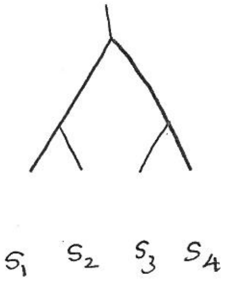
PARSIMONY

Parsimony

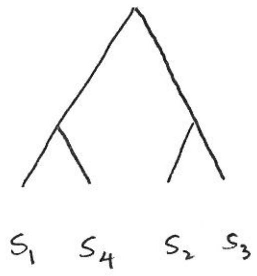
Consider the sequences

S_1:	A	A	T	G	L
S_2:	A	G	T	G	S
S_3:	A	G	T	C	S
S_4:	A	A	T	C	L
	DNA				morphological

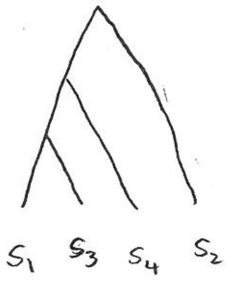
Which of the 4 (out of $(2n-3)!! = 5!! = 15$) rooted trees, do we prefer? and why?



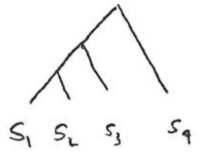
ps = 5



ps = 4

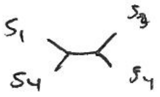


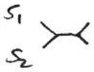
ps = 6



These numbers are called the PARSIMONY SCORES of T, ps(T)

Subtle point: Parsimony scores are really counts for unrooted trees.

For example, trees 1 and 4 are rooted versions of  and $ps(T_1) = ps(T_4)$

The three other rooted versions of  would also have the same ps.

Rooting just provides a way (= algorithm) for computing ps.

These numbers are known as the PARSIMONY SCORES of T

Theorem: $ps(T^{\rho}) = ps(T)$

i.e. the parsimony score of a rooted tree

= " " " " its unrooted version

$\Rightarrow$ PARSIMONY SCORES compare unrooted trees only

The underlying principle: The best tree (or trees in the case of ties) is the one requiring the minimal amount of changes. (substitutions/ state changes)

Defn: A column in aligned sequences is called a CHARACTER (usually denoted by χ) and a character can take on any of

s STATES

Eg: DNA character $s = 4$ morphological char $s=2$

Data: $\chi_1\ \chi_2\ \chi_3 \ldots$

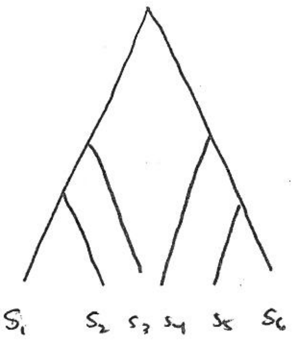
How should we compute parsimony scores?

(or have a computer)

Do alg development + Eg Simultaneously

Fix a binary tree T

Alg.



A	A
A	A
T	A
T	G
G	G
C	C

Sum over all characters χ_i, $i=1,\dots,m$

find $ps_{\chi_i}(T)$ by

0. If unrooted, arbitrarily root tree
1. place χ_i at leaves of tree
2. starting at the leaves, label each ancestral node by
   - the union of the two sets below if they are disjoint in which case the parsimony count is augmented by 1
   - the intersection of the two state sets below if they share elts (and do not augment the parsimony count)
3. Continue up tree until reach root   Return $ps_{\chi_i}(T)$

Comments:

1. This is called the FITCH-HARTIGAN ALGORITHM and it does compute the $ps(T)$ Thm: FH count on $T = ps(T)$

2. computer scientists would call this the SMALL-PARSIMONY PROBLEM (fix one single tree and compute its ps)

How complex is this?   If T is an X-tree with $n=|X|$ leaves

Roughly, n internal nodes to visit

each node might have any of s states and

must loop over m characters

$\Rightarrow \approx nsm \equiv \Theta(nsm)$ "big-Θ"

$\Rightarrow$ time is $\leq$ (constant)(nsm)

3. The LARGE-PARSIMONY PROBLEM is the one of interest

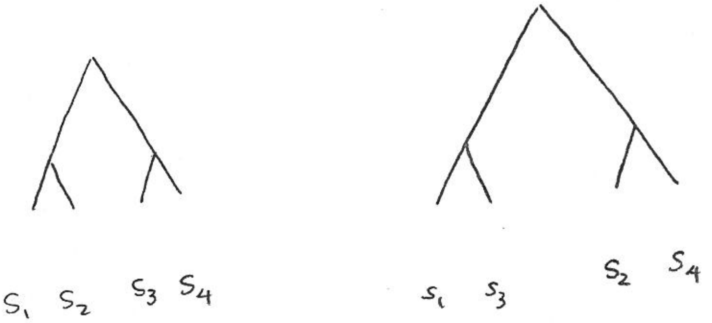
Search over all $(2n-5)!!$ unrooted trees

compute the parsimony score for each

return those with best score

UAF grad Ron Graham + Foulds proved this is NP-hard

4. Ways to speed this up



S_1:	A	A	A	A	A	A
S_2:	A	G	C	C	A	A
S_3:	A	A	T	C	C	T
S_4:	A	A	G	T	C	T

Remove the constant sites or, more generally, the non-informative sites

Defn: If χ is a character, with S states, then a PARSIMONY-INFORMATIVE CHARACTER in which 2 states appear at least twice

A	A	A
A	A	C
T	A	G
T	T	T

S	S
L	S
L	L
L	L

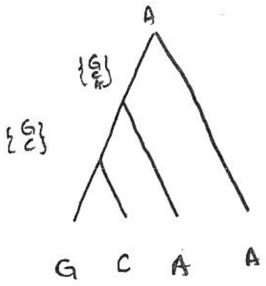
To speed up parsimony, preprocess to remove non-informative sites

recognize that all informative patterns, for instance, X X Y Y contribute equally to parsimony count

i.e. particular states don't matter, only pattern of states.

5. Reconstructing ancestral states

(Beware FH.)

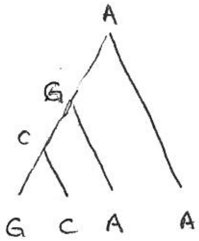


②

A EXTENSION $\tilde{\chi}$ of χ is a assignment of states to the internal nodes of T consistent with χ at the leaves.

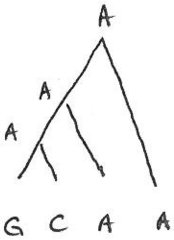
$\tilde{\chi}$ is minimal, if it computes $ps_{\chi}(T)$

Eg



$\tilde{\chi}$ is minimal ext. of χ as is

However,



This extension also gives rise to ps = 2, but does not arise from FH

extensions
Bad News: Not all labellings that give rise to ps for χ on T arise from FH algorithm

Good news: ② You can get some by a second pass down the tree.

Modify FH to get all ...

or use Sankoff algorithm

① The state set assigned to ρ by FH is exactly the state-set for any minimal ext $\tilde{\chi}$ of character χ

WEIGHTED PARSIMONY or the Sankoff algorithm

Underlying idea:

not all characters are equal $\rightsquigarrow$ Character weighting

not all state changes are equal. $\rightsquigarrow$ Sankoff alg.

Character-weighting: CODONS: 2 3 1 2 3 1

$\nearrow$ x_1 x_2 x_3

twice as important

$$ps(T) = 2\,ps_{x_1}(T) + ps_{x_3}(T) + ps_{x_4}(T)$$

Any scheme for character-weighting is legitimate if coefficients = weights > 0

In practice, software usually wants integer weights

Weighting state Changes.

A transition is a state change $A \leftrightarrow G$, $C \leftrightarrow T$

transversion between purines and pyrimidines

$$\{A,G\} \longleftrightarrow \{C,T\}$$

Empirically, transitions are observed much <u>more</u> <u>frequently</u> than transversions

FH all state changes are weighted equally

Perhaps better: introduce a WEIGHT MATRIX, a COST MATRIX, a STEPMATRIX

Eg.

$$W = \begin{array}{cc|cccc} & & & \text{to} & & \\ & & A & G & C & T \\ \hline & A & 0 & 1 & 2 & 2 \\ \text{from} & G & 1 & 0 & 2 & 2 \\ & C & 2 & 2 & 0 & 1 \\ & T & 2 & 2 & 1 & 0 \end{array}$$

Comments: this weight matrix is symmetric, but this is <u>not</u> necessary

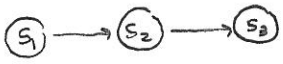
We will use w_{ij} = cost of changing from state i to state j.

Indeed, the weight matrix is user-defined judgment on the cost of changes

W = transversions cost 2x transitions

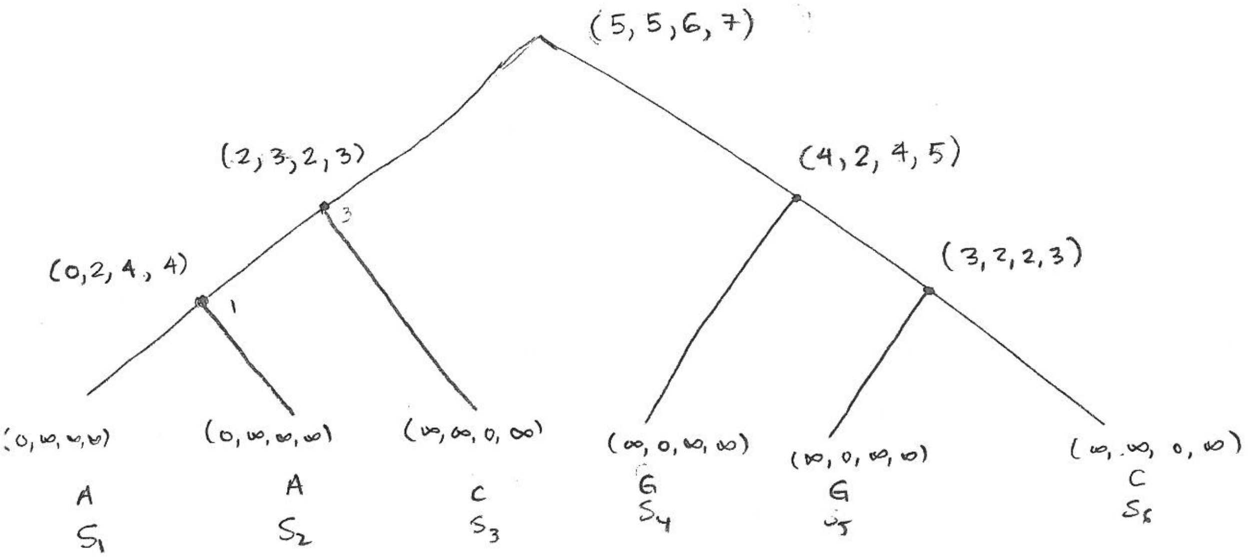
Alternative: χ 3-state character

$$\begin{array}{c|ccc} & s_1 & s_2 & s_3 \\ \hline s_1 & 0 & 1 & 2 \\ s_2 & \infty & 0 & 1 \\ s_3 & \infty & \infty & 0 \end{array}$$



See for eg PAUP* manual on web pp 16-20

The Sankoff algorithm for weighted-parsimony by example: with transversions twice the cost of transition. W= from to:



Main idea: starting at the leaves, and moving up toward root ρ

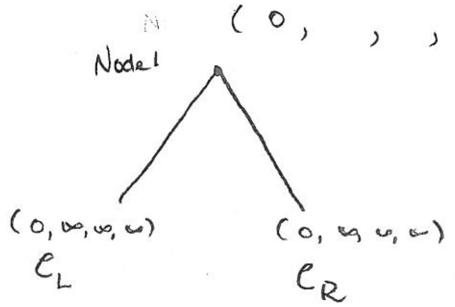
Place a $S = 4$ element vector with each entry containing the minimal cost of labelling that node with state s_i

For this particular example, we order the vectors

purines first, pyrimidines 2nd

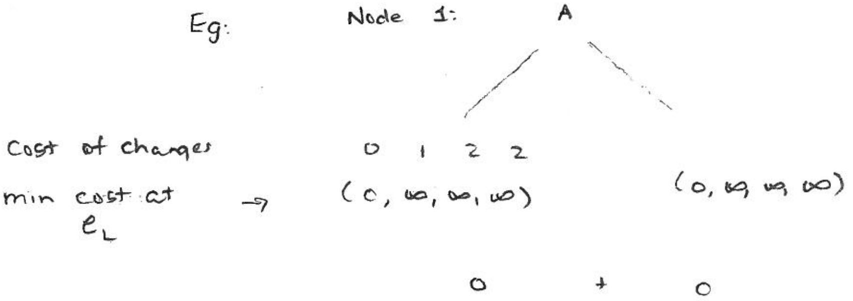
At each tip, form its count vector by placing a 0 in the ith slot if the tip is in state i; an ∞ elsewhere.

Move "up" through the tree, focusing on an internal node and its immediate children e_L + e_R



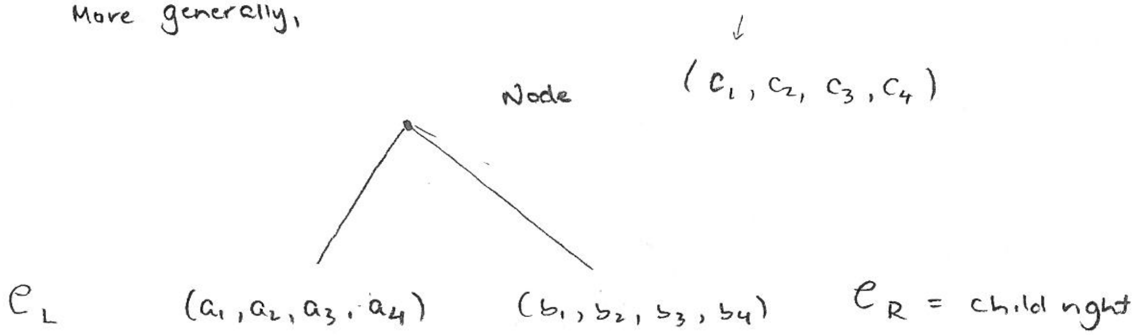
Use the two children and weights w_{ij} to compute the minimal count vector.

Eg:



minimize $\{(\text{cost of change } a \to j) + (\text{cost of } j \text{ at } e_L)\}$ +

More generally,



$$\min\left\{\begin{array}{c}\text{Cost of } A \to j + \\ \text{Cost of } j \text{ at } e_L\end{array}\right\} + \min\left\{\begin{array}{c}\text{cost of } A \to k + \\ \text{Cost of } e_R(k)\end{array}\right\}$$

$$= \min\{w_{1j} + a_j\} + \min\{w_{1k} + b_k\}$$

Sankoff algorithm to compute the parsimony score for an s-state character χ on T with weight matrix W

i.e. compute $ps_{\chi}(T)$ using weights $W = (w_{ij})$

0. Arbitrarily root T to get T^{ρ}

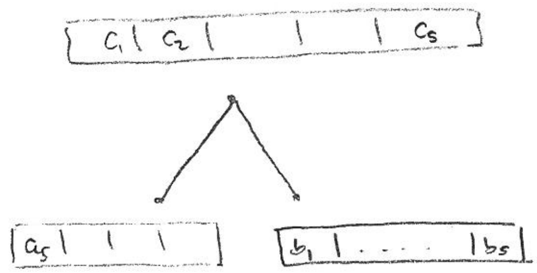
1. At each tip of the tree, place an s-element vector

If tip is in state i, place a 0 in the i-th position

Place an ∞ elsewhere.

2. Move up the tree through all internal nodes.

At each internal node, place an s-element vector with entries computed as follows:



$$\text{Let } c_i = \min_{\substack{\text{states}\\ 1,2,\dots,s}}\{w_{ij} + a_j\} + \min_{j\in S}\{w_{ij} + b_j\}$$

cost of state change $i \to j$

+ cost left child node in state j

3. When you reach the root ρ,

Consider its Count vector $[\,c_1 \mid c_2 \mid \;\mid\; \mid c_s\,]$

Then $ps_{\chi}(T) =$ minimum entry.

Theorem: The Sankoff algorithm does compute $ps_\chi(T)$ with weights W

Comments: This is an example of a "dynamic programming algorithm"

One advantage: all minimal exts are computable by examining the partial count vectors.

- If the weight matrix W is symmetric, then ps evaluates unrooted trees
  is <u>not</u> symmetric, " " is a measure on <u>rooted</u> trees

Reflections:

1. There are many other types of parsimony possible — See Chap 7 Felsenstein

   (Good project possibly?)

2. Strengths vs. Weaknesses

Change is rare is reasonable.

For morphological data, this seems a good choice.

heuristics are needed since we can't explore tree space $(2n-5)!!$

What if taxon have evolved over a "long" period of time? Is it reasonable to avoid consideration of multiple changes!

$L \to M \to S$

parallel evolution?

What is the meaning of the parsimony scores?

What is the meaning of W_{ij}?