

### Parsimony, Patterns & Processes

**1. Hennig and Parsimony:** Hennig was not concerned with parsimony as an optimality criterion, but rather his general paradigm was consistent with parsimony as a guiding principle (e.g. Occam's Razor as a heuristic rule of thumb). The connection is in Hennig's Auxiliary Principle – *to assume homology if there is no evidence to suggest otherwise*. Hennig provided fundamental methods that made the relationship between character evidence and cladograms explicit, but he did not provide a clear method for choosing among competing alternatives.

Parsimony as used in phylogenetics is often defined as "*minimizing evolutionary changes*." In a sense this is correct, but it should not be construed to mean that one thinks evolution is parsimonious. If our character matrix consists of characters that have undergone rigorous character analysis to establish conjectural or primary homology, we then should seek hypotheses (trees) that maximizing our homologies. Conversely, we prefer trees 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 explanations* when we fail to get the primary homology right.

The two views that parsimony is "minimizing evolutionary changes" or "minimizing ad hoc explanations" is part of a larger tension between **pattern and process**.

Pattern cladists or transformed cladists are one extreme end of the spectrum. They put forward the idea that cladistic (in this case = strict parsimony) methods do not need, and in fact are better off without an evolutionary (process) justification. Three things are needed to justify building trees based 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. 2000. Evolution Is Not a Necessary Assumption of Cladistics. Cladistics 16, 143–154). Also part of the pattern v. process debates was the accusation of circularity, e.g. 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.*" Much debate exists in the literature in regard to parsimony. Is it assumption free, assumption minimizing or just a case where assumptions are ignored?

The clear understanding of patterns we observe and summarize, as opposed to processes that explain such patterns is important (e.g. *multiple substitutions and insertions/deletions are inferred events, all 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.

For "reconstructionists" evolutionary models are definitely part of the character analysis and should be used (with caution) in tree building. In the "estimation school" maximum parsimony (MP) is a just one model of evolution and many would say a rather poor one. Maximum likelihood (ML), and methods of that sort, benefit from an evolutionary model based justification. Exactly which model should be used, and where the values for parameters in the model should come from, are issues and this problem is addressed through assessment of the impact of model violations and statistical support.

Parsimony and likelihood are best viewed as belonging to a family of methods. They are character-matrix based, using information about individual hypotheses of homology, unlike the distance methods we talked about last time. The connection between parsimony and likelihood is shown clearly in the case of the “no common mechanism” model (Penny et. al. 1994, Tuffley and Steel 1997). This model loosens the assumptions of rate change so that there is potentially a different rate for every combination of branch and character across the tree, which comes back to the parsimony model. Essentially parsimony has both the property of being the simplest model (straightforward summing changes of observed states) and the most complex model (each character assumed to have its own rate), thus the relationship between MP and ML is more like a circle than a spectrum.

## 2. Parsimony as an optimality criterion.

Minimize the total number of steps over the tree where there is a set of parsimonious solutions.

How do we measure steps (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.

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

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.

**Explicit Evolutionary models:** (Typically implemented by the software) *Irreversible characters* ("Camin-Sokal parsimony")- Multiple gains allowed, no losses *Dollo parsimony*- Multiple losses allowed, multiple gains not. Implicit in all of these are a character state **step-matrix**, or **cost matrix** (Sankoff, 1975), assigning costs to changes.

Unordered					Ordered					Irreversible				
	0	1	2	3		0	1	2	3		0	1	2	3
0	0	1	1	1	0	0	1	2	3	0	0	1	2	3
1	1	0	1	1	1	1	0	1	2	1	∞	0	1	2
2	1	1	0	1	2	2	1	0	1	2	∞	∞	0	1
3	1	1	1	0	3	3	2	1	0	3	∞	∞	∞	0

-Step matrices can be used for any number of transformation or weighting schemes, even asymmetrical ones and ones with non-zero diagonals, e.g., transversions cost more. Values in the step matrix can be steps or probabilities or any relative measure:

**Weighting vs. Cost-** A character has a cost, i.e. the total number of steps or state transformations on a given topology adjusted by modifiers in the step matrix. A character also has a weight, i.e. a factor applied to any change in the character states. This acts the same as having many characters with the exact same state distribution in the matrix. Although the vast majority of people agree that all characters are not equally “good”, equal-weights (which is a kind of weighting) is most commonly used.

*A priori* weight.

Equal weights is viewed by some as a minimal necessary assumption. For others differential weighting, or the *a priori* selection of a weight 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.

*A posteriori* weighting:

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

#### 4. Tree space and tree Searching

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

Number of OTUs-	Number of rooted, resolved trees-
-----------------	-----------------------------------

2	1
3	3
4	15
5	105
6	945
7	10395
10	34459455
20	$\sim 10^{21}$

This is a proven NP complete problem.

Strategies typically used to find most parsimonious trees (MPTs):

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 forth to the edge that creates the shortest four OTUs network and so on until all OUTs 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 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).

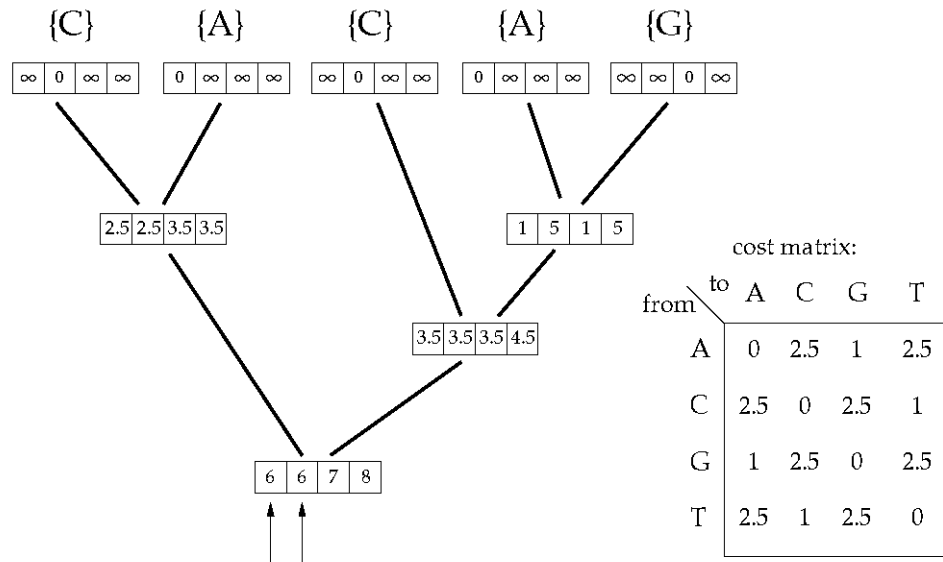


Figure 2.2: The Sankoff algorithm applied to the tree and site of the previous figure. The cost matrix used is shown, as well as the  $S$  arrays computed at each node of the tree.

modified from Felsenstein 2004 chp 2

A	0
B	1
C	1
D	0
E	0

