Phylogenetic trees

Weighted Parsimony

Outline

- Weighted Parsimony task
- Dynamic programming solution

Weighted Parsimony

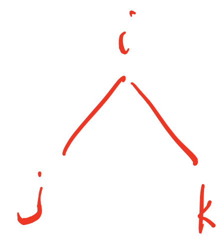
- [Sankoff & Cedergren, 1983]
- instead of assuming all state changes are equally likely, use different costs $S(a,b)$ for different changes $a \rightarrow b$

Weighted Parsimony

- Dynamic programming!
- Subproblem: want to determine minimum cost $R_i(a)$ for the subtree rooted at i of assigning character a to node i
- for leaves:

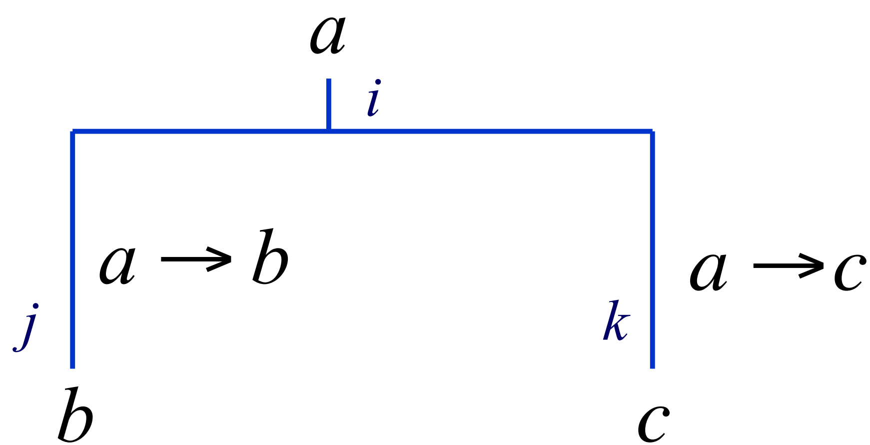
$$R_i(a) = \begin{cases} 0, & \text{if } a \text{ is character at leaf} \\ \infty, & \text{otherwise} \end{cases}$$

Weighted Parsimony



- for an internal node *i* with children *j* and *k*:

$$R_i(a) = \min_b (R_j(b) + S(a,b)) + \min_c (R_k(c) + S(a,c))$$



Example: Weighted Parsimony

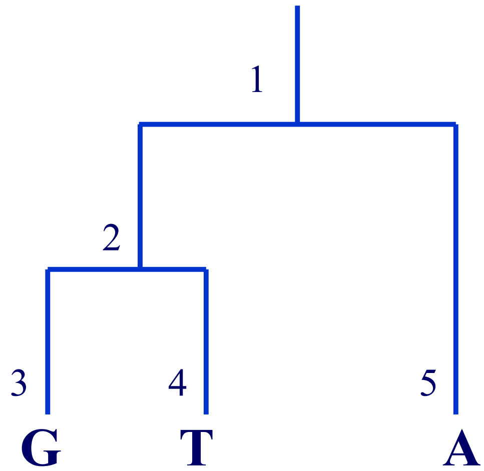
$$R_3[A] = \infty, R_3[C] = \infty, R_3[G] = 0, R_3[T] = \infty$$

$$R_4[A] = \infty, R_4[C] = \infty, R_4[G] = \infty, R_4[T] = 0$$

$$R_2[A] = R_3[G] + S(A,G) + R_4[T] + S(A,T)$$
$$\vdots$$
$$R_2[T] = R_3[G] + S(T,G) + R_4[T] + S(T,T)$$

$$R_5[A] = 0, R_5[C] = \infty, R_5[G] = \infty, R_5[T] = \infty$$

$$R_1[A] = \min\left(R_2[A] + S(A,A), \quad \ldots \quad , \quad R_2[T] + S(A,T)\right) \; + \; R_5[A] + S(A,A)$$
$$\vdots$$
$$R_1[T] = \min\left(R_2[A] + S(T,A), \quad \ldots \quad , \quad R_2[T] + S(T,T)\right) \; + \; R_5[A] + S(T,A)$$

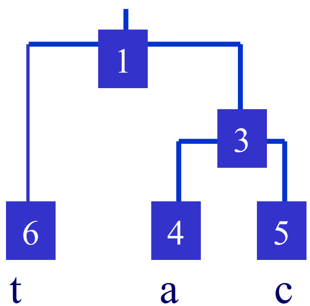


Weighted Parsimony: Traceback

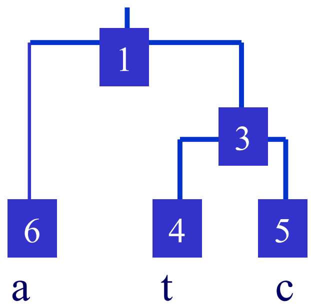
- do a pre-order (from root to leaves) traversal of tree
- for root node:
  - select minimal cost character
- for each other internal node:
  - select the character that resulted in the minimum cost explanation of the character selected at the parent (could use traceback pointers)

Weighted Parsimony Example

Consider the two simple phylogenetic trees shown below, and the symmetric cost matrix for assessing nucleotide changes. The tree on the right has a cost of 0.8



0-8

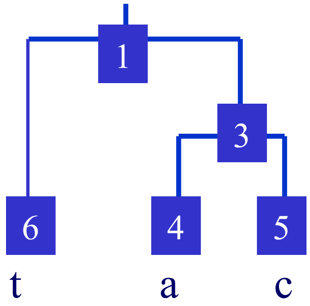


	a	c	g	t
a	0	0.8	0.2	0.9
c	0.8	0	0.7	0.5
g	0.2	0.7	0	0.1
t	0.9	0.5	0.1	0

What are the minimal cost characters for the internal nodes in the tree on the left?

Which of the two trees would the maximum parsimony approach prefer?

Weighted Parsimony Example



	a	c	g	t
a	0	0.8	0.2	0.9
c	0.8	0	0.7	0.5
g	0.2	0.7	0	0.1
t	0.9	0.5	0.1	0

$$R_3(a) = 0 + 0.8 = 0.8$$

$$R_3(c) = 0.8 + 0 = 0.8$$

$$R_3(g) = 0.2 + 0.7 = 0.9 \quad S(g,a) + S(g,c)$$

$$R_3(t) = 0.9 + 0.5 = 1.4$$

$S(a,t)$ $R_3(a)$ $S(a,c)$ $R_3(c)$ $S(a,g)$ $R_3(g)$ $S(a,t)$ $R_3(t)$

$$R_1(a) = 0.9 + \min\{0.8,\ \ 0.8 + 0.8,\ \ 0.2 + 0.9,\ \ 0.9 + 1.4\} = 1.7$$

$$R_1(c) = 0.5 + \min\{0.8 + 0.8,\ \ 0.8,\ \ 0.7 + 0.9,\ \ 0.5 + 1.4\} = 1.3$$

$$R_1(g) = 0.1 + \min\{0.2 + 0.8,\ \ 0.7 + 0.8,\ \ 0.9,\ \ 0.1 + 1.4\} = 1.0$$

$$R_1(t) = 0 + \min\{0.9 + 0.8,\ \ 0.5 + 0.8,\ \ 0.1 + 0.9,\ \ 1.4\} = 1.0$$

The minimal cost character for node 1 is either **g** or **t**. The minimal cost character for node 3 is **g**. The maximum parsimony approach would prefer the other tree, because it has a smaller cost (0.8).

Summary

- Extension of parsimony to weighted costs
- Dynamic programming solution
  - Postorder fill stage
  - Preorder traceback stage