

## **Trees II: Parsimony**

### **I. Parsimony, Patterns & Processes**

Hennig provided fundamental methods for the use of character data to form phylogenies and made the relationship between character evidence and cladograms explicit in a way that had not been done previously.

He was not, as far as I know, concerned with parsimony as an optimality criterion, but rather his general paradigm was consistent with parsimony as a guiding principle.

Hennig’s Auxiliary Principle –*to assume homology if there is no evidence to suggest otherwise*— requires that we seek hypotheses maximizing our homologies or conversely, that overturn as few as possible of our initial homologies, given that these initial hypotheses are the result of careful character analysis. The result is to minimize ad hoc hypotheses needed to explain when we fail.

Pattern Cladists- put forward that cladistic (in this case = strict parsimony) methods do not need and in fact are better off without an evolution (process) justification. Three things are needed to justify building trees base on synapomorphies, 1. discoverability of characters, 2. hierarchy is the best representation of the natural world and 3. parsimony as an epistemological approach (Brower, A.V.Z. 2000. Evolution Is Not a Necessary Assumption of Cladistics. Cladistics 16, 143–154.)

Circularity? Mitter (1981. "Cladistics" in botany. Syst. Zool., 30:373–376.) "there is widespread (but not universal) agreement that ... systematic methods should be as free as possible from assumptions about how evolution works, because these assumptions are in general not testable without reference to systematic results."

In the “estimation school of thought” this problem is addressed through assessment of the impact of model violation and statistical support. Therefore ML and methods of that sort benefit from an evolutionary model based justification.

For “reconstructionists” evolutionary models are definitely part of the character analysis and are used with caution in tree building.

Though the clear understanding patterns we observe and summarize and processes that explain such patterns is important (multiple substitutions and insertions/deletions are inferred events --processes-- not observations.). However, a strict pattern view, which denies a role for evolution, does not provide a good explanation as to why any given character should or should not be included in an analysis.

## II. Parsimony as an optimality criterion.

-We know that there are a huge number of possible cladograms for any modest number of OTUs.

-We know that there is character conflict, i.e. character state distributions support groups that are not compatible.

So....we need some way to choose among the universe of possible trees. Parsimony can be used as an optimality criterion for an overall best fit =minimum length, steps or cost.

**III. How do we measure length?** ---- A character has a length that is the number of independent origins of character states on any given cladogram. This is measured as steps or costs and is weighted depending on the model assumptions.

Character and tree consistency indices:  $c_i = m/s$  and  $CI = \sum m / \sum s$

Two kinds of equally parsimonious trees: 1. same topology but different character state distributions (optimization) 2. Different topologies.

Optimizations: ACCTRAN and DELTRAN. Two “extremes” of optimization that may alter the resolution of the tree and the implied transformational history of the character.

*A priori* weight.

Equal weights is viewed by some as a minimal necessary assumption. Differential weighting, or at least the *a priori* selection of a certain difference introduces an unacceptable subjectivity. Exactly what the difference in weight should be cannot be unambiguously answered.

Others argue that equal weight for all characters and character state changes is a whopper of an assumption. Imagine 100 columns of DNA data 40 columns have only A and G, 40 columns have only C and T and 20 are mixed. The observation suggests that transitions occur more frequently than transversions. As such a 1:1 weighting is more of an assumption than 2:1 or 4:1. According to some, at least for molecular data, a slight weighting in the right direction is much better than none at all.

An extreme and not generally used scheme is character compatibility weighting.

*A posteriori* weighting:

1. Character consistency weighting. (aka, successive approximations weighting, Farris 1969)
2. Implied weighting (Goloboff 1993)

## IV. Tree Searching

1. Enumeration- look at every possible cladogram and sum length of all characters. Of course, this is not practical for any significant matrix. Thus we need heuristic methods and methods to escape local optima...
2. Basic strategy of a search
  - a. Get a near optimal tree
    - i. This is often done by randomly putting three OTUs in a network and adding a fourth to the edge that creates the shortest four OTUs network and so on until all OTUs are joined. This does not guarantee a shortest tree, but it usually is not too bad for a start.
  - b. Take starting tree and make small rearrangements to get nearby trees (think of trees as being in tree-space with similar ones close together)
  - c. If one of these nearby trees is better (shorter) then retain it, discard the old one and make rearrangements on the new tree.
  - d. This will lead you to the local optimum but may not be the global optimal solution.
  - e. This is a basic “greedy algorithm” that always takes the first more optimal tree.
3. Much more sophisticated searches are employed by programs. Here are some of the common ones.
  - a. Random Sequence addition. As described in getting a near optimal tree above. Random start and addition is like to lead to trees of different length. These may or may not be globally shortest trees.
  - b. BB, Branch and Bound: An upper bound for length is found, usually on an optimal or near optimal starting tree and branch lengths are calculated only up to that length. Trees that exceed this length are not completely checked. This is a more effective means of searching but is still quite computationally intensive.
  - c. NNI, Nearest Neighbor Interchange: Swap on two adjacent branches.
  - d. SPR, Subtree Pruning and Regrafting: remove a branch and reinsert it at all possible points.
  - e. TBR, Tree Bisection and Reconnection: Break tree into two subtrees and reconnect the two in all possible ways.
  - f. Tree Fusion: Two optimal or near optimal trees are found and subgroups are exchanged. For example tree 1 has ((AB)(CD)) and tree 2 has ((AC)(BD)), these would be exchanged and length checked.
  - g. Genetic Algorithms: “simulated evolution”
    - i. “genotype” describes the tree
    - ii. “fitness function” is the optimality measure
    - iii. populations of trees are made that are similar among themselves and they can recombine by tree fusion or use SPR and TBR to “mutate”
    - iv. “fittest” trees can share their attribute and “reproduce” into other populations.
  - h. Tree Windows or Sectorial Searches: Extensive search of a subtree using BB or Enumeration for small numbers of OTUs (<15) or larger numbers

by using TBR (35-55). The larger the window or sector (number of OTUs) the less extensive the search need be to have a chance of escaping the local optima.

- i. Character Reweighting methods
  - i. “Parsimony Ratchet”, randomly select 5-25% of the characters in the matrix and increase their weight.
  - ii. Do TBR on the reweighted matrix
  - iii. Reset weights and calculate lengths on the set of trees found
  - iv. Keep best trees and repeat
  - v. Character sample can also be non-random
    1. bootstrap and identify weak areas of the tree and sample and reweight characters that are changing in those parts of the tree.
    2. reweight characters that best fit the tree
- j. Simulated annealing is a wandering algorithm method using Metropolis Algorithm that has a “temperature” parameter which dictates how severe the wandering permutations are and this decreases overtime and presumably can wander to the global optima
- k. Tree Drifting is similar to annealing but uses a Relative Fit Functions (Goloboff in TNT).