assumption allows taking the likelihood and decomposing it into a product, one term for each site:

$$L = P(D \mid T) = \Pi P(D^{(i)} \mid T) \quad \textbf{(4)}$$

where $D^{(i)}$ is the data (nucleotide) at the i^{th} site and multiplication is over *n* sites.

LIKELIHOOD FOR ONE NUCLEOTIDE SITE



(A) Example of four OTUs, aligned sequences, 9 out of n sites are shown.

(B) One of three possible trees for the four OTUs with ancestral states (HTUs, hypothetical taxonomic units).

(C) The likelihood of one site, site 5, equals the sums of the 16 probabilities (P) of every possible reconstruction of ancestral states at nodes 5 and 6. Some of the possibilities are, of course, less plausible than others, but each has a non-zero probability of generating any pattern of observed nucleotides at the four tips of the tree. The probabilities are defined by a model of substitution.

(D) The likelihood of a tree in (B) is the product of the individual likelihoods for all n sites.

(E) The likelihood is usually evaluated by summing up logarithms of the likelihoods at each site => log likelihood of the tree.

LIKELIHOOD OF A TREE WITH BRANCH LENGTHS, ONE SITE

- The likelihood of the tree for the site (A,C,C,C,G) is the sum, over all possible nucleotides that may have existed at the interior nodes of the tree, of the probabilities of each scenario of events:

$$P(D^{(i)} \mid T) = \sum_{x}\sum_{y}\sum_{z}\sum_{w} P(A,C,C,C,G, x,y,z,w \mid T) \qquad \textbf{(5)}$$

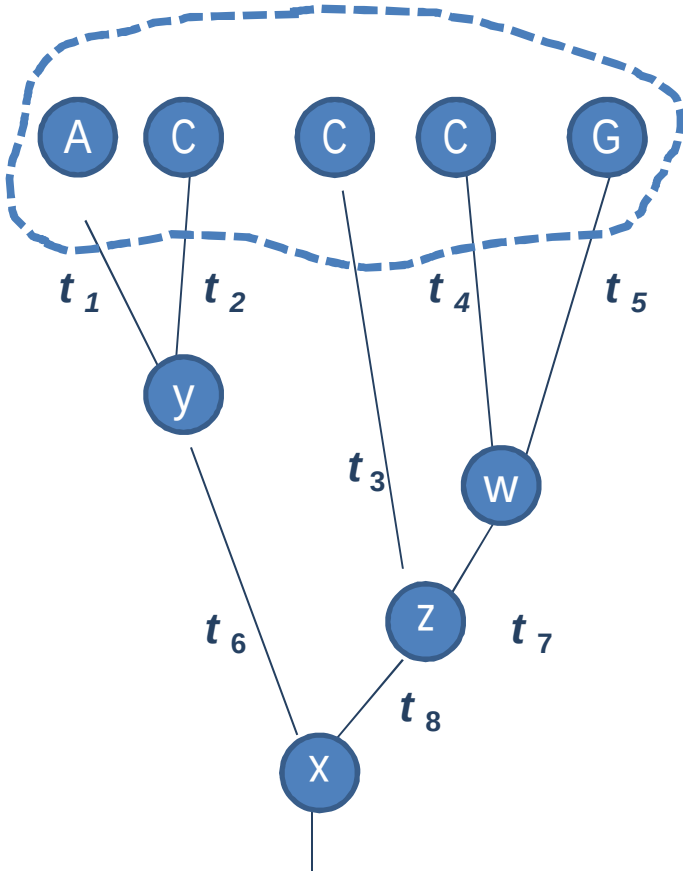
- Each summation runs over all four nucleotides.

- Assumtions (1) and (2) allow decomposing the probability in equation (5) into a product of terms:

$$\begin{aligned} &P(A,C,C,C,G,x,y,z,w \mid T) = \\ &P(x)\quad P(y \mid x,t_6)\quad P(A \mid y,t_1)\quad P(C \mid y,t_2) \\ &\qquad P(z \mid x,t_8)\quad P(C \mid z,t_3) \\ &\qquad\qquad P(w \mid z,t_7)\quad P(C \mid w,t_4)\quad P(G \mid w,t_5) \end{aligned} \qquad \textbf{(6)}$$

- The probability of x can be regarded as the probability that - at a random point on an evolving lineage - nucleotide x (where x=A,C,G or T) would be seen.

5 OTUs, one site as an example



LIKELIHOOD OF A TREE WITH BRANCH LENGTHS, ONE SITE

- Assuming that evolution has been proceeding for a very long time according to the particular model of nucleotide substitution that is used, it is reasonable to take P(x) to be *the equilibrium probability of nucleotide x under that model.* The other probabilities are derived from the model of nucleotide substitution. The change in each lineage is independent of that in all other lineages, once the nucleotides at the start of each lineage have been specified.

- **Computing expression (6)**

The individual probabilities are not problematic, they are the probabilities which are specified in a nucleotide substitution model (for example Jukes-Cantor, Kimura 2-parameter, or a more sophisticated model). There are, however, many terms. Each site requires summing $4^4 = 256$ terms. The number of terms rises exponentially with the number of OTUs:

  - On a tree with n OTUs, there are $n-1$ interior nodes, and each can have one of 4 states.
  - So, 4^{n-1} terms are needed.
  - $n = 10$ => 262 144 states
  - $n = 20$ => 274 877 906 944 states for one site

THE PRUNING ALGORITHM

● The pruning alqorithm for economizing the computation was introduced by Joe Felsenstein in 1973. The method may be derived by trying to move summation signs in equation (6) as far right as possible and enclose them in parentheses where possible. Equation (6) can be rewritten

$$P(D^{(i)} \mid T) = \sum_x \sum_y \sum_w \sum_z P(x)\, P(y \mid x, t_6)\, P(A \mid y, t_1) \\ P(C \mid y, t_2) \\ P(z \mid x, t_8) P(C \mid z, t_3) \\ P(w \mid z, t_7) P(C \mid w, t_4 P(G \mid w, t_5) \quad \textbf{(7)}$$

and moving summation signs as far right as possible

$$P(D^{(i)} \mid T) = \quad \textbf{(8)}$$
$$\sum_x P(x) \left(\sum_y P(y \mid x, t_6) P(A \mid y, t_1) P(C \mid y, t_2) \right)$$
$$\sum_z P(z \mid x, t_8) P(C \mid z, t_3) \left(\sum_w P(w \mid z, t_7) P(C \mid w, t_4) P(G \mid w, t_5) \right)$$

● Note that the pattern of parentheses and tips in this expression is (A,C)(C,(C,G)), as is in the tree.

■ Pruning is a special case of the *peeling algorithm,* which was introduced in 1970´s for rapidly computing likelihoods on pedigrees in human genetics. Peeling, in turn, is a special case of Horner´s algorithm, which is for the efficient evaluation of polynomials in monomial form (Isaac Newton used this).

■ The flow of computation in expression (8) is from the inside of the innermost parentheses outwards. This suggests a flow of information down the tree, and an algorithm (to compute the equation) that works in this way is as follows: The probability of everything that is observed from node *k* on the tree on up, at site *i* , conditional on node *k* having state *s.* In equation (8) the term $P(C \mid w, t_4) P(G \mid w, t_5)$ is one of these quantities: the probability of everything seen at or above that node (the node that lies below the rightmost two tips), given that the node has nucleotide w. There will be four such quantities, corresponding two different values of w. The key to the pruning algorithm is that, once these four numbers are computed, they need not continually be recomputed.

THE PRUNING ALGORITHM

- The algorithm is expressed as a recursion that computes the $L^{(i)}(s)$ at each node on the tree from the same quantities in the immediate descendant nodes.

- Suppose that node *k* has immediate descendants *l* and *m,* which are at the top ends of branches of length t_l and t_m. Then

$$L_k^{(i)}(s) = \left(\sum_x P(x \mid s, t_l)\, L_l^{(i)}(x)\right)\left(\sum_y P(y \mid s, t_m)\, L_m^{(i)}(y)\right) \qquad \textbf{(9)}$$

- This is the probability of everything at or above node *k,* given that node *k* has state *s*, is the product of the events taking place on both descendant lineages.

- In the left lineage, it sums over all of the states to which *s* could have changed, and for each of those computes the probability of changing to that state, times the probability of everything at or above that node (node *l*), given that the state has changed to state x.

THE PRUNING ALGORITHM

- To start the process the values of the $L^{(i)}$ at the tips of the tree are needed.

If state A is found at a tip, the values of the $L^{(i)}$ at the tip will be

$$(L^{(i)}(\mathrm{A}), L^{(i)}(\mathrm{C}), L^{(i)}(\mathrm{G}), L^{(i)}(\mathrm{T})) = (1, 0, 0, 0) \quad \textbf{(10)}$$

whichever nucleotide is seen at the tip has the corresponding value of $L^{(i)}$ set to 1, and all others are 0.

- The algorithm is applied starting at the node that has all of its immediate descendants being tips (always at least one such node exists). Then it is applied successively to nodes further down the tree, not applying it to any node until all of its descendants have been processed. The result is the $L_{(0)}^{(i)}$ for the bottom-most node of the tree.

- Evaluation of the likelihood of this site is completed by making a weighted average of these over all four nucleotides, weighted by their prior probabilities under the probabilistic model

$$L^{(i)} = \sum \pi_x L_{(0)}^{(i)}(\mathrm{x}) \quad \textbf{(11)}$$

- Once the likelihood for each site is computed, the overall likelihood of the tree is the product of these, as noted in equation (4).

ML TREES, PHYLIP-PROGRAM

● The pruning algorithm for updating the likelihoods along a tree simplifies the task of finding the ML tree. This is, however, only one part of the total task. There is a space of trees with branch lengths. The optimum branch lengths for each given tree topology need to be found, and also the tree space must be searched for tree topologies for the topology that has a set of branch lengths that gives it the highest likelihood. No easy analytical solutions to the problem of finding the optimal branch lengths for a given tree topology exist and there may be multiple local maxima for likelihoods.

● The famous program package for ML-phylogeny inference is Joe Felsenstein´s PHYLIP (includes also distance matrix and parsimony methods) http://evolution.genetics.washington.edu/phylip/doc/dnaml.html. **This link includes many other program packages, ML and other phylogeny inference methods.**

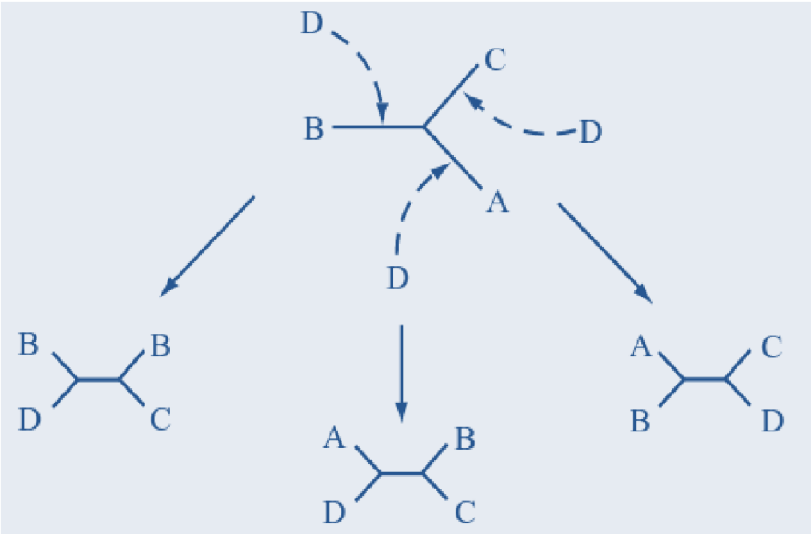
● PHYLIP uses a Hidden Markov Model (HMM) method of inferring different rates of evolution at different sites. It allows to specify that there will be a number of different possible evolutionary rates, what the prior probabilities of occurrence of each is, and what the average length of a patch of sites all having the same rate is. The rates can also be chosen by the program to approximate a Gamma distribution of rates, or a Gamma distribution plus a class of invariant sites. The program computes the likelihood by summing it over all possible assignments of rates to sites, weighting each by its prior probability of occurrence.

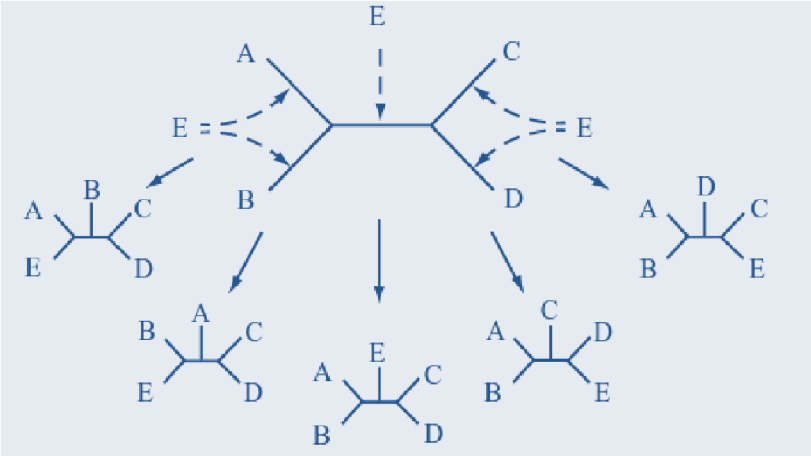
SEARCHING FOR THE ML-TREE

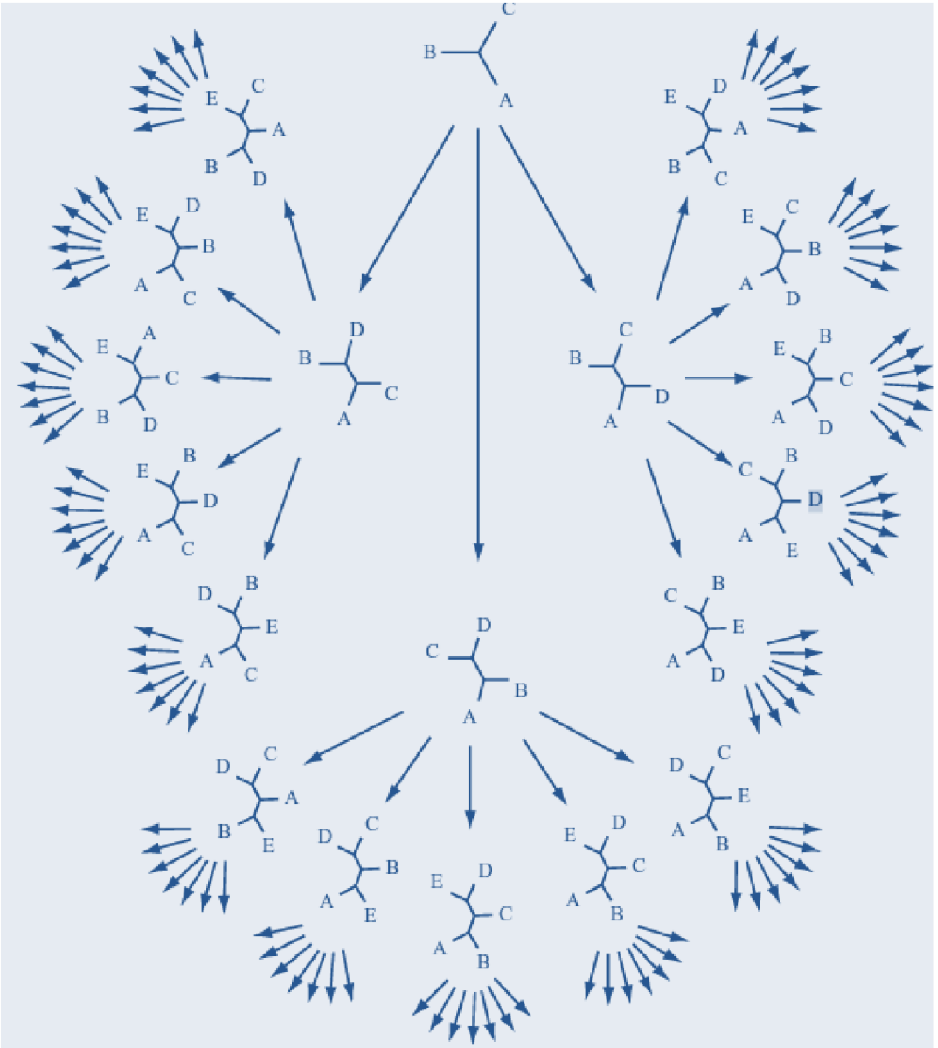
● Below, next page (taken from Lemey *et al., The phylogenetic handbook, 2009*) one procedure: The algorithm recursively adds the *t*th OTU in a stepwise fashion to all possible trees containing the first *t* – 1 OTUs until all *n* OTUs have been joined. For rooted trees the algorithm is modified by including one additional artificial OTU that locates the root of each tree. In this case, the first three trees generated represent each of the three possible rootings of an unrooted three-OTU tree, and the algorithm proceeds as in the unrooted case. Thus, the number of rooted trees for *n* OTUs is equal to the number of unrooted trees for *n* + 1 OTUs.

● Six OTUs (A,B,C,D,E,F), start with A,B,C, the fourth, D, connected to each of the three branches, fifth, E, connected to each three trees → all 15 possible trees generated....→ all 105 possible trees generated and their lengths evaluated.

GENERATION OF ALL POSSIBLE TREES





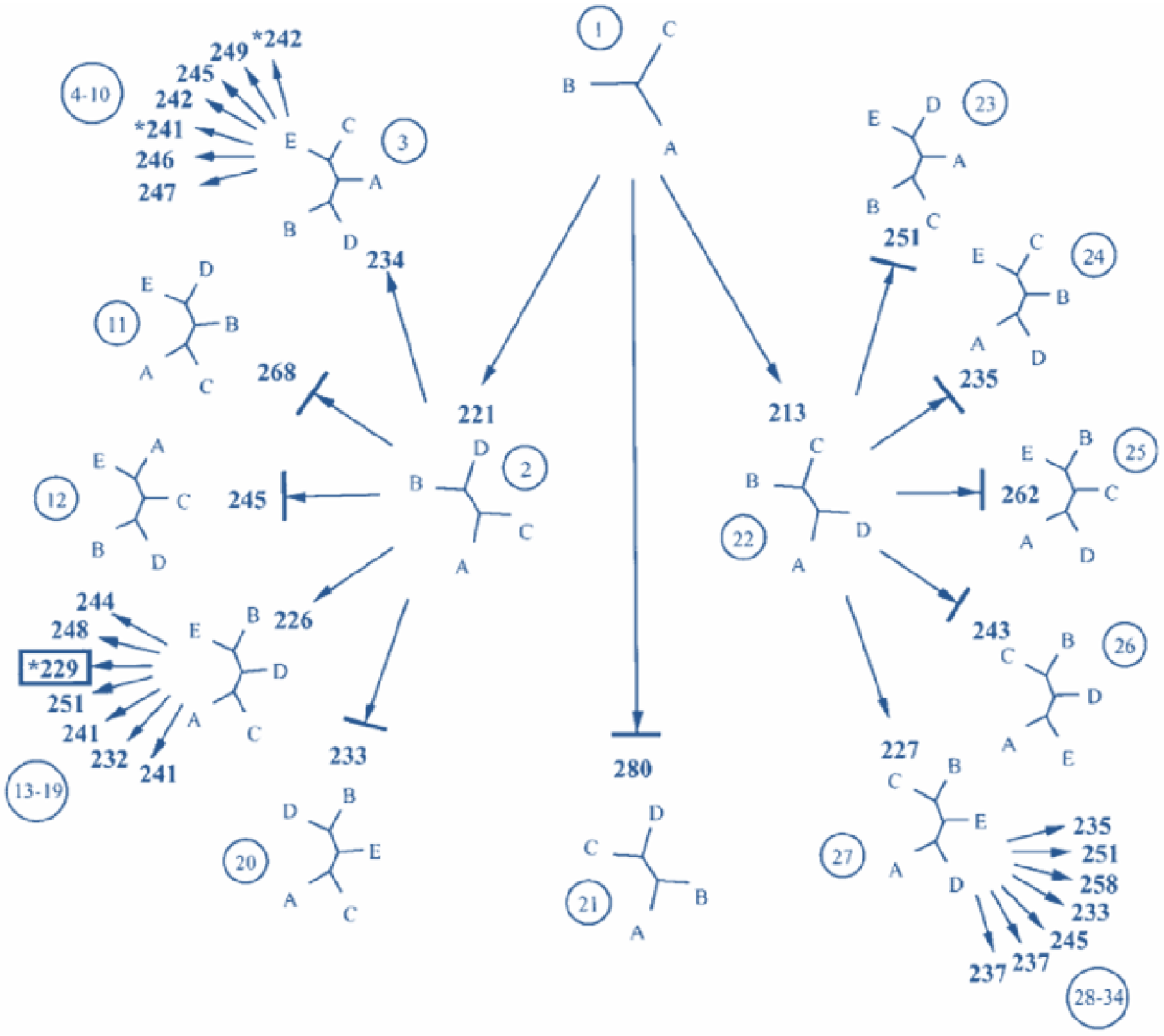


BRANCH-AND-BOUND ALGORITHM, AN EXACT METHOD by HENDY & PENNY (1982)

- In the example 6 OTUs. The method operates by implicitly evaluating all possible trees, but cutting off paths of the search tree when it is determined that they cannot possibly lead to optimal trees.

- The algorithm effectively traces the same route through the search tree as used in the previous example, but the length of each tree encountered at the node of the search tree is evaluated even if it does not contain the full set of OTUs.

- Throughout the traversal, an upper bound on the length of the optimal tree(s) in maintained. Initially the upper bound can simply be set to inifinity.



BRANCH-AND-BOUND ALGORITHM

● ...continued..
The traversal starts by moving down the left branch of the search tree successively connecting OTU D and E to the initial tree with lengths of 221 and 234 steps, respectively.

● Then, connecting OTU F provides the first set of full-tree lengths. After this connection, it is known that a tree of 241 steps exists, although it is not yet known whether this tree is optimal. Therefore this number is taken as a new upper bound on the length of the optimal tree (i.e. the optimal tree cannot be longer than 241 steps because a tree at this length has already been identified).

● After this, the algorithm backtracks on the search tree and takes the second path out of the 221-step, 4-OTU tree. The 5-OTU tree containing OTU E obtained by following this path requires 268 steps. Thus: there is no point in evaluating the seven trees produced by connecting taxon F to this tree because they cannot possibly require fewer than 268 steps, and a tree of 241 steps has already been found. By cutting off paths in this way, large portions of the search tree may be avoided and a considerable amount of computation time saved.

● The algorithm proceeds to traverse the remainder of the search tree, cutting off paths where possible, and storing optimal trees when they are found. In the example, a new optimal tree is found at a length of 229 steps, allowing the upper bound on the tree length to be further reduced. Then, when the 233-step tree containing the first five OTUs is encountered, the seven trees that would be derived from it can be immediately rejected because they would also require at least 233 steps. The algorithm terminates when the root of the search tree has been visited for the last time, at which all optimal trees will have been identified.

● This methods is said to be feasible for 12-25 OTUs.

BRANCH-AND-BOUND ALGORITHM

- Refinements to the branch-and-bound algorithm to improve its performance:
  - Including a *heuristic method,* like stepwise addition (described below).
  - Including neighbor-joining algorithm (cf. distance matrix methods of phylogeny inference) to find a tree whose length provides a smaller initial upper bound, which allows earlier termination of search paths in the early stages of the algorithm.
  - Ordering the sequential addition of OTUs in a way that promotes earlier cutoff of paths, rather than just adding them in order of their appearance in the data matrix.
  - Using techniques such as pairwise character incompatibility to improve the lower bound on the minimum length of trees that can be obtained by counting traversal of the search tree → allows earlier cutoffs.

- **The algorithm can ne used for any optimality criterion - in addition to parsimony, also maximum likelihood - whose objective function is guaranteed to be non-decreasing as additional OTUs are connected to the tree.**
  - For parsimony and maximum likelihood approaches this is true: increasing the variability of the data by adding additional OTUs cannot possibly lead to a decrease in tree length.
  - For minimum-evolution distance criterion this does not work: One objective function is optimized for the computation of branch lengths (i.e. least-squares fit), but a different one is used to score the trees (i.e. sum of branch lengths).

HEURISTICS, GREEDY ALGORITHMS: STEPWISE ADDITION, FARRIS (1970)

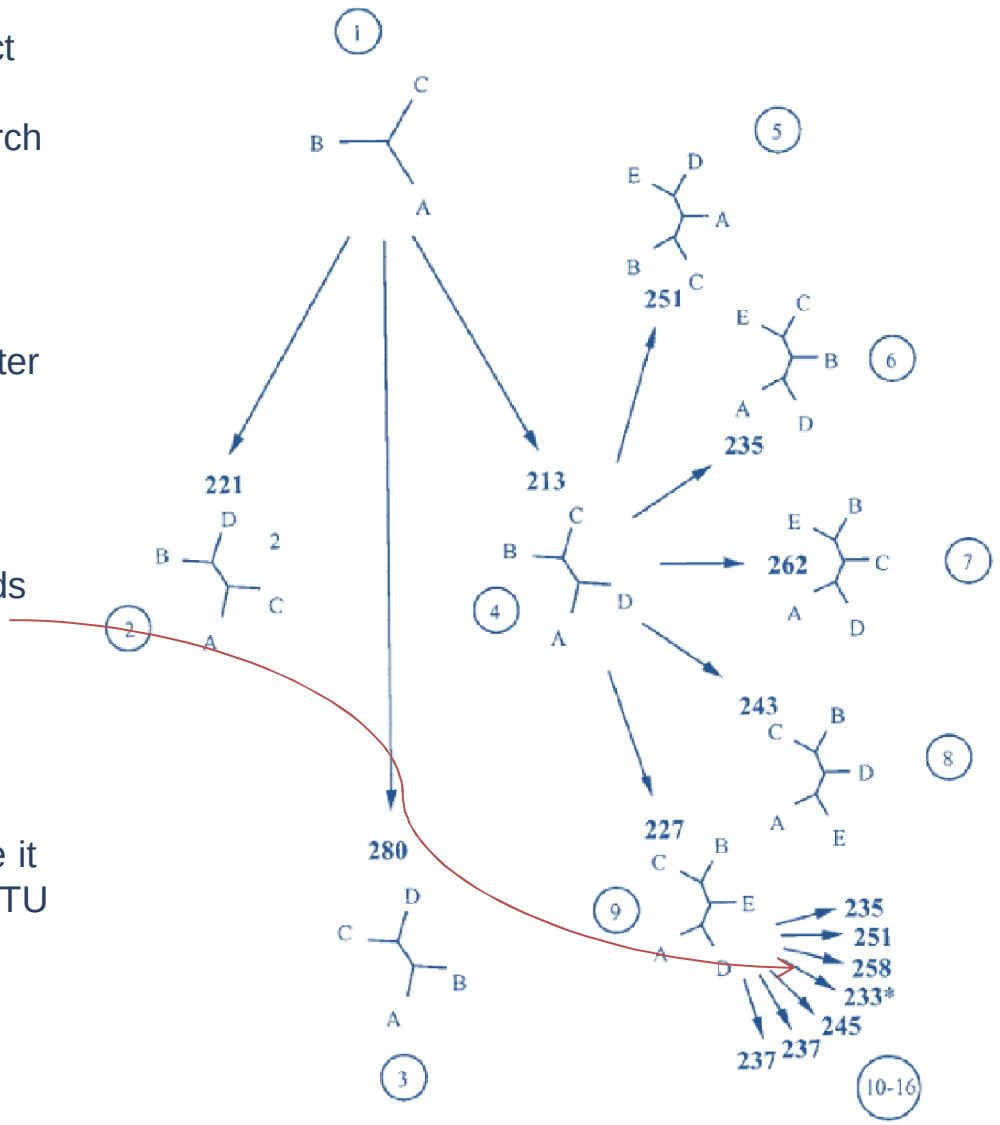
● Follows the same kind of search tree as the branch-and-bound method, but unlike the exact exhaustive enumeration, stepwise addition commits to a path out of each node on the search tree that looks most promising at the moment. *This might not lead to a global optimum.*

● In the previous example of exact branch-and –bound, tree 22 is shorter than trees 2 or 21. Thus only trees derivable from tree 22 remain as candidates.

● Following this path ultimately leads to selection of a tree of 233 steps which is *only a local* rather than a global optimum.

● The path leading to the optimal 229-step tree was rejected because it appeared less promising at the 4-OTU stage.

● Greedy heuristics are called *local-search methods* because of their tendency to become "stuck" in local optima.



HEURISTICS BY BRANCH-SWAPPING: NEAREST-NEIGHBOR INTERCHANGE (NNI)

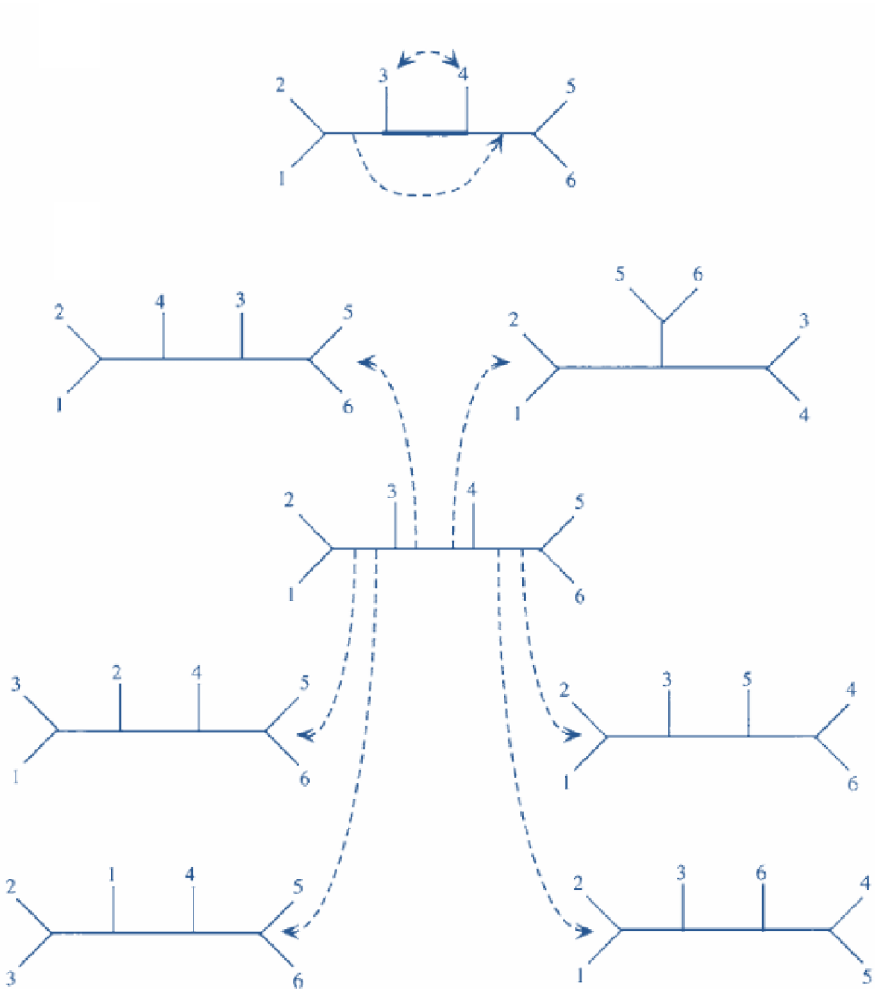
● Branch-swapping methods involve cutting off one or more pieces of a tree (subtrees) and reassembling them in a way that is locally different from the original tree.

● Nearest-neighbor interchange (NNI) is the simplest type of rearrangement.

● For any binary tree containing *T* terminal OTUs, there are *T* – 3 internal branches. Each branch is visited, and the two topologically distinct rearrangements that can be obtained by swapping a subtree connected to one end of the branch with a subtree connected to the other end of the branch are evaluated.

● This proceduce generates a relatively small number of perturbations whose lengths or scores can be compared to the original tree.

● A more extensive rearrangement scheme is subtree pruning and regrafting (next page).



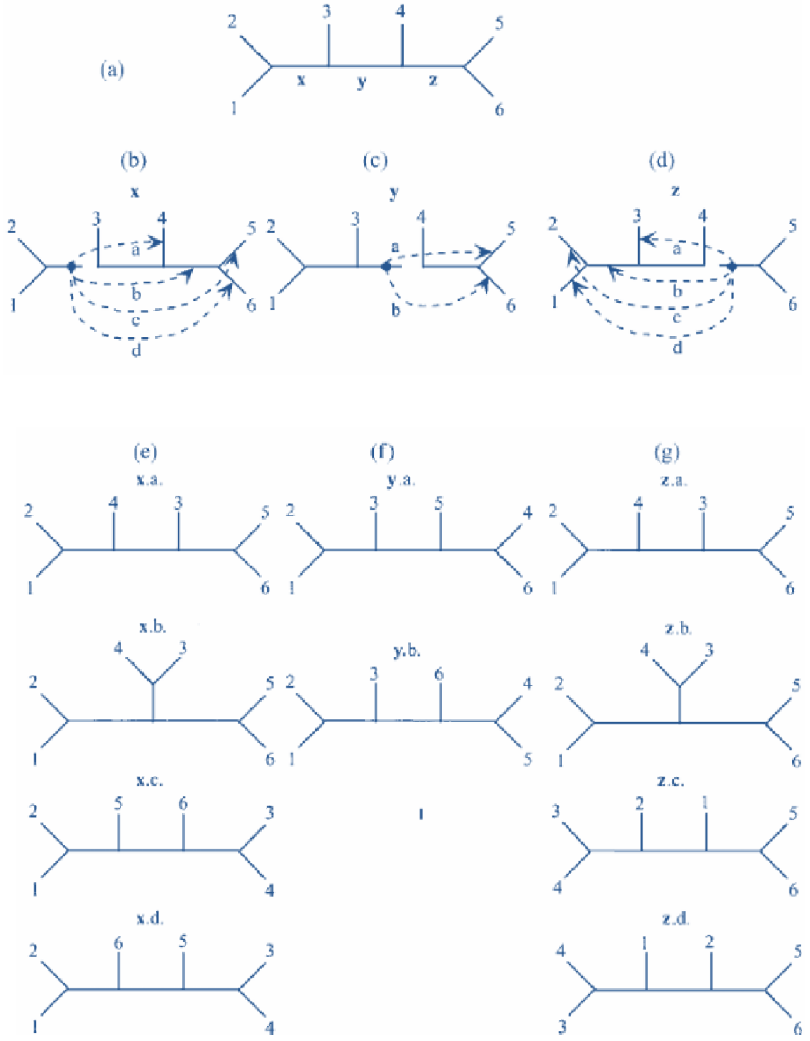
HEURISTICS BY BRANCH-SWAPPING: SUBTREE PRUNING AND REFRAFTING (SPR)

● The method subtree pruning and regrafting (SPR) involves clipping off all possible subtrees from the main tree and reinserting them at all possible locations, but avoiding pruning and grafting operations that would generate the same tree redundantly.

(a) The tree to be rearranged

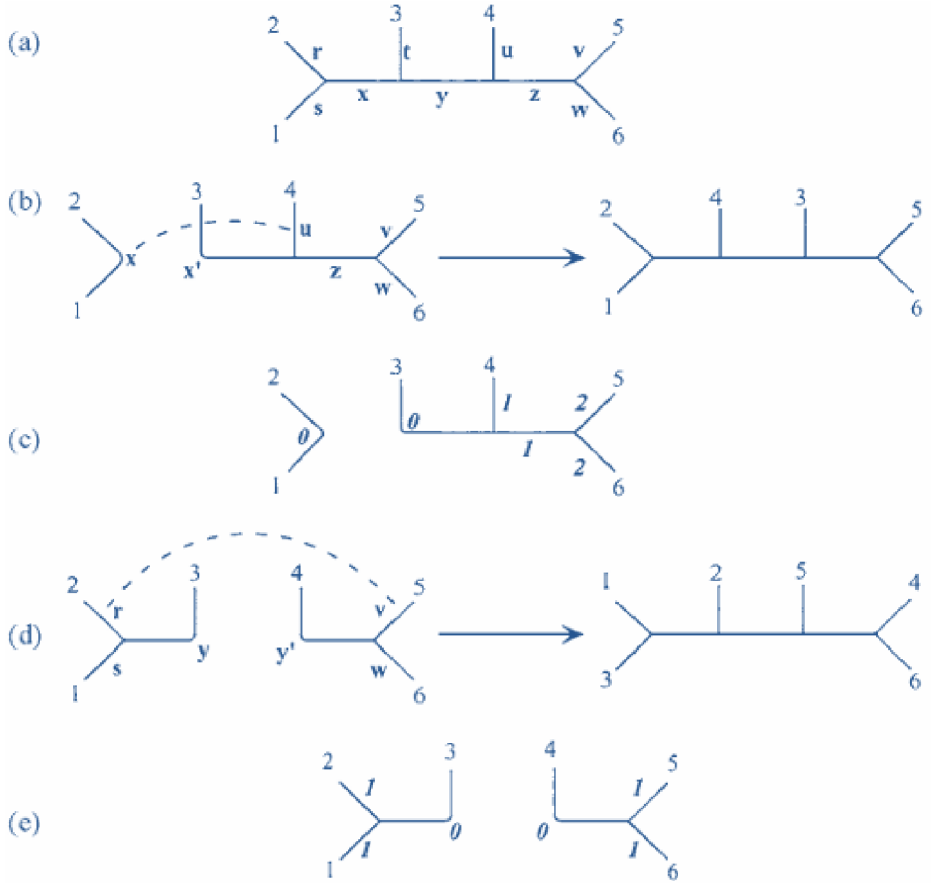
(b), (c), (d)
SPRs resulting from pruning of branches, x, y, z, respectively. In addition to these rearrangements, all terminal OTUs (leaves) would be pruned and reinserted elsewhere on the tree.

(e), (f), (g)
Trees resulting from regrafting of branches x, y, z, respectively, to other parts of the tree.



HEURISTICS BY BRANCH-SWAPPING: TREE BISECTION AND RECONNECTION (TBR)

● Tree bisection and reconnection (TBR) involve cutting a tree into two subtrees by cutting one branch, anf then reconnecting the two subtrees by creating a new branch that joins a branch on one subtree to a branch on the other. All possible pairs of branches are tried, avoiding redundancies.



(a) The tree to be rearranged.

(b) Bisection of branch x and reconnection to branch u. Other TBRs would connect x to z,v and w, respectively.

(c) Branch numbering for reconnection distances involving branch x.

(d) Bisection of branch y and reconnection of branch r to v. Other TBRs would connect r to w, r to y´, s to v, s to w, s to y´, y to v and y to w, respectively.

(e) Branch numbering for reconnection distances involving branch y. All other branches, both internal and external, also would be cut in a full round of TBR swapping.

RELATIONSHIPS BETWEEN NNI, SPR, TBR AND THEIR PERFORMANCE

● The set of possible NNIs for a tree is a subset of the possible SPR rearrangements and the set of possible SPR rearrangements, in turn, a subset of the possible TBR rearrangements.

● For TBR rearrangements, a “reconnection distance” can be defined by numbering the branches from zero starting at the cut branch (see the figure, (c) and (d)) . The reconnection distance is then equal to the sum of numbers of the two branches that are reconnected. The reconnection distance is then equal to the sum of numbers of the two branches that are reconnected and have the following three properties:

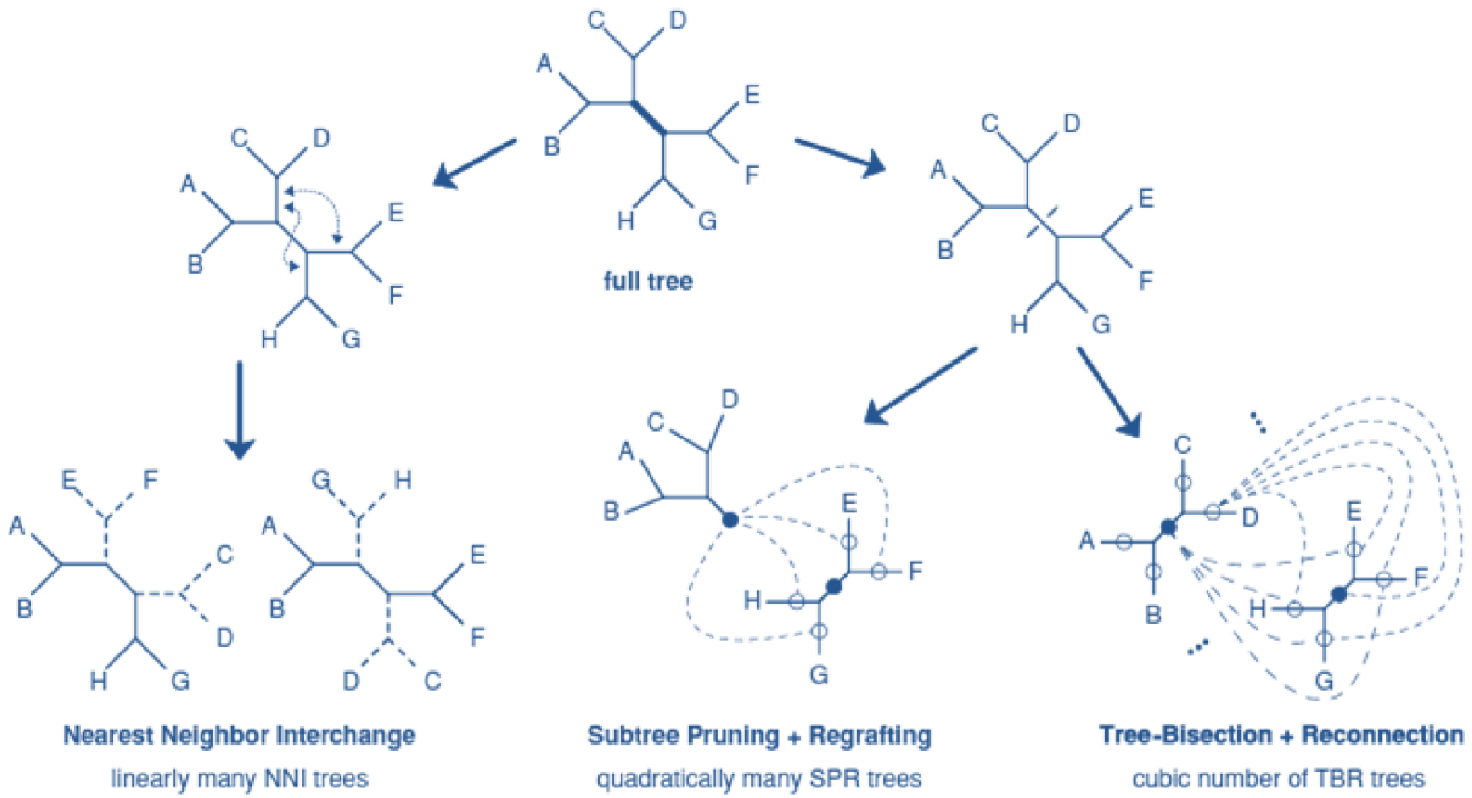
- NNIs are the subset of TBRs that have a reconnection distance of 1.
- SPRs are the subset of TBRs so that exactly one of the two reconnected branches is numbered zero.
- TBRs that are neither NNIs nor SPRs are those for which both reconnected branches have non-zero numbers.

● The reconnection distance can be used to limit the scope of TBR rearrangements tried during the branch-swapping procedure.

● The default strategy used for each of these rearrangement methods is to visit branches of the “current” tree in some arbitrary and predefined order. At each branch, all of the non-redundant branch swaps are tried and the score of each resulting tree is obtained.

● If a rearrangement is succesful in finding a shorter tree, the previous tree is discarded and the rearrangement process is restarted on this new tree. If all possible rearrangements have been tried without success in finding a better tree, the swapping process terminates.

RELATIONSHIPS BETWEEN NNI, SPR, TBR AND THEIR PERFORMANCE



● The three basic rearrangement operations on the thick branch in the full tree. In SPR and TBR all pairs of "circled" branches among the two subtrees will be connected (dashed lines), except the two filled circles to each other, since this yields to full tree again.

RELATIONSHIPS BETWEEN NNI, SPR, TBR AND THEIR PERFORMANCE

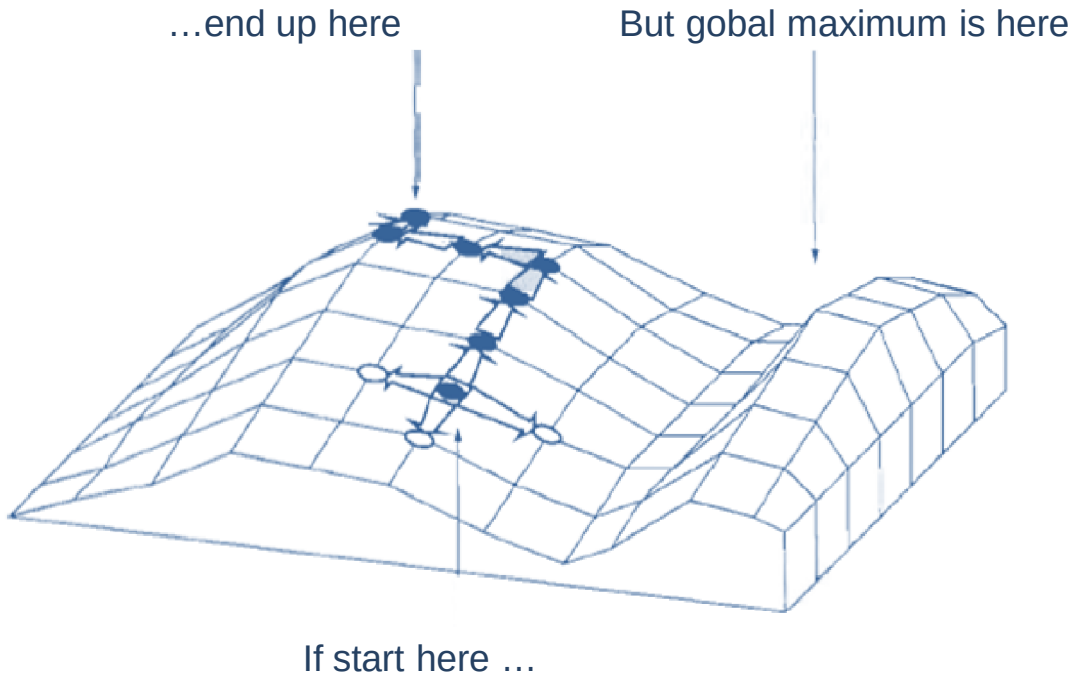
- Optionally, when trees are found that are equal in score to the current tree (e.g. equally parsimonious trees or trees that have identical likelihoods within round-off error), they are appended to a list of optimal trees.
    - In this case, when the arrangement of one tree finishes, the next tree in the list is obtained and input to the branch-swapping algorithm.
    - If the rearrangement of this next tree yields a better tree than any found so far, all trees in the current list are discarded and the entire process is restarted using a newly discovered tree.
- The algorithm terminates when every possible rearrangement has been tried on each of the stored trees.

- In addition to identifying multiple and equally good trees, this strategy often identifies better trees than would be found if only a single tree were stored at any one time. This can happen when all of the trees within one rearrangement of the current tree are no better than the current tree. However, some of the adjacent trees can, in turn, be rearranged to yield trees that are better.

- By only accepting proposed rearrangements that are equal to, or better than, the current best tree, these “hill-climbing algorithms” eventually reach the peak of the slope on which they start. However, the peak may not represent a global optimum.

    - Some phylogeny software packages have options to begin the search from several starting points (randomly chosen tree topologies) in the hope that at least one of them will result in climbing the right hill.

RELATIONSHIPS BETWEEN NNI, SPR, TBR AND THEIR PERFORMANCE



● A surface rising above a two-dimensional plane. Thwe process of climbing uphill on the surface is illustrated, as well as the failure to find a higher peak by a”greedy” method.

RELATIONSHIPS BETWEEN NNI, SPR, TBR AND THEIR PERFORMANCE

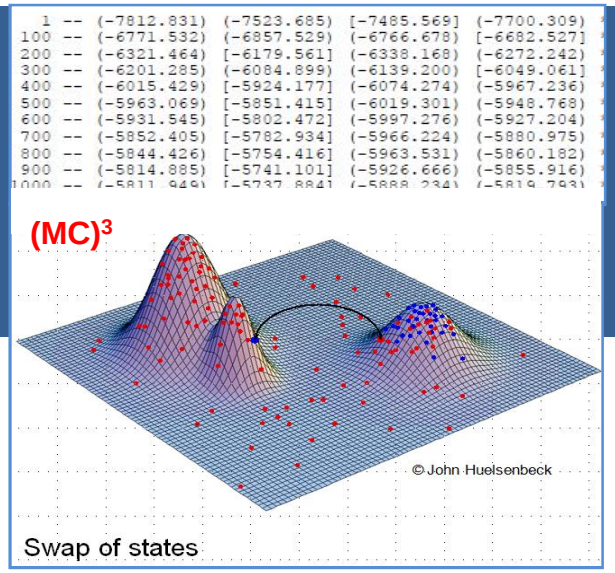
- An alternative method takes advantage of the fact that, for data sets of non-trivial size and complexity, varying the sequence (order) in which OTUs are added during stepwise addition may produce different tree topologies that each fit the data reasonably well .
- Starting branch-swapping searches from a variety of random-addition-sequence replicates thereby provides a mechanism for performing multiple searches that each begins at a relatively high point of some hill, increasing the probability that the overall search will find an optimal tree.

- Random-addition-order searches are also useful in identifying multiple “islands” of trees that may exist. Each island represents all of the trees that can be obtained by an order of rearrangements, starting from any tree in the island, keeping and rearranging all optimal trees that are discovered. If two optimal trees exist so that it is impossible to reach one tree by an order of rearrangements starting from the other without passing through trees that are suboptimal, these trees are on different islands. Because trees from different islands tend to be topologically dissimilar, it is important to detect multiple island when they exist.

- All these methods are said to be effective for data sets containing up to ~100 OTUs.

- For larger data sets, some methods that use a variety of stochastic-search and related algorithms that are better able to avoid entrapment in local optima.

BAYESIAN PHYLOGENY INFERENCE



● Maximum likelihood chooses amongst hypotheses by selecting the one which maximizes the likelihood, i.e. which renders the data most plausible. The likelihood of a hypothesis is equal to the probability of observing the data, given the hypothesis.

● Bayesian approach is closely related and differs in the use of *prior distribution* of the quantity being inferred.

● Equation (1) in the description of maximum likelihood (see the slides ”Maximum likelihood phylogeny inference”) gives Bayes´ theorem in an odds-ratio form.

● Given and hypothesis *H* and some data *D,* the probability of the hypothesis given the data is

$$P(H \mid D) = P(H \,\&\, D) / P(D) \qquad \textbf{(12)}$$

● The joint probability of *H* and *D,* P(*H* & *D*), can be written as product of the probability of *H* and the conditional probability of *D* given *H*

$$P(H \,\&\, D) = P(H)\, P(D \mid H) \qquad \textbf{(13)}$$

● Substituting (2) into (1): $P(H \mid D) = [\, P(H)\, P(D \mid H\,] / P(D)$ **(14)**

● **This is Bayes´ theorem in its simplest form.**
The denominator *P* (*D*) is the sum of the numerators *P* (*H & D*) over all possible hypotheses *H* and is the quantity that is needed to normalize them so that they add up to 1.

● This leads to the **more usual form of the theorem**

$$P(H \mid D) = [\, P(H)\, P(D \mid H\,] / \Sigma_H\, P(H)\, P(\, D \mid H\,) \qquad \textbf{(15)}$$

● When not in odds-ratio form, Bayes´ theorem allows turning a prior distribution into a posterior distribution. It computes the probabilities of different hypotheses in the light of the data.

BAYESIAN PHYLOGENY INFERENCE

● Recap once more equation (1) in maximum likelihood inference. The odds favoring one hypothesis over another are the odds the person gave them in itially (the prior odds), multiplied by the ratio of the likelihoods under the data. Suppose that there are two possible hypotheses, and in advance we favor H_1 over H_2 , giving them odds 3:2. Now the data is examined. The likelihood ratio $P(D \mid H_1) / P(D \mid H_2)$ turns out to be ½, so that the data is half as probable given hypothesis 1 as it is given hypothesis 2. Bayes´ theorem tells us to compute the posterior odds ratio by multiplying these to get (3/2) x (1/2) = ¾. After looking at the data we now give odds in favor of H_1 of only 3:4 . This very reasonable, given the correctness of the prior odds. Controversial is whether usable prior odds exist.

● Bayesian statistics tries to come up with valid prior probabilities and use formulas such as (1) to infer valid posterior probabilities of the various hypotheses.

● Non-Bayesians are sceptical of the ability to come up with valid prior probabilities. They may prefer that hypothesis that maximizes the likelihood $P(D \mid H)$. This may not end up being the hypothesis that has the largest posterior probability, but if the amount of data is large, the chance that it is, is good. As the amount of data increases, this maximum likelihood estimate will become more and more likely to be the best estimate as well, as equation (3) becomes dominated by the quantity in large parentheses.

BAYESIAN PHYLOGENY INFERENCE

● The essence of the Bayesian viewpoint is that there is no logical distinction between model parameters and data. Both are ***random variables*** with a ***joint probability distribution*** that is specified by a probabilistic model. From this viewpoint, ***data*** are observed variables and ***parameters*** are unobserved variables.

● The joint distribution is a product of the likelihood and the ***prior.***

● The prior includes information about the values of a parameter before examining the data in the form of a probability distribution.

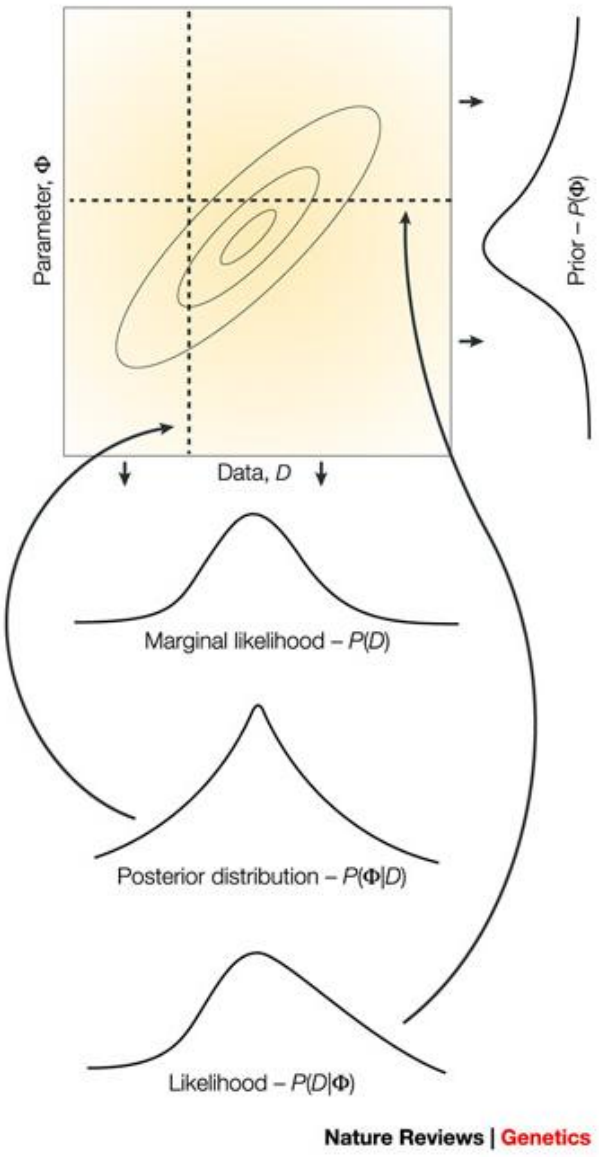
● The likelihood is a ***conditional distribution*** that specifies the probability of the observed data given any particular values for the parameters and is based on a model of the underlying process.

● Together these two functions combine all available information about the parameters.

● Bayesian statistics involves manipulating this joint distribution in various ways to make inference about the parameters, or the probability model, given the data.

● The main aim of Bayesian inference is to calculate the ***posterior distribution*** of the parameters, which is the conditional distribution of parameters given the data.

BASIC FEATURES THAT UNDERLIE BAYESIAN INFERENCE



- Data D can take any value that is measured along the x-axis.
- Similarly, the parameter value Φ can take any value that is measured along the y-axis.
- Bayesian inference involves creating the joint distribution of parameters and data, P(D, Φ), illustrated by the contour intervals in the figure. This distribution can be obtained simply as the product of the prior P(Φ) and the likelihood P(D|Φ).
- Typically, the likelihood will arise from a statistical model in which it is necessary to consider how the data can be 'explained' by the parameter(s).
- The prior is an assumed distribution of the parameter that is obtained from background knowledge.
- Marginal distributions are obtained by summing (integrating) the joint distribution either over the data, recovering the prior (the distribution on the right of the joint distribution), or over the values of the parameter, giving the marginal likelihood (the first distribution directly below the joint distribution).
- Conditional distributions ('|' in notation) are indicated by the dotted lines in the figure, and represent taking a 'slice' through the joint distribution and then rescaling the distribution so that the sum (integral) of possible values is equal to one. The scaling factor that is needed is given by the marginal distribution. Any conditional distribution is simply the joint distribution divided by a marginal distribution. For example, the likelihood can be recovered by dividing the joint distribution by the prior.
- The posterior distribution, P(Φ|D) — the key quantity in Bayesian inference — is the joint distribution divided by the marginal likelihood. It is the computation of the marginal likelihood (that is, the integrations denoted by the arrows that point down from the joint distribution) that is typically problematic.

From: Beaumont & Rannala, Nature Reviews Genetics 5:251-261.

A SHORT HISTORY OF BAYESIAN PHYLOGENY INFERENCE

● 1966, *Gomberg*, *"Bayesian" postdiction in an evolution process*. Unpublished manuscript, Pavia, Italy (referred to in the book by *J. Felsenstein, Inferring phylogenies*). A Bayesian approach to inferring phylogenies from characters that change according to a Brownian motion process.

● In 1970´ discussions about possibilities of using random models of branching and extinction to place priors on phylogenetic trees. Conclusions were that this was not computationally practical. Also, controversies between scientists that favor parsimony methods over all other methods and those that invoked Bayesian approaches.

● The first (almost) fully Bayesian approach is from 1987 by *Smouse* and *Li* (Evolution 41: 1162-1176). They used a birth-and-death process prior on the trees in an analysis of DNA sequences using molecular clock. For a fixed interval of time since tyhe start of the process, they inferred the birth, death and subsitution rates for a given substitution model (HKY) by finding values that maximized the posterior probabilities summed over all trees. Fixing these parameters at their estimated values, they used the probability contributed to the posterior by each tree topology as its posterior probability. This approach used numerical integration of the posterior probabilities over all interior node times for each given tree topology. As the number of topologies is huge even for quite small number of OTUs and there is the need to integrate over many dimensions, only a small number of OTUs could be used (in 1987, computers were not as they are now).

BAYESIAN PHYLOGENY INFERENCE, MrBAYES

● Currently the program package MrBayes http://mrbayes.csit.fsu.edu/ is the widely used approach in phylogeny data analysis.

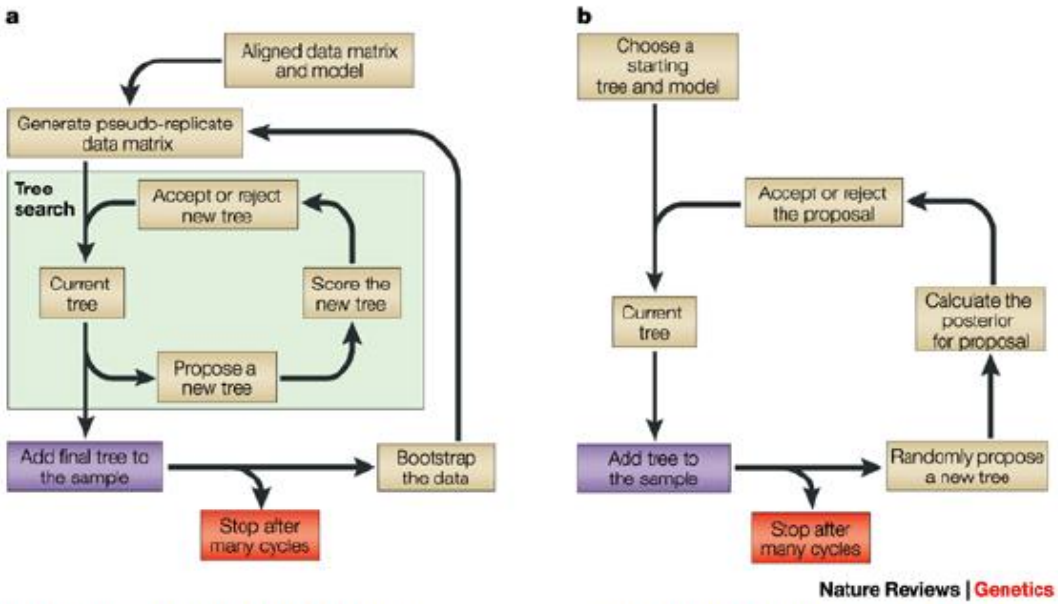
● Very practical, however, not necessarily easy, see this:
https://lists.sourceforge.net/lists/listinfo/mrbayes-users

● Has solved the following problem: The expression for the posterior distribution has a denominator that can be very difficult to compute. It involves summing up all possible hypotheses. Solution:

- Samples from the posterior distribution can be drawn by using a **Markov chain** that does not need to know the denominator.
- Markov chain **Monte Carlo** methods are now in widespread use in phylogeny inference (as well as in many other contexts).

● A Markov chain generates a series of random variables such that the probability distribution of future states is completely determined by the current state at any point in the chain. Under certain conditions, a Markov chain will have a **'stationary distribution**', meaning that if the chain is iterated for a sufficient period, the states it visits will tend to a specific probability distribution that no longer depends on the iteration number or the initial state of the variable.

BOOTSTRAPPING vs MCMC IN PHYLOGENY CONFIDENCE / CREDIBILITY



The bootstrapping approach .When optimality-criterion methods are used, a tree search (green box) is performed for each data set, and the resulting tree is added to the final collection of trees. A wide variety of tree-search strategies have been developed, but most are variants of the same basic strategy. An initial tree is chosen, either randomly or as the result of an algorithm — such as neighbour joining . Changes to this tree are proposed; the type of move can be selected randomly or the search can involve trying every possible variant of a particular type of move .The new tree is scored and possibly accepted. Some search strategies are strict hill-climbers — they never accept moves that result in lower scores; others (genetic algorithms or simulated annealing) occasionally accept worse trees in an attempt to explore the tree space more fully.

The Markov chain Monte Carlo (MCMC) methodology is similar to the tree-searching algorithm, but the rules are stricter. From an initial tree, a new tree is proposed. The moves that change the tree must involve a random choice that satisfies several conditions. The MCMC algorithm also specifies the rules for when to accept or reject a tree. MCMC yields a much larger sample of trees in the same computational time, because it produces one tree for every proposal cycle versus one tree per tree search (which assesses numerous alternative trees) in the traditional approach. However, the sample of trees produced by MCMC is highly auto-correlated. As a result, millions of cycles through MCMC are usually required, whereas many fewer (of the order of 1,000) bootstrap replicates are sufficient for most problems.

BAYESIAN PHYLOGENY INFERENCE

● MCMC: class of methods that rely on simulating a Markov chain, to study properties of a complicated probability distribution that cannot be easily studied using analytical methods.

● The basic idea that underlies all MCMC methods is to construct a Markov chain with a stationary distribution that is the probability distribution of interest, and then to sample from this distribution to make inferences.

● In Bayesian analysis, this distribution is usually the joint posterior distribution of one or more parameters.

● MCMC has also been used for estimating likelihoods and other purposes in maximum-likelihood inference.

● Example: If 96% of the samples from the posterior distribution of phylogenetic trees have (human, chimp) as a monophyletic group, then we can say that the probability that these are a monophyletic group is 96%.

MCMC

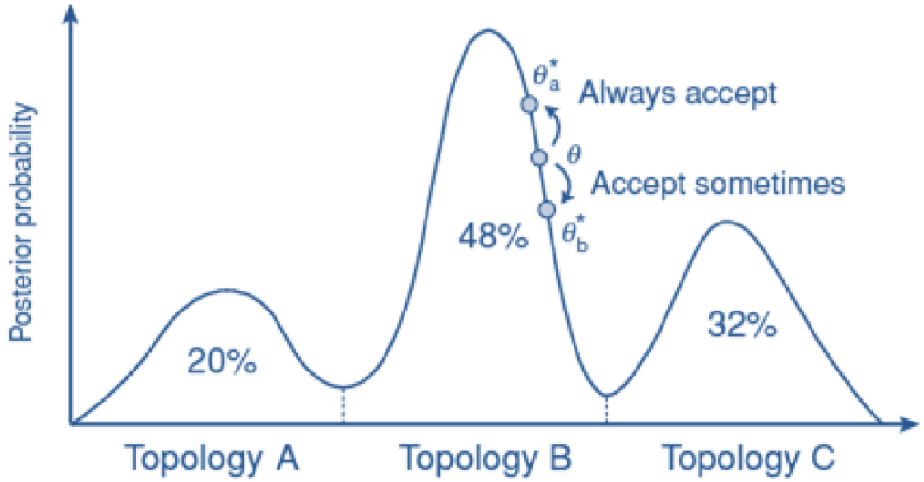
● The simplest form of MCMC is **Monte Carlo integration** which is widely applied in statistical genetics. The MC simulation method has the advantage that the estimates obtained are unbiased and the standard error of the estimates can be accurately estimated because the simulated random variables are i.i.d.

● The **Metropolis–Hastings** (MH) algorithm is similar to the MC simulation procedure in that it aims to sample from a stationary Markov chain to simulate observations from a probability distribution. Rather than simulating independent observations from the stationary distribution, it simulates sequential values from the chain until it converges and then samples simulated values at intervals from the chain to mimic independent samples from the stationary distribution.

● The MH algorithm has the advantage that it can improve the efficiency of simulations when the state space is large because it focuses the simulated variables on values with high probability in the stationary chain. Disadvantages include the fact that in most practical applications, there are no rigorous methods available to determine when the chain has converged or what the optimal intervals between samples are to extract the most information while preserving independence between observations.

Text is copied from:

MCMC



.From: Lemey *et al* ,
The phylogenetic handbook, 2009

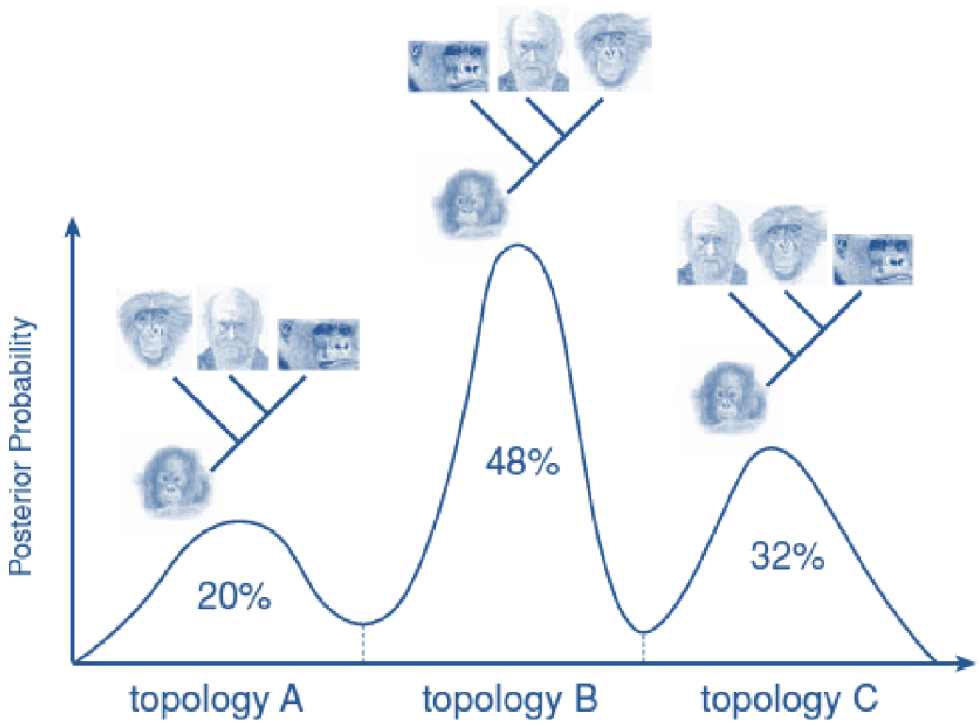
- MCMC is used to generate a valid sample from the posterior probability

- First, a Markov chain that has the posterior as its stationary distribution is set up.
The chain is then started at a random point and run until it converges onto this distribution.

- In each step (generation) of the chain, a small change is made to the current values of the model parameters (step 2).

- The ratio *r* of the posterior probability of the new and current states is then calculated
    - If $r > .1$, movement is uphill and the move is always accepted.
    - If $r < 1$, movement is downhill and the new state is accepted with probability *r.*

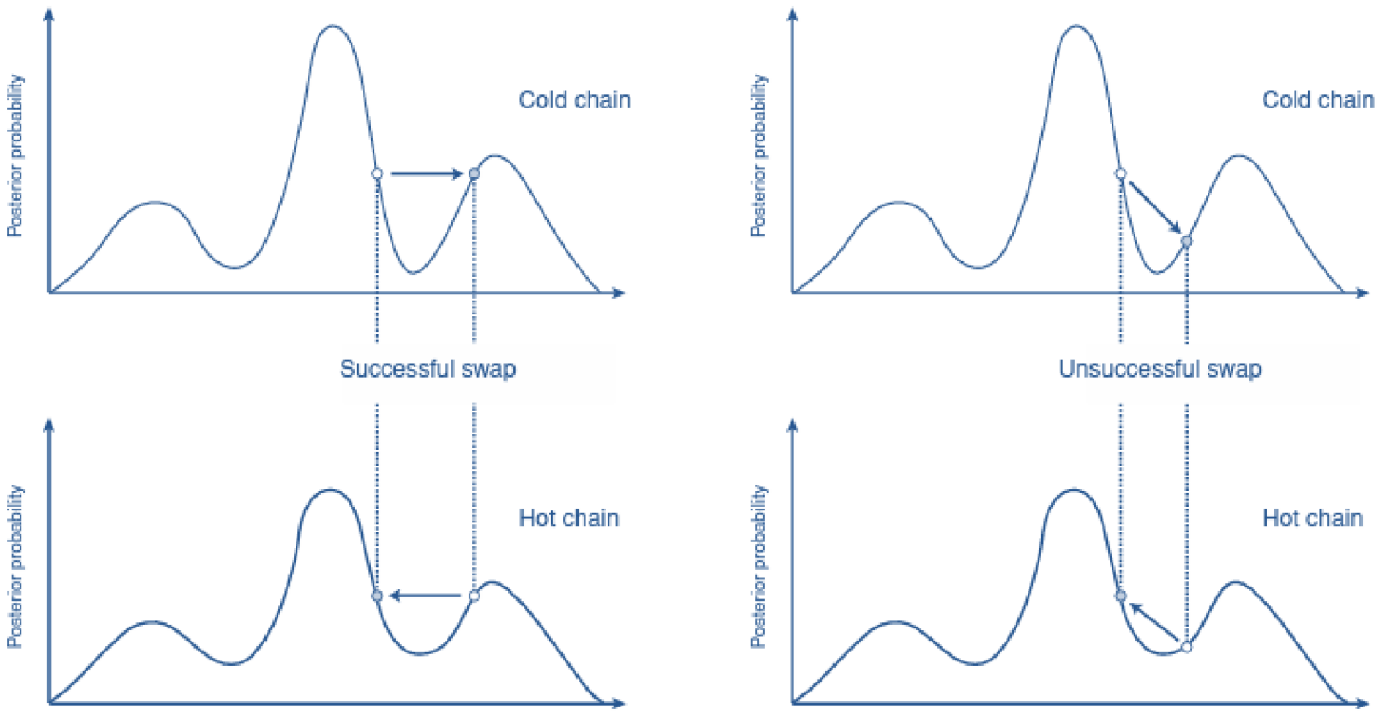
AN EXAMPLE OF TOPOLOGIES



From: Lemey *et al* ,
The phylogenetic handbook, 2009

● Posterior probability distribution for a phylogenetic analysis (human, chimpanzee, gorilla, orangutang). The x-axis is an imaginary one-dimensional representation of the parameter space. It falls into three different regions corresponding to the three different topologies. Within each region, a point along the axis corresponds to a particular set of branch lengths on that topology. It is difficult to arrange the space such that optimal branch length combinations for different topologies are close to each other. Therefore, the posterior distribution is multimodal. The are under the curve falling in each tree topology region is the posterior probability of that tree topology.

MCMCMC



From: Lemey *et al* ,
The phylogenetic handbook, 2009

● Metropolis coupling uses one or more *heated* chains to accelerate mixing in the so-called *cold* chain sampling from the posterior distribution. The heated chains are flattened out versions of the posterior, obtained by raising the posterior probability to a power smaller than one. The heated chains can move more readily between peaks in the landscape because the valleys between peaks are shallower. At regular intervals, one attempts to swap the states between chains. If a swap is accepted, the cold chain can jump between isolated peaks in the posterior in a single step, accelerating its mixing over complex posterior distributions.