8-phylogenetics-distance

(credit to lecture materials from Tandy Warnow & Dannie Durand)
(thanks to Erin Molloy for advice)

Saturday, March 7, 2026 3:11 PM

What can we do with a collection of genomes?

One thing: look at potential consensus sequences. e.g. generate human genome

Ex
```
GATTACACCTTACCAG
GATTACAC-TTACGAG
 TTACATCTTACGAG
GATTACACCTTACGAG
```
← majority vote

Another thing: Use statistics to predict correlations with phenotype
└ GWAS, polygenic risk scores, etc.

Today: Phylogenetics: study of evolutionary relationships among organisms.

Recall: Kraken used taxonomic trees in its classification.
└ how can we construct these trees from a collection of genomes

---

Def. A tree is an undirected graph $G=(V,E)$ that is connected and has no cycles.

Notice: $|E| = |V| - 1$ in a tree.

Def. A rooted tree has a special root node, usually drawn at the top, giving an implicit orientation away from the root.

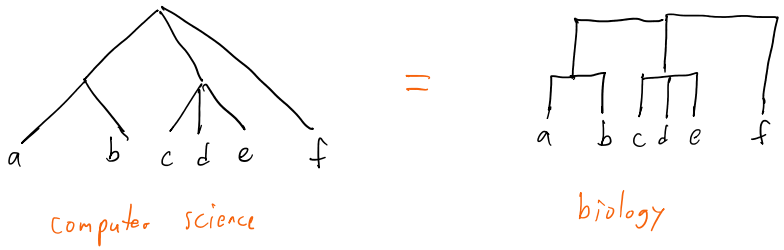
Def. A polytomy in an unrooted tree is a vertex with deg ≥ 4.
In an unrooted tree, a vertex with ≥ 2 children is also called a polytomy.

Def. A tree is binary if it does not have any polytomies.

Notice: this allows us to generalize binary trees to the unrooted case.

} we'll often assume that real phylogenetic trees are binary in evolution

Drawings: Two equivalent ways of drawing a (rooted) tree.
(we've only labeled leaves here)



computer science — biology

horizontal lines are NOT edges

Def. A clade is all the leaves descended from a common ancestor.
(mathematically equivalent to a subtree)

Example: $\{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{F\}, \{a,b\}, \{c,d,e\}, \{a,b,c,d,e,f\}$

Why are we focused so much on leaves? Because that's all we can directly measure.

Exercise: Prove that if two rooted trees have the same clades, and all their non-leaf nodes have at least two children, then they are the same tree.

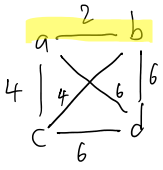
Exercise: Give an algorithm to generate a rooted tree from a set of clades. Can you do this for any sets of clades on a leaf set?

Def. A set A of subsets is compatible iff there is a rooted tree T (with labeled leaves) where $A \subseteq \text{Clades}(T)$.

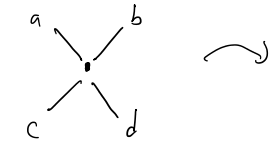
Distance-based tree estimation

Given a collection of genomes, how do we compute the phylogenetic tree?

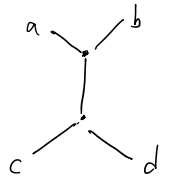
We can measure pairwise distances between leaves of the tree using genome seq. alignment.



pairwise distances

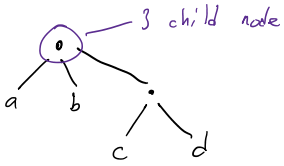


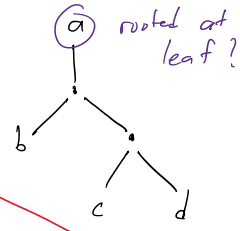
starting tree



unrooted binary tree

root via node



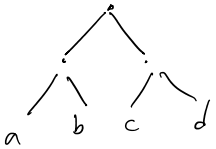


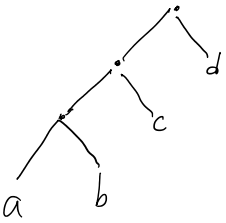
root via edge

Exercise: What kinds of genomic distances would work here?

put closest pair of genomes together?

rooting by picking up an edge is typically preferred because we get a rooted binary tree





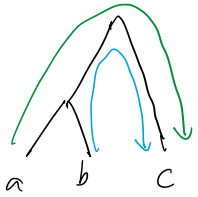
rooted binary tree

But how do we know where to put the root?

One idea: maybe there is some natural central edge such that somewhere on the edge is maximally far from every leaf.

Define: An ultrametric matrix is a matrix corresponding to distances between leaves of a rooted edge-weighted tree where the distance from the root to every leaf is constant.

Notice: In any ultrametric tree, given any three leaves a, b, c, the pairwise distances $\{d(a,b), d(b,c), d(a,c)\}$ cannot all be distinct. In fact, the 2 largest distances must always be the same. This is known as the **3-point condition.**



Fact: If a matrix satisfies the 3-point condition, it is ultrametric.

Ultrametricity happens when evolution is strongly clocklike.
e.g. if all genomes accumulate mutations at a constant rate.

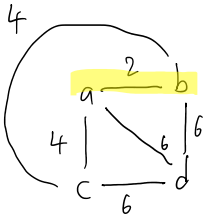
Can you think of times when this assumption fails?

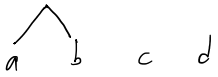
When this assumption holds, simple greedy algorithm works.

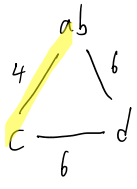
UPGMA: Unweighted Pair Group Method with Arithmetic mean ← usually called this
(of Agglomeration) ← sometimes called this

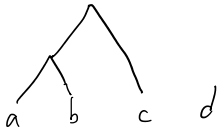
UPGMA is a bottom-up agglomerative clustering method that greedily merges nearest clusters, using arithmetic mean cluster distance.

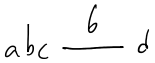
Initialize all leaves into their own cluster
Repeat until only one cluster:
↳ Calculate distances tween all pairs of clusters.
↳ Merge two nearest clusters. (Keep track of history of merges)

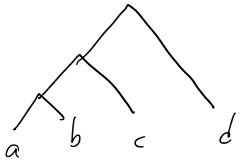












Exercise: What is the runtime of UPGMA? Can you come up with an $O(n^2 \log n)$ algorithm?

Exercise: Are there other ways to speed things up if you are given the promise that the matrix is ultrametric?

Problem: If the tree is not ultrametric, things might go wrong.

What's a way for things to go wrong?

Problem: If the tree is not ultrametric, things might go wrong.

What does it mean for things to go wrong?

Define: Given a generative model $\mathcal{G}$ for evolving sequences of length k, a method M for estimating the phylogenetic tree is statistically consistent if $\forall \varepsilon > 0$, $\exists$ a constant k' s.t. the probability M returns the correct tree is $\geq 1 - \varepsilon$ for $k > k'$.

Intuitively, if genomes are long enough, we recover the correct tree.
(obviously, a genome of length 1 is too short for ANY method)

Key: A generative model $\mathcal{G}$ to evaluate correctness, because otherwise we don't have any ground truths.

Analogy: Biased coin, and determining if bias is towards heads.

Statistically consistent estimator: flip k times, and see if more heads.

"Bad" estimator: Vote-counting meta analysis of p-values ← don't ever do this!

Study: Flip coin 100 times, and see if $p < 0.05$ for hypothesis that coin is biased towards heads.

Repeat this experiment (e.g. in different labs around the world) k times.

If a majority of labs say the coin is biased towards heads, say it is biased towards heads.

Bad case: Suppose $\text{Prob}(\text{heads}) = 0.52$. Correct answer is YES.

In order to reject null hypo that coin is not biased toward heads, need X heads out of 100.

Solve $\text{Prob}(\text{Binom}(100, 0.5) \geq X) = 0.05$

$\Rightarrow$ Need $X \geq 59$ to have p-value of < 0.05.

But $\text{Prob}(\text{Binom}(100, 0.52) \geq X) \approx 0.096$.

So each study only has a 9.6% chance of saying YES.

As $k \to \infty$, only 9.6% of studies say YES, so the meta-analysis will say NO, so this is not statistically consistent.

Why? The strict p-value throws away weak evidence of YES before aggregation.

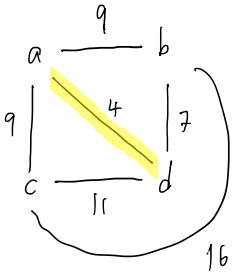
↳ don't pool filtered data

Pool raw data instead!

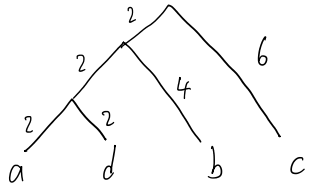
Exercise: Come up with statistically consistent and inconsistent estimators for the biased coin probability, rather than just the YES/NO question.

(statistically consistent here means that the estimated probability converges to the true probability as $k \to \infty$)

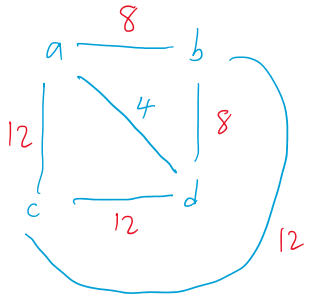
Going back to UPGMA: This is statistically consistent when the generative model is strongly clock-like, but not otherwise.



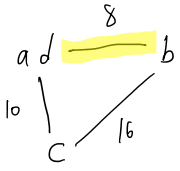
UPGMA:

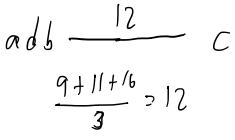


tree distances →



does not match original distances!





Obtained wrong distances because tree is not ultrametric.

Notice: For any unrooted binary tree T, tree distances satisfy

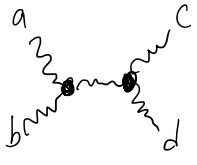
$\forall\ a, b, c, d$, consider

$s_1 = d(a,b) + d(c,d)$

$s_2 = d(a,c) + d(b,d)$

$s_3 = d(a,d) + d(b,c)$.

WLOG,



This is the 4-point condition

Then $s_1 < s_2 = s_3$.

↳ since middle path taken

$\rightsquigarrow$ = path of ≥ 1 edges.

(clearly, some pairs must be closer in tree hops)

We say T induces the quartet tree $ab|cd$.

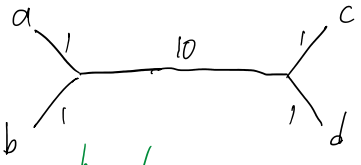
Def. A matrix of leaf distances is **additive** if it satisfies the 4-pt condition

The 4-pt method allows us to split any 4 leaves into two sets of two leaves each.

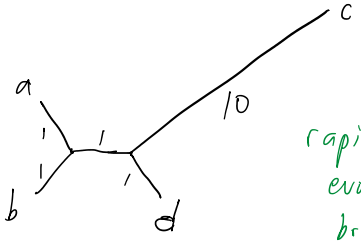
We'll come back this this later.

Neighbor joining: Greedy agglomerative correction to UPGMA taking into account correction factor for branches of tree that have faster mutation rates

Intuition:



shared ancestry



rapidly evolving branch

Instead of picking two nearest neighbors, adjusted pairwise distance subtracts the "average distance" to all other nodes.

$$Q_{i,j} = d(i,j) - (r_i + r_j) \quad \text{where} \quad r_i = \frac{1}{n-2} \sum_{k \neq i} d(i,k).$$

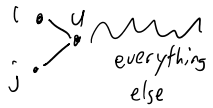
why $n-2$? has to do with # times we traverse from i or j to common junction node.

c in the right example above is far from everything, so subtracting that out correctly shows that it is actually close to d.

Intuition for $n-2$: Let's multiply everything by $(n-2)$.

$$(n-2)Q_{i,j} = (n-2)d(i,j) - \sum_{k \neq i} d(i,k) - \sum_{k \neq j} d(j,k)$$

If i,j are to be clustered, then there is a node u connecting them as "siblings"



$$\Rightarrow \quad d(i,j) = d(i,u) + d(j,u)$$

$$(n-2)Q_{i,j} = (n-2)d(i,u) + (n-2)d(j,u) - \sum_{k \neq i} \underbrace{[d(i,u) + d(u,k)]}_{n-1 \text{ choices of } k} - \sum_{k \neq j} \underbrace{[d(j,u) + d(u,k)]}_{n-1 \text{ choices of } k}$$

The $(n-2)d(i,u)$ cancel out all but one copy of $d(i,u)$ in the sum

Similarly for $(n-2)d(j,u)$.

$$= -d(i,u) - \sum_{k \neq i} d(u,k) - d(j,u) - \sum_{k \neq j} d(u,k)$$

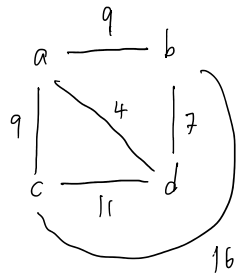
$$= -\sum_k d(u,k) - \sum_k d(u,k) = -2\sum_k d(u,k)$$

↳ minimizing this is equivalent to maximizing the distance of u from all the leaves

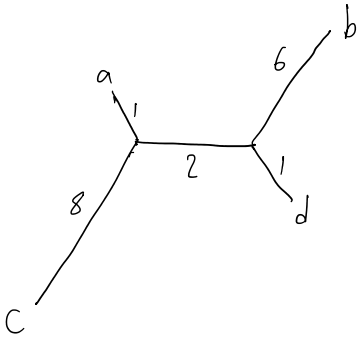
$\Rightarrow$ we are finding the internal node that is furthest away from the leaves and clustering together its two leaves.

If the distance matrix is (exactly) additive, this happens to be guaranteed to reconstruct the (unrooted) tree

Exercise: Work out by hand the neighbor joining algorithm on



Correct solution is:



Exercise: It would be convenient to use $\tilde{r}_i = \frac{1}{n}\sum_{k=1}^{n} d(i,k)$ instead. Show that this fails.

Back to quartets: Instead of taking a greedy agglomerative approach, we can also build a collection of quartets via the 4-pt method, and then combine them.

All Quartets Method: Say on leaves S.

Compute all quartets Q. (assume more than 1, else return T = quartet)
Find pair of taxa i,j that are always grouped together in all quartets ← If no such pair, return No compatible tree.
(Merge i,j) / remove i. ← equivalent as far as quartets go.
Recursively compute T' on $S - \{i\}$
Return $T = \{$insert i as a sibling of j in $T'\}$

Naive Quartets Method: Relax computation of quartets with relaxed 4-pt condition. (for non-additive case with errors)

↳ Given taxa a, b, c, d, group into pairs to minimize pairwise sum $S_1 = d(a,b) + d(c,d)$
$S_2 = d(a,c) + d(b,d)$

{ If additive, we have $S_1 < S_2 = S_3$

Given taxa a, b, c, d, ...

pairwise sum

$s_1 = d(a,b) + d(c,d)$

$s_2 = d(a,c) + d(b,d)$

$s_3 = d(a,d) + d(b,c)$

If additive, we have $s_1 < s_2 = s_3$

Here we just take $s_1 < s_2 < s_3$.

Still might end up with incompatible quartets.

Alternate idea: Generate trees that maximize compatibility with quartets. (since every unrooted tree implies set of quartets)

NP-hard, but many good heuristics used in practice.

Next time: Maximum parsimony, maximum likelihood, consensus trees.