Phylogenetics and Evolution

An Overview
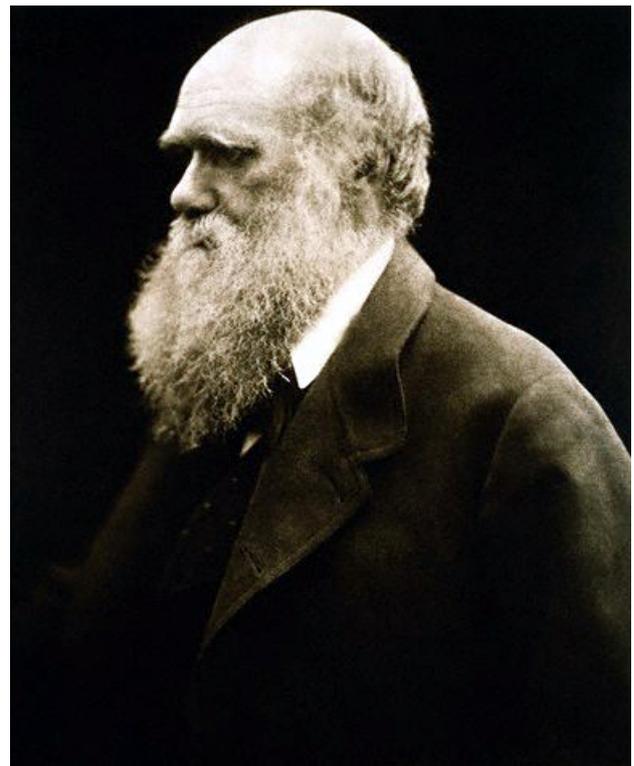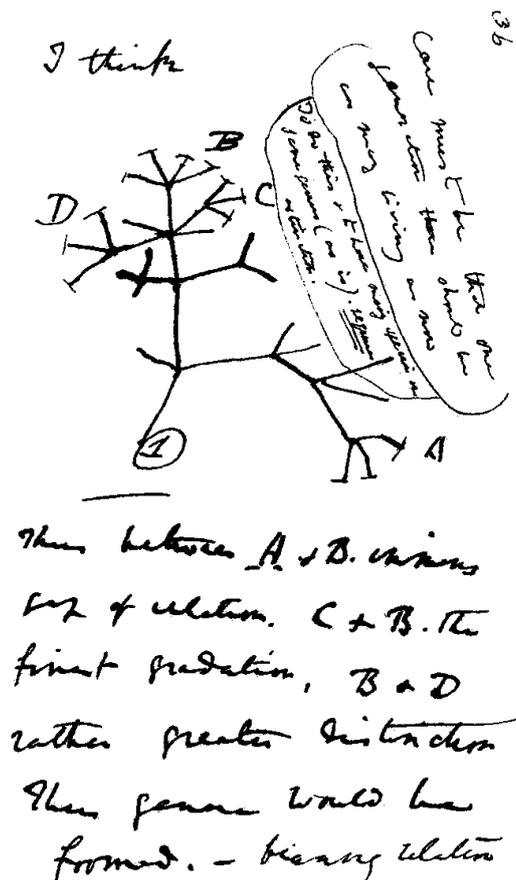
Heiko A. Schmidt

Center for Integrative Bioinformatics Vienna (CIBIV)
Max Perutz Labs
University of Vienna
Vienna, Austria
http://www.cibiv.at

Charles Darwin: Evolutionary Relationships

Charles Darwin (1809-1882)

Ernst Haeckel: Evolutionary Trees

Genealogical Tree of Humanity.

Ernst Haeckel (1834-1919)

A more recent view of the Tree of Life
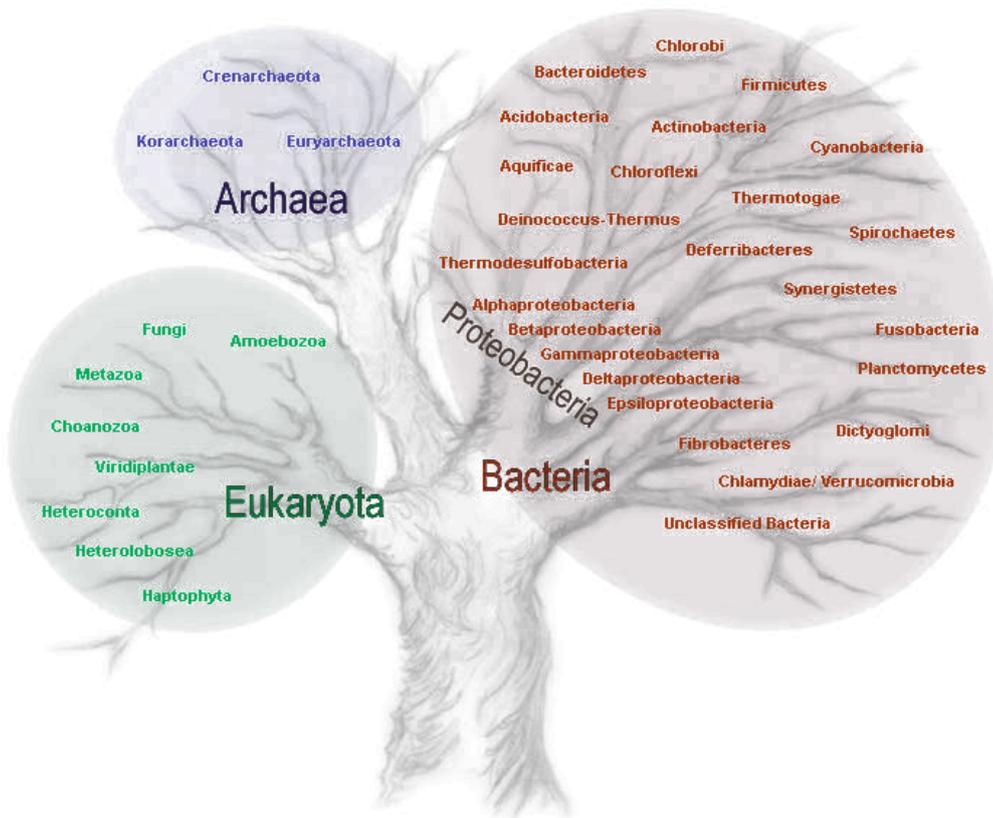
Source: genome.jgi-psf.org

Theodosius Dobzhansky: The Light of Evolution

Theodosius Dobzhansky (1900-1975)

Nothing in Biology Makes Sense Except in the Light of Evolution.

Dobzhansky, 1973

Traces of Sequence Evolution
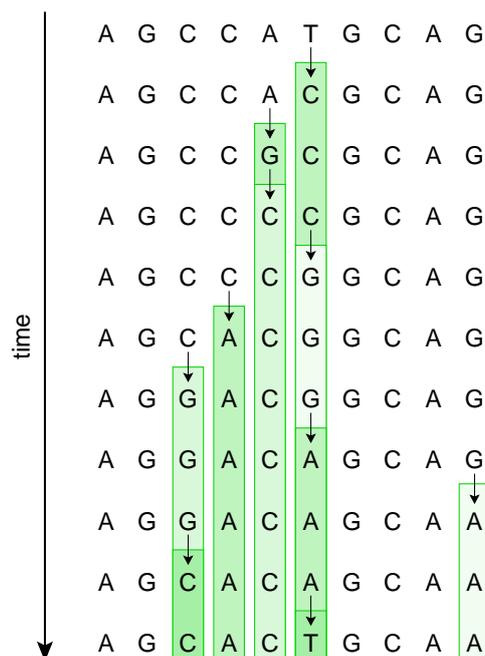
Some Notation
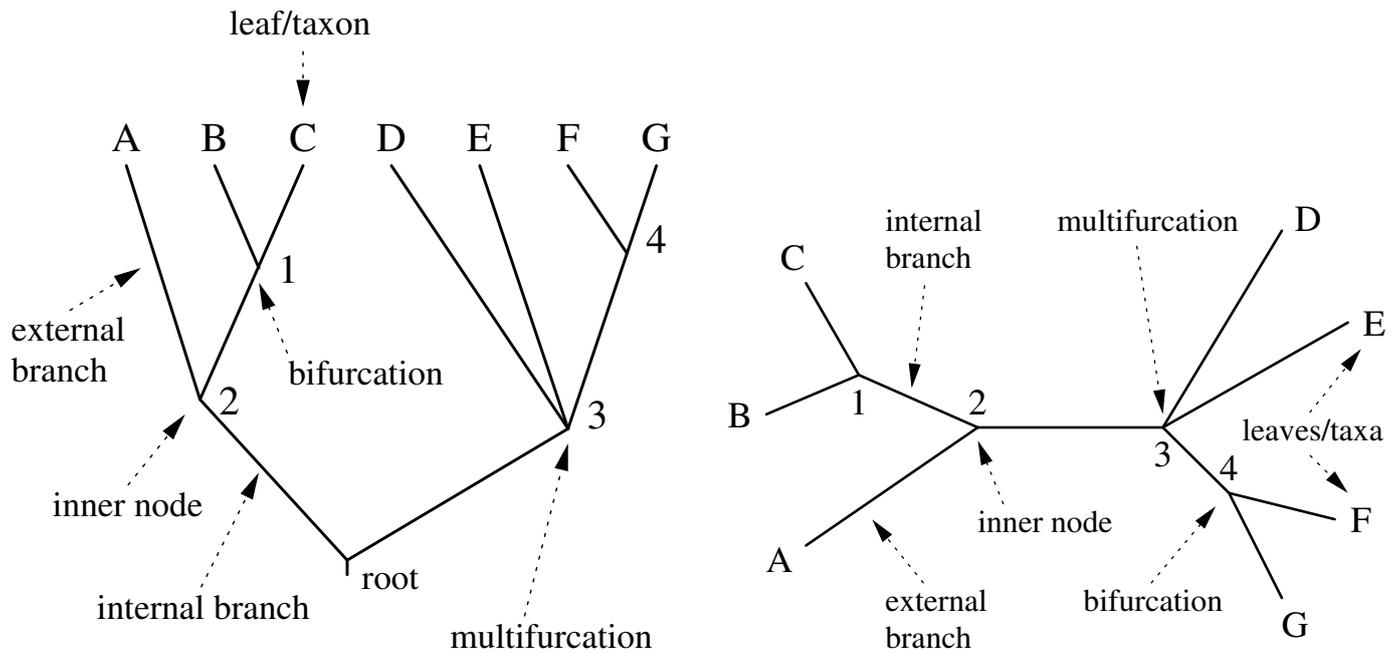
Note: branch = edge = split, external node = leaf = taxon = sequence are used interchangebly.

Main Types of Phylogenetic Methods

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

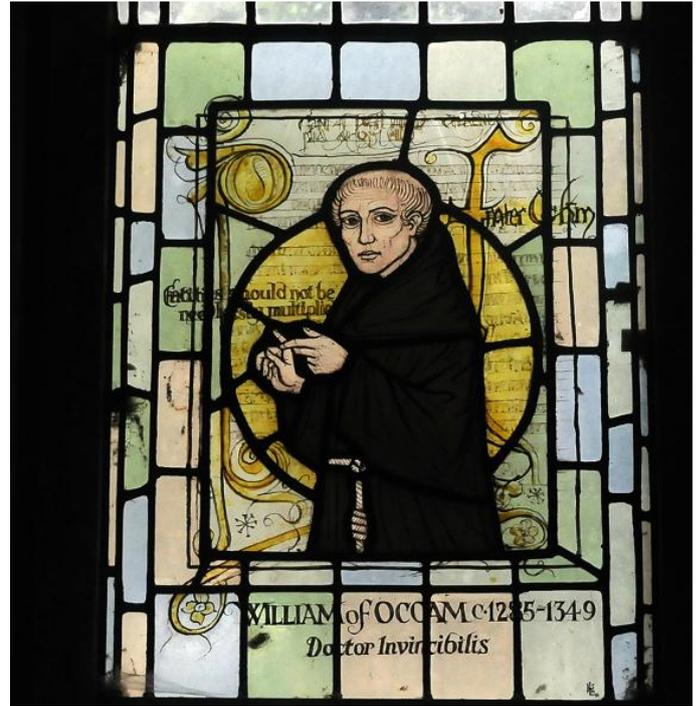# William of Ockham: The Law of Parsimony

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 (MP)

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

> **(Parsimony) Informative Sites: 2-2-2-rule**
>
> To be an informative site for the parsimony principle the column has to contain at least two different character states, and at least two of these states have to occur at least twice.

Maximum Parsimony: Fitch's (1970) algorithm

What is the minimum of mutations required?

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

{AC}*   {AG}*

1: ...C...
2: ...A...   {ACG}*
3: ...C...
4: ...A...
5: ...G...
                {AC}
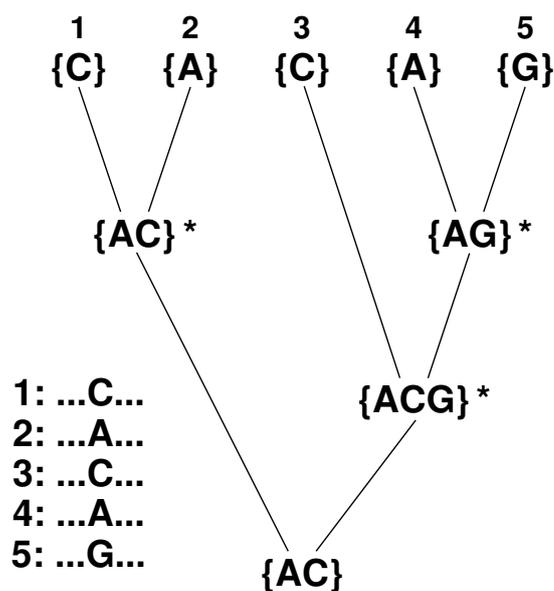
Note: We need 1 substitution per union in tree T (tree-length = substitutions needed).

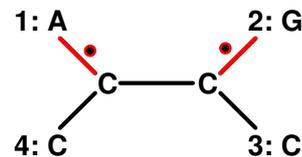1. Initialize state set S_k at each leaf k with the characters from the alignment.
2. Construct the state sets of all internal nodes in a post-order-traversal starting at the root.
3. Let k be the current node and i, j its decendents, then build the intersection of S_i and S_j:
   - If $S_i \cap S_j$ 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.
4. 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 its character and S_{root}.

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.

Simplicity with Caution – Einstein's Principle

Everything Should Be Made as Simple as Possible, But Not Simpler!
        (attributed to) Albert Einstein
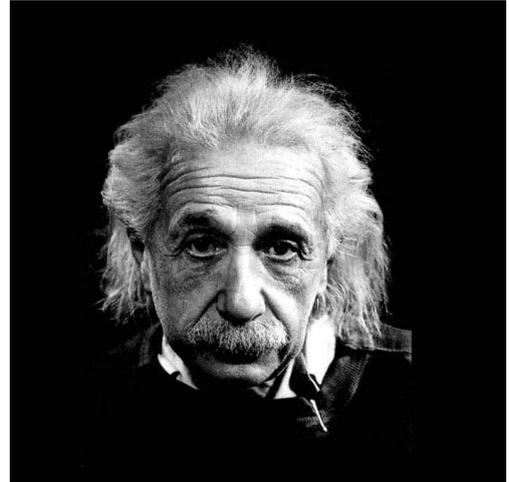
Albert Einstein (1879-1955)

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

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

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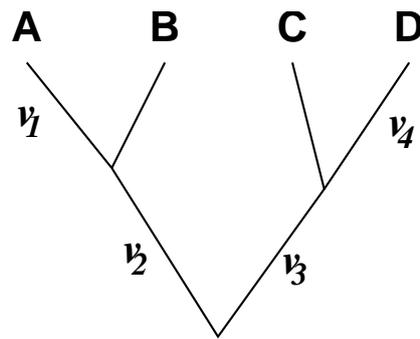
Aim: Find branch lengths v_b such that the sum of the branch lengths connecting any two leaves gets close to the measured distances between all pairs of leaves. That is, for instance
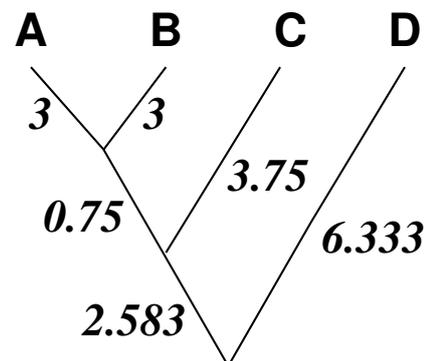
$$D_{A,D}^{measured} = v_1 + v_2 + v_3 + v_4$$

Distance Methods: UPGMA

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

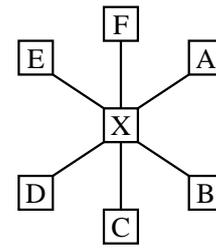|	A	B	C	D
A	0	6	7	13
B	6	0	8	14
C	7	8	0	11
D	13	14	11	0

$\Rightarrow$

Note:

- In the reconstructed rooted tree all sequences are equally far away from the root.
- If the distance matrix does not comply to this, UPGMA will likely reconstruct the wrong tree.

Distance Methods: Neighbor Joining (NJ)

A widely used distance method is Neighbor-Joining:

1 begin with a star tree with N leaves:

2 For each taxon i compute the net divergence r

$$r_i = \sum_{k=1}^{N} D_{i,k}$$

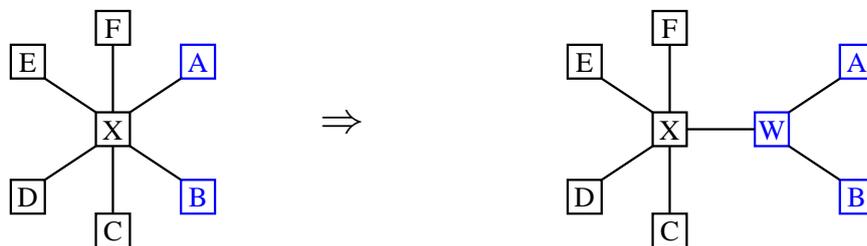For all pairs of taxa i, j, compute rate-corrected distances

$$M_{ij} = D_{i,j} - \frac{r_i + r_j}{N - 2}$$

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

Distance Methods: Neighbor Joining (NJ)

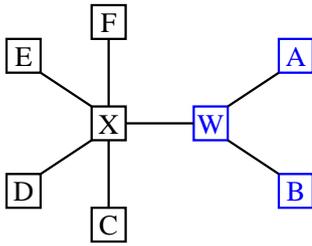4 cluster (A, B) and define an interior node W representing them:

$\Rightarrow$

5 compute the branch lengths for the external edges:

$$v_{A,W} = \frac{1}{2}\left(D_{A,B} + \frac{r_A - r_B}{N - 2}\right)$$
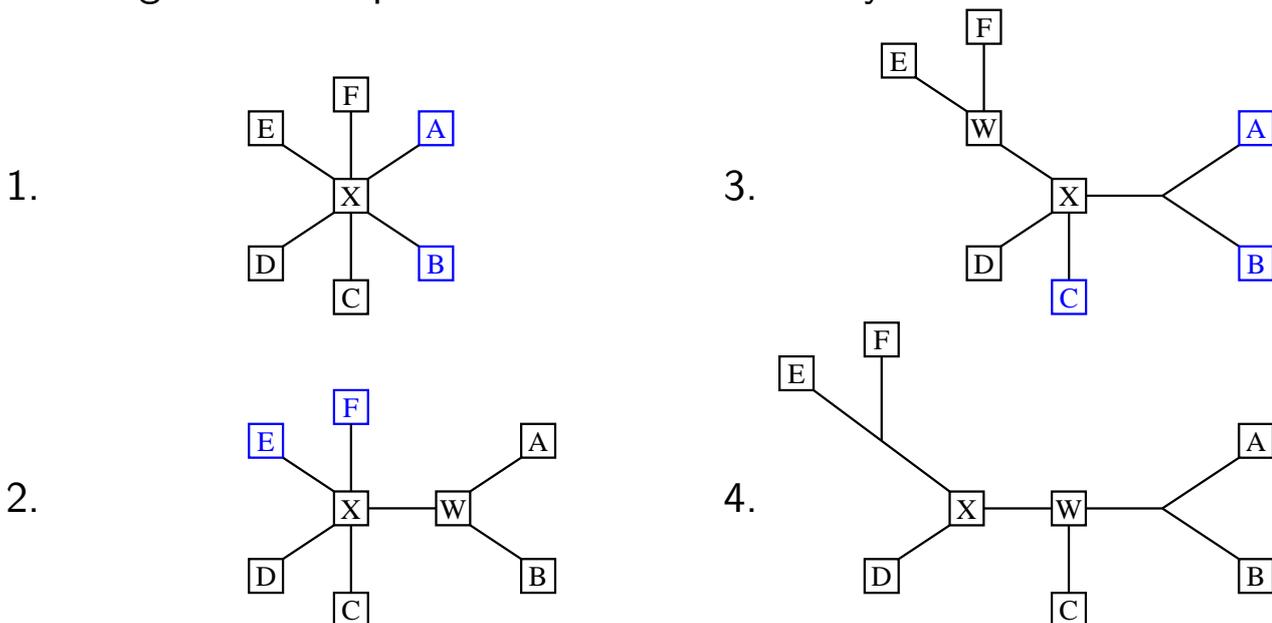
$$v_{B,W} = D_{A,B} - v_{A,W}.$$

6   compute the distances of W to the remaining $N - 2$ leaves k:

$$D_{W,k} = \frac{1}{2}\left(D_{A,k} + D_{B,k} - D_{A,B}\right))$$

7   continue at step 2 with the reduced set of leaves

Distance Methods: The NJ Tree Step-by-step

The algorithm is repeated until the tree is fully resolved:

1.

2.

3.

4.

The most simple tree: How to get distances?

The most simple tree could be seen as two
sequences and the distance between them.

Distances can be computed in various ways...

Jukes-Cantor Correction for Multiple Mutations

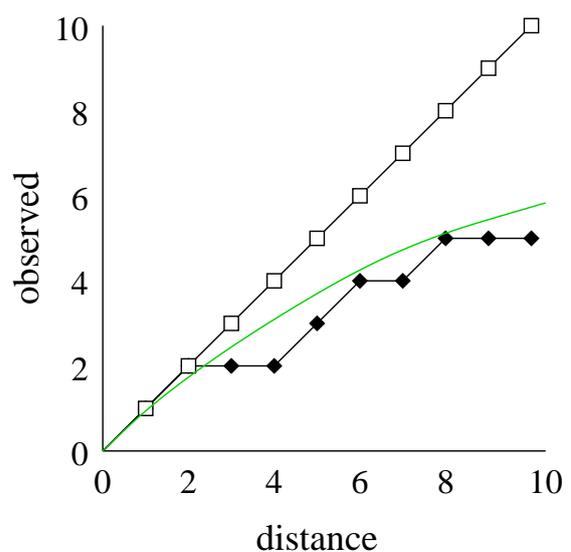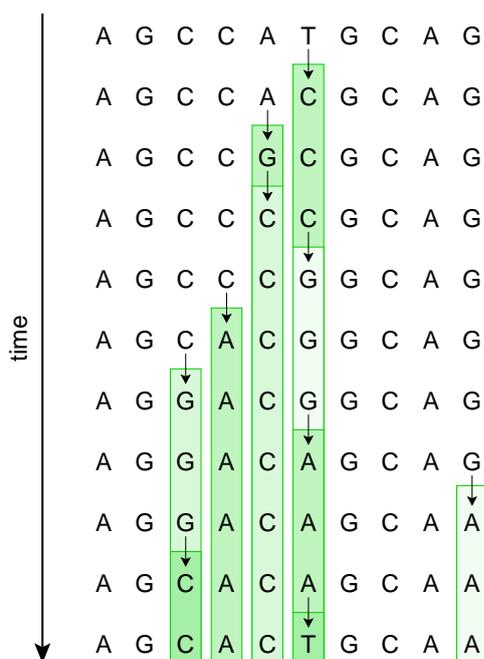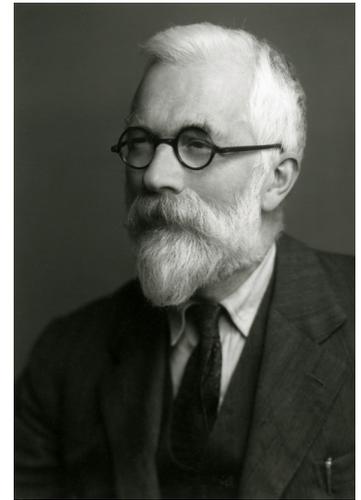The substitution process is commonly modeled as a Markov process.
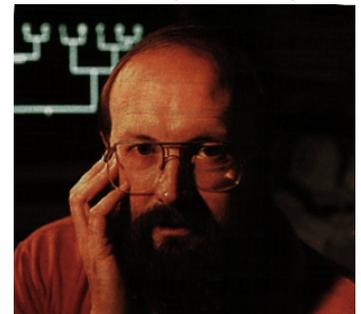
The most simple tree: How to get distances?

The most simple tree could be seen as two sequences and the distance between them.

Distances can be computed in various ways...

Usually via Maximum Likelihood (ML).

Ronald Fisher (1890-1962)

Joe Felsenstein (born 1942)

Introduction: ML on Coin Tossing

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

We take out one coin and toss 20 times:

$$H, T, T, H, H, T, T, T, T, H, T, T, H, T, H, T, T, H, T, T$$

$$
\begin{aligned}
\textbf{Probability} & \quad\quad \textbf{Likelihood} \\
p(k \text{ heads in } n \text{ tosses}|\theta) \;\equiv\; & L(\theta|k \text{ heads in } n \text{ tosses}) \\
=\; & \binom{n}{k}\theta^k(1-\theta)^{n-k} \\
& \text{(here binomial distribution)}
\end{aligned}
$$

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

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

coin 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

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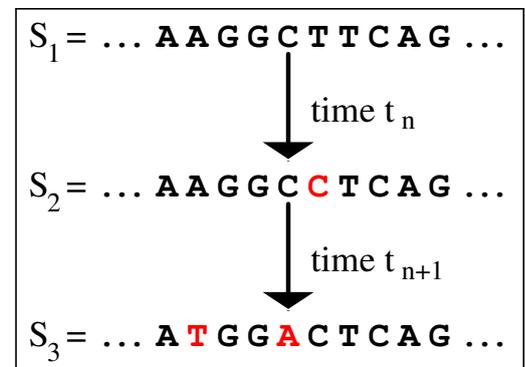
That means, a high dimensional optimization problem.
Hence, some parameters are often estimated/set separately.

Modeling Evolution

- Evolution is usually modeled as a

    stationary, time-reversible Markov process.

- What does that mean?

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.

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

time t_n

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

time t_{n+1}

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

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:

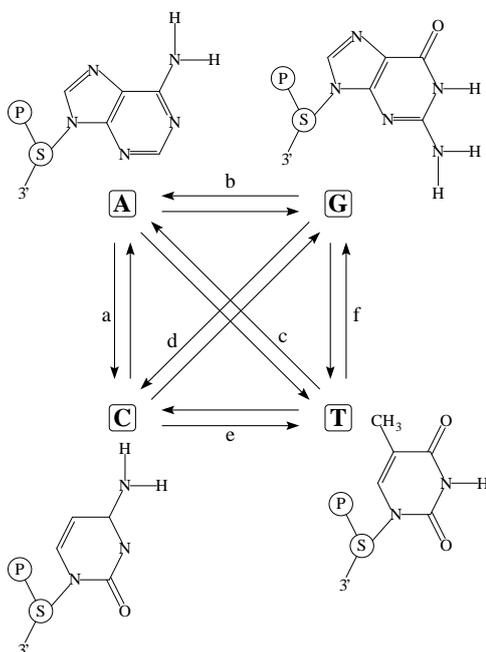Mutations in either direction are equally likely

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

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

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

Substitution Models

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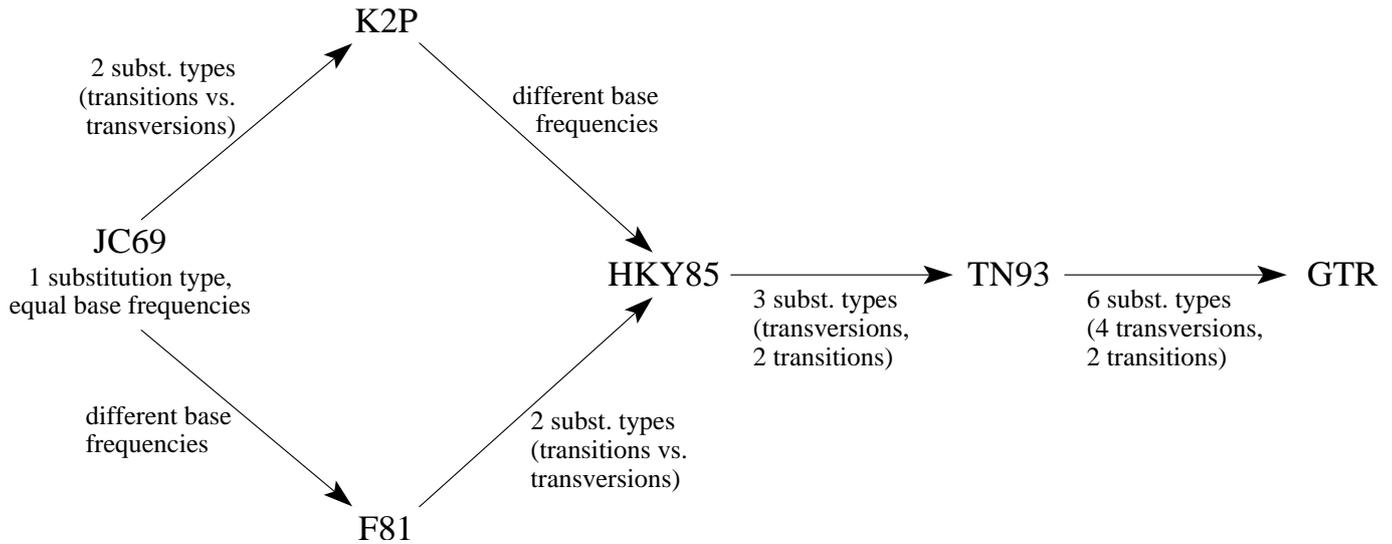

$$R = \begin{pmatrix} - & a & b & c \\ a & - & d & e \\ b & d & - & f \\ c & e & f & - \end{pmatrix} \begin{matrix} A \\ C \\ G \\ T \end{matrix}$$

with columns $A\ \ C\ \ G\ \ T$

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

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

Relations between DNA models

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, chloroplast 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

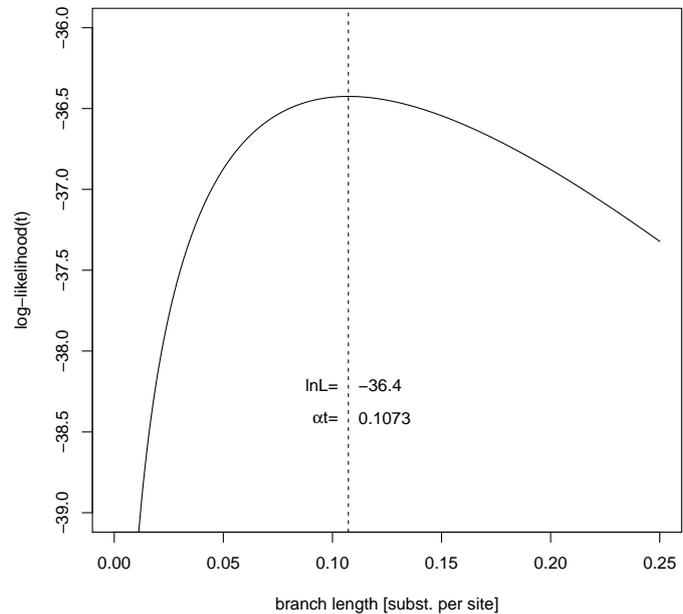# Computing ML Distances Using $P_{ij}(t)$

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

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

Likelihood surface for two sequences under JC69:

TGATCCTGAGTGAACTAAGC $= s'$
TGGTCCTGACTGAACTAAGC $= s$

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

log−likelihood(t)

branch length [subst. per site]

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:

$$L\left(\genfrac{}{}{0pt}{}{C \quad C}{G \quad C}\right) = L\left(\genfrac{}{}{0pt}{}{C \;_{A \; A}\; C}{G \quad C}\right) + L\left(\genfrac{}{}{0pt}{}{C \;_{A \; C}\; C}{G \quad C}\right) + \cdots + L\left(\genfrac{}{}{0pt}{}{C \;_{G \; C}\; C}{G \quad C}\right) + \cdots + L\left(\genfrac{}{}{0pt}{}{C \;_{T \; T}\; C}{G \quad C}\right)$$

for every column in the alignment... but there is a faster algorithm.
(Peeling Algorithm by Felsenstein, 1981)

Likelihoods of Trees (Single alignment column, given tree)

For a single alignment column and a given tree:
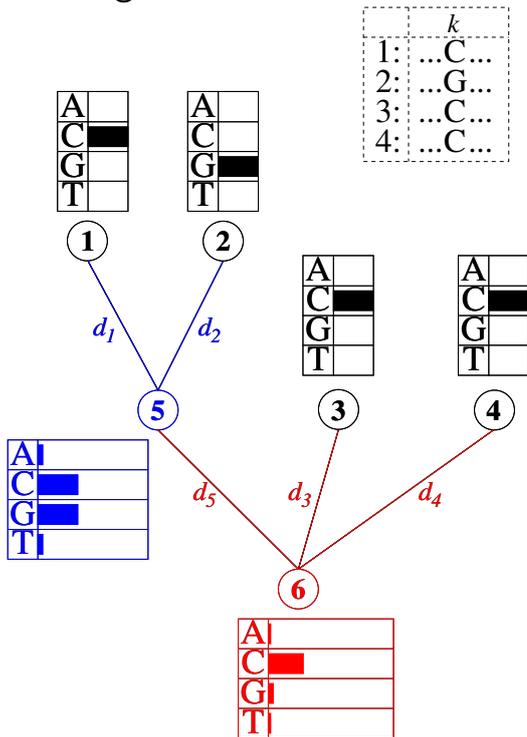
Likelihoods of nucleotides i at inner nodes:

$$L_5(i) = [P_{iC}(d_1) \cdot L_1(C)] \cdot [P_{iG}(d_2) \cdot L_2(G)]$$
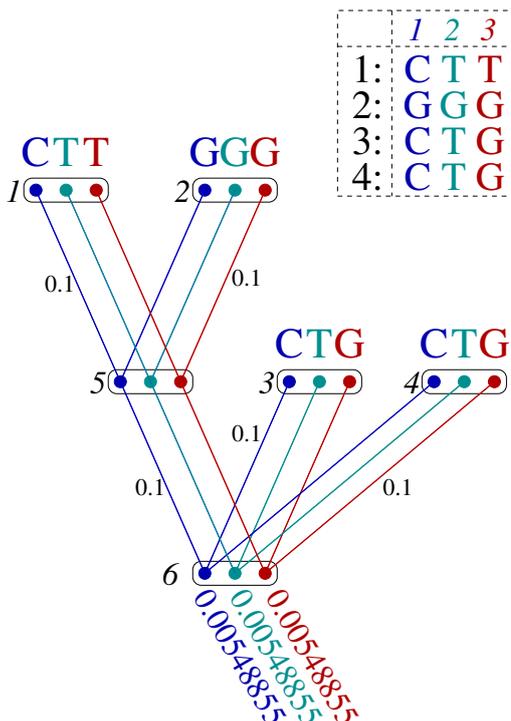
$$L_6(i) = \prod_{v=\{3,4,5\}} \left[\sum_{j=\{ACGT\}} P_{ij}(d_v) \cdot L_v(j) \right]$$

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 $P_{ij}(0.1) = \begin{cases} .9068 & i = j \\ .0313 & i \neq j \end{cases}$ (JC)

Likelihoods of Trees (multiple columns)

Considering this tree with $n = 4$ sequences of length $m = 3$ the tree likelihood of this tree is

$$
\begin{aligned}
\mathcal{L}(T) &= \prod_{k=1}^{m} L^{(k)} \\
&= 0.0054886 \cdot 0.0054886 \cdot 0.0054886 \\
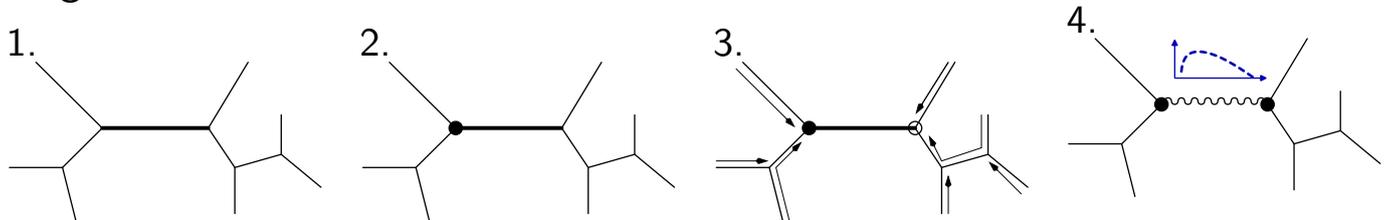&= 0.0000001653381
\end{aligned}
$$

or the log-likelihood

$$\ln \mathcal{L}(T) = \sum_{k=1}^{m} \ln L^{(k)} = -15.61527$$

Adjusting Branch Lengths Step-By-Step

To compute optimal branch lengths do the following. Initialize the branch lengths.
(1.) Choose a branch. (2.) Move the virtual root to an adjacent node.
(3.) Compute all partial likelihoods recursively. (4.) Adjust the branch length to maximize the likelihood value.

1.   2.   3.   4.

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

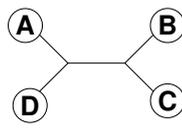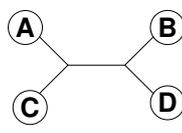This is based on the Pulley-Principle (Felsenstein, 1981) which states that the root can be moved on the tree but the likelihood doesn't change.

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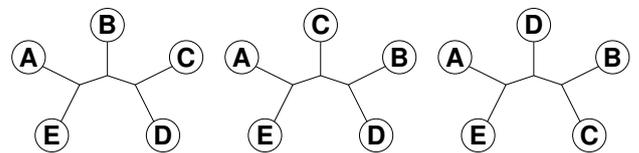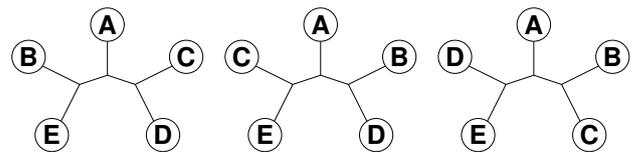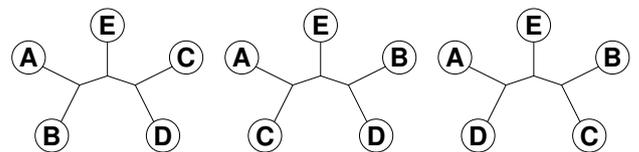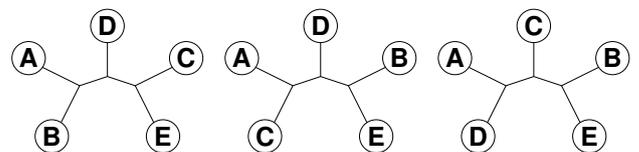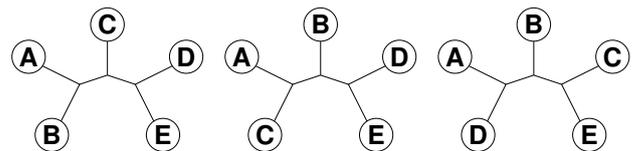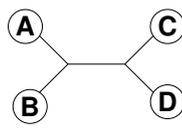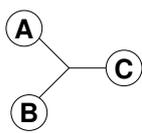
$$B(n) = \frac{(2n-5)!}{2^{n-3}(n-3)!}$$
$$B(10) = 2027025$$
$$B(55) = 2.98 \cdot 10^{84}$$
$$B(100) = 1.70 \cdot 10^{182}$$

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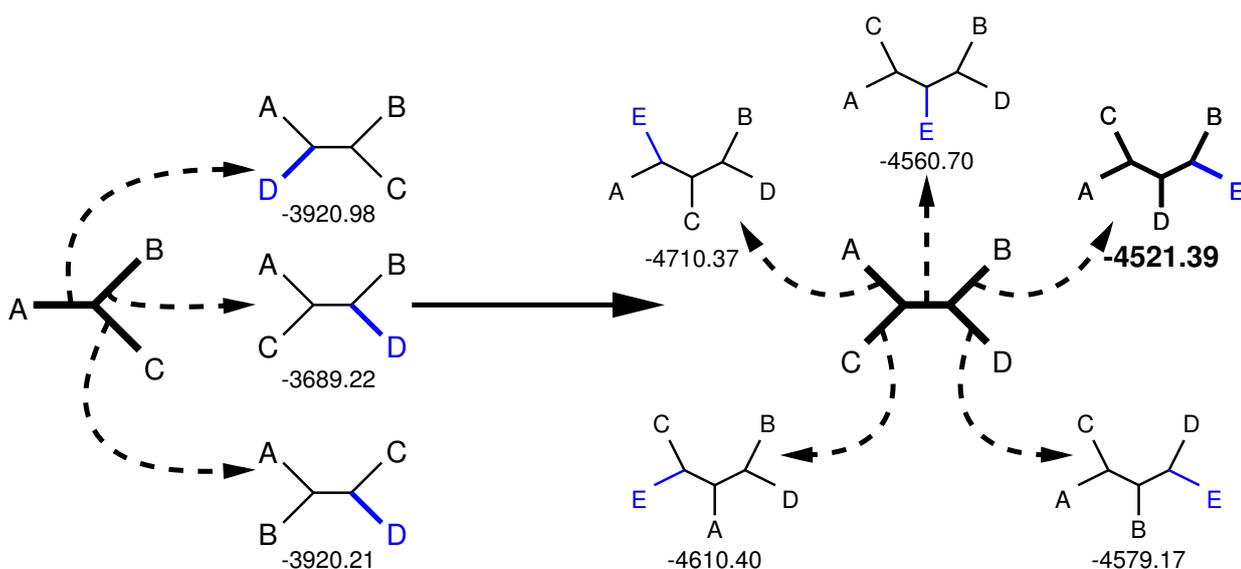
Is also used for other (non-ML) methods like parsimony.

Local Maxima

What if we have multiple maxima in the likelihood surface?
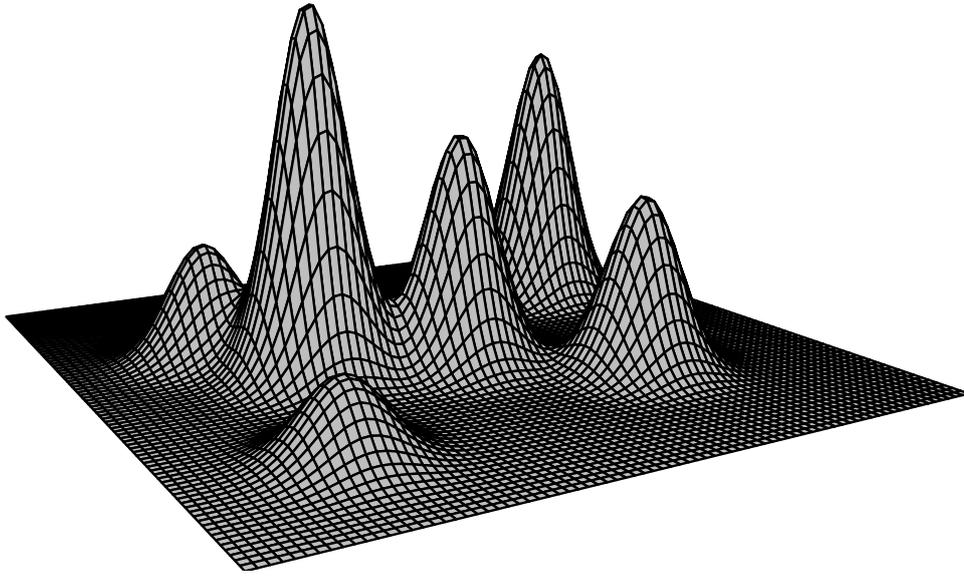
Tree rearrangements to escape local maxima.

Tree Rearrangements: Scanning a Tree's Neighborhood

Nearest Neighbor Interchange (NNI)

O(n) NNI trees

Subtree Pruning + Regrafting (SPR)

O(n²) SPR trees

Tree–Bisection + Reconnection (TBR)

O(n³) TBR trees
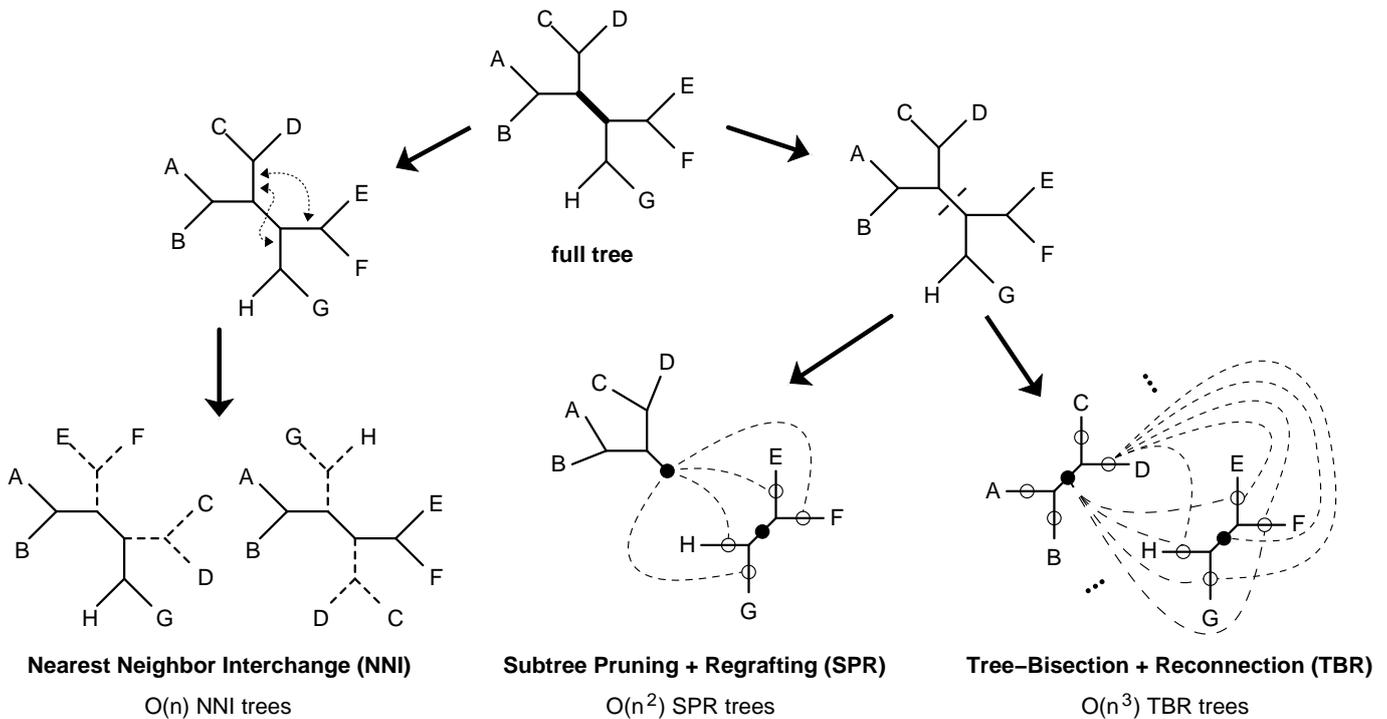
From a current tree construct other trees by rearranging its subtrees and evaluate all resulting trees. Repeat with the best tree found, until no better tree can be found. This also used for other (non-ML) methods, like parsimony.

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

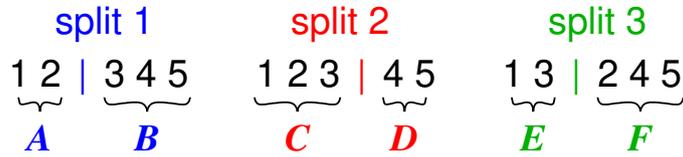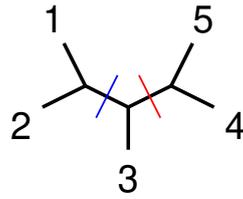- **Bootstrapping** creates many pseudo-alignments by sampling alignment columns with replacement from the original alignment.
- From the pseudo-alignment we reconstruct trees.
- From the trees we collect and count all splits.
- From the splits we construct a consensus tree.

- **Definition:** A split $\mathcal{A}|\mathcal{B}$ in the tree is the bipartition of the leaves/taxa into two subsets $\mathcal{A}$ and $\mathcal{B}$ induced by removing an edge or branch from the tree.
- **Definition:** A trivial split is a split induced by an external branch, because they have to be present in every tree. Otherwise a leaf would not be connected to the tree.
- **Definition:** Two splits $\mathcal{A}|\mathcal{B}$ and $\mathcal{C}|\mathcal{D}$ are compatible (i.e. not contradictory) and, thus, fit into one tree, if at least one intersection of $\mathcal{A} \cap \mathcal{C}$, $\mathcal{A} \cap \mathcal{D}$, $\mathcal{B} \cap \mathcal{C}$, $\mathcal{B} \cap \mathcal{D}$ is empty.

Example: Chacking Split Compatibility

split 1   split 2   split 3

$1\,2 \mid 3\,4\,5$ $1\,2\,3 \mid 4\,5$ $1\,3 \mid 2\,4\,5$

A B C D E F

split 1 vs. split 2	
$A \cap C$	$\{1, 2\}$
$A \cap D$	$\{\}$
$B \cap C$	$\{3\}$
$B \cap D$	$\{4, 5\}$
compatible	

$\leftarrow$

split 1 vs. split 3	
$A \cap E$	$\{1\}$
$A \cap F$	$\{2\}$
$B \cap E$	$\{3\}$
$B \cap F$	$\{4, 5\}$
incompatible	

split 2 vs. split 3	
$C \cap E$	$\{1, 3\}$
$C \cap F$	$\{2\}$
$D \cap E$	$\{\}$
$D \cap F$	$\{4, 5\}$
compatible	

$\leftarrow$
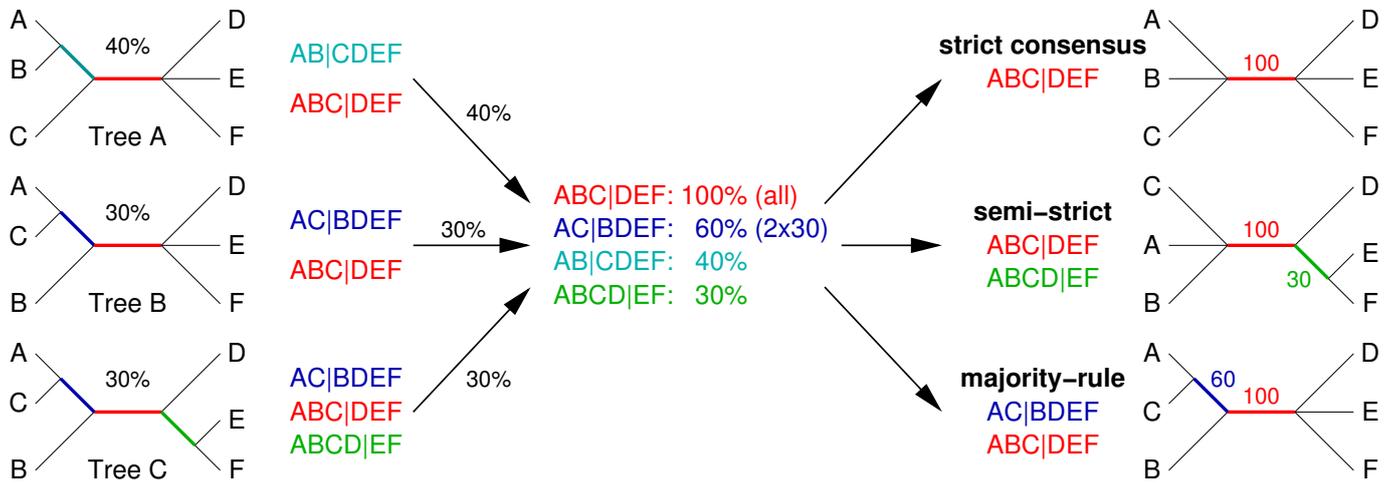
Estimating Confidence: The Bootstrap
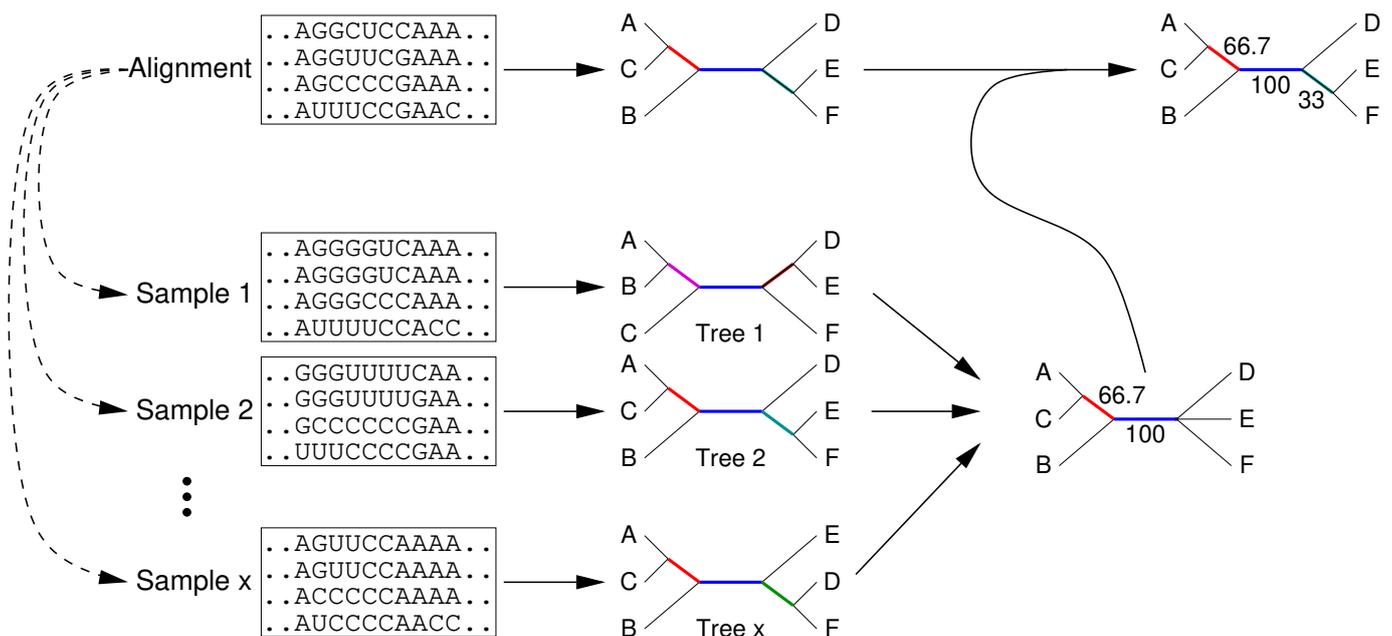
Summarizing Trees: Consensus Methods

- Strict consensus: contains all splits occuring in all input trees.
- 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%.
- Majority Rule extended (MRe): starting from the most to the least frequent splits one collects compatible splits. If a split is incompatible to the already collected ones, it is discarded and the next is examined.