Trees & Biological Data

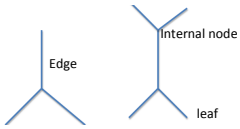
Alex Krasnitz
Center for Quantitative Biology

Trees are a way of organizing biological data

- Phylogeny: tree of life
  - Paralogues
  - orthologues
- Clustering: group objects based on features
- Classification: can we identify cell types by their expression profiles?

Binary, unrooted, rooted

Consider only binary trees.

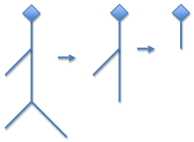


We can make one leaf special and call it a root.

Node & edge counting

For n leaves and a root, how many internal nodes?

(n=3)



Generally, n-1

For n leaves and a root, how many edges?

For each leaf & internal node, 1 edge pointing to the root

➔ $2n - 2$ edges (not counting the root edge)

Counting (cont'd)

An <u>unrooted</u> tree with n leaves can be obtained from a <u>rooted</u> tree with n-1 leaves by making the root an extra leaf and its edge. Since doing so does not change the number of internal nodes,

➔ $(n - 1) - 1 = n - 2$ internal nodes

Counting the leaves,

➔ $2n - 2$ nodes in total

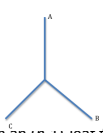
The former root edge is now counted:

➔ $2(n - 1) - 2 + 1 = 2n - 3$ edges

How many labeled trees?

Let $N(n)$ be the number of labeled unrooted trees with n leaves

Clearly, $N(3) = 1$



Construct an n-leaf tree from an (n-1)-leaf tree by adding a leaf.

There are $2(n - 1) - 3 = 2n - 5$ edges where a leaf can be added.

$N(n) = (2n - 5)\, N(n - 1)$ ➔ $N(n) = (2n - 5)!!$

Quiz: how many labeled rooted trees with n leaves?

Distance trees & hierarchical clustering

Idea: organize objects as leaves on a tree.
Objects: species, genes, cells.
Described by: sequence and/or other features.
More similar objects should be closer on a tree.
➔ Introduce dissimilarity d_{ij}
To make d_{ij} a distance:

$d_{ii} = 0$; $d_{ij} = d_{ji}$; $d_{ij} + d_{jk} \geq d_{ik}$ (triangle inequality)

Distance between sequences

Let f be the fraction of sites where the 2 sequences differ.

$0 \leq f \leq 1$.

For 2 identical sequences $f = 0$.
For 2 unrelated sequences $f = ¾$.

Jukes-Cantor distance: $d = -¾ \ln(1 - 4f/3)$
0 for identical sequences; ∞ for unrelated sequences.
Proportional to the time it takes 2 sequences to diverge.

Hierarchical Clustering: Linkage (agglomeration rule)

A prescription to define dissimilarity between composite objects (groups of objects).

For 2 groups of objects, a & b

$$D_{ab} = \min_{i \in a, j \in b} d_{ij} \qquad D_{ab} = \max_{i \in a, j \in b} d_{ij}$$

(simple linkage) (complete linkage)

$$D_{ab} = \frac{1}{C_a C_b} \sum_{i \in a, j \in b} d_{ij}$$

(Average, or Unweighed Pair Group Method Using Arithmetic Averages (UPGMA))

HC: Linkage (cont'd)

Exercise: prove that if group *b* is obtained by joining groups p and q, then for UPGMA

$$D_{ab} = \frac{D_{ap}C_p + D_{aq}C_q}{C_p + C_q}$$

Similarly, for simple linkage:

$$D_{ab}=\min(D_{ap},D_{aq})$$

For maximal linkage:

$$D_{ab}=\max(D_{ap},D_{aq})$$

These are update rules for matrix *D* .

Hierarchical clustering

With a linkage defined, we can grow a tree hierarchically:

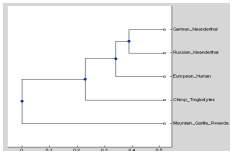
Initialize: dissimilarity matrix $d = D$ $(n\times n)$

Iterate while dim(D) > 1×1:

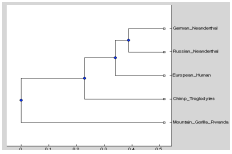
find the minimal off-diagonal element of D and the corresponding p, q

update D using the update rule

Height of a node is ½ the distance between children



Hierachical clustering (cont'd)



Is this what really happened? Is this the tree of life for hominidae?

Assume that this is the true tree and that the distances among leaves are measured along the edges.

Ultrametric property: In the absence of inversions, two of the distances along edges between any 3 leaves are exactly equal and the third is no greater than these.

Jukes-Cantor *d* ~ time, constant mutation rate ➔ ultrametric *d*.

If d is ultrametic and UPGMA is used as linkage, the tree will be reconstructed correctly.

Neighbor joining algorithm

What if the sequences do not all evolve at the same rate and/ or are not observed at the same time (fossils)?

We can (re)construct a tree such that the distance between any two leaves equals the sum of connecting edges (additivity).

Any 4 leaves in an additive tree can be labeled *i,j,k,l* such that

$d_{ij}+d_{kl} \leq d_{ik}+d_{jl} = d_{il}+d_{jk}$

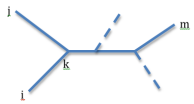
(a 4-point condition)

Neighbor joining algorithm (cont'd)

If the 4-point condition holds for all combinations of 4 leaves, there is an underlying additive tree from which the $n \times n$ distance matrix d_{ij} comes. But how can we find the tree?

Suppose that *i* and *j* are neighbors on the tree, with a parent node *k* ➔ drop *i* & *j* in favor of *k*: for any other leaf *m* set

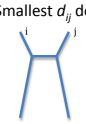
$d_{im} = d_{ik} + d_{km}$
$d_{jm} = d_{jk} + d_{km}$
➔ $d_{km} = ½(d_{im} + d_{jm} - d_{ij})$



The new $(n\text{-}1)\times(n\text{-}1)$ distance matrix still comes from a tree (is additive)!

Neighbor joining algorithm (cont'd)

How do we find neighbors? Smallest d_{ij} does not guarantee that *i* & *j* are neighbors.



But smallest $D_{ij} = d_{ij} - r_i - r_j$ does!
(NJ theorem, w/o proof [show for n=4])

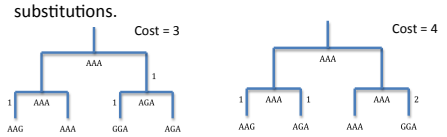
$$(r_i = \frac{1}{n-2}\sum_k d_{ik})$$

NJ algorithm:
Use *D* to find 2 neighboring leaves *i* & *j*. Prune *i* & *j*.
Their parent *k* becomes a leaf with distances $d_{km} = ½(d_{im} + d_{jm} - d_{ij})$
Repeat as long as there are > 2 leaves in the tree.

Parsimony

Alternative approach to phylogeny. Instead of growing a tree find a tree (among many) that best "explains" its *n* leaves. Leaves must be sequences of characters.

Example: 4 aligned sequences AAG, AAA, GGA, AGA. For a given tree with these as leaves, find ancestral sequences (for the internal nodes and the root) that require fewest substitutions.



Parsimony (cont'd)

Ideally:

(a) Enumerate all (labeled) trees for a given set of leaves (sequences).

(b) Find the minimal cost (number of substitutions) for each tree.

(c) Select the tree(s) with the smallest overall minimal cost.

Approach to (b): find the minimal cost for each site in the sequence and sum over sites.

A slight generalization: make cost dependent on the substitution (weighted parsimony). E.g., $S(\mathrm{A,C}) \neq S(\mathrm{G,T})$.

Parsimony (cont'd)

Recursive solution for cost minimization:

Consider a labeled tree with residues $x_1,\ldots x_n$ given at each leaf.
Let $S_k(a)$ be the minimal cost of placing a at node k.
Start at the root: $k = 2n - 1$.
Compute $S_k(a)$ recursively:

If k is a leaf, set $S_k(a = x_k) = 0$, $S_k(a \neq x_k) = \infty$.

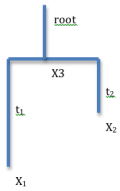
Else compute $S_i(b)$, $S_j(b)$ for all b at daughters i, j and set

$$S_k(a) = \min_b(S_i(b) + S(a,b)) + \min_b(S_j(b) + S(a,b)).$$

Terminate: minimal cost of tree = $\min_a S_{2n-1}(a)$.

Probabilistic phylogeny

Another character-based method. The idea is similar to that of parsimony: choose the "best" among the many possible trees with a given set of leaves. "Best" ➔ the most likely tree, assuming we know the nucleotide substitution rates



The probability of observing leaf nucleotides x_i given tree T (branching pattern, edge lengths t_i and ancestral nucleotides):

$P(x_1,x_2,\ldots \mid T,t_1,t_2\ldots x_3,\ldots)$ = product over edges

Leaves are the data, T is a model. Turn P into likelihood.

Edge lengths are free parameters ➔ use to maximize P

Edge probabilities

Jukes-Cantor base substitution probability: $s(t)=¼\,(1 - e^{-4\alpha t})$

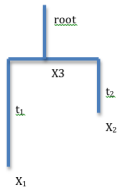
No change of base: $r(t)=¼\,(1 + 3e^{-4\alpha t})$

Example: Assuming that x_1,x_2,x_3 above are all different,

$P(x_1,x_2 \mid T,t_1,t_2,x_3) = (\,¼\,)^2\, s(t_1)\, s(t_2)$

Probabilistic phylogeny (cont'd)

If xi are sequences of length > 1, assume that each site evolves independently ➔P is a product over sites.



Example: x_1 = GC, ,x_2 = GG, x_3 = GT

$P(x_1,x_2 \mid T,t_1,t_2,x_3) = (\,¼\,)^4\, s(t_1)\, s(t_2)\, t(t_1)\, r(t_2)$

Maximize *w.r.t.* t_1 and t_2: $e^{-4\alpha t} = ½$, or $t_{1,2} = (\ln 2)/(4\alpha)$

Ancestral sequences (x_3 here) usually are unknown ➔ average over them, with equal probabilities for all sequences.