Phylogenetics

The small and the large phylogeny problems

CMSC 423 · Fall 2026

From many aligned sequences to the history that produced them.

UNIVERSITY OF MARYLAND

Some material adapted from slides by Carl Kingsford.
Figures redrawn; published figures credited on their slides.

From Lecture 10 to today

Lectures 8 – 10

How do two sequences correspond?

global · local · affine · linear space

→

Today

What history explains many sequences?

parsimony · likelihood · distances · tree search

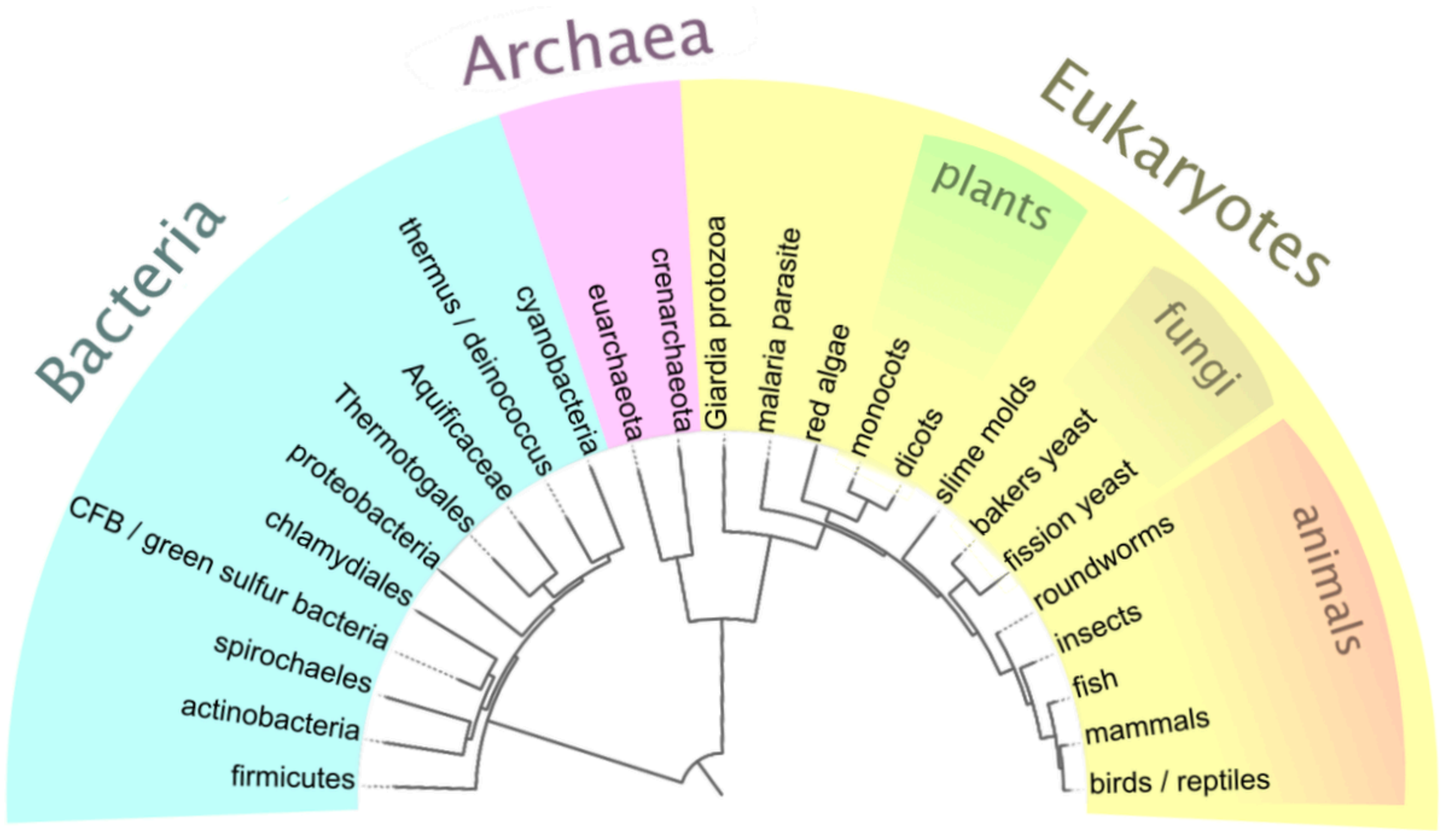
- **Input**: a multiple alignment of n sequences (the *taxa*). Each column is one *character*.
- **Output**: a tree. Leaves are the taxa; internal nodes are ancestors we never observe.
- Two problems: **small** – the tree is given, label its ancestors; **large** – find the tree itself.

Why trees

Evolution leaves a branching record in sequences

Why trees – Parsimony – Likelihood – Distance methods – Tree search – Support

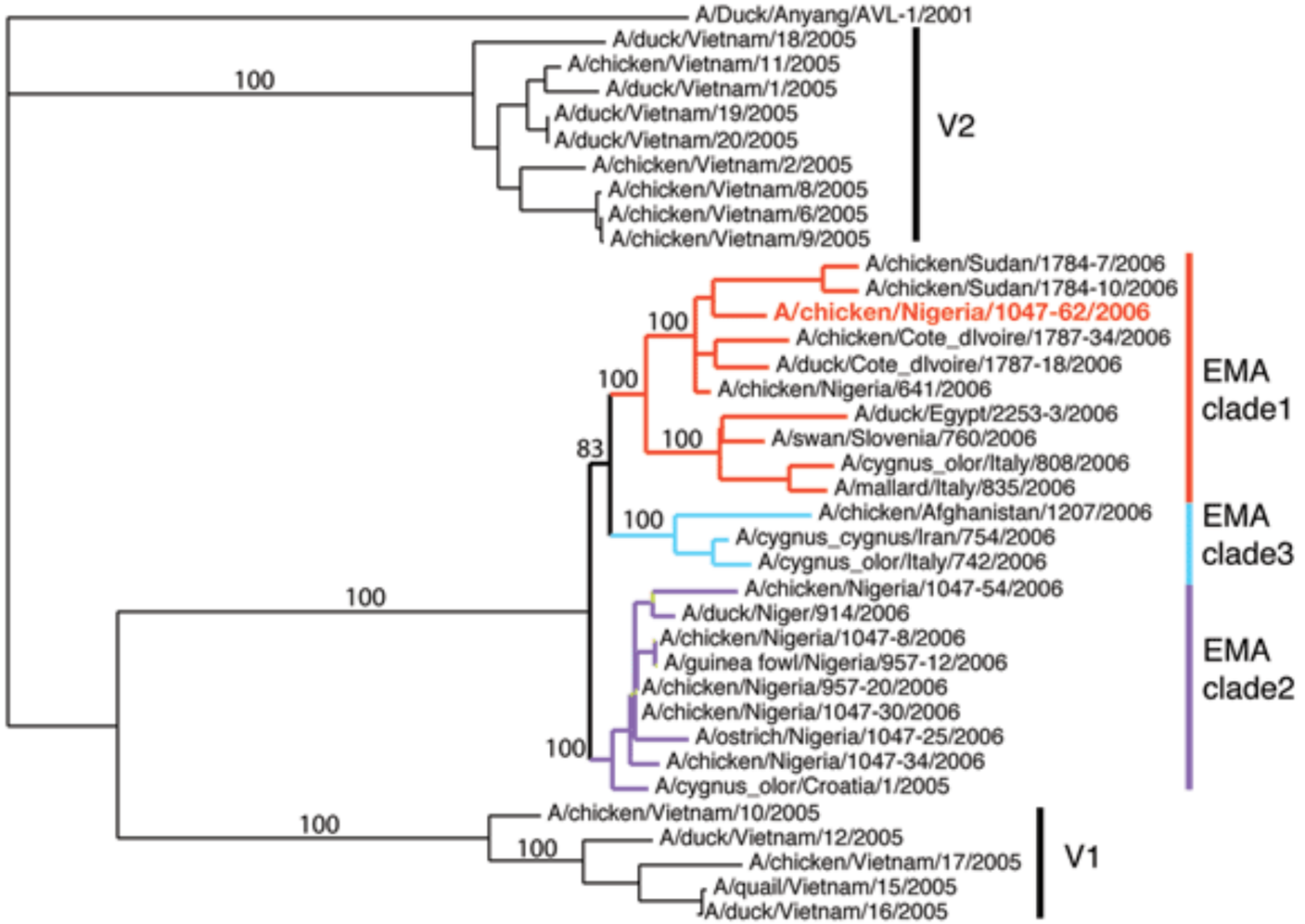
The tree of life



- Three domains: **Bacteria**, **Archaea**, **Eukaryotes**.
- Inferred from sequences: ribosomal RNA first (Woese, 1977), now whole genomes.
- The Open Tree of Life combines thousands of published trees into one synthesis.

Image: Wikimedia Commons, “Tree of life” (simple.wikipedia.org)

Tracking an outbreak: H5N1 influenza



- Whole-genome trees of H5N1 isolates from Europe, the Middle East and Africa (EMA).
- The EMA isolates fall into **three well-supported clades** (coloured). Placing a new isolate on the tree shows which lineage it came from.
- The numbers on edges are **bootstrap support**: how often a clade reappears when the data are resampled (end of lecture).

Figure: Salzberg, Kingsford et al., Emerg. Infect. Dis. 13(5), 2007

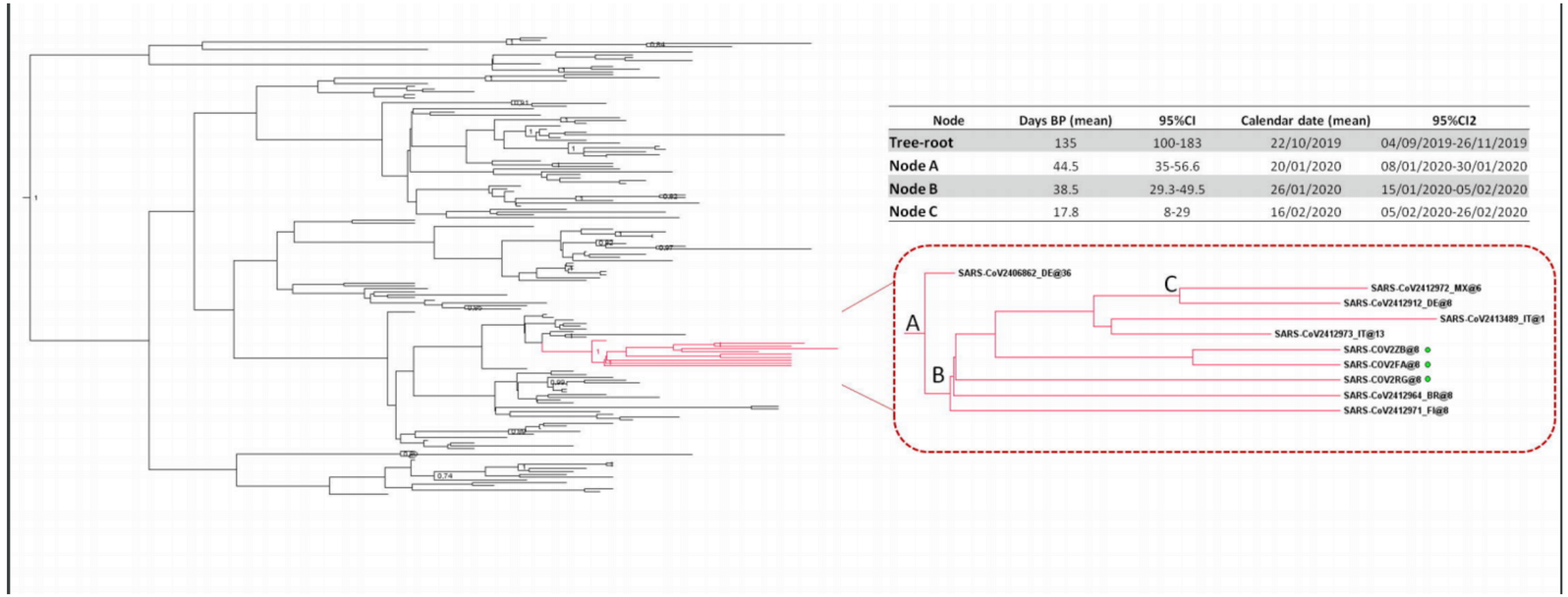
Dating an introduction: SARS-CoV-2 in Italy

J Med Virol. 2020 Sep;92(9):1637-1640. doi: 10.1002/jmv.25794. Epub 2020 Apr 10.

Genomic characterization and phylogenetic analysis of SARS-COV-2 in Italy

Gianguglielmo Zehender [1 2 3], Alessia Lai [1 2], Annalisa Bergna [1], Luca Meroni [4], Agostino Riva [4], Claudia Balotta [1], Maciej Tarkowski [1], Arianna Gabrieli [1], Dario Bernacchia [4], Stefano Rusconi [1 4], Giuliano Rizzardini [5], Spinello Antinori [1 4], Massimo Galli [1 2 4]

- Branch lengths calibrated in *time* date the most recent common ancestor (tMRCA) of the Italian clade.
- Estimate: the virus entered northern Italy in late January – early February 2020, **weeks before** the first Italian case was identified.



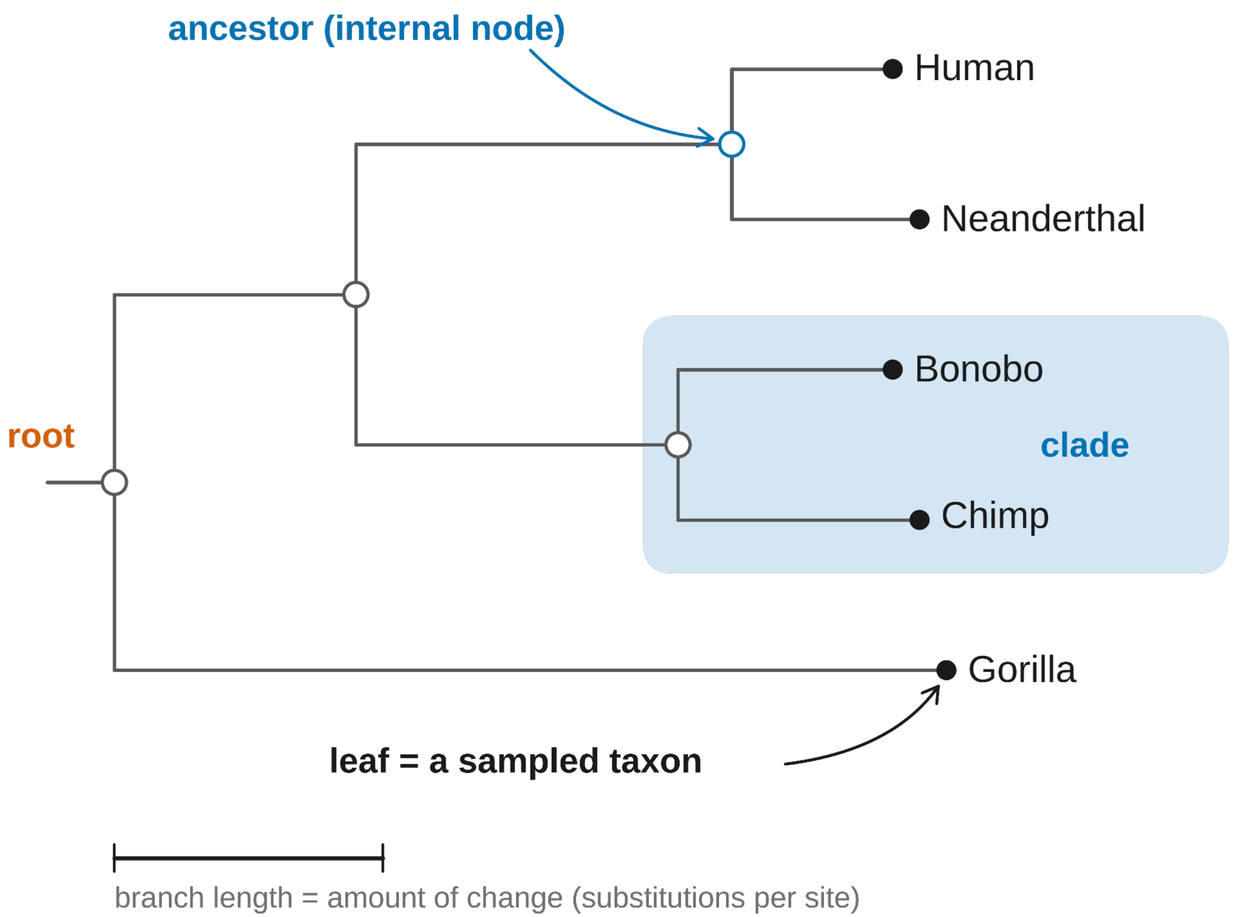
Node	Days BP (mean)	95%CI	Calendar date (mean)	95%CI2
Tree-root	135	100-183	22/10/2019	04/09/2019-26/11/2019
Node A	44.5	35-56.6	20/01/2020	08/01/2020-30/01/2020
Node B	38.5	29.3-49.5	26/01/2020	15/01/2020-05/02/2020
Node C	17.8	8-29	16/02/2020	05/02/2020-26/02/2020

Figure: Zehender et al., J. Med. Virol. 92(9), 2020

Questions a phylogeny can answer

- How many times has a feature arisen? Been lost?
- How is a pathogen evolving to evade the immune system?
- What was the sequence of an ancestral protein?
- Which species are most closely related?
- How fast do new species form?
- Do gains and losses of a trait track the environment, or geography?
- Which features are ancestral to a clade, and which are derived?
- Which structures are homologous, and which are analogous?

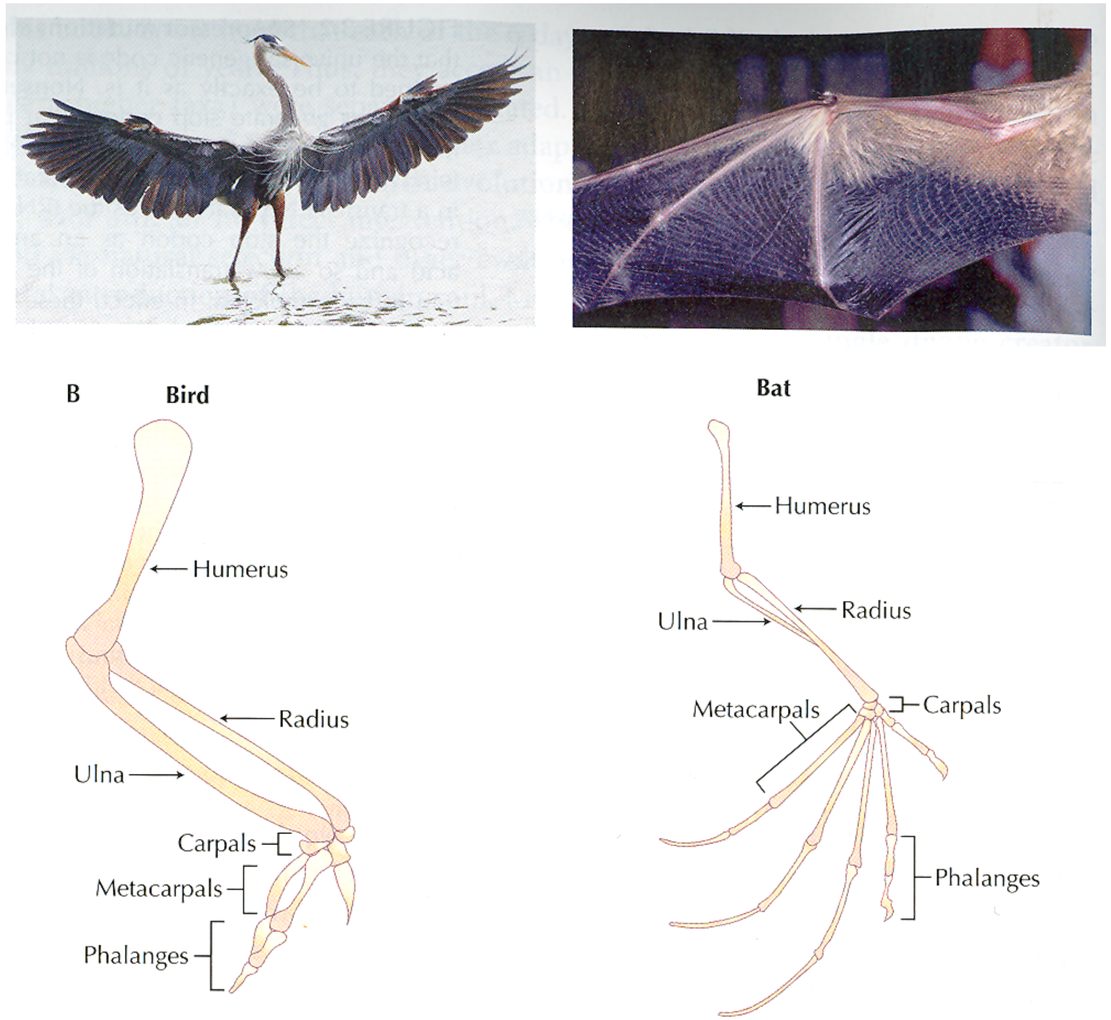
Anatomy of a tree



- **Rooted**: the root is the common ancestor, and time runs from it outward.
- **Unrooted**: only who-is-next-to-whom. Most methods return unrooted trees; an *outgroup* (here Gorilla) places the root.
- **Newick** format writes a tree as nested parentheses:

```
(((Human,Neanderthal),(Bonobo,Chimp)),Gorilla);
```

Characters must be homologous



- **Analogous**: bird and bat wings arose *independently* (convergent evolution).
- **Homologous**: the forelimb bones (humerus, radius, ulna, …) come from a common ancestor.
- So "has wings" is a bad character: it would group birds with bats. A character must compare homologous things.
- For sequences, homology is what an **alignment** asserts: each column holds residues descended from one ancestral position.

Figure: Barton et al., Evolution (Cold Spring Harbor Laboratory Press, 2007)

Appearance can mislead in the other direction

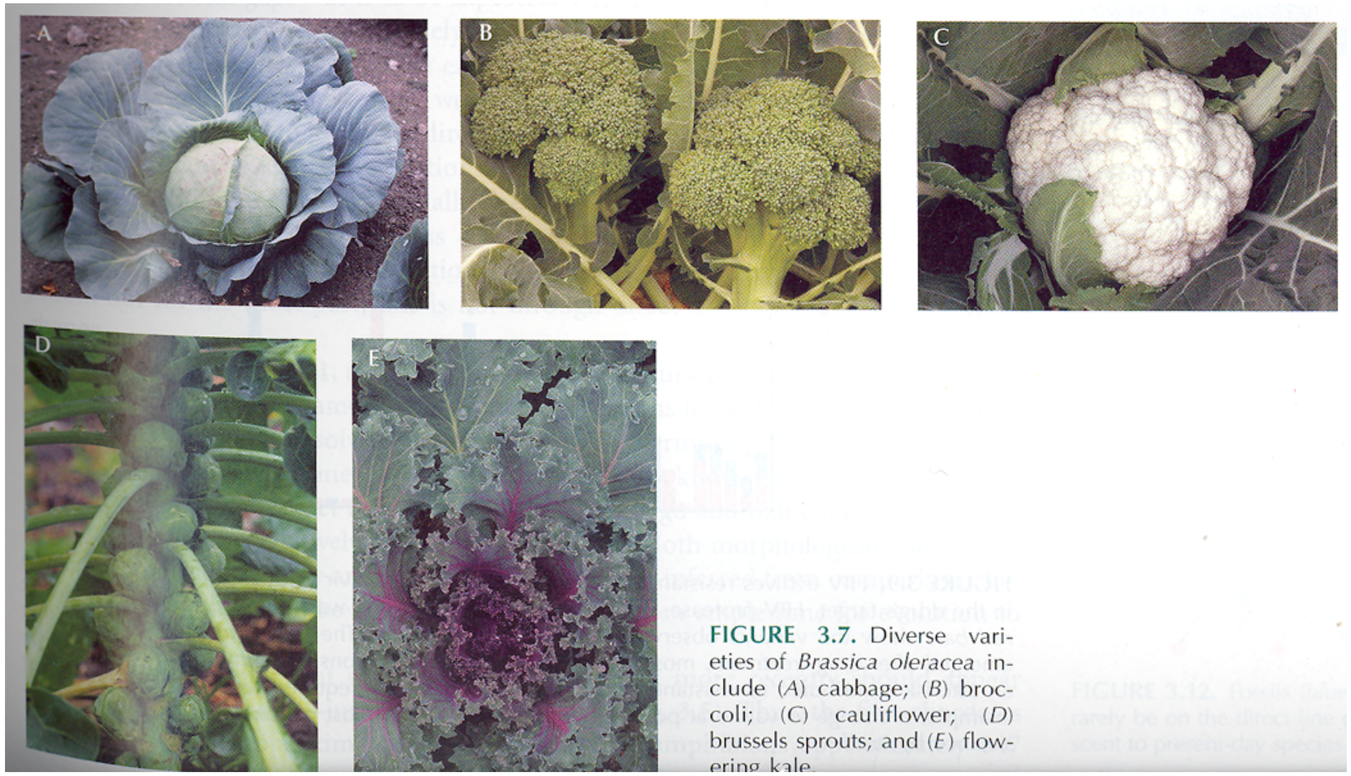


FIGURE 3.7. Diverse varieties of *Brassica oleracea* include (*A*) cabbage; (*B*) broccoli; (C) cauliflower; (*D*) brussels sprouts; and (*E*) flowering kale.

- Cabbage, broccoli, cauliflower, Brussels sprouts and kale are **one species**: *Brassica oleracea*.
- Breeders selected different organs; the plants look unrelated but their genomes are nearly identical.
- Obvious phenotypes are poor markers. Sequences give many characters, all directly comparable.

Figure 3.7: Barton et al., Evolution (Cold Spring Harbor Laboratory Press, 2007)

Study design

Taxon sampling

- How many individuals represent each species?
- Which outgroup roots the tree?
- Can the organisms be collected, cultured, sequenced?

Marker selection

- Reflects the history of the organisms (not recombined, not horizontally transferred)
- Single copy, so we compare orthologs
- Changes enough to separate species, little enough to align reliably

From an alignment to characters

column	1	2	3	4	5	6
Human	**A**	C	G	T	A	A
Neanderthal	**A**	C	G	T	A	G
Bonobo	**C**	C	A	T	G	G
Chimp	**C**	C	A	C	G	G
Gorilla	**T**	C	A	C	G	A

(toy data)

one character; its states are A, C, T

- Rows are taxa; each **column** is a **character**, and the residues in it are its **states**.
- We assume **sites evolve independently**: solve one column at a time, then add (or multiply) over columns.
- Gap characters are usually treated as missing data.
- The alignment comes from Lectures 8–10. Progressive aligners (ClustalW, MAFFT) align along a quick *guide tree* — itself a quick, rough answer to the large problem (NJ or UPGMA).

Two phylogeny problems

The small phylogeny problem

Given: a tree topology, and the states of each character at the leaves

Find: states for the internal nodes that minimise the total cost (or maximise the likelihood)

The large phylogeny problem

Given: the states of each character at the leaves

Find: a tree topology and internal states that optimise the same objective, over all trees

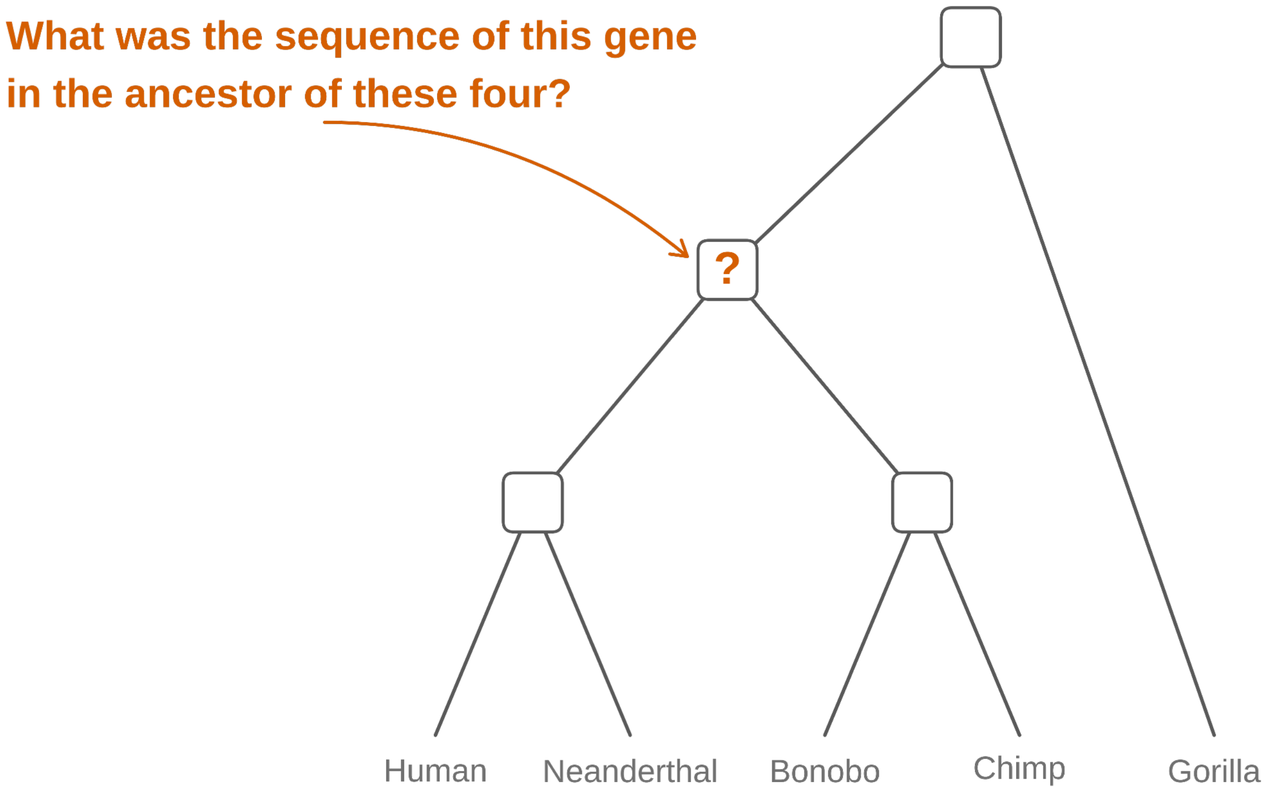
The small problem is solved **exactly** by dynamic programming over the tree. The large one is NP-hard; we search.

Parsimony

The small problem: fewest changes

Why trees – **Parsimony** – Likelihood – Distance methods – Tree search – Support

The small phylogeny problem



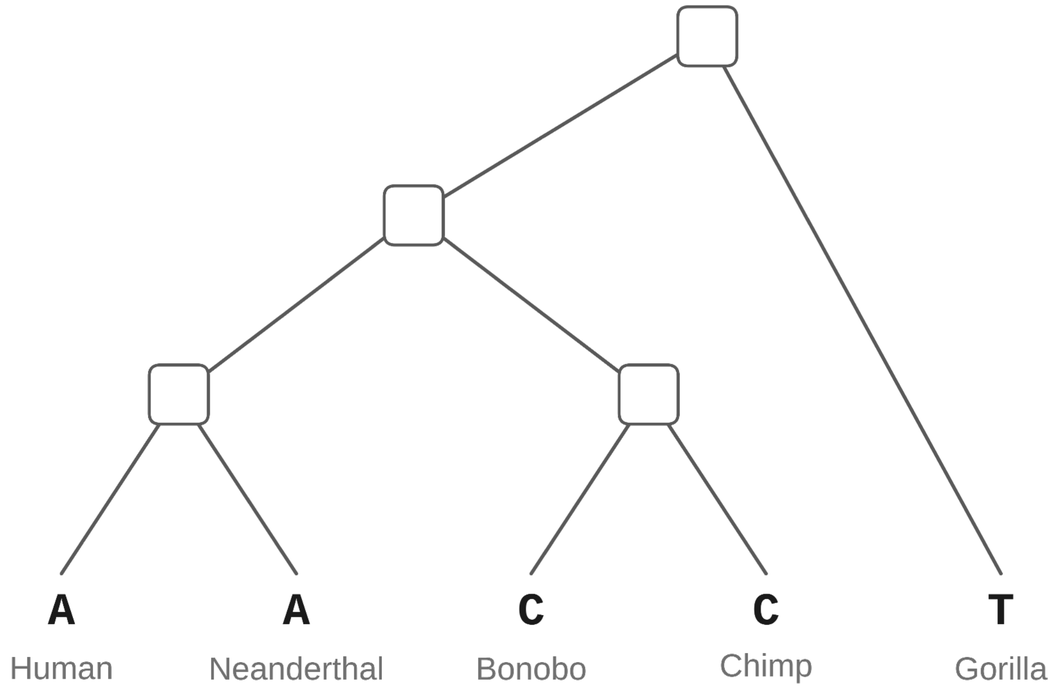
- The tree is **given** (from other data, or from the large problem).
- Leaves: one gene per species, aligned.
- Boxes: the ancestral sequences we want.
- Solve one column at a time.

Parsimony: posit as few changes as possible

One column: the states at the leaves are A A C C T. Choose states for the four boxes to **minimise the number of edges whose two ends differ**.

unit cost c(a, b)

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

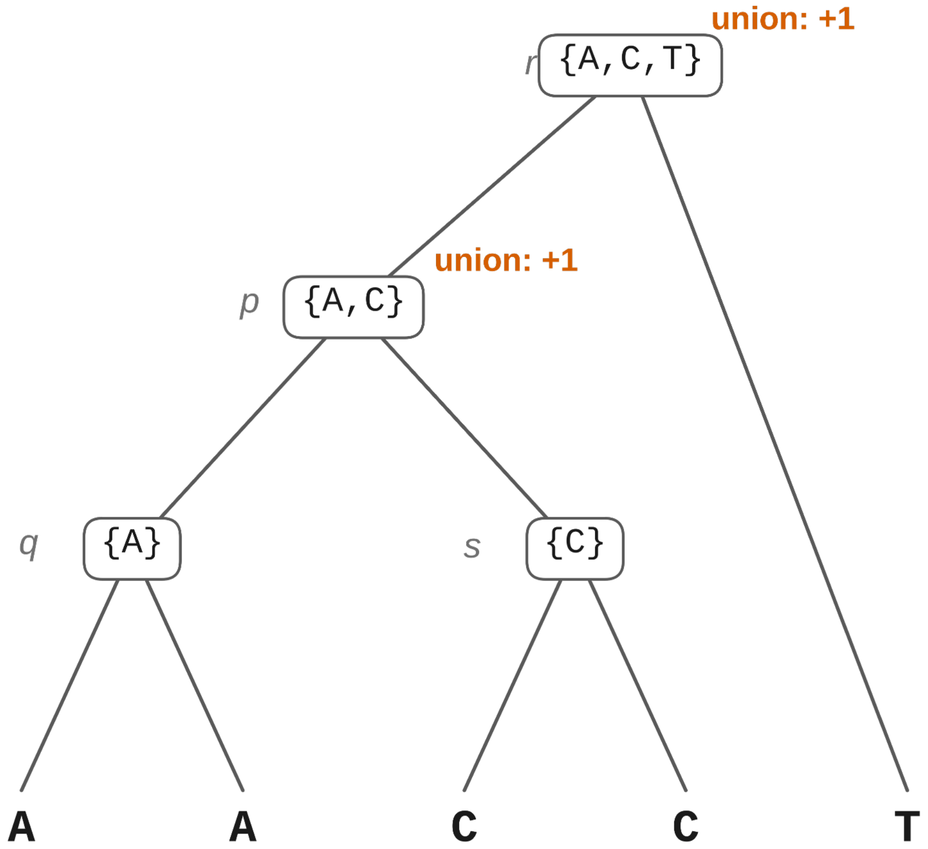


Here every change costs 1. Later: *weighted* costs (Sankoff), then probabilities (likelihood).

Fitch's algorithm, phase 1: sets, bottom-up

Fitch (1971)

```
Phase 1 — post-order (leaves up)
  leaf v:      F(v) = { state at v }
  internal v:  F(v) = F(a) ∩ F(b)  if non-empty
               F(v) = F(a) ∪ F(b)  otherwise;  cost += 1
Phase 2 — pre-order (root down)
  root:  pick any state in F(root)
  v:     keep the parent's state if it is in F(v),
         else pick any state in F(v)
```

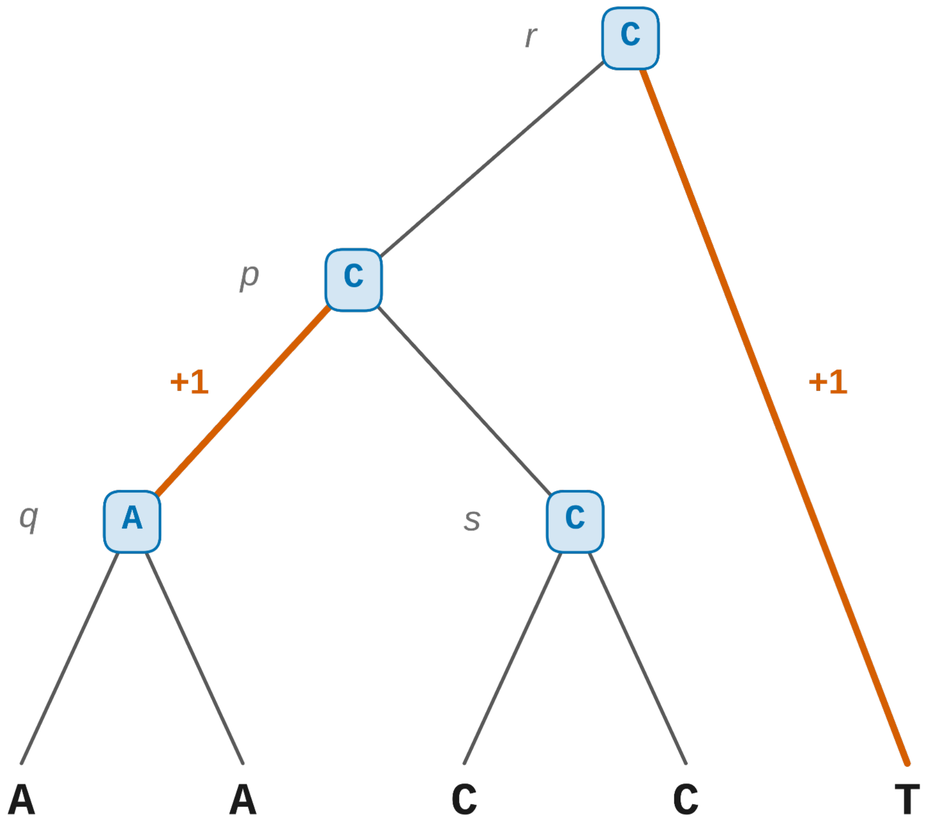


- q: {A} ∩ {A} = {A}. s: {C} ∩ {C} = {C}.
- p: {A} ∩ {C} = ∅, so take the union {A, C} and pay 1.
- root: {A, C} ∩ {T} = ∅, so {A, C, T} and pay 1.
- **Cost = number of unions = 2.**

Fitch's algorithm, phase 2: states, top-down

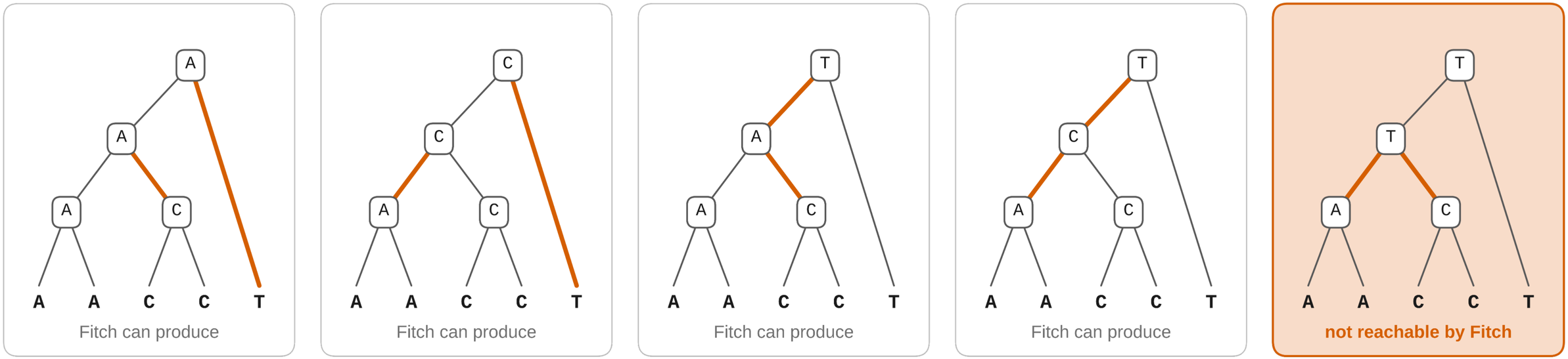
Fitch (1971)

```
Phase 1 — post-order (leaves up)
  leaf v:      F(v) = { state at v }
  internal v:  F(v) = F(a) ∩ F(b)  if non-empty
               F(v) = F(a) ∪ F(b)  otherwise;  cost += 1
Phase 2 — pre-order (root down)
  root:  pick any state in F(root)
  v:     keep the parent's state if it is in F(v),
         else pick any state in F(v)
```



- Root: pick C from {A, C, T}. p: C ∈ {A, C}, keep it.
- q: C ∉ {A}, so q = A (a change). s: C ∈ {C}, keep it.
- Changes on p→q and root→Gorilla: **cost 2**, as phase 1 promised.
- Per column: O(n|Σ|) time for n leaves; O(nk|Σ|) for k columns.

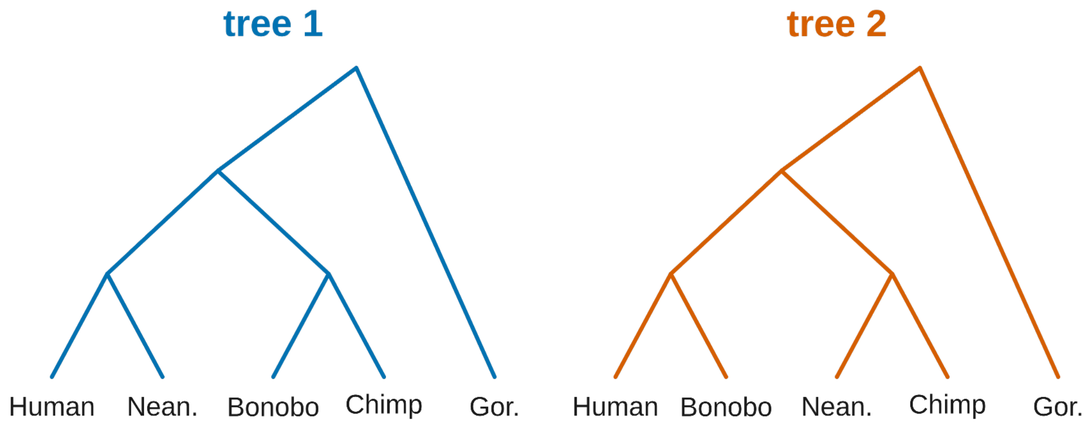
How many optimal labellings? Five



- Every labelling here costs 2. Fitch's phase 2 finds four of them; the fifth puts T at p, but $T \notin F(p) = \{A, C\}$.
- **Lesson**: F(v) is the set of best states for v's *subtree*, not the set of states v takes in some optimum of the whole tree.
- To **count** optima, carry $N_v(a)$ = number of best labellings of v's subtree with v in state a; multiply over children, summing over the children's tied best states. Here $N_{root} = 1\ (A) + 1\ (C) + 3\ (T) = 5$.

Scoring a tree on a whole alignment

	1	2	3	4	5	6	
Human	A	C	G	T	A	A	
Neanderthal	A	C	G	T	A	G	
Bonobo	C	C	A	T	G	G	
Chimp	C	C	A	C	G	G	
Gorilla	T	C	A	C	G	A	
tree 1	**2**	**0**	**1**	**2**	**1**	**2**	**= 8**
tree 2	**3**	**0**	**2**	**2**	**2**	**2**	**= 11**



- Sites are independent: the score of a tree is the **sum of per-column scores**, O(nk|Σ|) for k columns.
- Tree 1 needs 8 changes, tree 2 needs 11: tree 1 is **more parsimonious**.
- Column 2 is constant and costs 0 on every tree: it cannot tell trees apart.
- Choosing the best tree is the **large** problem – later.

Sankoff: when changes have different costs

A cost matrix c(a, b) (e.g. transitions cheaper than transversions). Dynamic programming over the tree (Sankoff, 1975):

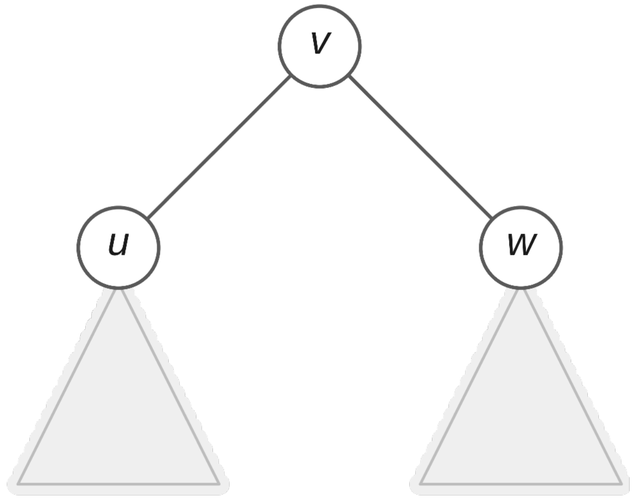
$S_v(a)$ = minimum cost of v's subtree, given that v is in state a

leaf v: $S_v(a) = 0$ if a is the state at v, ∞ otherwise

internal v with children u, w:

$$S_v(a) = \min_b [\, c(a, b) + S_u(b) \,] + \min_b [\, c(a, b) + S_w(b) \,]$$

answer: $\min_a S_{root}(a)$



choose u's best state and w's, independently, given a at v

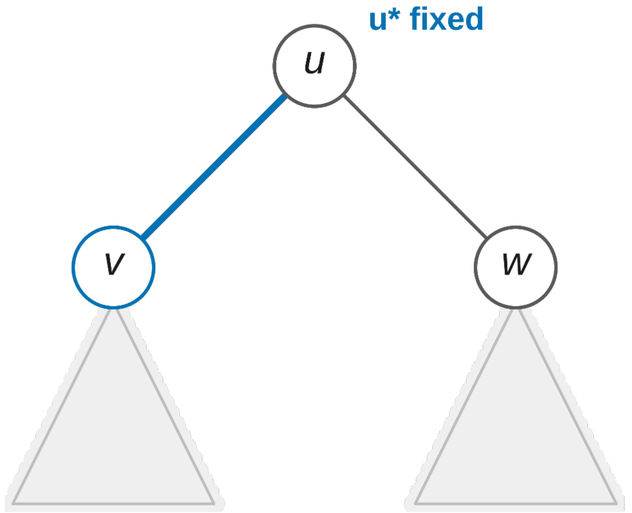
- The two children's subtrees share nothing but v, so given v's state their costs add.
- Per column: |Σ| states × |Σ| choices per edge: **O(n|Σ|²)**. With unit costs, Fitch's F(v) is exactly the set of states minimising S_v.

Sankoff, phase 2: trace the choices back down

root: choose $r^* = \text{argmin}_a\, S_{\text{root}}(a)$

child v of a node u already set to u^*:

$$v^* = \text{argmin}_b \left[\, c(u^*, b) + S_v(b) \,\right]$$



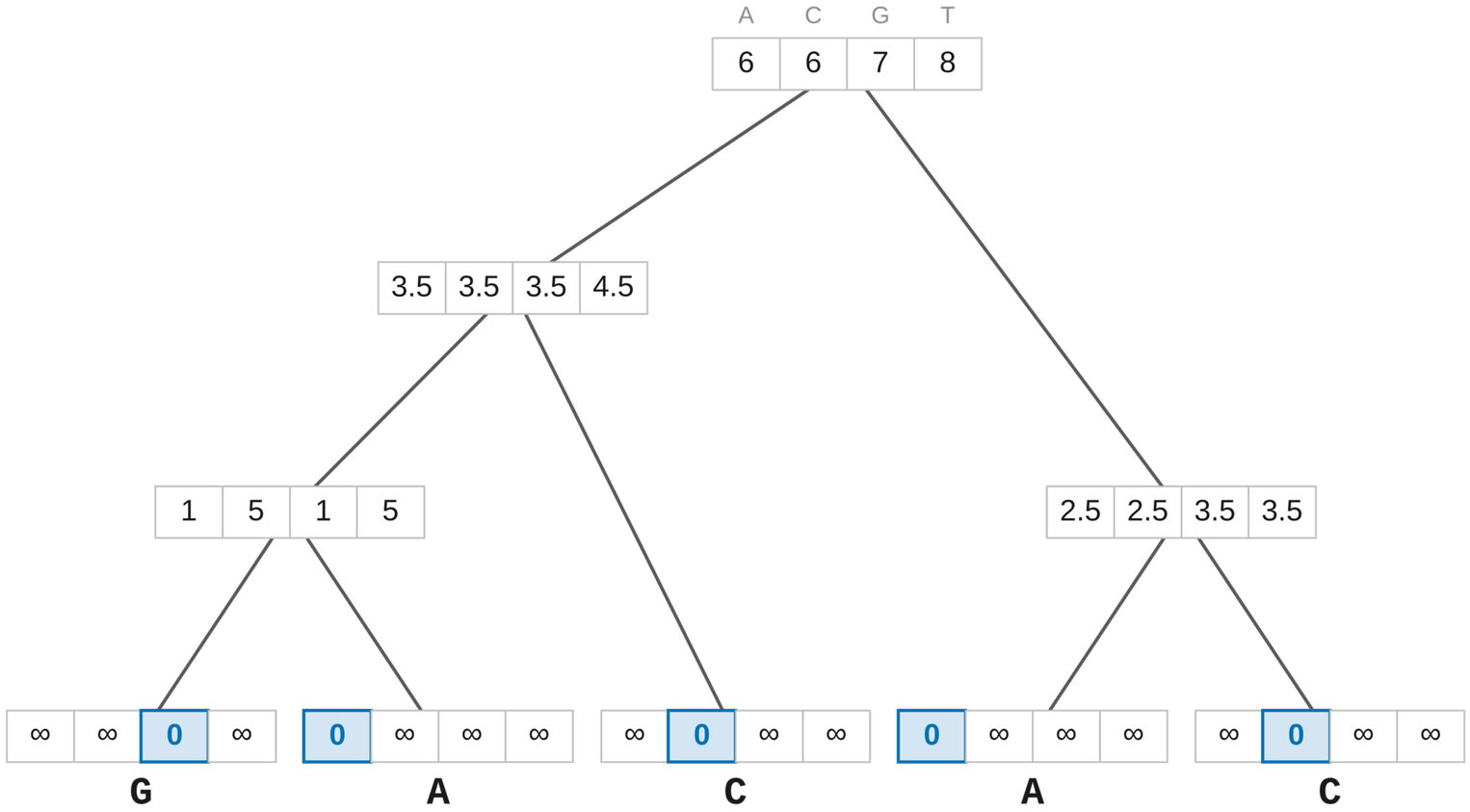
- The same pattern as every traceback in Lectures 8–10: fill a table bottom-up, then re-derive which choice achieved each optimum.
- Store the argmin per (edge, state) in phase 1 to make phase 2 O(n).
- Ties are real: several labellings can be optimal (next slides).

Sankoff, worked: transitions 1, transversions 2.5

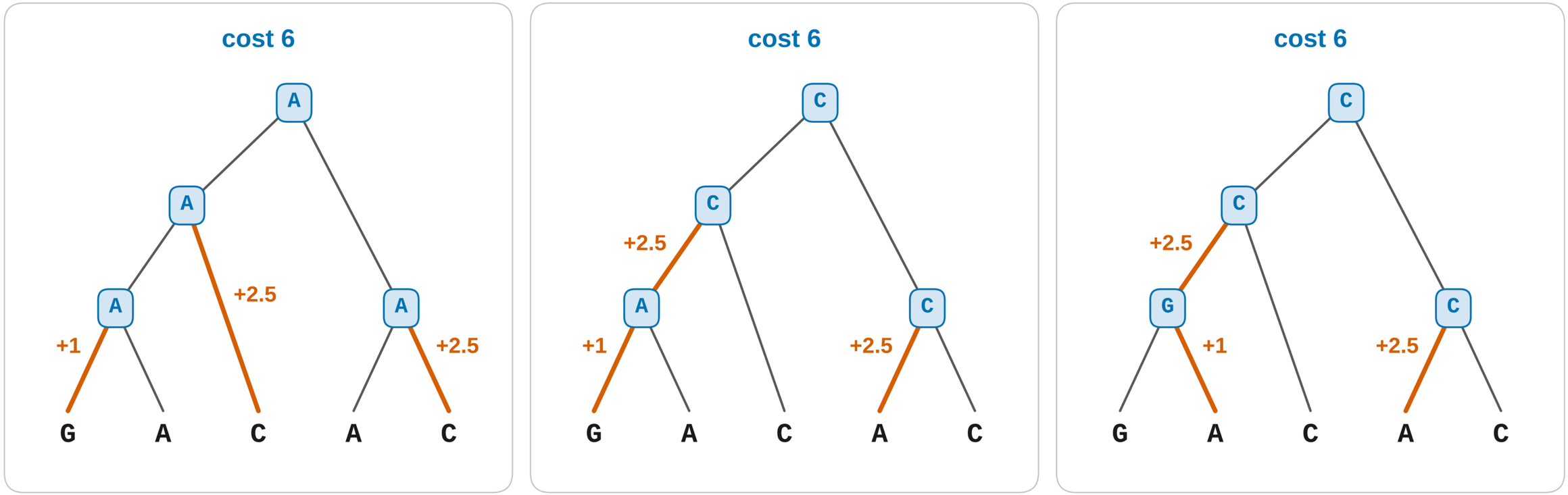
cost c(a, b)

	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

- Each node shows $S_v(A)$, $S_v(C)$, $S_v(G)$, $S_v(T)$.
- Node above G and A: S(A) = c(A,G) + c(A,A) = 1, S(C) = 2.5 + 2.5 = 5.
- Root: **6 6 7 8**. The optimum is 6, and A and C tie.



Sankoff, traceback: three optimal labellings



- One traceback (left) hides the others. The root ties (A or C, both 6); with C at the root, the node above G and A may be A or G.
- Counting optima uses the same counting DP as for Fitch: **3** here, out of $4^4 = 256$ possible labellings.

Likelihood

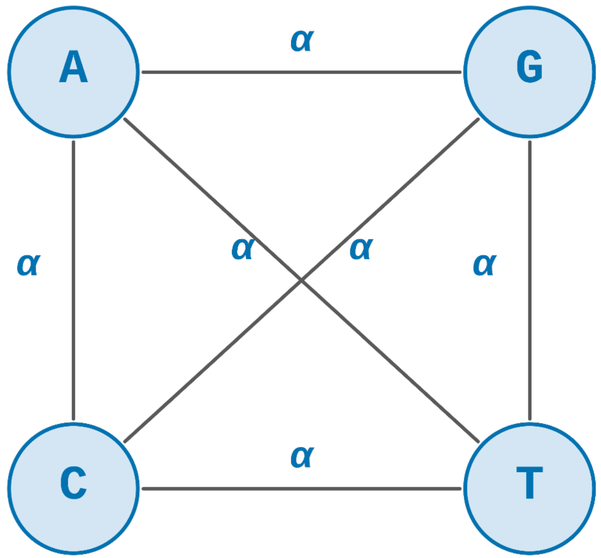
The small problem with a model of evolution

Why trees – Parsimony – **Likelihood** – Distance methods – Tree search – Support

Why a model?

- Parsimony counts changes but ignores **branch lengths**: a change on a long branch is far more likely than on a short one.
- It also ignores **multiple hits**: A → G → A on one branch looks like no change.
- Worse, it can be **consistently wrong** – more data makes it more sure of the wrong tree (long-branch attraction, later).
- Likelihood: write down a **probabilistic model** of how a site changes along a branch, then ask which ancestral states (and later, which trees) make the data most probable.
- Standard model: each site evolves independently, along each branch, as a continuous-time **Markov chain** on {A, C, G, T}.

The Jukes–Cantor model



every change happens at the same rate α

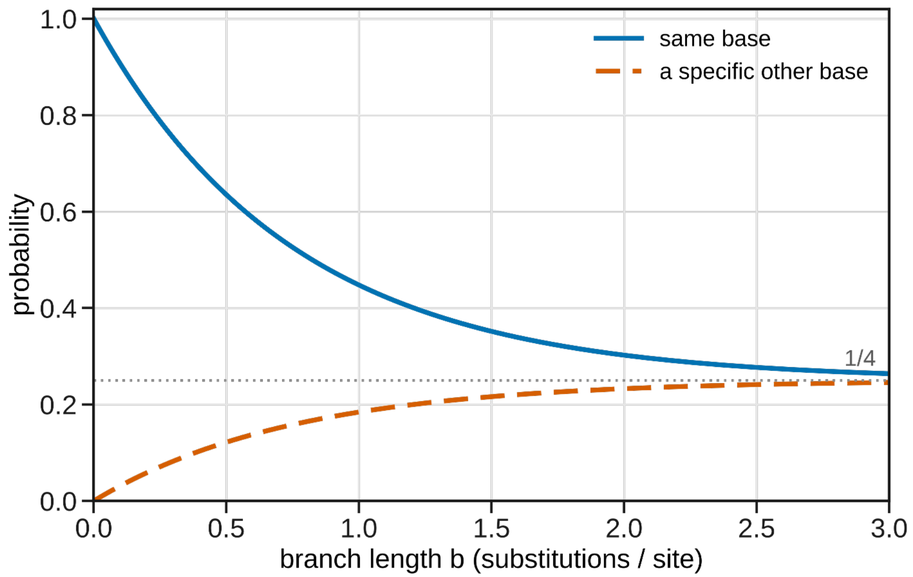
α is a *rate*, not a probability. As b grows, every base becomes equally likely: the sequence forgets where it started.

P(same base after time t) = $1/4 + 3/4\ e^{-4\alpha t}$

P(*a* specific other base) = $1/4 - 1/4\ e^{-4\alpha t}$

branch length $b = 3\alpha t$ = expected substitutions per site

$P_{aa}(b) = 1/4 + 3/4\ e^{-4b/3}$



Jukes & Cantor (1969)

Richer models: GTR and friends

rate matrix Q (rows sum to 0)

	A	C	G	T
A	·	a π_C	b π_G	c π_T
C	a π_A	·	d π_G	e π_T
G	b π_A	d π_C	·	f π_T
T	c π_A	e π_C	f π_G	·

Everything below works for any such model: we only ever need $P_{ab}(t)$ and π.

- **GTR** (general time-reversible): six exchange rates a … f and base frequencies π. Diagonal: minus the row sum.
- **Time-reversible**: $\pi_i Q_{ij} = \pi_j Q_{ji}$ – the chain looks the same run backwards.
- Probabilities over a branch of length t: **$P(t) = e^{Qt}$** (matrix exponential).
- Special cases: **JC** (all rates equal, π uniform), **K80** (transitions ≠ transversions), **HKY** (K80 + unequal π).

Tavaré (1986); Kimura (1980)

The small problem, by maximum likelihood

Given

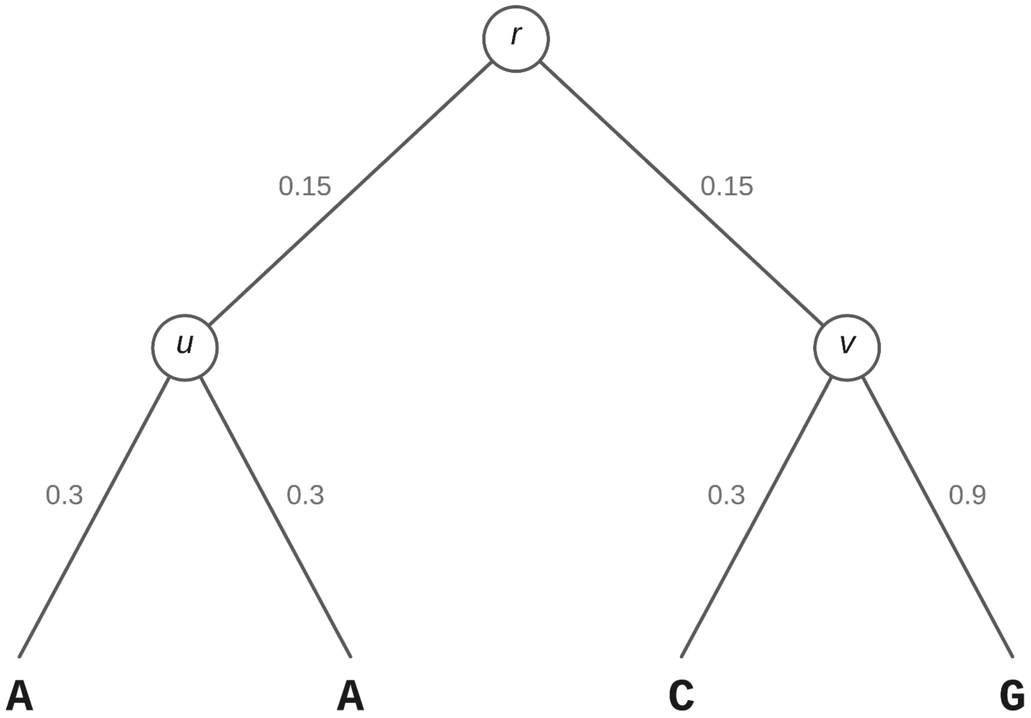
a tree with branch lengths, the leaf states, and *a* model ($P_{ab}(t)$, π)

Compute

the likelihood: the probability of the leaf states, summed over every possible assignment of ancestral states

- Parsimony picks the *best* history. Likelihood **sums over all histories**, each weighted by its probability.
- Sites are independent: L(tree) = ∏ over columns of L(column); in practice we add log-likelihoods.
- Then: which ancestral states are most probable? (end of this section.)

One site, by brute force



For one choice of ancestral states r, u, v:

$$\Pr(\text{leaves}, r, u, v) = \pi_r \cdot P_{ru}(0.15)\, P_{uA}(0.3)\, P_{uA}(0.3) \cdot P_{rv}(0.15)\, P_{vC}(0.3)\, P_{vG}(0.9)$$

We never see r, u, v, so sum them out:

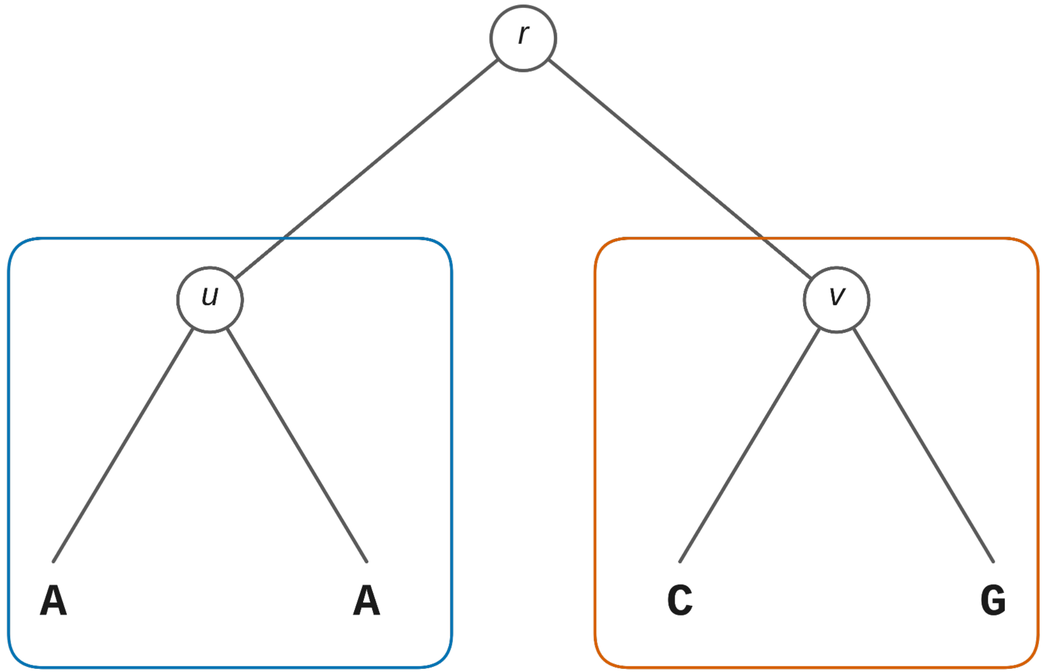
$$L = \Sigma_r \Sigma_u \Sigma_v \Pr(\text{leaves}, r, u, v)$$

- Here $4^3 = 64$ terms. With n leaves: 4^{n-1} terms — $4^{49} \approx 3 \times 10^{29}$ for 50 taxa.
- Branch lengths in substitutions per site; JC model.

Push the sums inside (distributivity)

$$L = \Sigma_r \pi_r \cdot [\, \Sigma_u P_{ru}(0.15) \cdot P_{uA}(0.3) \cdot P_{uA}(0.3) \,] \cdot [\, \Sigma_v P_{rv}(0.15) \cdot P_{vC}(0.3) \cdot P_{vG}(0.9) \,]$$

- The u-bracket mentions only u and its leaves; the v-bracket only v and its leaves. So **each bracket is computed once per value of r**.
- 4 × (4 + 4) terms instead of 64. The nesting of brackets is the tree: ((1, 2), (3, 4)) – its Newick string.
- This is Horner's rule, and the same factoring as a sum-product DP.



Felsenstein's pruning algorithm

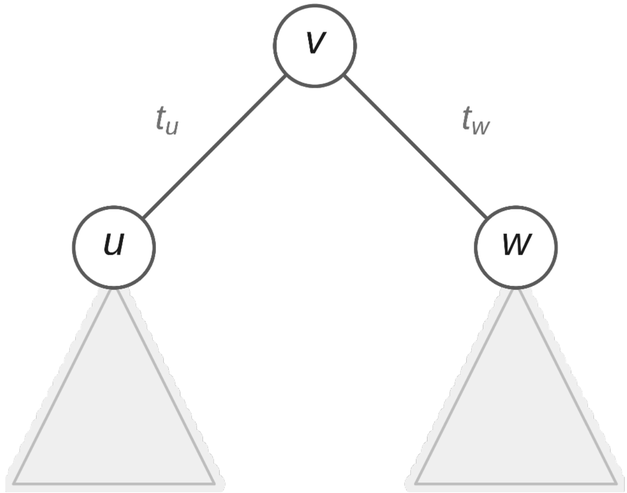
$L_v(a)$ = Pr(the leaves below v | v is in state a)

leaf v: $L_v(a) = 1$ if a is the observed state, 0 otherwise

internal v, children u and w, branch lengths t_u and t_w:

$$L_v(a) = [\ \Sigma_b\ P_{ab}(t_u)\ L_u(b)\ ] \cdot [\ \Sigma_b\ P_{ab}(t_w)\ L_w(b)\ ]$$

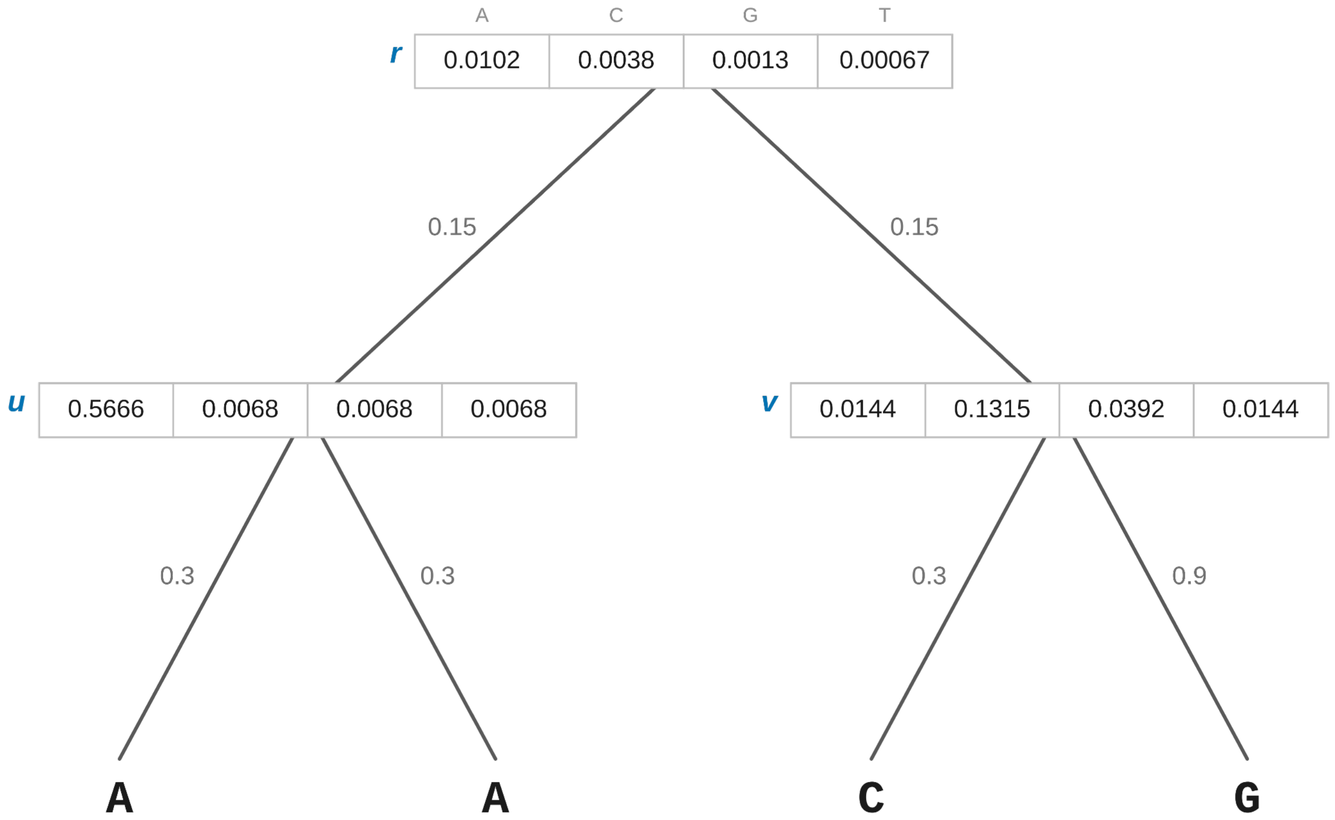
whole tree: $L = \Sigma_a\ \Pi_a\ L_{root}(a)$



- A post-order traversal, exactly like Sankoff – with **(+, min)** replaced by **(×, Σ)**.
- Per site: **O(n|Σ|²)**; for k sites O(nk|Σ|²). Identical columns need computing only once.
- Thousands of factors below 1 underflow a double: store log L, or rescale each L_v and keep the scale factors.

Felsenstein, J. Mol. Evol. 17 (1981)

Pruning, worked



$P_{AA}(0.3) = 0.7527$, $P_{AC}(0.3) = 0.0824$, $P_{AA}(0.9) = 0.4759$, $P_{AC}(0.9) = 0.1747$

- u: both leaves are A, so $L_u(A) = 0.7527^2 = 0.5666$; $L_u(C) = 0.0824^2 = 0.0068$.
- v: leaves C and G. C is best for v (0.1315): it explains the C on the short branch.
- Root: $L = \frac{1}{4} \sum_a L_r(a) =$ **3.991×10^{-3}** – the same as summing all 64 terms.

What the likelihood buys us

- **Compare trees** on the same data: higher likelihood = better explanation (under the model).
- **Fit branch lengths**: maximise L over the t's (numerically, one branch at a time).
- **The root doesn't matter** for a reversible model (Felsenstein's “pulley principle”). Rerooting our example on another edge gives the same $L = 3.991 \times 10^{-3}$.
- **Ancestral states**: which state was each ancestor in? Two different questions — next.

Ancestral states: marginal or joint?

Marginal

For one node, which state is most probable, summing over all states of every other node?

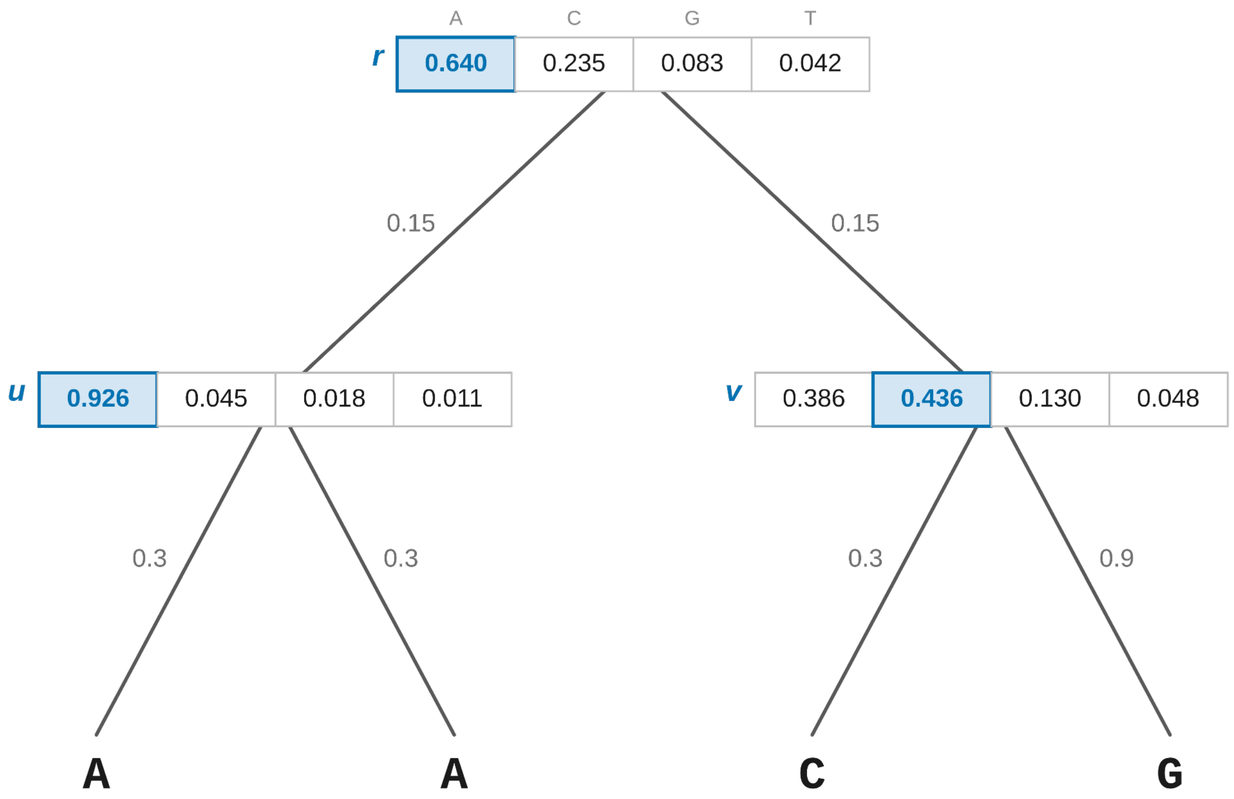
Joint

Which single assignment of states to all internal nodes is most probable?

At the root: $\Pr(\text{root} = a \mid \text{data}) = \pi_a \, L_{\text{root}}(a) / L$

- For any other node v, L_v sees only the subtree below v. Also need the **rest of the tree**: a second, top-down pass — or reroot at v (fine for a reversible model).
- Joint: the same DP with Σ replaced by max (Pupko et al., Mol. Biol. Evol. 2000) — Sankoff with probabilities.
- Marginal maximisation is exact, not greedy. It answers a *different question*, so the answers can disagree.

Marginal ≠ joint: an example



	r	u	v	Pr(leaves, r, u, v)
joint	A	A	A	1.52×10^{-3}
marginal	A	A	C	7.29×10^{-4}

- Posterior at v: C (0.436) beats A (0.386): C is favoured by v's own leaves.
- But A at v agrees with A at u and r, and that *combination* is the single most probable history.
- Stitching the marginal winners together gives a history **less than half as likely** (0.48×) as the joint optimum.

The model is part of the answer

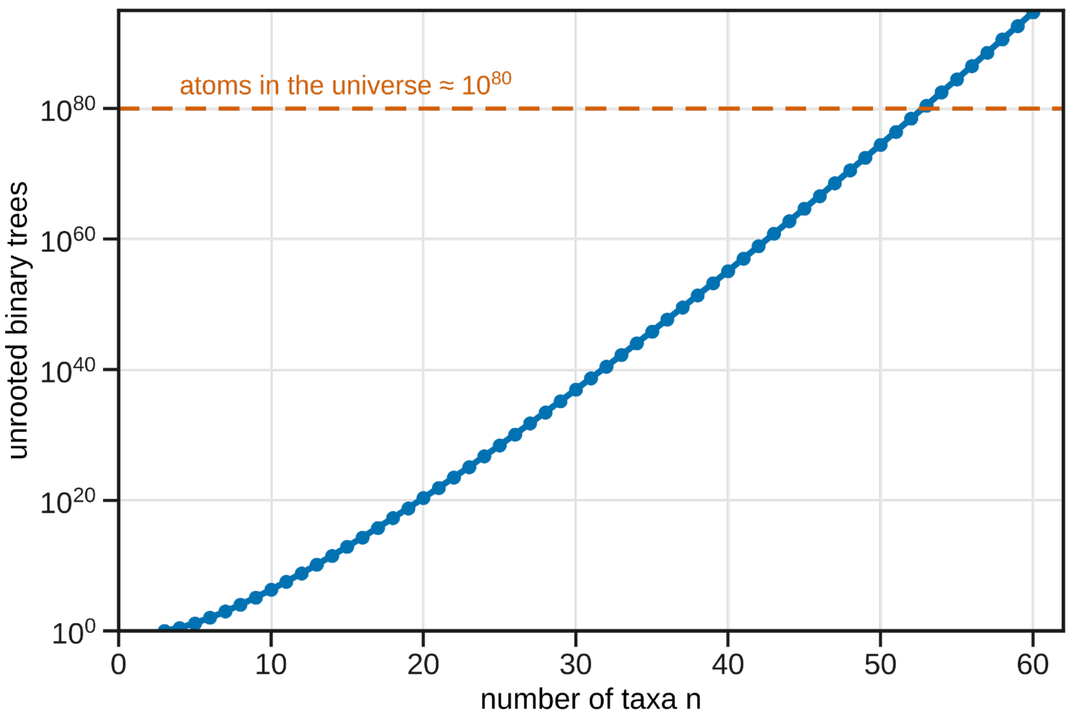
- Likelihood considers *all* histories, weighted by probability, and fits branch lengths. It is powerful, **but every answer is conditional on the model**.
- Wrong parameters? (rates, base frequencies) — estimate them too.
- Rates differ between sites? — add per-site rate categories (Γ).
- Sites not independent, or not Markovian? (RNA stems, codons, selection) — richer models, or accept the bias.
- Model choice is itself a statistical question (likelihood-ratio tests, AIC/BIC).

Distance methods

The large problem, part 1: trees from a distance matrix

Why trees – Parsimony – Likelihood – **Distance methods** – Tree search – Support

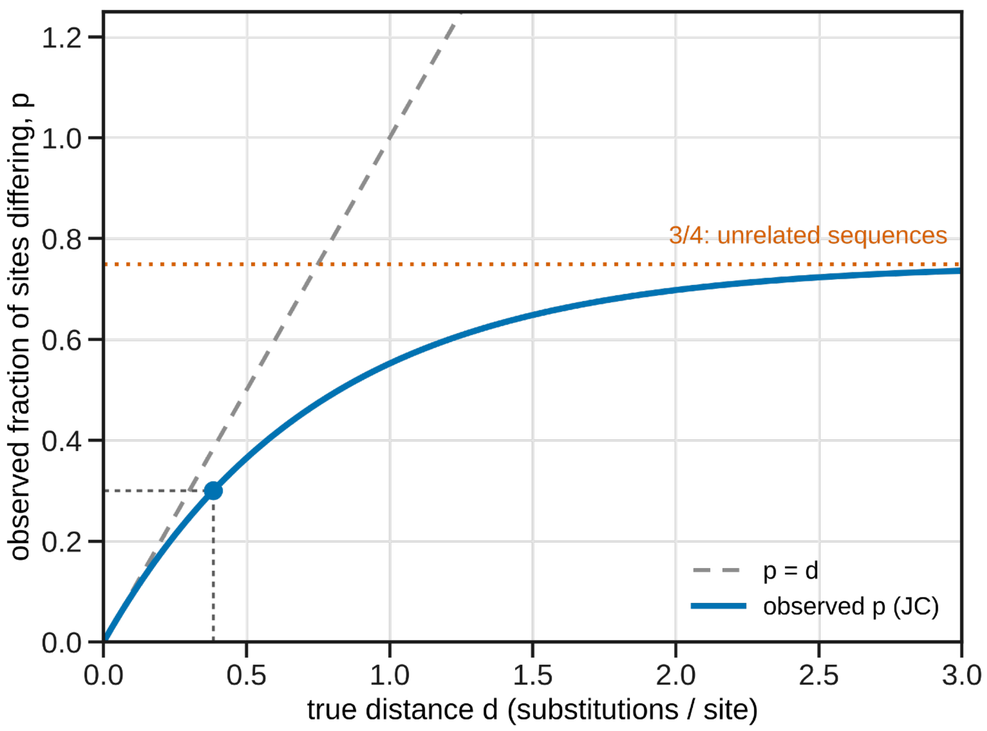
The large problem: there are too many trees



taxa n	unrooted binary trees
4	3
5	15
10	2,027,025
20	≈ 2.2×10^{20}
50	≈ 2.8×10^{74}

- $(2n-5)!! = 3 \cdot 5 \cdot 7 \cdots (2n-5)$: add leaves one at a time; the k-th new leaf can go on any of $2k-5$ edges.
- Exhaustive search is hopeless past ~12 taxa. Three strategies:
- **distances** → a fast algorithm (NJ); **optimise a score** by local search; **sample** trees (Bayesian MCMC).

From sequences to distances

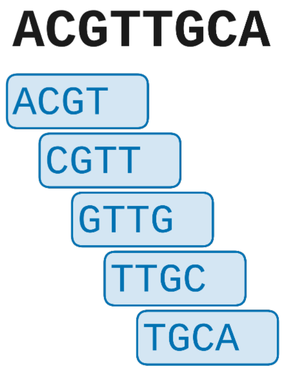


- **p-distance**: the fraction of aligned sites that differ. It *under-counts*: multiple hits at a site are invisible, and the expected p levels off at 3/4.
- **Jukes–Cantor correction**: invert the JC curve.

$$d = -3/4 \cdot \ln(1 - 4p/3)$$

- Example: p = 0.30 → d = 0.383 substitutions per site (the dot).
- d estimates the total branch length between the two sequences; it is undefined for p ≥ 3/4.

Alignment-free distances: k-mers and MinHash



the 4-mers of a sequence

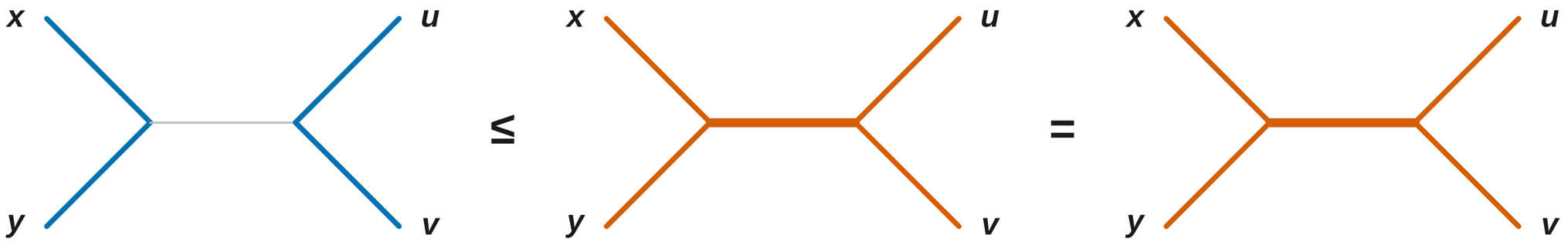
$J(A, B) = |A \cap B| \,/\, |A \cup B|$ for k-mer sets A, B

Mash distance $D = -(1/k) \cdot \ln(\, 2J \,/\, (1 + J)\,)$

- Treat a genome as its **set of k-mers**. No alignment needed. A substitution destroys up to k k-mers.
- D estimates the per-base divergence. Example: two 10 kb sequences, 1% substitutions, k = 21: J ≈ 0.68, D ≈ 0.010.
- **MinHash**: keep only the s smallest hash values of each set; the overlap of these sketches estimates J. Comparing two genomes costs O(s), not O(genome).
- Mash builds distance matrices for tens of thousands of genomes – then NJ makes the tree.

Ondov et al., Genome Biology 17 (2016)

When does a distance matrix come from a tree?



M(x,y) + M(u,v) **M(x,u) + M(y,v)** **M(x,v) + M(y,u)**

the internal edge counted twice

- M is **additive** if some tree with positive edge lengths has path length between i and j = M(i, j) for every pair.
- **Four-point condition** (Buneman 1971): M is additive iff every four taxa can be named so that the sums above hold.
- Then the tree is **unique**, and neighbor joining finds it exactly.

UPGMA: average-linkage clustering

UPGMA (Sokal & Michener, 1958)

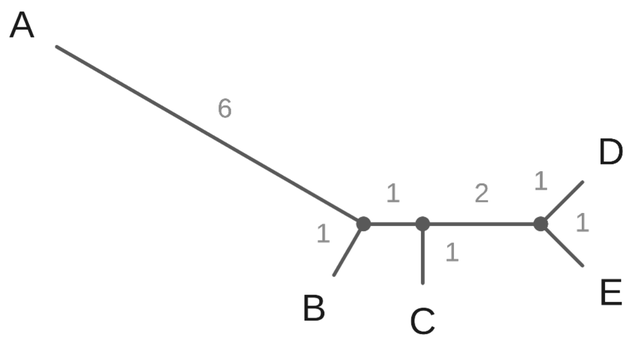
```
repeat until one cluster:
    merge the two clusters i, j with the smallest M(i, j)
    new cluster k = i ∪ j, placed at height M(i, j) / 2
    M(k, z) = ( |i| M(i, z) + |j| M(j, z) ) / ( |i| + |j| )
```

- Simple, $O(n^2)$ per merge, gives a rooted tree.
- It assumes a **molecular clock**: every leaf equally far from the root (an *ultrametric* M).
- If rates differ between lineages, the closest pair need not be neighbours in the tree — and UPGMA can return the **wrong topology**, even with perfect, additive distances.

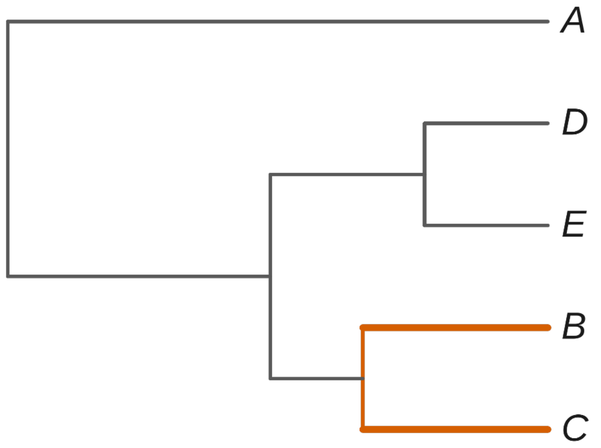
UPGMA fails on a perfect tree; NJ does not

	A	B	C	D	E
A	·	7	8	10	10
B	7	·	3	5	5
C	8	3	·	4	4
D	10	5	4	·	2
E	10	5	4	2	·

(a) true tree



(b) UPGMA: groups B with C



- A's branch is long (6); B is next to A. Yet B is *closer* to C (3) than to A (7), so UPGMA merges D, E, then **B with C** – a split not in the true tree.
- Neighbor joining on the same matrix returns tree (a) exactly, with every branch length.

Neighbor joining

$R(i) = \Sigma_k M(i, k)$ (total distance from i)

$Q(i, j) = (n - 2)\, M(i, j) - R(i) - R(j)$

join the pair i, j with the smallest Q into *a* new node u:

$M(i, u) = M(i, j)/2 + (R(i) - R(j)) / (2(n - 2))$, $M(j, u) = M(i, j) - M(i, u)$

$M(u, k) = (M(i, k) + M(j, k) - M(i, j)) / 2$ for every other k

repeat on $n - 1$ nodes; join the last two.

- Q rewards pairs that are **close to each other** but **far from everything else** – that corrects for unequal rates.
- If M is additive, NJ returns the true tree. Each step is $O(n^2)$, so $O(n^3)$ in all. Fast, and the default for big distance matrices.

Saitou & Nei, Mol. Biol. Evol. 4 (1987); Studier & Keppler (1988)

Neighbor joining, worked: step 1

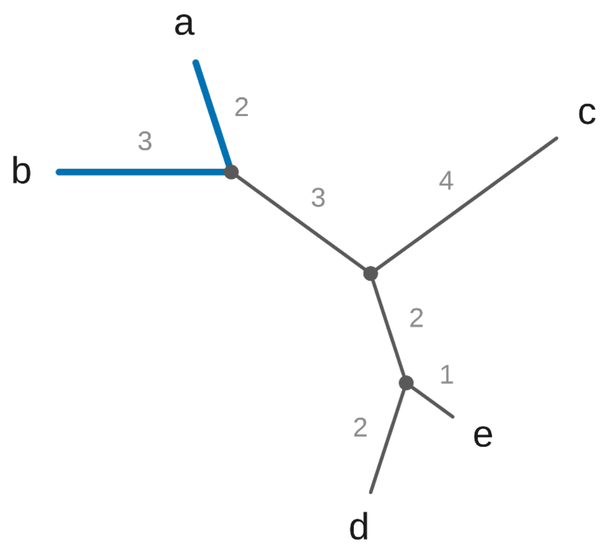
distances M

	a	b	c	d	e	R
a	·	5	9	9	8	31
b	5	·	10	10	9	34
c	9	10	·	8	7	34
d	9	10	8	·	3	30
e	8	9	7	3	·	27

$Q(i, j) = 3\,M(i, j) - R(i) - R(j)$

	a	b	c	d	e
a	·	**-50**	-38	-34	-34
b	**-50**	·	-38	-34	-34
c	-38	-38	·	-40	-40
d	-34	-34	-40	·	-48
e	-34	-34	-40	-48	·

- Smallest Q: (a, b), −50. Join them into u.
- M(a, u) = 5/2 + (31 − 34)/6 = **2**; M(b, u) = 5 − 2 = **3**.
- M(u, c) = (9 + 10 − 5)/2 = 7, M(u, d) = 7, M(u, e) = 6.



- Repeat: the next step ties (c, u) with (d, e) at −28; either choice gives the same final tree, shown here.
- Its path lengths reproduce M exactly – M was additive.

Example: Wikipedia, “Neighbor joining”

Tree search

The large problem, part 2: optimise a score over trees

Why trees – Parsimony – Likelihood – Distance methods – **Tree search** – Support

Maximum parsimony and maximum likelihood trees

Maximum parsimony

Find *a* tree T and sequences at its internal nodes minimising Σ over edges (u, v) of Hamming(s_u, s_v).

NP-hard (Foulds & Graham, 1982). Scoring one tree: Fitch, $O(nk|\Sigma|)$.

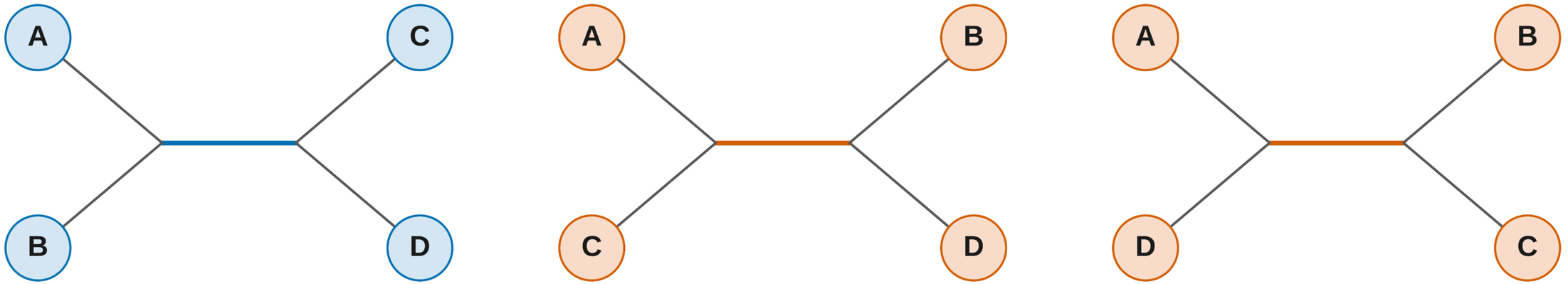
Maximum likelihood

Find *a* tree T and branch lengths maximising $\Pr(\text{alignment} \mid T, t, \text{model})$.

NP-hard (Chor & Tuller 2006; Roch 2006). Scoring one tree: pruning plus branch-length optimisation.

Scoring a tree is easy; **finding** the best one is hard. So: start somewhere, and improve by local moves.

Local search: nearest-neighbour interchange (NNI)



current tree: AB | CD

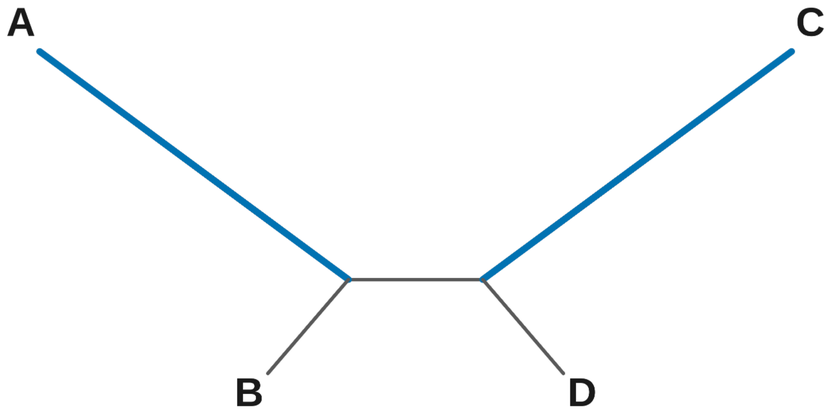
swap B and C: AC | BD

swap B and D: AD | BC

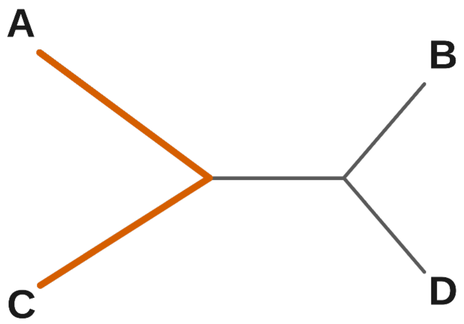
A, B, C, D are whole subtrees

- Pick an internal edge; swap one subtree across it. Each of the n – 3 internal edges gives 2 neighbours: **2(n – 3)** in all.
- Hill-climb: move to the best neighbour while the score improves. Stops at a **local** optimum.
- Bigger moves search wider: **SPR** (prune a subtree, regraft anywhere) and **TBR** (bisect, reconnect). Restart from many starting trees.

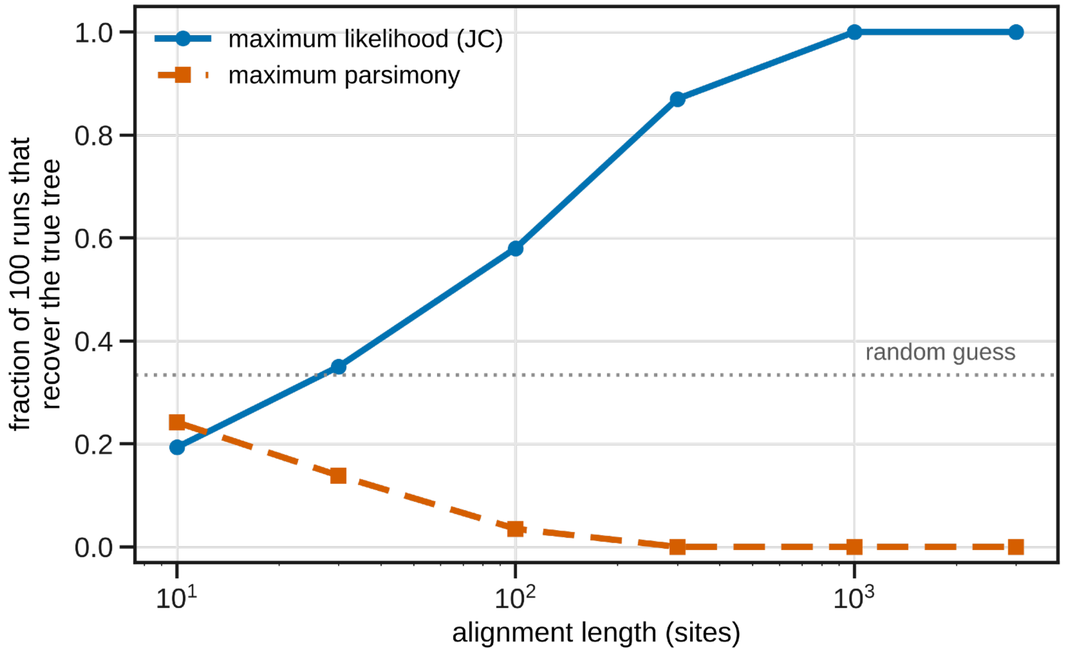
Long-branch attraction



true tree: AB | CD



parsimony's answer: AC | BD



- A and C sit on long branches (0.75); every other branch is 0.05. Parallel changes on the two long branches look like shared history.
- Parsimony becomes **more certain of the wrong tree** as data grow (statistically inconsistent). ML under the right model converges to the **true tree**.

Felsenstein, Syst. Zool. 27 (1978)

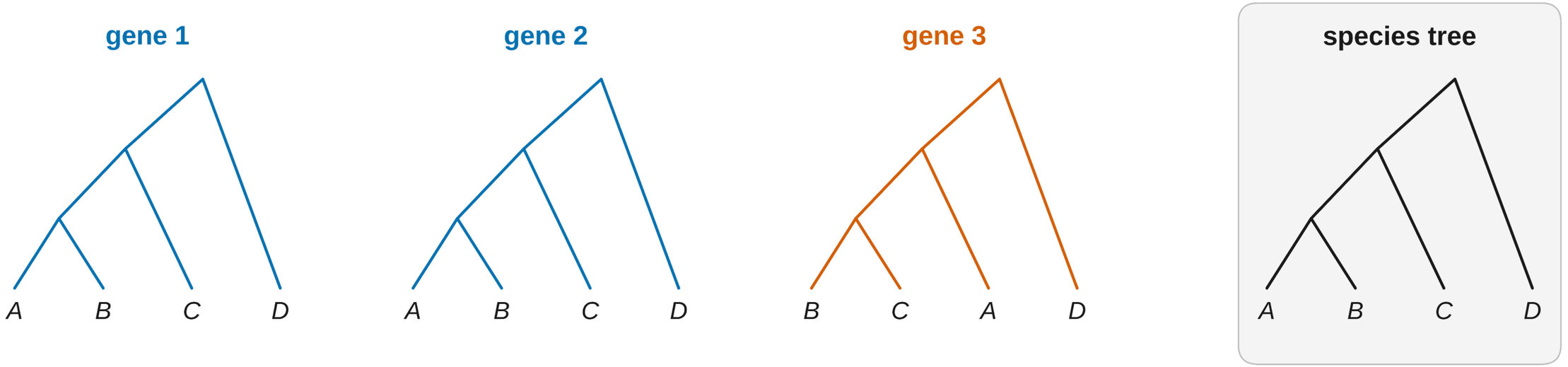
Bayesian inference: sample trees instead

Metropolis–Hastings MCMC

```
start at some tree T (with branch lengths and model parameters)
repeat many times:
    propose T' by a small random change (an NNI, a new branch length)
    accept T' with probability
        min( 1,  Pr(data | T') Pr(T')  /  Pr(data | T) Pr(T) )
    (times a proposal correction if moves are not symmetric)
```

- The walk is a Markov chain whose stationary distribution is the **posterior** Pr(T | data).
- After a “burn-in”, the fraction of samples with a given clade estimates that clade's posterior probability.
- Output: a **distribution** of trees, not one tree. MrBayes; BEAST for time-calibrated trees.

Phylogenomics: many genes, many trees



- Each gene has its own history. Gene trees can **disagree** with the species tree: incomplete lineage sorting (ancestral polymorphism), horizontal transfer, paralogy, or just noise.
- **Concatenation**: glue all genes into one alignment and infer one ML tree.
- **Summary methods**: infer a tree per gene, then find the species tree that agrees with the most gene-tree quartets (ASTRAL).

In practice

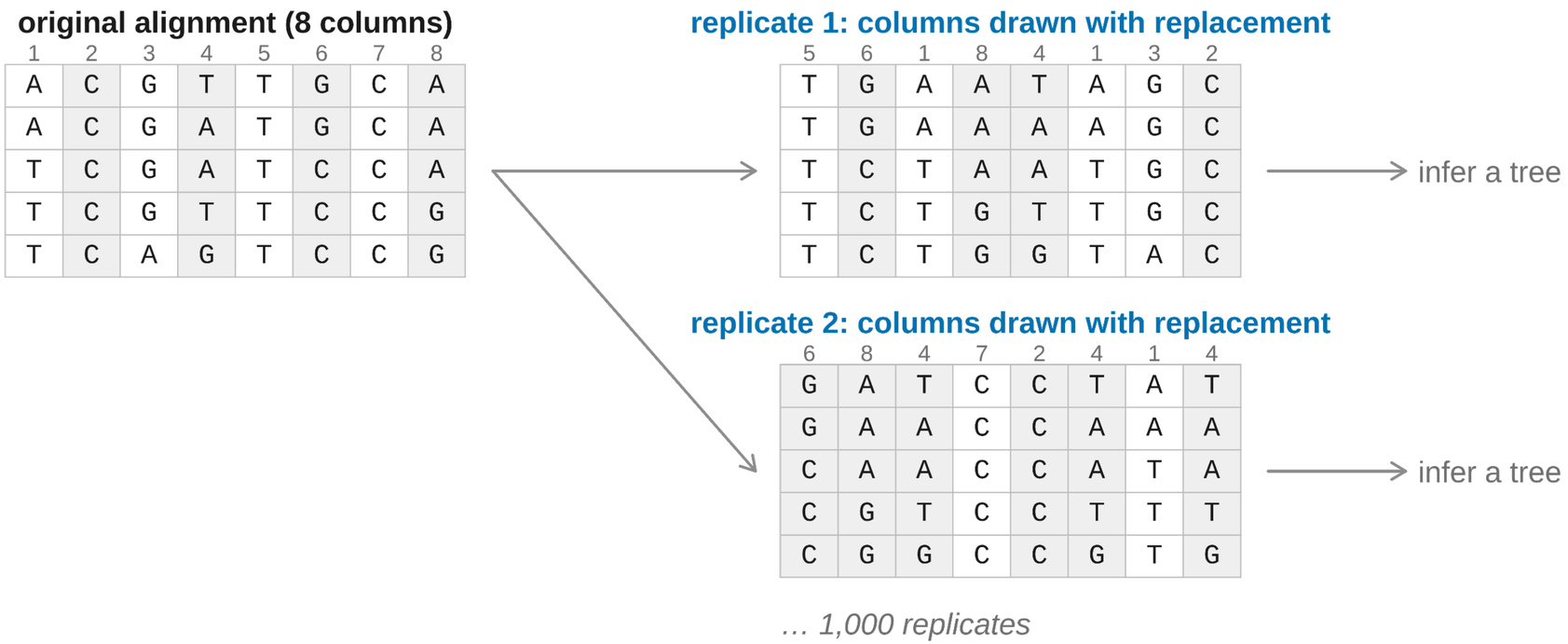
- **Maximum likelihood**: IQ-TREE (built-in model selection, ultrafast bootstrap), RAxML-NG; FastTree for very large alignments.
- **Bayesian**: MrBayes; BEAST for time-calibrated trees (epidemics).
- **Distances**: NJ on corrected or k-mer distances (Mash) when there are too many genomes to align.
- **Placement**: add new sequences to a fixed big tree. UShER places SARS-CoV-2 genomes on a tree of millions by *parsimony* — Fitch at pandemic scale (Turakhia et al., Nat. Genet. 2021).
- **Guide trees**: MSA tools build a quick NJ/UPGMA tree first, then align along it.

Support

How much should we believe each edge?

Why trees – Parsimony – Likelihood – Distance methods – Tree search – **Support**

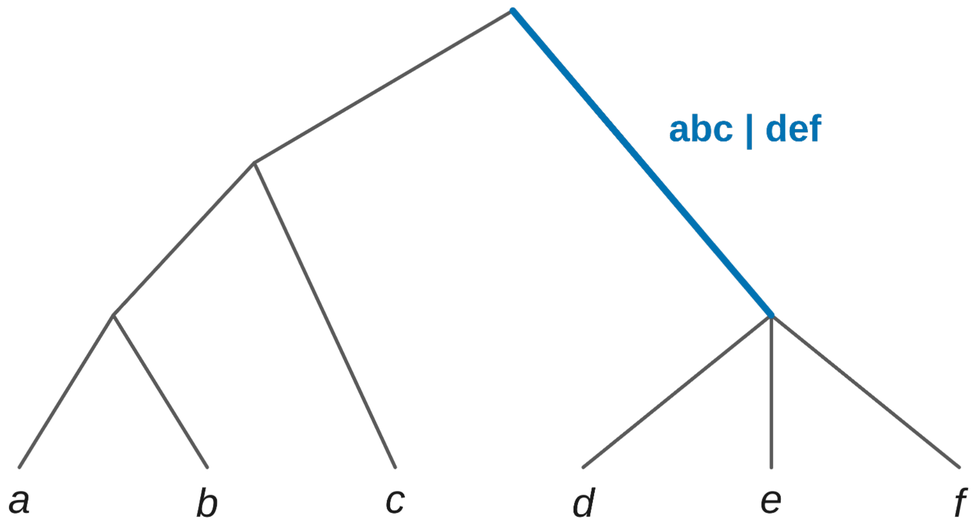
The bootstrap



- Support of an edge = % of replicate trees that contain it (the numbers on the H5N1 tree).
- It measures **stability under resampling**, not the probability that the clade is true.

Felsenstein, Evolution 39 (1985)

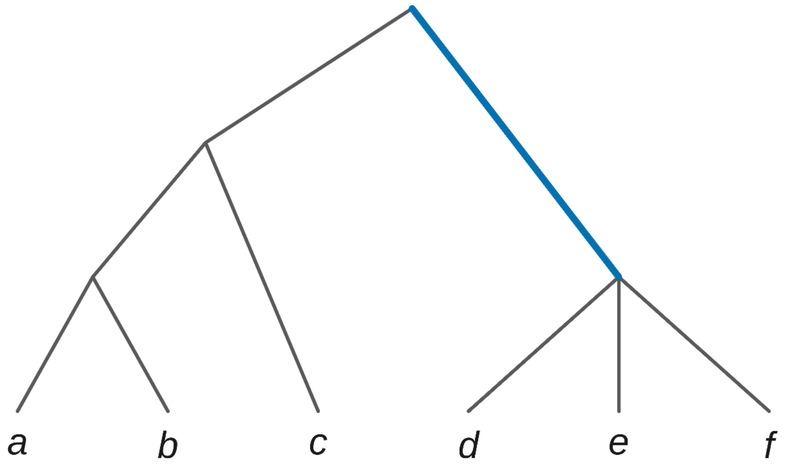
Edges are splits



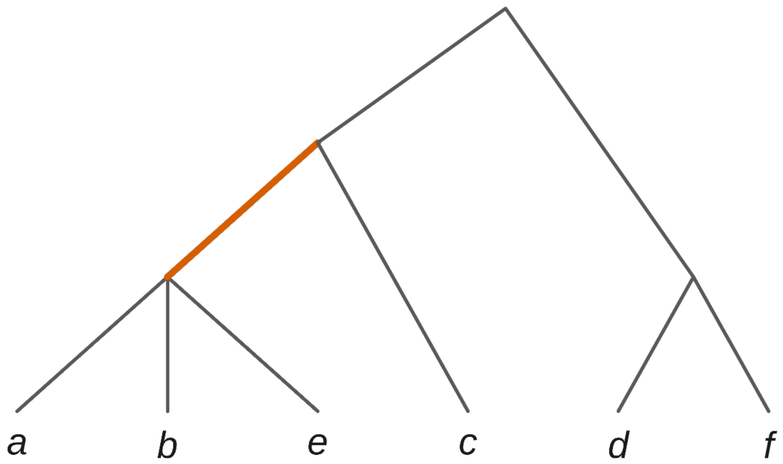
- Removing an edge cuts the taxa into two sides: a **split** (bipartition).
- This tree: {ab | cdef, abc | def} plus trivial splits (one leaf against the rest).
- A tree is determined by its set of splits. A binary unrooted tree on n taxa has n − 3 non-trivial splits (this one is not binary: d, e, f hang from one node).
- **Robinson–Foulds distance**: the number of splits found in exactly one of the two trees — the standard way to compare trees.

Compatible and incompatible splits

tree 1: abc | def



tree 2: abe | cdf

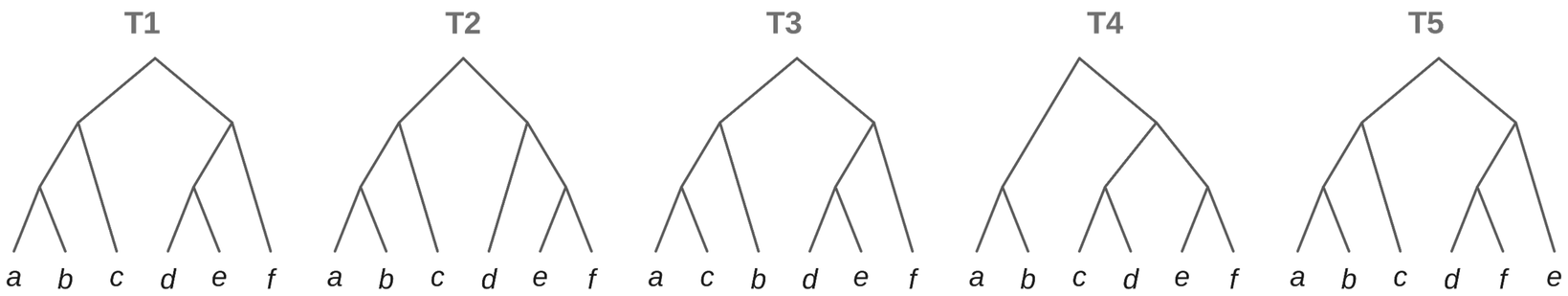


intersections

	abe	cdf
abc	ab	c
def	e	df

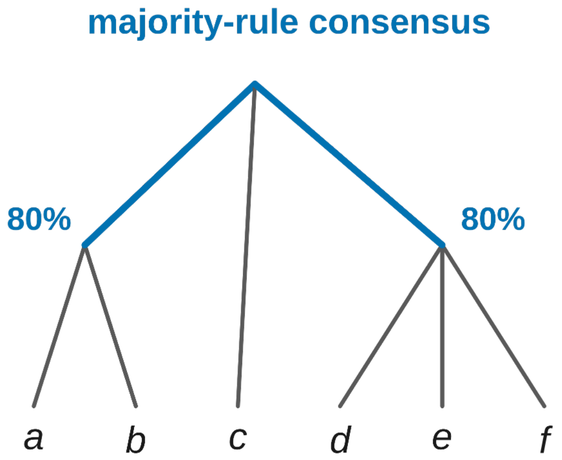
- Splits A | A′ and B | B′ are **compatible** (can sit in one tree) iff at least one of A ∩ B, A ∩ B′, A′ ∩ B, A′ ∩ B′ is empty.
- Here all four are non-empty (table): abc | def and abe | cdf are **incompatible**.
- Splits that are pairwise compatible always fit together in one tree (Buneman, 1971).

Majority-rule consensus



split	trees	
ab | cdef	4 / 5	**kept**
abc | def	4 / 5	**kept**
abcf | de	2 / 5	dropped
abcd | ef	2 / 5	dropped
ac | bdef	1 / 5	dropped
abef | cd	1 / 5	dropped
abce | df	1 / 5	dropped

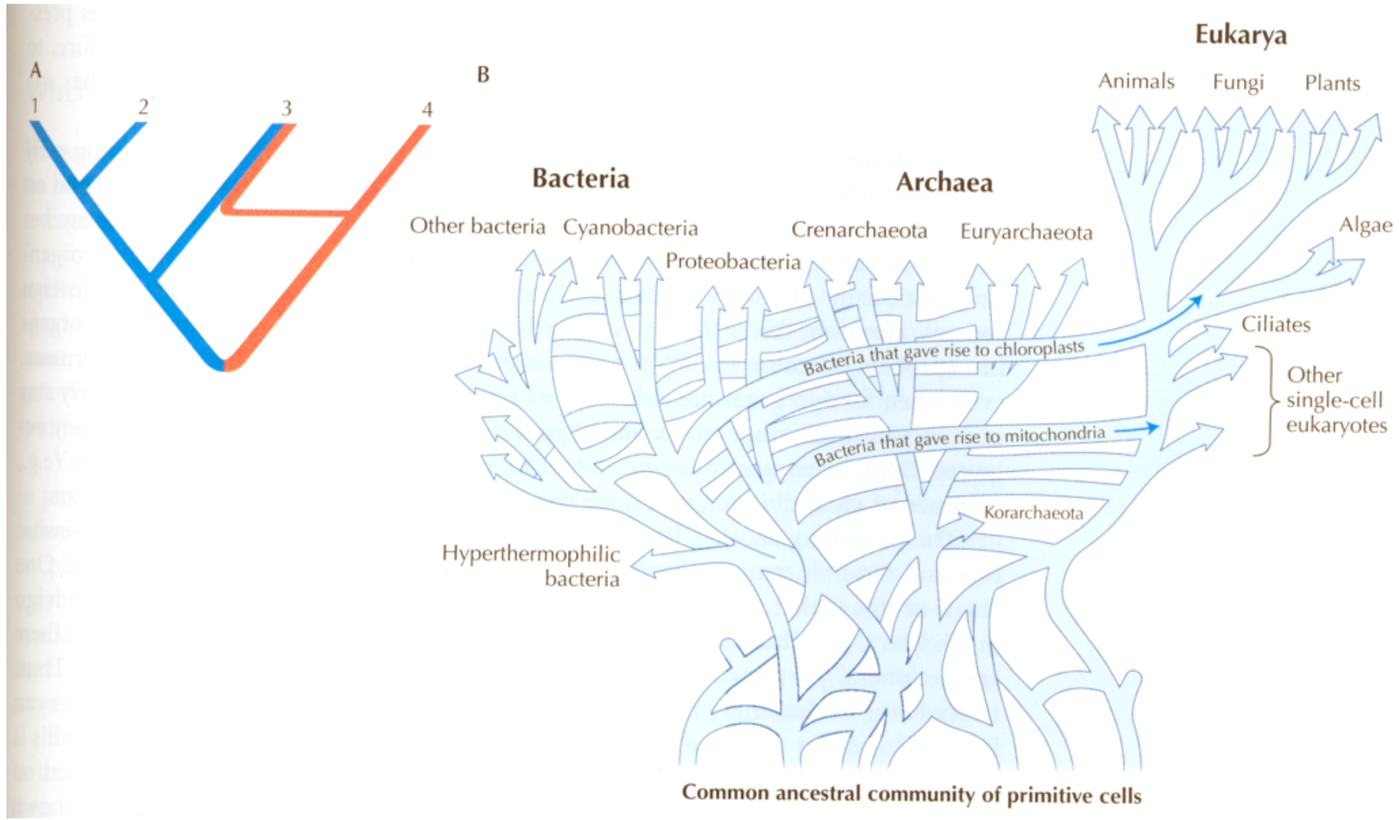
- Collect the splits of every tree (from bootstrap replicates, MCMC samples, or equally good trees) and count them.
- Keep the splits in **more than half** of the trees. Their frequency is the edge's support.
- Unresolved conflict becomes a **polytomy**: here d, e, f.



The majority-rule consensus always exists

- Let S be the set of splits that occur in more than half of the m trees.
- Take any two splits s, s′ ∈ S. Each is in more than m/2 trees, so by **pigeonhole** some tree contains both.
- Two splits in one tree are compatible. So the splits of S are **pairwise compatible**.
- Pairwise-compatible splits always form a tree (Buneman). ∎
- *The same argument fails at 50%: two incompatible splits can each appear in exactly half the trees.*

When a tree is the wrong picture



- **Horizontal gene transfer**: bacteria take up DNA from other species (plasmids, phages, uptake). Different genes, different trees.
- **Reassortment**: a host infected by two influenza strains can produce progeny with a mix of the segments; implicated in past pandemics.
- Then a **phylogenetic network** – a DAG, not a tree – is the better model.

Figure: W. F. Doolittle, Science 284 (1999)

Recap

- **Small problem, parsimony**: Fitch (unit costs), Sankoff (weighted); post-order DP + traceback, $O(nk|\Sigma|^2)$.
- **Small problem, likelihood**: a substitution model, Felsenstein's pruning — the same DP with $(\times, \Sigma)$. Marginal and joint ancestors differ.
- **Distances**: correct p for multiple hits (or use k-mers); NJ is exact on additive data, UPGMA needs a clock.
- **Large problem**: $(2n - 5)!!$ trees, NP-hard; local search (NNI/SPR), or sample trees by MCMC. Parsimony can be inconsistent.
- **Support**: bootstrap, splits, majority-rule consensus.