"I believe greatness is an evolutionary process that changes and evolves era to era."
-- Michael Jordan

Constructing Phylogenetic Trees

There are two main methods of constructing phylogenetic trees:

* **distance-based methods** such as UPGMA and neighbour-joining,

* **character-based methods** such as maximum parsimony, maximum likelihood, or Bayesian inference.

UPGMA (Unweighted Pair Group Method with Arithmetic Mean) is a simple agglomerative (bottom-up) hierarchical clustering method. Generally attributed to Sokal and Michener (1958).

Tree building - UPGMA

UPGMA works by progressively clustering the most similar taxa until all the taxa form a rooted clock-like tree.

1. Find the smallest entry in the distance matrix, say *d(x,y)*.
2. Form a new internal node, *z*, that is a parent to *x* and *y* and set the edge lengths from *z* to *x* and *z* to *y* to half *d(x,y)*.
3. Update the distance matrix by setting the distances from the new node *z* to all the other taxa to be the *average distance* between groups *x* and *y*.

REPEAT until all groups have been joined.

What precisely is meant by the average distance?

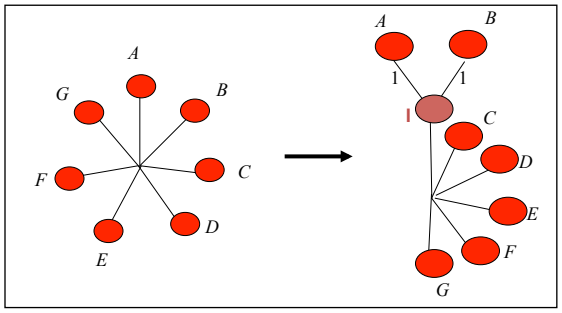
- If we a joining two groups *x* and *y* that already have n_x and n_y members we update the distances using

$$D_{(x,y),k} = \left(\frac{n_x}{n_x + n_y}\right) D_{x,k} + \left(\frac{n_y}{n_x + n_y}\right) D_{y,k}$$

Step 1 – Find the smallest entry in the distance matrix

d(i,j)	*A*	*B*	*C*	*D*	*E*	*F*
A	-					
B	2	-				
C	4	4	-			
D	4	4	**2**	-		
E	7	7	7	7	-	
F	5	5	5	5	6	-
G	8	8	8	8	9	5

Step 2 - Cluster taxa *A* and *B*, form a new internal node *I*
Calculate the lengths of the new edges *d(A,I)=d(B,I)=1/2 d(A,B)=1*

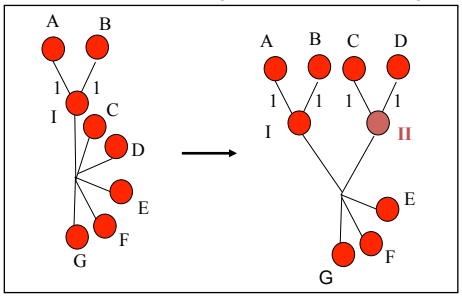


Step 3 – Update the distance matrix
d(C,I) = ½(d(A,C) + d(B,C))
= 4
etc...

Step 1 – Find the smallest entry in the distance matrix

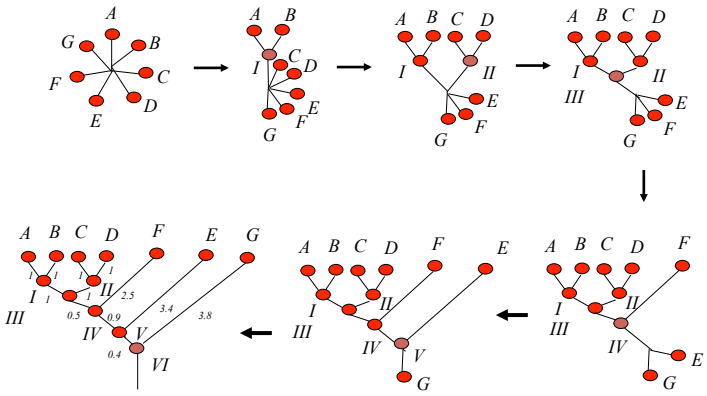
d(i,j)	*I (A+B)*	*C*	*D*	*E*	*F*
I (A+B)	-				
C	4	-			
D	4	**2**	-		
E	7	7	7	-	
F	5	5	5	6	-
G	8	8	8	9	5

Step 2 - Cluster taxa *C* and *D*, form a new internal node *II*
Calculate the lengths of the new edges *d(C,II)=d(D,II)=1/2 d(C,D)=1*



Step 3 – Update the distance matrix
d(I,II)=1/2(d(I,C)+d(I,D))
= 4
d(E,II) = ½(d(E,C) + d(E,D))
= 7
etc...

And so on...



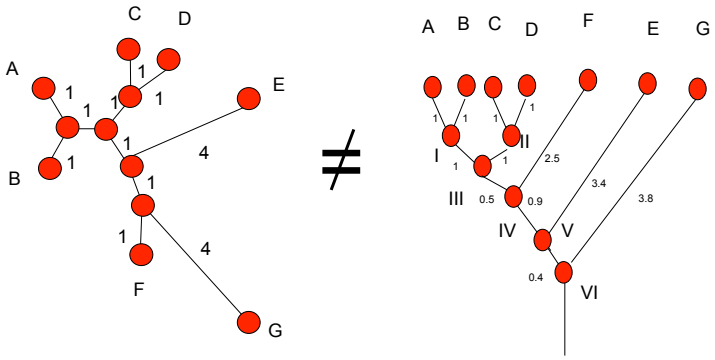
...until we have a rooted tree.

But, is it the right tree?

UPGMA is not consistent for additive distances

d(i,j)	*A*	*B*	*C*	*D*	*E*	*F*
A	-					
B	2	-				
C	4	4	-			
D	4	4	**2**	-		
E	7	7	7	7	-	
F	5	5	5	5	6	-
G	8	8	8	8	9	5

The tree that matches the distances is not recovered by UPGMA.



Inconsistency

- When a method is given "perfect" data but still gets the wrong tree it is said to be **inconsistent**.
- UPGMA is inconsistent for data that isn't ultrametric (clock-like).
- Next we'll look at a method that is consistent for any additive data.

Constructing Phylogenetic Trees

There are two main methods of constructing phylogenetic trees:

* **distance-based methods** such as UPGMA and neighbour-joining,

* **parsimony-based methods** such as maximum parsimony, maximum likelihood, or Bayesian inference.

Neighbor joining is a bottom-up (agglomerative) clustering method developed originally by Naruya Saitou and Masatoshi Nei in 1987.

Neighbor-joining (NJ)

NJ works by progressively clustering taxa until all the taxa form an unrooted tree.

1. Rather than using the distance matrix directly to determine which taxa should be clustered at each stage, NJ uses the "cost matrix" Q where

   $Q(i,j) = (N-2)d(i,j) - R(i) - R(j)$

   N is the number of taxa.

   $R(i)$ is the sum of the ith row in the distance matrix.

   $R(j)$ is the sum of the jth row in the distance matrix.

2. Find the smallest entry in the Q matrix, say $Q(x,y)$.

NJ Example

D=

	Cat	Dog	Rat
Dog	3		
Rat	4	5	
Cow	6	7	6

Step 1 →

Q=

	Cat	Dog	Rat
Dog	-22		
Rat	-20	-20	
Cow	-20	-20	-22

← Step 2

R(cat) = 13
R(dog) = 15
R(rat) = 15
R(cow) = 19

e.g. Q(cat,dog) = (4-2)x3 – 13 – 15 = -22
Q(cat,rat) = (4-2)x4 – 13 – 15 = -20

NJ tree distance updates

3. Form a new internal node, z, that is a parent to x and y and calculate the edge lengths from z to x and z to y.

$d(x,z) = 1/(2(N-2))[(N-2)d(x,y) + R(x) - R(y)]$

$d(y,z) = d(x,y) - d(x,z)$

4. Update the distance matrix to all other nodes (w)

$d(w,z) = ½ (d(x,w) + d(y,w) - d(x,y))$

REPEAT until only two things are left to be joined.

NJ Example

D=

	Cat	Dog	Rat
Dog	3		
Rat	4	5	
Cow	6	7	6

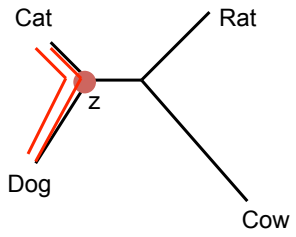
Step 1 →

Q=

	Cat	Dog	Rat
Dog	-22		
Rat	-20	-20	
Cow	-20	-20	-22

← Step 2

Step 3
d(cat,z) = ¼[2d(cat,dog) + R(cat) – R(dog)]
= ¼ [6 + 13 – 15]
= 1
d(dog,z) = 3-1
= 2



NJ Example

D=

	Cat	Dog	Rat
Dog	3		
Rat	4	5	
Cow	6	7	6

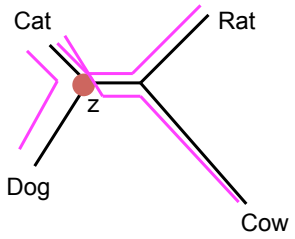
Step 1 →

Q=

	Cat	Dog	Rat
Dog	-22		
Rat	-20	-20	
Cow	-20	-20	-22

← Step 2

Step 3
d(cat,z) = ¼[2d(cat,dog) + R(cat) – R(dog)]
= ¼ [6 + 13 – 15]
= 1
d(dog,z) = 3-1
= 2



NJ Example

D=

	Cat	Dog	Rat
Dog	3		
Rat	4	5	
Cow	6	7	6

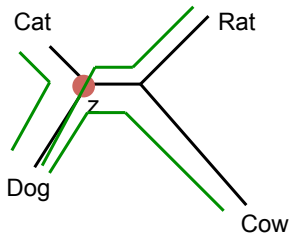
Step 1 →

Q=

	Cat	Dog	Rat
Dog	-22		
Rat	-20	-20	
Cow	-20	-20	-22

← Step 2

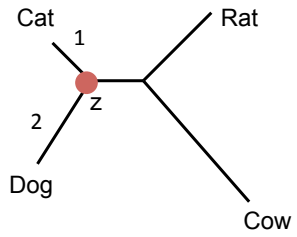
Step 3
d(cat,z) = ¼[2d(cat,dog) + R(cat) – R(dog)]
= ¼ [6 + 13 – 15]
= 1
d(dog,z) = 3-1
= 2



Step 4
d(z,rat) = ½ [d(cat,rat) + d(dog,rat) – d(cat,dog)]
= ½ [4 + 5 – 3]
= 3

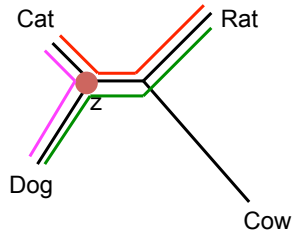
d(z,cow) = ½ [6 + 7 – 3]
= 5

	Z	Rat
Rat	3	
Cow	5	6



Step 4
d(z,rat) = ½ [d(cat,rat) + d(dog,rat) – d(cat,dog)]
= ½ [4 + 5 – 3]
= 3

d(z,cow) = ½ [6 + 7 – 3]
= 5

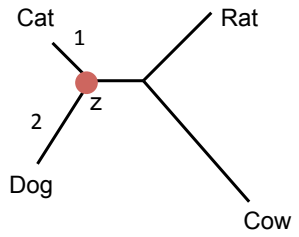


Step 4
d(z,rat) = ½ [d(cat,rat) + d(dog,rat) – d(cat,dog)]
= ½ [4 + 5 – 3]
= 3

d(z,cow) = ½ [6 + 7 – 3]
= 5

D=

	Z	Rat
Rat	3	
Cow	5	6



Global vs Local methods

- UPGMA and NJ are **local** construction methods. At each step they pick they best pair of taxa to cluster, once a decision is made it cannot be unmade. This makes these methods very fast.
- There are also **global** methods for making trees based on distances. These evaluate an optimality criterion on each possible tree and then pick the tree with the best score. Because the number of trees grows very quickly with the number of taxa, these methods are slow.
- Examples of global methods for distance data include **least squares** and **minimum evolution**. Nearly all character methods are global in nature.

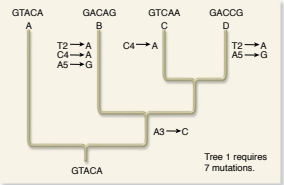
Constructing Phylogenetic Trees

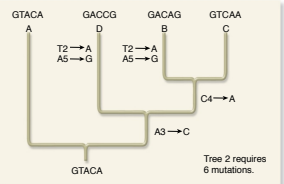
There are two main methods of constructing phylogenetic trees:

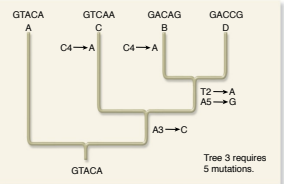
* **distance-based methods** such as UPGMA and neighbour-joining,

* **character-based methods** such as maximum parsimony, maximum likelihood, or Bayesian inference.

Parsimony is a 'less is better' concept of frugality, economy, stinginess or caution in arriving at a hypothesis or course of action. The word derives from Latin *parsimonia*, from *parcere*: **to spare**.







Maximum Parsimony

• There is no algorithm to quickly *generate* the most-parsimonious tree.

•All possible trees are determined for each position of the sequence alignment

• Each tree is given a score based on the number of evolutionary step needed to produce said tree

• The most parsimonious tree is the one that has the ***fewest*** evolutionary changes for all sequences to be derived from a common ancestor

• Usually several equally parsimonious trees result from a single run.

The principle of parsimony

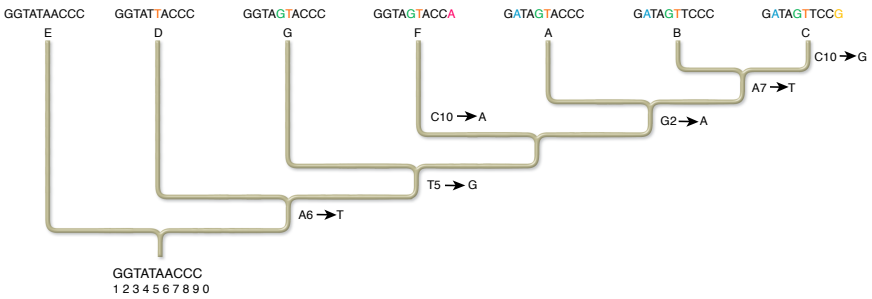
Derive a phylogenetic tree for seven sequences:

```
   1234567890
A: GATAGTACCC
B: GATAGTTCCC
C: GATAGTTCCG
D: GGTATTACCC
E: GGTATAACCC
F: GGTAGTACCA
G: GGTAGTACCC
```

Positions with variation:
2: 3A, 5G
5: 5G, 2T
6: 1A, 6T
7: 5A, 2T
10: 5C, 1G, 1A

Notice that the patterns of variation are different, for example between columns 5 and 7 – different sequences have the "minor" allele.

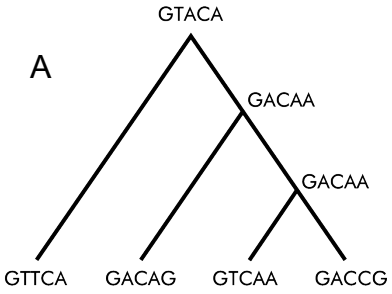
Cladogram relationship involving sequences from seven species



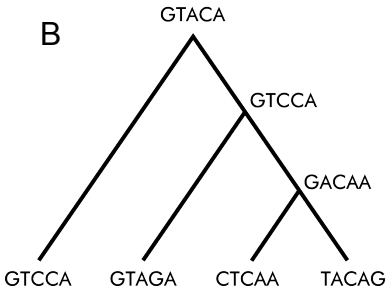
Notice that given a tree, the criteria of parsimony is easy to score (# of changes implied by the topology).

Which of these trees follows the principle of parsimony?

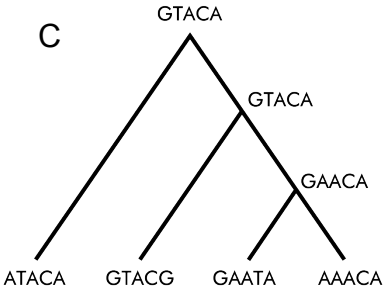
A



B



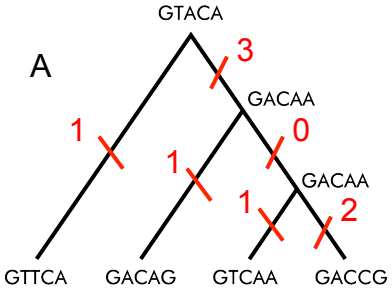
C



Which of these trees follows the principle of parsimony?

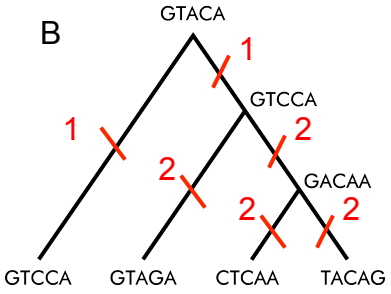
Parsimony looks for the tree that requires the fewest number of changes.

A



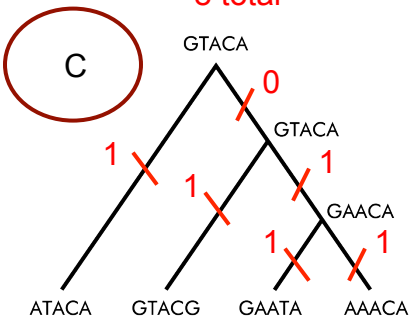
= 8 total

B



= 10 total

C



= 5 total

Constructing Phylogenetic Trees

There are two main methods of constructing phylogenetic trees:

* **distance-based methods** such as UPGMA and neighbour-joining,

* **character-based methods** such as maximum parsimony, maximum likelihood, or Bayesian inference.

Maximum Likelihood methods (and related Bayesian approaches) determine the tree topology, branch lengths, and parameters of the evolutionary model that maximize the probability of observing the sequences at hand. By many considered the most accurate approach, however undoubtedly the most computationally expensive.

Methods in Phylogenetic Reconstruction

Maximum Likelihood

• Creates all possible trees like Maximum Parsimony method but instead of retaining trees with shortest evolutionary steps……

• Employs a model of sequence evolution (a method of converting characters to conditional probabilities)

• Each tree generated is calculated for the probability that it reflects each position of the sequence data.

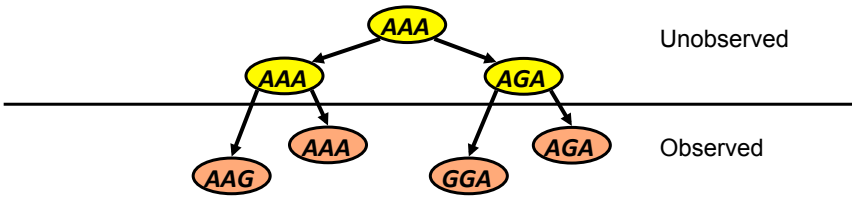
• Calculation is repeated for all nucleotide sites

• Finally, the tree with the best probability is shown as the maximum likelihood tree - usually only a single tree remains

• It is a more realistic tree estimation because it can capture distinct evolutionary rates of character conversion (example: transitions versus transversions)

Maximum Likelihood Approach

Consider the phylogenetic tree to be a stochastic process.



The probability of transition from character a to character b is given by parameters $\theta_{b|a}$. The probability of letter a in the root is q_a. These parameters are defined via rates of change per time unit times the time unit.

Given the complete tree, the probability of data is defined by the values of the $\theta_{b|a}$'s and the q_a's.

Probabilistic Methods

- The phylogenetic tree represents a generative probabilistic model (like HMMs) for the observed sequences.
- Background probabilities: $q(a)$
- Mutation probabilities: $P(b \mid a, t)$
- Models for evolutionary mutations
  - Jukes Cantor
  - Kimura 2-parameter model
- Such models are used to derive the probabilities

Substitution models: general framework

The substitution probability matrix can be written thus:

$$P(b \mid a, t) = P_t = \begin{bmatrix} P_{AA} & P_{AC} & P_{AG} & P_{AT} \\ P_{CA} & P_{CC} & P_{CG} & P_{CT} \\ P_{GA} & P_{GC} & P_{GG} & P_{GT} \\ P_{TA} & P_{TC} & P_{TG} & P_{TT} \end{bmatrix}$$

where P_{AC} is the prob that site that started with A had nuc. C after time t.

Substitution models: general framework

- The base composition of the sequences can be represented by a vector:

  $q(a) = [p_A, p_C, p_G, p_T]$

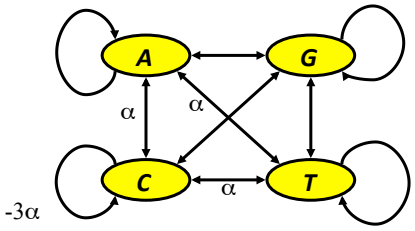
Where the p_A is the equilibrium frequency of A.

In some models, $p_A=p_C=p_G=p_T$ so all 4 bases are proportional, but in other models they can vary.

The Jukes-Cantor model (1969)

We need to develop a formula for DNA evolution via Prob(b | a, t) where b and a are taken from {A, C, G, T} and t is the time length.

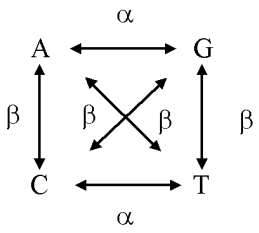
Jukes-Cantor assumes equal rate of change:



$$P_t = \begin{array}{c} \\ A \\ C \\ G \\ T \end{array} \begin{array}{c} \begin{array}{cccc} A & C & G & T \end{array} \\ \begin{bmatrix} -3\alpha & \alpha & \alpha & \alpha \\ \alpha & -3\alpha & \alpha & \alpha \\ \alpha & \alpha & -3\alpha & \alpha \\ \alpha & \alpha & \alpha & -3\alpha \end{bmatrix} \end{array}$$

Kimura's K2P model (1980)

Jukes-Cantor model does not take into account that transitions rates (between purines) A↔G and (between pyrmidine) C↔T are different from transversions rates of A↔C, A↔T, C↔G, G↔T.

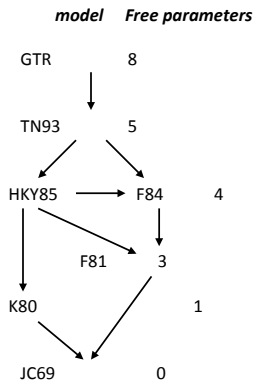


Kimura used a different rate matrix:

$$P_t = \begin{array}{c} \\ A \\ C \\ G \\ T \end{array} \begin{array}{c} \begin{array}{cccc} A & C & G & T \end{array} \\ \begin{bmatrix} -2\beta-\alpha & \beta & \alpha & \beta \\ \beta & -2\beta-\alpha & \beta & \alpha \\ \alpha & \beta & -2\beta-\alpha & \beta \\ \beta & \alpha & \beta & -2\beta-\alpha \end{bmatrix} \end{array}$$

The phylogenetic inference process

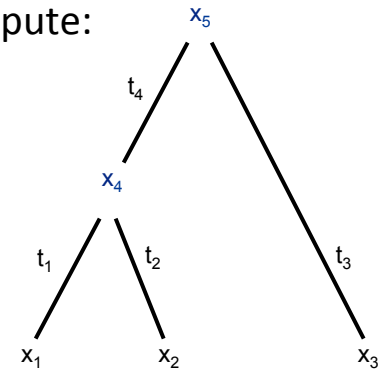
- There are many models of sequence evolution.
- Increasing model complexity improves the fit to the data but also increases variance in estimated parameters.
- Model selection strategies attempt to find the appropriate level of complexity on the basis of the available data.



- Model complexity can often lead to computational intractability.

Probabilistic Approach

- Given P,q, the tree topology and branch lengths, we can compute:



$$P(x_1, x_2, x_3, x_4, x_5 \mid T, t) =$$

$$q(x_5)p(x_4 \mid x_5, t_4)p(x_3 \mid x_5, t_3)p(x_1 \mid x_4, t_1)p(x_2 \mid x_4, t_2)$$

Computing the Tree Likelihood

- We are interested in the probability of **observed** data given tree and branch "lengths":

$$P(x_1, x_2, x_3 \mid T, t) = \sum_{x_4, x_5} P(x_1, x_2, x_3, x_4, x_5 \mid T, t)$$

- Computed by summing over internal nodes
- This can be done efficiently using a tree upward traversal pass.

Maximum Likelihood (ML)

- Score each tree by
  - Assumption of independent positions "m"
- Branch lengths *t* can be optimized
  - Gradient Ascent
  - EM
- We look for the highest scoring tree
  - Exhaustive
  - Sampling methods (Metropolis)

Maximum likelihood

- Advantages
  - Statistically well founded
  - Based on a model of evolution
  - Evaluates different topologies
  - Uses all sequence information
  - Often yields estimates that have lower variance than other methods
- Disadvantages
  - Very slow (computationally intensive)
  - Dependent on the model of evolution used

Likelihood: Choose the tree that makes the data the most likely

Equivalent to maximizing $P(D \mid T)$

Bayesian: Choose the most probable tree (tree with the highest posterior probability)

Equivalent to maximizing: P(T|D)

$$P(T \mid D) = \frac{P(D \mid T)P(T)}{P(D)}$$

Comparison of Methods

Distance	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