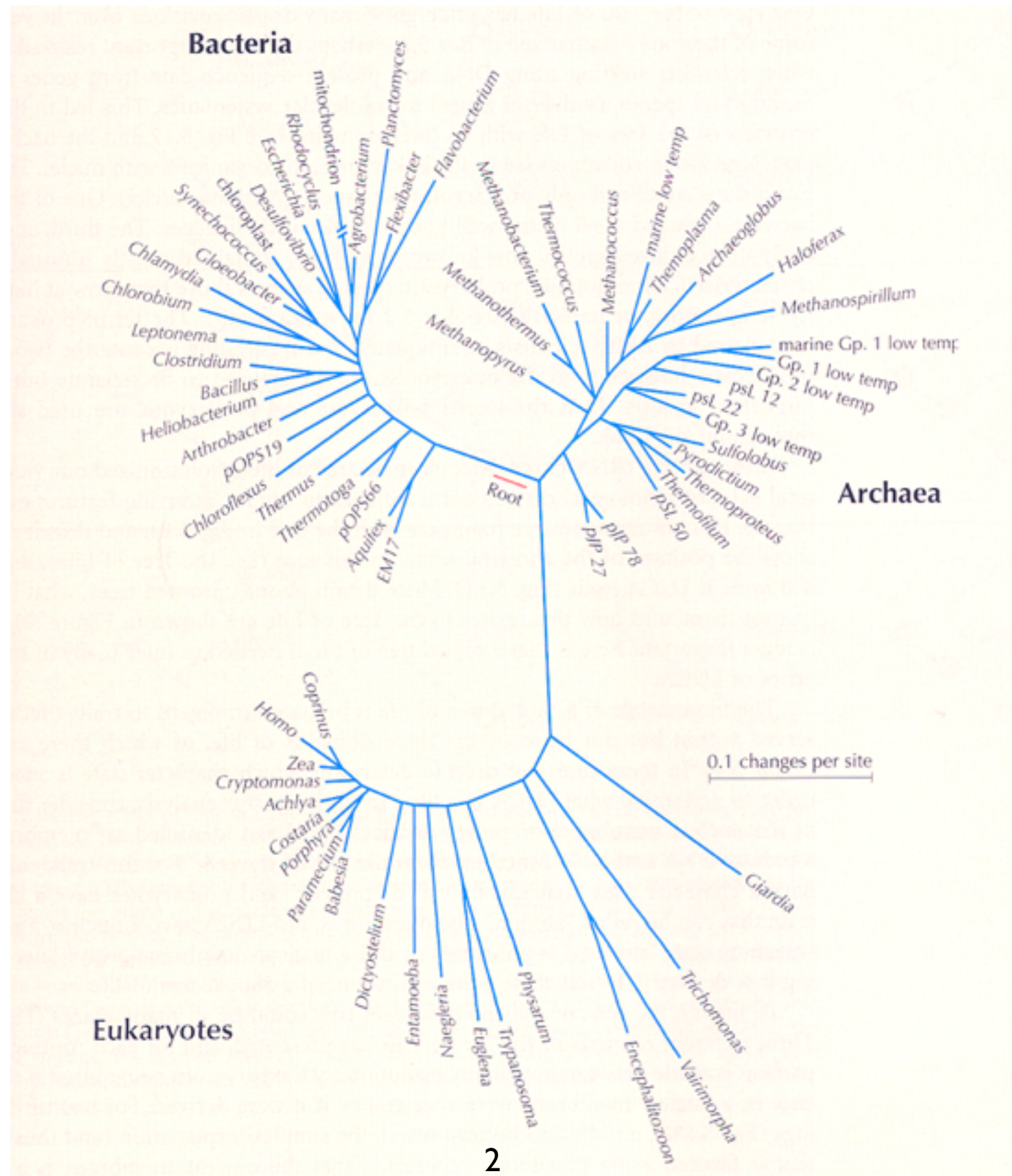


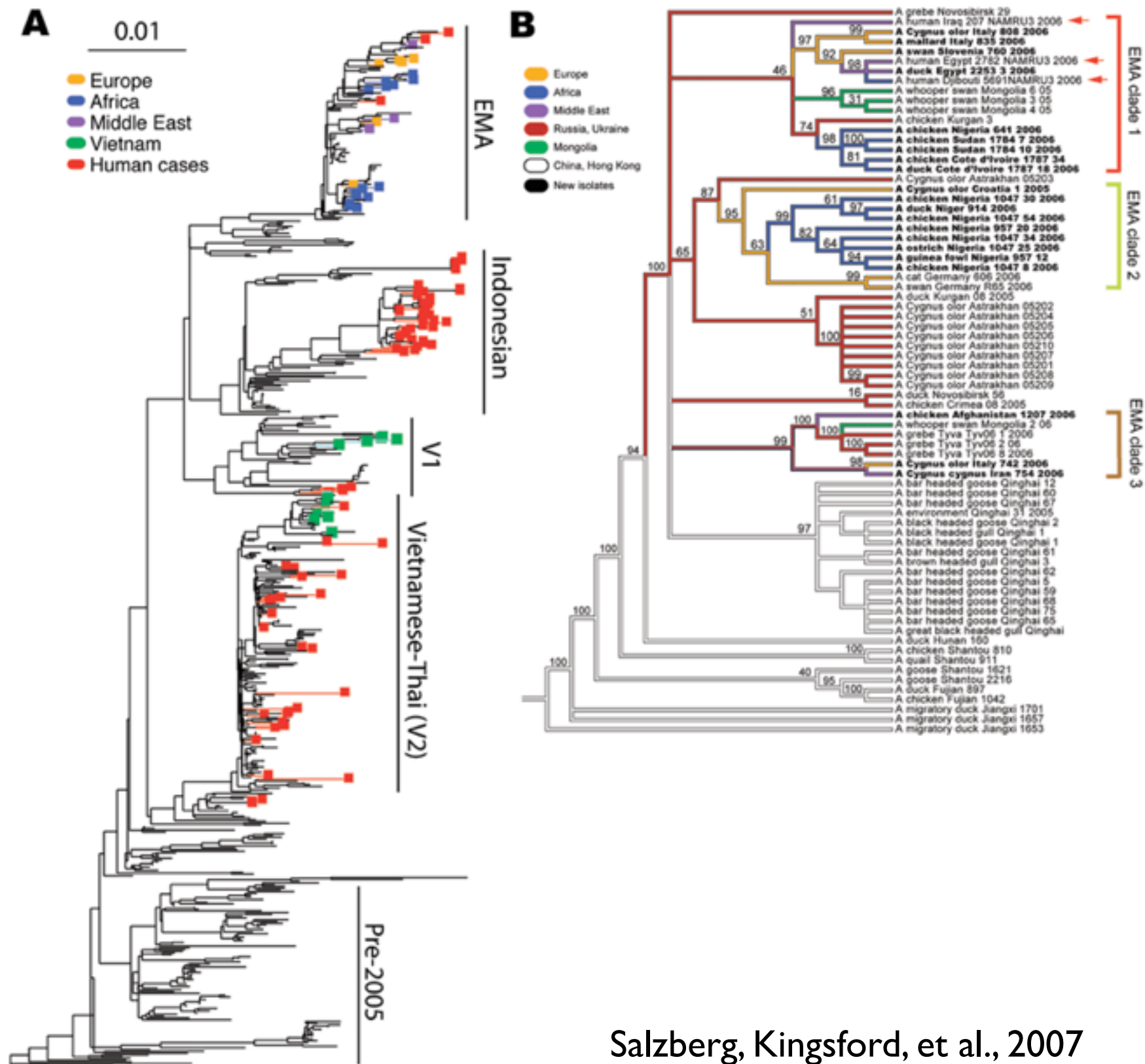
# CSE 549: Computational Biology

Phylogenomics

# Tree of Life



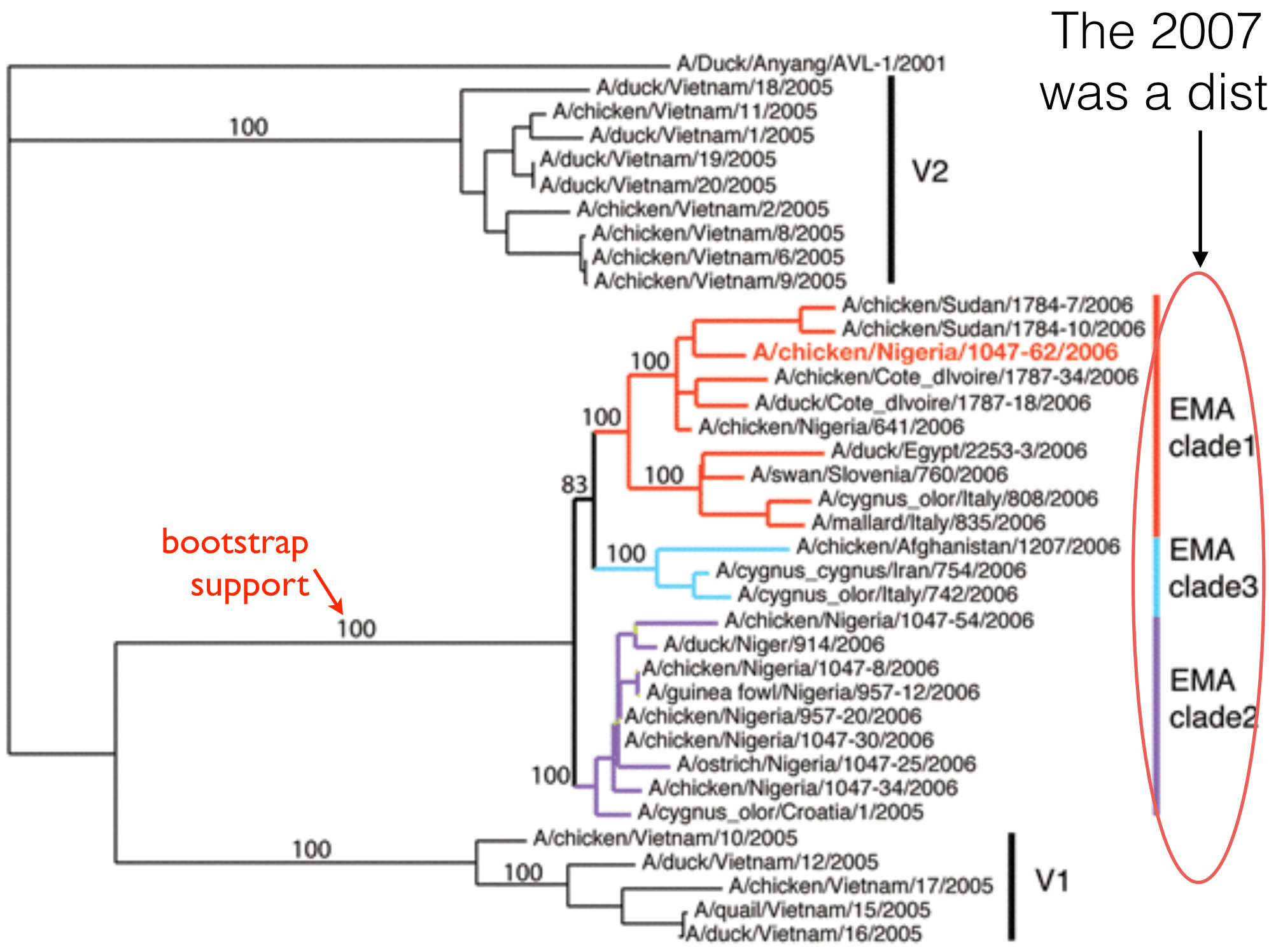
# H5N1 Influenza Strains



Salzberg, Kingsford, et al., 2007



# H5N1 Influenza Strains



The 2007 outbreak was a distinct strain

# Questions Addressable by Phylogeny

- How many times has a feature arisen? been lost?
- How is a disease evolving to avoid immune system?
- What is the sequence of ancestral proteins?
- What are the most similar species?
- What is the rate of speciation?
- Is there a correlation between gain/loss of traits and environment? with geographical events?
- Which features are ancestral to a clade, which are derived?
- What structures are homologous, which are analogous?

# Study Design Considerations

- **Taxon sampling:**

- how many individuals are used to represent a species?
- how is the “outgroup” chosen?
- Can individuals be collected or cultured?

- **Marker selection: Sequence features should:**

- be Representative of evolutionary history (unrecombined)
- have a single copy
- be able to be amplified using PCR
- able to be sequenced
- change enough to distinguish species, similar enough to perform MSA

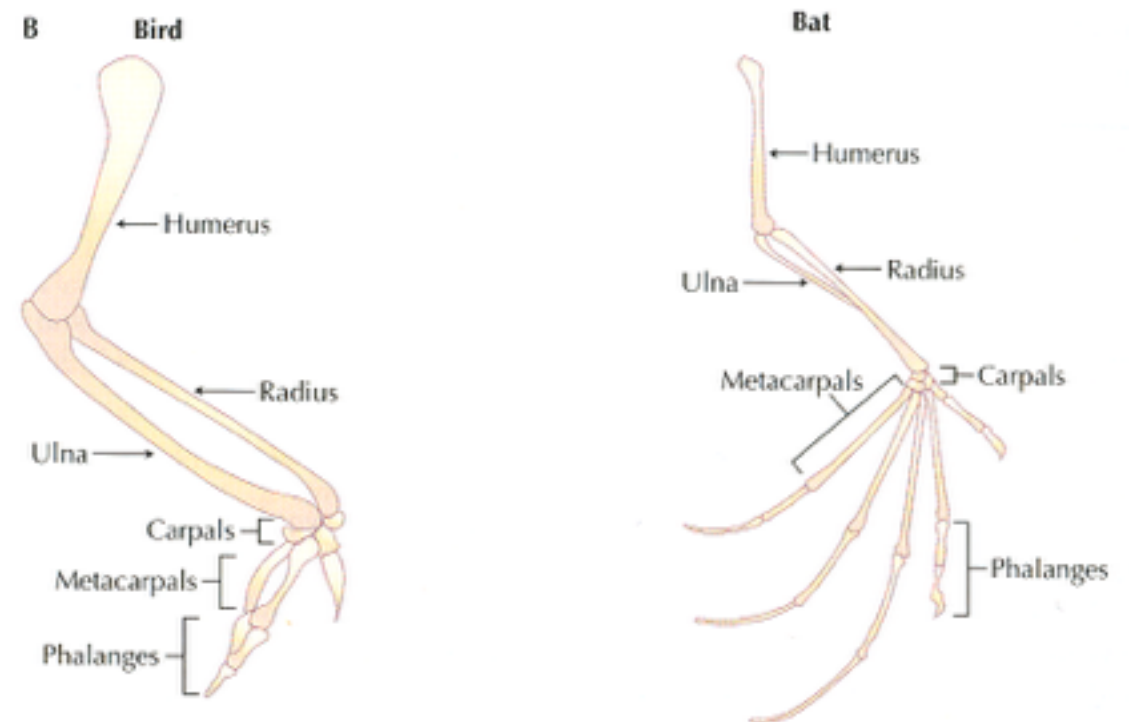
# Convergent Evolution



Bird & bat wings arose independently (analogous)

“Has wings” is thus a “bad” trait for phylogenetic inference

Bone structure has common ancestor (homologous)





# “Divergent” Evolution



“Obvious” phenotypic traits are not necessarily good markers

These are all one species!

**FIGURE 3.7.** Diverse varieties of *Brassica oleracea* include (A) cabbage; (B) broccoli; (C) cauliflower; (D) brussels sprouts; and (E) flowering kale.



# Two phylogeny “problems”

*Note:* “Character” below is not a letter (e.g. A,C,G,T), but a particular characteristic under which we consider the phylogeny (e.g. column of a MSA). Each character takes on a *state* (e.g. A,C,G,T).

## The **small** phylogeny problem

**Given:** a set of characters at the leaves (extant species), a set of states for each character, the cost of transition from each state to every other, and the *topology* of the phylogenetic tree

**Find:** a labeling for each internal node that minimizes the *overall* cost of transitions.

## The **big** phylogeny problem

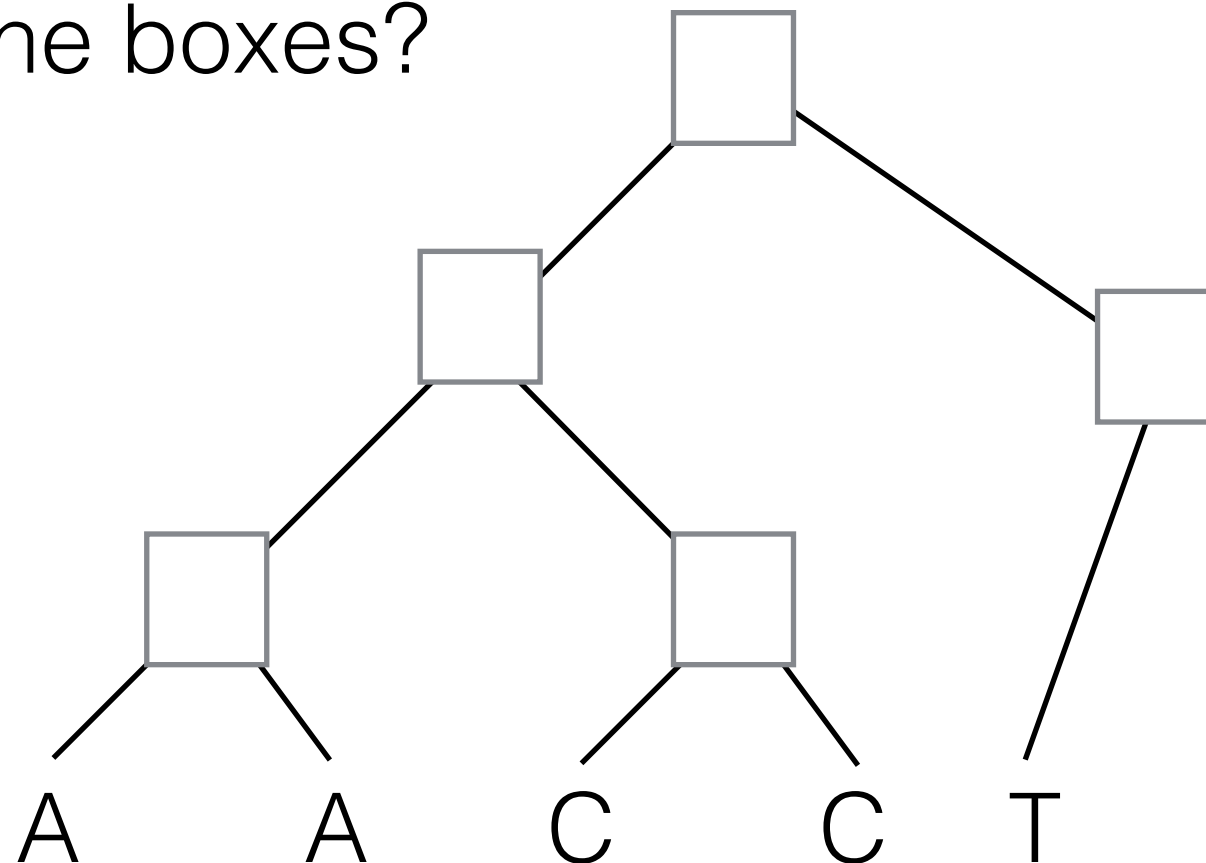
**Given:** a set of characters at the leaves (extant species), a set of states for each character, the cost of transition from each state to every other

**Find:** a tree topology and labeling for each internal node that minimizes the *overall* cost (over all trees and internal states)

# Small phylogeny problem — parsimony

One way to define the lowest *cost* set of transitions is to maximize *parsimony*. That is, posit as few transitions as necessary to produce the observed result.

What characters should appear in the boxes?



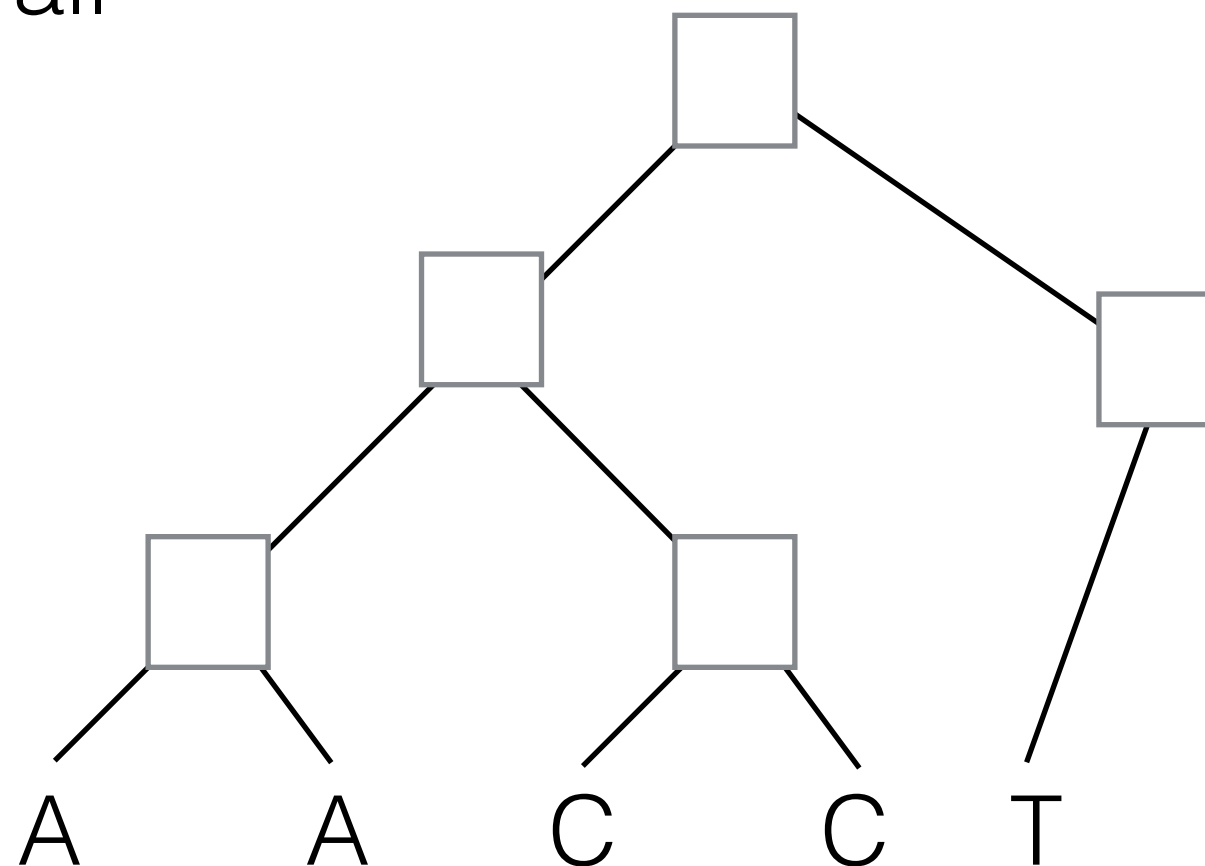


# Small phylogeny problem — parsimony

One way to define the lowest *cost* set of transitions is to maximize *parsimony*. That is, posit as few transitions as necessary to produce the observed result.

Assume transitions all have unit cost:

	A	C	G	T
A	0	1	1	1
C	1	0	1	1
G	1	1	0	1
T	1	1	1	0



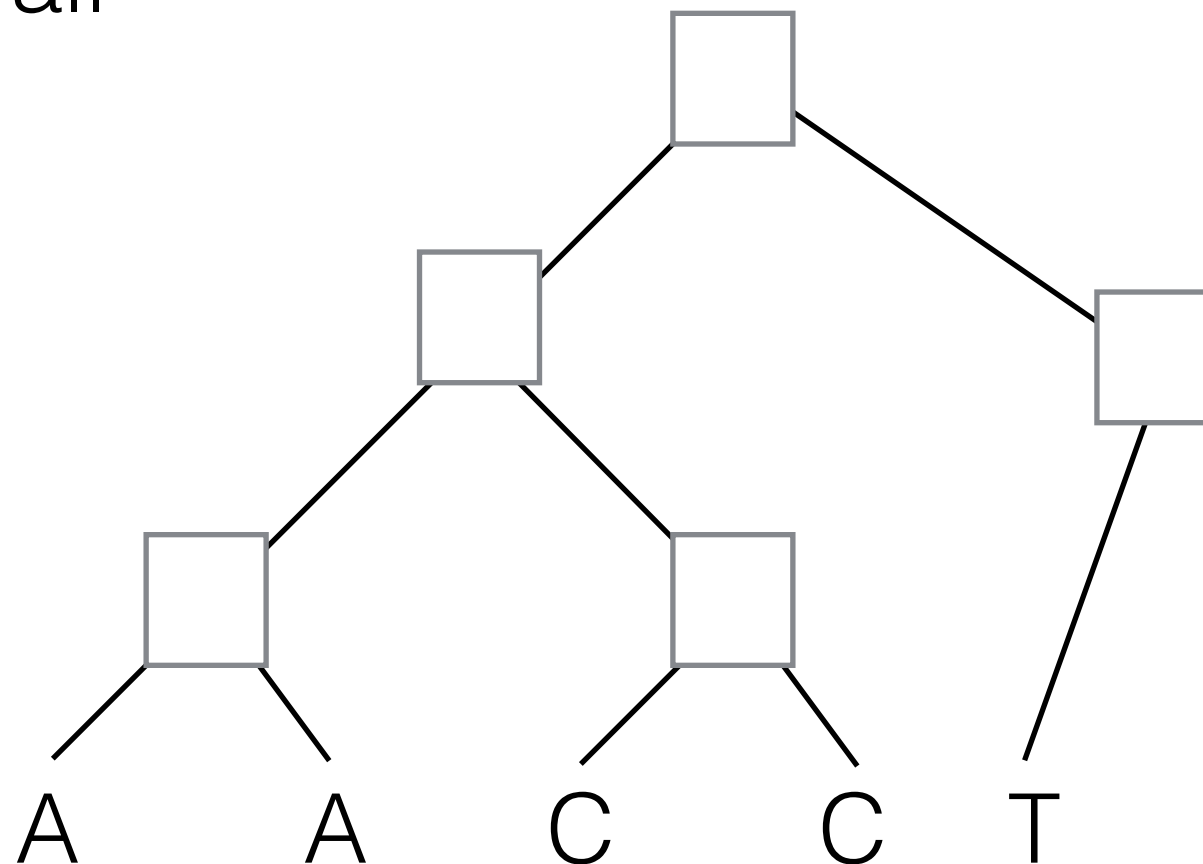
# Small phylogeny problem — parsimony

One way to define the lowest *cost* set of transitions is to maximize *parsimony*. That is, posit as few transitions as necessary to produce the observed result.

Assume transitions all have unit cost:

	A	C	G	T
A	0	1	1	1
C	1	0	1	1
G	1	1	0	1
T	1	1	1	0

Fitch's algorithm provides a solution.





# Small phylogeny problem — parsimony

Fitch's algorithm (2-pass):

Visit nodes in *post-order* traversal:

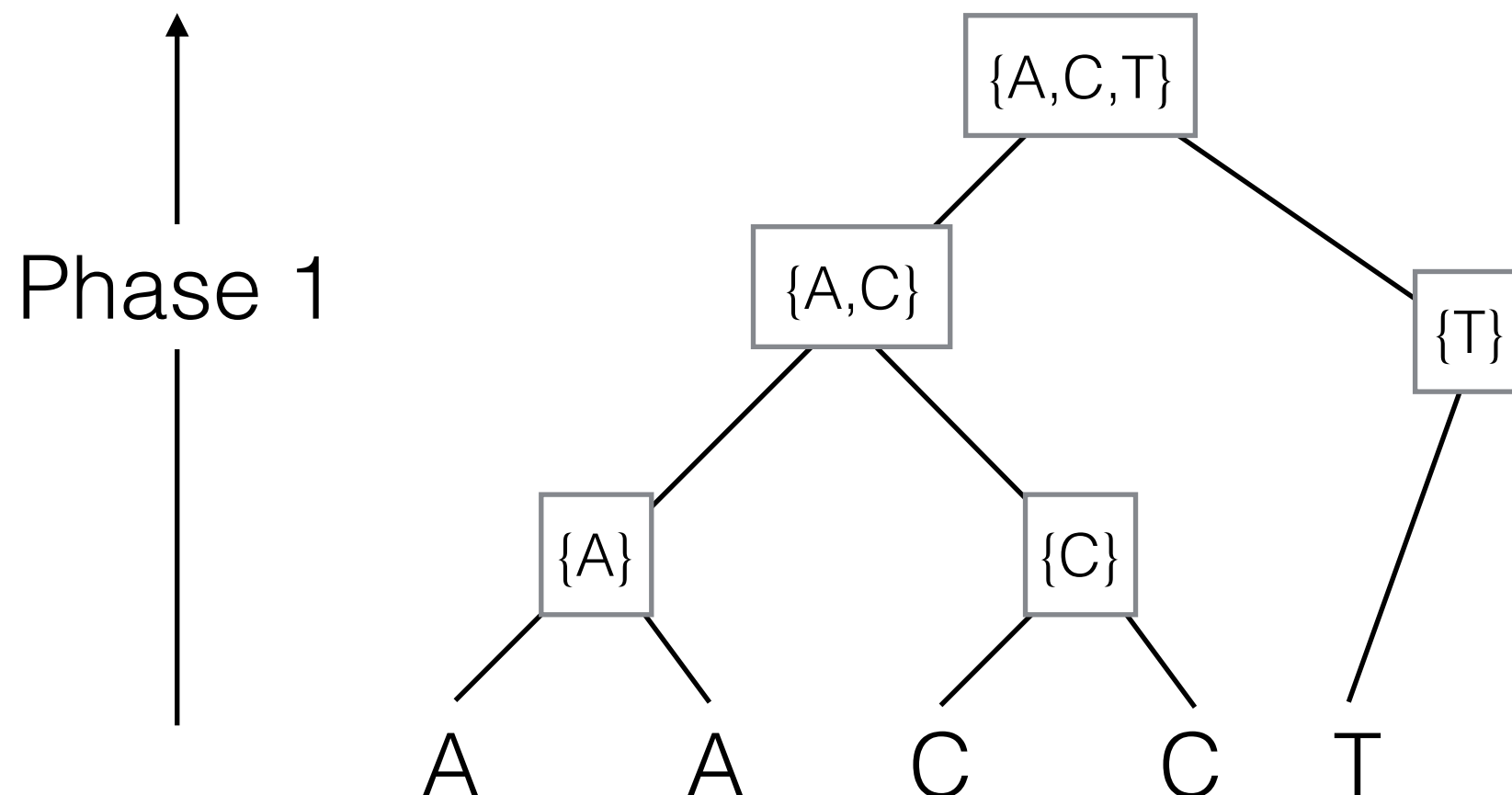
- store a *set* of characters at each node

- take the intersection of child's set if not empty; else take the union

Visit nodes in *pre-order* traversal:

- If a child's character set has its parent's label, choose it.

- Otherwise, select any character in this node's character set.



# Small phylogeny problem — parsimony

Fitch's algorithm (2-pass):

Visit nodes in *post-order* traversal:

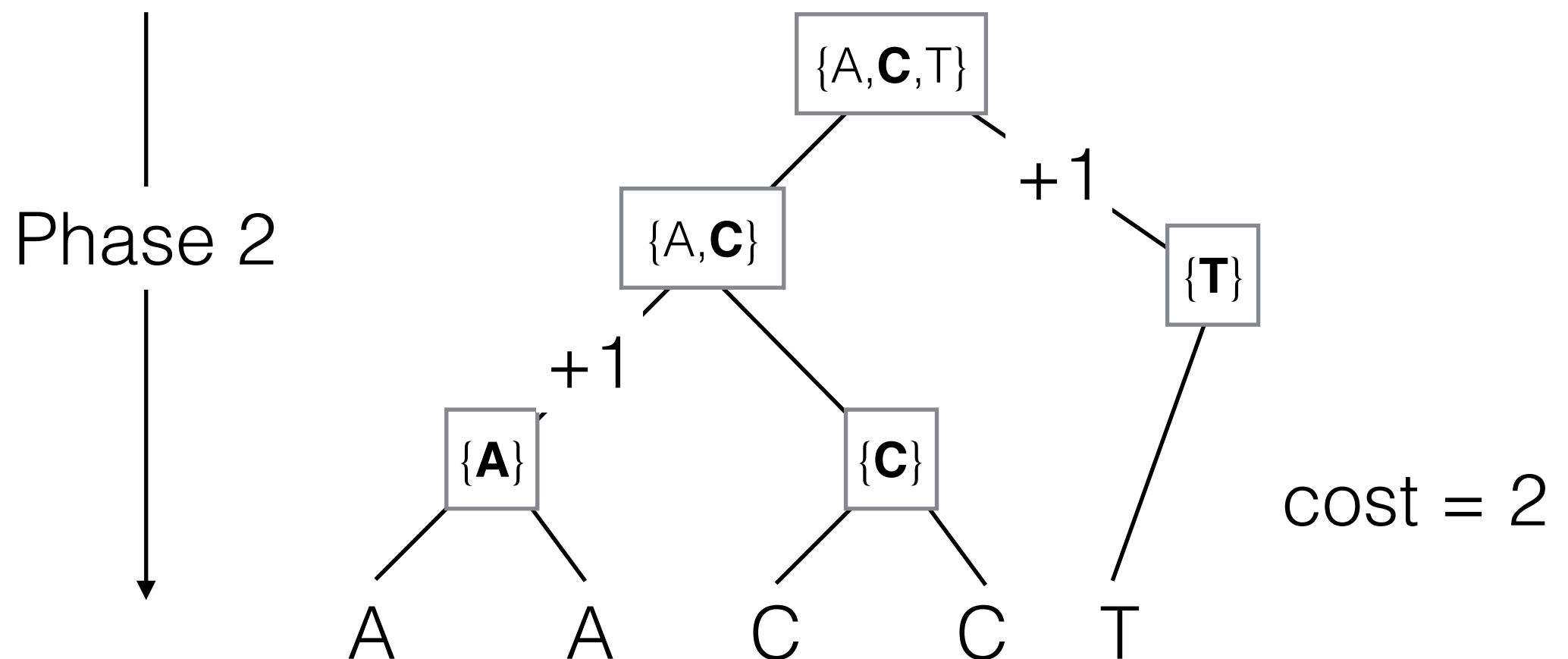
- store a *set* of characters at each node

- take the intersection of child's set if not empty; else take the union

Visit nodes in *pre-order* traversal:

- If a child's character set has its parent's label, choose it.

- Otherwise, select any character in this node's character set.





# Small phylogeny problem — parsimony

Fitch's algorithm (2-pass):

Visit nodes in *post-order* traversal:

store a *set* of characters at each node

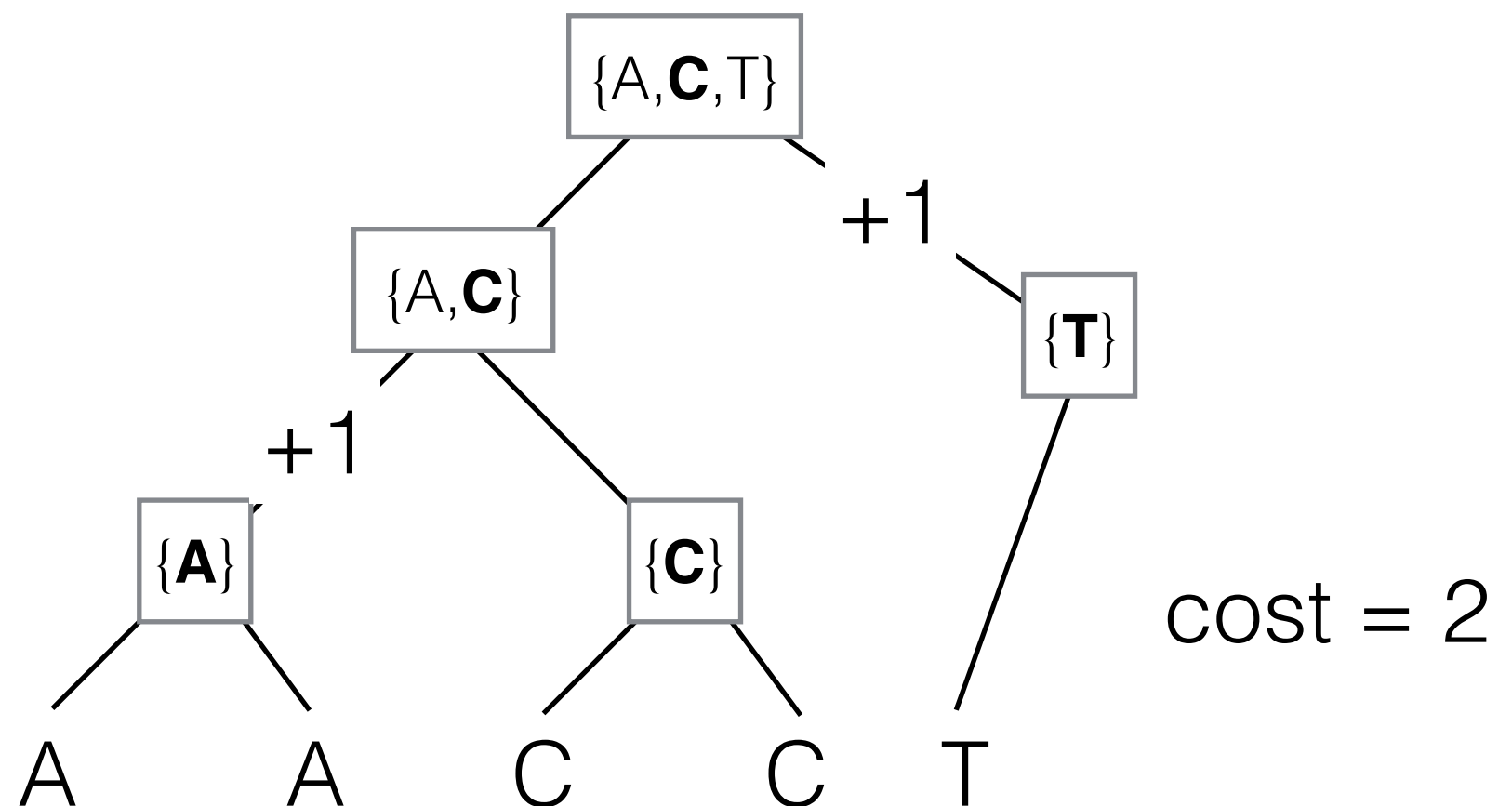
take the intersection of child's set if not empty; else take the union

Visit nodes in *pre-order* traversal:

If a child's character set has its parent's label, choose it.

Otherwise, select any character in this node's character set.

*Note:* There are generally *many* solutions of optimal cost.



# Small phylogeny problem — parsimony

What if there are different costs for each transition?

Sankoff\* provides a dynamic program to solve this case.

For simplicity, consider only a single character,  $c$

Phase 1 (post-order):

For each leaf  $v$ , state  $t$ , let  $S_t^c(v) = \begin{cases} 0 & \text{if } v_c = t \\ \infty & \text{otherwise} \end{cases}$

For each internal  $v$ , state  $t$ , let  $S_t^c(v) = \min_i \{C_{ti}^c + S_i^c(u)\} + \min_j \{C_{tj}^c + S_j^c(w)\}$

Phase 2 (pre-order):

Let the root take state  $r_c = \arg \min_t S_t^c(r)$

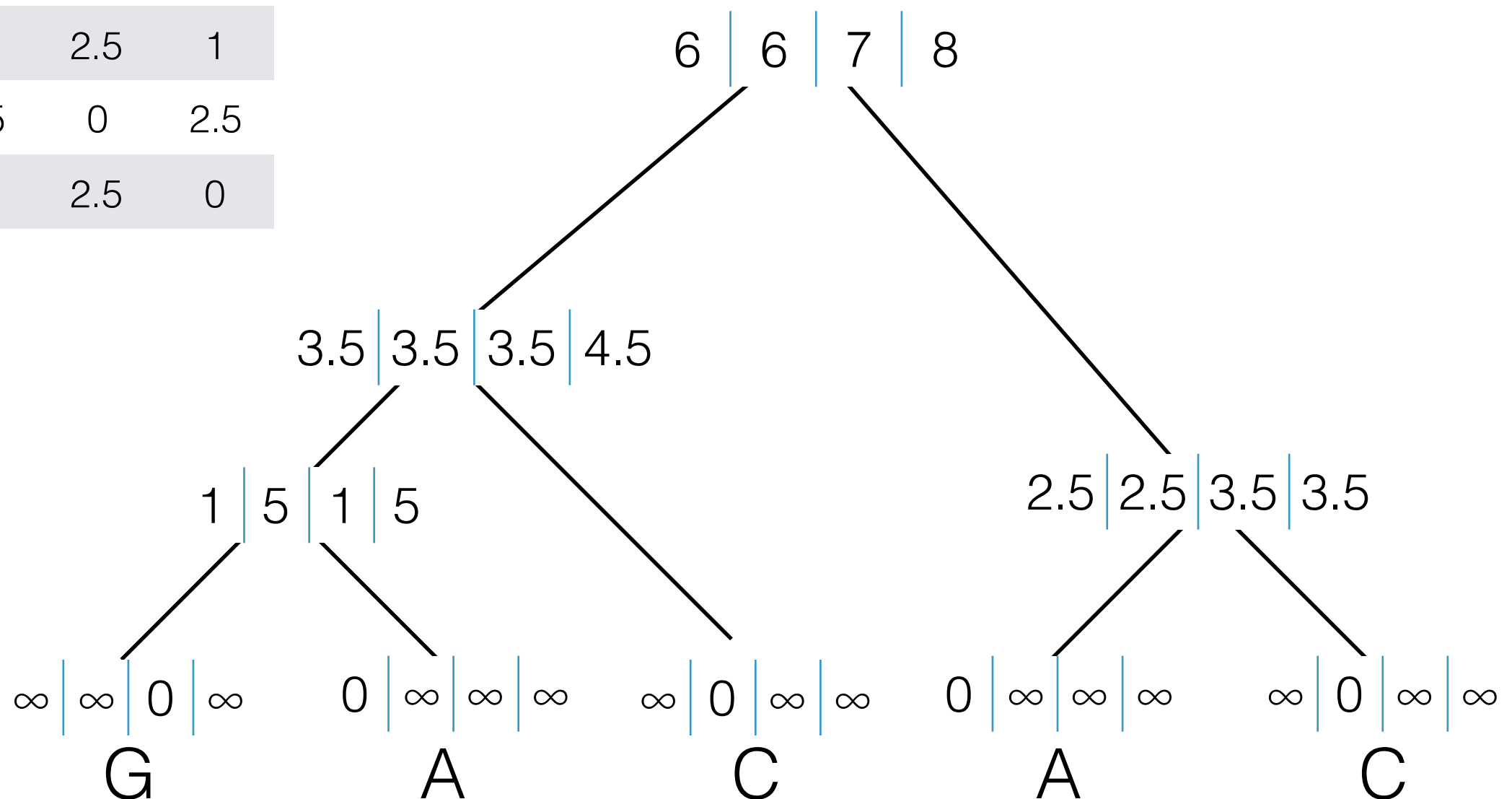
For all other  $v$  with parent  $u$ , let:  $v_c = \arg \min_t (C_{u_c t}^c + S_t^c(v))$



# Small phylogeny problem — parsimony

Consider the following tree and transition matrix:

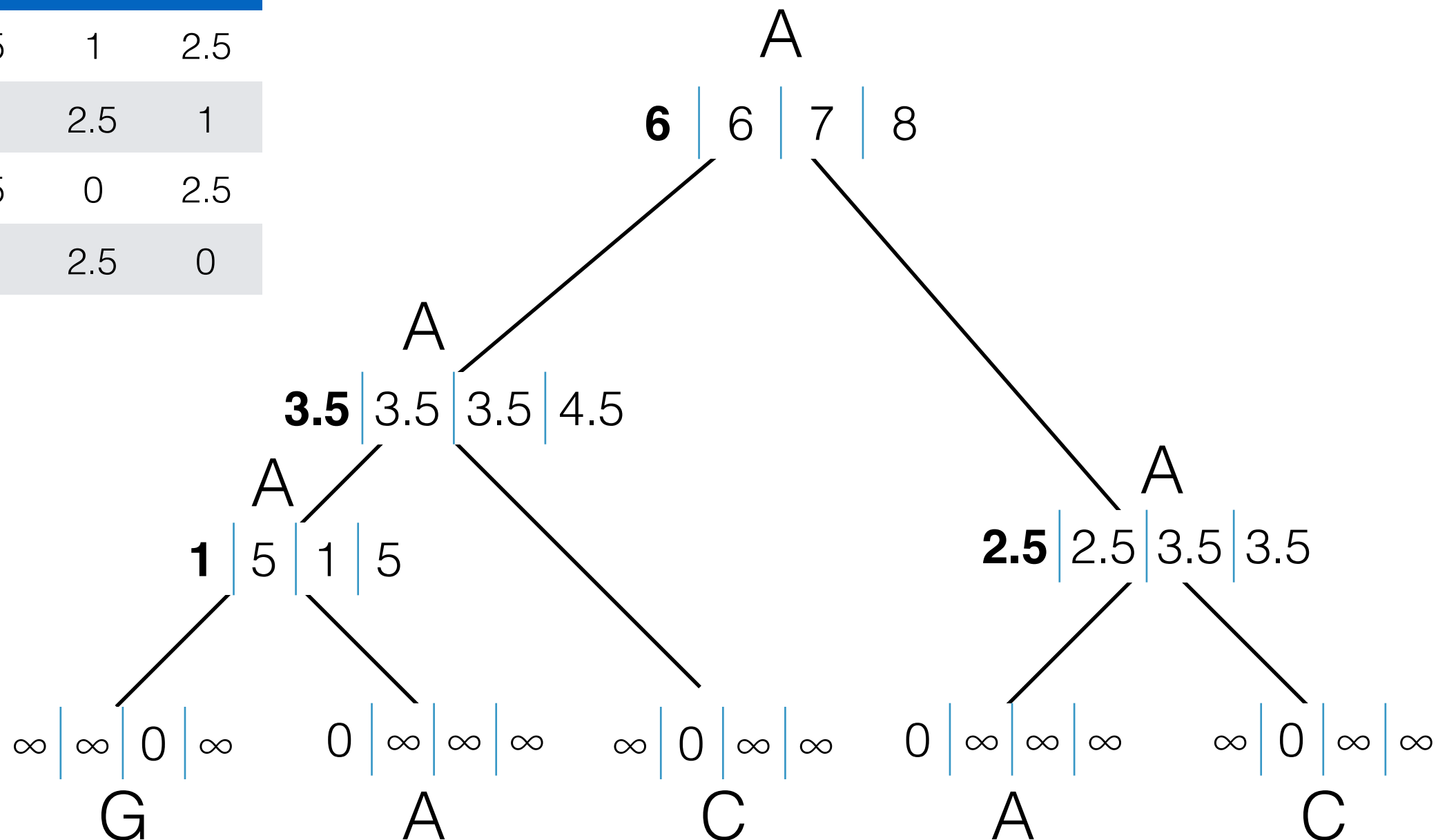
	A	C	G	T
A	0	2.5	1	2.5
C	2.5	0	2.5	1
G	1	2.5	0	2.5
T	2.5	1	2.5	0



# Small phylogeny problem — parsimony

Consider the following tree and transition matrix:

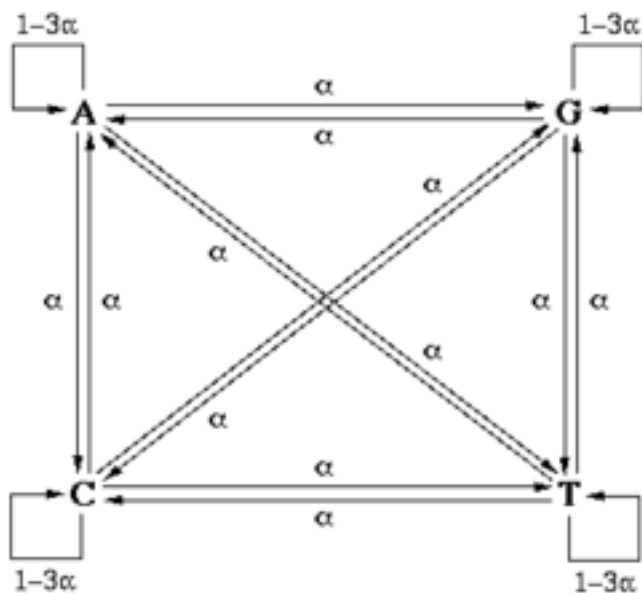
	A	C	G	T
A	0	2.5	1	2.5
C	2.5	0	2.5	1
G	1	2.5	0	2.5
T	2.5	1	2.5	0



# Small phylogeny problem — Maximum Likelihood

Imagine we assume a specific, probabilistic model of sequence evolution. For example:

Jukes-cantor



$\alpha$  is the probability to mutate (per-unit time)

or General Time Reversible

Time reversible:

$$\pi_i Q_{ij} = \pi_j Q_{ji}$$

Base frequencies:

$$\Pi = (\pi_T, \pi_C, \pi_A, \pi_G)$$

Rate matrix (per unit time):

$$Q = \begin{pmatrix} -(\alpha\pi_C + \beta\pi_A + \gamma\pi_G) & \alpha\pi_C & \beta\pi_A & \gamma\pi_G \\ \alpha\pi_T & -(\alpha\pi_T + \delta\pi_A + \epsilon\pi_G) & \delta\pi_A & \epsilon\pi_G \\ \beta\pi_T & \delta\pi_C & -(\beta\pi_T + \delta\pi_C + \eta\pi_G) & \eta\pi_G \\ \gamma\pi_T & \epsilon\pi_C & \eta\pi_A & -(\gamma\pi_T + \epsilon\pi_C + \eta\pi_A) \end{pmatrix}$$

Transition matrix at time t:

$$P(t) = e^{Qt} = \sum_{n=0}^{\infty} Q^n \frac{t^n}{n!}$$

$$\begin{aligned} \alpha &= r(T \rightarrow C) = r(C \rightarrow T) \\ \beta &= r(T \rightarrow A) = r(A \rightarrow T) \\ \gamma &= r(T \rightarrow G) = r(G \rightarrow T) \\ \delta &= r(C \rightarrow A) = r(A \rightarrow C) \\ \epsilon &= r(C \rightarrow G) = r(G \rightarrow C) \\ \eta &= r(A \rightarrow G) = r(G \rightarrow A) \end{aligned}$$

## Small phylogeny problem — Maximum Likelihood

Imagine we assume a specific, probabilistic model of sequence evolution.

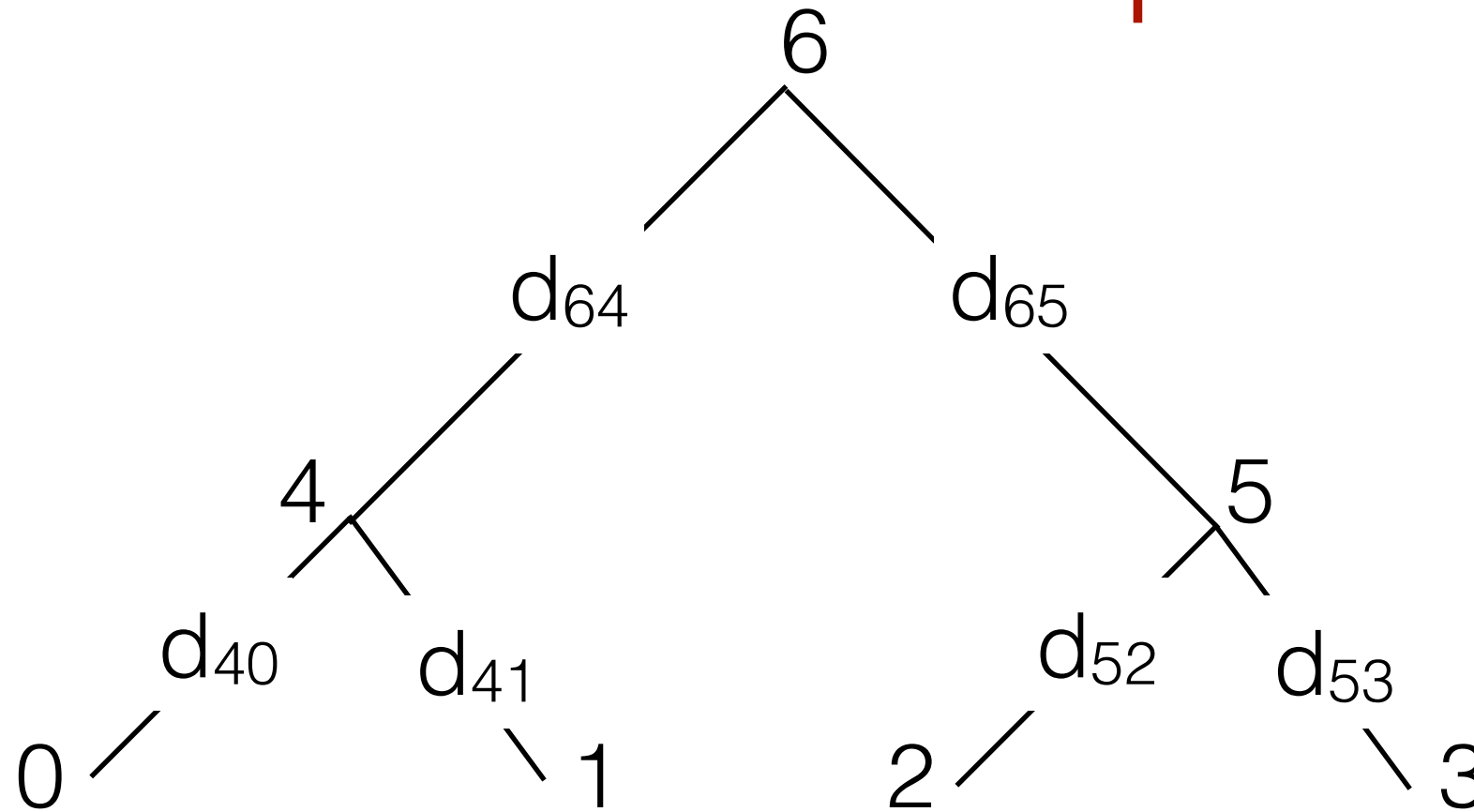
**Given** a tree topology (with branch lengths), a set of states for each character, and the assumed model of state evolution

**Find** the states at each internal node that *maximizes* the likelihood of the observed data (i.e. states at the leaves)

Rather than choosing the *best* state at each site, we are summing over the possibility of *all* states (phylogenetic histories)



# Consider the simple tree



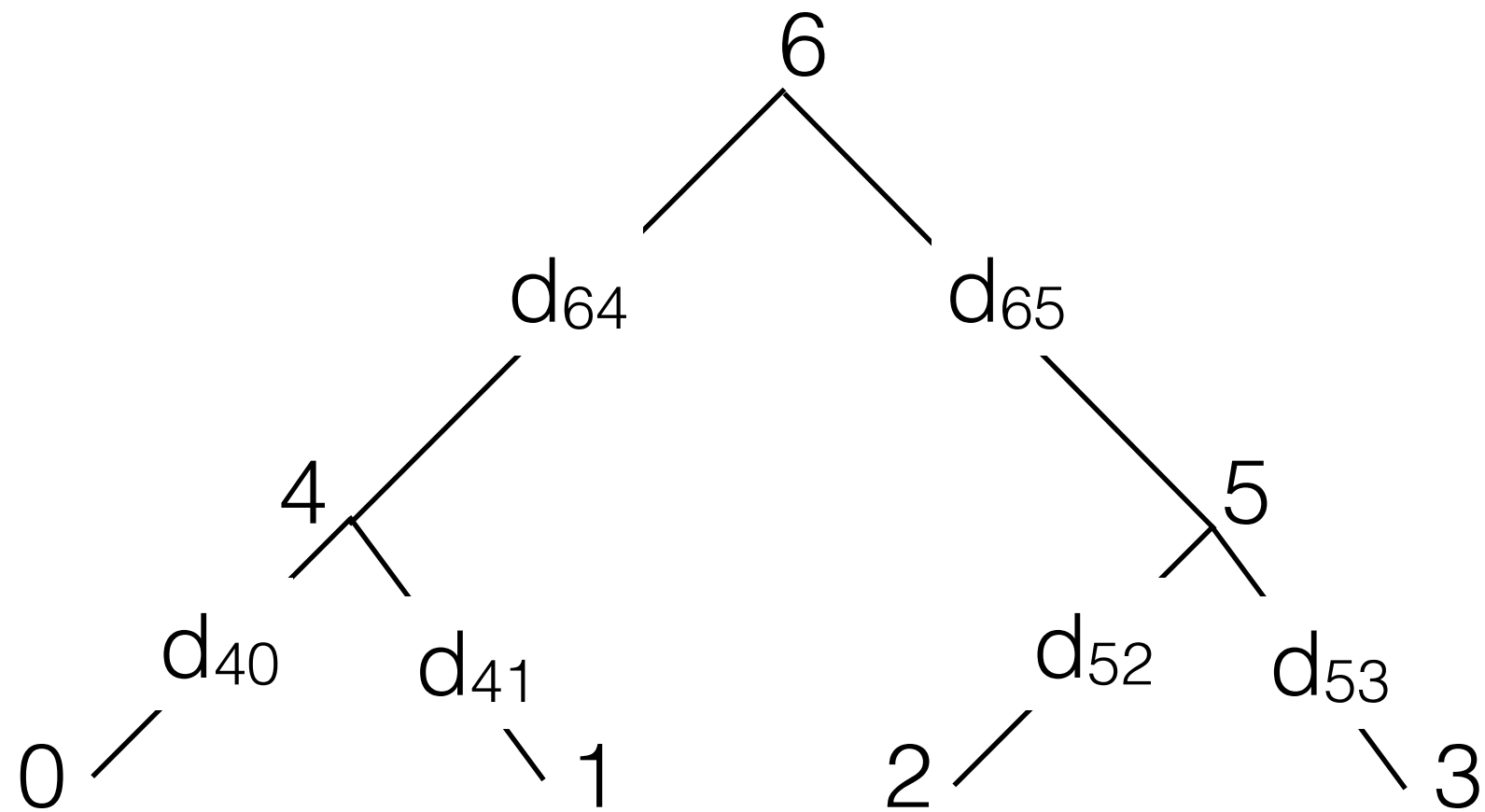
For particular ancestral states  $s_6$ ,  $s_4$  and  $s_5$ , we can score their likelihood as:

$$\tilde{\mathcal{L}}(s_6, s_4, s_5) = p_{s_6 \rightarrow s_4}(d_{64}) \cdot p_{s_6 \rightarrow s_5}(d_{65}) \cdot p_{s_4 \rightarrow s_0}(d_{40}) \cdot p_{s_4 \rightarrow s_1}(d_{41}) \cdot p_{s_5 \rightarrow s_2}(d_{52}) \cdot p_{s_5 \rightarrow s_3}(d_{53})$$

Since we don't know these states, we must *sum over* them:

$$\mathcal{L} = \sum_{s_6} \sum_{s_4} \sum_{s_5} \pi_{s_6} \tilde{\mathcal{L}}(s_6, s_4, s_5)$$

# Small phylogeny problem — Maximum Likelihood



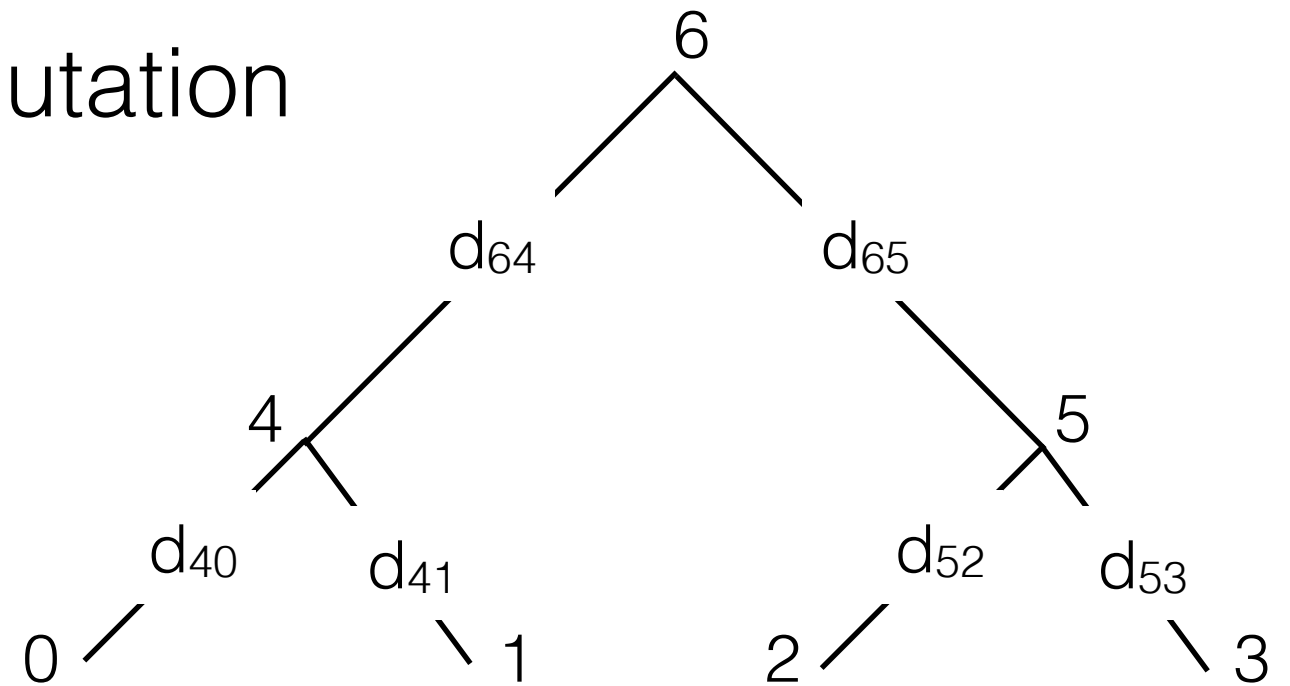
It turns out that this objective (maximum likelihood) can also be optimized in polynomial time.

This is done by re-arranging the terms and expressing them as conditional probabilities.

The algorithm is due to Felsenstein\* — again, it is a dynamic program

# Small phylogeny problem — Maximum Likelihood

Idea 1: Re-arrange the computation to be more favorable

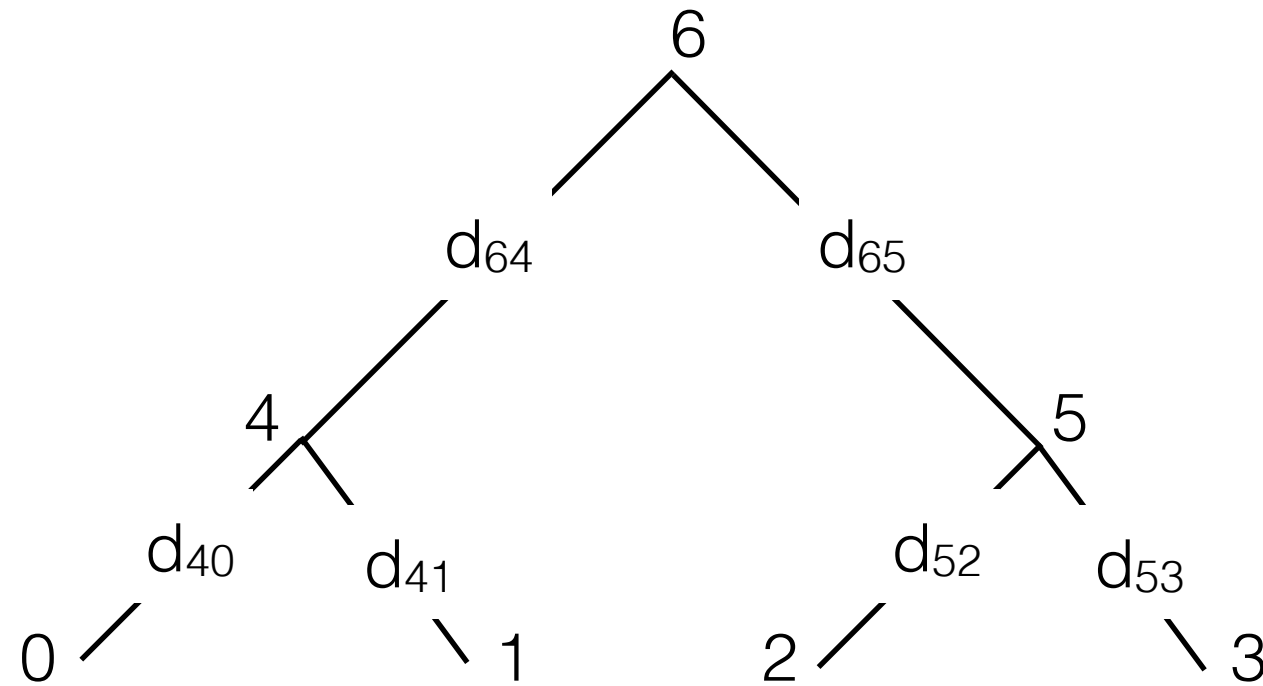


$$\mathcal{L} = \sum_{s_6} \sum_{s_4} \sum_{s_5} \pi_{s_6} \tilde{\mathcal{L}}(s_6, s_4, s_5)$$

via. Horner's method (push summations to the right)

$$= \sum_{s_6} \pi_{s_6} \times \left\{ \begin{array}{l} \sum_{s_4} p_{s_6 \rightarrow s_4} d(s_{64}) (p_{s_4 \rightarrow s_0} d(s_{40}) p_{s_4 \rightarrow s_1} d(s_{41})) \\ \times \\ \sum_{s_5} p_{s_6 \rightarrow s_5} d(s_{65}) (p_{s_5 \rightarrow s_2} d(s_{52}) p_{s_5 \rightarrow s_3} d(s_{53})) \end{array} \right\}$$

# Small phylogeny problem — Maximum Likelihood



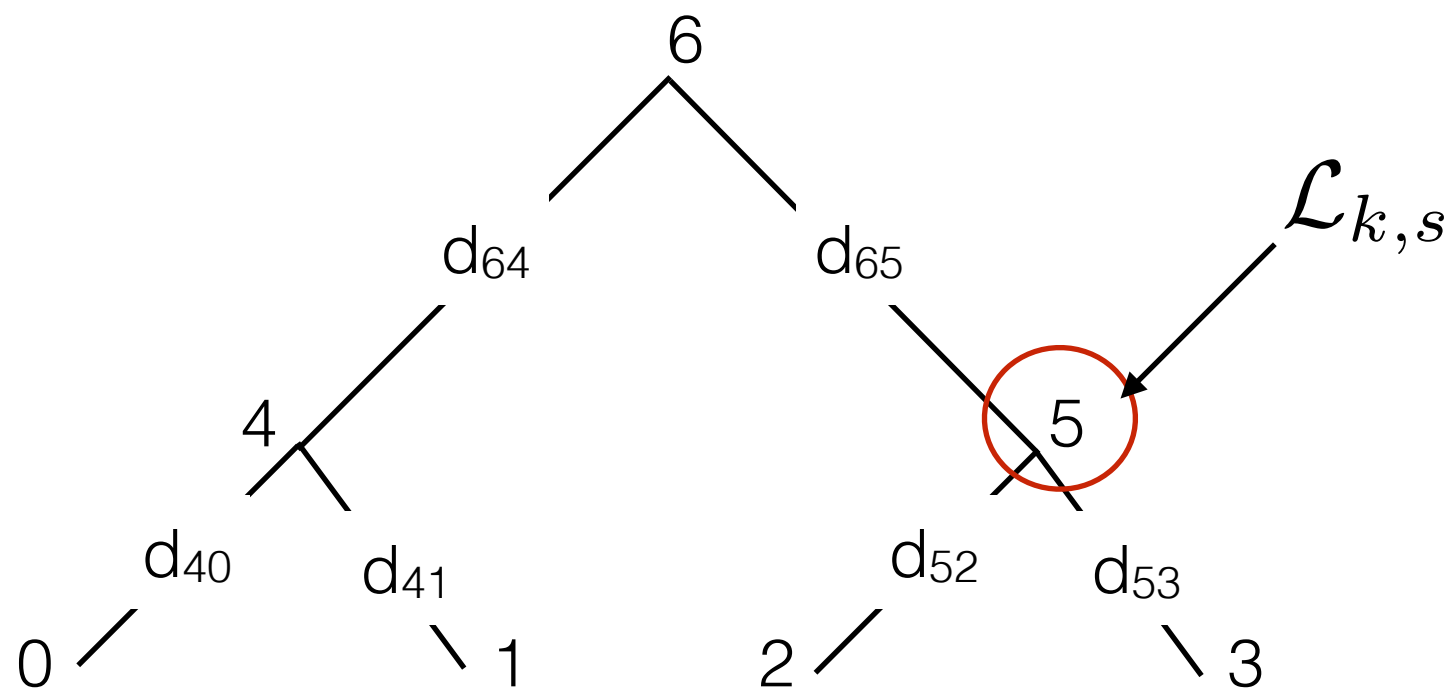
$$\sum_{s_6} \pi_{s_6} \times \left\{ \begin{array}{l} \sum_{s_4} p_{s_6 \rightarrow s_4} d(s_{64}) (p_{s_4 \rightarrow s_0} d(s_{40}) p_{s_4 \rightarrow s_1} d(s_{41})) \\ \times \\ \sum_{s_5} p_{s_6 \rightarrow s_5} d(s_{65}) (p_{s_5 \rightarrow s_2} d(s_{52}) p_{s_5 \rightarrow s_3} d(s_{53})) \end{array} \right\}$$

The structure of the equations here *matches* the structure of the tree  $((\dots)(\dots))$  — see e.g. nested parenthesis encoding of trees.



# Small phylogeny problem — Maximum Likelihood

Idea 2: define the total likelihood in terms of *conditional* likelihoods.



Conditional likelihood of the *subtree rooted at k*, assuming *k* takes on states *s*.

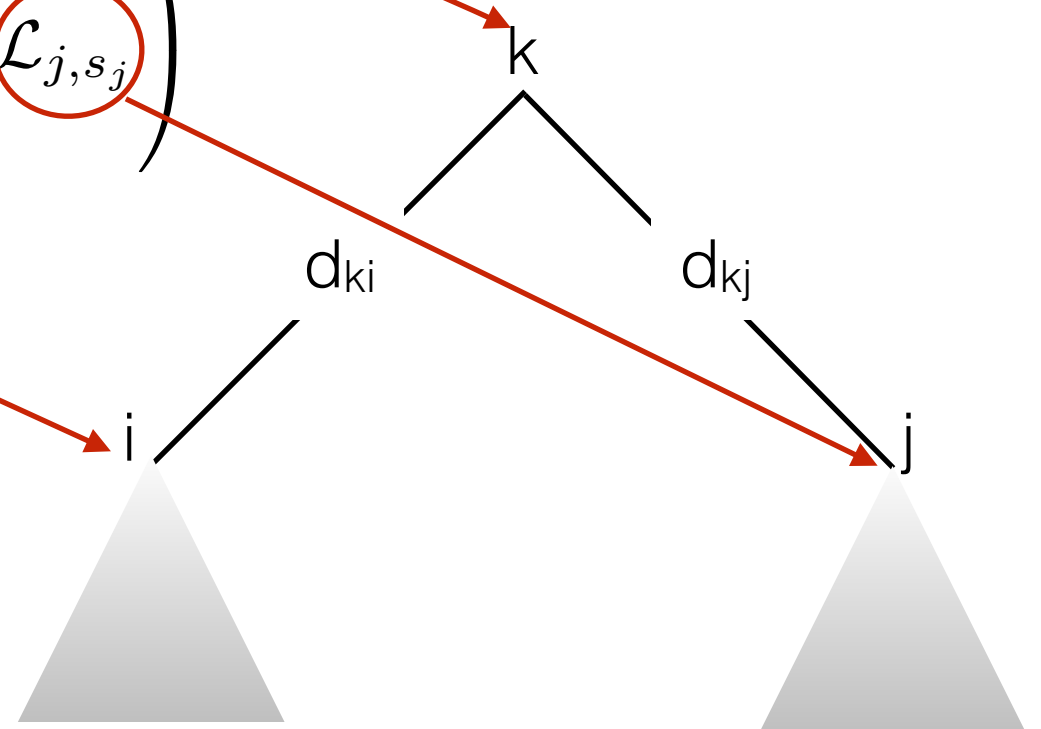
# Small phylogeny problem — Maximum Likelihood

Now, we can define likelihood recursively!

$$\mathcal{L}_{k,s} = \Pr(s_k = s) \quad \text{if } k \text{ is a leaf}$$

$$\mathcal{L}_{k,s} = \left( \sum_{s_i} p_{s_k \rightarrow s_i}(d_{ki}) \mathcal{L}_{i,s_i} \right) \left( \sum_{s_j} p_{s_k \rightarrow s_j}(d_{kj}) \mathcal{L}_{j,s_j} \right)$$

... how can we do this efficiently?



# Small phylogeny problem — Maximum Likelihood

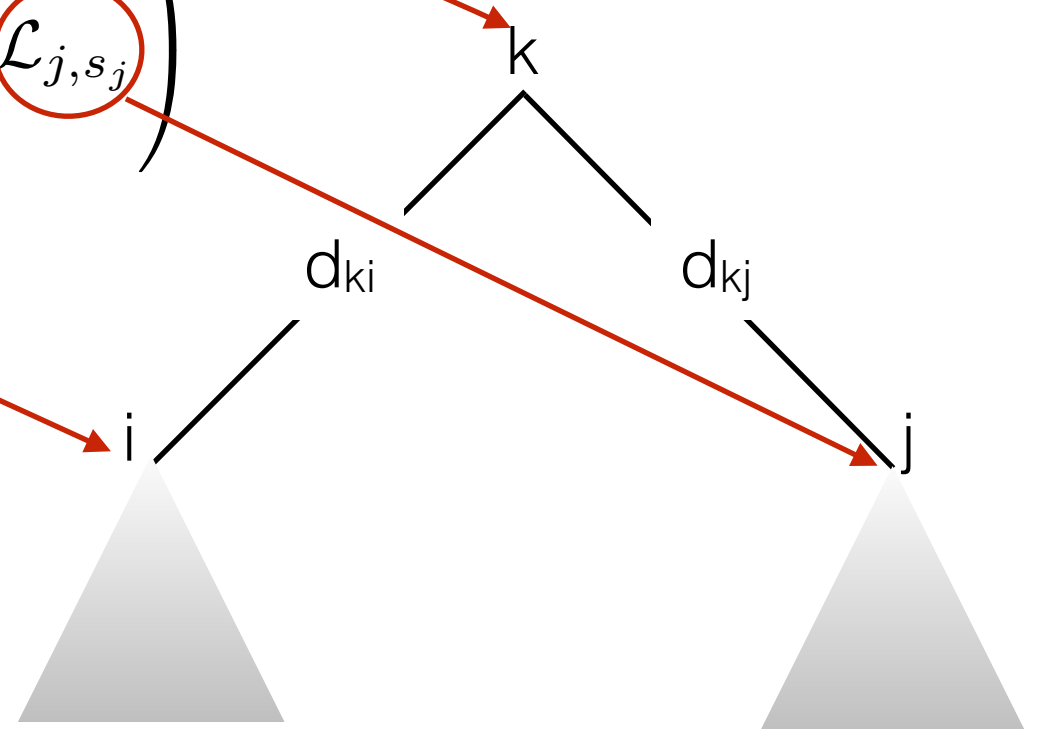
Now, we can define likelihood recursively!

$$\mathcal{L}_{k,s} = \Pr(s_k = s) \quad \text{if } k \text{ is a leaf}$$

$$\mathcal{L}_{k,s} = \left( \sum_{s_i} p_{s_k \rightarrow s_i}(d_{ki}) \mathcal{L}_{i,s_i} \right) \left( \sum_{s_j} p_{s_k \rightarrow s_j}(d_{kj}) \mathcal{L}_{j,s_j} \right)$$

... how can we do this efficiently?

**Dynamic programming:** post-order traversal of the tree!

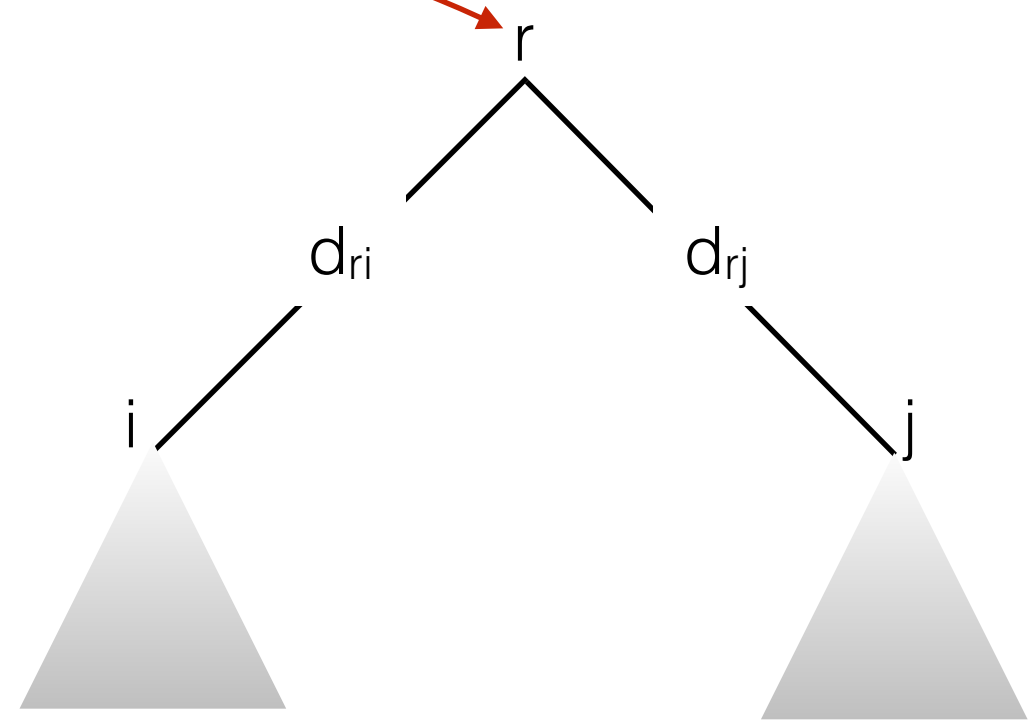


# Small phylogeny problem — Maximum Likelihood

At the root, we simply sum over all possible states to get the likelihood for the entire tree:

$$\mathcal{L} = \sum_{s_r} \pi_{s_r} \mathcal{L}_{r, s_r}$$

Using these likelihoods, we can ask questions like:



What is the probability that node k had state 'A'?

What is the probability that node k didn't have state 'C'?

At node k, how likely was state 'A' compared to state 'C'?



# Small phylogeny problem — Maximum Likelihood

This maximum likelihood framework is very powerful.

It allows us to consider *all* evolutionary histories, weighted by their probabilities.

Also lets us evaluate other tree parameters like branch-length.

**But** we there can be many assumptions baked into our *model* (and such a model is part of our ML framework)

What if our parameters are wrong?

What if our assumptions about “Markovian” mutation are wrong?

What if the *structure* of our model is wrong (correlated evolution)?

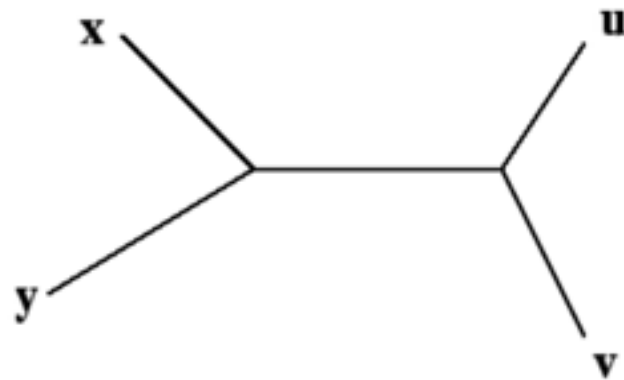
# Large phylogeny problem — searching for trees

- Distance-based methods:
  - \* Sequences -> Distance Matrix -> Tree
  - \* Neighbor joining, UPGMA
- Maximum Likelihood:
  - \* Sequences + Model -> Tree + parameters
- Bayesian MCMC:
  - \* Markov Chain Monte Carlo: random sampling of trees by random walk

# Additivity (for distance-based methods)

- A distance matrix  $M$  is **additive** if a tree can be constructed such that  $d_T(i,j) = \text{path length from } i \text{ to } j = M_{ij}$ .
- Such a tree faithfully represents all the distances
- 4-point condition: A metric space is additive if, given any 4 points, we can label them so that

$$M_{xy} + M_{uv} \leq M_{xu} + M_{yv} = M_{xv} + M_{yu}$$



- If our metric is additive, there is exactly one tree realizing it, and it can be found by successive insertion<sup>#</sup>

# Neighbor Joining

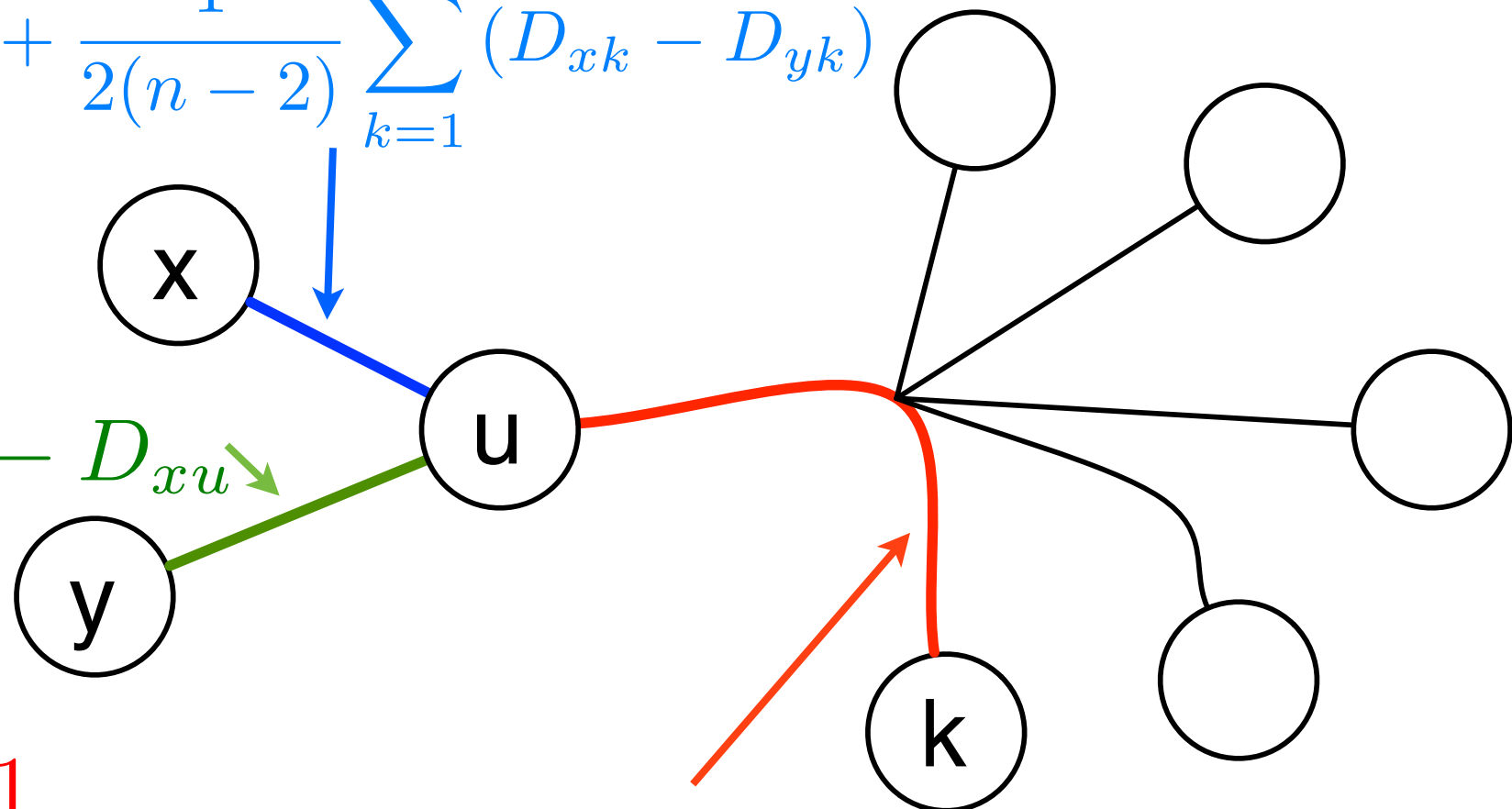
Choose  $x, y$  to merge that minimize:

$$Q(x, y) := (n - 2)D_{xy} - \left( \sum_{k=1}^n D_{xk} + \sum_{k=1}^n D_{yk} \right)$$

Update lengths:

$$D_{xu} := \frac{1}{2}D_{xy} + \frac{1}{2(n-2)} \sum_{k=1}^n (D_{xk} - D_{yk})$$

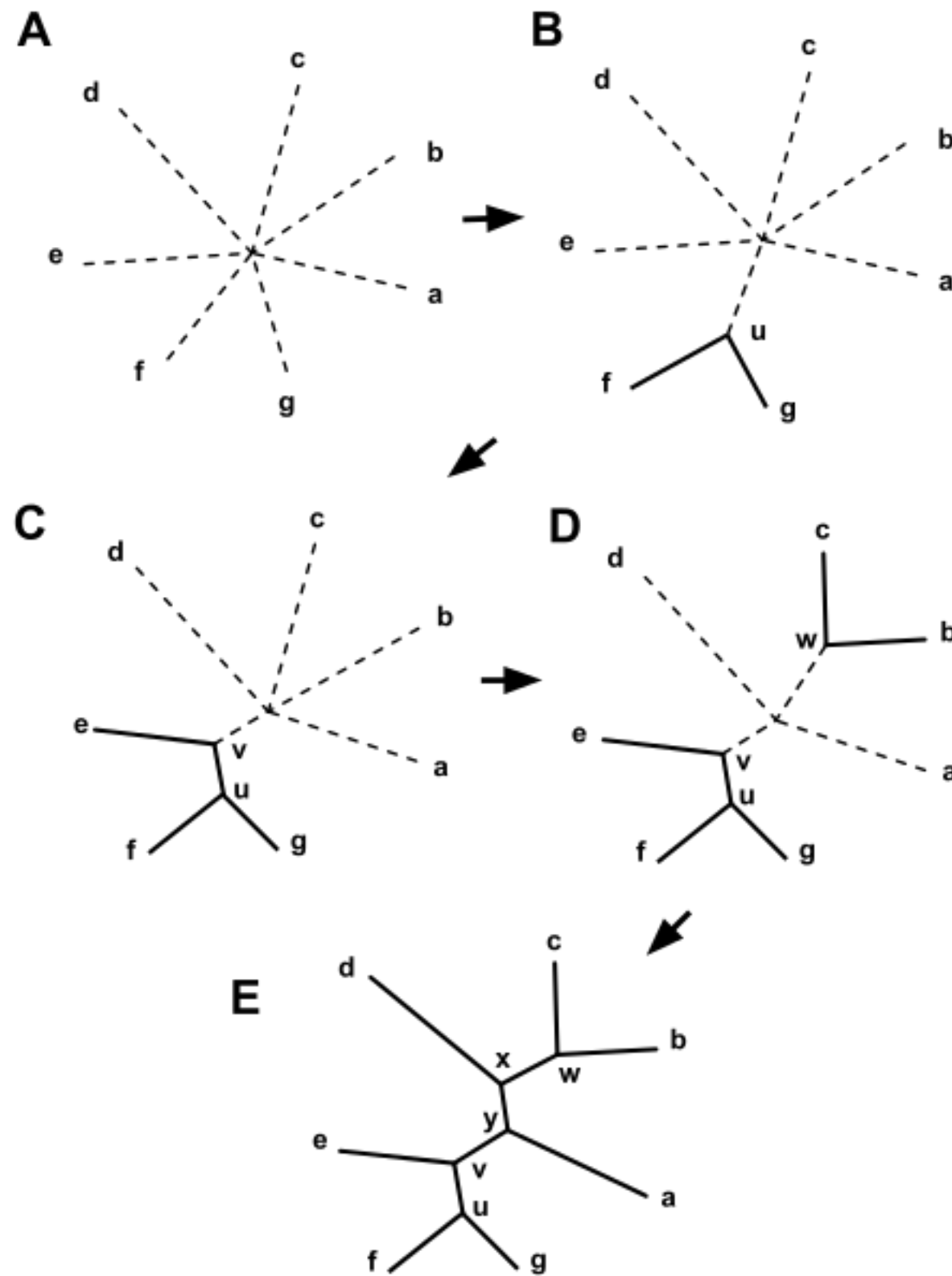
$$D_{yu} := D_{xy} - D_{xu}$$



$$D_{uk} := \frac{1}{2} (D_{xk} + D_{yk} - D_{xy})$$



# Neighbor Joining — Example



What if our distances aren't so nice?

## UPGMA

- Find two most similar taxa (ie. such that  $M_{ij}$  is smallest)
- Merge into new “OTU” (operational taxonomic unit)
  - distance from  $k$  to new OTU = average distance from  $k$  to each of OTUs members
- Repeat.
- **Even if there is perfect tree, it may not find it.**

# Maximum Parsimony

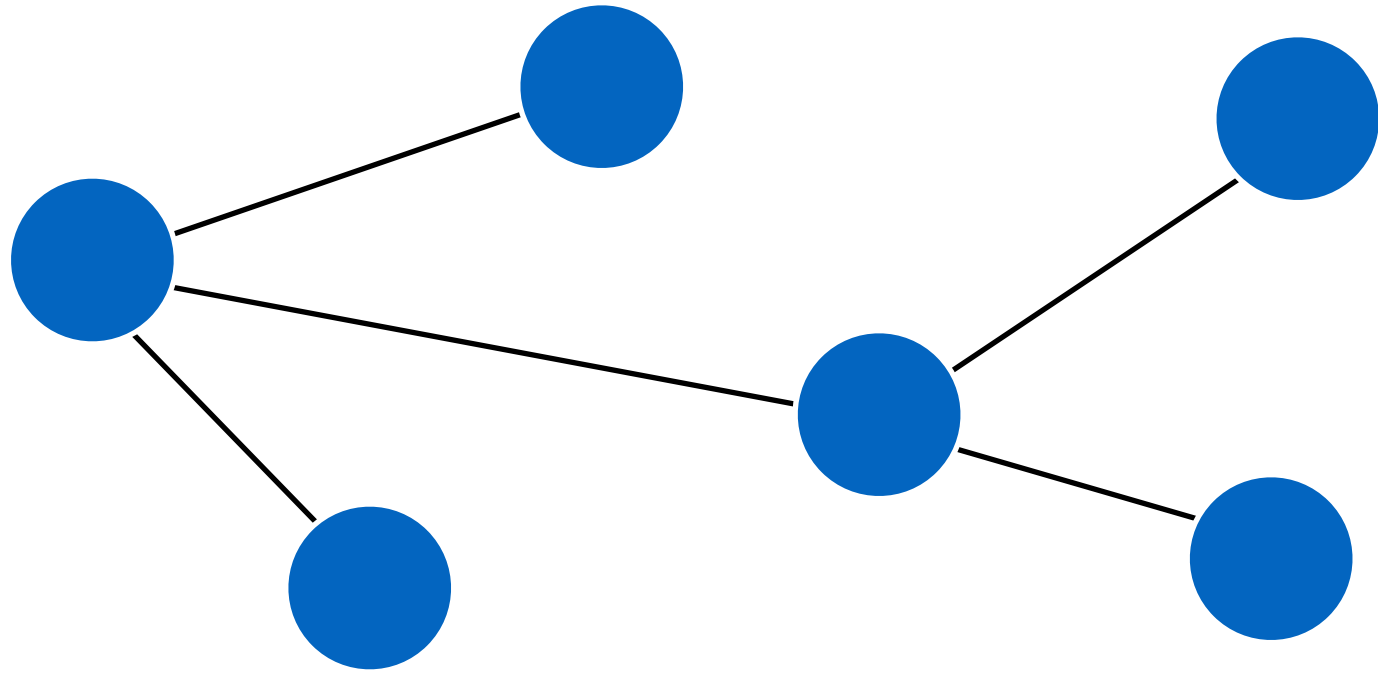
- **Input:**  $n$  sequences of length  $k$
- **Output:** A tree  $T = (V, E)$  and a sequence  $s_u$  of length  $k$  for each node  $u$  to minimize:

$$\sum_{(u,v) \in E} \text{Hamming}(s_u, s_v)$$

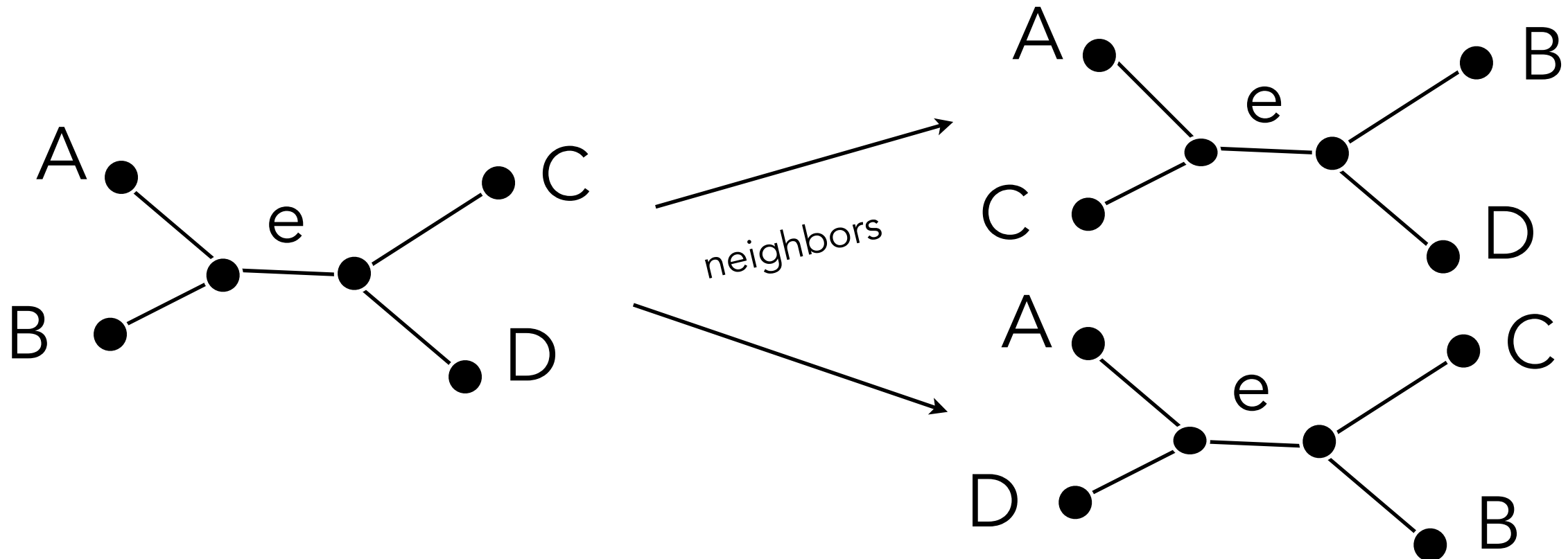
NP-hard (reduction from Hamming distance Steiner tree)

Can score a given tree in time  $O(|\Sigma|nk)$ .

# Heuristic: Nearest Neighbor Interchange



Walk from tree T to its neighbors, choosing best neighbor at each step.



# Maximum Likelihood

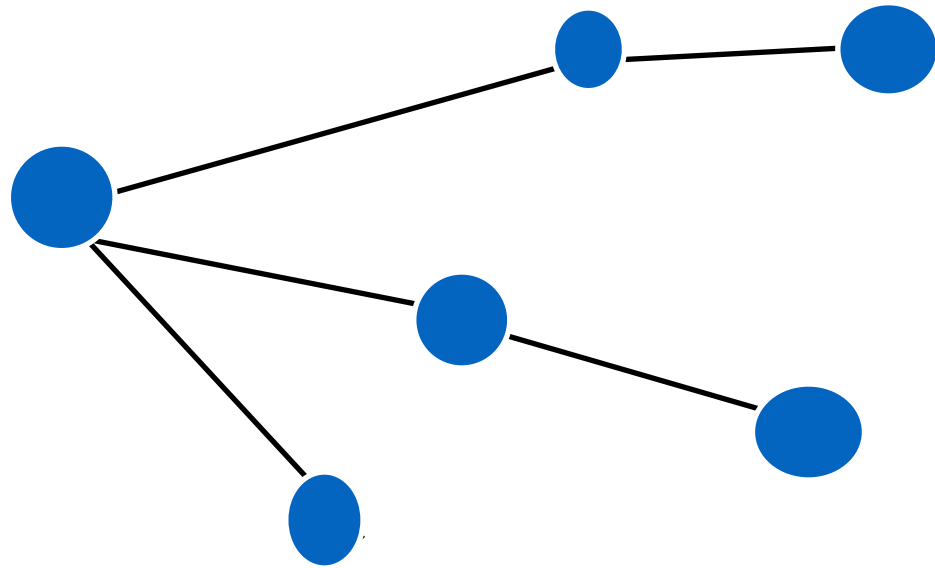
- ⑥ **Input:**  $n$  sequences  $S_1, \dots, S_n$  of length  $k$ ; choice of model
- ⑥ **Output:** Tree  $T$  and parameters  $p_e$  for each edge to **maximize:**

$$\Pr[S_1, \dots, S_n \mid T, p]$$

NP-hard if model is Jukes-Cantor; probably NP-hard for other models.



# Bayesian MCMC



Walk from tree  $T$  to its neighbors, choosing a particular neighbor at each step with probability related to its improvement in likelihood. This walk in the space of trees is a Markov chain.

Under “mild” assumptions, and after taking many samples, trees are visited proportional to their true probabilities.

- ⑥ # of times you visit a tree (after “burn in”) = probability of that topology
- ⑥ Outputs a distribution of trees, not a single tree.

# Bootstrapping

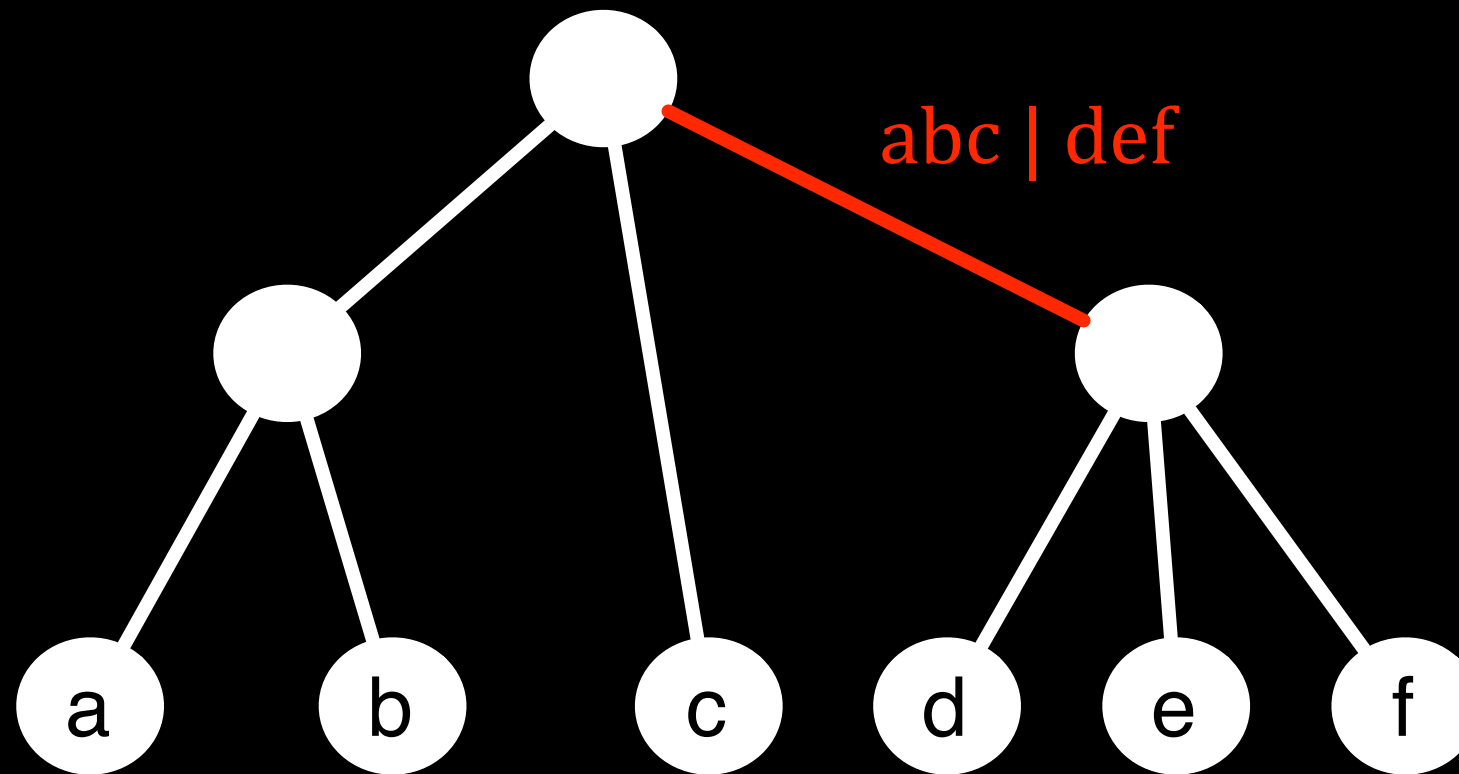
- How confident are we in a given edge?
- Bootstrapping:
  1. Create (e.g.) 1,000 data sets of same size as input by sampling markers (MSA columns) with replacement.
  2. Repeat phylogenetic inference on each set.
  3. Support for edge is the % of trees containing this edge (bipartition).
- **Interpretation:** probability that edge would be inferred on a random data set drawn from the same distribution as the input set.

# Going from an “ensemble” to a single tree

Even if we can generate such an ensemble (e.g. a collection of trees where each is proportional to its true probability).

How can we “extract” a single, meaningful, tree from this ensemble?

# Splits



Every edge  $\Rightarrow$  a **split**, a bipartition of the taxa

- taxa within a clade leading from the edge
- taxa outside the clade leading from the edge

Example: this tree = {abc|def, ab|cdef + 'trivial' splits}

# Consensus

- Multiple trees: from bootstrap, from Bayesian MCMC, trees with sufficient likelihood, same parsimony:

$$T = \{T_1, \dots, T_n\}$$

- Splits of  $T_i := C(T_i) = \{ b(e) : e \in T_i \}$

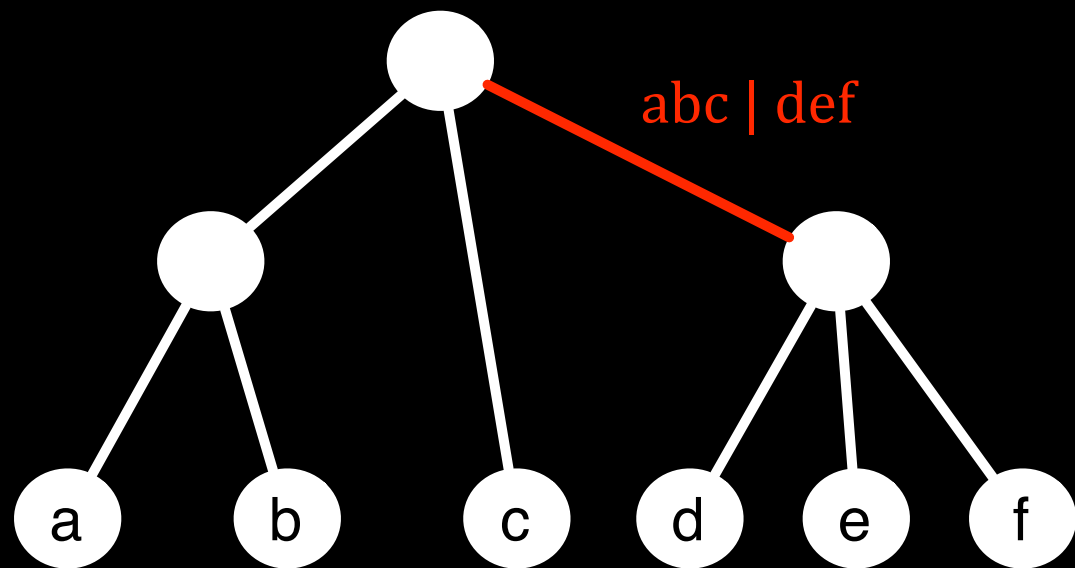
$b(e)$  is the split (bipartition) for edge  $e$ .

- **Majority consensus:** tree given by splits which occur in  $>$  half inferred trees.

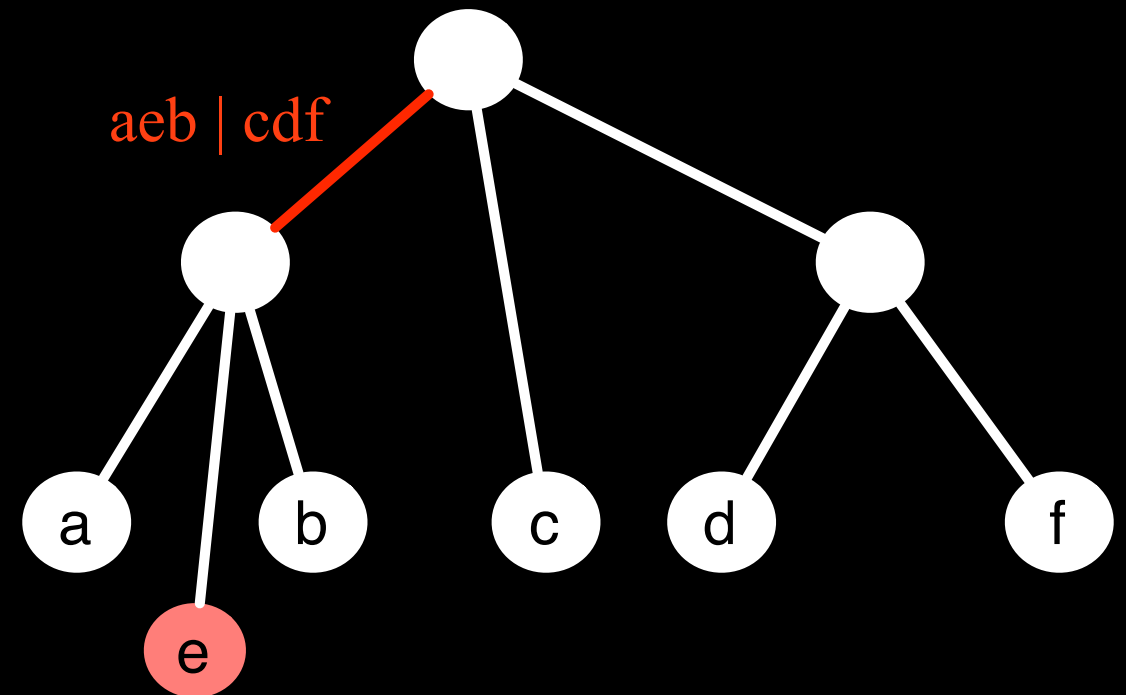


# Incompatibility

Tree 1



Tree 2



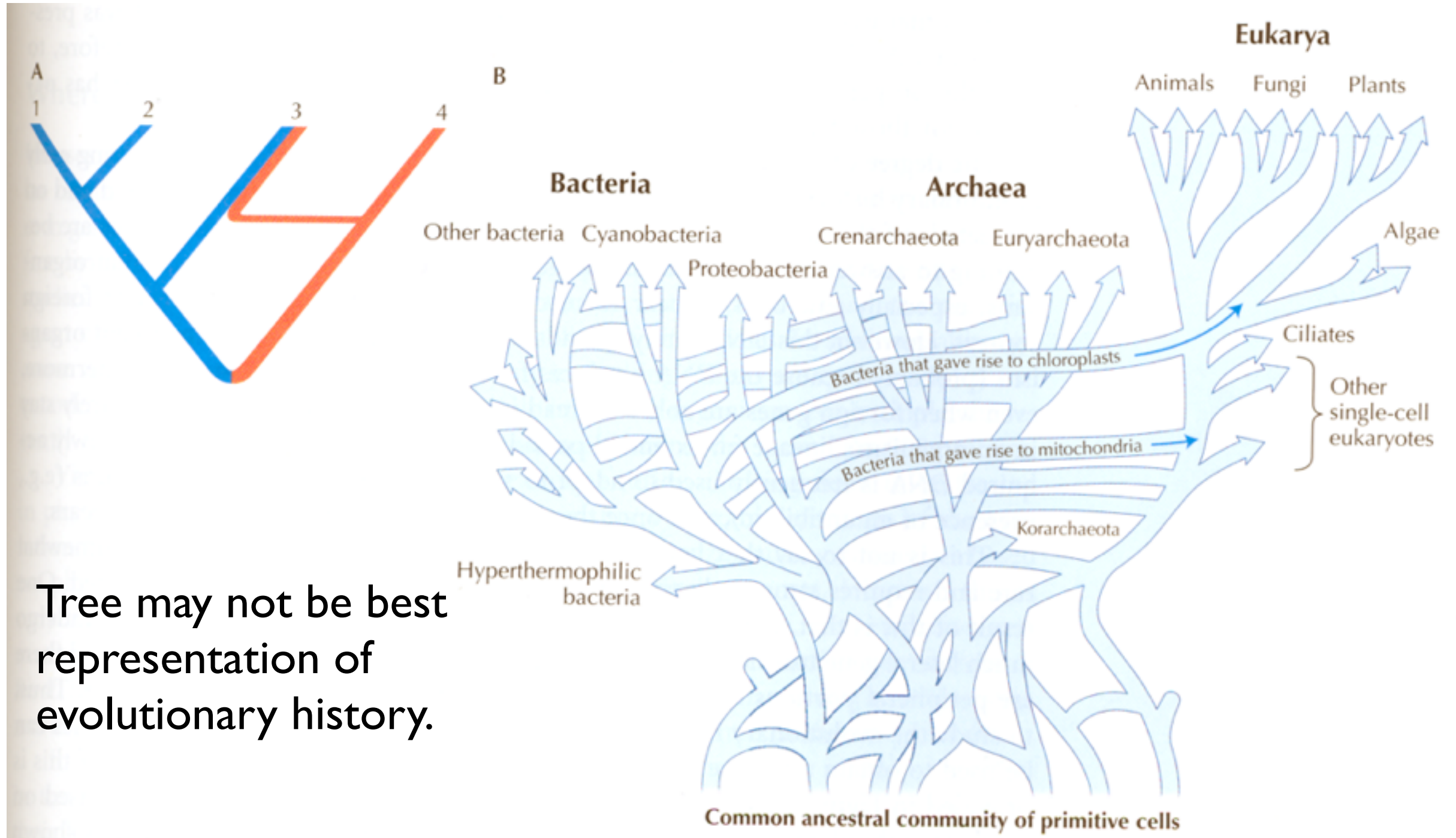
Two splits are **incompatible** if they cannot be in the same tree.

# Majority Consensus Always Exists

- **Proof:**

1. Let  $\{s_k\}$  be the splits in  $>$  half the trees.
  2. Pigeonhole: for each  $s_i, s_j$  in  $\{s_k\}$  there must be a tree containing both  $s_i$  and  $s_j$ .
  3. If  $s_i$  and  $s_j$  are in same tree they are compatible.
  4. Any set of compatible splits forms a tree.
- $\Rightarrow$  The  $\{s_i\}$  are pairwise compatible and form a tree.

# Horizontal Gene Transfer



Tree may not be best representation of evolutionary history.

DNA uptake; retroviruses