Phylogeny Reconstruction
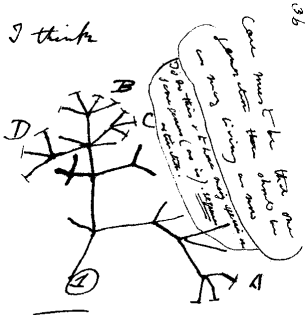
Center for Integrative Bioinformatics Vienna (CIBIV)
Max F. Perutz Laboratories (MFPL)
Vienna, Austria
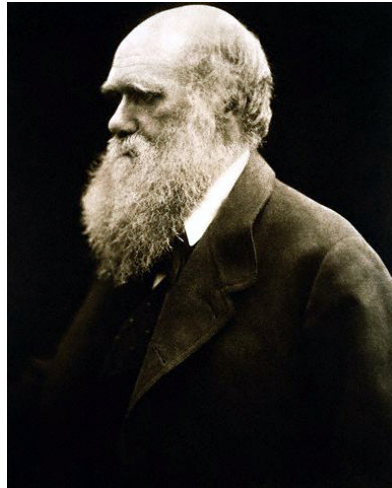
Notes

Charles Darwin: Evolutionary Relationships

Charles Darwin (1809-1882)

Notes

Ernst Haeckel: Evolutionary Trees
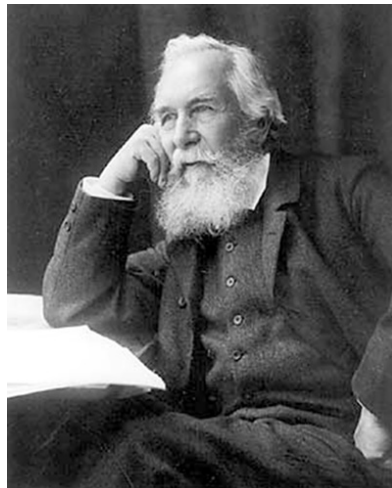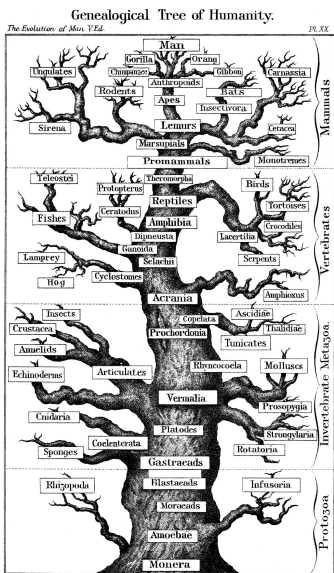
Ernst Haeckel (1834-1919)

Notes

Theodosius Dobzhansky: The Light of Evolutionary

Theodosius Dobzhansky (1900-1975)

Nothing in Biology Makes Sense Except in the Light of Evolution.
Dobzhansky, 1973

Some Notation

Note: branch = edge = split, external node = leaf = taxon are used interchangebly.

Main Types of Phylogenetic Methods

Data	Method	Evaluation Criterion
Characters (Alignment)	**Maximum Parsimony**	Parsimony
	Statistical Approaches: Likelihood, Bayesian	Evolutionary Models
Distances	**Distance Methods**	

William of Ockham: The Law of Parsimony

Occam's Razor (law of parsimony) states:

> *Pluralitas non est ponenda sine necessitate.*

> *Plurality should not be posited without necessity.*

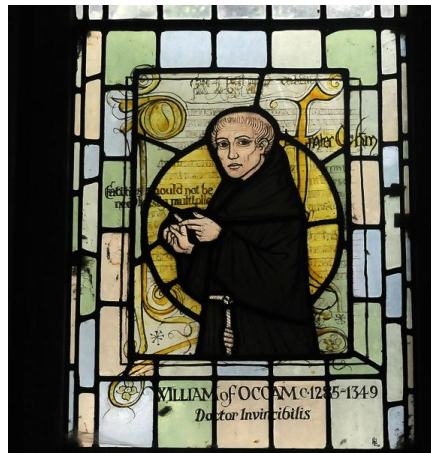The principle gives precedence to simplicity; of two competing theories, the simplest explanation of an entity is to be preferred.

William of Ockham (1285-1347/49)

Maximum Parsimony

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

Tree a: ((1,2),(3,4))

Tree b: ((1,3),(2,4))

Tree c: ((1,4),(2,3))

Parsimony Informative Sites

We have seen that not all variable columns are informative for the parsimony reconstruction.

To be an *informative site* for the parsimony principle the column has to contain at least two different character states, and for at least two of these states have to occur at least twice.

Maximum Parsimony: Fitch's (1970) algorithm

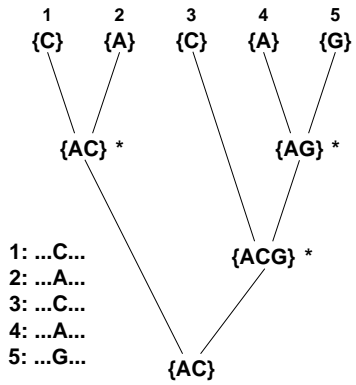| 1	2	3	4	5
{C}	{A}	{C}	{A}	{G}

{AC} *   {AG} *

1: ...C...
2: ...A...
3: ...C...
4: ...A...
5: ...G...

{ACG} *

{AC}

Note: We need 1 substitution per union in tree T (tree-length = substitutions needed).

❶ Initialize state set S_k at each leaf k with the characters from the alignment.

❷ Construct the state sets of all internal leaves in a post-order-traversal starting a the root.

❸ Let k be the current node and i, j its decendents, then build the intersection of S_i and S_j:
   - If $S_i \cap S_j$ non-empty: set $S_k = S_i \cap S_j$,
   - if $S_i \cap S_J$ empty: set $S_k = S_i \cup S_j$ and increase the tree-length by 1.

❹ Continue with the traversal until you have reconstructed the state set S_{root} of the root of T. If we have a sequence for the root, repeat Step 3 for the its character and S_{root}.

Notes

Maximum Parsimony: Objective Function

Aim: Find the tree T that minimizes the following function:

$$MP(T) = \sum_{k=1}^{B} \sum_{j=1}^{L} w_j \cdot \text{diff}(x_{k'j}, x_{k''j}).$$

diff: Scoring matrix for substitutions (often 1 for changes, 0 otherwise)

w_j: alignment-specific weight (often 1)

L: alignment length

B: number of edges in T

k' and k'': endnodes of edge k.

Notes

Maximum Parsimony: Better Substitution Costs

diff with substitution cost 1 and more elaborate costs:

	A	G	C	T
A	-	1	1	1
G	1	-	1	1
C	1	1	-	1
T	1	1	1	-

	A	G	C	T
A	-	1	5	5
G	1	-	5	5
C	5	5	-	1
T	5	5	1	-

Notes

How to find the Most Parsimonious Tree?

Ideally we would evaluate all trees and take the one(s) with the lowest
tree-length.
However, there are too many trees. This problem affects almost every
method that aims to find trees with optimal score.
So we need other strategies (which we will see later).

Problems with Parsimony

- Parsimony is often considered model-free. This is not entirely correct.
- One has no choice of a model, but nevertheless the algorithm assumes
  a very simple model.
- Parsimony assumes that substitutions are rare and that
  back-mutations do not occur.
- Although this was often true for morphological data, it is certainly not
  true for distantly related DNA sequences which only have four
  character states.

Distance-based methods

seq 1	A	G	C	T	T	A	C	C	T	G	T	T	A	C	T
seq 2	C	G	T	A	A	A	T	T	T	C	C	C	G	A	T
seq 3	C	G	C	A	A	G	T	T	T	C	C	C	G	A	T
seq 4	C	A	C	T	T	A	T	T	A	G	T	C	A	A	C

$$\Downarrow (d_{ij})_{i,j=1,\dots,4}$$

	seq 1	seq 2	seq 3	seq 4
seq 1	0	11	11	8
seq 2	11	0	2	10
seq 3	11	2	0	9
seq 4	8	10	9	0

Distance Methods: Aim
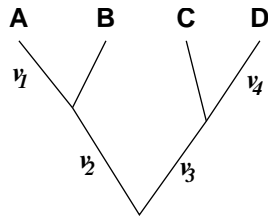
Aim: Find branch lengths v_b such that the sum of the branch lengths connecting any two leaves gets close to the measured distances between all pairs of leaves. That is, for instance
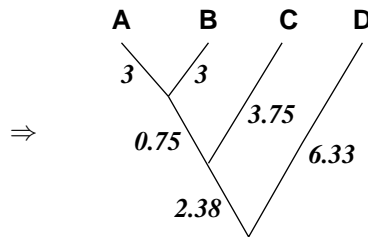
$$d_{A,D}^{measured} = v_1 + v_2 + v_3 + v_4$$

Distance Methods: UPGMA

One possibility are clustering methods like UPGMA = Unweighted Pair Group Methods using Arithmetic means.

	A	B	C	D
A	0	6	7	13
B	6	0	8	14
C	7	8	0	11
D	13	14	11	0

$\Rightarrow$

Distance Methods: Clustering Methods

- Clustering methods well, if sequences evolve according to a molecular clock
- or equivalently: if the ultrametric inequality

$$D_{AB} \leq \max D_{AC}, D_{BC}$$

  holds for each triple (A, B, C).
- Then the data is ultrametric, that means according to a molecular clock.

Distance Methods: Four-point Condition

- On the other hand, a distance matrix D can only be presented as a tree, if and only if the Four-Point-Condition

$$d_{uv} + d_{xz} \leq max(d_{ux} + d_{vz}, d_{uz} + d_{vx})$$

  holds for all orderings of four taxa u, v, x, z.

- Or equivalently:
- For all sets of four taxa there exists a labelling of the elements, say A, B, C, D such that
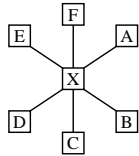
$$d_{AB} + d_{CD} \leq d_{AC} + d_{BD} = d_{AD} + d_{BC}).$$

- This means, that the distance matrix is additive, i.e. it fits fits one tree.

Distance Methods: Neighbor Joining (NJ)

A widely used distance method is Neighbor-Joining:

1. begin with a start tree:
2. compute for each pair $1, 2$ the net-divergence

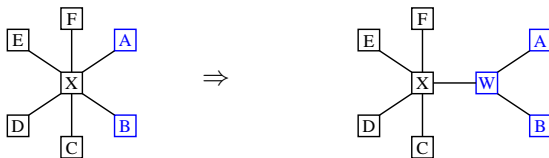$$\frac{1}{2(N-2)} \sum_{k=3}^{N} (D_{1k} + D_{2k}) + \frac{1}{2} D_{12} + \frac{1}{N-2} \sum_{3 \leq i < j} D_{ij}.$$

3. Choose the pair (A, B) that minimizes this equation.

Distance Methods: Neighbor Joining (NJ)

4. cluster (A,B) and define an interior node W:

5. compute the branch lengths for the external edges:

$$v_{AW} = \frac{1}{2}\left(D_{AB} + \frac{1}{m-2} \sum_{k=1}^{m} (D_{Ak} - D_{Bk}) \right)$$

$$v_{BW} = \frac{D_{AB}}{2} - v_{AW}.$$

Distance Methods: Neighbor Joining (NJ)

6 compute distance W to the remaining m-2 leaves:
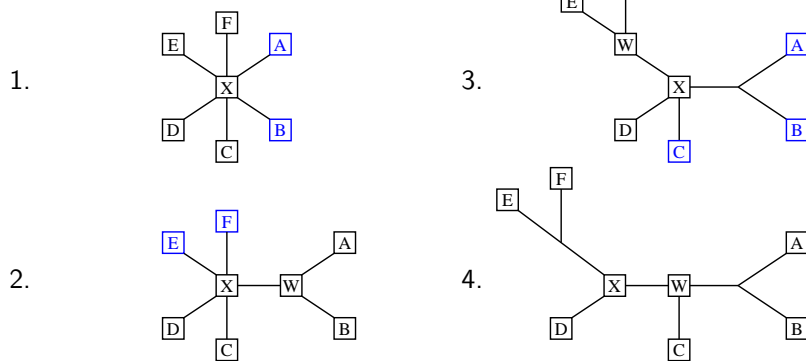
$$D_{Wk} = \frac{1}{2}\left(d_{Ak} + D_{Bk} - D_{AB}\right))$$

7 continue with the reduced set of leaves

Distance Methods: The NJ Tree Step-by-step

The algorithm is repeated until the tree is fuly resolved:

Distance Methods: Least Square

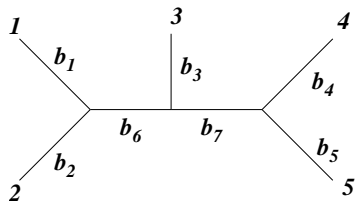- Aim: Find the tree T that minimizes the squared error:

$$E(T) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} (d_{ij} - \rho_{ij})^2$$

where $\rho_{i,k}$ is the length of the unique path in T connecting leaves i and k.

Distance Methods: Least Square

$$d_{12}=b_1+b_2$$
$$d_{13}=b_1+\quad b_3+\qquad b_6$$
$$d_{14}=b_1+\qquad b_4+\quad b_6+b_7$$
$$d_{15}=b_1+\qquad\qquad b_5+b_6+b_7$$
$$d_{23}=\quad b_2+b_3+\qquad b_6$$
$$d_{24}=\quad b_2+\quad b_4+\quad b_6+b_7$$
$$d_{25}=\quad b_2+\qquad b_5+b_6+b_7$$
$$d_{34}=\qquad b_3+b_4+\qquad\quad b_7$$
$$d_{35}=\qquad b_3+\quad b_5+\quad b_7$$
$$4_{45}=\qquad\qquad b_4+b_5$$

Distance Methods: Least Square

$$d_{12}=b_1+b_2$$
$$d_{13}=b_1+\quad b_3+\qquad b_6$$
$$d_{14}=b_1+\qquad b_4+\quad b_6+b_7$$
$$d_{15}=b_1+\qquad\qquad b_5+b_6+b_7$$
$$d_{23}=\quad b_2+b_3+\qquad b_6$$
$$d_{24}=\quad b_2+\quad b_4+\quad b_6+b_7$$
$$d_{25}=\quad b_2+\qquad b_5+b_6+b_7$$
$$d_{34}=\qquad b_3+b_4+\qquad\quad b_7$$
$$d_{35}=\qquad b_3+\quad b_5+\quad b_7$$
$$4_{45}=\qquad\qquad b_4+b_5$$

$$d = Ab$$

d: distance vector, b: branch length vector

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

Distance Methods: Least Square Estimate

For

$$d = Ab$$

$$\begin{pmatrix} \rho_{12} \\ \rho_{13} \\ \rho_{14} \\ \rho_{15} \\ \rho_{23} \\ \rho_{24} \\ \rho_{25} \\ \rho_{34} \\ \rho_{35} \\ \rho_{45} \end{pmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{pmatrix}$$

with d, ρ: distance vectors, b: branch length vector
If the distances are additive, then $\rho_{ij} = d_{ij}$ and we can solve it directly.

Distance Methods: Least Square Estimate

For
$$d = Ab$$

If they are not additive we try to minimize the square differences (i.e., least square error) between d and ρ:

$$E = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} (d_{ij} - \rho_{ij})^2$$

The least square solution $\hat{b}$ for this can be obtained mathematically using Matrix operations:

$$\hat{b} = (A^t A)^{-1} A^t d$$

Notes

How to get distances?

Distances can be computed in various way. . .
Usually via Maximum Likelihood.

Notes

Introduction: ML on Coin Tossing

Given a box with 3 coins of different fairness $\left(\frac{1}{3}, \frac{1}{2}, \frac{2}{3} \text{ heads}\right)$

We take out one coin an toss 20 times:
$$H, T, T, H, H, T, T, T, T, H, T, T, H, T, H, T, T, H, T, T$$

| **Probability**		**Likelihood**
$p(k \text{ heads in } n \text{ tosses}\lvert\theta)$	$\equiv$	$L(\theta\lvert k \text{ heads in } n \text{ tosses})$
	$=$	$\binom{n}{k}\theta^k(1-\theta)^{n-k}$
		(here binomial distribution)

Aim: The ML approach seaches for that parameter set θ for the generating process which maximizes the probability of our given data.

Hence, *"likelihood flips the probability around."*

Notes

Notes

Introduction: ML on Coin Tossing (Estimate)

coint tossing: 7 heads, 13 tails
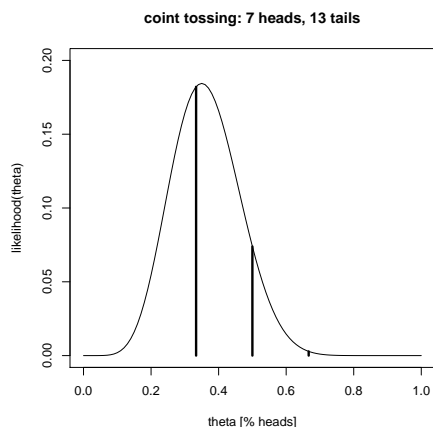
Three coin case

$$L(\theta | 7 \text{ heads in } 20) = \binom{20}{7}\theta^7(1-\theta)^{13}$$

for each coin $\theta \in \left\{\frac{1}{3}, \frac{1}{2}, \frac{2}{3}\right\}$

For infinitely many coins
$\theta = (0...1)$

ML estimate: $L(\hat{\theta}) = 0.1844$ where coin shows $\hat{\theta} = 0.35$ heads

Notes

From Coins to Phylogenies?

While the coin tossing example might look easy, in phylogenetic analysis, the parameter (set) θ comprises:

- evolutionary model
- its parameters
- tree topology
- its branch lengths

That means, a high dimensional optimization problem.
Hence, some parameters are often estimated/set separately.

Notes

Modeling Evolution

- Evolution is usually modeled as a

    stationary, time-reversible Markov process.

- What does that mean?

Notes

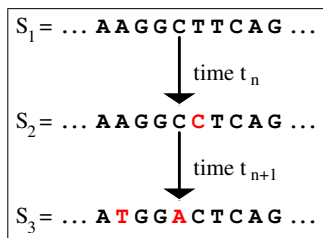## Assumptions on Evolution

Markov Process

The (evolutionary) process evolves without memory, i.e. sequence S_2 mutates to S_3 during time t_{n+1} independent of state of S_1.

$S_1 = \ldots \textbf{A A G G C T T C A G} \ldots$

$\downarrow$ time t_n

$S_2 = \ldots \textbf{A A G G C C T C A G} \ldots$

$\downarrow$ time t_{n+1}

$S_3 = \ldots \textbf{A T G G A C T C A G} \ldots$

Assumptions on Evolution

Stationary:
The overall character frequencies π_j of the nucleotides or amino acids are in an equilibrium and remain constant.

Time-Reversible:
Mutations in either direction are equally likely

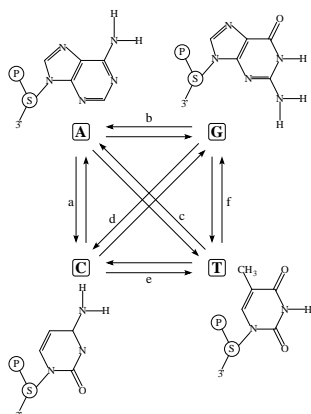$$\pi_i \cdot P_{ij}(t) = P_{ji}(t) \cdot \pi_j$$

This means a mutation is as likely as its back mutation.

$$P(i \to j) = P(i \leftarrow j) \qquad \text{(JC69)}$$

Substitution Models

Evolutionary models are often described using a substitution rate matrix R and character frequencies Π. Here, 4×4 matrix for DNA models:
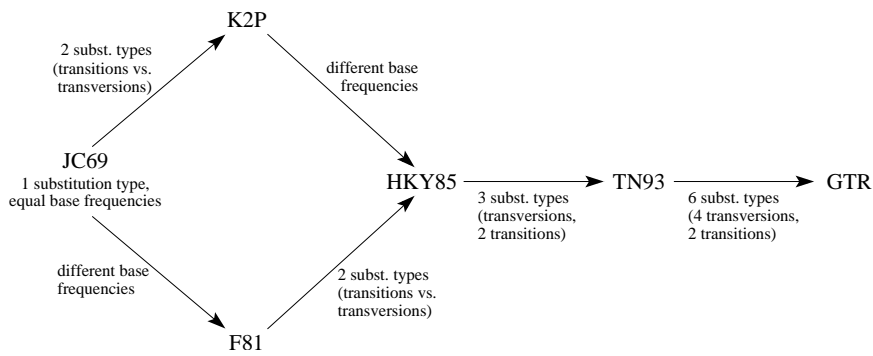

$$R = \begin{pmatrix} & A & C & G & T \\ & - & a & b & c \\ & a & - & d & e \\ & b & d & - & f \\ & c & e & f & - \end{pmatrix}$$

$$\Pi = (\pi_A, \pi_C, \pi_G, \pi_T)$$

From R and Π we reconstruct a substitution probability matrix P, where $P_{ij}(t)$ is the probability of changing $i \to j$ in time t.

Relations between DNA models

K2P

2 subst. types
(transitions vs.
transversions)

different base
frequencies

JC69
1 substitution type,
equal base frequencies

HKY85 → TN93 → GTR

3 subst. types
(transversions,
2 transitions)

6 subst. types
(4 transversions,
2 transitions)

different base
frequencies

2 subst. types
(transitions vs.
transversions)

F81

Further modification:
rate heterogeneity: invariant sites, Γ-distributed rates, mixed.

Protein Models

Generally this is the same for protein sequences, but with 20×20
matrices. Some protein models are:

- Poisson model ("JC69" for proteins, rarely used)
- Dayhoff (Dayhoff *et al.*, 1978, general matrix)
- JTT (Jones *et al.*, 1992, general matrix)
- WAG (Whelan & Goldman, 2000, more distant sequences)
- VT (Müller & Vingron, 2000, distant sequences)
- mtREV (Adachi & Hasegawa, 1996, mitochondrial sequences)
- cpREV (Adachi *et al.*, 2000, cloroplast sequences)
- mtMAM (Yang *et al.*, 1998, Mammalian mitochondria)
- mtART (Abascal *et al.*, 2007, Arthropod mitochondria)
- rtREV (Dimmic *et al.*, 2002, reverse transcriptases)
- . . .
- ~~BLOSUM 62 (Henikoff & Henikoff, 1992)~~ → for database searching

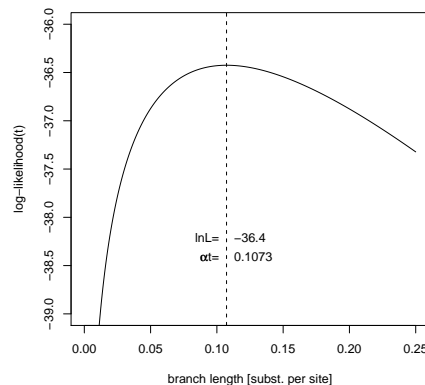---

Computing ML Distances Using $\mathbf{P_{ij}(t)}$

The Likelihood of sequence s evolving to s' in time t:

$$L(t|s \rightarrow s') = \prod_{i=1}^{m} \left(\Pi(s_i) \cdot P_{s_i s_i'}(t) \right)$$

Likelihood surface for two
sequences under JC69:

GATCCTGAGAGAAATAAAC $= s'$
GGTCCTGACAGAAATAAAC $= s$

Note: we do not compute the
probability of the distance t
but that of the data $D = \{s, s'\}$.

lnL= −36.4
αt= 0.1073

Computing Likelihood Values for Trees

Given a tree with branch lengths and sequences for all nodes, the computation of likelihood values for trees is straight forward.
Unfortunately, we usually have no sequences for the inner nodes (ancestral sequences).
Hence we have to evaluate every possible labeling at the inner nodes:

$$L\left(\begin{smallmatrix}C\\G\end{smallmatrix}\!\!\times\!\!\begin{smallmatrix}C\\C\end{smallmatrix}\right) = L\left(\begin{smallmatrix}C\\G\end{smallmatrix}\!\!\overset{A\ A}{\times}\!\!\begin{smallmatrix}C\\C\end{smallmatrix}\right) + L\left(\begin{smallmatrix}C\\G\end{smallmatrix}\!\!\overset{A\ C}{\times}\!\!\begin{smallmatrix}C\\C\end{smallmatrix}\right) + \cdots + L\left(\begin{smallmatrix}C\\G\end{smallmatrix}\!\!\overset{G\ G}{\times}\!\!\begin{smallmatrix}C\\C\end{smallmatrix}\right) + \cdots + L\left(\begin{smallmatrix}C\\G\end{smallmatrix}\!\!\overset{T\ T}{\times}\!\!\begin{smallmatrix}C\\C\end{smallmatrix}\right)$$

for every column in the alignment... but there is a faster algorithm.

Likelihoods of Trees (Single alignment column, given tree)

For a single alignment column and a given tree:

Likelihoods of nucleotides i at inner nodes:

$$L_5(i) = [P_{iC}(d_1) \cdot L(C)] \cdot [P_{iG}(d_2) \cdot L(G)]$$

$$L_6(i) = \prod_{v=\{3,4\}} \left[\sum_{j=\{ACGT\}} P_{ij}(d_v) \cdot L_v(j) \right]$$

Site-Likelihood of an alignment column k:

$$L^{(k)} = \sum_{i=\{ACGT\}} \pi_i \cdot L_6(i) = 0.005489$$

with all $d_x = 0.1$ and $P_{ij}(0.1) = \begin{cases} .91 & i \neq j \\ .03 & i = j \end{cases}$ (JC)

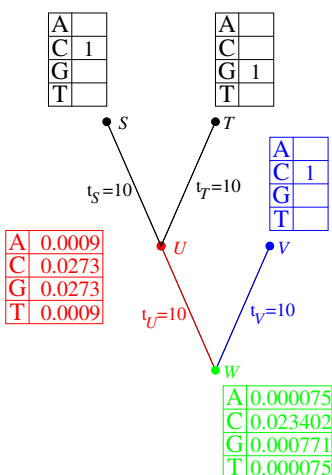# Likelihoods of Trees (Single column $\begin{smallmatrix}C\\G\\C\end{smallmatrix}$, given tree)

Likelihoods of nucleotides at inner nodes:

$$L_U(i) = [P_{iC}(10) \cdot L(C)] \cdot [P_{iG}(10) \cdot L(G)]$$

$$L_W(i) = \left[\sum_{u=ACGT} P_{iu}(t_U) \cdot L_U(u) \right] \cdot$$

$$\left[\sum_{v=ACGT} P_{iv}(t_V) \cdot L_V(v) \right]$$

Site-Likelihood of an alignment column k:
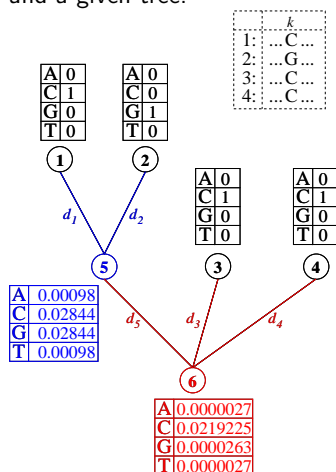
$$L^{(k)} = \sum_{i=ACGT} \pi_i \cdot L_W(i) = 0.024323$$

Likelihoods of Trees (Single alignment column, given tree)

For a single alignment column and a given tree:

Likelihoods of nucleotides i at inner nodes:

$$L_5(i) = [P_{iC}(d_1) \cdot L(C)] \cdot [P_{iG}(d_2) \cdot L(G)]$$

$$L_6(i) = \prod_{v=\{2,3,4\}} \left[\sum_{j=\{ACGT\}} P_{ij}(d_v) \cdot L_v(j) \right]$$

Site-Likelihood of an alignment column k:

$$L^{(k)} = \sum_{i=\{ACGT\}} \pi_i \cdot L_6(i) = 0.005489$$

with all $d_x = 0.1$ and $P_{ij}(0.1) = \begin{cases} .91 & i \neq j \\ .03 & i = j \end{cases}$ (JC)

Likelihoods of Trees (multiple columns)
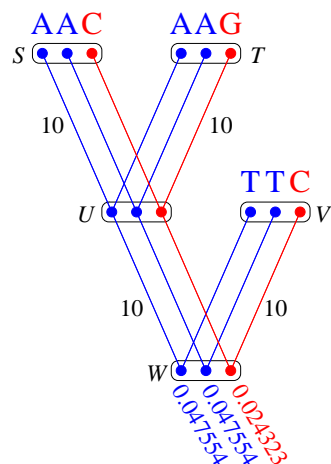
Considering this tree with $n = 3$ sequences of length $m = 3$ the tree likelihood of this tree is

$$\mathcal{L}(T) = \prod_{k=1}^{m} L^{(k)} \quad = \quad 0.047554^2 \cdot 0.024323$$
$$= \quad 0.000055$$

or the log-likelihood

$$\ln \mathcal{L}(T) = \sum_{k=1}^{m} \ln L^{(k)} = -9.80811$$

Adjusting Branch Lengths Step-By-Step

To compute optimal branch lengths do the following. Initialize the branch lengths.
Choose a branch (A.). Move the virtual root to an adjacent node (B.). Compute all partial likelihoods recursively (C.). Adjust the branch length to maximize the likelihood value (D.).
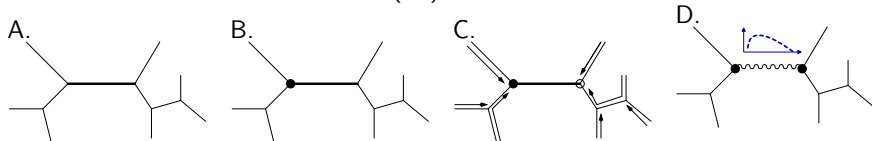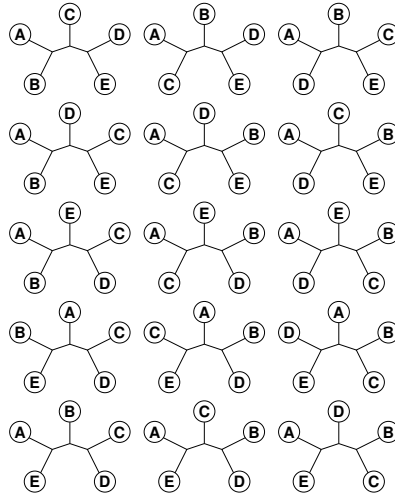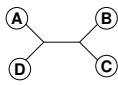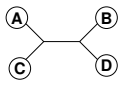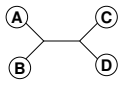

Repeat this for every branch until no better likelihood is gained.

Number of Trees to Examine...

$$B(n) = \frac{(2n-5)!}{2^{n-3}(n-3)!}$$
$$B(10) = 2027025$$
$$B(55) = 2.98 \cdot 10^{84}$$
$$B(100) = 1.70 \cdot 10^{182}$$

Finding the ML Tree

Exhaustive Search: guarantees to find the optimal tree, because all trees are evaluated, but not feasible for more than 10-12 taxa.
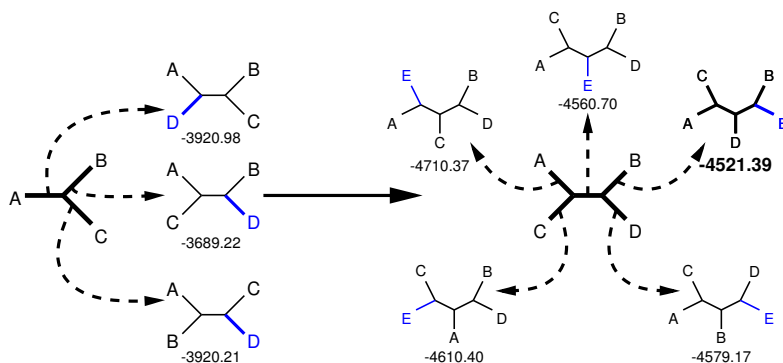
Branch and Bound: guarantees to find the optimal tree, without searching certain parts of the tree space – can run on more sequences, but often not for current-day datasets.

Heuristics: cannot guarantee to find the optimal tree, but are at least able to analyze large datasets.
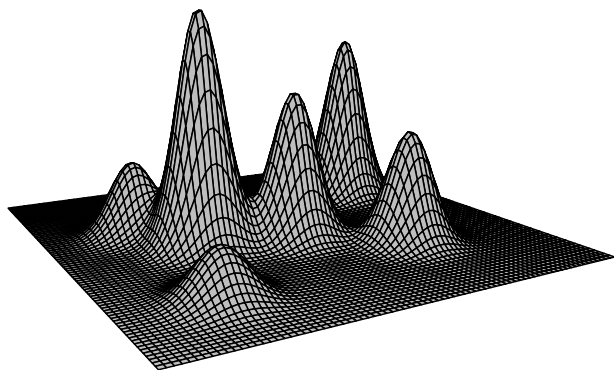

Build up a tree: Stepwise Insertion

Is also used for other (non-ML) methods like parsimony.

Local Maxima

What if we have multiple maxima in the likelihood surface?

Tree rearrangements to escape local maxima.
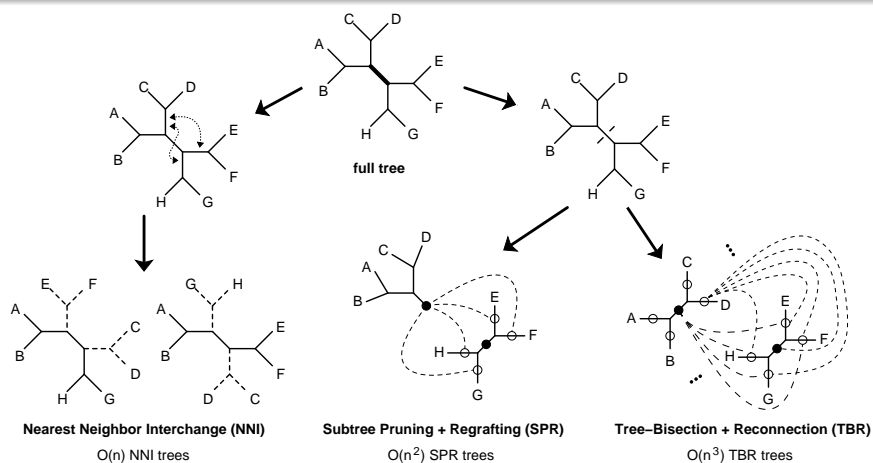
Tree Rearrangements: Scanning a Tree's Neighborhood

From a current tree construct other trees by rearranging its subtrees and evaluates all resulting trees. Repeat with the best tree found, until no better tree can be found. This also used for other (non-ML) methods, like parsimony.

How reliable is the reconstructed tree:

- Usually programs deliver a single (best) tree, but without confidence values for the subtrees.
- How can we assess reliability for the subtree?

Bootstrap and Consensus Tree

- Bootstrapping creates many pseudo-alignments by sampling alignment columns with replacement from the original alignment.
- From the pseudo-alignment we reconstruct trees.
- From the tree we collect and count all splits.
- From the splits we construct a consensus tree.

- **Definition:** A split $A|B$ in the tree is the bipartition of the leaves/taxa into two subsets A and B induced by removing an edge or branch from the tree.
- **Definition:** Two splits $A|B$ and $C|D$ are compatible, i.e. not contradictory, if at least one intersection of $A \cap C$, $A \cap D$, $B \cap C$, $B \cap D$ is empty.
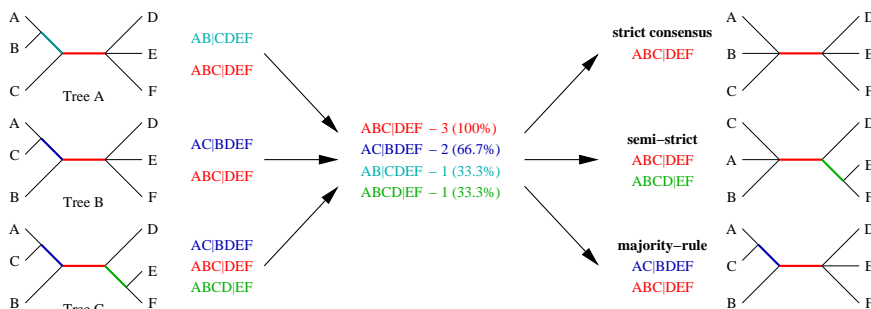
Notes

Estimating Confidence: The Bootstrap
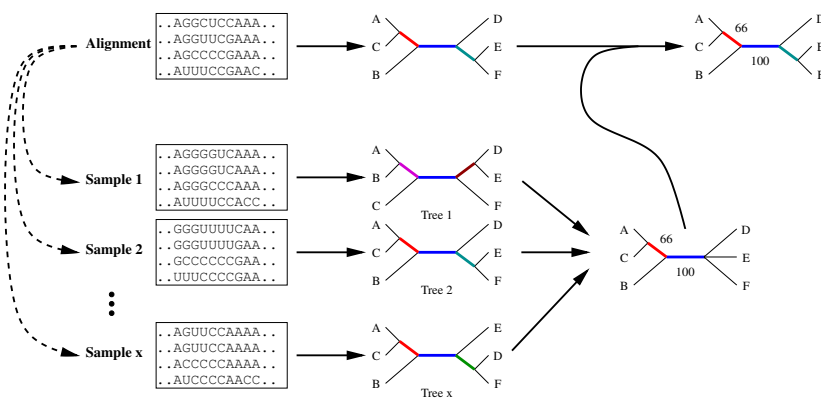
Notes

Summarizing Trees: Consensus Methods

- Strict consensus: contains all splits occuring in all input tree.
- Semi-strict consensus: contains all splits which are not contradicted by any tree.
- Majority consensus M_ℓ: contains all splits which occur in more than ℓ input trees, where $\ell \geq 50\%$ typically exactly 50%.

Notes