Chapter 4

Phylogenetic Trees

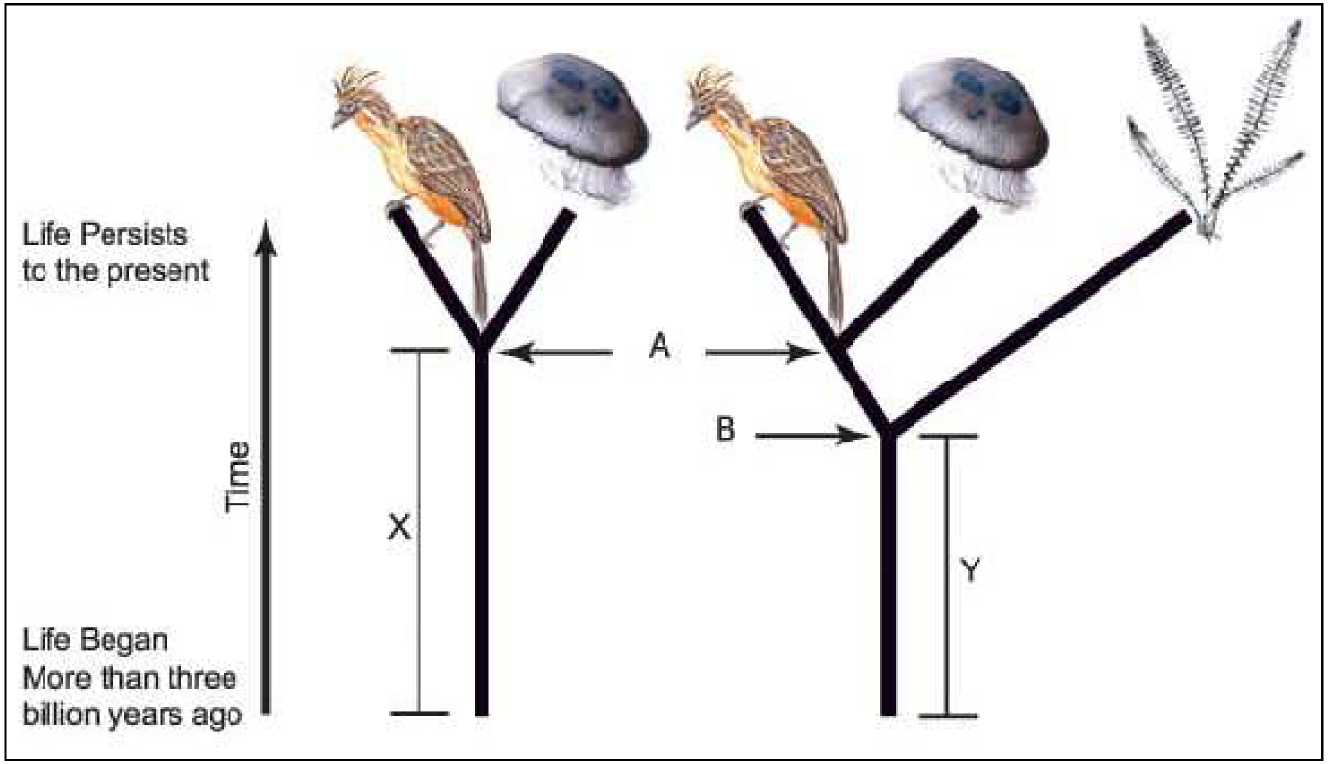


Figure 8.1 in R. Shamir, Orly Stettiner, R. Gabor: Algorithms for Molecular Biology 8.1 Preface: Phylogenetics and Phylogenetic Trees

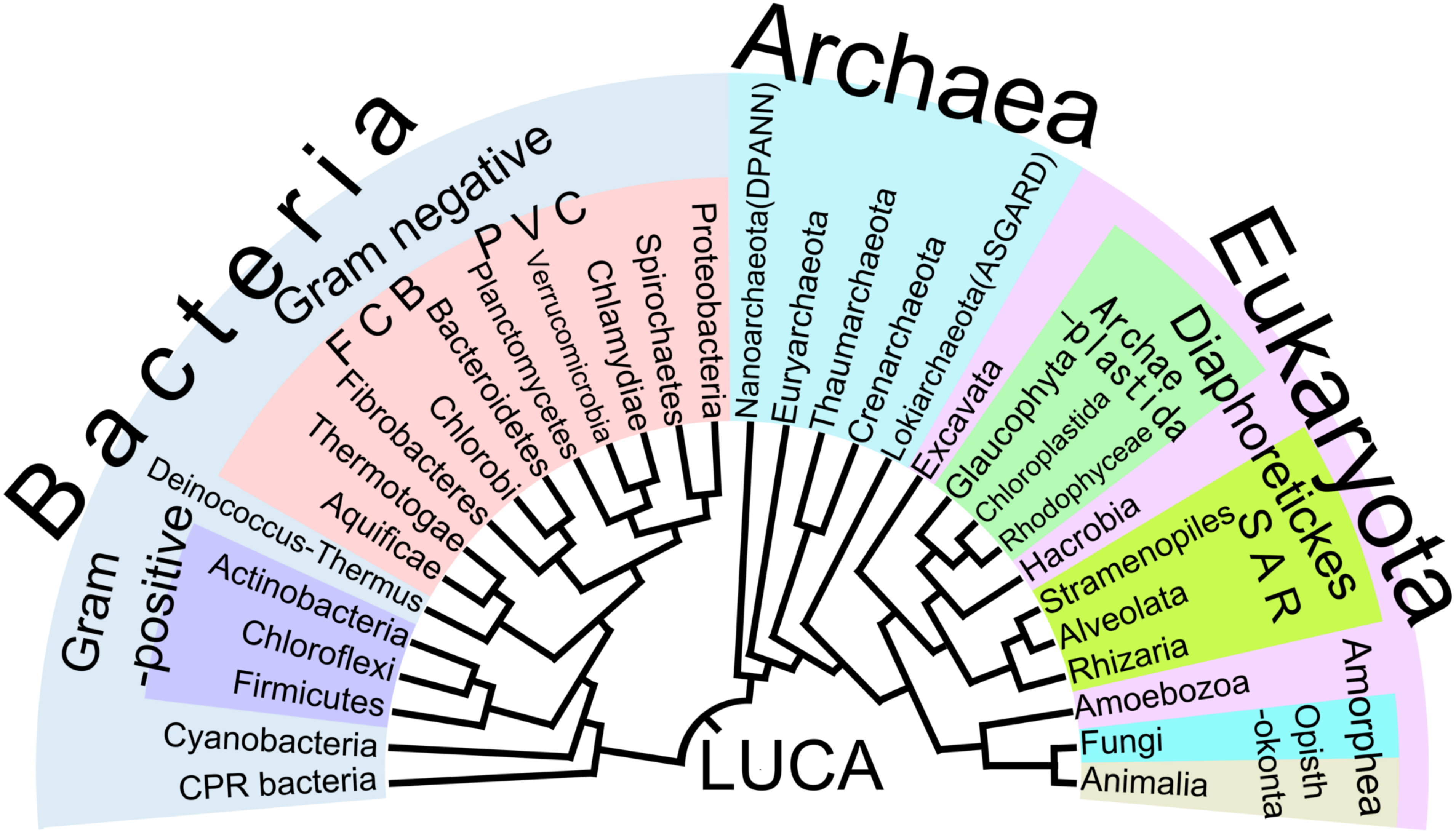
- A: most recent common ancestor of bird and jellyfish
- X: portion of history shared by bird and jellyfish

What is Phylogenetics?

- Find the **evolutionary relationships** between species.
- **Basic idea**: compare specific features of the species.
  Assumption: similar species are genetically close.
- The term **phylogeny** refers to these relationships, usually presented as a **phylogenetic tree.**
- Classic phylogeny: **physical or morphological features** – size, color, shape, number of legs, ...
- **Modern phylogeny** uses information extracted from genetic material – mainly **DNA and protein sequences.**

  Features (characters): DNA or protein sites within **conserved blocks of multiple alignments.**

The Tree of Life



LUCA: Last universal common ancestor: the most recent common ancestor of all current life on Earth.

User:Crion / CC BY (https://creativecommons.org/licenses/by/3.0)

https://commons.wikimedia.org/wiki/File:Phylogenetic_Tree_of_Life.png

Approaches To Phylogenetic Tree Construction

Distance based methods:

require definition of a **distance function between objects**.
Construct tree so that the pairwise distances can be mapped to the tree **as accurately as possible**.

Character based methods:

character or **trait** $=$ a discrete property of an object. E.g.

- “mammal” (all animals are either mammals or not)
- “unicellular” (either unicellular or multicellular)

Species are grouped according to similarity of characters.

Probabilistic methods:

Classification may be based on the **likelihood** of a certain tree explaining the set of objects.
Alternative: **Bayesian approach.** Combine likelihood with prior over trees $\leadsto$ posterior distribution of trees.

Phylogenetic Trees

- Phylogenetic information usually represented as a **tree:**
  - every node represents a species,
  - edges represent the genetic connections.

- Difference between leaf nodes ⇝ **real species**,
  and internal nodes ⇝ **hypothetical evolutionary ancestors**.

- **Phylogenetic trees** take several forms:
  - **rooted** ⇝ one of the nodes is the root
    ⇝ **direction of ancestral relationships is determined**,
  - **unrooted** ⇝ induces no hierarchy,
  - **binary** (or bifurcating) ⇝ a node has only 0 to 2 subnodes,
  - general (not covered here).

A Simple Solution?

Trivial solution: enumerate over all possible trees and calculate the target function for each one.

Problem: number of non-isomorphic, labeled, binary, rooted trees containing n leaves, is super-exponential:

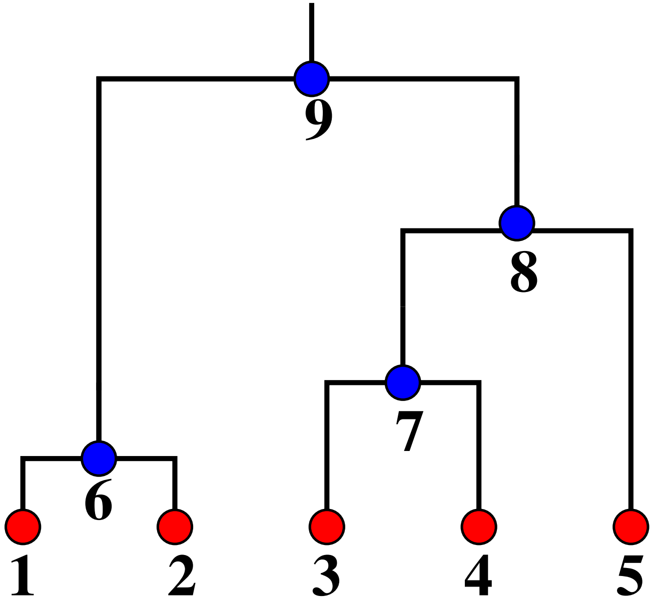
$$(2n-3)!! = \prod_{i=2}^{n}(2i-3) = 1 \cdot 3 \cdot 5 \cdot 7 \cdots (2n-3)$$

(or $(2n-5)!!$ for unrooted trees). For $n = 20$: about 10^{21} trees → infeasible even for a relatively small number of species.

Theorem: Phylogenetic Tree Construction (for almost all reasonable models) is NP-Complete.

Number of nodes and edges

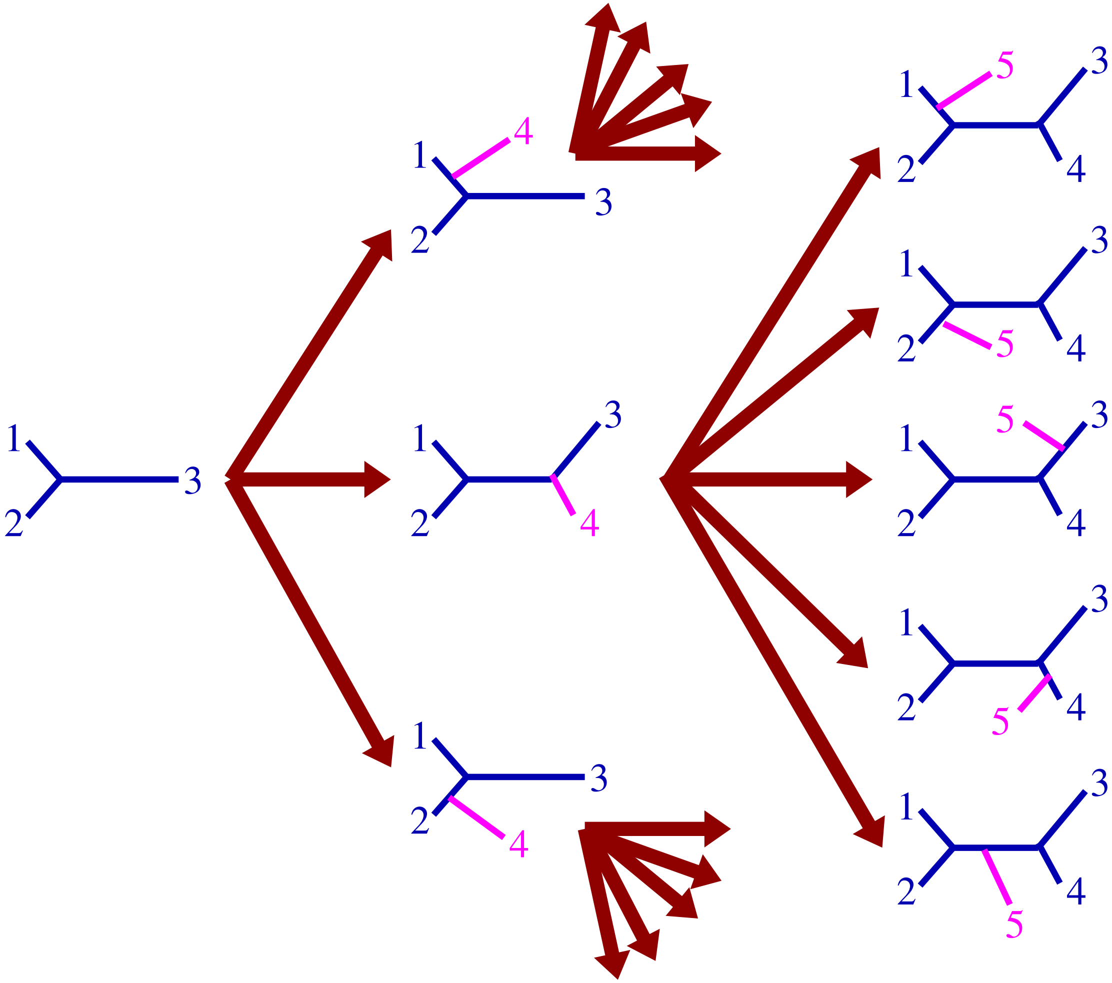
- Suppose there are n leaves in a **rooted tree**. As we move up the tree, two edges join as a new node is reached.
  Each time, the number of edges is reduced by one.
- So there must be $(n-1)$ inner nodes $\rightsquigarrow$ $(2n-1)$ nodes and $(2n-2)$ edges (not counting the edge above the root)



- **Unrooted tree:** $(2n-2)$ nodes and $(2n-3)$ edges.

Consider an **unrooted tree with n leaf nodes:**

- An **extra edge** with new label at its leaf can be added at any edge
  $\rightsquigarrow$ there are $(2n-3)$ times as many trees with $n+1$ leaves.
- Instead of an extra edge, we can **add a root**
  $\rightsquigarrow$ there are $(2n-3)$ times as many rooted trees as unrooted ones.
- There are $1 \cdot 3 \cdot 5 \cdot \cdots \cdot (2n-5) = (2n-5)!!$ unrooted trees with n leaves
  $\rightsquigarrow$ $(2n-3)!!$ rooted trees.



Distance-Based Methods

- Assume we have a **measure of distance** between each pair of species.

- **Approach:** find a tree that predicts the observed set of distances as closely as possible.

- This **leaves out some of the information** contained in the raw sequence (due to reduction to table of pairwise distances).

- It seems that in many cases **most of the evolutionary information** is conveyed in these distances.

Least Squares Methods

- **Idea:** approximate an observed distance matrix.
- **Goal:** find a tree T, whose leaves are the n given species, and that predicts distances d_{ij}^T between the species, so that the following expression is minimized:

$$SSQ(T) \equiv \sum_{i=1}^{n} \sum_{j \neq i} (d_{ij} - d_{ij}^T)^2$$

where d_{ij} is the observed distance between species i and j.

- SSQ is a measure of the discrepancy between the **observed distances d_{ij}** and the **path distances d_{ij}^T in the tree T.**

The Least Squares Tree Problem

Problem: Least Squares Tree.
INPUT: The distance d_{ij} between species i and j, for each $1 \leq i, j \leq n$, arranged in distance matrix D.
QUESTION: Find the phylogenetic tree T, with the species as its leaves, that minimizes $SSQ(T)$.

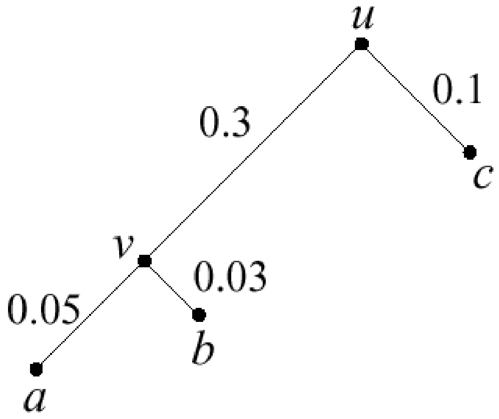
- Difficult problem, due to optimization over discrete set of topologies. One can show: NP-complete problem.
- Two polynomial heuristics - *UPGMA* and *Neighbor-Joining*. These are efficient algorithms, but they will only work in some particular cases.

Ultrametric trees: All the leaves have the same distance to the root $\rightsquigarrow$ all species evolve at the same rate.

In such cases, UPGMA will find the correct topology.

Efficiently solvable Special Cases

Additive distance matrices: There exists a tree that represents **exactly** the given distances between species:
$d_{ij} = \sum$ of all edge lengths in the path between leaves i and j.



	a	b	c
a	0	0.08	0.45
b	0.08	0	0.43
c	0.45	0.43	0

In such cases, neighbor joining will find the correct topology.

In general, given a set of pairwise distances ($\leadsto$ scales **quadratically** in n) it is not possible to find a set of internal edges ($\leadsto$ number is **linear** in n) that explain all the observed distances as path distances in the tree.

UPGMA

- **UPGMA**, or *Unweighted Pair Group Method with Arithmetic mean*, is a heuristic algorithm that often generates satisfactory results.

- The algorithm iteratively **joins the two nearest clusters** (or groups of species), until one cluster is left.

Initialization:

1. Initialize n clusters C_i, one species per cluster.

2. Set the size of each cluster to 1: $n_i \leftarrow 1$.

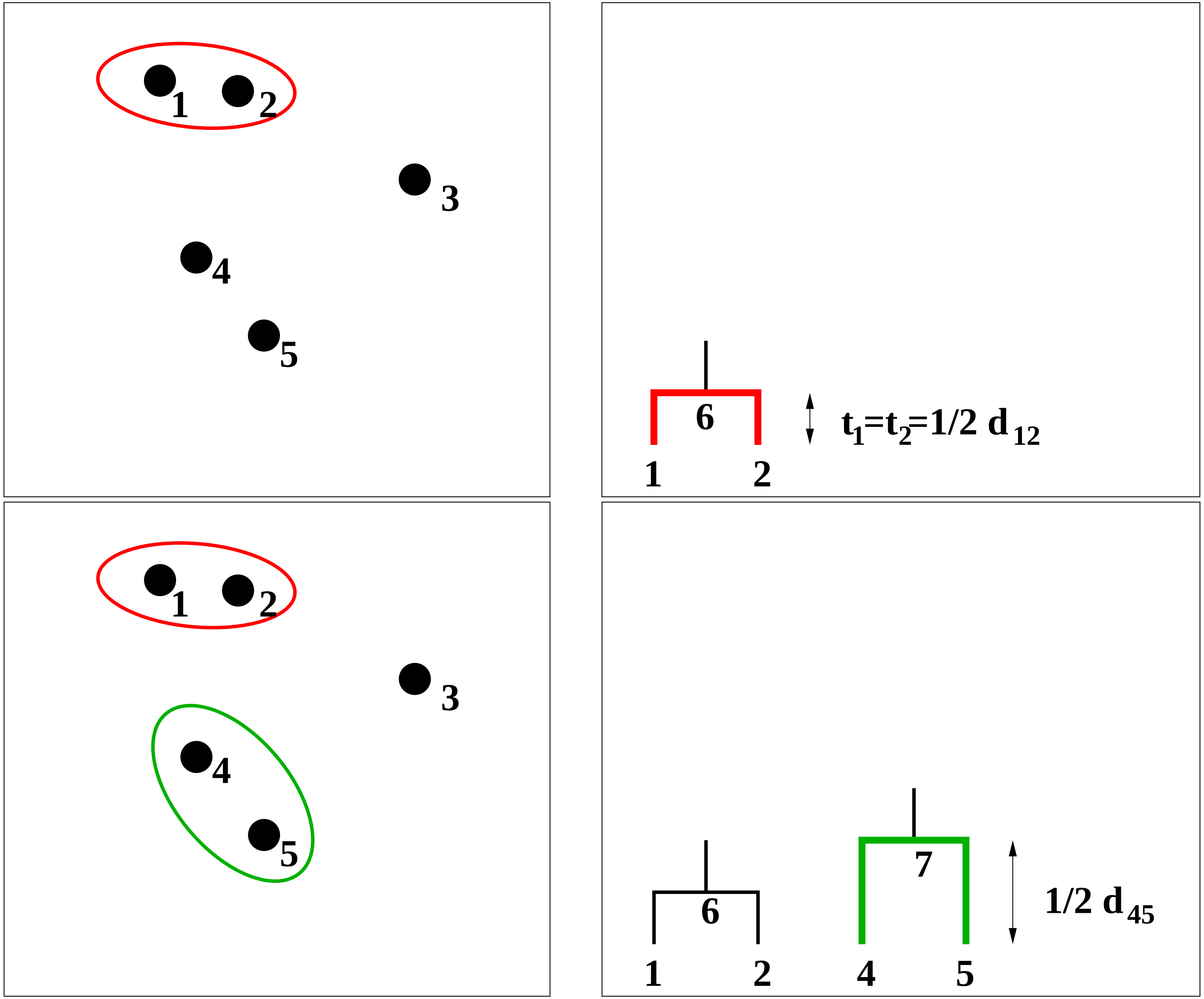
3. In the output tree T, assign a leaf for each species.

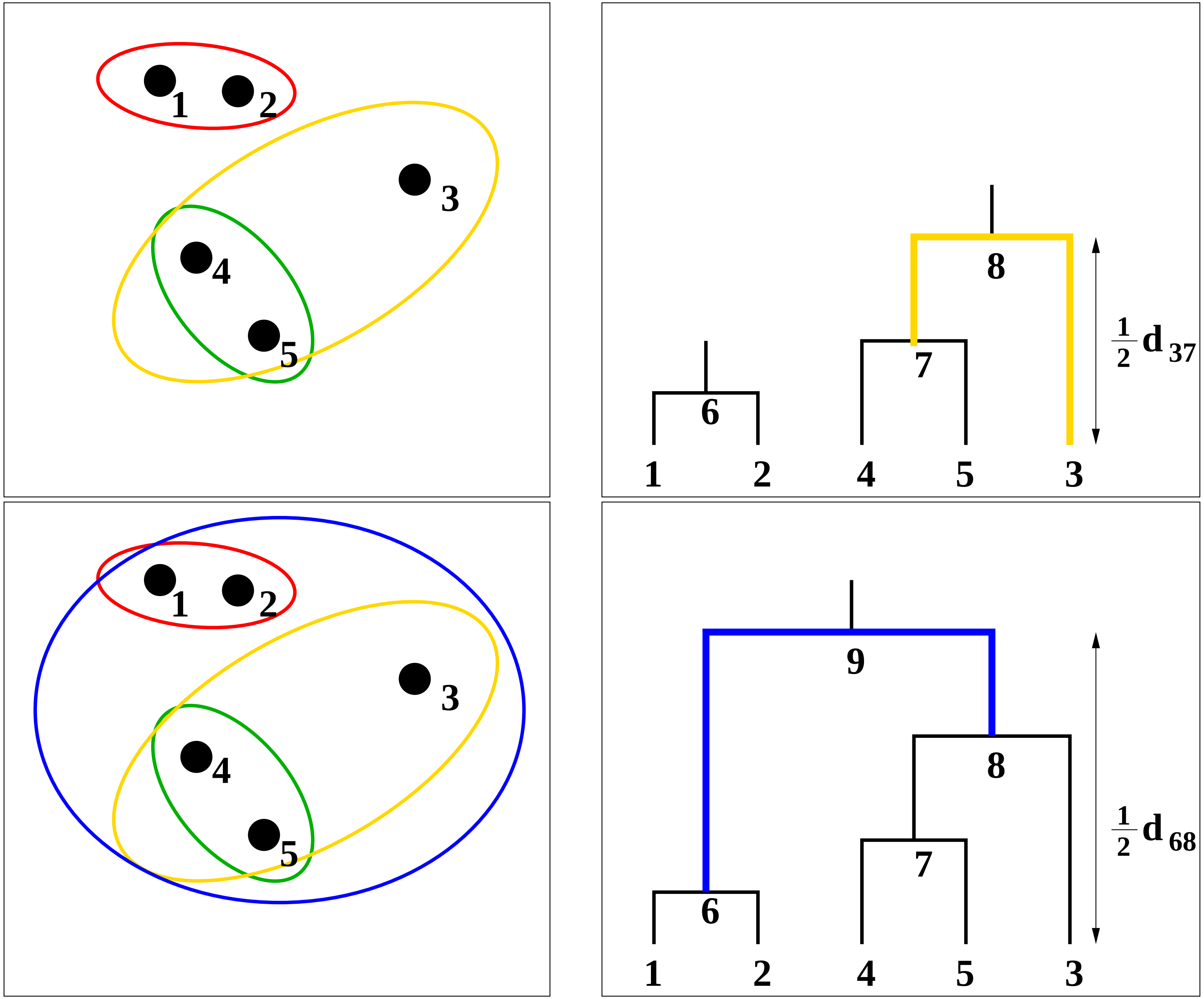
Iteration:

1. Find the i and j that have the smallest distance d_{ij}.
2. Create a new cluster - k by $C_k = C_i \cup C_j$ and compute the distance from the new cluster to all other clusters as a weighted average of the distances from its components:

$$d_{kl} = (\frac{n_i}{n_i + n_j})d_{il} + (\frac{n_j}{n_i + n_j})d_{jl}.$$

3. Connect i and j on the tree to the new node k, and place it at height $d_{ij}/2$. Note: vertical axis represents time. Horizontal connections do not contribute to path-length computations.
4. Delete the columns and rows in D that correspond to clusters i and j, and add a column and row for cluster k.
5. Return to 1 until there is only one cluster left.


1
2
3
4
5
6
1
2
t_1=t_2=1/2 d_{12}
1
2
3
4
5
6
7
1
2
4
5
1/2 d_{45}


1
2
3
4
5
8
7
6
1 2 4 5 3
$\frac{1}{2}\,\mathbf{d}_{37}$
1
2
3
4
5
9
8
7
6
1 2 4 5 3
$\frac{1}{2}\,\mathbf{d}_{68}$

UPGMA: Analysis

- A metric on a set of objects O is given by the assignment of a real number $d(x,y)$ to every pair $x, y \in O$ where $d(x,y)$ has to fulfill the following requirements:

$$\begin{aligned} d(x,y) &> 0 \qquad for\ x \neq y\,, \qquad d(x,y) = 0 \qquad for\ x = y \\ d(x,y) &= d(y,x) \\ d(x,y) &\leq d(x,z) + d(z,y) \qquad \forall\, x, y, z \quad \text{(triangle inequality)} \end{aligned}$$

- An ultrametric has to fulfill a restricted triangle inequality

$$d(x,y) \ \leq \max\left(d(x,z),\ d(y,z)\right).$$

- A **clocklike**, or **ultrametric** tree is a rooted tree, in which the total branch length from the root to any leaf is equal
$\rightarrow$ **molecular clock** that ticks in a constant pace, and all the observed species are at an equal number of ticks from the root.

UPGMA: Analysis

- One can show: If the input data are ultrametric then UPGMA is **guaranteed to return the optimal solution**.
- For substantially **non-clocklike** trees, the algorithm might give seriously **misleading results**.

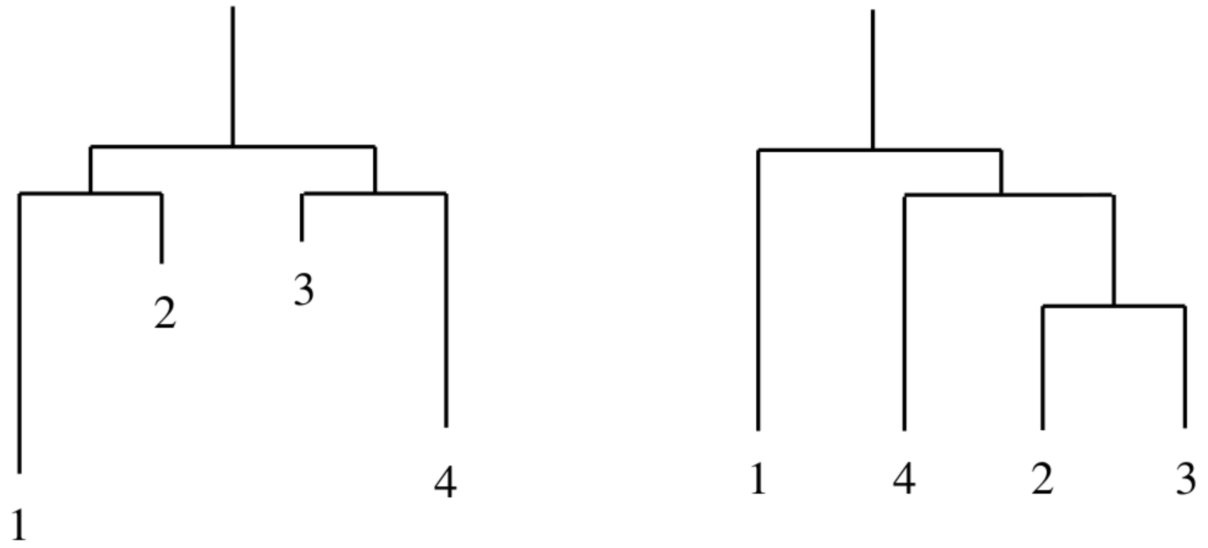


Figure 7.5 *A tree (left) that is reconstructed incorrectly by UPGMA (right).*

Durbin et al., Cambridge University Press. https://doi.org/10.1017/CBO9780511790492.004

Additive trees

- **Ultrametric tree**:
  - $\#$(mutations) $\propto$ temporal distance(node,ancestor),
  - mutations took place with the same rate in all paths.

- But it's a fact, that the evolutionary clock is running **differently** for different species (and even for different regions in a sequence).

- Generalization: **additive trees** (i.e. trees built form additive distance matrices). Unrooted tree, reflection of our ignorance as to where the common ancestor lies.

- All nodes (except for the leaves) have degree three $\rightsquigarrow$ **unrooted binary tree**. More general, but undirected.

Additive distance matrix

Distance matrix D is additive iff there exists a tree T with $d_{ij}^T = d_{ij}$
$\rightsquigarrow SSQ(T) = \sum_{i=1}^n \sum_{j \neq i} (d_{ij} - d_{ij}^T)^2 = 0$.

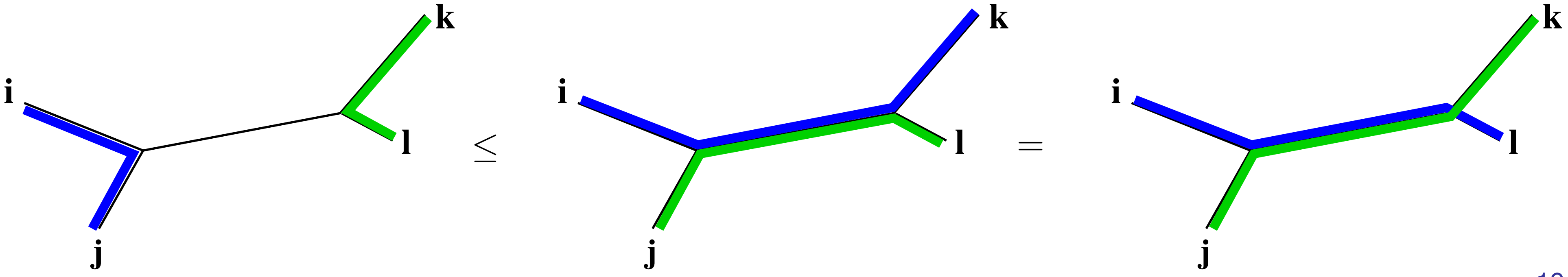
Every ultrametric is additive, but the converse is not true.

Simple test for additivity?

Four point condition: For every set of four leaves i, j, k and l, two of the distances $d_{ij} + d_{kl}, d_{ik} + d_{jl}$ and $d_{il} + d_{jk}$ must be equal and larger than the third. For instance

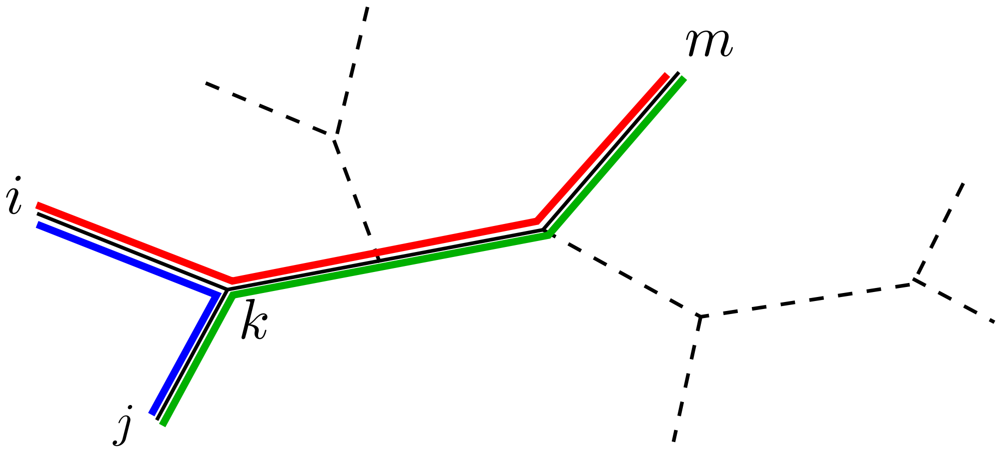
$$d(i,j) \ + \ d(k,l) \ \leq \ d(i,k) \ + \ d(j,l) \ = \ d(i,l) \ + \ d(j,k).$$

Generalizes the triangle inequality (take $k = l$).



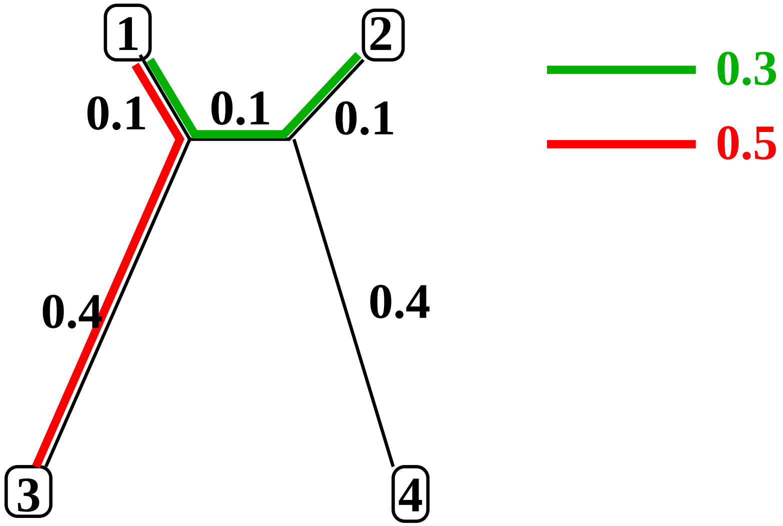
Neighbor Joining

- Neighbor-Joining approximates the least squares tree, assuming **additivity**, but **without resorting to the assumption of a molecular clock**.
- **Idea:** Find direct ancestor of two species, join them, iterate.
- **Distance computation:** Asume we join i and j with ancestor k → remove i, j from list of leaves → add k to list with distances to other leaves m defined as $d_{km} = \frac{1}{2}(d_{im} + d_{jm} - d_{ij})$.



$$\left.\begin{array}{l} d_{jm} = d_{jk} + d_{km} \\ d_{im} = d_{ik} + d_{km} \\ d_{ij} = d_{ik} + d_{kj} \end{array}\right] \Longrightarrow \boxed{d_{km} = \frac{1}{2}(d_{im} + d_{jm} - d_{ij})}$$

Correcting distances



Problem: it is not sufficient to pick simply the two closest leaves.

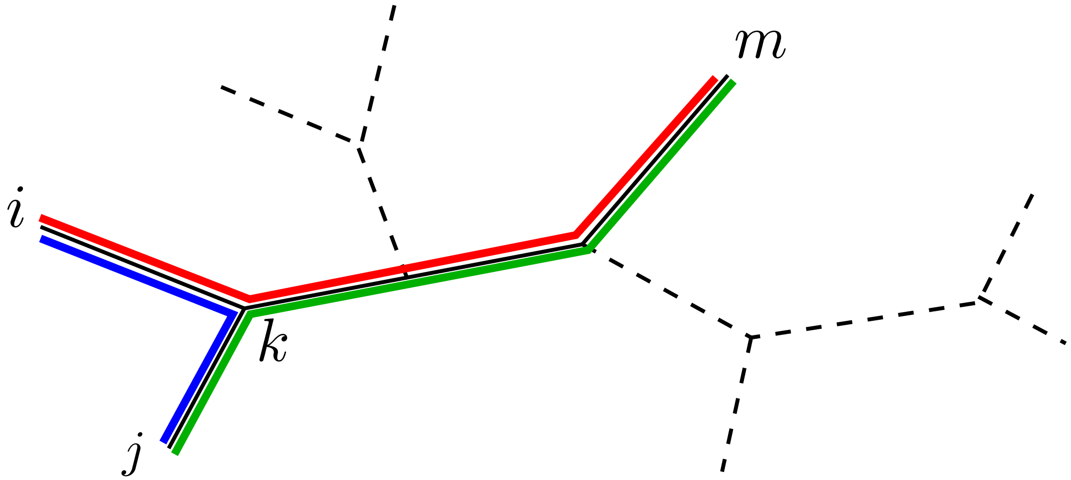
Solution: Join clusters that are not only close, but are also far from the rest. For node i, define average distance u_i to all other leaves: $u_i = \sum_{k \neq i} \frac{d_{ik}}{(n-2)}$, and “correct” distances: $q_{ij} = d_{ij} - (u_i + u_j)$.

$$D = \begin{bmatrix} 0 & 0.3 & 0.5 & 0.6 \\ & 0 & 0.6 & 0.5 \\ & & 0 & 0.9 \\ & & & 0 \end{bmatrix}, \quad Q = \begin{bmatrix} * & -1.1 & -\mathbf{1.2} & -1.1 \\ & * & -1.1 & -\mathbf{1.2} \\ & & * & -1.1 \\ & & & * \end{bmatrix}$$

Neighbor Joining Theorem

(Studier & Keppler, Molecular Biology and Evolution 5:729-731, 1988): For a tree with additive lengths, q_{ij} minimal implies i, j are neighboring leaves.

We know how to compute the branch lengths from a new node k to all other nodes $m \neq (i, j)$.



$$\left.\begin{array}{r} d_{jm} = d_{jk} + d_{km} \\ d_{im} = d_{ik} + d_{km} \\ d_{ij} = d_{ik} + d_{kj} \end{array}\right] \Longrightarrow \boxed{d_{km} = \tfrac{1}{2}(d_{im} + d_{jm} - d_{ij})}$$

Neighbor-Joining: Distance Computation

What about i and j? $d_{ik} = \frac{1}{2}(d_{ij} + d_{im} - d_{jm}), \forall m \neq (i,j)$.
If observed distances are indeed fully additive, we can pick any $m \neq (i,j)$. In practice, it might be better to average:

$$
\begin{aligned}
d_{ik} &= \frac{1}{2}(d_{ij} + d_{im} - d_{jm}), \forall m \neq (i,j) \quad \Rightarrow \text{average over m} \Rightarrow \\
&= \frac{1}{2} \cdot \frac{1}{n-2} \sum_{m \neq (i,j)} (d_{ij} + d_{im} - d_{jm}) \\
&= \frac{1}{2} d_{ij} + \frac{1}{2} \cdot \frac{1}{n-2} \sum_{m \neq (i,j)} (\overbrace{q_{im} + u_i + u_m}^{d_{im}} - q_{jm} - u_j - u_m) \\
&= \frac{1}{2}(d_{ij} + u_i - u_j) + \frac{1}{2} \cdot \frac{1}{n-2} \cdot \underbrace{\sum_{m \neq (i,j)} (q_{im} - q_{jm})}_{=0}
\end{aligned}
$$

Neighbor-Joining algorithm: Initialization:

1. Initialize n clusters with the given species, one species per cluster.
2. Set the size of each cluster to 1: $n_i \leftarrow 1$.
3. In the output tree T, assign a leaf for each species.

Iteration:

1. For each species, compute $u_i = \sum_{k \neq i} \frac{d_{ik}}{(n-2)}$
2. Choose the i and j for which $d_{ij} - u_i - u_j$ is smallest.
3. Join clusters i and j to new cluster, with corresponding node k and set

$$d_{km} = \frac{1}{2}(d_{im} + d_{jm} - d_{ij}) \quad \forall m \neq (i,j) \in \text{Nodes}(T).$$

Calculate the branch lengths from i and j to the new node as:

$$d_{ik} = \frac{1}{2}(d_{ij} + u_i - u_j) \ , \quad d_{jk} = \frac{1}{2}(d_{ij} + u_j - u_i).$$

4. Delete clusters i and j from T and add k.

5. If more than two nodes remain, go back to 1. Otherwise, connect the two remaining nodes by a branch of length d_{ij}.

Reconstructing Trees from Non-additive Matrices

- Q: What if the distance matrix is **not** additive?
- A: We could still run NJ!
- Q: But can **anything** be said about the resulting tree?
- A: Not really. Resulting tree topology could even vary according to way **ties are resolved** on the way.

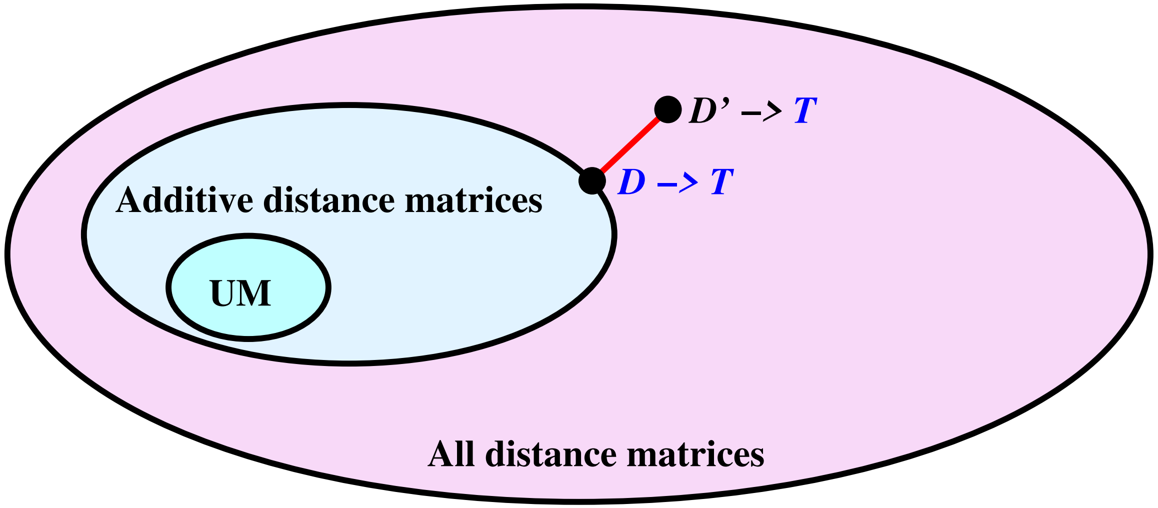
Almost Additive Distance Matrices

A distance matrix D' is called **"almost additive"** if there exists an additive matrix D (with corresponding tree T) such that

$$|D - D'|_\infty = \max_{i,j}\{|d_{i,j} - d'_{i,j}|\} \leq \min_{e}\{l(e)/4\},$$

where e is an edge in the tree T (corresponding to the additive matrix D) with length $l(e)$.

Theorem: If D' is almost additive with respect to a tree T, then the output of NJ is a tree T' with the **same topology** as T.



Character Based Methods

Problem: Optimal Phylogenetic Tree. **INPUT:**

- A set of n **species**.
- A set of m **characters** pertaining to all of these species,
- For each species, the **values** of each of the characters.
- **Notation:** $n \times m$ matrix M, where $M_{i,j}$ represents the value of the j-th character of the i-th species. The **value** of each character is taken from a known alphabet Σ.

Question: What is the fully labeled phylogenetic tree that best explains the data, i.e., maximizes some target function.

Limiting assumptions: (probably not exactly correct in practice)

- Characters are **mutually independent** ($\rightarrow$ change in one character has no effect on the distribution of another character).
- After two species diverged in the tree, they **continue to evolve independently**.

Character-based Methods: Parsimony

- Intuitive **score** for tree: **number of changes** along edges.
- Minimizing this score is called **parsimony**.

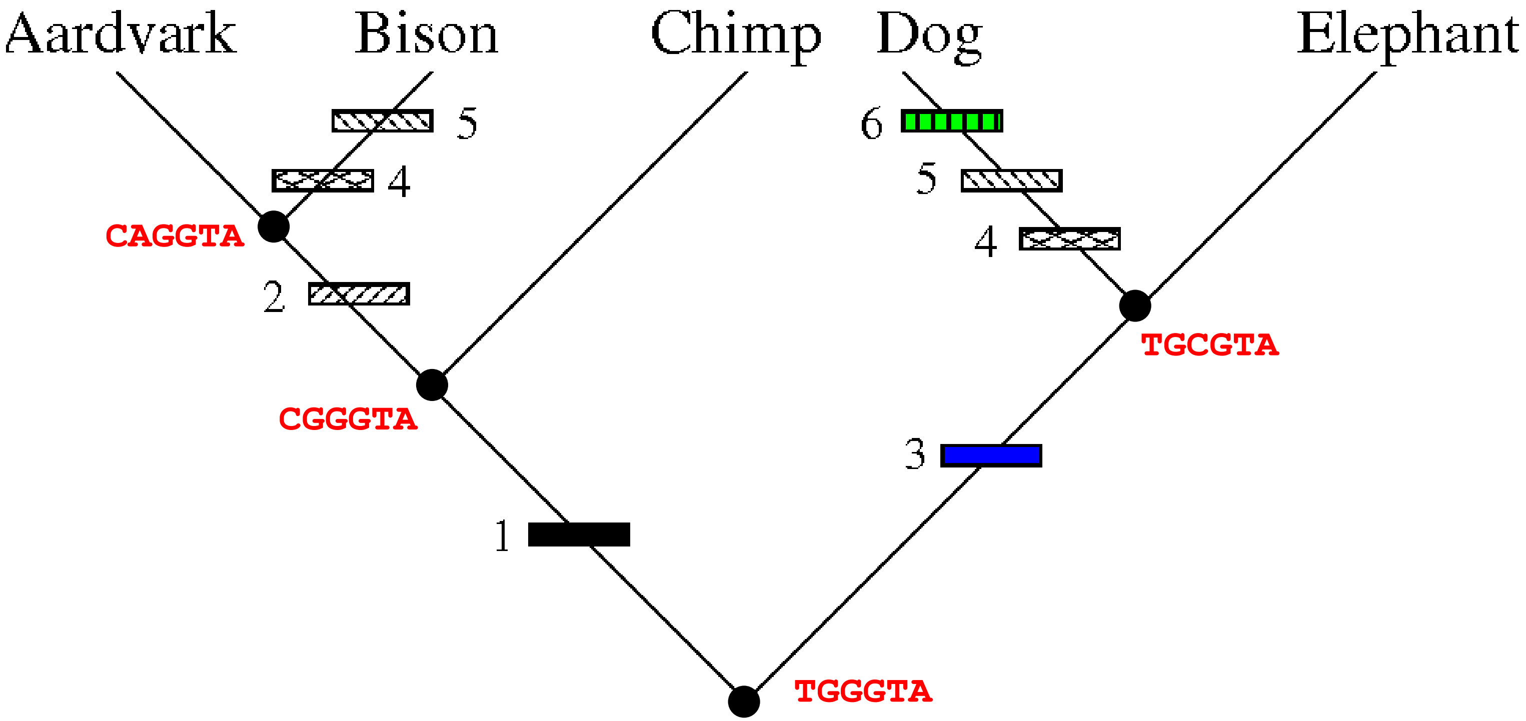
Notation: $V(T)$: **vertices** of a tree, $E(T)$: **edges**.
v_j: **value** of j-th character at vertex $v \in V(T)$.

Given a phylogenetic tree T, its **parsimony score** is defined as

$$S(T) = \sum_{(v,u) \in E(T)} |\{j : v_j \neq u_j\}|$$

That is - the total number of times the **value of some character changes** along some edge.

Most parsimonious 5-species phylogeny for 6 characters:



Adapted from Figure 8.8 in R. Shamir, Orly Stettiner, R. Gabor: Algorithms for Molecular Biology 8.1 Preface: Phylogenetics and Phylogenetic Trees

	1	2	3	4	5	6
Aardvark	C	A	G	G	T	A
Bison	C	A	G	A	C	A
Chimp	C	G	G	G	T	A
Dog	T	G	C	A	C	T
Elephant	T	G	C	G	T	A

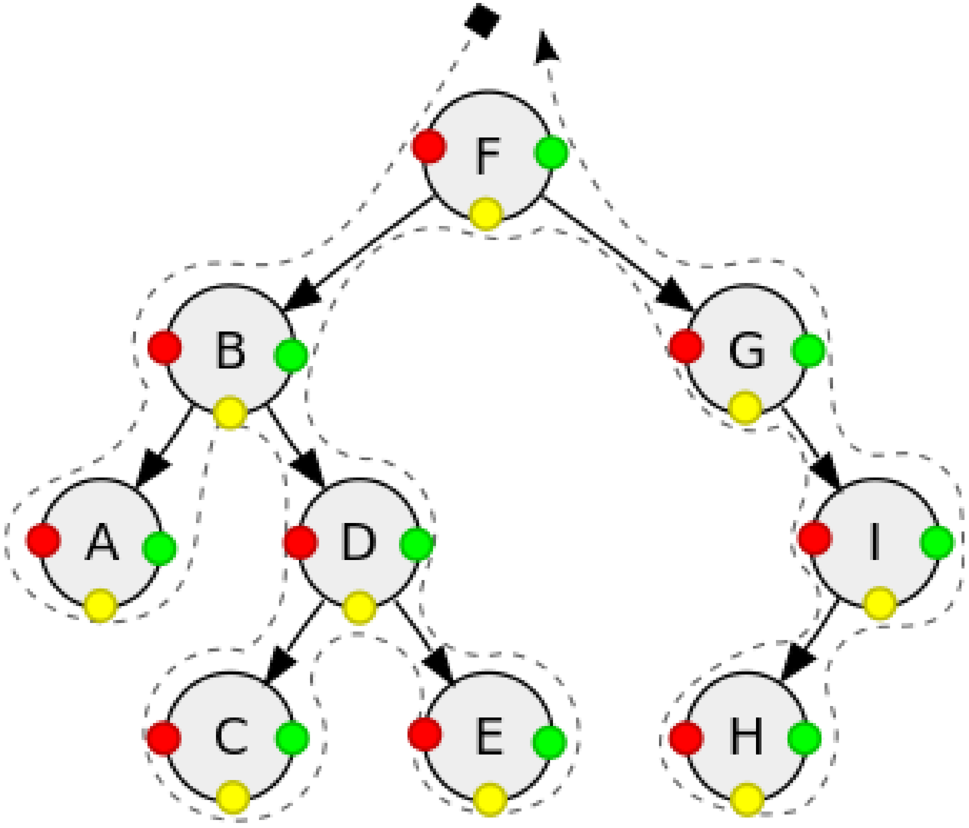
Weighted Small Parsimony

- Cost of a change is not necessarily constant:
  $C^c_{ij} =$ cost of the character c changing from state i to state j.

- **Goal:** minimize the total cost of the tree given the topology and the leaf labels.

Problem: Weighted Small Parsimony.
INPUT: The topology of a rooted phylogenetic tree with leaves having labels in Σ. The costs C^c_{ij} for $i, j \in \Sigma$. There are k possible character values, $|\Sigma| = k$.
QUESTION:
1. What is the minimum possible cost for this topology?
2. What is the optimal labeling of the internal nodes?

Recall: Tree traversals



Public Domain, https://commons.wikimedia.org/w/index.php?curid=83230146

Depth-first traversal of an example tree:

pre-order (red): F, B, A, D, C, E, G, I, H

in-order (yellow): A, B, C, D, E, F, G, H, I

post-order (green): A, C, E, D, B, H, I, G, F.

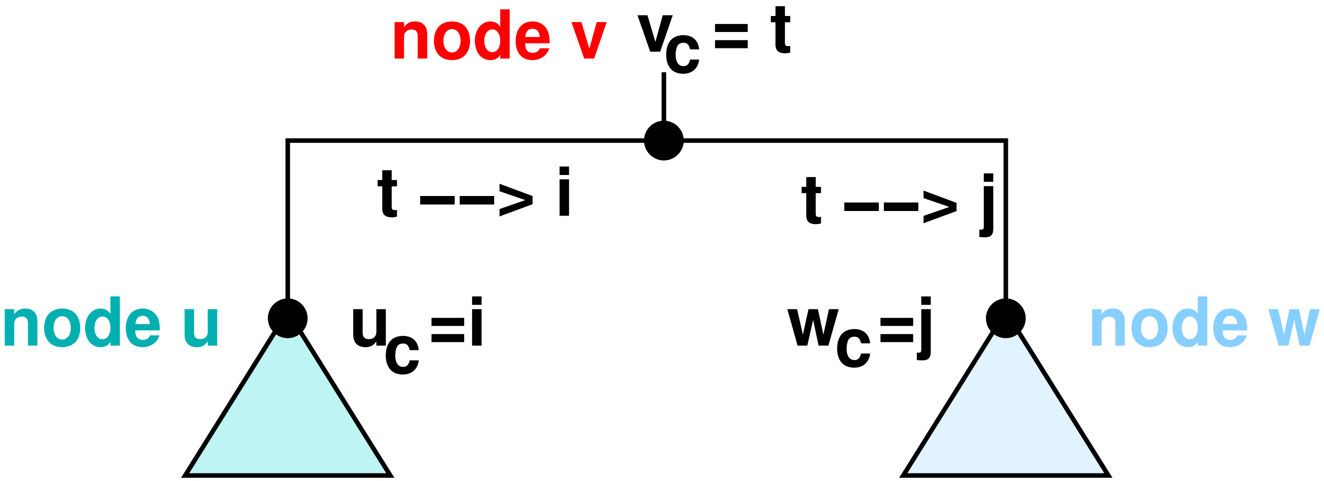
Sankoff's algorithm

Step 1: for each node v and each state t compute quantity $S_t^c(v)$: minimum cost of the subtree whose root is v, assuming that the character value at v is t, i.e. $(v_c = t)$. **In postorder:** for each leaf v:

$$S_t^c(v) = \begin{cases} 0 & v_c = t \\ \infty & \text{otherwise} \end{cases}$$

For an internal node v, with subnodes u and w:

$$S_t^c(v) = \min_i \{C_{ti}^c + S_i^c(u)\} + \min_j \{C_{tj}^c + S_j^c(w)\}$$



Sankoff's algorithm

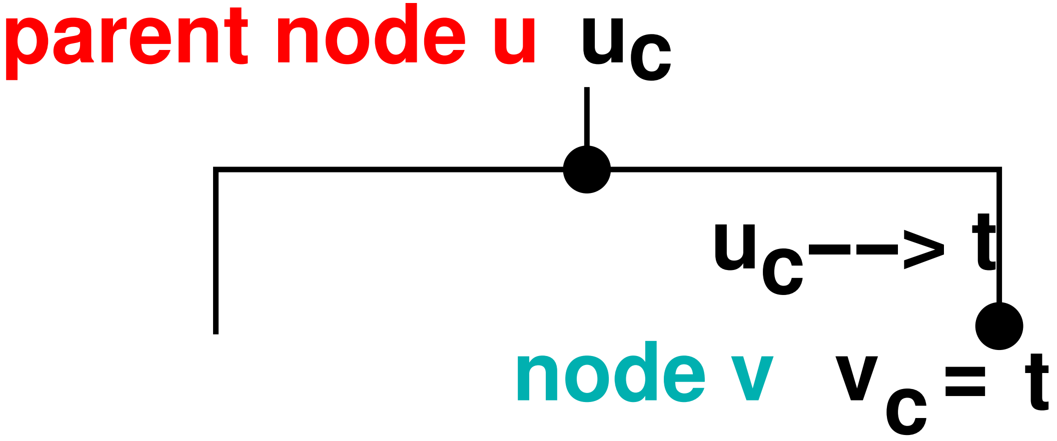
For m characters, minimum total cost of a tree with root r:

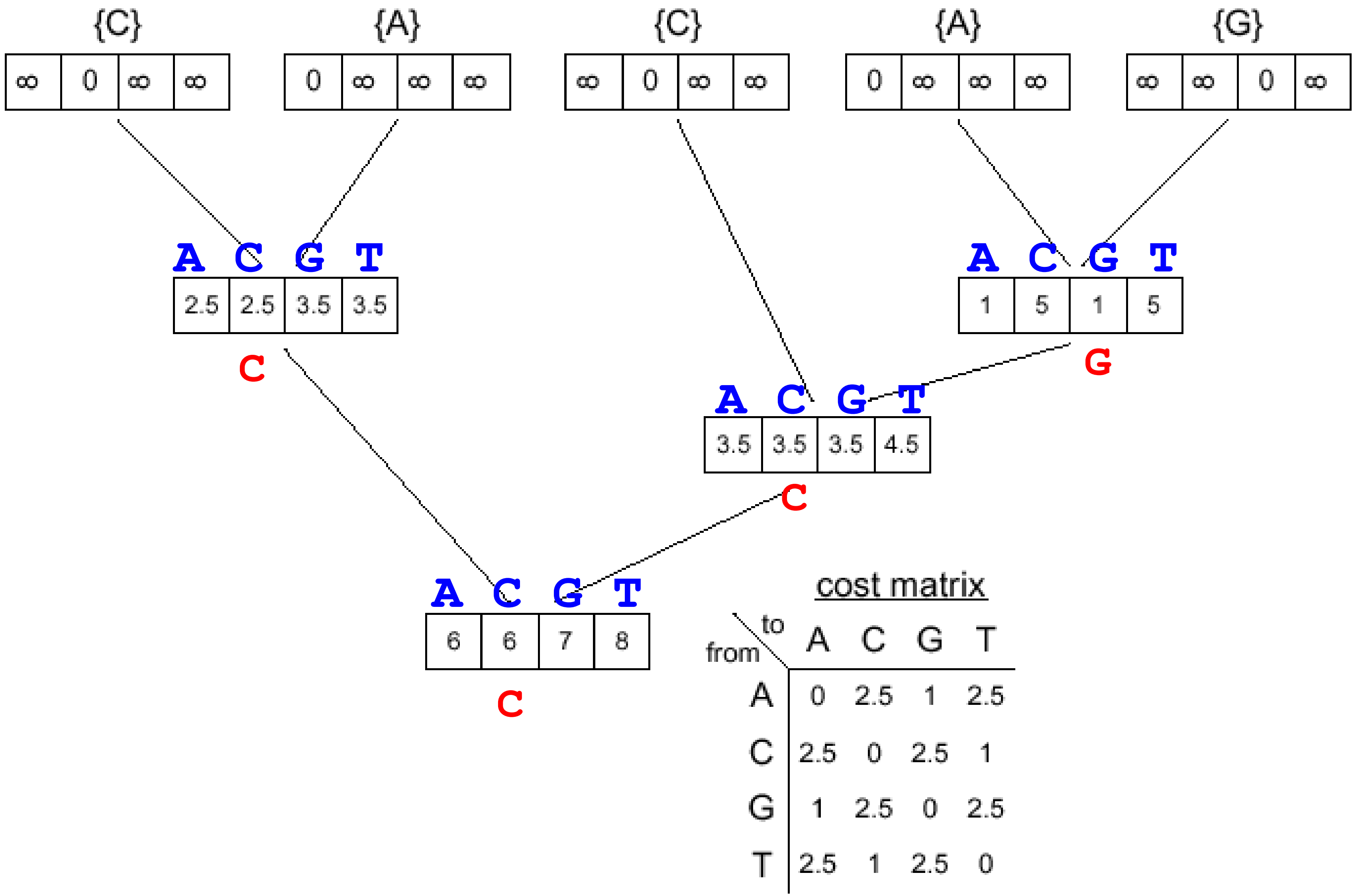
$$S(T) = \sum_{c=1}^{m} \min_t S_t^c(r)$$

Step 2: Based on $S_t^c(v)$, determine the **optimal values for each character** in internal nodes. **Preorder:** For the root node r, choose character value $r_c = \arg\min_t S_t^c(r)$.

For any other node v, with parent node u,

$$v_c = \arg\min_t (C_{u_c t}^c + S_t^c(v))$$





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

Adapted from Figure 8.10 in R. Shamir, Orly Stettiner, R. Gabor: Algorithms for Molecular Biology 8.1 Preface: Phylogenetics and Phylogenetic Trees

Large Parsimony

Final goal: find the optimal phylogeny, not just the optimal internal labeling of a given phylogeny.

Problem: Large Parsimony.
INPUT: A matrix M describing m characters of a set of n species,
QUESTION: What is the optimal phylogeny for these species, i.e., the one minimizing the parsimony score?

Remark: weighted and a non-weighted version, but difference is not essential. It can be shown that this **problem is NP-hard**. However, several approximation heuristics exist.

Branch and Bound

- **Branch-and-Bound** (B&B) deals with optimization problems over a **search space** that can be presented as the **leaves of a tree**.

- First used for parsimony by (Hendy and Penny, 1982).

- Works when the **search tree is monotonous:** the score of each node in the search tree is at least as bad as that of any of its ancestors.

- B&B is **guaranteed to find the optimal solution**, but its complexity in the worst case is as high as that of exhaustive search.

- Basic version: Tree is traversed in some order, cost of the best leaf found so far is kept as a bound C'. When a node is reached whose cost is $C > C'$, the tree is **pruned at that node**.

Branch and Bound for Parsimony

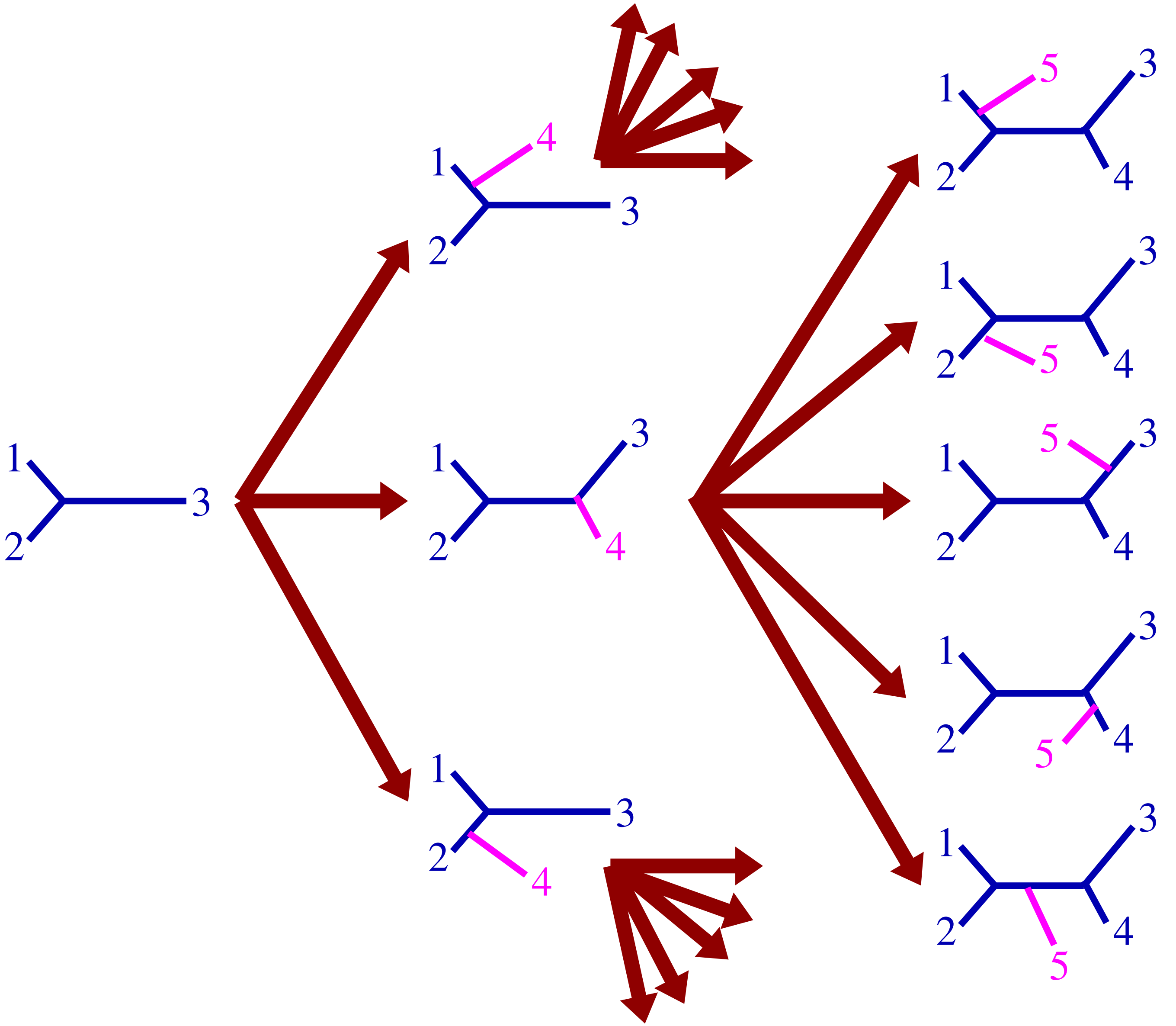
- **Parsimony:** present the search-space as a search tree:
  - k-th level of search tree: nodes represent all possible phylogenetic trees with k leaves for the first k species,
  - Children of such a node: all phylogenetic trees created by adding the $(k+1)$-th species.

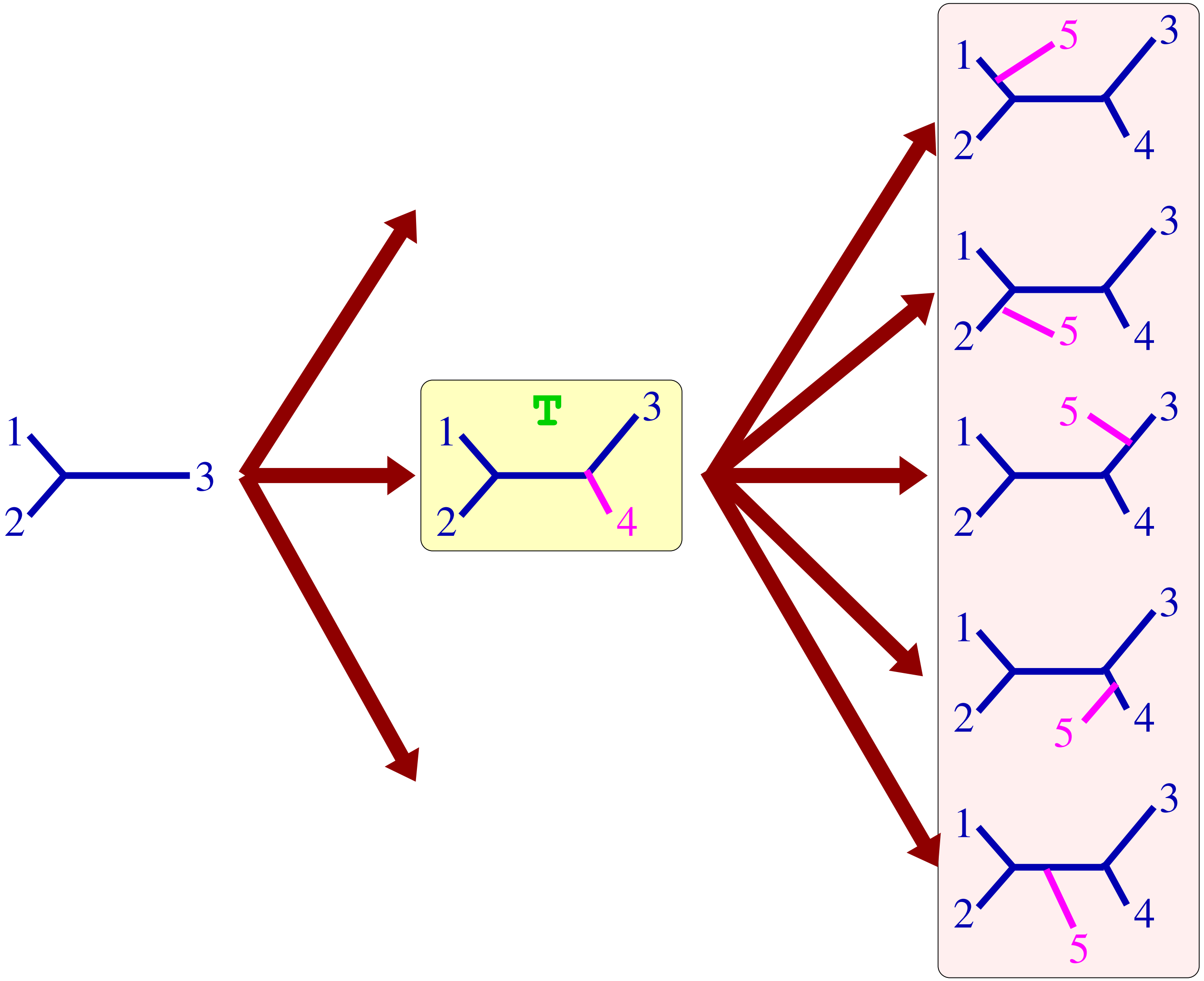
- Search tree is **monotonous**, since adding a node to a given tree can **never reduce** its parsimony score.

- **Does not lower worst-case time complexity**. However, in real-life test cases it proved to speed up the search considerably.

- Plausible strategy: Start with distance-based approach. Neighbor joining $\rightsquigarrow$ initial topology T' $\rightsquigarrow$ compute its parsimony cost C' $\rightsquigarrow$ use this as initial bound.

Branch and Bound for Parsimony (cont'd)





If partial tree T has $cost = C$,

   & the best complete tree seen has $cost = C' < C$

Then prune expansions of T

Maximum Likelihood Methods

- Given a tree, we often wish to have a statistical measure of how well it describes our data.
- **Likelihood function:** $P($**Data**$|$**Parametrized model**$)$, treated as a function of the parameters.
- In our case, the model is a phylogenetic tree, parametrized by its topology T and the set of edge lengths t, representing biological time, or genetic distance, between two connected nodes.
- **Problem 1:** For a set of species with observed values M, what is the likelihood score of a given tree (T, t)?
- **Problem 2 (Maximum likelihood inference):** What is the tree that maximizes $P(M|T, t)$, i.e. best explains the observations?

Computing the Likelihood of a Tree

- **Labels** are the sets of m character values associated with each species, or node in the tree.
- A **reconstruction** is a full labeling of the tree's internal nodes.
- A **branch length** t_{vu} measures the biological time, or genetic distance, between the species associated with these nodes.
- **Assumptions:**
  - characters are pairwise independent,
  - branching is a **Markov process:** probability of a node having a given label is a function only of the **state of its parent node and the branch length t between them.**
  - character frequencies are fixed throughout the evolutionary history, and that they are given as $P(x)$.

The Maximum Likelihood Problem

Problem: Likelihood of a Tree.

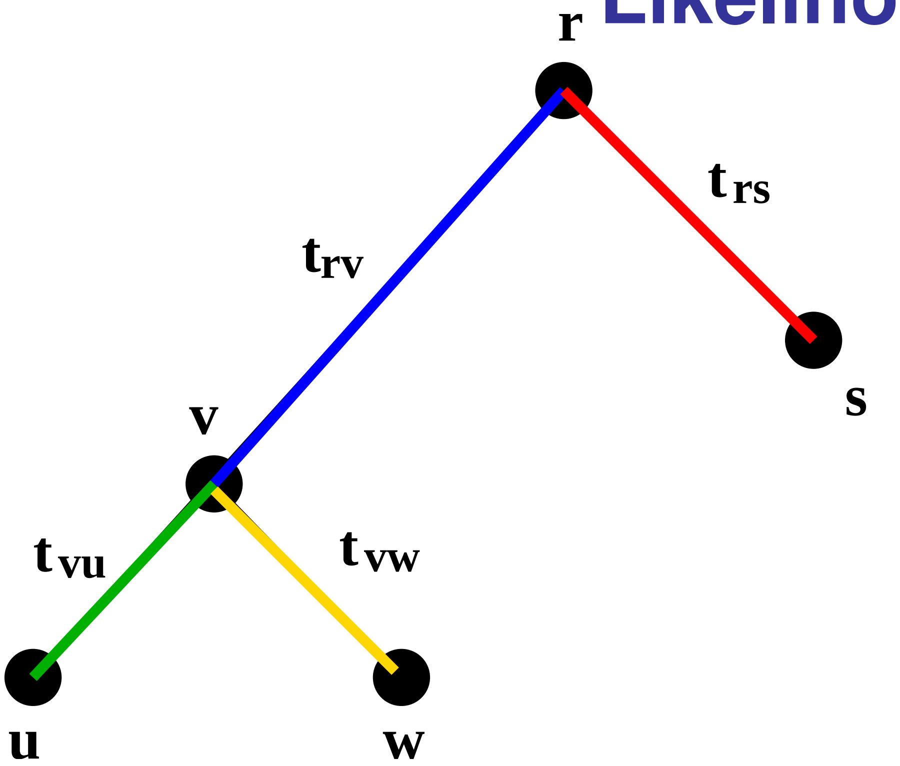
INPUT:

- A matrix M describing a set of m characters for each one of n given species.
- A tree with given topology T, with the above species as the leaves and with known branch lengths t_{vu}.

QUESTION: Calculate the likelihood L of the tree, assuming the m characters are independent:

$$L = P(M|T,t) = \prod_{\text{character } j} P(M_j|T,t)$$

Likelihood of a tree



Labels of internal nodes are unknown ⇝ sum over all possible reconstructions (=labelings of internal nodes).

$$L = \sum_{r}\sum_{v} P(r) \cdot P_{r\to s}(t_{rs}) \cdot P_{r\to v}(t_{rv}) \cdot P_{v\to u}(t_{vu}) \cdot P_{v\to w}(t_{vw})$$

Multiple independent characters:

$$L = \prod_{\text{character } j} P(M_j|T,t) = \prod_{\text{character } j} \left\{ \sum_{\text{reconstruction } R} P(r) \cdot \prod_{\text{edges}} P_{u\to v}(t_{uv}) \right\}$$

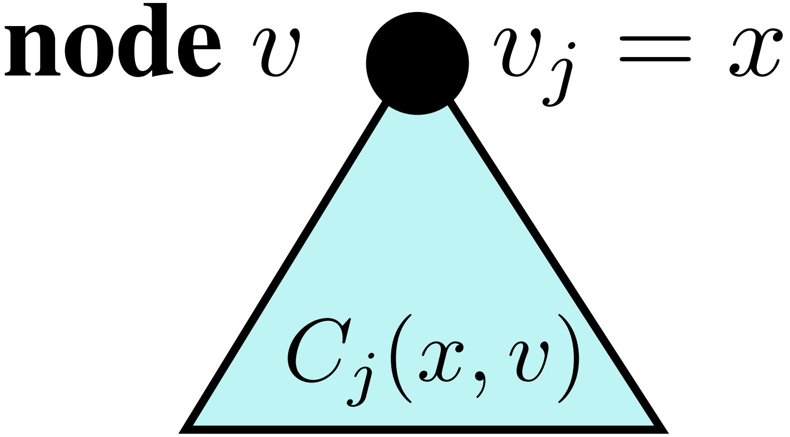
Computing the Likelihood

Dynamic-programming algorithm [Felsenstein, Evolutionary trees from DNA sequences: a maximum likelihood approach. J. Mol. Evol. (1981)17:386-376].

Notation:

Likelihood of v's subtree, given that v has the label x at position j:

$$C_j(x,v) = P(\text{ subtree whose root is } v \,|\, v_j = x\,)$$

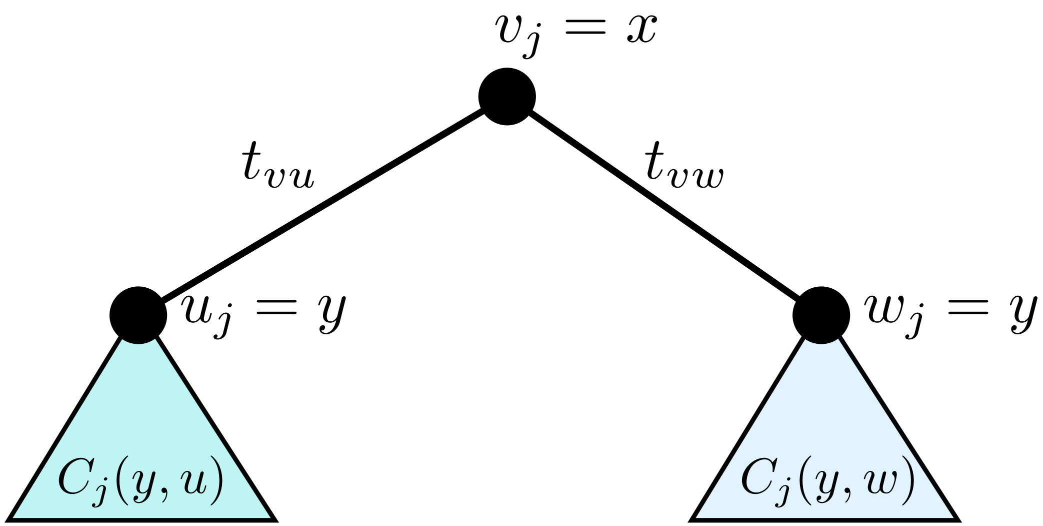


Initialization: For each leaf v and label x:

$$C_j(x,v) = \begin{cases} 1 & \text{if } v_j = x \\ 0 & \text{otherwise} \end{cases}$$

Recursion: Traverse the tree in postorder. For an internal node v with children u and w, compute for each possible label x:

$$C_j(x,v) = \Big[\sum_y C_j(y,u) \cdot P_{x\to y}(t_{vu})\Big] \cdot \Big[\sum_y C_j(y,w) \cdot P_{x\to y}(t_{vw})\Big].$$



Final solution: $L = \prod_{j=1}^{m} \big[\sum_x C_j(x, root) \cdot P(x)\big]$.

Maximizing the Likelihood

- **Optimal Branch Lengths**. Given the topology, find the optimal branch length (optimality = maximum likelihood).

  No analytical solution known. Use numerical methods such as conjugate gradients, based on the derivatives $\frac{\partial}{\partial t_{vw}} P_{x \to y}(t_{vw})$.

- **Optimal topology**. Even harder problem.

  EM-like methods have been proposed:
  Iteratively optimize topology and branch lengths, e.g. “Structural EM” [Friedman et al, J Comput Biol. 2002; 9(2):331-353].

Bayesian approaches

- Instead of solving for the maximum likelihood tree, investigate the **distribution of trees, given the observations:**
  $\leadsto$ **Posterior distribution of trees.**
  M: observed characters. T: topology. t: edge lengths.

$$\underbrace{P(T,t|M)}_{\text{posterior}} = \frac{\overbrace{P(M|T,t)}^{\text{likelihood}}}{P(M)} \cdot \underbrace{P(T,t)}_{\text{prior}}.$$

- Typically, we do not have the posterior in analytic form, but we might be able to **draw samples from the posterior.**
- Law of large numbers: Frequency of a **property in the sample** will converge to the posterior probability.
- Example: If a particular tree topology is present in some fraction r of the samples, then r is an estimate of the posterior probability of this topology.

The Metropolis Method

- A method for drawing samples from a posterior distribution.
- Proposal mechanism: A procedure f that generates a tree $(\tilde{T}, \tilde{t})$ randomly based on the current tree (T, t) by sampling from a **proposal distribution**.
- Define posteriors $P_1 = P(T, t|M)$ and $P_2 = P(\tilde{T}, \tilde{t}|M)$.
- Step 1: Build a random tree (T, t) and calculate P_1.
- Step 2: Build a new $f(T, t) = (\tilde{T}, \tilde{t})$ and calculate P_2.
- Step 3: Accept new tree if $P_2 > P_1$.
  If $P_2 < P_1$, accept only with probability P_2/P_1.
  If accepted, new sample is $(\tilde{T}, \tilde{t})$, otherwise sample is (T, t).
- Step 4: If an appropriate number of samples have been taken, stop. Else, go to Step 2.

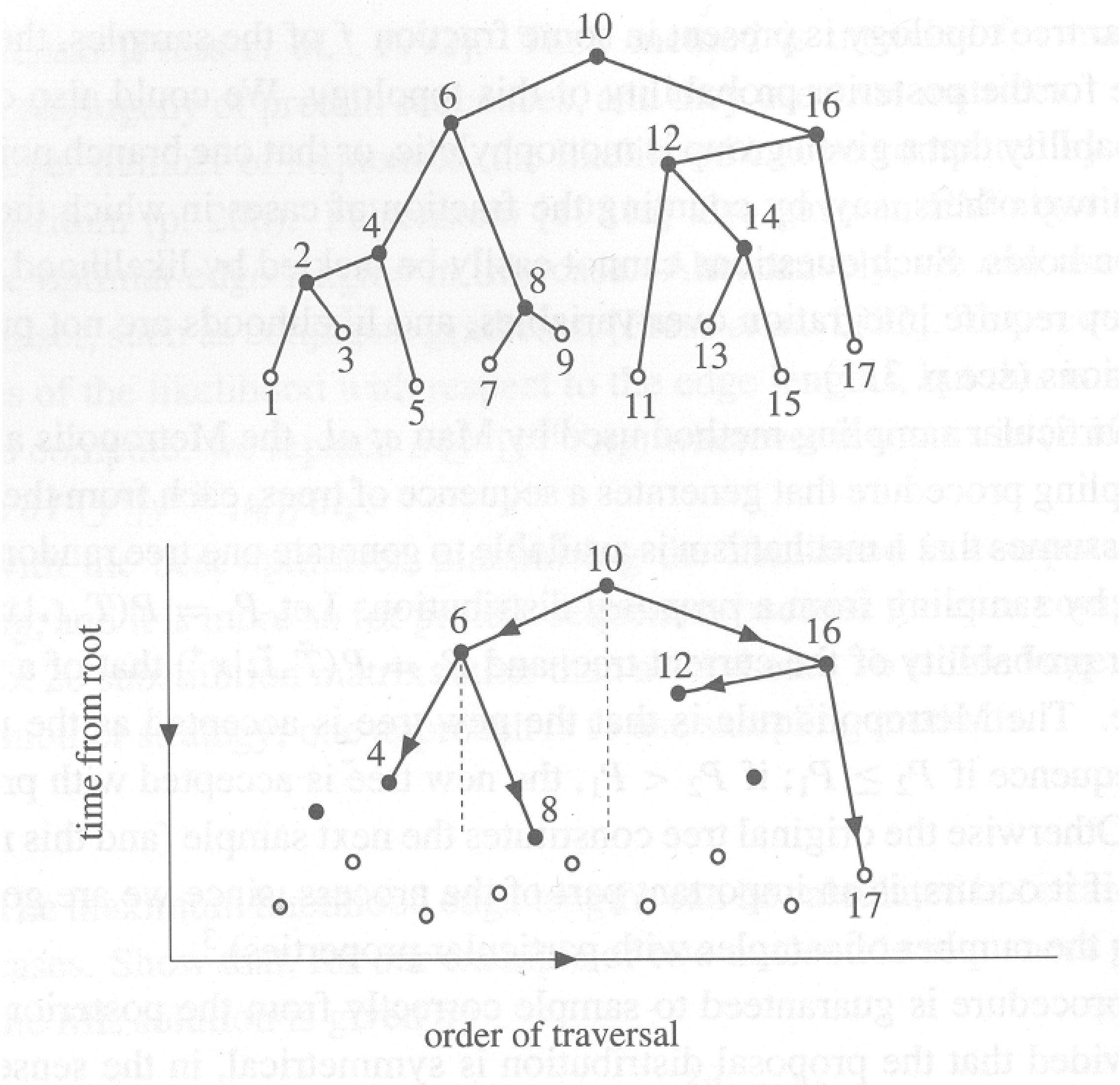
The Metropolis Method (cont'd)

- Note: only the rate P_2/P_1 must be calculated
  $\leadsto$ exponentially large sum $P(M) = \sum_{\text{all trees } (T,t)} P(M,T,t)$
  in Bayes formula is avoided!
- Guaranteed to **asymptotically sample correctly from the posterior distribution**, if the proposal distribution is symmetric: Proposing $(\tilde{T},\tilde{t})$ from (T,t) is the same as proposing (T,t) from $(\tilde{T},\tilde{t})$.
- Crucial point: find suitable proposal distribution for trees. Exploration-exploitation trade-off:
  - If proposed tree is merely sampled randomly, the posterior probabilities will be low $\rightarrow$ low acceptance rate.
  - If proposed tree is too close to the current tree, many steps will be needed to explore the space of trees.

A Proposal Distribution for Trees

- (Mau et al., 1996): **Traversal profile.** Equivalent to the original tree (so tree can be reconstructed from profile), but allowing more convenient manipulations of the topology.
- Node is placed at height h = sum of the edge lengths from root to that node.
- Nodes are regularly spaced horizontally, in the order given by an in-order traversal of the tree.
- For a node k, all left children have numbers $< k$, and all right children $> k$.
- Proposal procedure: Randomly shifting the positions of nodes up and down.
- Relative heights of nodes switched $\rightsquigarrow$ new topology produced.
- Additional proposal mechanism reorders the leaves.

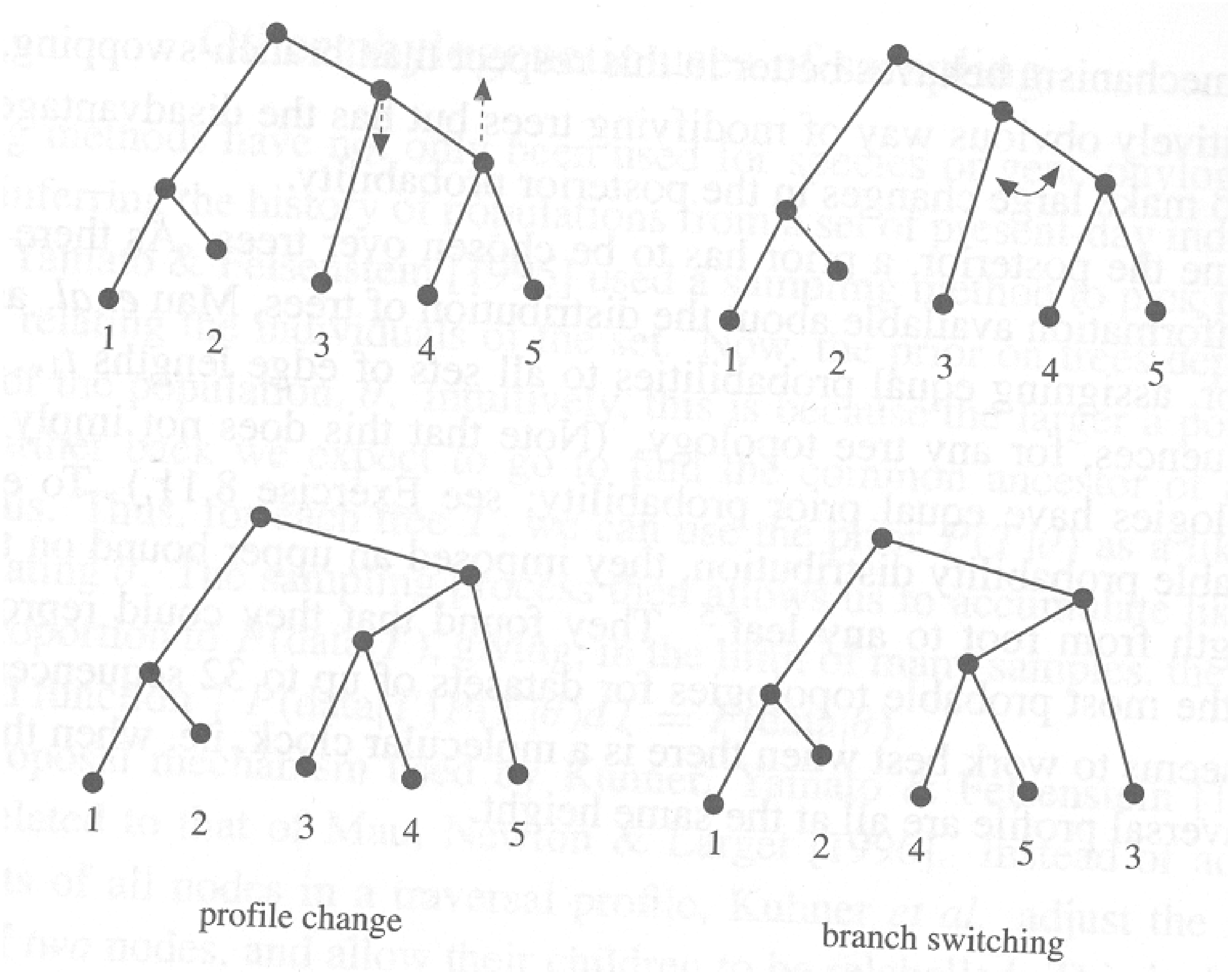
A Proposal Distribution for Trees



Above: an example of a tree with its nodes numbered in the order of the traversal profile. Below: Reconstruction of the tree from the traversal profile.

Durbin et al., Cambridge University Press. https://doi.org/10.1017/CBO9780511790492.004

A Proposal Distribution for Trees (cont’d)



Durbin et al., Cambridge University Press. https://doi.org/10.1017/CBO9780511790492.004