Phylogenetic Tree Construction

- **In constructing a phylogenetic tree, the taxa we wish to relate are usually ones currently living.**
- **We have information, such as DNA sequences, from the terminal taxa and no information from the ones represented by internal vertices.**
- **We do not even know which internal vertices should exist, because we do not yet know the tree topology.**
- **Distance methods attempt to build tree using information that we believe describes the total distances between terminal taxa along the tree.**

Let's try to find evolutionary relationship between four species S1, S2, S3, and S4

- choose a particular orthologous gene from their genomes and align the sequences

- compute Jukes-Cantor distances between each pair of sequences.

These are our estimates of distances along the tree.

	S_1	S_2	S_3	S_4
S_1		.45	.27	.53
S_2			.40	.50
S_3				.62

Methods for Phylogeny

- **Distance Methods**
  - UPGMA
  - Fitch-Margoliash Algorithm
  - Neighbour-Joining Algorithm
- **Character-based Methods**
  - Maximum Parsimony Methods
  - Maximum Likelihood Methods

Distance Methods

- **Uses the number of changes between pairs of sequences in a group to construct a tree**
- **Sequences with fewest changes are neighbours, *i.e.* they share a node to which they are joined by a branch**
- **Aim is to position neighbours correctly and to compute branch lengths that best fit the data**

Tree Construction: UPGMA

Unweighted Pair-Group Method with Arithmetic Means (UPGMA)

- is the simplest method for tree construction

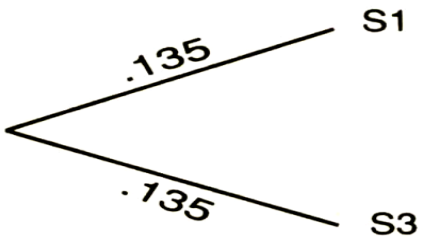
- assumes that the rate of change along the branches of a tree is constant, i.e., it assumes a molecular clock.

This method produces a rooted tree.

Tree Construction: UPGMA

Table: Distances Between Taxa

	S_1	S_2	S_3	S_4
S_1		.45	.27	.53
S_2			.40	.50
S_3				.62



Step -1: pick the two closest taxa, S1 and S3, .27 distance apart.

Draw the edges, each of length 0.27/2 = 0.135

i.e., the edges are drawn equidistant from the common ancestor.

UPGMA

Combine S1 & S3 into a group, compute distance of remaining sequences from this group, e.g., distance between S1-S3 and S2 is

(.45 + .40)/2 = .425

	S_1	S_2	S_3	S_4
S_1		.45	.27	.53
S_2			.40	.50
S_3				.62

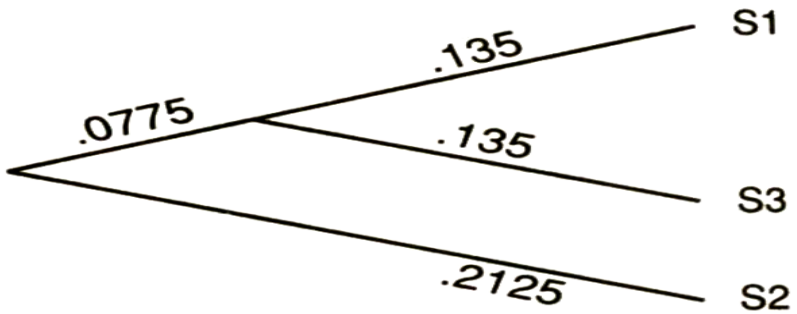
The table then collapses to:

Table: Distances between Groups: UPGMA Step-1

	S1-S3	S2	S4
S1-S3		.425	.575
S2			.50

UPGMA

Step-2: Repeat the process, using the collapsed table. Because the closest taxa and/or groups in the new table are S1-S3 and S2, which are .425 apart:



Edge to S2 will have length .425/2 = .2125, while the other new edge will be .2125 - .135 = .0775

	S1-S3	S2	S4
S1-S3		.425	.575
S2			.50

UPGMA

Step-3: Again combining taxa, we form a group S1-S2-S3, and compute its distance from S4 as: (.53 + .5 + .62)/3 = .55

	S1-S2-S3
S4	.55

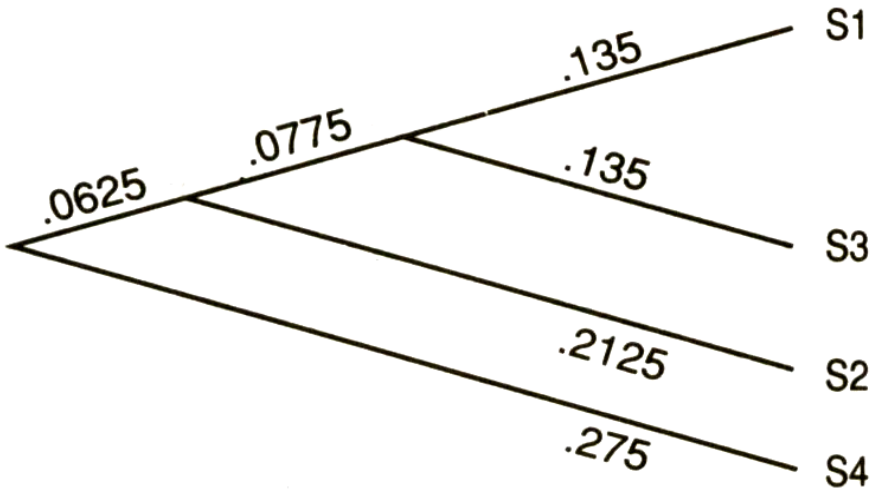
	S_1	S_2	S_3	S_4
S_1		.45	.27	.53
S_2			.40	.50
S_3				.62

Final tree is drawn by estimating S4 as .55/2 = .275 from the root.

The other edge has length 0.275 – 0.2125 = .0625, since that places all other taxa .275 from the root as well.

UPGMA

	S_1	S_2	S_3	S_4
S_1		.45	.27	.53
S_2			.40	.50
S_3				.62



Does the constructed tree exactly fit the data?

Distance on the tree from S3 to S4 =.55, while according to the original data, it is.62!

Tree constructed for the data does not exactly fit the data.

However, the tree distances are reasonably close to the distances given by the data

UPGMA

Note that the molecular clock assumption is implicit in UPGMA.

In this example, when we placed S1 & S3 at the ends of equal length branches, we assumed that the amount of mutation each underwent from their common ancestor was equal.

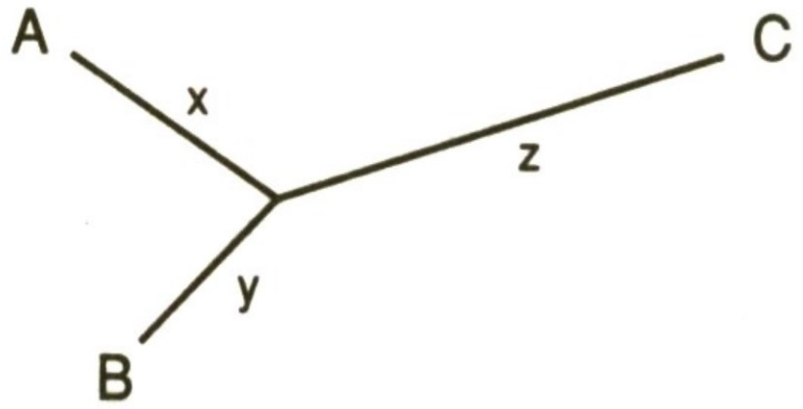
UPGMA always places all the taxa at the same distance from the root, so that the amount of mutation from the root to any taxon is identical.

Fitch-Margoliash Algorithm

More complicated than UPGMA, but builds on the same basic approach.

It attempts to drop the molecular clock assumption of UPGMA.

First, let's put 3 taxa on an unrooted tree:



Distance data defined as:

$x + y = d_{AB}$

$x + z = d_{AC}$

$y + z = d_{BC}$

For 3 taxa, we can assign lengths to the edges to fit data exactly

Fitch-Margoliash Algorithm

These equations can be solved to give

$$x = (d_{AB} + d_{AC} - d_{BC})/2$$

$$y = (d_{AB} + d_{BC} - d_{AC})/2$$

$$z = (d_{AC} + d_{BC} - d_{AB})/2$$

- 3-point formula for fitting taxa to a tree

Fitch-Margoliash algorithm uses the 3 taxa case to handle more taxa.

Fitch-Margoliash Algorithm

	S1	S2	S3	S4	S5
S1	-	0.31	1.01	0.75	1.03
S2	-	-	1.00	0.69	0.90
S3	-	-	-	0.61	0.42
S4	-	-	-	-	0.37
S5	-	-	-	-	-

As in UPGMA, choose the closet pair of taxa to join (S_1 & S_2 in this case)

Fitch-Margoliash Algorithm

Step – 1:

- Join S_1 & S_2 **without** placing them at an equal distance from a common ancestor

  - reduce to 3-taxa case by combining all other taxa into a group (i.e., group S_3-S_4-S_5)

- Compute the distance of S_1 and S_2 from the group as the average of their respective distances from S_3, S_4, and S_5

Fitch-Margoliash Algorithm

Distance from S_1 to S_3-S_4-S_5:

$d(S_1, S_3\text{-}S_4\text{-}S_5) = (1.01+.75+1.03)/3 = 0.93$

Distance from S_2 to S_3-S_4-S_5:

$d(S_2, S_3\text{-}S_4\text{-}S_5) = (1.00+.69+.90)/3 = 0.863$

This gives us the table:

	S1	S2	S3-S4-S5
S1	-	0.31	0.93
S2	-	-	0.863

Fitch-Margoliash Algorithm

Fitting the data in the table to obtain the tree using 3-point formula:

$x + y = dS_1S_2$, $x + z = dS_1G$,

$y + z = dS_2G$, G: S_3-S_4-S_5

Then,

$x = (dS_1S_2 + dS_1G - dS_2G)/2$

$= (0.31 + 0.93 - 0.863)/2 = 0.1885$

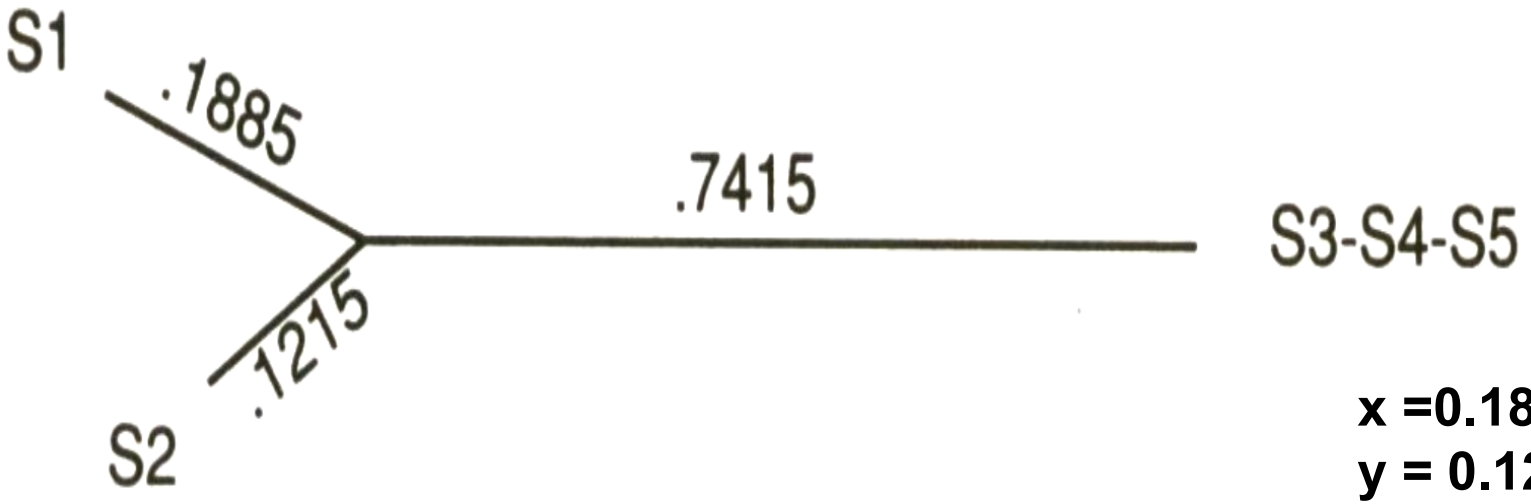
$y = (dS_1S_2 + dS_2G - dS_1G)/2$

$= (0.31 + 0.863 - 0.93)/2 = 0.1215$

$z = (dS_1G + dS_2G - dS_1S_2)/2$

$= (0.93 + 0.863 - 0.31)/2 = 0.7415$

Fitch-Margoliash Algorithm



x =0.1885
y = 0.1215
z = 0.7415

FM algorithm: Step 1

Note: S1 & S2 are not equidistant from the internal node

Also note that x + y = 0.31 = dS_1S_2

Fitch-Margoliash Algorithm

Step–2:

- Keep only edges ending at S_1 & S_2 and return to original table
- Join S_1 & S_2 into a group, compute distances of remaining taxa from the group S_1-S_2:

Distance from S_3 to S_1-S_2:

$$d(S_3, S_1\text{-}S_2) = (1.01+1.00)/2 = 1.005$$

Distance from S_4 to S_1-S_2:

$$d(S_4, S_1\text{-}S_2) = (0.75+0.69)/2 = 0.72$$

Distance from S_5 to S_1-S_2:

$$d(S_5, S_1\text{-}S_2) = (1.03+0.90)/2 = 0.965$$

Fitch-Margoliash Algorithm

On collapsing S1-S2 into a group:

	S1-S2	S3	S4	S5
S1-S2	-	1.005	0.72	0.965
S3	-	-	0.61	0.42
S4	-	-	-	0.37

Again look for the closest pair.

Fitch-Margoliash Algorithm

Step–3:

- Join the closest pair S_4 & S_5
- Compute the distances of S_4 & S_5 from a single temporary group S_1-S_2-S_3:

$$d(S_4, S_1\text{-}S_2\text{-}S_3) = (.75+.69+.61)/3 = 0.683$$

$$d(S_5, S_1\text{-}S_2\text{-}S_3) = (1.03+.90+.42)/3 = 0.783$$

This gives us the table:

	S1-S2-S3	S4	S5
S1-S2-S3	-	0.683	0.783
S4	-	-	0.37

Fitch-Margoliash Algorithm

Applying the 3-point formula to the table:

$x + y = dS_4S_5$, $x+z = dS_4G$, $y+z=dS_5G$, $G:S_1\text{-}S_2\text{-}S_3$

$x = (dS_4S_5 + dS_4G - dS_5G)/2$

$= (0.37 + 0.683 - 0.783)/2 = 0.135$

$y = (dS_4S_5 + dS_5G - dS_4G)/2$

$= (0.37 + 0.783 - 0.683)/2 = 0.235$

$z = (dS_4G + dS_5G - dS_4S_5)/2$

$= (0.683 + 0.783 - 0.37)/2 = 0.548$

Fitch-Margoliash Algorithm

x =0.135
y = 0.235
z = 0.548

FM algorithm: Step 2

Fitch-Margoliash Algorithm

Keep the edges joining S_4 & S_5, discarding the edge leading to the temporary group S_1-S_2-S_3.

So far we have joined two groups, S_1-S_2 & S_4-S_5.

Next, compute a new table containing these two groups:

$$d(S_1\text{-}S_2, S_4\text{-}S_5) = (0.75+1.03+0.69+0.90)/4 = 0.8425$$

$$d(S_3, S_4\text{-}S_5) = (0.61+0.42)/2 = 0.515$$

From step-2, $d(S_1\text{-}S_2, S_3) = 1.005$.

	S1-S2	S3	S4-S5
S1-S2	-	1.005	0.8425
S3	-	-	0.515

Fitch-Margoliash Algorithm

Applying the 3-point formula again:

$x + y = dG_1S_3$, $x+z = dG_2S_3$, $y+z = dG_1G_2$,

G_1: S_1-S_2 G_2: S_4-S_5.

$x = (dG_1S_3 + dG_2S_3 - dG_1G_2)/2$

$= (1.005+0.515-0.8425)/2 = 0.33875$

$y = (dG_1S_3 + dG_1G_2 - dG_2S_3)/2$

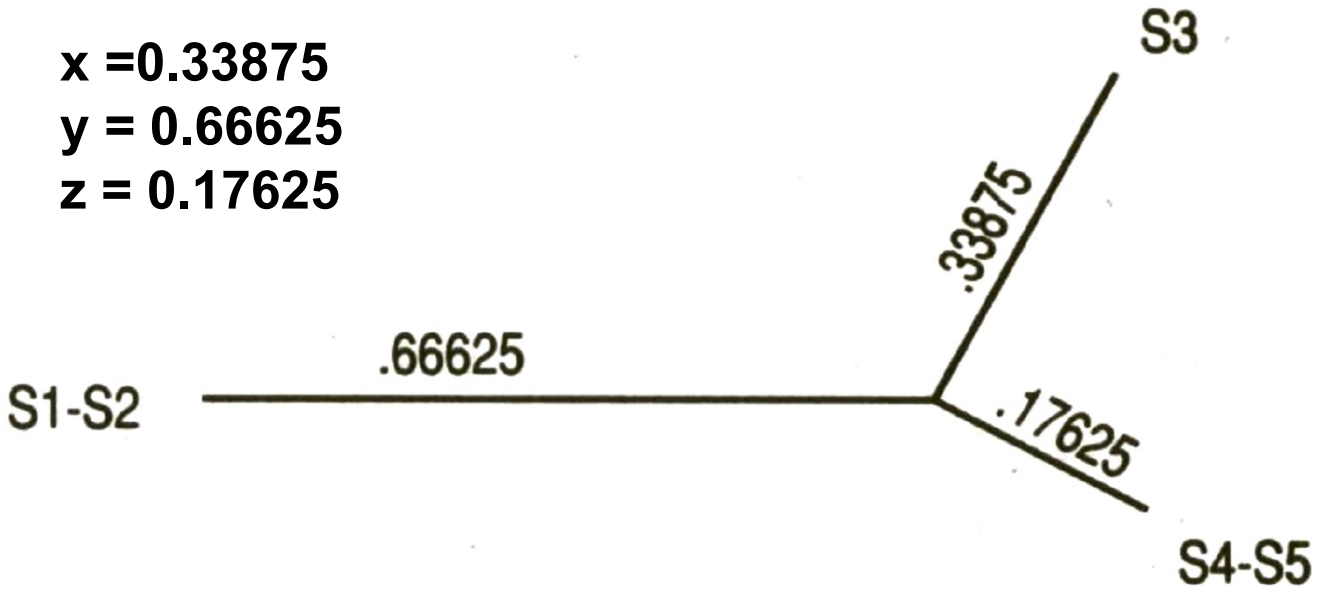
$= (1.005+ 0.8425 - 0.515)/2 = 0.66625$

$z = (dG_2S_3 + dG_1G_2 - dG_1S_3)/2$

$= (0.515+ 0.8425 - 1.005)/2 = 0.17625$

Fitch-Margoliash Algorithm

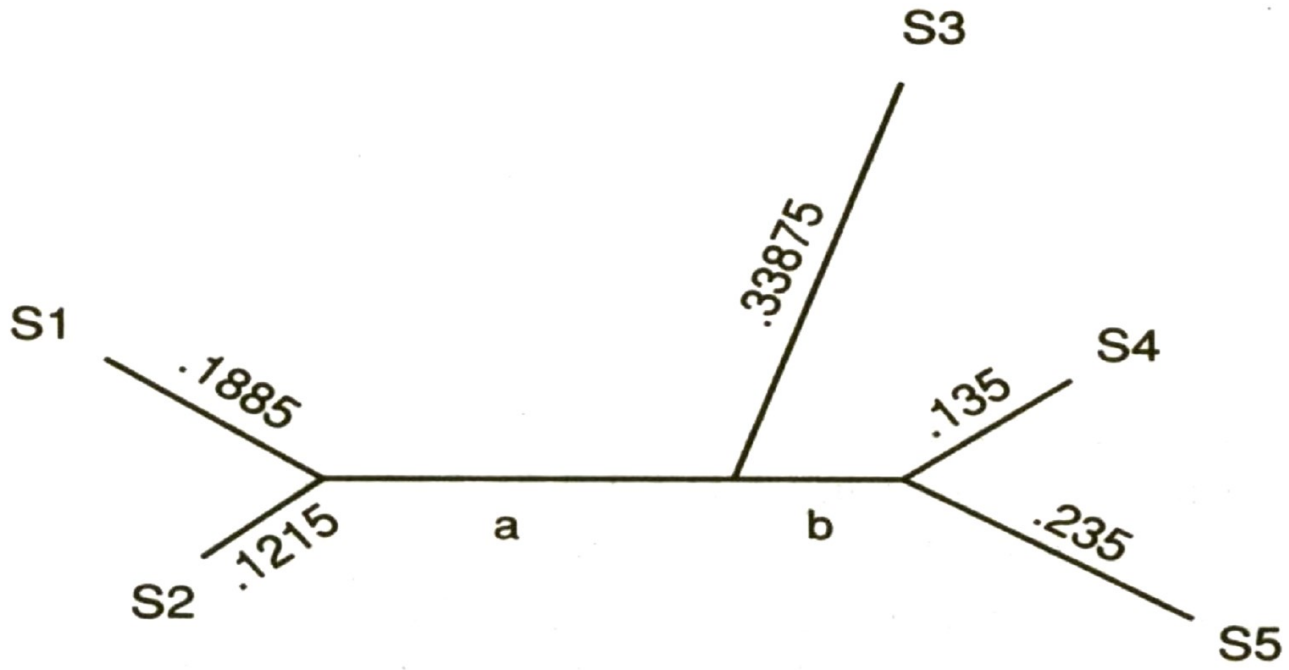
x =0.33875
y = 0.66625
z = 0.17625



FM algorithm: Step 3

Fitch-Margoliash Algorithm

Replacing the groups already determined in the earlier steps gives us the following tree:



FM algorithm: Final Tree

Final step is to compute the lengths *a* and *b*.

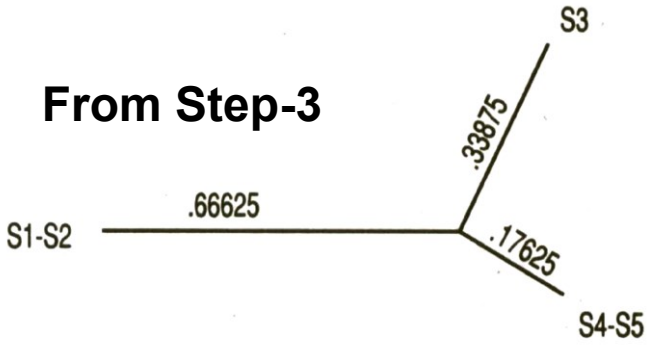
Fitch-Margoliash Algorithm

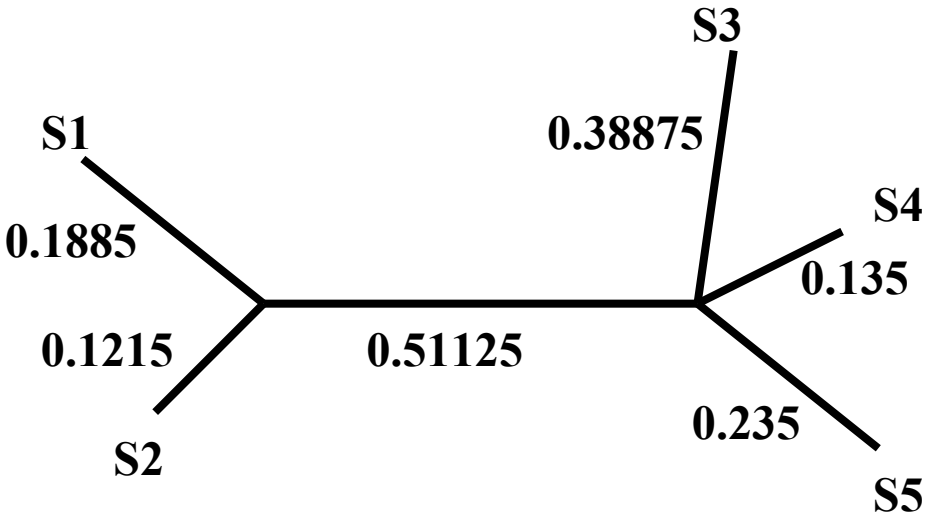
Since S_1 & S_2 are on average (0.1885 + 0.1215)/2 = 0.155 from the vertex joining them

S_4 & S_5 are on average (0.135 + 0.235)/2 = 0.185 from the vertex joining them, hence

$$a = 0.66625 - 0.155 = 0.51125,$$

$$b = 0.17625 - 0.185 = -0.00875.$$





Fitch-Margoliash Algorithm

FM algorithm and UPGMA both produce exactly the same topological tree when applied to a data set.

The reason being, when deciding which taxa or groups to join at each step, both methods consider exactly the same collapsed data table and both choose the pair corresponding to the smallest entry in the table.

Only the metric features of the resulting trees differ, undermining the hope that FM algorithm is much better than UPGMA

Fitch-Margoliash Algorithm

To summarize,

FM produces a better metric tree, but topologically it never differs from UPGMA

FM does not assume molecular clock hypothesis

It produces an unrooted tree, while UPGMA gives a rooted tree

Rooting a Tree

Finding a root is often desirable.

When applying any phylogenetic tree method that produces an unrooted tree, an additional taxon can be included, which is chosen so that it is known to be more distantly related to each of the taxa of interest than they are to each other, and is known as an outgroup.

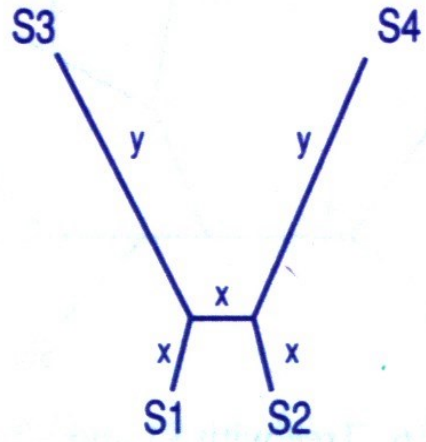
The root is located where the edge to the outgroup joins the rest of the tree.

Neighbour Joining

Both UPGMA & FM algorithm have a flaw.

Consider the metric tree with 4 taxa, where, x and y represent specific lengths, with x << y.

S_1 & S_3 are neighbours



S_2 & S_4 are neighbours

A 4-taxon metric tree with distant neighbours, x << y

No molecular clock operating

Neighbour Joining

If y > 2x, vertices S_1 & S_2 are closest by distance

	S_1	S_2	S_3	S_4
S_1		3x	x+y	2x+y
S_2			2x+y	x+y
S_3				x+2y

The very first joining step will be incorrect in FM or UPGMA, and once we join non-neighbours, we will not recover the true tree.

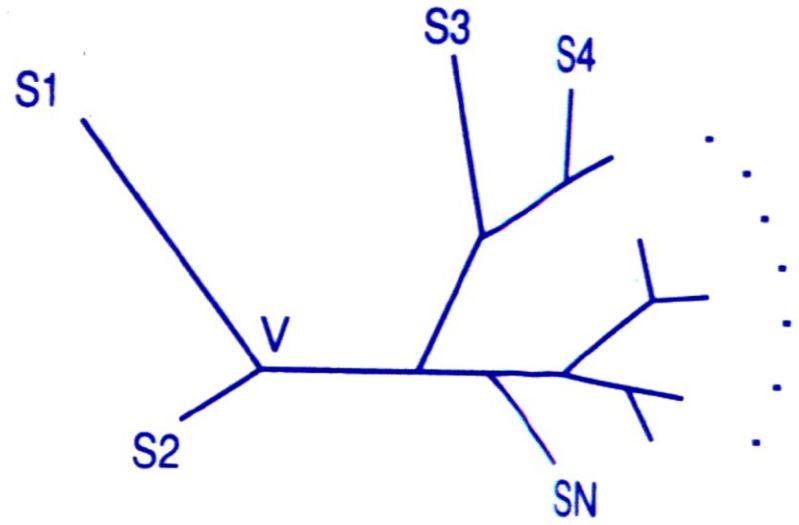
Neighbour Joining

Choosing the closest taxa can be misleading.

We need a more sophisticated criterion for choosing the taxa to join.

Consider a tree in which taxa S_1 and S_2 are neighbours joined at vertex V, with V joined to the remaining taxa S_3, S_4, … , S_N.

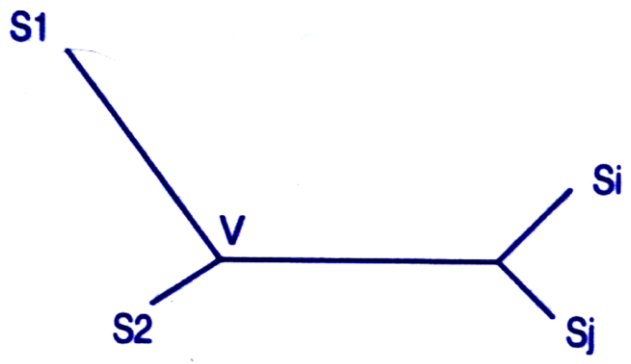
Tree with S1 and S2 neighbours



Neighbour Joining

If the given data exactly fit this metric tree, then for every *i*, *j* = 3, 4, ..., N, the tree would include a subtree as shown below:

Subtree of the previous tree



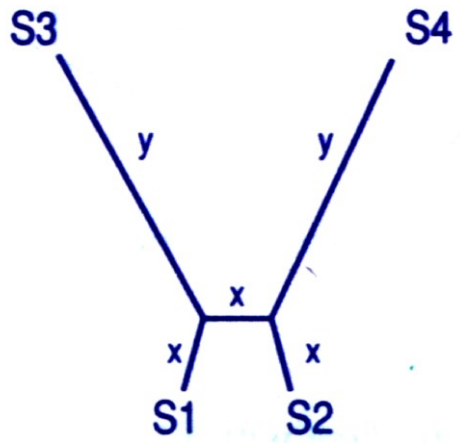
If S_1 and S_2 are neighbours, then for any choice of *i*, *j* between 3 and N:

$$d(S_1, S_2) + d(S_i, S_j) < d(S_1, S_i) + d(S_2, S_j)$$

- 4-point condition for neighbours, is the basis for Neighbour Joining method

Neighbour Joining

Check this 4-pt condition for the tree:



$$d(S_1, S_3) + d(S_2, S_4) < d(S_1, S_2) + d(S_3, S_4)$$

$$(x+y) + (x+y) < 3x + (x+2y)$$

$$\Rightarrow \quad 2x + 2y < 4x + 2y$$

This criterion holds true irrespective of whether x is < or > y.

Neighbour Joining

For <u>fixed</u> *i*, there are *N* - 3 possible choices of *j* with $3 \le j \le N$ and $j \ne i$. Adding up the 4-point inequalities for all *j*, we get

$$(N-3)d(S_1,S_2)+\sum_{\substack{j=3\\ j\neq i}}^{N} d(S_i,S_j) < (N-3)d(S_1,S_i)+\sum_{\substack{j=3\\ j\neq i}}^{N} d(S_2,S_j)$$

For N=4, i=3, only 1 subtree possible: S_1-S_2-S_3-S_4

For N=5, i=3, 2 subtrees are possible:

S_1-S_2-S_3-S_4 , S_1-S_2-S_3-S_5

Neighbour Joining

To simplify this relation,

$$(N-3)d(S_1,S_2)+\sum_{\substack{j=3\\ j\neq i}}^{N} d(S_i,S_j) < (N-3)d(S_1,S_i)+\sum_{\substack{j=3\\ j\neq i}}^{N} d(S_2,S_j)$$

Define total distance from taxon S_i to all other taxa as

$$R_i = \sum_{j=1}^{N} d(S_i,S_j) \qquad d(S_i, S_i) = 0$$

Neighbour Joining

$$(N-3)d(S_1,S_2)+\sum_{\substack{j=3\\ j\neq i}}^{N} d(S_i,S_j) < (N-3)d(S_1,S_i)+\sum_{\substack{j=3\\ j\neq i}}^{N} d(S_2,S_j)$$

Adding $d(S_i, S_1) + d(S_i, S_2) + d(S_1, S_2)$ to each side of inequality, we obtain

$$(N-2)d(S_1,S_2)+R_i < (N-2)d(S_1,S_i)+R_2$$

Subtracting $R_1 + R_2 + R_i$ from each side of inequality gives the more symmetric form

$$(N-2)d(S_1,S_2)-R_1-R_2 < (N-2)d(S_1,S_i)-R_1-R_i$$

Neighbour Joining

Generalizing to any S_n & S_m, rather than to S_1 & S_2,

$$M(S_n, S_m) = (N-2)d(S_n, S_m) - R_n - R_m$$

Then, if S_n and S_m are neighbours,

$$M(S_n, S_m) < M(S_n, S_k) \quad for \quad all \quad k \neq m$$

- criterion used for Neighbour Joining

Neighbour Joining

Criterion used for Neighbour Joining:

From the distance data $d(S_i, S_j)$, compute a new table of values for $M(S_i, S_j)$.

Then, choose to join the pair of taxa with the smallest value of $M(S_i, S_j)$,

i.e., if S_1 and $S2$ are neighbours, their corresponding M value will be the smallest in the distance table

Outline of the NJ Method

Step – 1: Given distance data for ***N*** taxa, compute a new table of values of ***M***.

Choose the smallest ***M*** value to determine which taxa to join.

Step – 2: If S_i & S_j are to be joined at a new vertex V, temporarily collapse all other taxa into a single group G, and determine lengths of the edges from S_i & S_j to V using the 3-point formulas, as in FM algorithm.

Outline of the NJ Method

Distances of S_i and S_j to the internal vertex V are given by

$$d(S_i, V) = \frac{d(S_i, S_j)}{2} + \frac{R_i - R_j}{2(N-2)}$$

$$d(S_j, V) = \frac{d(S_i, S_j)}{2} + \frac{R_j - R_i}{2(N-2)}$$

which can be written as

$$d(S_j, V) = d(S_i, S_j) - d(S_i, V)$$

Outline of the NJ Method

Step – 3: Determine distances from each taxa S_k in G to V by applying 3-point formulas to the distance data for the 3 taxa S_i, S_j and S_k. Now include V in the table of distance data, and drop S_i and S_j.

$$d(S_k, V) = \frac{d(S_i, S_k) + d(S_j, S_k) - d(S_i, S_j)}{2}$$

Step – 4: Distance table now includes N - 1 taxa. If there are only 3 taxa, use the 3-point formula to finish. Otherwise, go back to Step-1 and repeat.

Accuracy of Various Methods

Testing the accuracy of various tree construction methods:

- **Simulate DNA mutation according to certain specified phylogenetic trees and then apply the methods to see how often we recover the correct tree**
- **Studies carried out with real taxa related by a known phylogenetic tree, and tree constructed from DNA sequences using various methods compared with the known correct tree.**

These tests suggest NJ method more reliable than UPGMA or FM methods, especially if no molecular clock is operating.

Example – NJ Method

Consider the distance table to construct a tree using NJ algorithm:

	S_1	S_2	S_3	S_4
S_1		0.83	0.28	0.41
S_2			0.72	0.97
S_3				0.48

(a) Compute R1, R2, R3, R4, and the table of M values for the four taxa: S1, S2, S3, & S4.

$$R_i = \sum_{j=1}^{N} d(S_i, S_j)$$

$$M(S_n, S_m) = (N-2)d(S_n, S_m) - R_n - R_m$$

Example – NJ Method

R1 = d(S1, S2) + d(S1, S3) + d(S1,S4)
= 0.83 + 0.28 + 0.41 = **1.52**

R2 = d(S2, S1) + d(S2, S3) + d(S2,S4)
= 0.83 + 0.72 + 0.97 = **2.52**

R3 = d(S3, S1) + d(S3, S2) + d(S3,S4)
= 0.28 + 0.72 + 0.48 = **1.48**

R4 = d(S4, S1) + d(S4, S2) + d(S4,S3)
= 0.41 + 0.97 + 0.48 = **1.86**

Example – NJ Method

Since,

$$M(S_n, S_m) = (N-2)d(S_n, S_m) - R_n - R_m$$

M(S1, S2) = (4 - 2) x 0.83 – 1.52 – 2.52 = - 2.38

M(S1, S3) = (4 - 2) x 0.28 – 1.52 – 1.48 = - 2.44

M(S1, S4) = (4 - 2) x 0.41 – 1.52 – 1.86 = - 2.56

M(S2, S3) = (4 - 2) x 0.72 – 2.52 – 1.48 = - 2.56

M(S2, S4) = (4 - 2) x 0.97 – 2.52 – 1.86 = - 2.44

M(S3, S4) = (4 - 2) x 0.48 – 1.48 – 1.86 = - 2.38

Example – NJ Method

(b) We obtain a tie for the smallest value of M. Consider any one of these smallest values, say, M(S1, S4) = - 2.56, and join S1 & S4 first.

For the new vertex V where S1 & S4 join, compute d(S1, V) & d(S4, V) as given in Step-2, *viz.*,

$$d(S_1, V) = \frac{d(S_1, S_4)}{2} + \frac{R_1 - R_4}{2(N-2)}$$

$$d(S_4, V) = d(S_1, S_4) - d(S_1, V)$$

Example – NJ Method

Step (b) contd.

$$d(S_1, V) = \frac{d(S_1, S_4)}{2} + \frac{R_1 - R_4}{2(N-2)}$$

$$= \frac{0.41}{2} + \frac{1.52 - 1.86}{2 \times (4-2)} = \frac{0.41}{2} - \frac{0.34}{4} = \frac{0.82 - 0.34}{4} = \frac{0.48}{4} = 0.12$$

$$d(S_4, V) = d(S_1, S_4) - d(S_1, V)$$

$$= 0.41 - 0.12 = 0.29$$

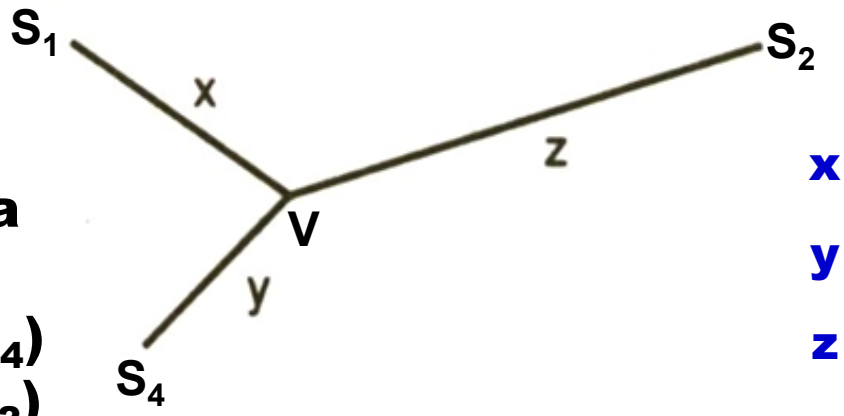
$d(S_1, V) = 0.12$, $d(S_4, V) = 0.29$

Example – NJ Method

(c) Compute $d(S_2,V)$ & $d(S_3,V)$ using 3-point formula:

$$d(S_2, V) = \frac{d(S_1, S_2) + d(S_4, S_2) - d(S_1, S_4)}{2}$$

$$d(S_3, V) = \frac{d(S_1, S_3) + d(S_4, S_3) - d(S_1, S_4)}{2}$$



Distance data defined as:

$x + y = d(S_1,S_4)$
$x + z = d(S_1,S_2)$
$y + z = d(S_4,S_2)$

$x = (d_{14} + d_{12} - d_{42})/2$

$y = (d_{14} + d_{42} - d_{12})/2$

$z = (d_{12} + d_{42} - d_{14})/2$

- 3-point formula for fitting taxa to a tree

Example – NJ Method

Step (c) contd.

$$d(S_2, V) = \frac{d(S_1, S_2) + d(S_4, S_2) - d(S_1, S_4)}{2}$$

$$= \frac{0.83 + 0.97 - 0.41}{2} = \frac{1.39}{2} = 0.695$$

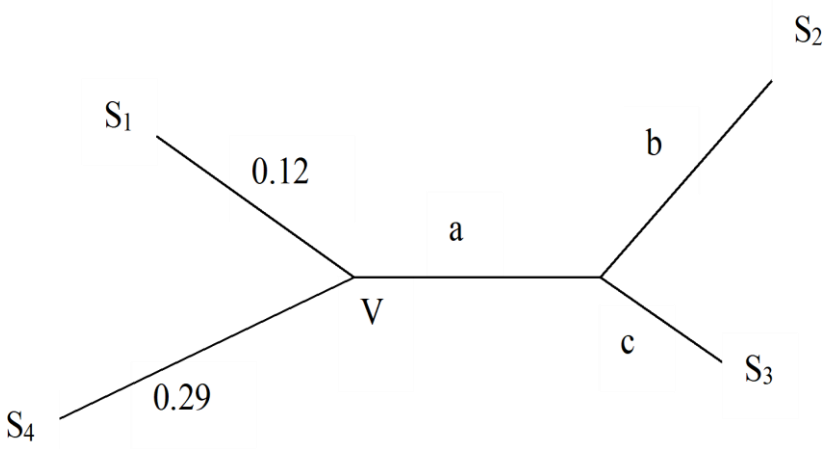
$$d(S_3, V) = \frac{d(S_1, S_3) + d(S_4, S_3) - d(S_1, S_4)}{2}$$

$$= \frac{0.28 + 0.48 - 0.41}{2} = \frac{0.35}{2} = 0.175$$

	V	S_2	S_3
V		0.695	0.175
S_2			0.72

Example – NJ Method

	V	S_2	S_3
V		0.695	0.175
S_2			0.72



From the table above, we have

a + b = 0.695, a + c = 0.175, b + c = 0.72

Solving these equations, we obtain

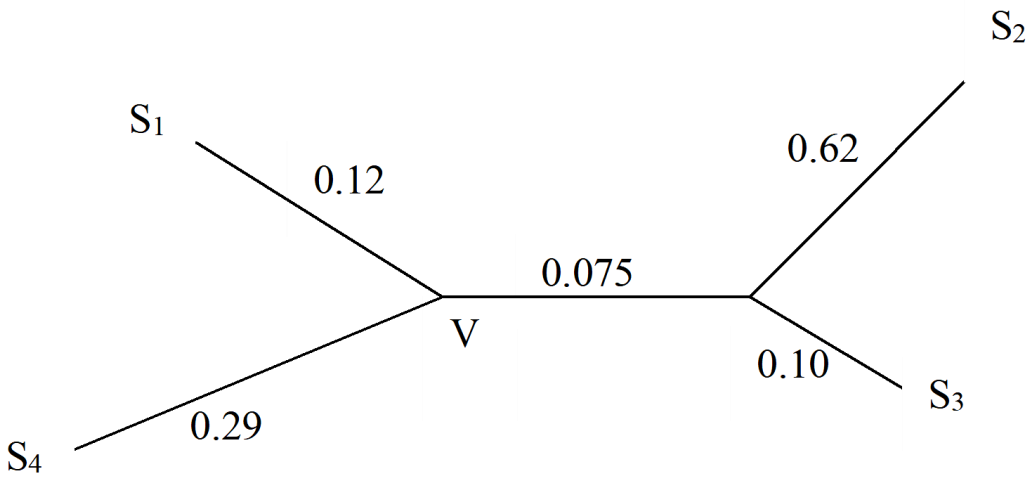
a = 0.075, b = 0.62, c = 0.10

Note: d(S2,S3) = 0.72 from the distance table

Example – NJ Method

(d) Because there are only 3 taxa left, we use the 3-point formulas to fit V, S_2 & S_3 to a tree.

(e) Draw the final tree by attaching S_1 & S_4 to V with the distances from step (b).



Distance Methods

- **Success of distance methods lies on the degree to which distances are additive**
- **Additivity for four sequences is defined as:**

$$d_{AB} + d_{CD} \leq d_{AC} + d_{BD} = d_{AD} + d_{BC},$$

AB & CD – neighbours

$\Rightarrow$ **4-point formula used in NJ approach is called the additive property of distance trees.**

- **Additivity - destroyed by homoplasy**

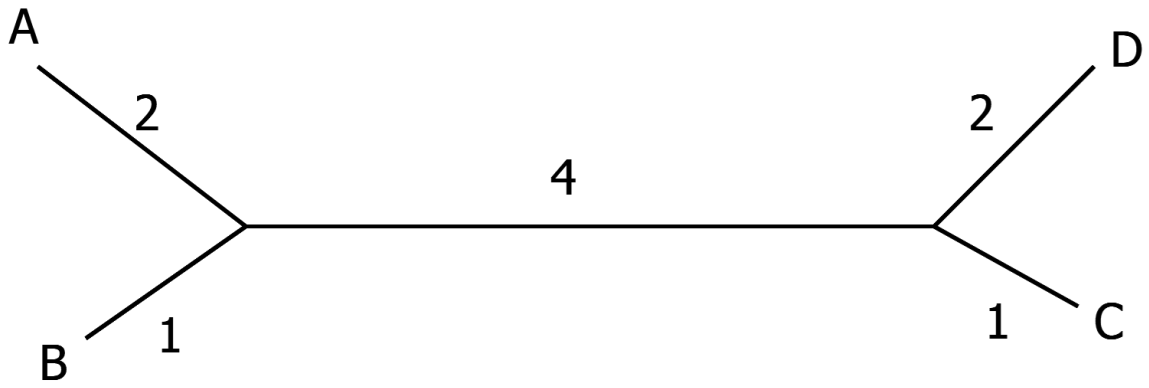
Homoplasy: identical character due to evolutionary convergence or reversal (evolutionary noise)

Distance Methods

Following example presents the ideal case when distances are completely additive:

Sequence A: A C G C G T T G G G C G A T G G C A A C
Sequence B: A C G C G T T G G G C G A C G G T A A T
Sequence C: A C G C A T T G A A T G A T G A T A A T
Sequence D: A C A C A T T G A G T G A T A A T A A T

Distance: $d_{AB} = 3$ $d_{AC} = 7$ $d_{AD} = 8$ $d_{BC} = 6$ $d_{BD} = 7$ $d_{CD} = 3$



$$d_{AB} + d_{CD} \leq d_{AC} + d_{BD} = d_{AD} + d_{BC}$$

Summarize

Method	Tree	Molecular Clock
UPGMA	Rooted	Exists
FM	Unrooted	Does not Exist
NJ	Unrooted	Does not Exist

Phylip Programs: Distance-based

- **dnadist** - computes distance among DNA sequences
- **protdist** - computes distance among protein sequences
- **fitch** - estimates branch length assuming additivity of branch lengths using Fitch-Margoliash method; molecular clock not assumed
- **kitsch** - same as fitch but assumes molecular clock
- **neighbor** - estimates phylogeny using neighbour-joining method

Maximum Parsimony

Criticism with distance methods

- reduces full DNA sequence data to a collection of pairwise distances between taxa

- they may not use all the information in the original sequences.

Maximum Parsimony method is a different approach that uses the entire sequences

- among all possible trees that might relate the taxa, it looks for the one that would require fewest possible mutations to have occurred. **Why?**

To assess the no. of mutations, distances are not computed, how mutations occur at each separate site in the sequences is considered.

Maximum Parsimony

Method:

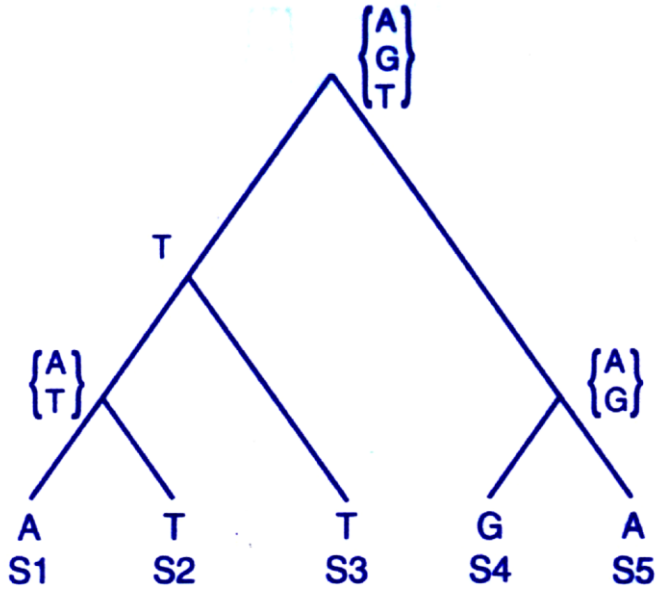
- **For a given tree, count the smallest number of mutations that would have been required if the sequences had arisen from a common ancestor according to that tree. This number is referred as the parsimony score of the tree.**
- **Compute parsimony score for all possible trees that might relate the given taxa.**
- **Choose the tree with smallest parsimony score**

  - the most parsimonious tree is considered to be optimal for the given sequence data.

Maximum Parsimony

e.g., suppose we look at a single site for 5-taxa:

S_1: A, S_2: T, S_3: T, S_4: G, S_5: A



No. of mutations = 3

parsimony score = 3

Considering taxa related by this tree, trace backward up the tree to determine what base might have been at each vertex, assuming fewest possible mutations occurred.

Maximum Parsimony

Problems with this method:

- **Not obvious that the method gives minimum possible mutations needed for the tree.**

  Also, one cannot assign bases to internal vertices in a way that requires fewer mutations

  – as there can be assignments of bases to these vertices that are not consistent with this method, yet achieve the same minimum number of mutations.

Maximum Parsimony

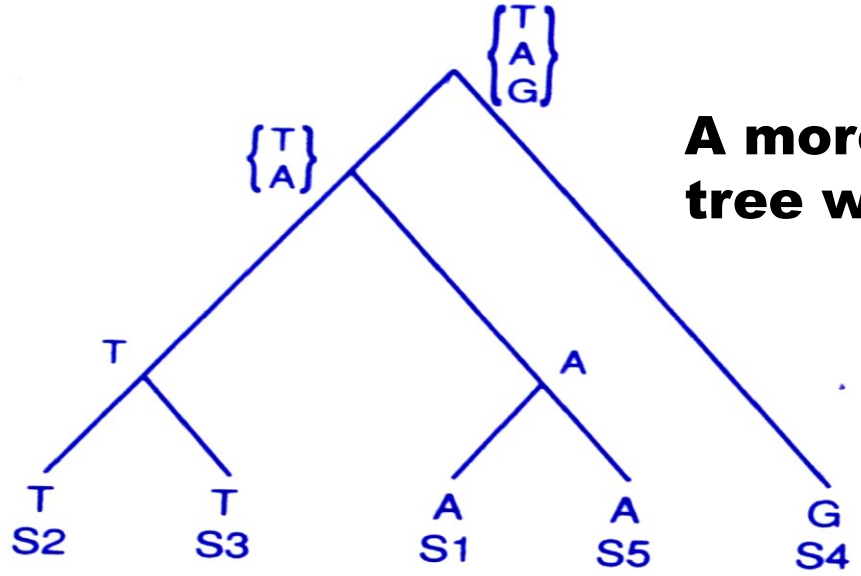
- **Parsimony score of a tree does not depend on the location of root.**

  - while the counting procedure requires temporarily inserting a root, one is really judging the fitness of an unrooted tree.

- **Because the method does not reliably construct the sequences at internal vertices, we have no way of knowing along which edges mutations occurred, i.e., we cannot assign a precise length to an edge by using the number of mutations occurring along it.**

Maximum Parsimony

Consider another tree relating the same 1-base sequences.



A more parsimonious tree with score 2

Labeling the internal vertices as before, we find this tree requires only two mutations. Thus, this tree is more parsimonious than the earlier one

Maximum Parsimony

To find the most parsimonious tree relating 5 taxa, we need to consider all 15 possible topologies of unrooted trees and compute the minimum number of mutations for each.

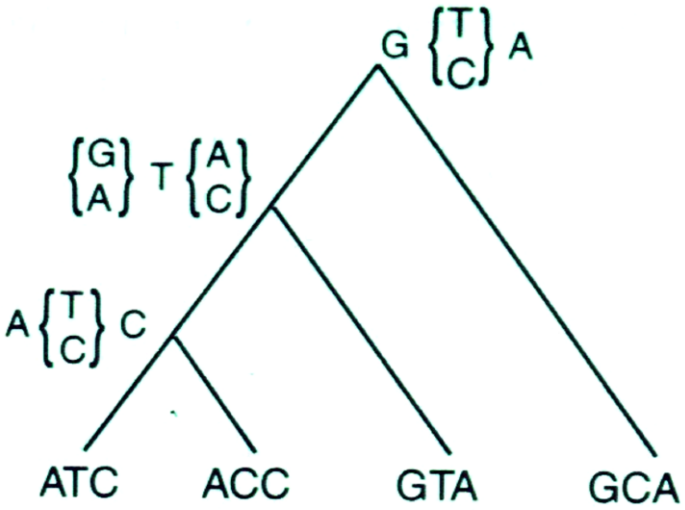
For this example, there are 5 trees having the same parsimony score 2, the method reports all 5 trees, as all are equally good by selection criterion

When dealing with real sequence data, we need to count the no. of mutations required for a tree along all sites in the sequences.

- done in the same manner, treating each site in parallel.

Maximum Parsimony

Consider an example with 3 sites:



parsimony score = 4

- comparing ATC and ACC, the mutation count is 1.

At the vertex where the 3rd taxa joins, the mutation count increases to 3.

At the root, we need a mutation in the 2nd site

Maximum Parsimony

As the no. of sites and the no. of taxa increase, no. of tree topologies that must be considered is huge

- impractical for large sequences.

Some effort in using the parsimony method can be saved, if we make the observation that not all sites will affect the number of mutations needed for a tree.

Maximum Parsimony

- If all sequences have the same base at a site, then all trees will need 0 mutations for that site

  - these columns can be eliminated before applying the algorithm.

- When at a site all sequences have the same base (say A), except for at most one sequence each with the other bases (C, T, and G). In this case, regardless of the tree topology, if we put an A at every interior vertex, then we have the minimum possible no. of mutations
- An informative site is one at which at least two different bases occur at least twice each among the sequences being considered.

Maximum Parsimony Example

Taxa	1	2	3	4	5	6	7	8	9
1	A	A	G	A	G	T	G	C	A
2	A	G	C	C	G	T	G	C	G
3	A	G	A	T	A	T	C	C	A
4	A	G	A	G	A	T	C	C	G

- Four taxa giving 3 possible trees
- Sites 1, 6 & 8 not informative - do not favour a tree over another
- Sites 2, 3, & 4 not informative - to be informative sites should have same character in at least 2 taxa;
- Only informative sites analyzed: 5, 7, 9

Maximum Parsimony

Maximum Parsimony method does not use any explicit model of DNA mutation, *viz.*, Jukes-Cantor model of molecular evolution.

Instead it carries an implicit assumption that mutation is rare, and the best explanation of evolutionary history is the one that requires the least mutation.

Parsimony Method: Discussion

- **Lake's method of invariance or evolutionary parsimony is another approach to identify long branches**

  - Four sequences are considered at a time, and only transversions are scored as changes

  - All transversions are assumed to occur at the same rate

- **dnainvar in Phylip computes Lake's and other phylogenetic invariants**

Phylip Programs: Parsimony

- **dnapars:** treats gaps as 5th state
- **dnapenny:** uses branch and bound method
- **dnacomp:** based on compatibility criteria; finds tree that supports largest number of sites
- **dnamove:** performs parsimony and compatibility interactively
- **protpars:** based on no. of mutations to change a codon for aa1 to codon for aa2 for non-synonymous changes only

Maximum Likelihood

- **Maximum likelihood is similar to maximum parsimony, in that analysis is performed for each column of the alignment, all possible trees are considered, and trees with fewest changes are usually more likely**
- **ML allows corrections for variations in the mutation rates by considering explicit evolutionary models**
- **Method can be used to explore relationships among more diverse sequences**

Maximum likelihood estimation

So what's the the concept of likelihood?

If the probability of an event *X* dependent on model parameters *p* is written as *P*(*X* | *p*)

then the likelihood of the parameters given the data is

$$L(p \mid X)$$

For most models, we find that certain data are more probable than other data.

Aim of maximum likelihood estimation is to find the parameter value(s) that makes the observed data most likely

Maximum likelihood estimation

In Probability theory, we are interested in making predictions, i.e., finding the probability of occurrence/non-occurrence of certain outcomes

In Data Analysis, we have already observed all the data: once they have been observed they are fixed, there is no 'probabilistic' part to them anymore.

We are now interested in the likelihood of the model parameters that underlie the fixed data.

Maximum likelihood estimation

Probability

Knowing parameters $\Rightarrow$ Predict outcome

Likelihood

Observation of data $\Rightarrow$ Estimate parameters

In data analysis, likelihood and log-likelihood functions are the basis for deriving estimators for parameters, given data

Maximum Likelihood method

Basic approach of maximum likelihood:

- **Specify a particular model of molecular evolution (such as Jukes-Cantor, Kimura, etc.).**
- **Consider a specific tree for relating our taxa. Assuming the model of evolution and specific tree are correct, compute the probability that the DNA sequence in our data could have been produced.**

  This is the likelihood of the tree, given our data.
- **The observed data here are the mutations observed in the sequences.**

Maximum Likelihood method

There should be a model that describes the mutational events.

The parameters of this model would give us the rate of mutation observed in these sequences.

- **Repeat this process for all other trees and compute likelihood value for each.**
- **Choose the tree with the greatest likelihood as the tree best fitting the data.**

Maximum Likelihood method

- **ML assumes a model of evolution, which defines the probability/rate with which nucleotides mutate**
- **Probability of each tree is product of mutation rates in each branch**
- **Likelihoods given by each column are multiplied to give the likelihood of the tree**
- **Phylip programs dnaml and dnamlk (same as dnaml except that it assumes a molecular clock)**

Maximum Likelihood method

Problems with this method:

- **First, depends on choosing a specific model of evolution, and if that model does not describe the real process well, one could question the validity of the method.**
- **Second, as with parsimony, the method requires considering all possible trees, and so is computationally very intensive.**

Which method is the Best?

One of the difficulties of picking a method is that one can find good arguments for and against them all.

Cautious approach - always use a number of different methods on the data.

Rather than trusting a single method to give an accurate tree, check to see if different methods give roughly the same results.

They often do, and if they do not, it is worth investigating why they don't.

Bootstrapping

Once a tree has been chosen by some method, it would be desirable to quantify how confident one is of it.

This is given by the statistical technique - bootstrapping.

In this procedure, the true data sequences are used to create a set of new pseudo-replicate sequences of the same length.

Bases at a particular site in the new sequences are chosen to be the bases appearing in a randomly chosen site in the original sequences.

Bootstrapping

A tree is constructed for the phylogeny of the pseudo-replicates and recorded.

This procedure is repeated many times, giving a large collection of bootstrap trees.

If a high percentage of bootstrap trees are in agreement with the one produced using original data, then we may be more confident of it.

Based on the concept that the data accurately reflects the variation in the population; by repeated sampling of the data the effect of this variation on any statistic of interest can be understood

Bootstrapping

An important caveat on using bootstrapping is that

- the technique only helps assess the effects on tree construction of variability within the sequences.

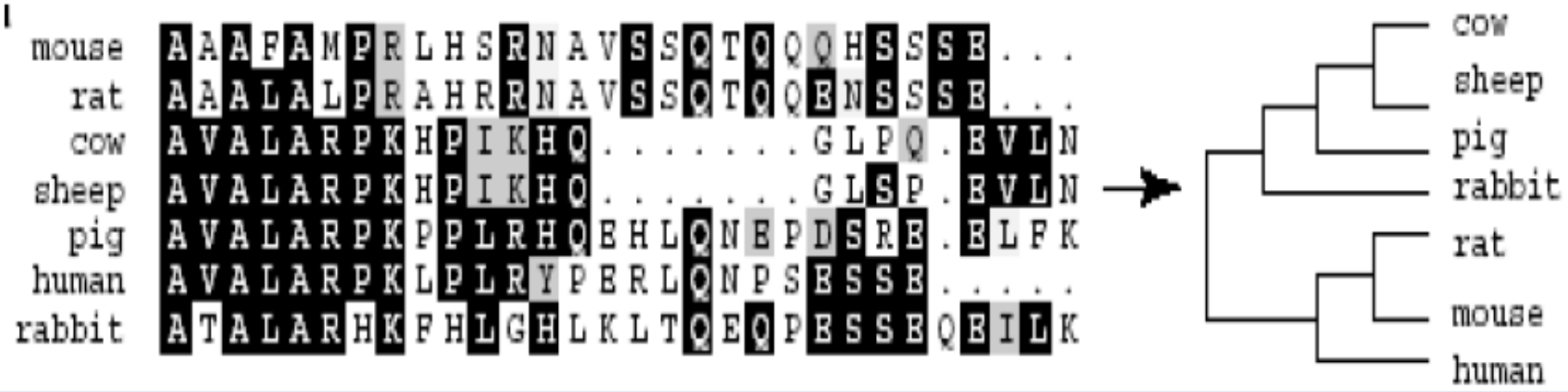
Bootstrapping says nothing about the fundamental soundness of the method by which we choose a tree

– it only indicates how variability in the data affects the outcome of the method.

Bootstrapping

Initial alignment and tree:

Each aligned site is considered independent

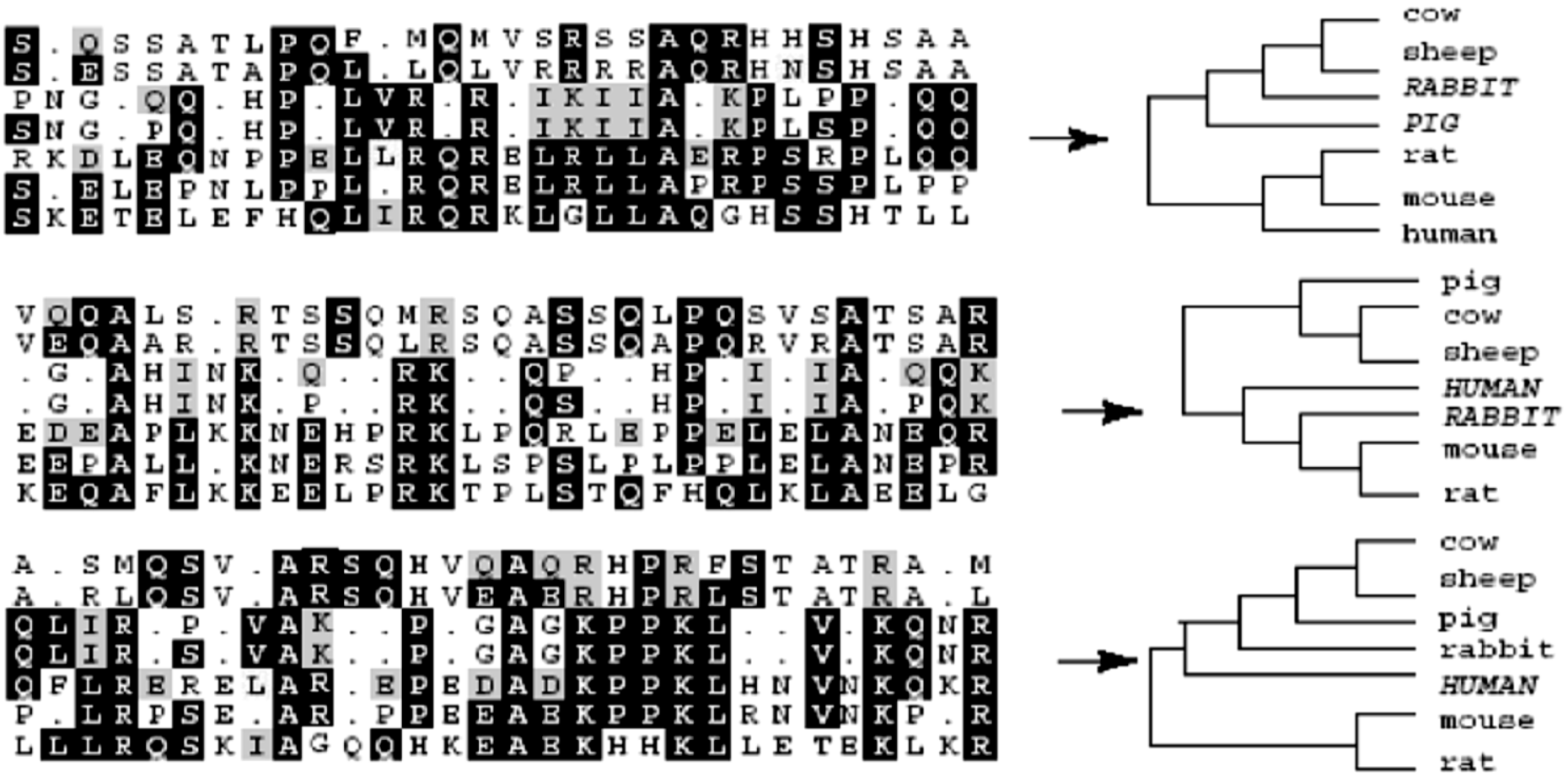


In bootstrap analysis:

- **All sites are considered independent as if freely available in a 'hat' to pick**
- **Available sites are picked up randomly to reconstruct a new alignment of the <u>original size</u> and a new phylogeny**

Bootstrapping

- **The process is repeated many times to ascertain the strength of clustering. New "alignment" may contain several sites multiple times while some other sites may be absent (sampling with replacement)**



Bootstrapping

- **Phylogenies are compared to calculate values [Bootstrap value] that signify the number of times a given branch/cluster occurred in the Multiple bootstrap trees**
- **Higher the value - higher the confidence of phylogenetic inference**
- **In general values < 50% provide very poor support**

Comparison of Methods

Neighbor-joining	Maximum parsimony	Maximum likelihood
Uses only pairwise distances	Uses only shared derived characters	Uses all data
Minimizes distance between nearest neighbors	Minimizes total distance	Maximizes tree likelihood given specific parameter values
Very fast	Slow	*Very* slow
Easily trapped in local optima	Assumptions fail when evolution is rapid	Highly dependent on assumed evolution model
Good for generating tentative tree, or choosing among multiple trees	Best option when tractable (<30 taxa, homoplasy rare)	Good for very small data sets and for testing trees built using other methods

Summarize

There are many approaches to phylogenetic inference that are not sequence-based.

Evidence of all should be weighed before making too strong a statement about the phylogeny of the species under consideration.

Application

Few examples of applications of phylogeny inference

I Investigating whether the evolution of several species parallel one another:

(i) Evolution of hosts & parasites can be studied by constructing separate phylogenetic trees for each.

- similarity of tree topologies can indicate whether the parasites evolved with the host, or if the parasites "jumped" from one host species to another

(ii) Likewise, trees for two symbiotic species, such as fungus-growing ants and the fungus they grow

- help indicate how far back in evolutionary history the symbiotic partnership stretches.

Application

- **Determining likely infection sources of HI by constructing trees from HIV sequences from a number of infected individuals:**

There have been several forensic applications of this:

- **the Florida Dentist AIDS cases, and**
- **the case of a doctor accused of intentionally injecting HIV into a former lover**

Application

- **Studies whether genes have entered the genome of a species through lateral transfer.**

  When a tree is constructed from DNA sequences for a gene, it is really a "gene tree" showing gene relationships that may or may not be the same as taxa relationship.

  Because some human genes are believed to have been obtained by lateral transfer from bacteria that infected us, for certain genes we may appear to be more closely related to some bacteria than other mammals.

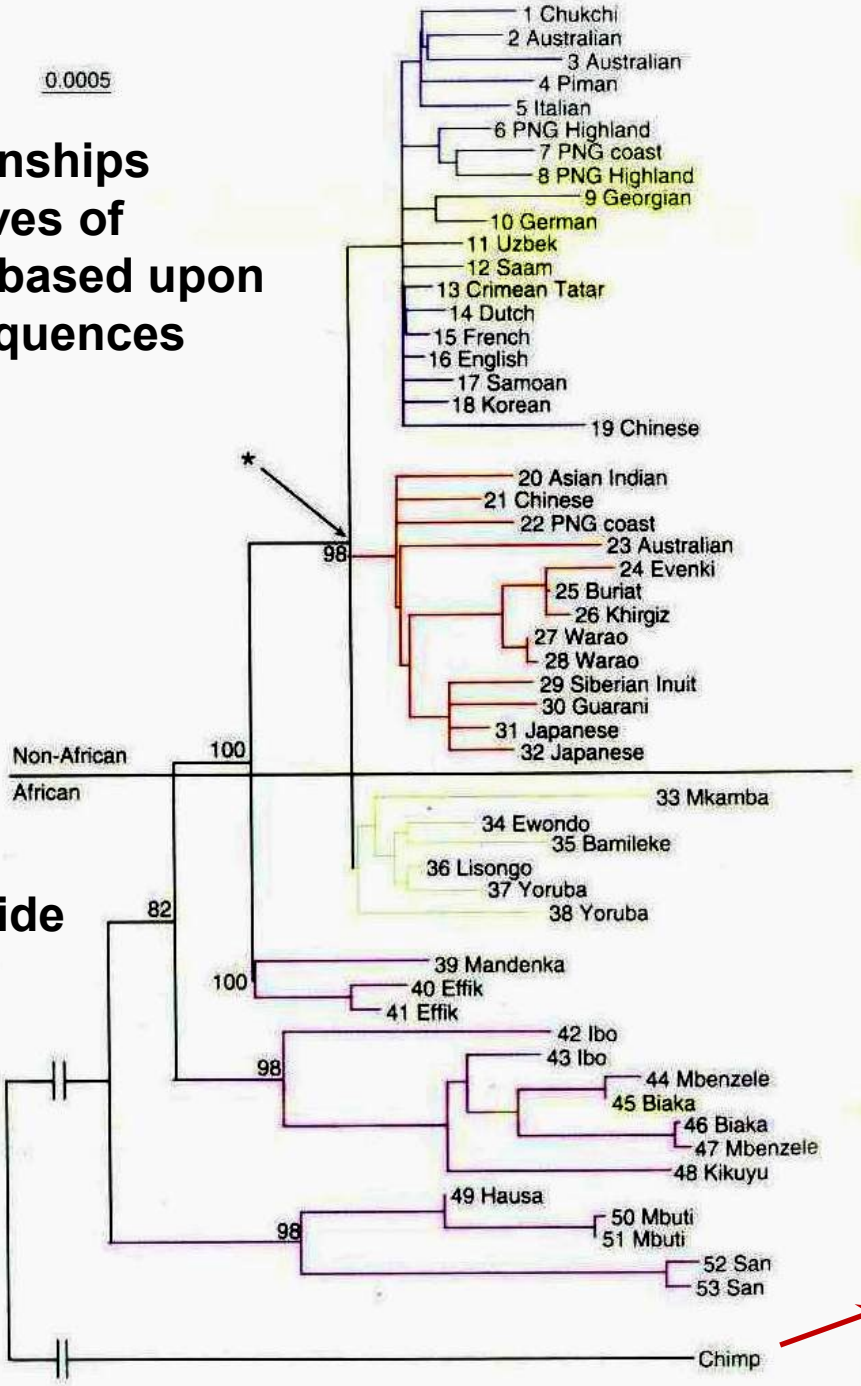
Application

IV Investigating the "Out of Africa" hypothesis of human origins:

The clustering pattern on a tree constructed from human DNA sequences from ethnic groups around the world should help indicate how human populations are related and hence how and from where they spread.

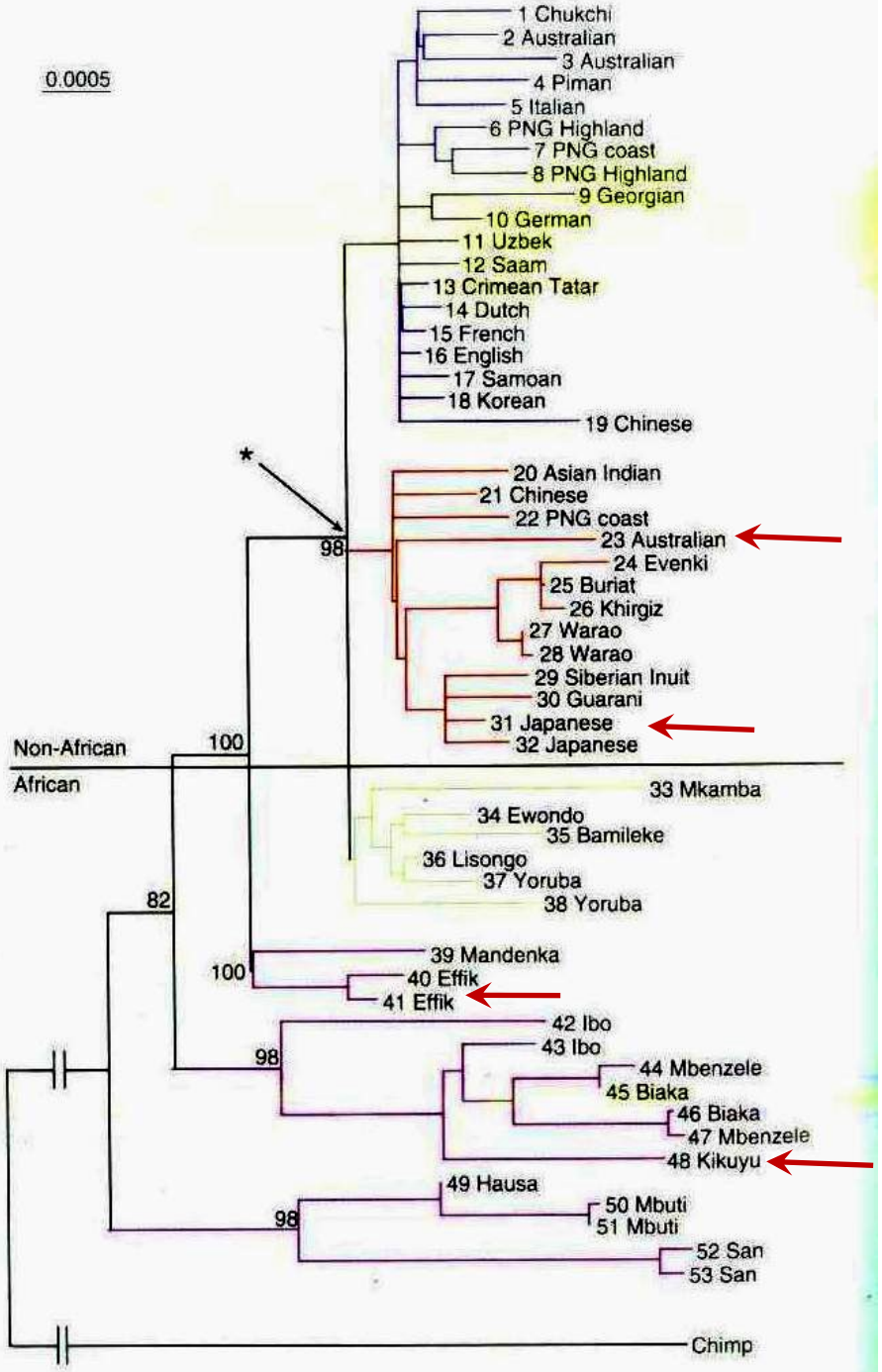
Populations of humans that share particular sets of mutations are understood to be more closely related to each other than they are to populations lacking these mutations.

Phylogenetic relationships among representatives of human populations based upon complete mtDNA sequences



Scale: no. of nucleotide differences per site

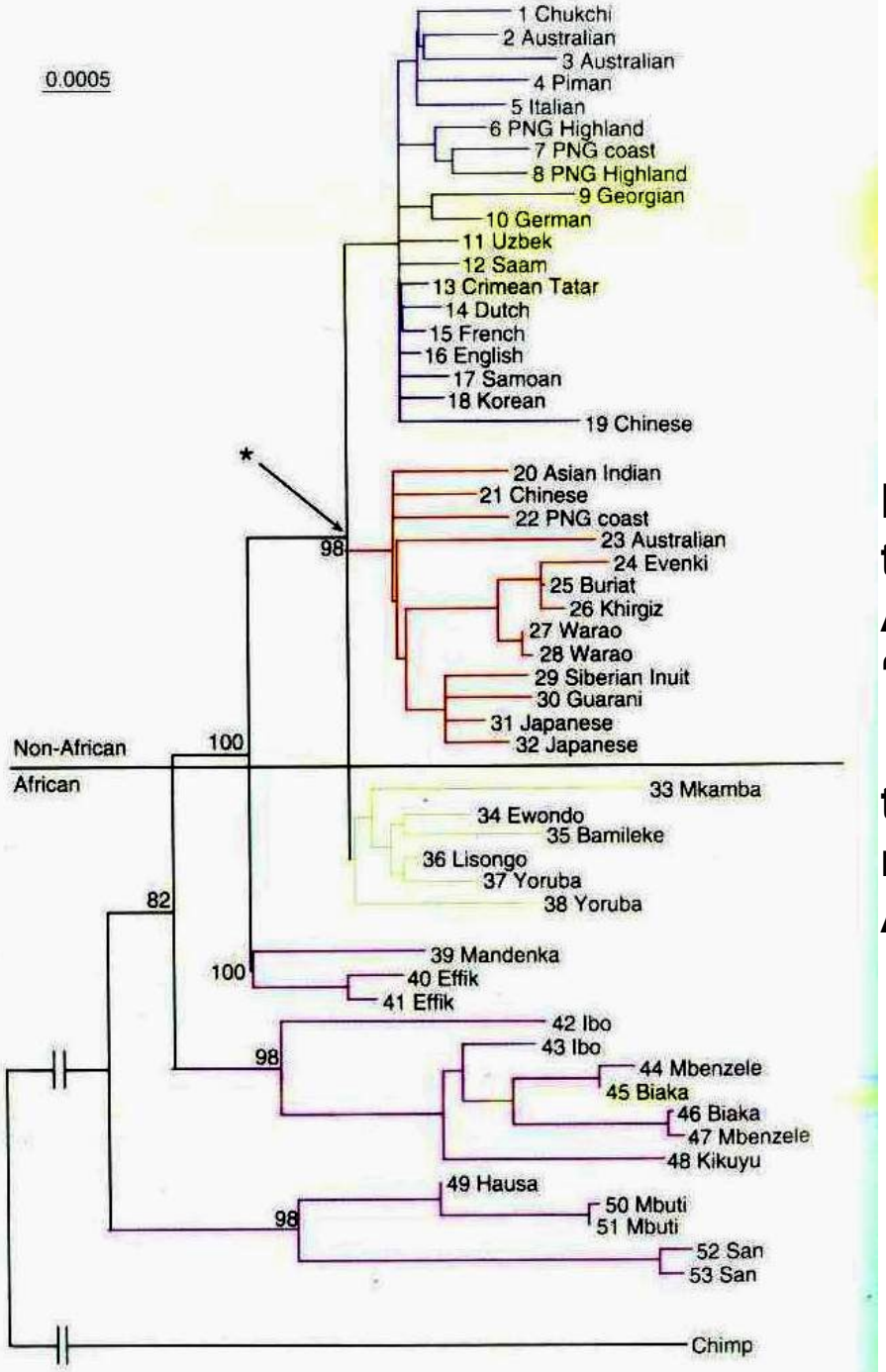
outgroup



Tree suggests native Australians are more closely related to Japanese than they are to the major groups of Africans.

Note: diversity of African populations: Kikuyu people differ more from Effik than Europeans differ from Papua New Guinean highlanders.

- they have had a longer time to evolve and accumulate mutations



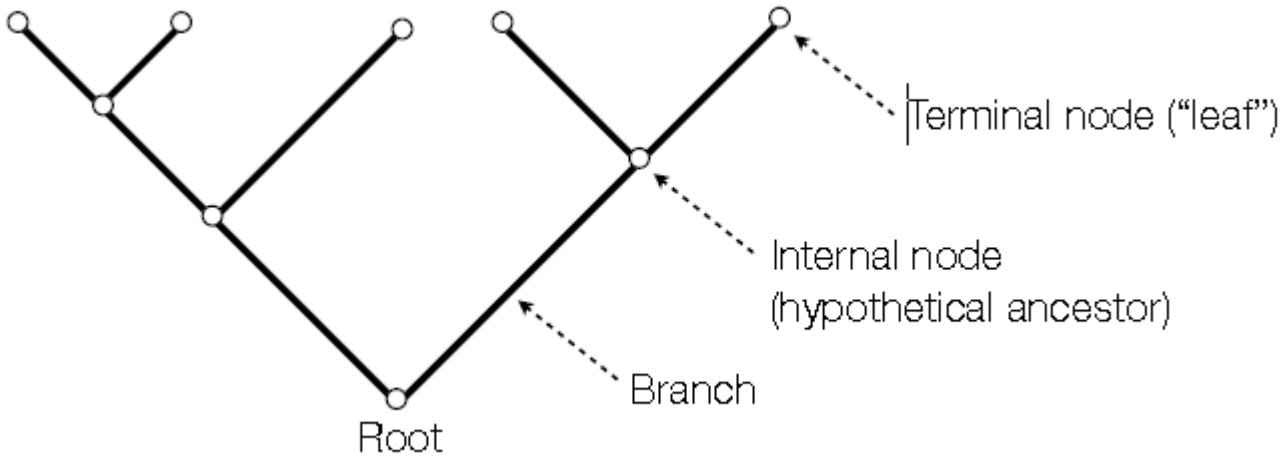
It is observed that populations from the rest of the world are a subset of African populations, supporting the "Out-of-Africa" hypothesis that

- modern humans in other parts of the world are descendants of migrants that originated from an African population.

Phylogenetic Trees

The sequences we want to relate could come from different species, subspecies, populations, or even individuals, each source of the DNA sequence is termed a *taxon*.

An equivalent term in common use is *operational taxonomic unit*, abbreviated as OTU.



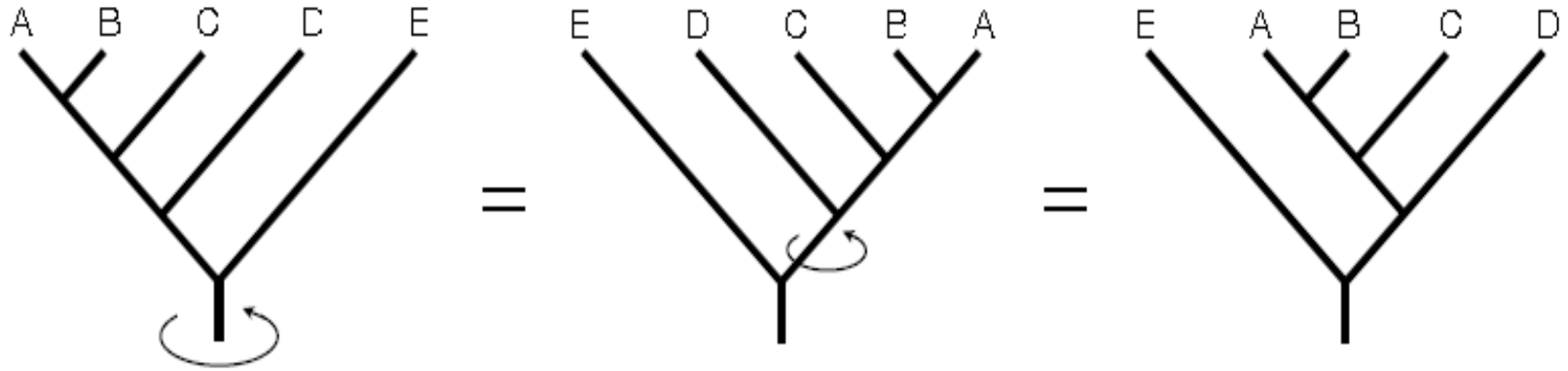
Topological Trees

Two trees are said to be topologically the same if we can bend and stretch the edges of either one to get the other.

We are not allowed to cut off an edge and reattach it elsewhere; doing that may give us a tree that is topologically distinct from the original one.

Topological Trees

Two rooted trees are topologically the same if one can be deformed into the other without moving the root. Edge lengths can be changed, but not the branching structure.

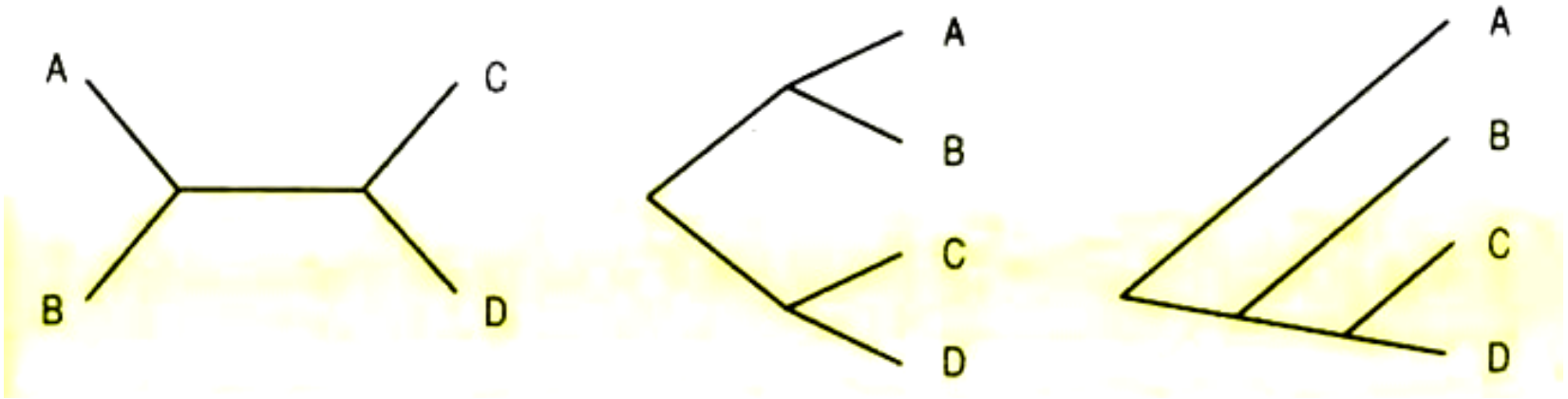


Three different representations of the same tree-topology

- **A rooted tree has directionality (nodes can be ordered in terms of "earlier" or "later").**
- **In rooted tree, distance between two nodes is represented along the time-axis only (the 2nd axis just helps spread out the leafs)**

Topological Trees

Topologically same trees:



In unrooted trees there is no directionality: we do not know whether a node is earlier or later than another node.

Distance along branches directly represents node distance.

Topological Trees

No. of rooted trees (for n species):

$$(2n - 3)!/(2(n-2)[n - 2]!)$$

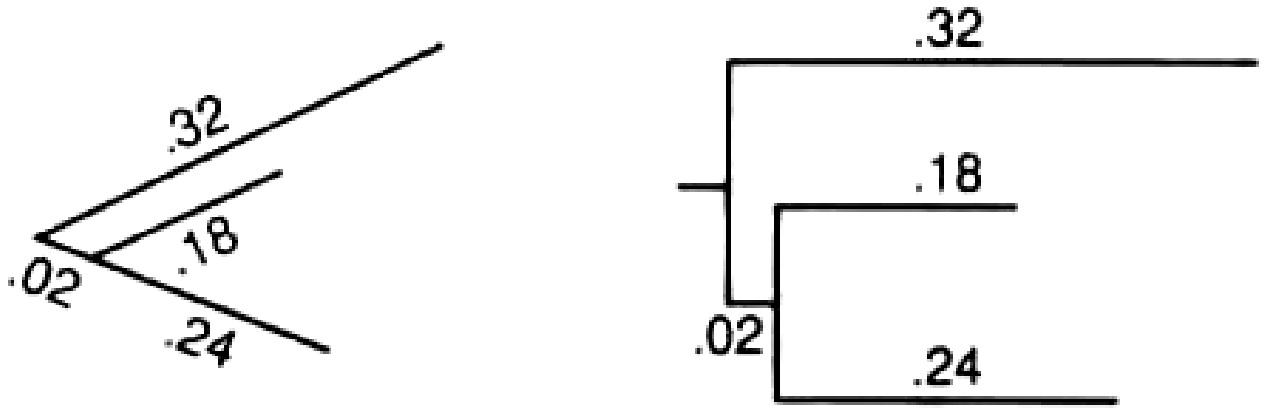
No. of unrooted trees:

$$(2n - 5)!/(2(n-3)[n - 3]!)$$

For rooted trees, the no. of distinct trees grow faster: for 4 species - the no. is 15, for 5 species it is 105, … , for 12 species the no. is more than 13 billion!

For unrooted trees the no. of possible trees is much less: for e.g., only 1 unrooted tree relating 3 taxa, 3 trees relating 4 taxa, 15 trees for 5 taxa, …, for 12 species it is a little more than half a billion

Metric Trees



Alternate depictions of the same metric tree

Problems with Tree Building

Numerous problems are associated with tree-building using molecular data.

We might wonder the adequacy of a given mutation model as a description of the data.

A number of assumptions have been made in modeling the mutation process:

- **Sites evolve independently of one another**
- **Sites evolve according to the same stochastic (Markov) model.**
- **The tree is rooted.**
- **The sequences are well-aligned.**

Problems with Tree Building

While interpreting the data in the context of a tree, we need to recognize that biological mechanisms may impose additional conditions on the distances under particular circumstances

- **A tree is said to be additive if the distance between any pair of leaves is the sum of the distances between these leaves and the first node they share in the tree.**
- **A rooted additive tree is called ultrametric (or clock-like) if the distance between any two leaves and their common ancestor are equal.**

Given a distance matrix, D, can we determine whether we can construct an additive/ultrametric trees corresponding to these distances?

- **Three-point condition for ultrametric trees:**
  D corresponds to an ultrametric tree if and only if for any three sequences *i*, *j*, *k*, the distances satisfy:

$$d_{ij} \leq \max (d_{ik}, d_{kj}).$$

- **Four-point condition for additive trees:**
  D corresponds to an additive tree if and only if for any four sequences, two of the sums $d_{12} + d_{34}$, $d_{13} + d_{24}$, $d_{14} + d_{23}$ are equal and greater than or equal to the third.

Distinction between a gene tree & a species tree:

A gene tree is a tree drawn from DNA or protein sequences corresponding to a particular gene shared by a set of organisms.

Each different gene may (or may not) produce different tree for the same set of organisms.

A species tree is often produced from sets of macroscopic characters but may also be produced from sequence data.

Generating a consensus species tree from a collection of gene trees is a difficult problem.

References

- **Mathematical Models in Biology: An Introduction, E.S. Allman and J.A. Rhodes**
- **Bioinformatics Sequence & Genome Analysis, David W. Mount**
- **Biological Sequence Analysis, Probabilistic Models of Proteins and Nucleic Acids, R. Durbin, S.R. Eddy, A. Keoghs and G. Mitchison**