

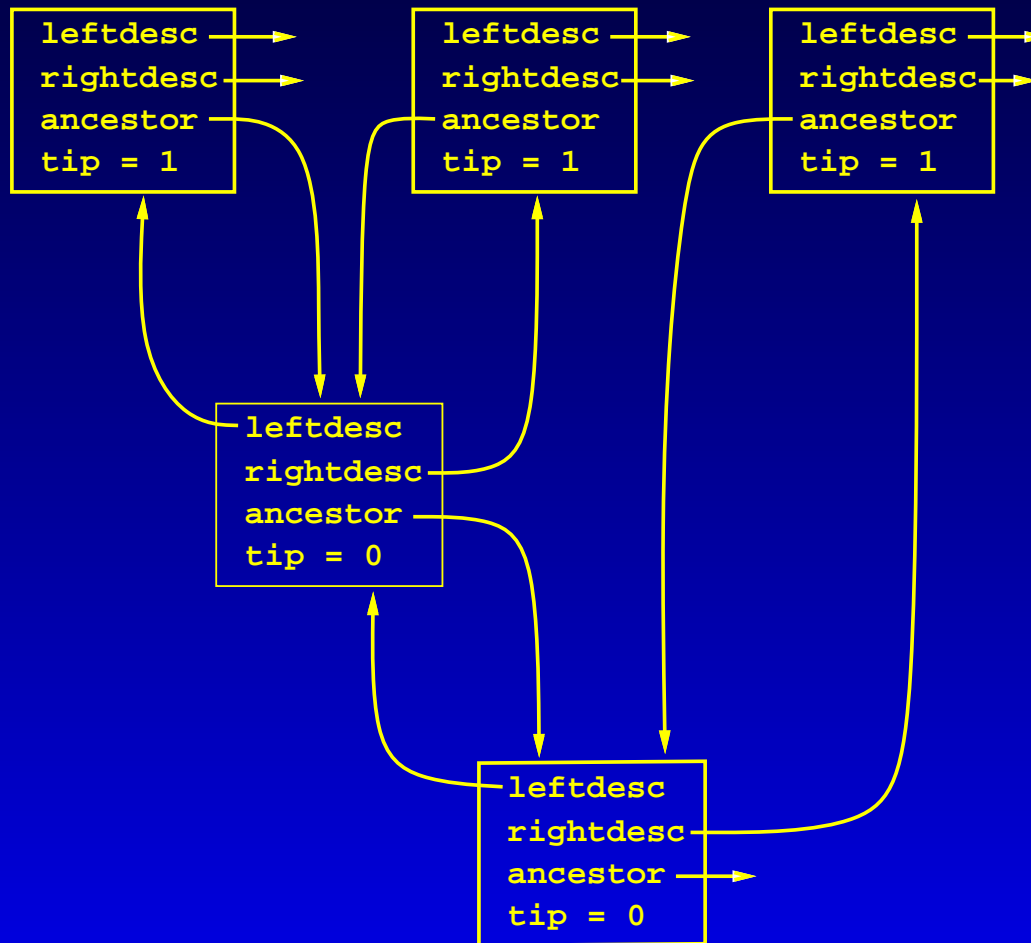
# Lecture 24. Phylogeny methods I (Parsimony and such)

Joe Felsenstein

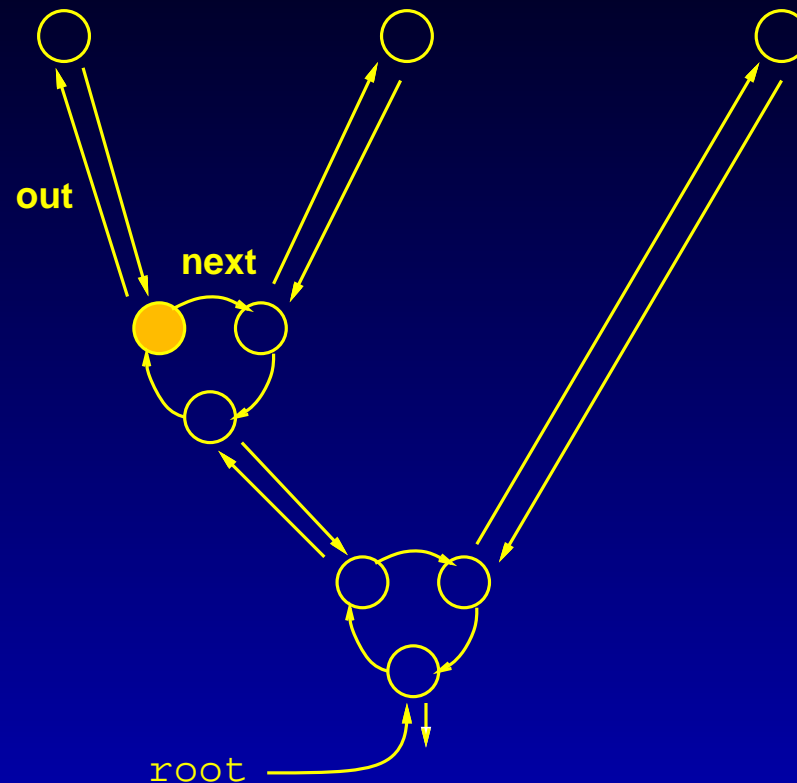
Department of Genome Sciences and Department of Biology

# Representing a tree in the computer

Using records (in C: structures, in Java and C++: classes) and pointers:  
Here is one record-pointer structure representing a small tree:



## A better representation, allowing multifurcation



This one allows multifurcations and is more easily rerootable. Each small circle represents a record with two pointers, "next" and "out", and a boolean variable "tip".

## A computer-readable notation for phylogenies

The Newick standard for computer readable trees represents the previous tree, with branch lengths on each branch, by nested parentheses:

$$((A:0.1,B:0.2):0.06,C:0.4);$$

Each interior node is a pair of parentheses, enclosing the subtrees coming from that node. Each branch length is placed after the node that is at the top of that branch.

See:

<http://evolution.gs.washington.edu/phylip/newicktree.html>

# Methods of reconstructing phylogenies (evolutionary trees)

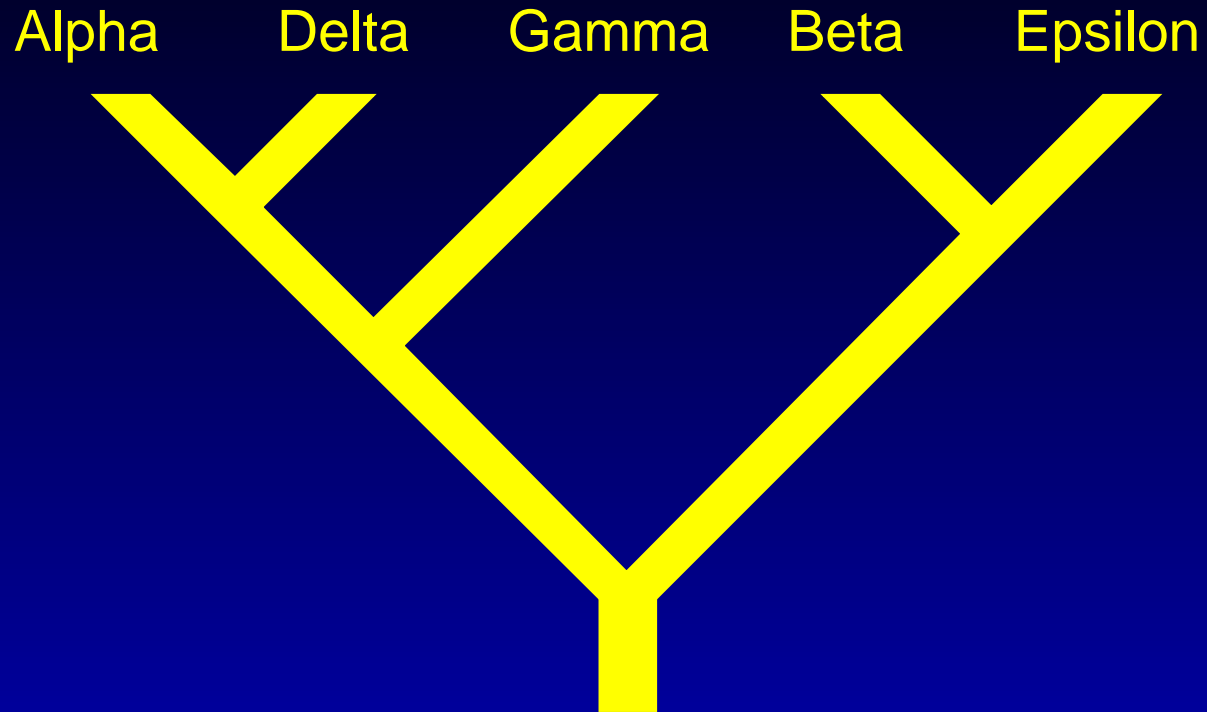
**Parsimony methods.** Tree that allows evolution of the sequences with the fewest changes. *Also compatibility methods: tree that perfectly fits the most states.*

**Distance matrix methods.** Tree that best predicts the entries in a table of pairwise distances among species. *Closely related to clustering methods.*

**Maximum likelihood.** Tree that has highest probability that the observed data would evolve. *Also Bayesian methods: tree which is most probable a posteriori given some prior distribution on trees.*

**Invariants.** Tree that predicts certain algebraic relationships among patterns in the data. *Mathematically fun though little-used as it ignores too much of the data.*

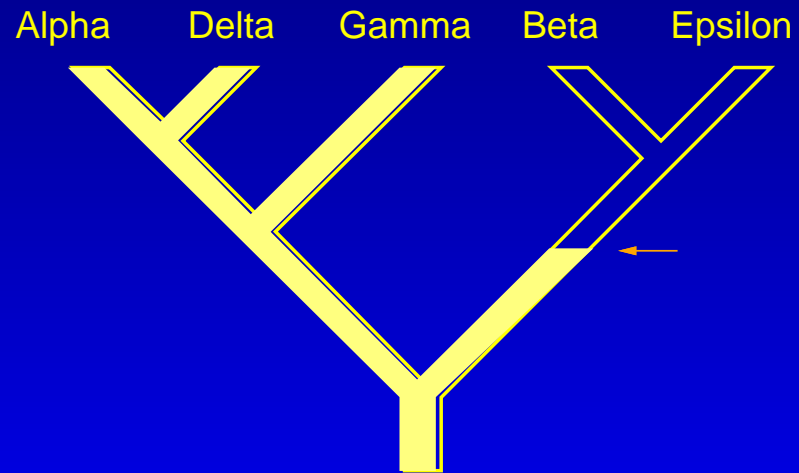
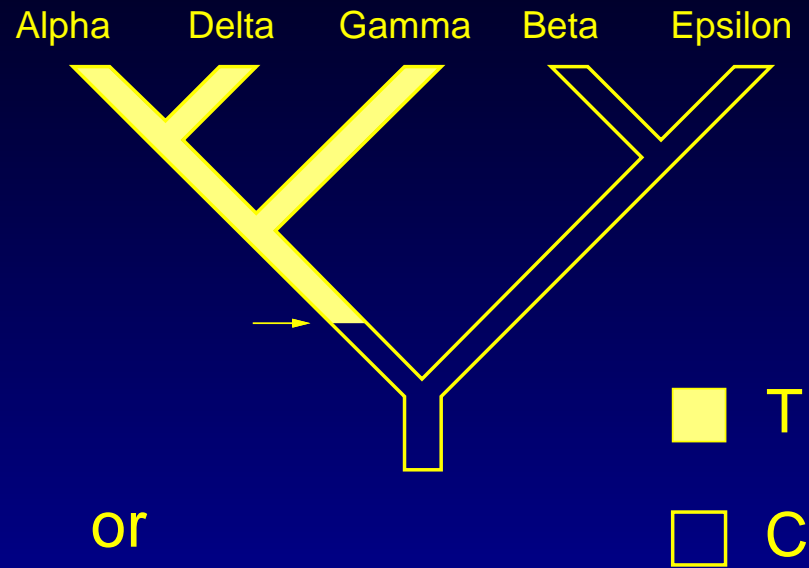
# A tree we will be evaluating



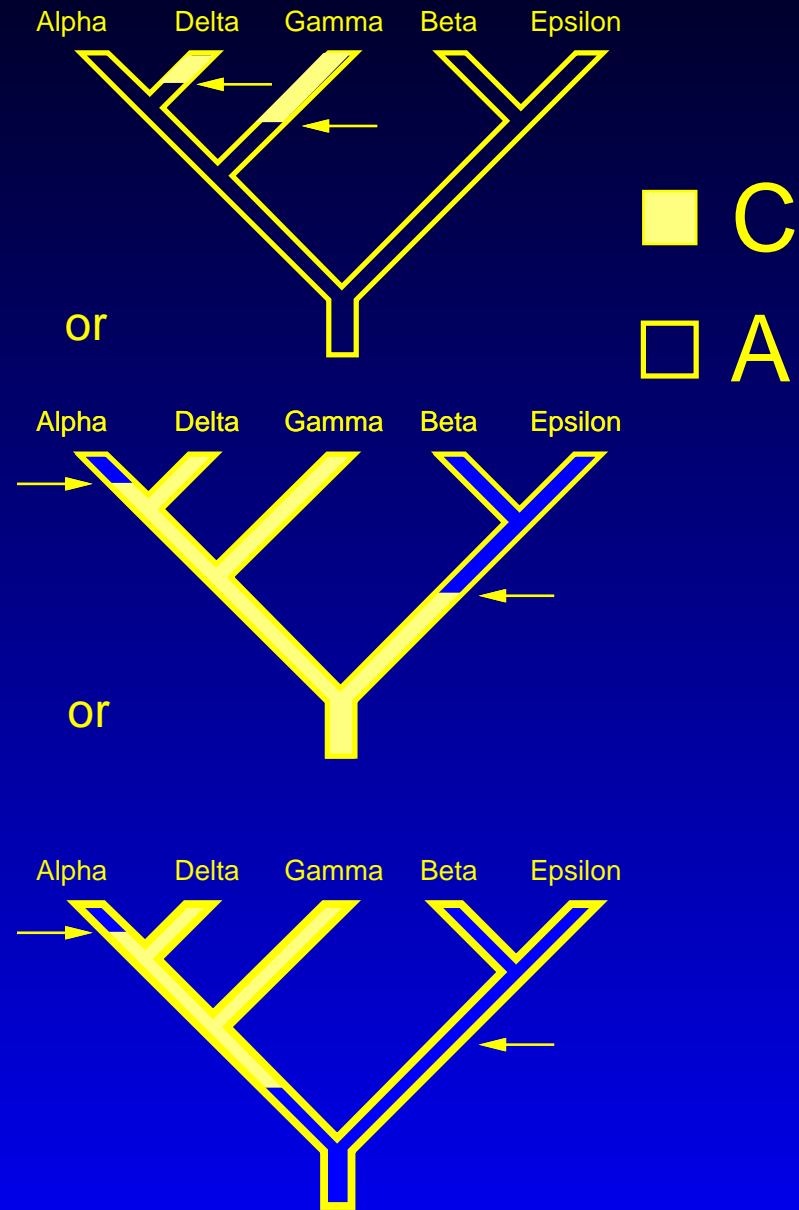
## A simple data set with nucleotide sequences

Species	Characters					
	1	2	3	4	5	6
Alpha	T	A	G	C	A	T
Beta	C	A	A	G	C	T
Gamma	T	C	G	G	C	T
Delta	T	C	G	C	A	A
Epsilon	C	A	A	C	A	T

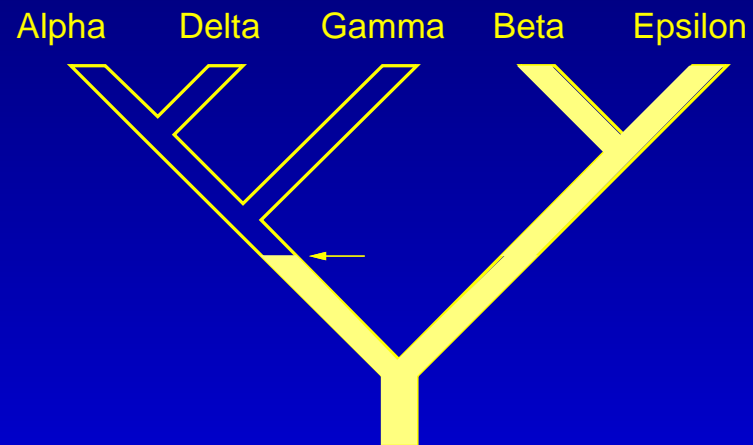
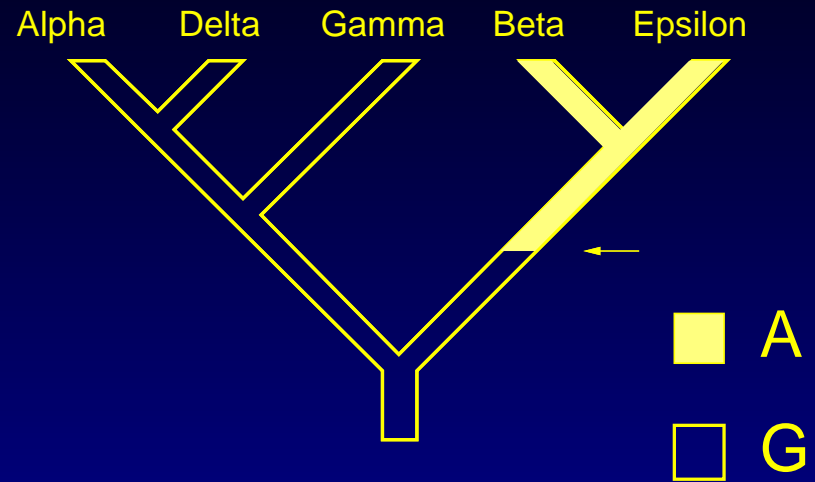
# Most parsimonious states for site 1



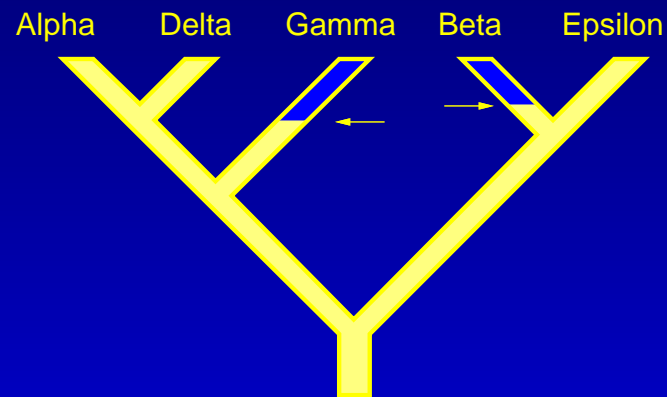
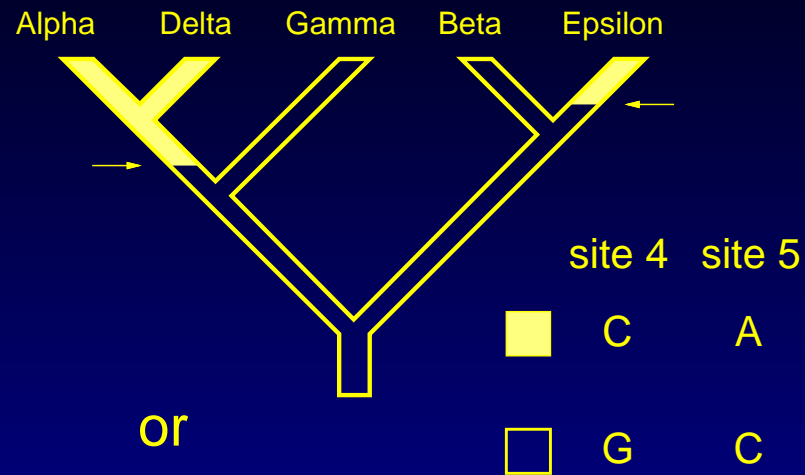
# Most parsimonious states for site 2



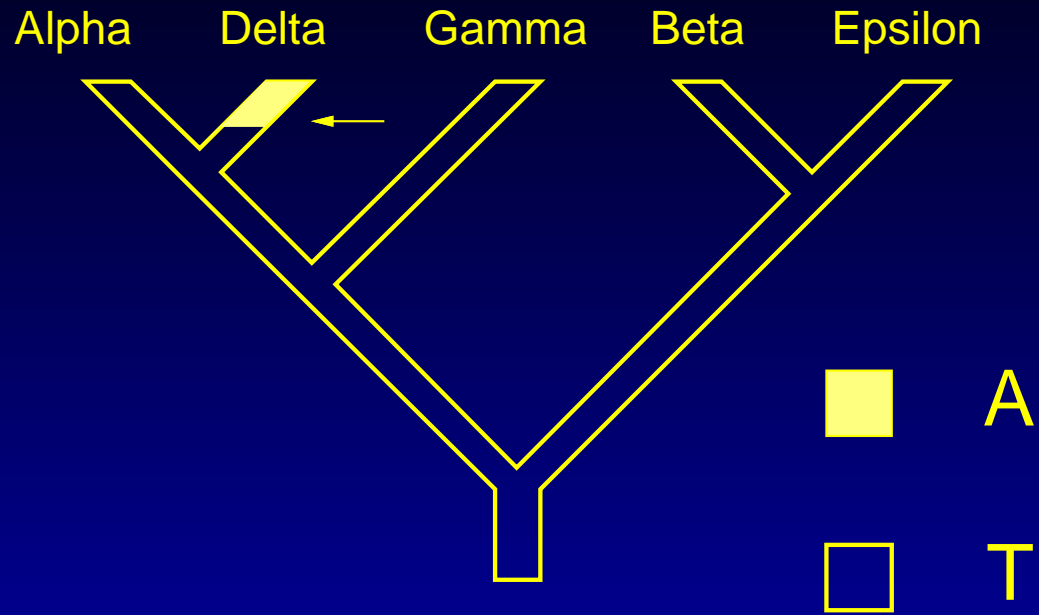
# Most parsimonious states for site 3



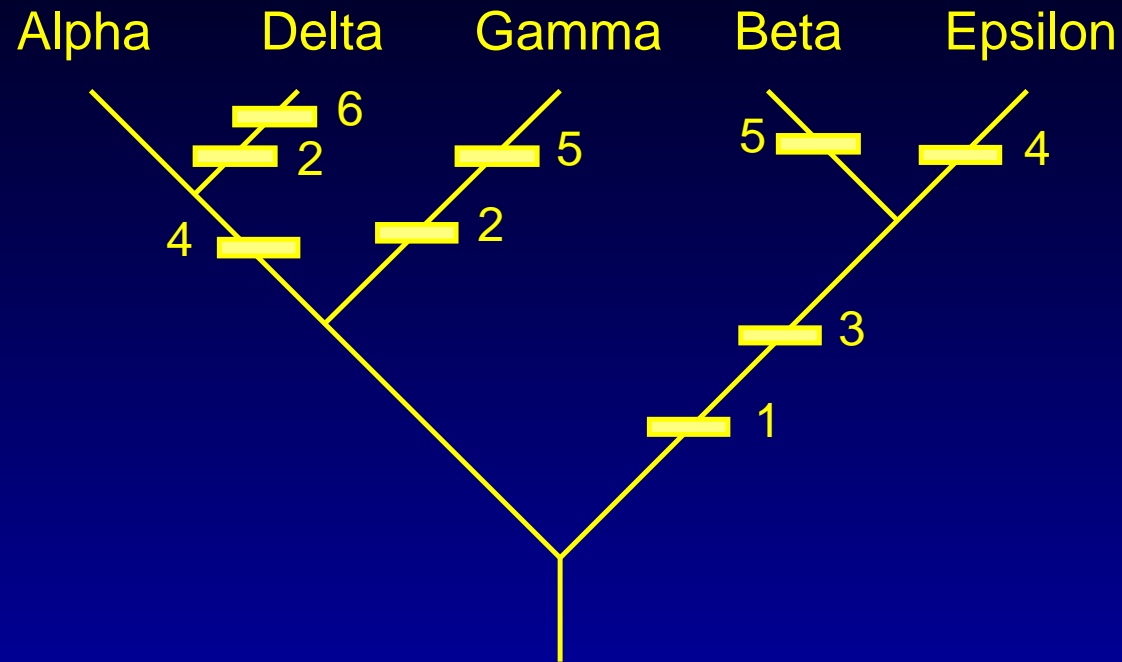
# Most parsimonious states for sites 4 and 5



# Most parsimonious states for site 6

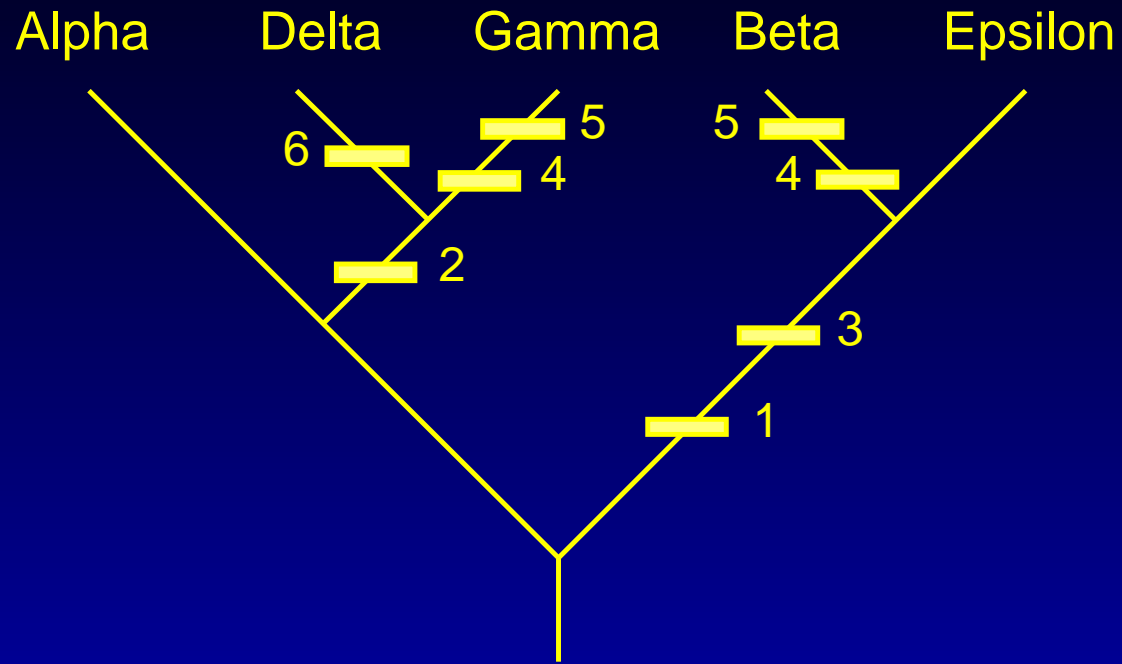


## Steps on this tree

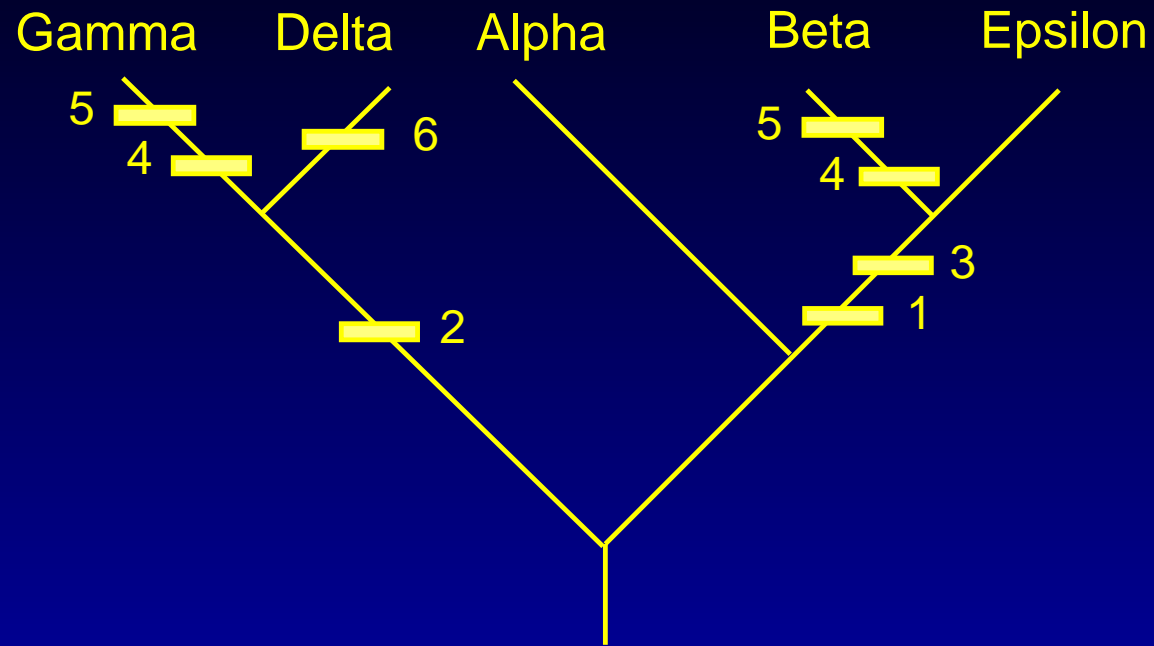


Steps on this tree, all characters, for one choice of reconstruction at each site. There are 9 steps in all

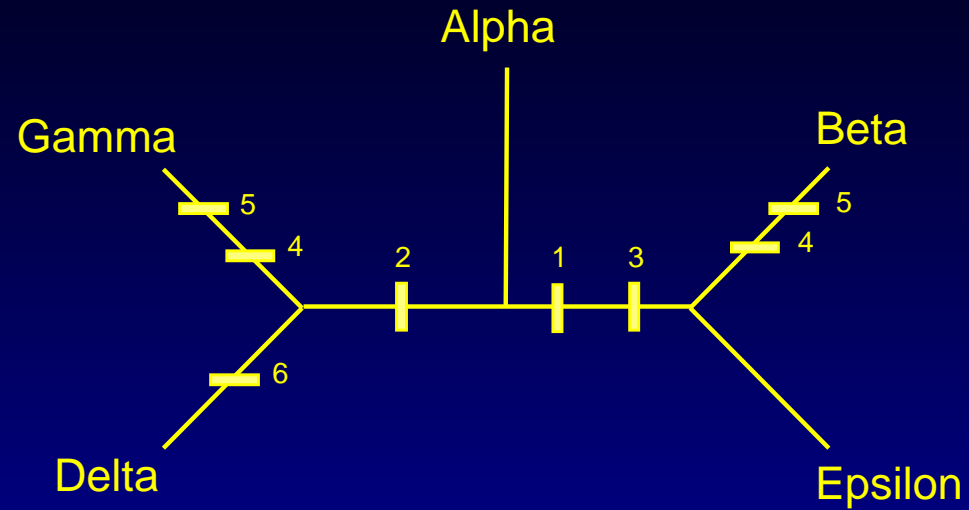
# Steps on another tree (8 in all)



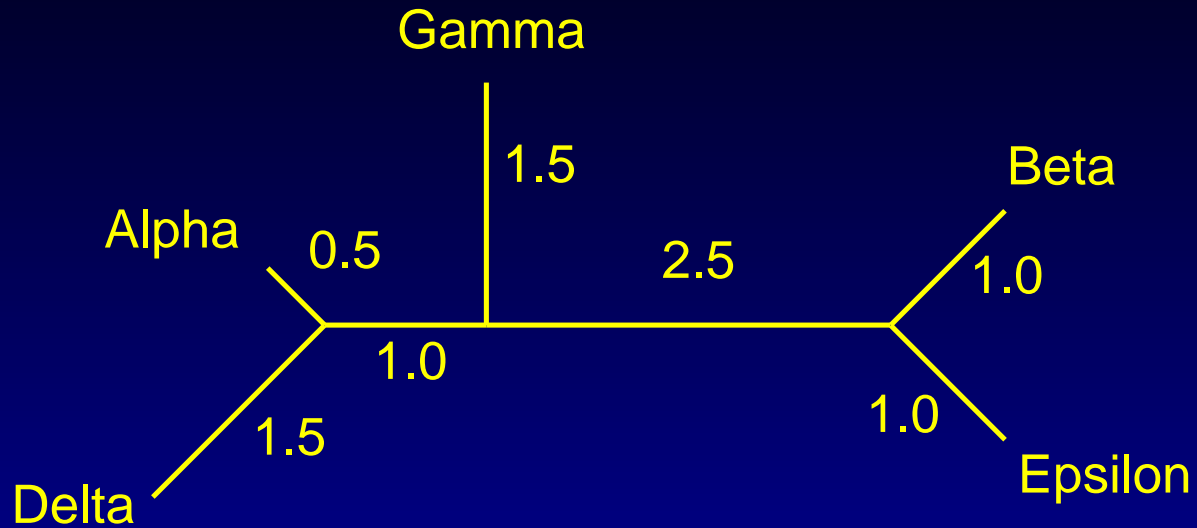
## The previous tree, rerooted (still 8 steps)



# State reconstruction on an unrooted tree



# Branch lengths



Averaged over all state reconstructions. This is not the most parsimonious tree but the first one we saw.

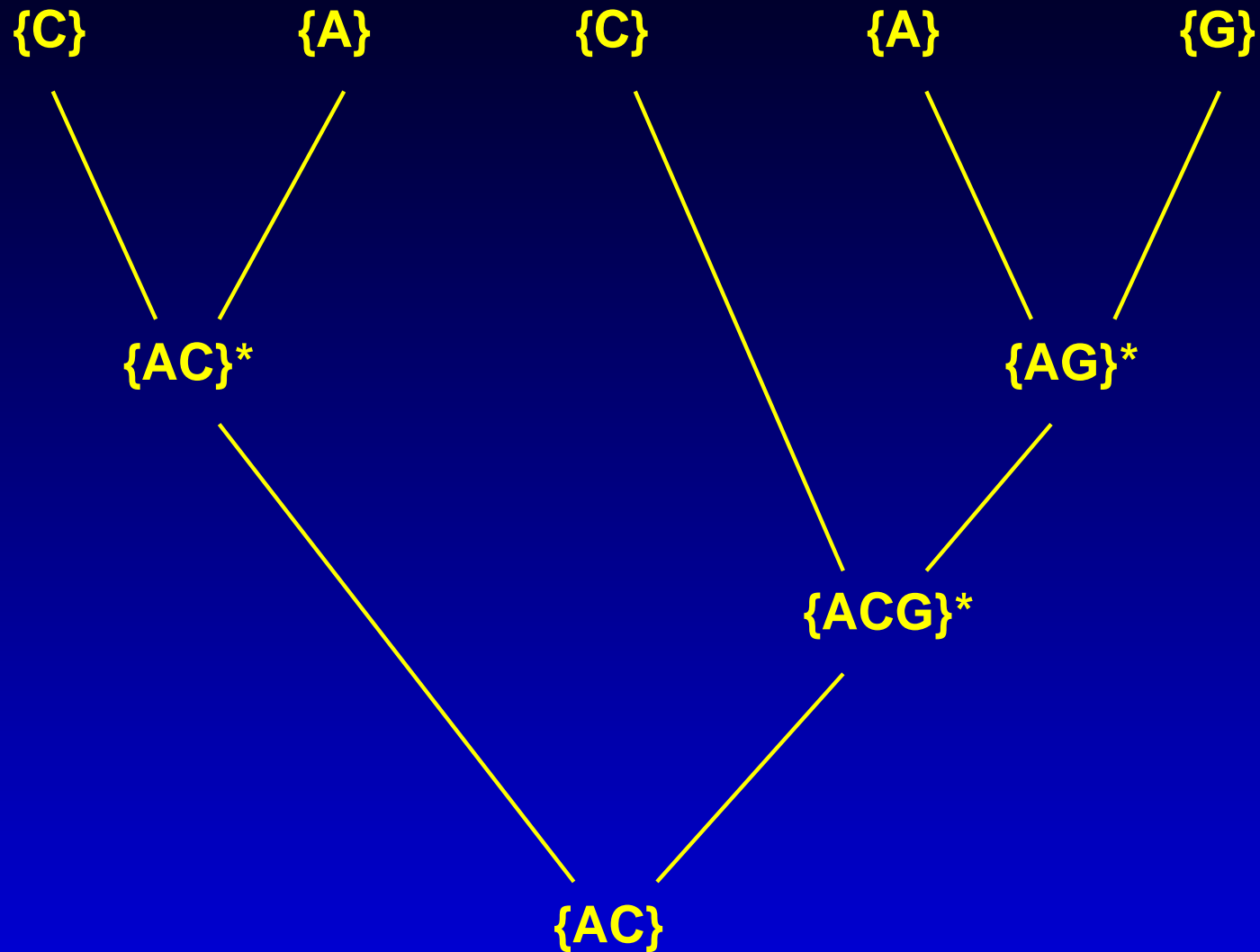
## Fitch's algorithm (for nucleotide sequences):

To count the number of steps a tree requires at a given site, start by constructing a set of nucleotides that are observed there (ambiguities are handled by having all of the possible nucleotides be there).

Go down the tree (postorder tree traversal). For each node of the tree consider its two immediate descendants' sets,  $S$  and  $T$ , and

- If  $S \cap T \neq \emptyset$ , write it down as the set in that node,
- If  $S \cap T = \emptyset$ , write down  $S \cup T$  and count one step.

# Fitch's algorithm counting the numbers of state changes



## Sankoff's algorithm

A dynamic programming algorithm for counting the smallest number of possible (weighted) state changes needed on a given tree.

Let  $S_j(i)$  be the smallest (weighted) number of steps needed to evolve the subtree at or above node  $j$ , given that node  $j$  is in state  $i$ .

Suppose that  $c_{ij}$  is the cost of going from state  $i$  to state  $j$ .

Initially, at tip (say)  $j$

$$S_j(\mathbf{i}) = \begin{cases} 0 & \text{if node } j \text{ has (or could have) state } i \\ \infty & \text{if node } j \text{ has any other state} \end{cases}$$

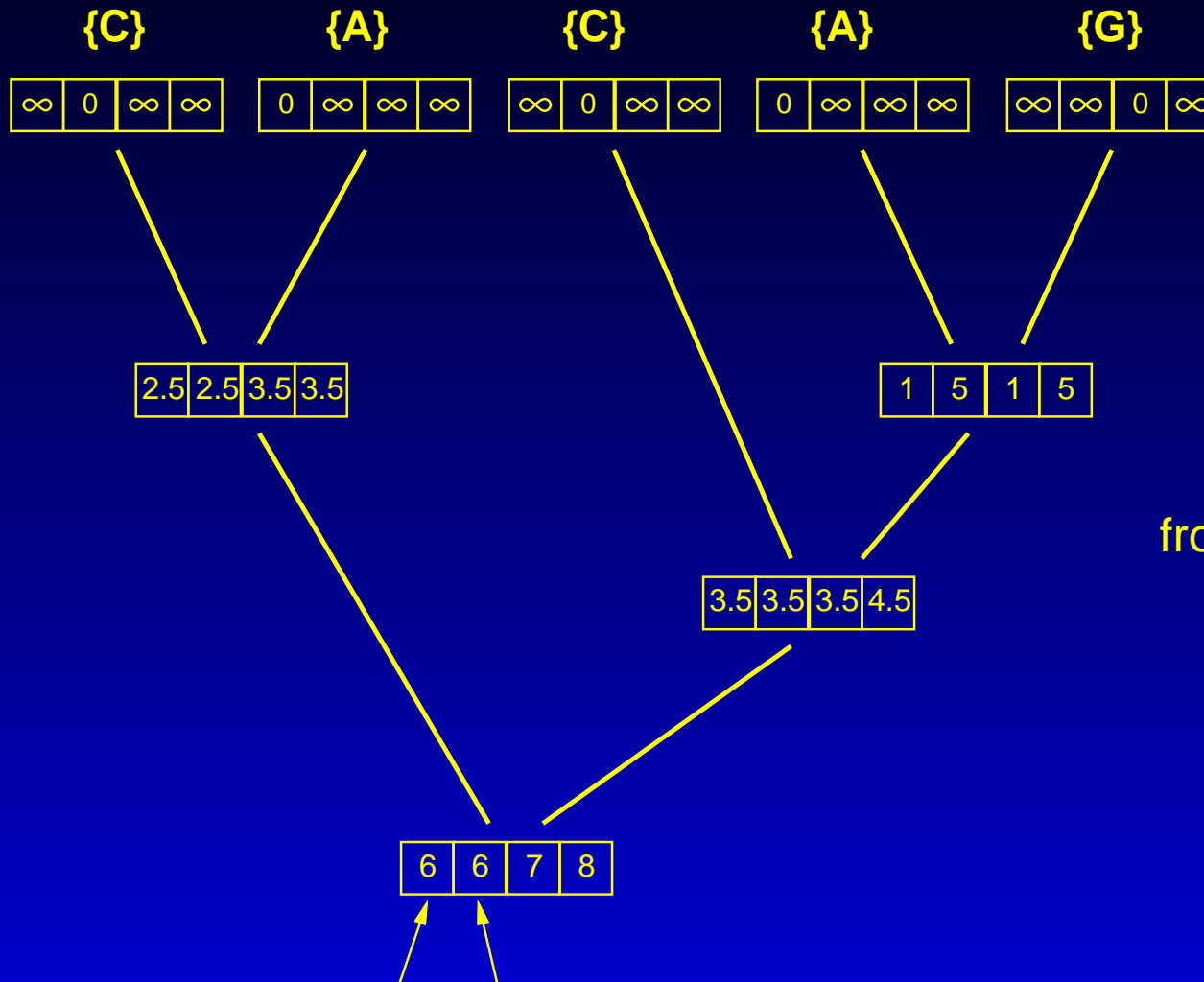
## Sankoff's algorithm (continued)

Then proceeding down the tree (postorder tree traversal) for node  $a$  whose immediate descendants are  $\ell$  and  $r$

$$\mathbf{S}_a(\mathbf{i}) = \min_j [ \mathbf{c}_{ij} + \mathbf{S}_\ell(\mathbf{j}) ] + \min_k [ \mathbf{c}_{ik} + \mathbf{S}_r(\mathbf{k}) ]$$

The minimum number of (weighted) steps for the tree is found by computing at the bottom node (0) the  $\mathbf{S}_0(\mathbf{i})$  and taking the smallest of these.

# An example using Sankoff's algorithm



cost matrix:

from \ to	A	C	G	T
A	0	2.5	1	2.5
C	2.5	0	2.5	1
G	1	2.5	0	2.5
T	2.5	1	2.5	0

# Compatibility

Compatibility is an alternative to parsimony. Instead of evaluating a tree by the sum of steps over all characters, we score each character as being either compatible with the tree or not. For one of our trees:

Species	Sites					
	1	2	3	4	5	6
Alpha	T	A	G	C	A	T
Beta	C	A	A	G	C	T
Gamma	T	C	G	G	C	T
Delta	T	C	G	C	A	A
Epsilon	C	A	A	C	A	T
States-1	1	1	1	1	1	1
Steps	2	2	2	1	1	1
Compatible?	n	n	n	y	y	y

Want to find the largest set of characters all compatible with the same tree.

# Compatibility Method

Two states are compatible if there exists a tree on which both could evolve with no extra changes of state.

**Pairwise Compatibility Theorem.** A set  $S$  of characters has all pairs of characters compatible with each other if and only if all of the characters in the set are jointly compatible (in that there exists a tree with which all of them are compatible).

(True for what kinds of characters?)

The compatibility test for sites 1 and 2 of the example data is:

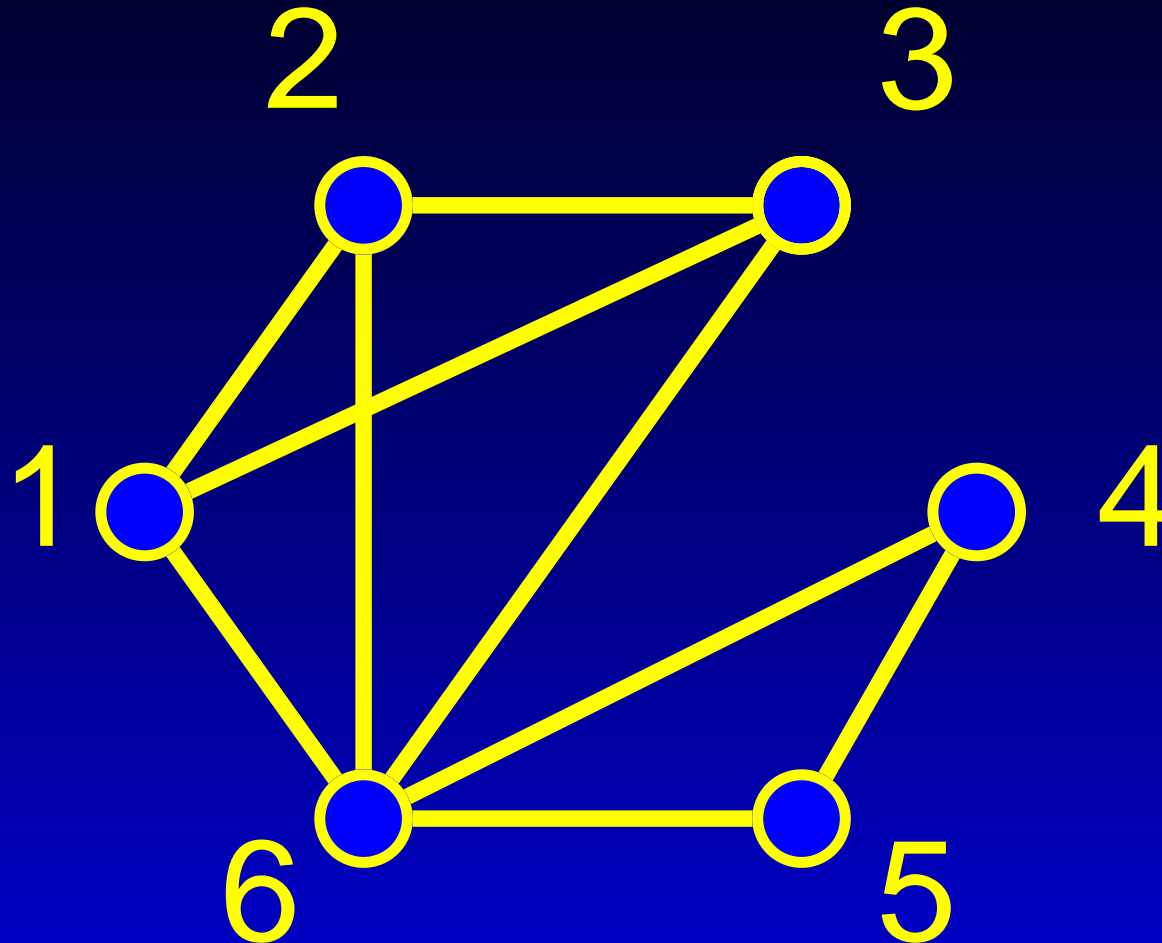
	site 2	C	A
site 1			
T		X	X
C			X

# Compatibility matrix for our example data set

	1	2	3	4	5	6
1	compatible	compatible	compatible	not	not	compatible
2	compatible	compatible	compatible	not	not	compatible
3	compatible	compatible	compatible	not	not	compatible
4	not	not	not	compatible	compatible	compatible
5	not	not	not	compatible	compatible	compatible
6	compatible	compatible	compatible	compatible	compatible	compatible

Legend:  
compatible (shaded cell)  
not (unshaded cell)

## The graph of pairwise compatibility



There are two “maximal cliques”, one larger than the other.

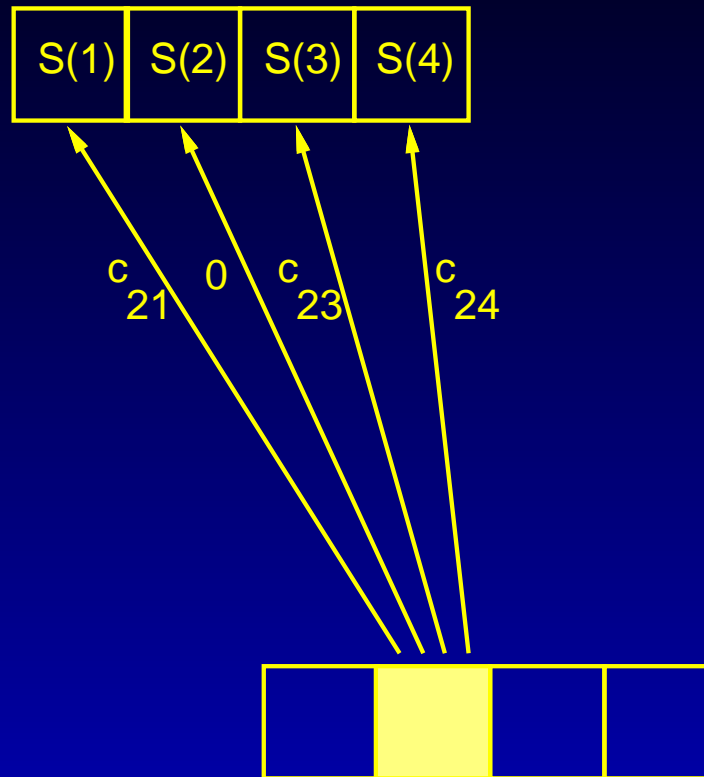


## Fitch's counterexample

Fitch's set of nucleotide sequences that have each pair of sites compatible, but which are not all compatible with the same tree.

Alpha	A	A	A
Beta	A	C	C
Gamma	C	G	C
Delta	C	C	G
Epsilon	G	A	G

# Reconstruction of ancestral states



The shaded state is the one that has been reconstructed at the lower of these two nodes in the tree. To decide what to reconstruct above it, we choose the smallest of  $c_{2i} + S(i)$



## Some references

- Edwards, A. W. F., and L. L. Cavalli-Sforza. 1964. Reconstruction of evolutionary trees. pp. 67-76 in *Phenetic and Phylogenetic Classification*, ed. V. H. Heywood and J. McNeill. Systematics Association Publ. No. 6, London. [The first parsimony paper, using gene frequencies]
- Camin, J. H. and R. R. Sokal. 1965. A method for deducing branching sequences in phylogeny. *Evolution* 19: 311-326. [The second parsimony paper, on discrete morphological characters]
- Eck, R. V. and M. O. Dayhoff. 1966. *Atlas of Protein Sequence and Structure 1966*. National Biomedical Research Foundation, Silver Spring, Maryland. [First parsimony on molecular sequences]

## references, cont'd

- Kluge, A. G. and J. S. Farris. 1969. Quantitative phyletics and the evolution of anurans. *Systematic Zoology* 18: 1-32. [An algorithm for parsimony with symmetrical change along a linear series of ordered states]
- Le Quesne, W. J. 1969. A method of selection of characters in numerical taxonomy. *Systematic Zoology* 18: 201-205. [Compatibility method]
- Estabrook, G. F., and F. R. McMorris. 1980. When is one estimate of evolutionary relationships a refinement of another? *Journal of Mathematical Biology* 10: 367-373. [Best proof of the Pairwise Compatibility Theorem]
- Fitch, W. M. 1971. Toward defining the course of evolution: minimum change for a specified tree topology. *Systematic Zoology* 20: 406-416. [The Fitch algorithm]

## references, cont'd

Sankoff, D. 1975. Minimal mutation trees of sequences. *SIAM Journal of Applied Mathematics* **28**: 35-42. [The Sankoff algorithm]

Swofford, D. L., G. J. Olsen, P. J. Waddell, and D. M. Hillis. 1996. Phylogenetic inference. pp. 407-514 in *Molecular Systematics*, 2nd ed., ed. D. M. Hillis, C. Moritz, and B. K. Mable. Sinauer Associates, Sunderland, Massachusetts. [Good review written for biologists]

Felsenstein, J. 2004. *Inferring Phylogenies*. Sinauer Associates, Sunderland, Massachusetts. [The best possible book on phylogenetic inference, of course]

## How it was done

This projection produced

- using the `prosper` style in LaTeX,
- using LaTeX to make a `.dvi` file,
- using `dvips` to turn this into a Postscript file,
- using `ps2pdf` to mill it into a PDF file, and
- displaying the slides in Adobe Acrobat Reader.

Result: nice slides using freeware.