

# 3. Parsimony Methods

EEOB563  
Spring 2020

## 1 Building/choosing phylogenetic trees

Two possibilities:

- can use original characters
- can convert them into a measure of overall similarity/dissimilarity (aka distances)

Only distances can be used in clustering algorithms

Both distances and original characters can be used to choose the best tree(s) based on some optimality criterion.

Optimality criterion is an objective function that returns a score for any input tree topology.

Unlike the case for clustering algorithms, the basis for preferring one tree over another based on an optimality criterion is mathematically precise.

Four groups of methods that can use optimality criteria:

- Maximum Parsimony
- Distance-based
- Maximum Likelihood
- Bayesian

## 2 Maximum parsimony

The basic idea of parsimony is to seek a tree, or a collection of trees that minimizes the amount of evolutionary change (or transformations of one character state into another) required to explain the data.

We can also say that the parsimony criterion favors hypotheses that maximize congruence and minimize homoplasy.

The idea of parsimony can be traced back to the idea of Ockham's razor: "*Pluraritas non est ponenda sine necessitas*" (plurality shouldn't be posited without necessity) (William of Ockham, English monk, c. 1285-1349) and is a common principle we use in science.

Maximum parsimony (MP) depends on the idea of the fit of a character to a tree, which we can define as the minimum number of steps required to explain the observed distribution of character states among taxa. The sum of steps over all characters is called Tree Length. Most parsimonious trees have the minimum tree length needed to explain the observed distributions of all the characters.

**Two questions we need to answer:**

- How to count (optimize) steps on a given tree? (small parsimony problem)
- How to evaluate different trees? (large parsimony problem: usually too many trees – which ones to evaluate?)

### 3 Counting steps

How we count steps (the number of edges on which change occurred) depends on the type of parsimony analysis:

- Wagner – ordered, additive (0-1-2)
- Fitch-Hartigan – unordered, non-additive (equal costs for all changes)
- Sankoff – generalized parsimony (variable costs for different kinds of changes)

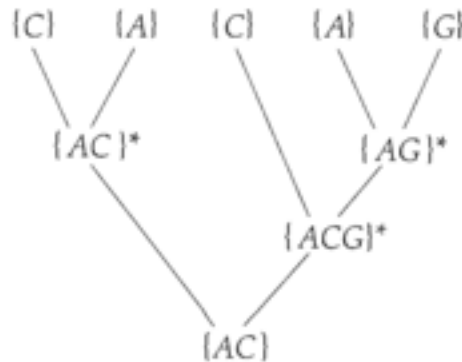


Figure 1: Example of F-H algorithm: 3 changes required, \* = 1 change

#### 3.1 Fitch-Hartigan algorithm:

- Build a set  $N_i$  at each leaf  $i$  containing the observed character state at that node.
- At each internal node with two non-empty descendent sets  $N_i$  and  $N_j$ , build a set  $N_k$  such that  $N_k = N_i \cup N_j$  if  $N_i \cap N_j = \emptyset$  and  $N_k = N_i \cap N_j$  otherwise.. Furthermore, if  $N_i \cap N_j = \emptyset$ , count +1 changes.

Please note (and remember for the exam!) that the parsimony score does not depend on the root location (see proof in the text).

### 3.2 Sankoff algorithm:

Let  $c_{ij}$  denote the cost of transitioning from state  $i$  to state  $j$ . Let  $S_i(a)$  denote the length of the subtree descending from node  $a$  assuming the state at node  $a$  is state  $i$ . Let  $l$  and  $r$  denote the two nodes immediately descendent from  $a$ . Then,

$$S_i(a) = \min_j [c_{ij} + S_j(l)] + \min_k [c_{ik} + S_k(r)]$$

and the total tree length is  $\min_i [S_i(R)]$ , where  $R$  is the root node.

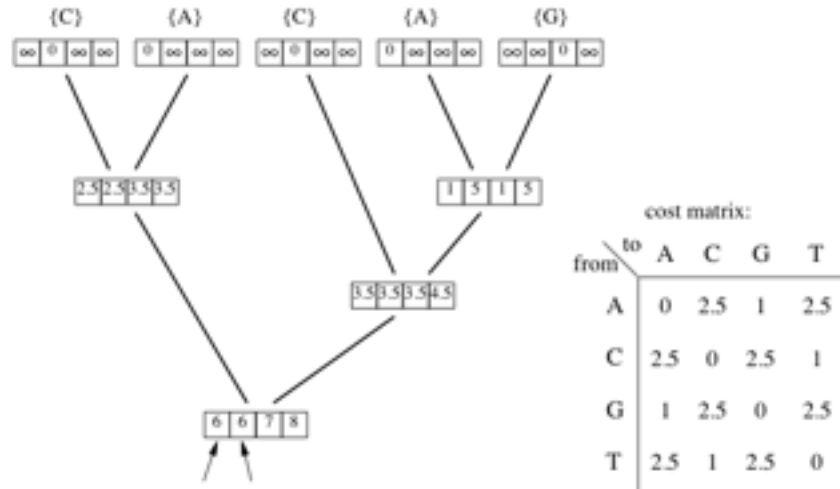


Figure 2: An example of using Sankoff's algorithm

## 4 Tree search

- Exhaustive
- Branch & Bound
- Heuristic

**Exhaustive search** is limited by the number of possible trees (10 species – 2,027,025 trees; 20 species – 221,643,095,476,699,771,875 trees; 53 species > number of atoms in the universe; 67 species > volume of the universe in cubic angstroms (  $10^{70,000,000}$ )). Usually < 12 species.

Have to devise a logical strategy to go through the tree space (*e.g.*, step-wise addition or start decomposition).

**Branch and bound search** allows one to find the optimal tree without looking at all possible trees.

**Heuristic search.** “The task of searching for an optimal tree by approximate methods is somewhat analogous to the plight of a myopic pilot who loses his glasses when forced to parachute from his airplane into a mountainous region. He suspects that there is a manned outpost at the top of highest peak in the area, and he must somehow grope his way there to have any hope of rescue” Swofford et al. 1996 in Molecular Systematics.

Two steps in a heuristic search:

1. Obtain a starting tree
  - (a) Stepwise addition
  - (b) Start decomposition
  - (c) Random
2. Rearrange this tree
  - (a) Don't bother
  - (b) Nearest-neighbor interchange (NNI)
  - (c) Subtree pruning and regrafting (SPR)
  - (d) Tree bisection and reconnection (TBR)

## 5 Measures of Character Fit

- Treelength
- Consistency index (*ci*)
- Retention index (*ri*)

When a character fits a cladogram perfectly, it can be considered 100% consistent with that particular cladogram. When a character does not fit a cladogram perfectly, it is less than 100% consistent. Such a character exhibits homoplasy. This discordance can be measured by various indices:

### 5.1 Consistency index

The [per-character] consistency index  $ci = m/s$ , where  $m$  = minimum number of steps on any cladogram ( $m = \#$  character states - 1) and  $s$  = actual number of steps on a particular cladogram. The ensemble consistency index  $CI$  is a similar index summed over all characters.

## 5.2 Retention index

The [per-character] retention index  $ri = (g - s)/(g - m)$ , where  $m$  and  $s$  are as above and  $g$  is the maximal number of steps for the character on any cladogram. The ensemble retention index RI is a similar index summed over all characters:  $RI = (G - S)/(G - M)$ . The retention index measures the amount of synapomorphy on the tree, and varies from 0 to 1.

## 6 Assessing support for individual clades

- Branch length (a good measure only for perfect “Hennigian” data)
- Bremer Support (=Decay Index)
- Statistical Procedures
  - Character re-sampling:
    - \* Bootstrap
    - \* Jackknife
  - Randomization tests
    - \* T-PTP

### 6.1 Bremer Support (=Decay Index)

The Bremer index (K. Bremer, 1988) for a particular node is the number of additional steps in the shortest tree(s) without the node in question.

To calculate:

- Find shortest tree(s): length = S
- Find shortest tree(s) without node i: length = Si
- Decay index = Si - S

### 6.2 Bootstrapping

Characters are re-sampled with replacement to create a large (100-1000) number of data sets of the same size as the original data set (each character can be present 0, 1, or more times). Each replicate data set is analyzed (e.g., the MP tree is found). Agreement among the resulting trees is summarized as a majority-rule consensus tree. Frequency of occurrence of a group in the replicate analysis (bootstrap percentage) is interpreted as a measure of support for that particular group in the data set.

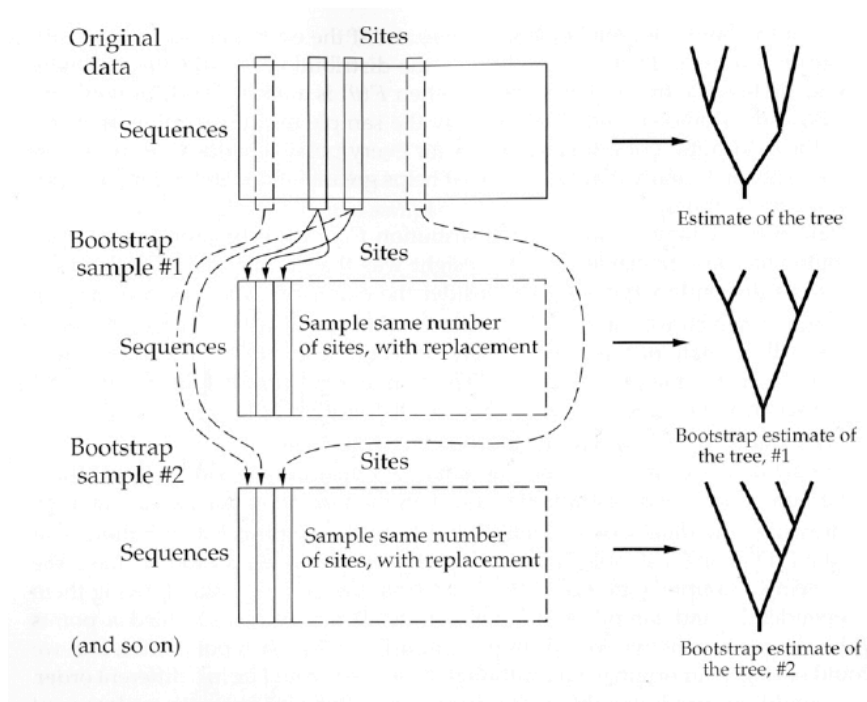


Figure 3: Bootstrapping algorithm

### 6.3 Jackknifing

Characters are re-sampled without replacement to create a large (100-1000) number of data sets usually of 50% size of the original (you can also say that 50% of characters are randomly deleted). Each replicate data set is analyzed (*e.g.*, the MP tree is found). Agreement among the resulting trees is summarized as a majority-rule consensus tree. Frequency of occurrence of a group in the replicate analysis (jackknife percentage) is interpreted as a measure of support for that particular group in the data set. 50% jackknife is statistically equivalent to bootstrap analysis.