9-more-phylogenetics

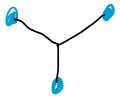
Tuesday, March 17, 2026 7:02 AM

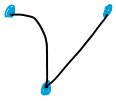
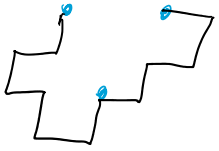
Last week we saw several different distance-based methods for constructing phylogenetic trees. These had the advantage of simplicity, but left out useful information, notably the identity of the mutations themselves.

Outline:

Maximum parsimony
Maximum likelihood
Consensus trees

Problem: Given N points in $\mathbb{R}^2$, construct a set of (overlapping) paths in $\mathbb{R}^2$ that connect them, and minimize the total length of the paths, but don't double-count overlaps.

Example: 

Non-minimum lengths:  

Exercise: Prove that the paths can be decomposed into a collection of line segments E. Let V be the set of all endpoints of E. Let $S = V - N$ be called Steiner points. Prove that $T = (V, E)$ is an (embedded) tree.

(Euclidean) Steiner tree: Given N points in $\mathbb{R}^2$, find a plane-embedded tree $T = (V, E)$ s.t. $N \subseteq V \subseteq \mathbb{R}^2$ and $\sum_{e \in E} \|e\|_2$ is minimized.
↑ Euclidean norm

Problem arises naturally (and historically) when one wants to build canals (or roads) between cities. Goal is to minimize construction cost.

Let's switch out from Euclidean space to Hamming space
↑ substitution distance b/t sequences.

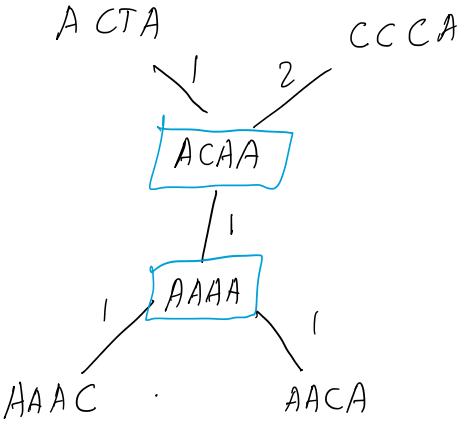
Maximum Parsimony: Given N taxa, represented as equal length strings of characters, find an unrooted phylogenetic tree that minimizes edge weights, where edges are weighted by Hamming distance between connected nodes. (note that we now have to assign sequences to internal nodes)

Example: ACTA

ACTA —2— CCCA —3— AAAC —2— AACA

Example:

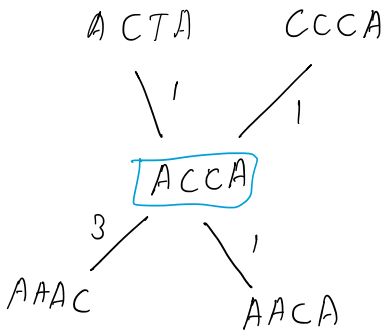
ACTA
CCCA
AAAC
AACA



parsimony score: 6

ACTA —2— CCCA —3— AAAC —2— AACA

parsimony score: 7



parsimony score: 6
(but not binary)

Scoring a proposed tree: walk up & down tree for each character

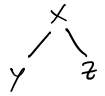
Say $S(x)$ is known state of x at a leaf.

(1) Root tree on arbitrary edge (so we have a binary tree)

(2) Let $A(x)$ be set of possible states for a character (e.g. ACGT for substitution variants)

(3) Set $A(x) = \{S(x)\}$ $\forall$ x leaves. (base case)

(4) Recursion formula:



$$A(x) = \begin{cases} A(y) \cap A(z) & \text{if } A(y) \cap A(z) \neq \emptyset \\ A(y) \cup A(z) & \text{else} \end{cases}$$

This recursion goes up from leaves to root.

(5) Walk back down tree, picking an arbitrary value of $A(r)$ at root as $S(r)$

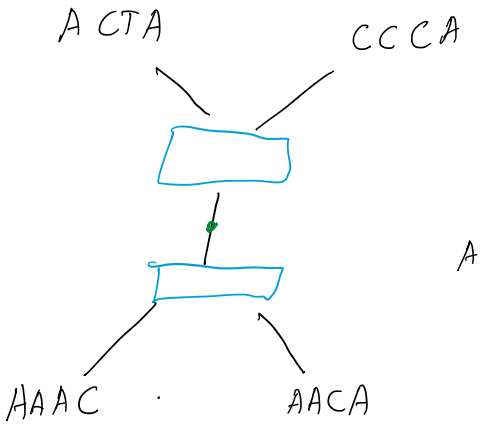
(6) Recursion formula:

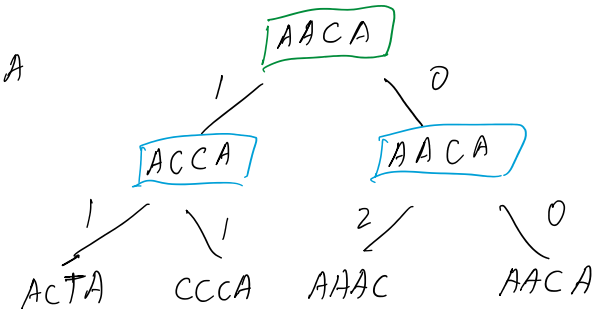
$$S(x) = \begin{cases} S(p) & \text{if } S(p) \in A(x) \\ \text{arbitrary } a \in A(x) & \text{otherwise.} \end{cases}$$

Notice that some nodes are forced into definite states but others have several possibilities

ACTA CCCA AACA

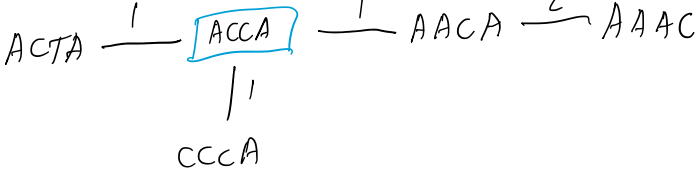
Example:





H more optimal score for this structure than above.

Better:



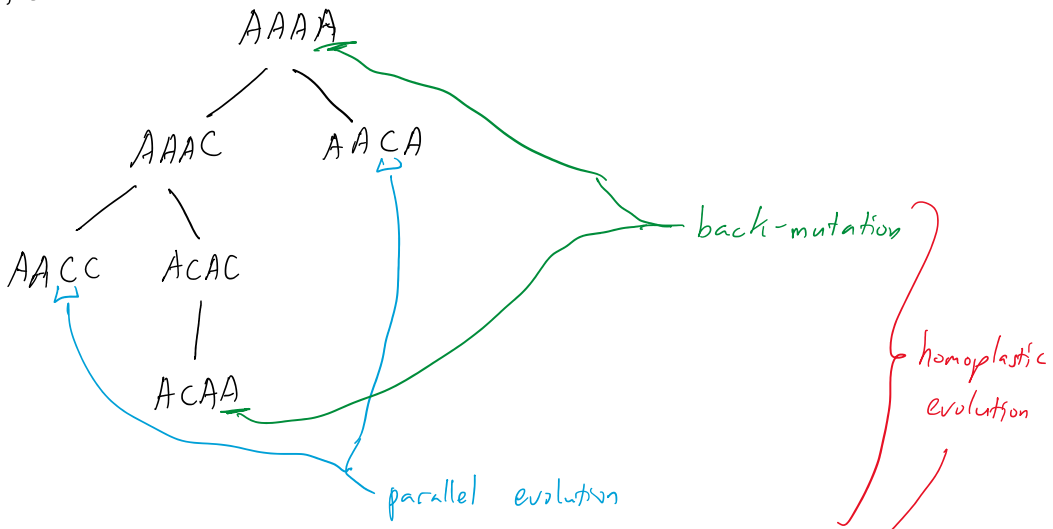
Given a candidate tree, it is fast to compute the parsimony score.
But, how do we find the right tree?

NP-hard problem.

- in practice, exact solutions only work for a few dozen taxa, so we need heuristics

---

Problem: Consider true tree:



Homoplasy is a problem because most parsimonious explanation is not real history.

the appearance of a state

explanation is not real history.

Also messes up distances.

↳ distant branches might accidentally look similar

the appearance of a state that already appears elsewhere in tree.

Exercise: Long branch attraction is a form of systematic error where distantly related species are (incorrectly) inferred to be closely related when they are on rapidly evolving branches. Explain how this might happen due to homoplasy, either for distance or parsimony-based methods.

Maximum likelihood estimation (MLE)

Suppose I flip a coin & it comes up heads 10 times in a row.
What is the chance the coin is fair? $P(\text{hypo} \mid \text{flips})$

If you are a Bayesian, you would need some prior distribution.

But, easy to answer question: how likely is it to see 10 heads IF coin was fair. $P(\text{flips} \mid \text{hypo})$

If coin fair, $\left(\frac{1}{2}\right)^{10} = \frac{1}{1024} \approx 0.001$.

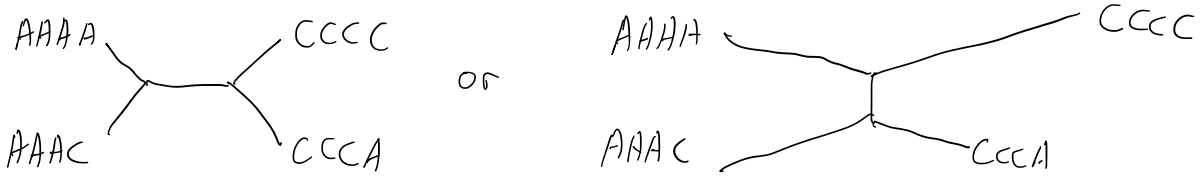
Alternate hypo: coin always comes up heads. Then $P(\text{flips} \mid \text{hypo}) = 1$

Hypo 3: coin is biased with $p_{\text{heads}} = 0.9$. Then $P(\text{flips} \mid \text{hypo}) = \left(\frac{9}{10}\right)^{10} \approx 0.349$

Idea behind MLE is that $P(\text{outcome} \mid \text{hypo})$ much easier to compute than $P(\text{hypo} \mid \text{outcome})$, so we do the former over a bunch of possible hypos and choose the hypo that maximizes likelihood.

Back to phylogenetics: Hypos are potential tree structures,

Ex.



Here, we have both topology & branch lengths as part of hypo.

We also need a model for how the sequences evolve with time.

Cavender-Farris: (for binary sequences).

At each site, 1 or 0 for each state.

Cavender-Farris: (for binary sequences)

At root, equal chance of 1 or 0 for each state.

Every edge e has a parameter $0 < p(e) < 0.5$ controlling the probability of flipping each bit.

} for simplicity, we assume mutation rate is the same.

Ex. Sequences 000, 001

Structure: 000 —(0.25)— 001 Likelihood: $0.75 \cdot 0.75 \cdot 0.25 = 0.140$

Hypo: 000 —(0.10)— 001 Likelihood: $0.9 \cdot 0.9 \cdot 0.1 = 0.081$

Hypo: 000 —(0.20)— 001 Likelihood: $0.8 \cdot 0.8 \cdot 0.2 = 0.128$

Hypo: 000 —(1/3)— 001 Likelihood: $\frac{2}{3} \cdot \frac{2}{3} \cdot \frac{1}{3} = 0.148$

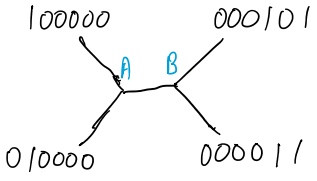
Exercise: Prove that the maximum likelihood branch length under the Cavender-Farris model and two sequences is $p(e) = \frac{M}{N}$, where M is the Hamming distance and N the total sequence length, so long as $\frac{M}{N} < 0.5$, and $0 < N < M$.

Can generalize to A, C, G, T in Jukes-Cantor Model or different nucleotide substitution rates

Notice: This model is reversible, in that the probability of going $A \to B$ or $B \to A$ are equal. Thus, we can arbitrarily assign a root & get the right probability.

Ex. Sequences 100000, 010000, 000101, 000011

Hypo (all edges length 1/6)



Brute force computation: (inefficient & people actually use tree-pruning)

Say A is root.

Likelihood: $\sum_{A \in \{0,1\}^6} \sum_{B \in \{0,1\}^6} P(A \to B)\, P(A \to 100000)\, P(A \to 010000)\, P(B \to 000101)\, P(B \to 000111)$

Felsenstein tree pruning: (only works if positions are independent)

- Look at a single character at a time.
- Recursively compute partial likelihoods of each node from children.

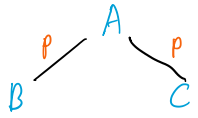
$L_n = (1,0)$, $L_i = (0,1)$ ← for known leaf values

- Recursively compute partial likelihoods of each node from children.

$L_0 = (1,0)$, $L_1 = (0,1)$ ← for known leaf values

likelihood 0 — likelihood 1

e.g. $L_0(0) = 1$ $L_0(1) = 0$

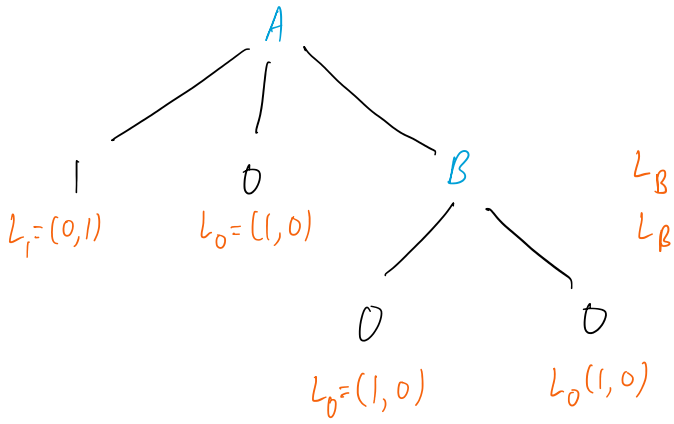


$$L_A(0) = \left[L_B(0)\underbrace{(1-p)}_{P(0|0)} + L_B(1)\underbrace{p}_{P(1|0)}\right]\left[L_C(0)(1-p) + L_C(1)p\right]$$

$$L_B(1) = \left[L_B(0)\underbrace{p}_{P(0|1)} + L_B(1)\underbrace{(1-p)}_{P(1|1)}\right]\left[L_C(0)p + L_C(1)(1-p)\right]$$

- Once we get to root, multiply its partial likelihoods by starting probabilities (e.g. uniform random 0/1) to get likelihood of tree for that character.
- Then multiply together likelihoods for each character together.

Ex. Take first char from above example: $p = \frac{1}{6}$ edge lengths



$$L_B(0) = (1-p)^2 = \frac{25}{36} \approx 0.69$$

$$L_B(1) = p^2 = \frac{1}{36} \approx 0.02$$

$$L_B = (0.69, 0.02)$$

$$L_A(0) = p(1-p)\left[0.69(1-p) + 0.02p\right] = 0.080$$

$$L_A(1) = (1-p)p\left[0.69p + 0.02(1-p)\right] = 0.018$$

$$L = L_A(0)\cdot\frac{1}{2} + L_A(1)\cdot\frac{1}{2} = 0.049$$

Assuming either char is equally likely at root

Exercise: Compute the runtime of Felsenstein tree pruning & the brute force method & compare how much faster it is

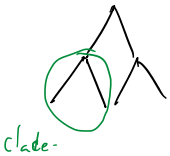
Problem: Given a proposed tree structure with both topology & branch lengths, we can compute likelihood. How do we optimize over trees?

NP-hard problem.

Tree Consensus

Recall that a rooted tree is defined by its clades (if all internal nodes have 2 kids)

What is the equivalent for unrooted trees?



clade

Define: A bipartition is the two disconnected sets of leaves that result from removing an edge in an unrooted tree.

Just like with clades & rooted trees, bipartitions fully define an unrooted tree if all internal nodes have deg 3.

Def. Let $C(T)$ be the bipartition encoding of an unrooted tree T given by breaking each edge in turn.

Def. A set A of bipartitions is compatible iff there is a tree T (with labelled leaves) in which $A \subseteq C(T)$

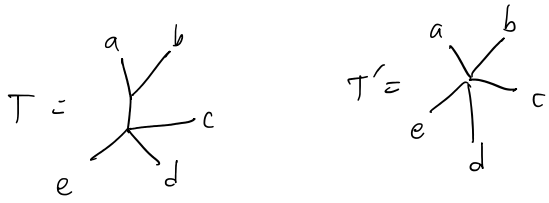
Fact: A set A of bipartitions is compatible iff every pair of bipartitions is pairwise compatible, where a pair $X=(X_1|X_2)$ & $Y=(Y_1|Y_2)$ is compatible iff at least one of the $X_i \cap Y_j = \emptyset$.

Note: Given two trees T & T' on the same sets of leaves, $C(T') \subseteq C(T)$ implies we can get T' from T be contracting edges.

T' is a contraction of T

T is a refinement of T'

Ex.



Exercise: Given a set A of bipartitions on the same leaf set, design an algorithm that determines if A is compatible, and if so, returns a tree T in which $A \subseteq C(T)$.

Hint: try turning A into a rooted tree first and get clades.

Def. Given a set $\{T_1, \ldots, T_k\}$ of unrooted trees on the same leaf-set, the strict consensus tree T is the tree where $C(T) = \bigcap C(T_i)$

Def. Given a set $\{T_1, \ldots, T_k\}$ of unrooted trees on the same leaf-set, the strict consensus tree T is the tree where $C(T) = \bigcap_i C(T_i)$.
(bipartitions must appear in all trees)

Def. The majority consensus tree T is the tree where $\forall x \in C(T)$, x appears in more than (strictly) half of $C(T_i)$.

Exercise: Prove that the bipartitions that appear in more than half the tree MUST be compatible.

Def. The greedy (or extended majority) consensus tree T starts from the majority consensus, and then repeatedly tries to add in bipartitions in order of most to least frequent occurence.
(might not be compatible.)

Consensus especially useful for parsimony, because often many maximally parsimonious trees.

Can be useful for MLE trees with random bootstrapping to get confidence intervals on the edges of the tree.