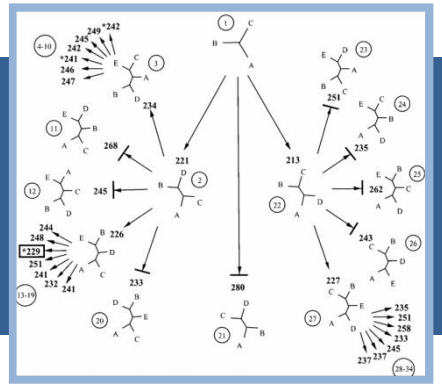
MAXIMUM PARSIMONY IN PHYLOGENY INFERENCE



- 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.

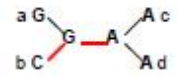
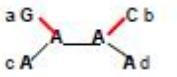
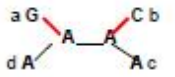
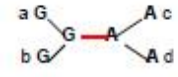
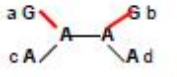
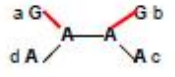
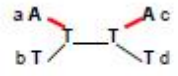
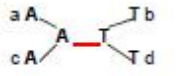
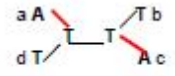
- Two subproblems:
  - Determining the amount of character change, or tree length, required by any given tree.
  - Searching over all possible tree topologies to find the tree that minimize this length.

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.

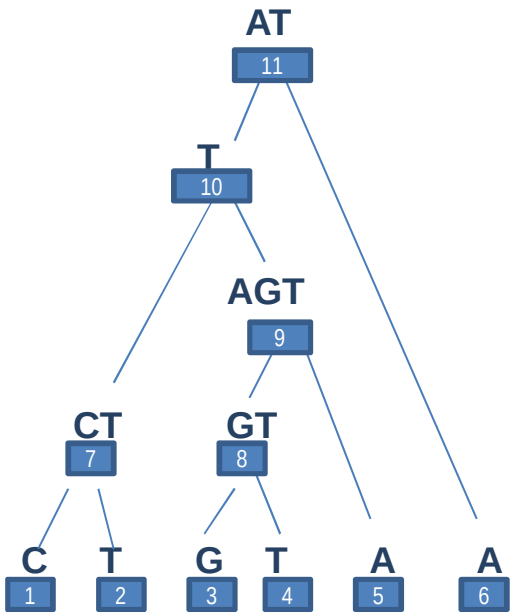
- The example next page (taken from textbook W-H Li, *Molecular evolution,* 1997) considers the case of six OTUs, and one particular *site*, at which the nucleotides are

....*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 favor many alternative trees.

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.

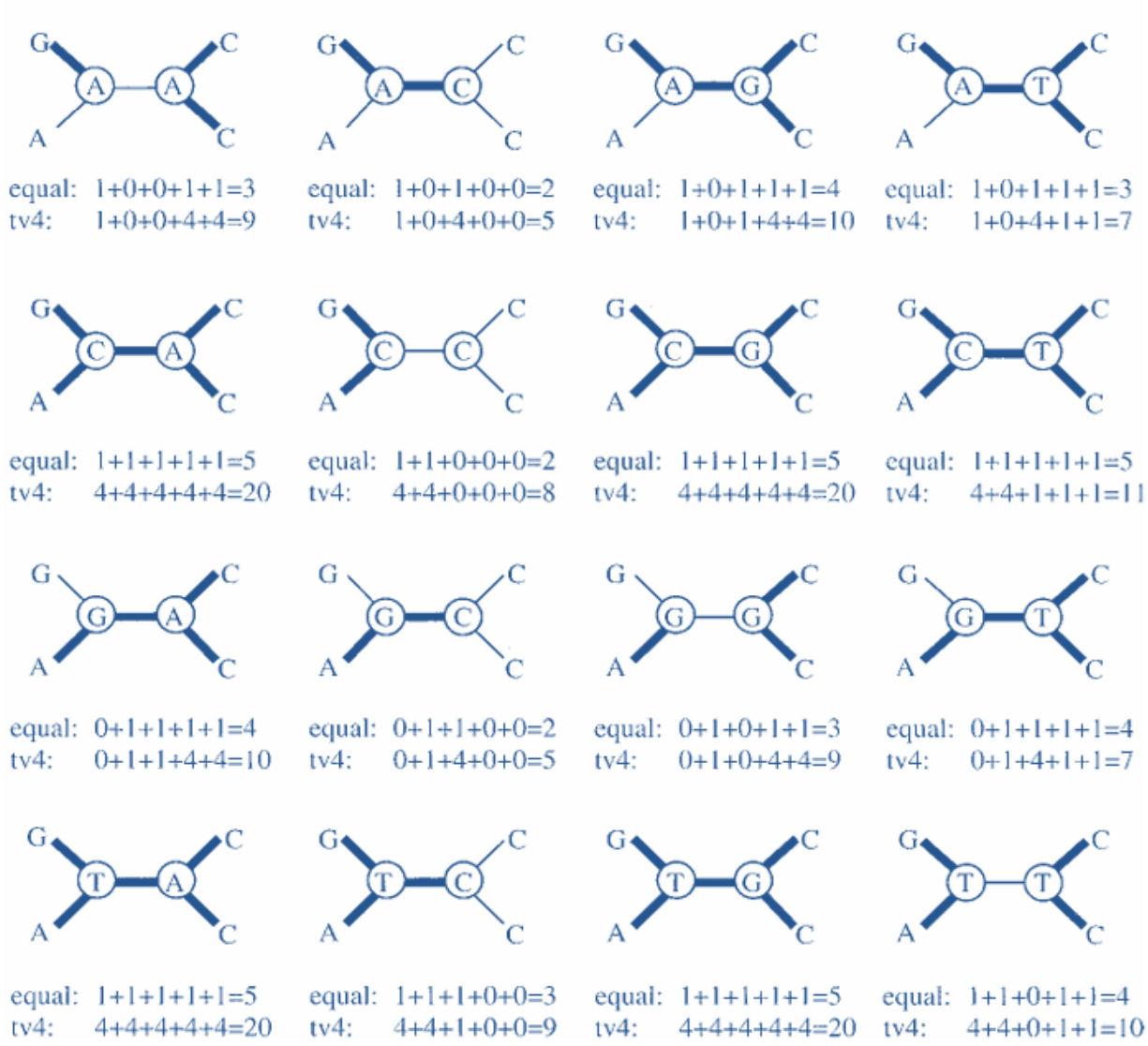
■ Next example (taken from Lemey *et al., The phylogenetic handbook, 2009*), calculation of tree length using "brute-force" approach of evaluating all possible character-state reconstructions. Four OTUs, W, X, Y and Z,

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



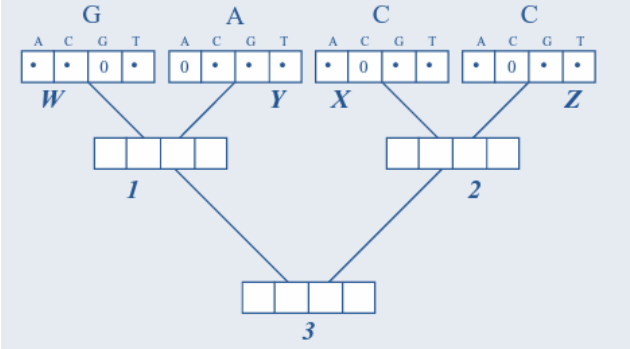
■ Two cost schemes, equal and transversions 4x weighted.

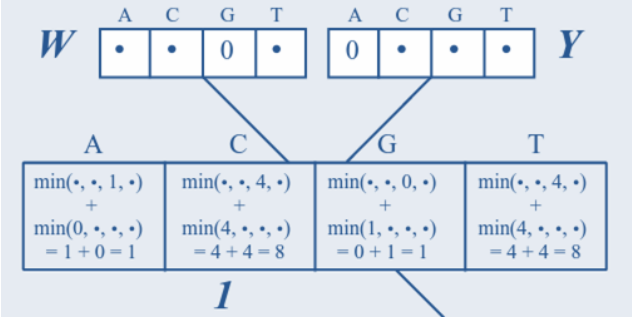
■ With equal costs, the minimum length is two steps and this length is achievable in three different ways, internal nodes assignment A-C, C-C and G-C. If a similar analysis for the other two possible trees, ((W,X),(Y,Z)) and ((W,Z),(Y,X)) is conducted, they are also found to have lengths of two steps. *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).





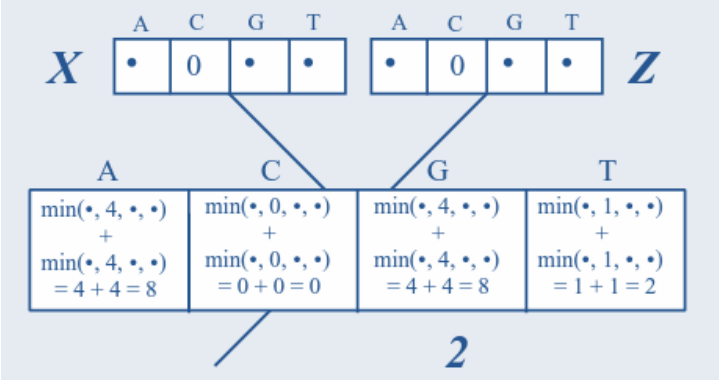
Node 1

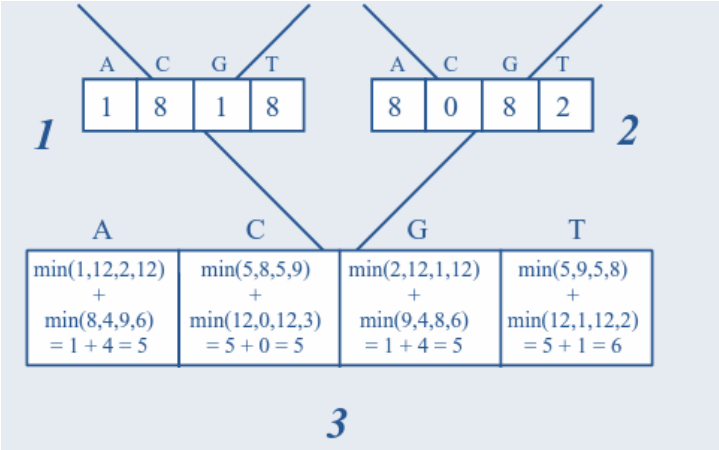
■ For each element *k* of this vector, consider the costs associated with each of the four possible assignments to each of the child nodes W and Y, and the cost needed to reach these states from state *k*, which is obtained from the cost matrix (4:1 transversion:transition is assumed here).

■ 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).

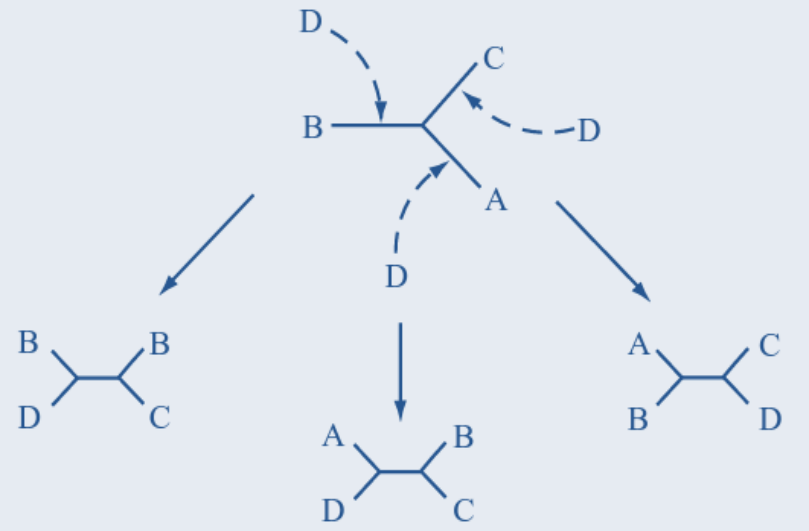
SANKOFF´s ALGORITHM - GENERATION OF ALL POSSIBLE TREES

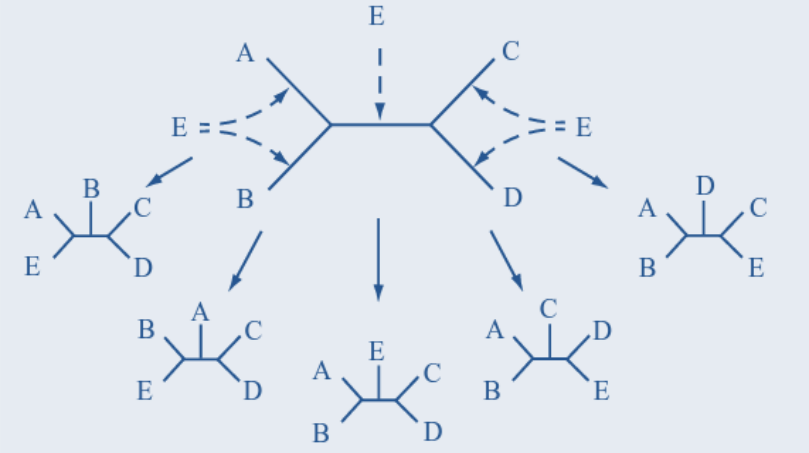
■ Sankoff´s algorithm provides a means of calculating the length required by any character on any tree under any cost scheme. The length of a given tree is obtained by repeating the procedure for each character and summing up all characters. In principle, the most parsimonious tree is found by generating 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.

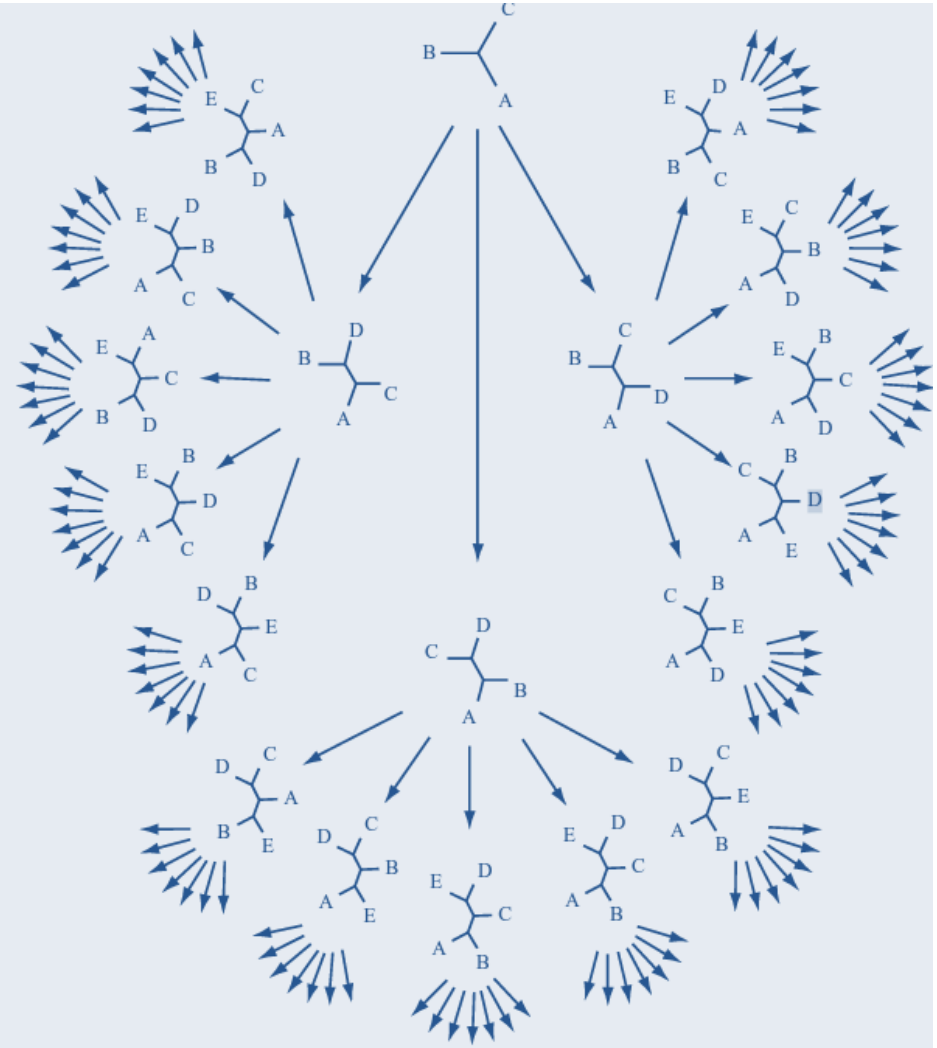
■ Below (next page, taken from Lemey *et al., The phylogenetic handbook, 2009*) one procedure: The algorithm recursively adds the *t*th OTU in a stepwise fashion to all possible trees containing the first *t* – 1 OTUs until all *n* OTUs have been joined. For rooted trees the algorithm is modified by 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.

GENERATION OF ALL POSSIBLE TREES





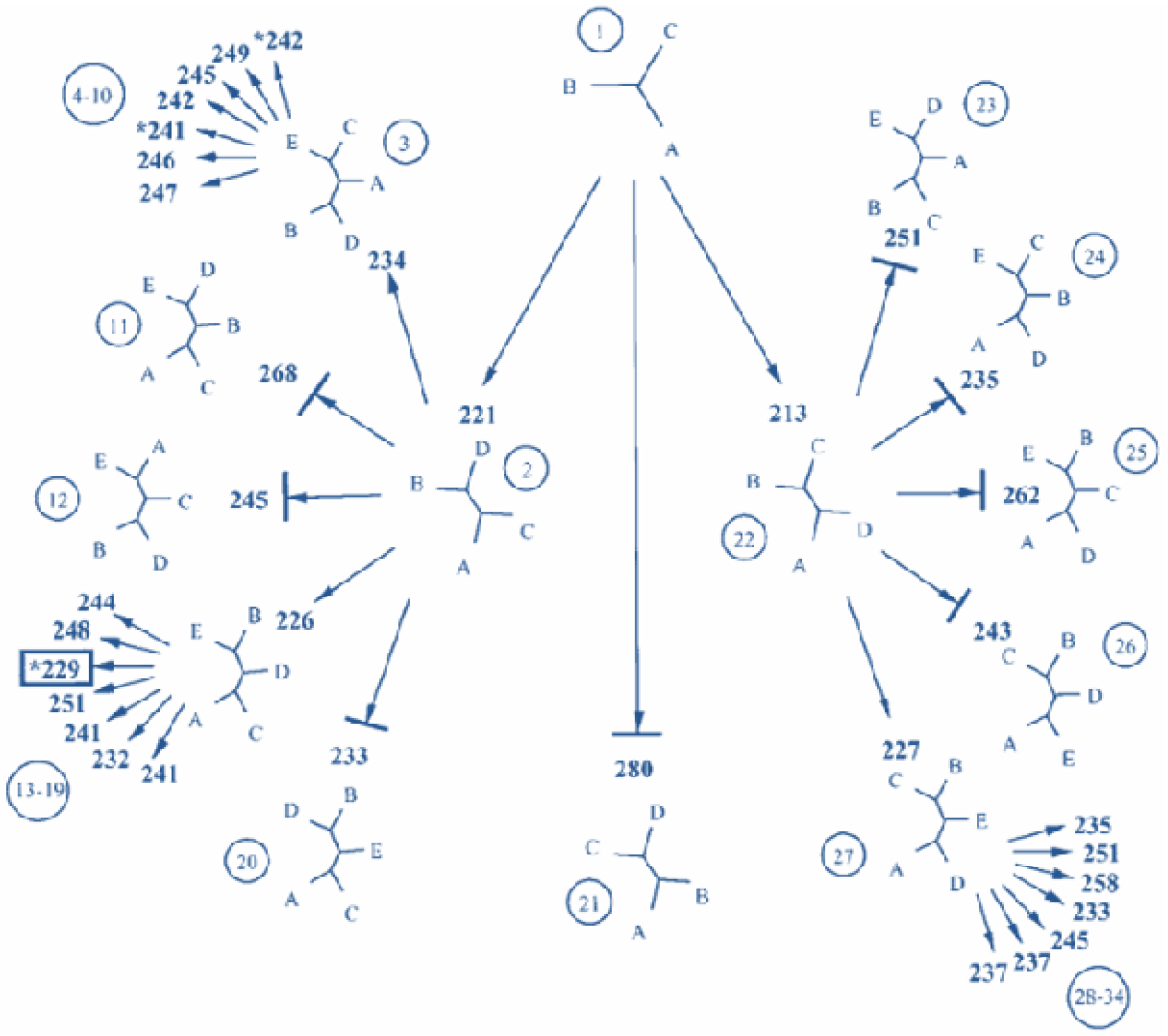


BRANCH-AND-BOUND ALGORITHM, AN EXACT METHOD by HENDY & PENNY (1982)

■ 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 (see 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)

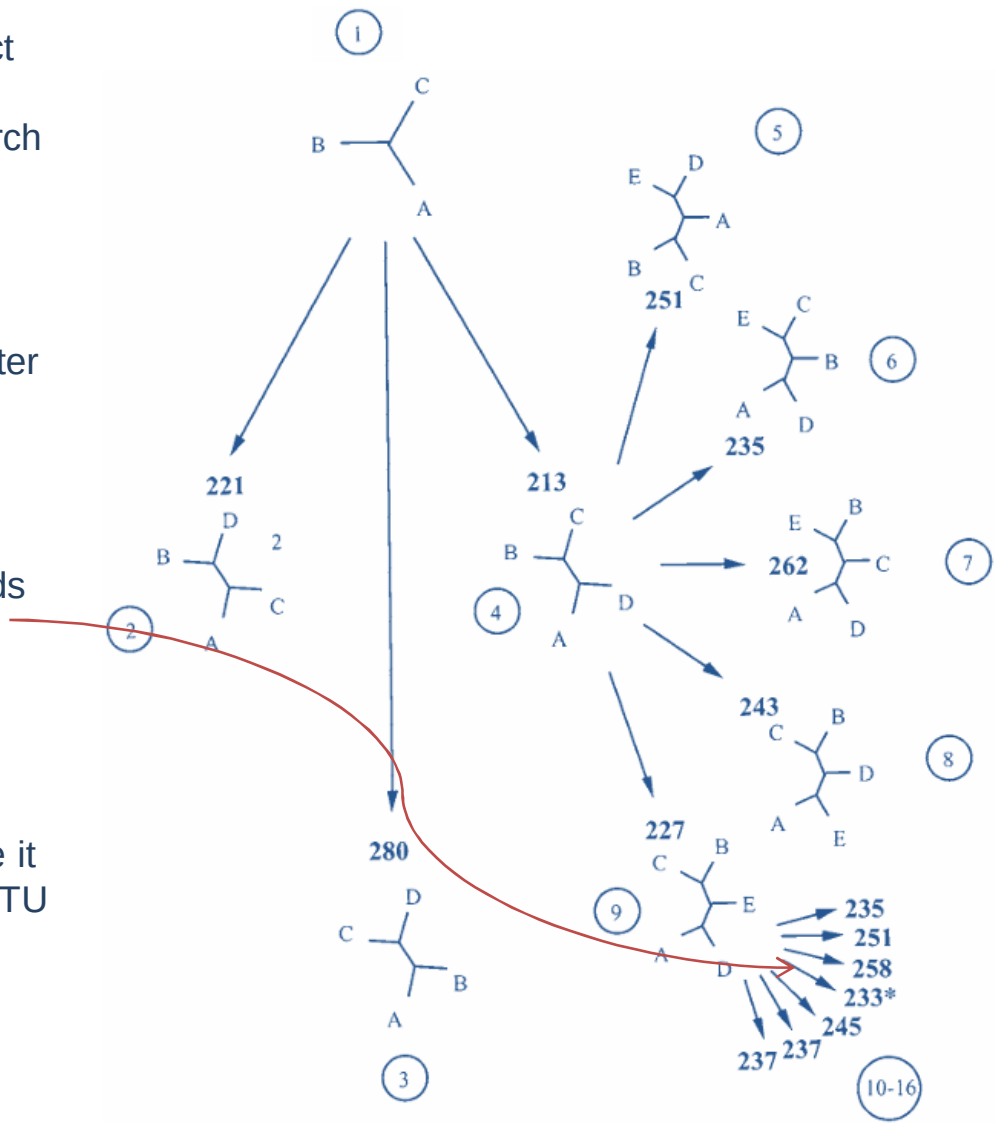
■ Follows the same kind of search tree as the branch-and-bound method, but unlike the exact exhaustive enumeration, stepwise addition commits to a path out of each node on the search tree that looks most promising at the moment. *This might not lead to a global optimum.*

■ 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)

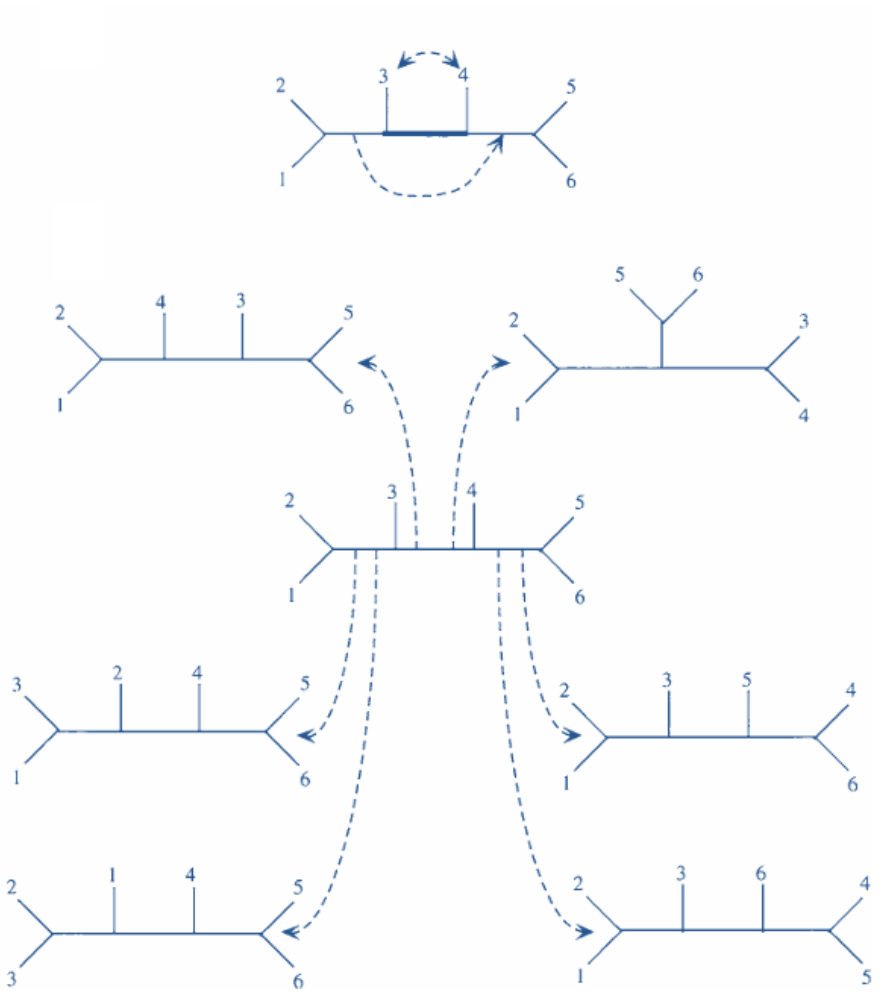
■ Branch-swapping methods involve cutting off one or more pieces of a tree (subtrees) and reassembling them in a way that is locally different from the original tree.

■ 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).



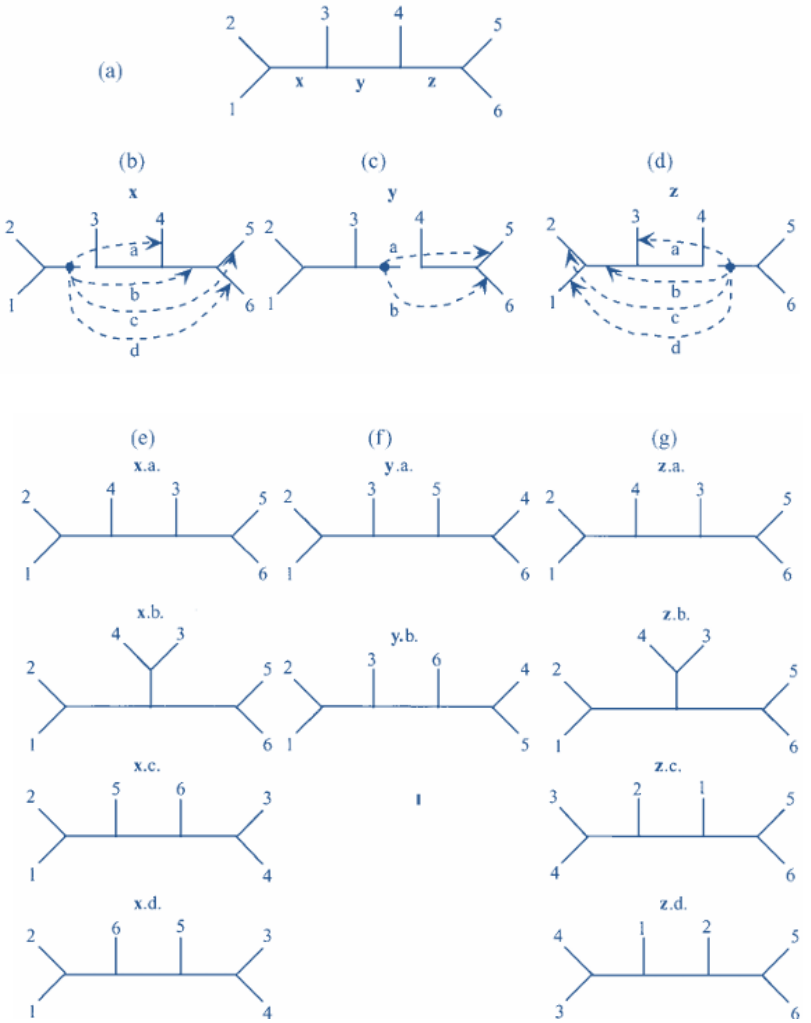
HEURISTICS BY BRANCH-SWAPPING: SUBTREE PRUNING AND REFRAFTING (SPR)

■ 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

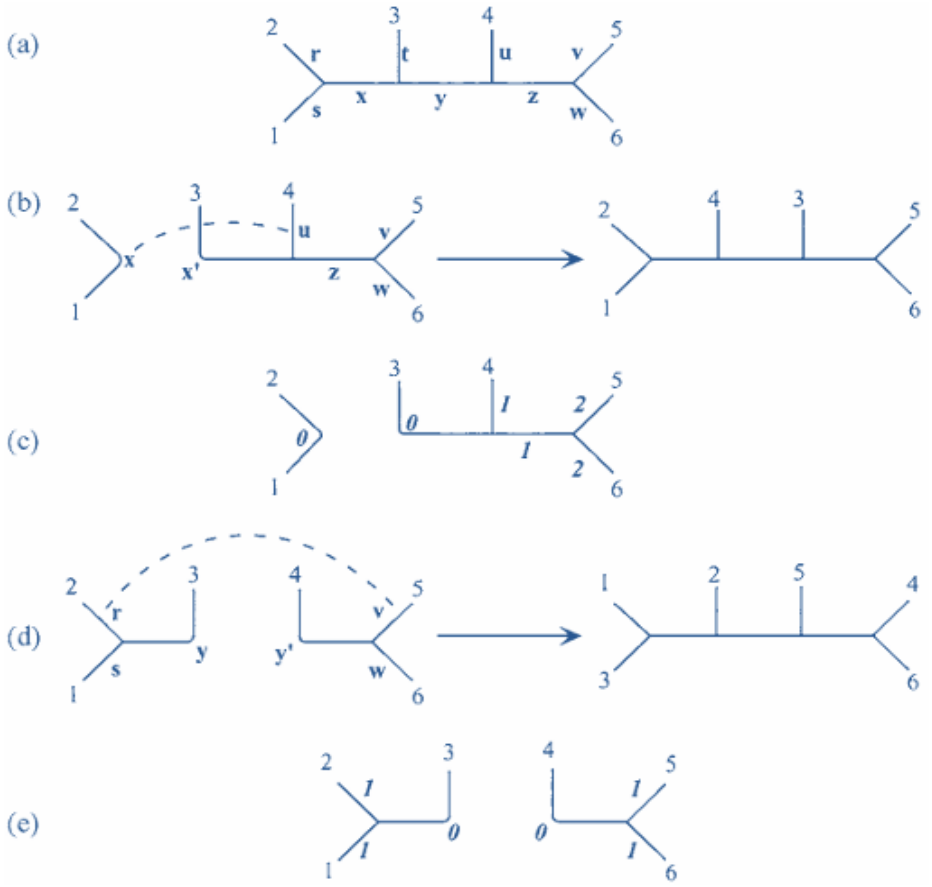
(b), (c), (d)
SPRs resulting from pruning of branches, x, y, z, respectively. In addition to these rearrangements, all terminal OTUs (leaves) would be pruned and reinserted elsewhere on the tree.

(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:

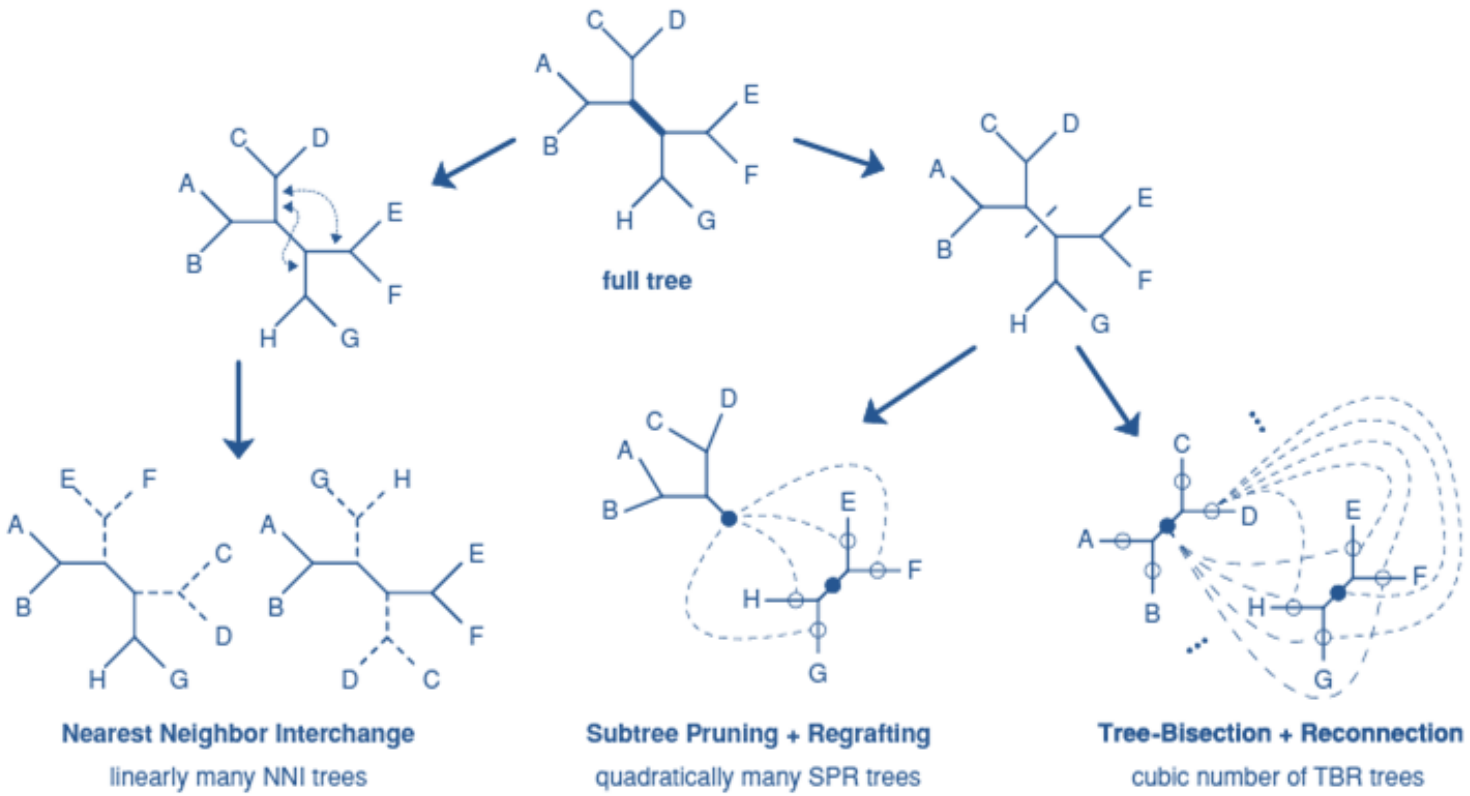
- NNIs are the subset of TBRs that have a reconnection distance of 1.
- SPRs are the subset of TBRs so that exactly one of the two reconnected branches is numbered zero.
- TBRs that are neither NNIs nor SPRs are those for which both reconnected branches have non-zero numbers.

■ 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

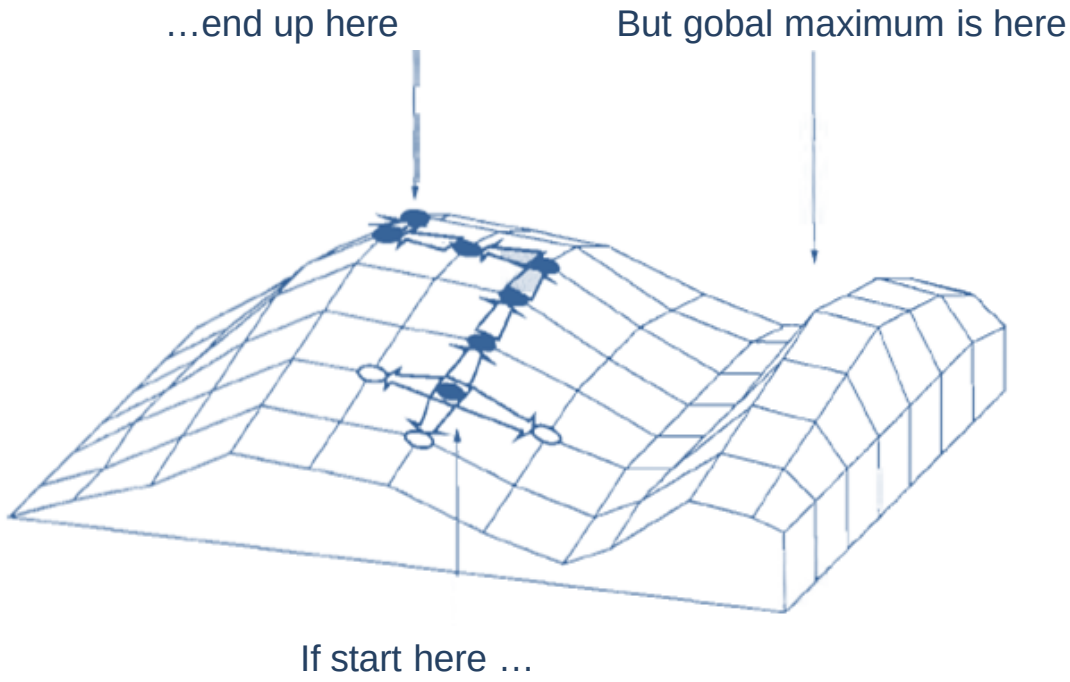
- Optionally, when trees are found that are equal in score to the current tree (e.g. equally parsimonious trees or trees that have identical likelihoods within round-off error), they are appended to a list of optimal trees.
    - In this case, when the arrangement of one tree finishes, the next tree 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 obtai ned 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.