Stat 882: Statistical Phylogenetics – Lecture 3

Laura S. Kubatko
lkubatko@stat.osu.edu

Contents

1 Introduction: Optimality Criteria

Optimality Criteria

- Thus far, we have described two criteria for evaluating phylogenetic trees
    - Parsimony: Prefer the tree(s) that minimize evolutionary change
        - Compute a score for each tree based on a cost associated with each type of change
    - Maximum likelihood: Prefer the tree(s) that have the highest likelihood
        - Compute the likelihood of each tree under a particular substitution model
- Using these criteria, we have a method for comparing different phylogenies

Optimality Criteria – Inference

- Recall that our goal is to infer a phylogeny
- To do this, we must search the set of possible phylogenies for the tree(s) that gives the best value of the selected criterion
- Recall that the space of phylogenies is very large - consider an unrooted tree with T taxa. There are
    - $T-2$ internal nodes
    - $2T-3$ branches
    - Number of possible trees is $\prod_{i=1}^{T-2}(2i-1)$

Number of Taxa	Number of Rooted Trees	Number of Unrooted Trees
5	105	15
10	34,459,425	2,027,025
20	8.2×10^{21}	2.21×10^{20}
50	2.75×10^{76}	2.83×10^{74}

2 Methods for Finding Optimal Trees: Exact Methods

Methods for Finding Optimal Trees

- Exact Methods
    - Exhaustive search
    - Branch and bound methods

- Heuristic methods
    - Divide-and-conquer
    - Stepwise addition and branch swapping
    - Parsimony ratchet
    - Numerous other possibilities

- Stochastic searches
    - Simulated annealing
    - Genetic algorithms

2.1 Exhaustive Search

Exhaustive Search

- Enumerate all possible trees

- Evaluate the criterion of interest on all trees

- Pick the tree that gives the optimal value of the criterion

- Advantage: Complete information about the problem

- Disadvantage: Not computationally feasible for large numbers of taxa

2.2 Branch and Bound

Branch and Bound

- Can be used with any criterion whose values are non-decreasing as taxa are added to the tree

- Basic idea: Eliminate portions of the tree space that do not contain the optimal tree, so that the criterion need never be evaluated for these trees

- Advantage: Guaranteed to find optimal tree

- Disadvantages:
  - Don't necessarily give info about near-optimal trees
  - May still be very time consuming
  - Limited to approx. 20 taxa or less

Branch and Bound – An Example

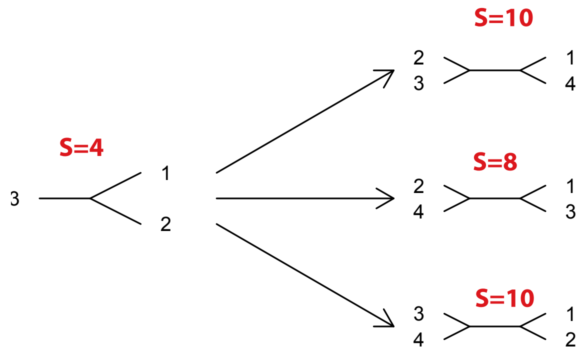
- Consider the following data simple set:

Taxon	Site Pattern						
Name	1	2	3	4	5	6	7
1	A	A	C	C	A	C	A
2	A	T	C	G	T	G	C
3	A	T	C	G	A	C	A
4	C	T	T	T	T	C	C
5	C	G	T	T	T	C	C

- Suppose we want to find the tree with the lowest score, S, under Fitch parsimony

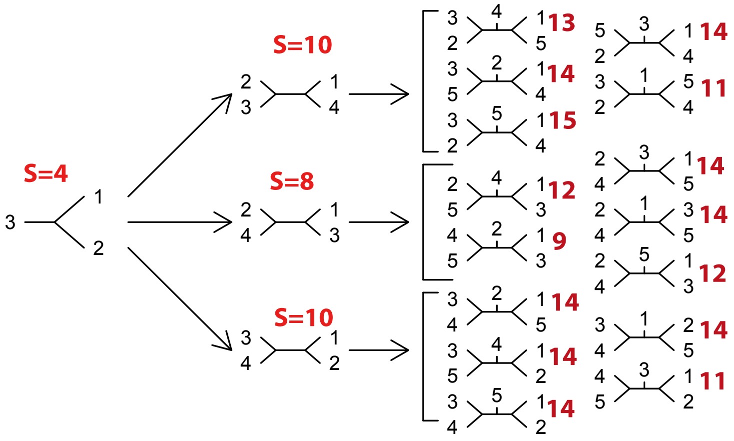
Branch and Bound – An Example

- Use an initial, quick search to find a tree whose length is 9
- Begin with three taxa, then add the fourth in all possible locations; evaluate all scores



Branch and Bound – An Example

- Then add the fifth taxon



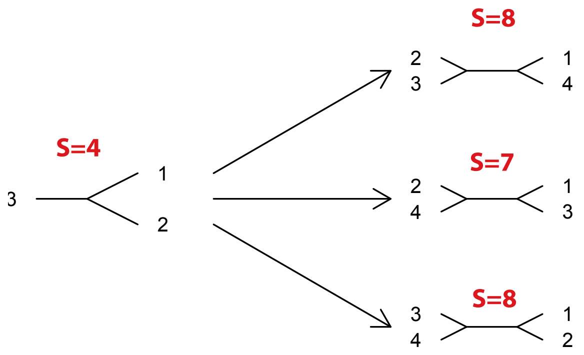
Branch and Bound – Worst Case

- Suppose instead our data were only the first 6 sites:

Taxon	Site Pattern					
Name	1	2	3	4	5	6
1	A	A	C	C	A	C
2	A	T	C	G	T	G
3	A	T	C	G	A	C
4	C	T	T	T	T	C
5	C	G	T	T	T	C

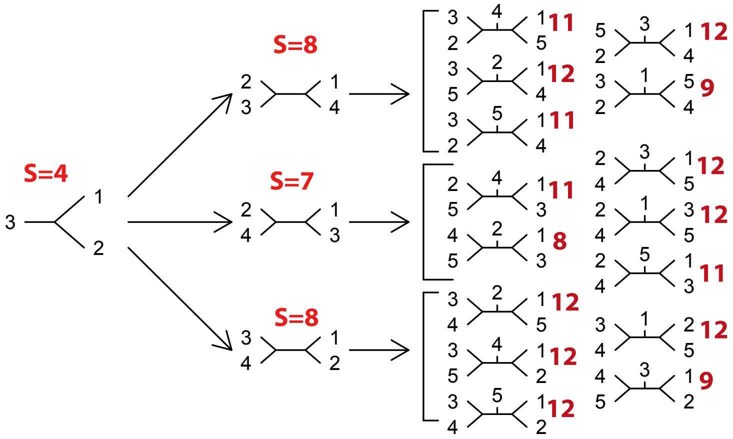
Branch and Bound – An Example

- Use an initial, quick search to find a tree whose length is 8
- Begin with three taxa, then add the fourth in all possible locations; evaluate all scores



Branch and Bound – An Example

- Then add the fifth taxon – note that now all trees must be evaluated



3 Methods for Finding Optimal Trees: Heuristic Methods

3.1 Divide-and-conquer Methods

Divide-and-Conquer Methods

- Divide the collection of taxa into subgroups
- Infer optimal phylogenetic trees for each of the subgroups
- Reassemble the subtrees into an overall supertree that includes all of the taxa
- Advantage: Quick
- Disadvantages:
  - How do we divide the taxa into subgroups?
  - How do we create supertrees from subtrees, especially if there is incongruence?

Divide-and-Conquer Methods

- Some of the most widely-used methods in this class are:
  - Disk-covering methods – Tandy Warnow's group Reference: U. Roshan, B. M. E. Moret, T. L. Williams, T. Warnow, Rec-I-DCM3: A Fast Algorithmic Technique for Reconstructing Large Phylogenetic Trees, Proceedings of the IEEE Computational Systems Bioinformatics (CSB04) Stanford (CA), USA, 2004
  - Quartet puzzling – implemented in PAUP* Reference: Strimmer, K. and A. von Haeseler. 1996. Quartet Puzzling: A Quartet Maximum-Likelihood Method for Reconstructing Tree Topologies. Molecular Biology and Evolution, Vol 13(7): 964-969.

3.2 Stepwise Addition and Branch Swapping

Stepwise Addition and Branch Swapping

- Most commonly used estimation method (PHYLIP, PAUP*, fastDNAML)
- Form an initial tree by stepwise addition of taxa

- At each step, perform rearrangements of the tree until no improvement in optimality criterion can be made

- Several strategies for performing rearrangements
    - Nearest Neighbor Interchagnes (NNI)
    - Subtree pruning and regrafting (SPR)
    - Tree bisection and reconnectin (TBR)

Stepwise Addition and Branch Swapping

- Advantages:
    - Easy to understand
    - Implemented in software
    - Works well for small sample sizes – e.g., < 50 taxon

- Disadvantages:
    - No guarantee of finding optimal tree (order of addition of taxa matters)
    - One proposed solution: Perform the search from several starting points (orderings of the taxa)
    - Computationally intensive

3.3 Ratchet

Ratchet

- Originally proposed for parsimony (Kevin Nixon, 1999, Cladistics)
- Extended to likelihood (Rutger Vos, 2003, Systematic Biology)
- Alogrithm:
    - Generate a starting tree by some quick method
    - Reweight a randomly selected subset of characters (e.g., give a weight of 2 to 50% of the characters and a weight of 1 to the other 50%)
    - Search on the current tree using any strategy (e.g., branch swapping)
    - Set all characters back to their original weights and search again for the current tree
    - Repeat these steps for many iterations

Ratchet

- Advantages:
  - Reweighting of characters allows the algorithm to explore more of the tree space than standard heuristics
  - Gain information about locally optimal trees
  - Easily implemented in PAUP* – Sikes and Lewis, PAUPRat

- Disadvantages:
  - Need to decide how many iterations to use and how weights should be selected
  - No guarantee of finding optimal tree

4 Methods for Finding Optimal Trees: Stochastic Methods

4.1 Simulated Annealing

Simulated Annealing

- A general method of function optimization
- Basic idea:
  - Move through the space of all trees by randomly rearranging a current tree to form a new tree.
  - The criterion of interest is evaluated on the new tree, and a decision is made about whether the new tree should be accepted as the current tree.
  - The key is that even new trees with worse values of the optimality criterion can be accepted - hopefully, this helps avoid finding only local optima.

Simulated Annealing

- Three steps

  1. From tree τ_i, generate candidate tree τ^* via a randomly-selected NNI move.

2. If $L(\tau^*) \geq L(\tau_i)$, set $\tau_i = \tau^*$. Otherwise, set $\tau_i = \tau^*$ with probability $exp\{\frac{L(\tau^*)-L(\tau_i)}{c_i}\}$.

3. Update the value of the control parameter, c_i, and set i to $i+1$. Go to step 1.

Simulated Annealing

- Advantages:
  - Quick, and can handle large data sets
  - Increased ability to find globally optimal tree
  - Easy to implement
  - Gives information about many trees

- Disadvantages:
  - Many parameters must be specified
  - No guarantee of finding optimal trees

4.2 Genetic Algorithms

Genetic Algorithms

- A general method of function optimization
- Basic idea:
  - Model the search for the optimal phylogenetic tree after the process of natural selection. Natural selection allows differential survival rates of individuals based upon their fitness.
  - This is applied to the tree search problem by letting a particular trees fitness be represented by its value of the optimality criterion. Trees with better values of the criteria are more likely to proceed to the next generation.
  - After many generations are simulated, we will hopefully have found the optimal tree.

Genetic Algorithms

- Some details:
  - Begin with a population of trees
  - Compute values of the optimality criterion for each tree

  - Mutation, natural selection, recombination, etc., act on each generation of the population to produce the next generation
    - Examples: Branch lengths may be mutated by multiplication by a factor selected from a gamma distribution Topology might be mutated by performing an SPR rearrangement
  - The tree with the highest value of the criterion is automatically placed in the next generation
  - Repeat the process for many generations

Genetic Algorithms

- Advantages:
  - Quick
  - Increased ability to find globally optimal trees
  - Potential information about many other trees
  - Implemented in the user-friendly program GARLI (Zwickl, 2006)

- Disadvantages:
  - Many parameters must be specified
  - No guarantee of finding optimal trees

A few other programs to mention ....

- RAxML - Randomly Axelerated Maximum Likelihood
  - Author: Alexandros Stamatakis
  - Website: http://icwww.epfl.ch/~stamatak/index-Dateien/Page443.htm

- TNT - Tree analysis using New Technology
  - Author: Pablo Goloboff, James Farris, Kevin Nixon
  - Website: http://www.cladistics.org/tnt.html