Lecture 15

Phylogeny I: Parsimony, and Tree Space — DRAFT

February 24, 1998
Lecturer: Joe Felsenstein
Notes: Ka Yee Yeung

General references on phylogeny are Swofford and Olsen 1996 cite..., Li 1998 cite..., and Felsenstein 1998 cite....

15.1. Phylogenies

Phylogenies are branching diagrams that show the history of life. A phylogeny show the ancestral relationships among a set of species. Note that in a phylogeny tree, the internal nodes may refer to hypothetical ancestors. When a hypothetical common ancestor diverges, the two different species are assumed to evolve independently of each other.

There are many methods to infer phylogenies given statistical data. Consider the following fictitious example:

Species	site 1	site 2	site 3	site 4	site 5	site 6
Aardvark	C	A	G	G	T	A
Bison	C	A	G	A	C	A
Chimp	C	G	G	G	T	A
Dog	T	G	C	A	C	T
Elephant	T	G	C	G	T	A

A possible phylogeny tree corresponding to site 1 is shown in Figure 15.1.

15.2. Parsimony

Parsimony is one way to find a phylogeny, given the data. A phylogeny constructed with the method of parsimony explains evolution with the fewest evolutionary changes.

The problem of finding a phylogeny, given the data, can be divided into two subproblems:

1. How to evaluate how good a given tree is.

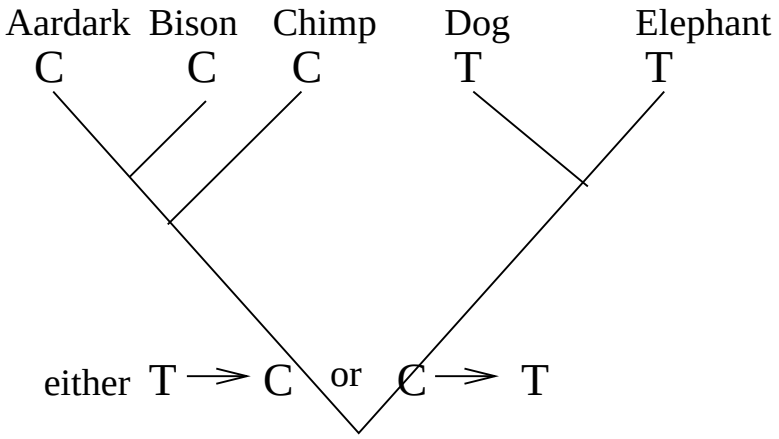

Figure 15.1: A phylogeny corresponding to site 1 in Table 15.1.

2. How to search through all possible trees efficiently.

The first subproblem is known as the *small parsimony* problem, and the second subproblem is known as the *large parsimony* problem.

15.2.1. The Small Parsimony Problem

The small parsimony problem is to determine the cost or score of a given phylogenetic tree. The cost or score of a phylogeny refers to the number of changes required to explain the data. There exists polynomial time algorithms to solve the small parsimony problem.

Let us re-visit the data given in Table 15.1. The phylogeny shown in Figure 15.1 corresponds to a phylogeny for site 1. Only one change (either C is changed to T or T is changed to C) is required to explain the data at site 1.

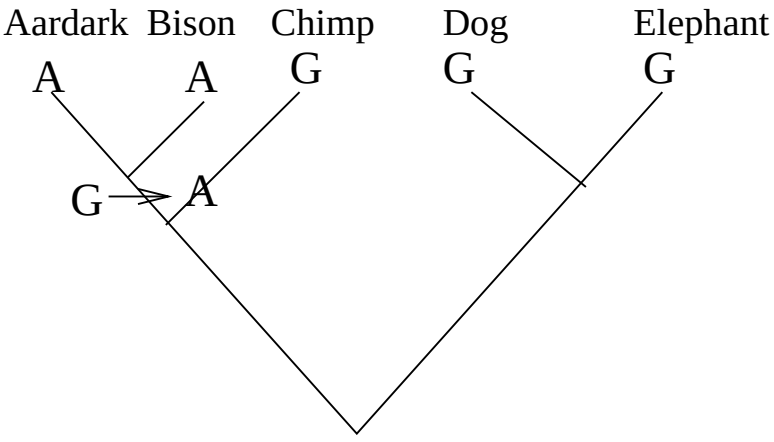

Figure 15.2: A phylogeny corresponding to site 2 in Table 15.1.

Figure 15.2.1 shows a phylogeny corresponding to site 2 in Table 15.1. Again, only one change (from

G to A) is needed. If we do the same for all sites, we get a total of 8 changes required as shown in Figure 15.2.1. Two changes are needed for both site 4 and site 5.

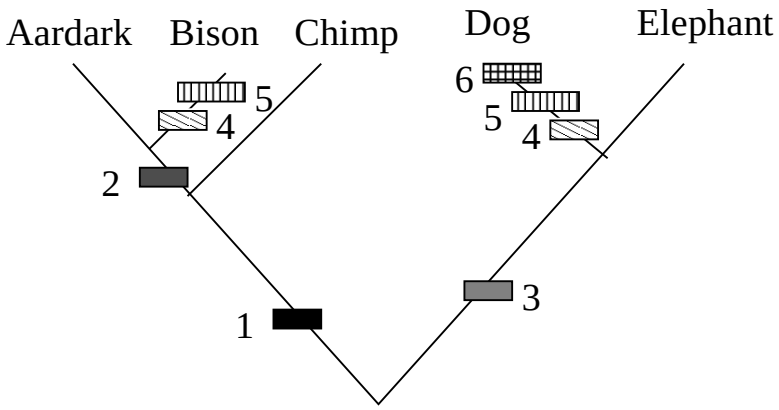

Figure 15.3: A phylogeny corresponding to all the sites in Table 15.1.

Given another phylogenetic tree as shown in Figure 15.2.1, 11 changes are required to explain the data. Therefore, the same data set can result in different phylogenies which in turn have different costs or scores.

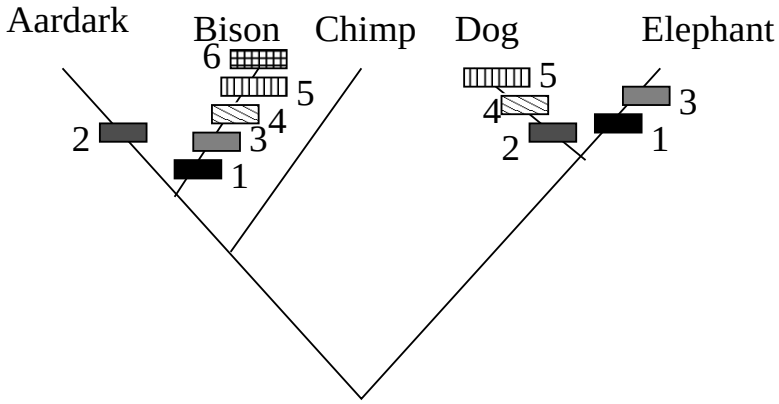

Figure 15.4: An alternative phylogeny corresponding to all the sites in Table 15.1.

Phylogenies are usually *rooted* trees. If the root is changed, the same number of changes are needed to explain the data as illustrated in Figure 15.2.1. In other words, the principle of parsimony is independent of where the root is in the tree.

Now, the problem is how to determine the cost or score of a given phylogenetic tree efficiently.

Two ways of viewing phylogentic trees

- A phylogenetic tree can be viewed as a member of a space of trees. Consider a phylogeny on four species. There are a total of 22 possible phylogenetic trees for four species. Each phylogeny has a numerical score indicating how many changes are needed to explain the data. Hence, each tree can be represented as a point in the tree space.

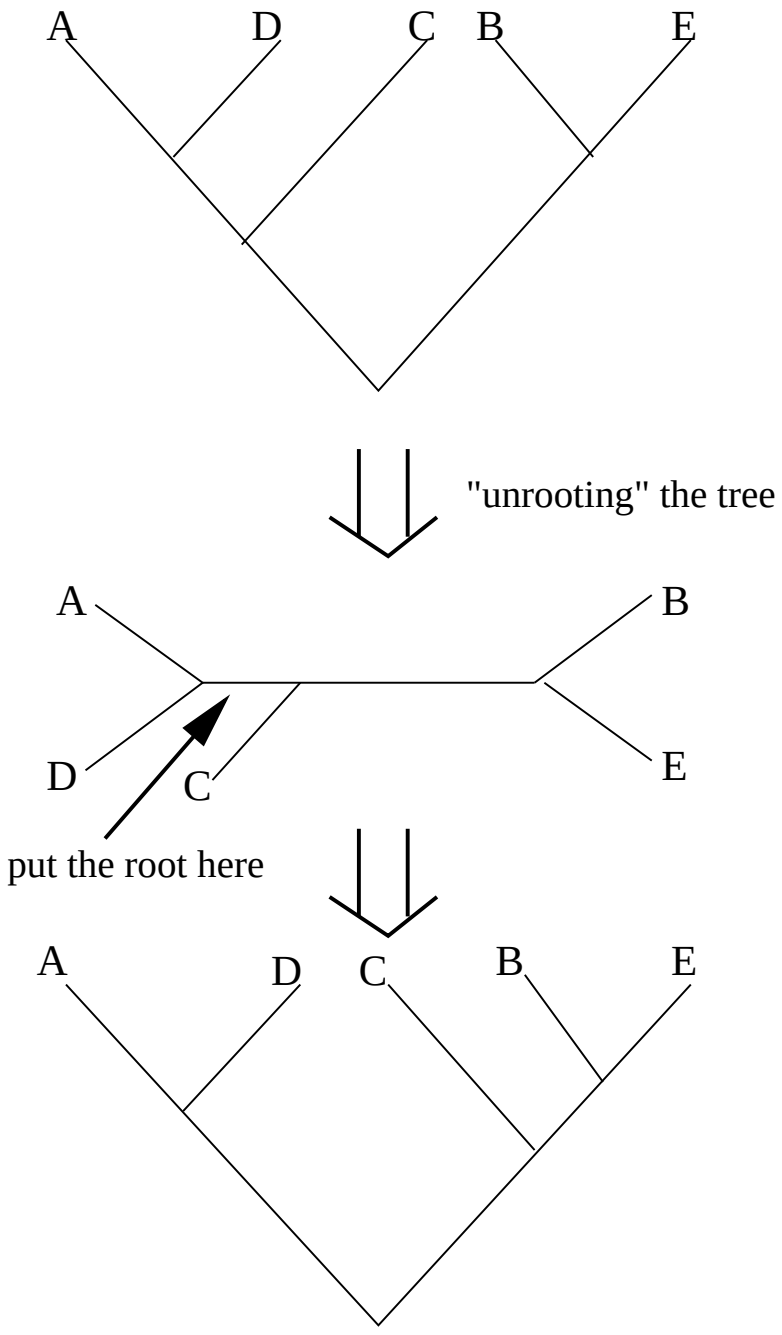

Figure 15.5: Changing the root in a given phylogeny.

- A phylogeny can also be viewed as a tree connecting points in a data space such that the total length of the tree is minimized. This is equivalent to the *Steiner tree* problem on a given graph as illustrated in Figure 15.2.1.

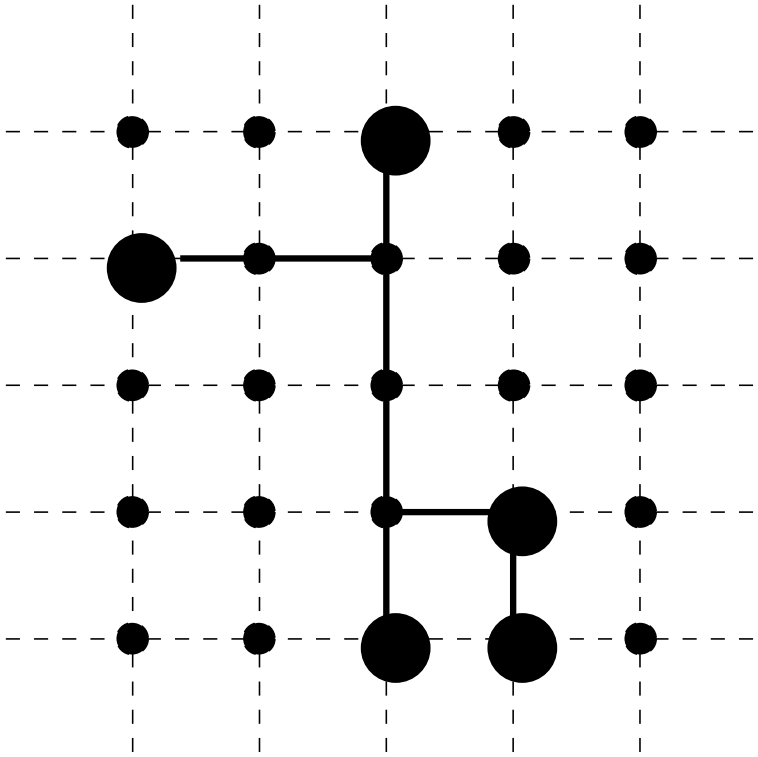

Figure 15.6: An example of a steiner tree on a data set.

Now, let us look at two algorithms which can be used to determine the cost or score of a given phylogeny.

Fitch's Algorithm

Fitch algorithm considers each site in a given DNA sequence separately. It computes the minimum number of changes required to explain evolution of a given phylogeny representing a site in a DNA sequence. It is a dynamic programming style algorithm.

The algorithm constructs a set of possible states (possible nucleotides) for each internal node, which is computed from the states of its children. We start at the leaves of the phylogeny. Each leaf is labeled with the singleton set containing the nucleotide at that particular site. Then the tree is traversed in a postorder manner (such that all of the children of the current node have been visited before the current node). In general, if m is an internal node with children l and r having states S_l and S_r respectively. The state of m, S_m, is computed as follows:

$$S_m = \begin{cases} S_l \cup S_r & \text{if } S_l \cap S_r \text{ is empty} \\ S_l \cap S_r & \text{otherwise} \end{cases}$$

Each application of the first rule $S_l \cup S_r$ contributes one count to the number of changes. An example of Fitch's algorithm is shown in Figure 15.2.1. The asterisk indicates the application of the first rule, and hence count one step. Summing up all the counts at each internal node gives the total number of changes for the given phylogeny. In the example, the minimum number of changes is 3, as indicated by 3 asterisks.

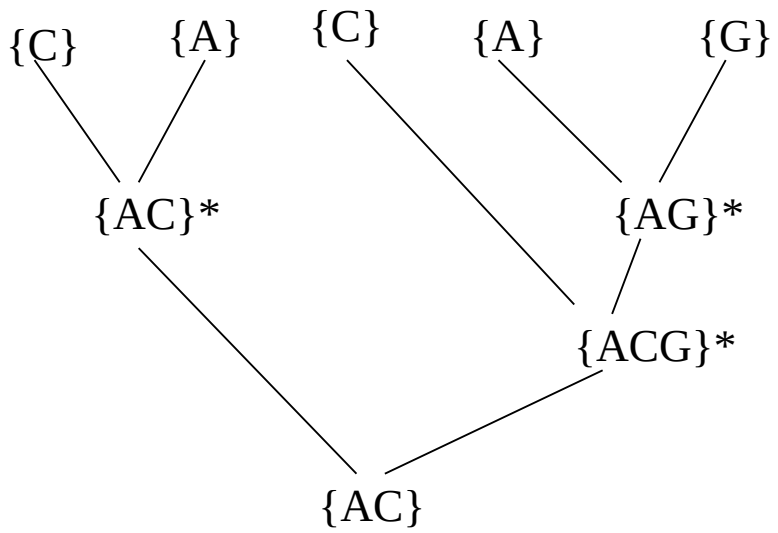

Figure 15.7: An example illustrating Fitch's Algorithm.

Sankoff's Algorithm

Sankoff's algorithm is a generalization of Fitch's algorithm in which arbitrary states and different costs between states are allowed. Suppose there are k possible states with costs $s_1, s_2, \ldots, s_k$ at each site (k = 4 for DNA sequences), and there is a cost matrix c, where c_{ij} is the cost of changing from state i to state j. Sankoff's algorithm works for arbitrary cost matrices, even if the cost of changing from state i to j is different than that from state j to i.

Sankoff's algorithm also starts at the leaves of the tree. At each leave, the cost of state i is zero if that state is being observed, and the cost is infinity otherwise. In the case of DNA sequences, the state corresponding to a nucleotide α is zero if that nucleotide is observed, otherwise the cost is assigned to be infinity. The cost of state i at node A, $s_i^{(A)}$, with children nodes L and R can be computed as follows:

$$s_i^{(A)} = \min_m (c_{im} + s_m^{(L)}) + \min_n (c_{in} + s_n^{(R)})$$

The cost of the entire tree is the minimum of the costs of all states at the root of the tree. Figure 15.2.1 is an example illustrating Sankoff's algorithm. In the following example, the cost of the entire tree is 6.

In the case of DNA sequences, nucleotides A and G are known as *purines* and nucleotides C and T are known as *pyrimidines*. Change of states within the same category is called *translation*, and change of states between purines and pyrimidines is called *transversion*. Translations are known to be easier than transversions. Hence, it is reasonable to assign lower costs for translational changes between A and G, and between C and T than transversion changes.

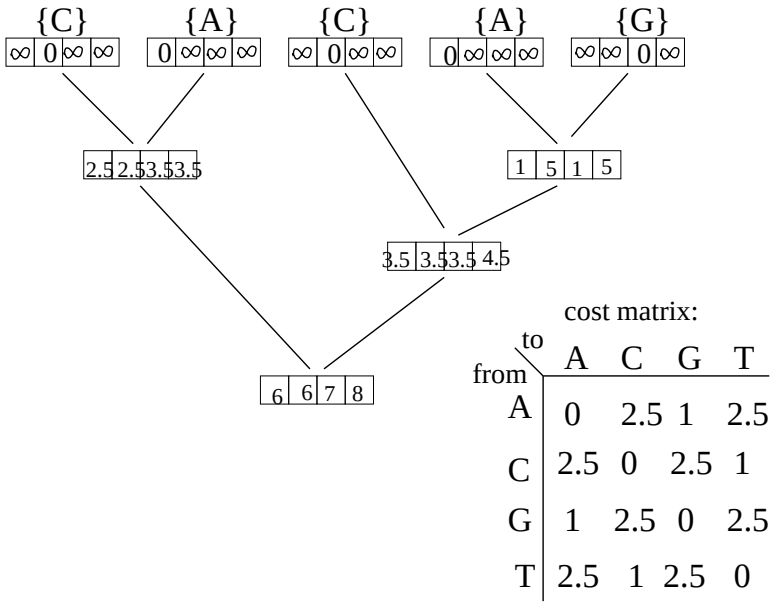

Figure 15.8: An example illustrating Sankoff's algorithm.

15.2.2. The Large Parsimony Problem

Given that we have a way of determining the cost or score of a given tree, we now consider the combinatorial problem of finding the best tree that explains the given data. In the large parsimony problem, we have to search through the space of all possible trees to find the best tree.

Assume that in evolution, each split is a two-way split, giving rise to bifurcating trees. We can either consider the space consisting of rooted or unrooted tree because it can be shown that the number of rooted trees with n leaves is equal to the number of unrooted trees with $n + 1$ leaves.

Consider a rooted tree with 2 leaves. There is only one such tree. There are 3 possible branches where we can add an extra leave, giving a rooted tree with 3 leaves. Given a rooted tree with 3 leaves, there are 5 branches where we can add the fourth leaf. Proceeding in this way, it can be shown that there are $1 * 3 * 5 * \ldots * (2n - 3) = (2n - 3)!!$ rooted bifurcating trees with labeled tips and unlabeled interior nodes. It is obvious that the number of bifurcating rooted trees grows exponentially with the number of leaves as shown in the following table cite ....:

n	rooted bifurcating	rooted multifurcating
2	1	1
3	3	4
4	15	26
5	105	236
6	945	
7	10,395	
8		660,032
10	34,459,455	282,137,824
15		$6.35 * 10^{15}$
20	$\approx 10^{21}$	$8.87 * 10^{23}$

The large parsimony problem is proved to be NP-complete cite...

15.3. Heuristics for the Large Parsimony Problem

15.3.1. Branch and Bound

Hendy and Penny cite.. proposed the branch and bound method to search through the tree space for the most parsimonious tree. Their algorithm considers unrooted trees. Starting with an unrooted tree with 3 tips, it computes the number of steps required to explain all the sites with the current tree, and then estimate the minimum number of changes required for an additional species. One crude way to do the estimation would be to count one step for each new state for the new species at each site. The estimate would be the sum over all the sites for adding the new species. At each step of the algorithm, if the sum of the number of steps for the current tree and the estimated minimum number of steps exceeds the current minimum number of steps found, the algorithm aborts on the branch starting at the current node.

15.3.2. Nearest Neighbour Interchanges (NNI)

The idea of NNI is to rearrange a tree by dissolving the connections between subtrees to form different trees. Consider the following example, the connections between subtrees S, T, U and V are dissolved. There are 3 possible subtrees that can be formed in this case: U next to V as shown in the left figure, or U can be connected to T as in the right figure, and the original tree.

If we repeat the above procedure for all possible subtrees for five tips, we would form a non-planar graph, in which each possible tree is connected to four others. The graph is known as *Peterson's graph*. A hill-climbing local search technique can be applied on this search space by always moving to the best immediate neighbour until no better neighbour exists.

15.3.3. Subtree Pruning and Regrafting (SPR)

Maddison cite... proposed the following heuristic: for each possible subtree, disconnect a branch and remove the subtree T_1 from the original tree T. Denote the original tree with the subtree T_1 removed as T_2.

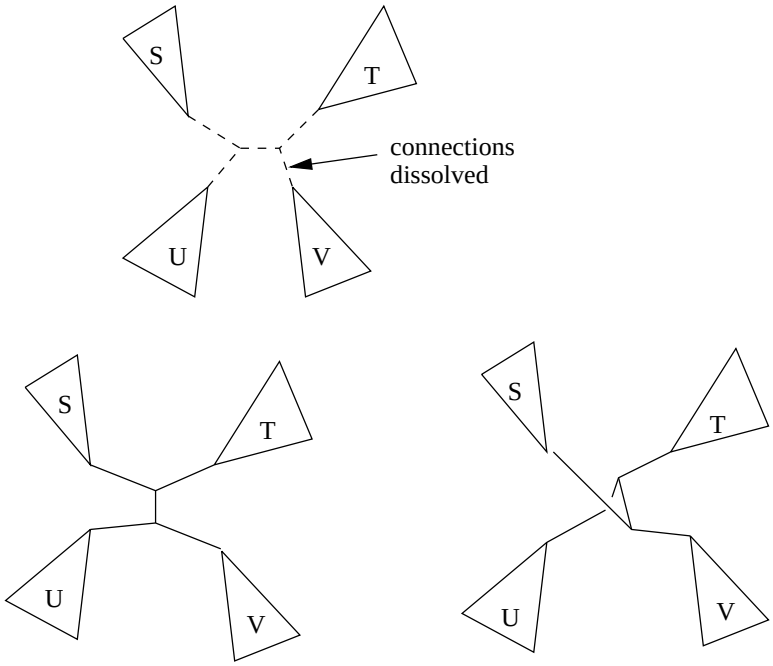

Figure 15.9: An example illustrating NNI.

The next step is to add T_1 back to T_2 by attaching it to any of the possible branches of T_2.

15.3.4. Tree Bisection and Reconnection (TBR)

The idea for TBR is to remove an internal branch of the original tree T and separate the subtree T_1. Again, let us call the tree with a branch and subtree T_1 removed T_2. Then connect the subtree T_1 to T_2 by adding a branch between any branch in T_1 and any branch in T_2. This method usually gives rise to more neighbours and thus have a higher chance of finding a good tree.

15.3.5. Phylip

Phylip is a software package written to solve the large parsimony problem. The idea is to start with 2 tips, then try to add in additional species, and then continue to add further species to the best tree obtained so far.

Computational Biology Winter, 1998
CSE 590bi J. Felsenstein

References on phylogeny methods

General introductions

Swofford, D. L, G. J. Olsen, P. J. Waddell, and D. M. Hillis. 1996. Phylogenetic inference. pp. 407-514 in *Molecular Systematics*, 2nd ed. ed. D. M. Hillis, C. Moritz, and B. K. Mable. Sinauer Associates, Sunderland, Massachusetts.

Li, W.-H. 1998. *Molecular Evolution.* Sinauer Associates. Sunderland, Massachusetts.

Felsenstein, J. 1998. *Inferring Phylogenies*. ASUW Publishing, HUB 113. (partial preprint of a book, available at start of spring quarter)

Pioneering papers:

Edwards, A. W. F. and L. L. Cavalli-Sforza. 1964. Reconstruction of evolutionary trees. pp. 67-76 in *Phenetic and Phylogenetic Classification,* ed. V. H. Heywood and J. McNeill. Systematics Association Publ. No. 6, London.

Camin, J. H., Sokal, R. R. 1965. A method for deducing branching sequences in phylogeny. *Evolution* **19:** 311-326.

Fitch's and Sankoff's algorithms for the Small Parsimony problem:

Fitch, W. M. 1971. Toward defining the course of evolution: minimum change for a specified tree topology. *Systematic Zoology* **20:** 406-416

Sankoff, D. D. 1975. Minimal mutation trees of sequences. *SIAM Journal of Applied Mathematics* **28:** 35-42.

Sankoff, D. D., and P. Rousseau, P. 1975. Locating the vertices of a Steiner tree in arbitrary space. *Mathematical Programming* **9:** 240-246.

The Pairwise Compatibility theorem:

Estabrook, G. F., C. S. Johnson, Jr., and F. R. McMorris. 1976. A mathematical foundation for the analysis of character compatibility. *Mathematical Biosciences* **23:** 181-187.

Parsimony and compatibility NP-nasty:

Foulds, L. R. and R. L. Graham. 1982. The Steiner problem in phylogeny is NP-complete. *Advances in Applied Mathematics* **3:** 43-49.

Graham, R. L. and L. R. Foulds. 1982. Unlikelihood that minimal phylogenies for a realistic biological study can be constructed in reasonable computational time. *Mathematical Biosciences* **60:** 133-142.

Day, W. H. E. 1986. Computational complexity of inferring phylogenies by compatibility. *Systematic Zoology* **35:** 224-229.

Trying Branch and Bound:

Hendy, M. D. and D. Penny. 1982. Branch and bound algorithms to determine minimal evolutionary trees. *Mathematical Biosciences* **60:** 133-142.

Swofford, D. L. and G. J. Olsen. 1990. Phylogeny reconstruction. Chapter 11, Pp. 411-501 *in* D. M. Hillis and C. Moritz, eds. Molecular Systematics. Sinauer Associates, Sunderland, Massachusetts.

Heuristic rearrangement strategies

Maddison, D. R. 1991. The discovery and importance of multiple islands of most-parsimonious trees. *Systematic Zoology* **40:** 315-328.

Distance matrix methods

Fitch, W. M., and E. Margoliash. 1967. Construction of phylogenetic trees. *Science* **155:** 279-284.

N. Saitou and M. Nei. 1987. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution* **4:** 406-425.

Likelihood methods:

Felsenstein, J. 1981. Evolutionary trees from DNA sequences: a maximum likelihood approach. *Journal of Molecular Evolution* **17:** 368-376.

Consensus trees

Adams, E. N., III. 1972. Consensus techniques and the comparison of taxonomic trees. *Systematic Zoology* **21:** 390-397.

Margush, T., and F. R. McMorris. 1981. Consensus n-trees. *Bulletin of Mathematical Biology* **43:** 239-244.

Adams, E. N., III. 1986. N-trees as nestings: complexity, similarity, and consensus. *Journal of Classification* **3:** 299-317.

Trees and alignment

Sankoff, D. D., C. Morel, and R. J. Cedergren. 1973. Evolution of 5S RNA and the nonrandomness of base replacement. *Nature New Biology* **245:** 232-234.

Sankoff D. D., and R. J. Cedergren. 1983. Simultaneous comparison of three or more sequences related by a tree. pp. 253-263 in *Time Warps, String Edits, and Macromolecules: the Theory and Practice of Sequence Comparison*, ed. D. Sankoff and J. B. Kruskal, Addison-Wesley, Reading, Massachusetts.