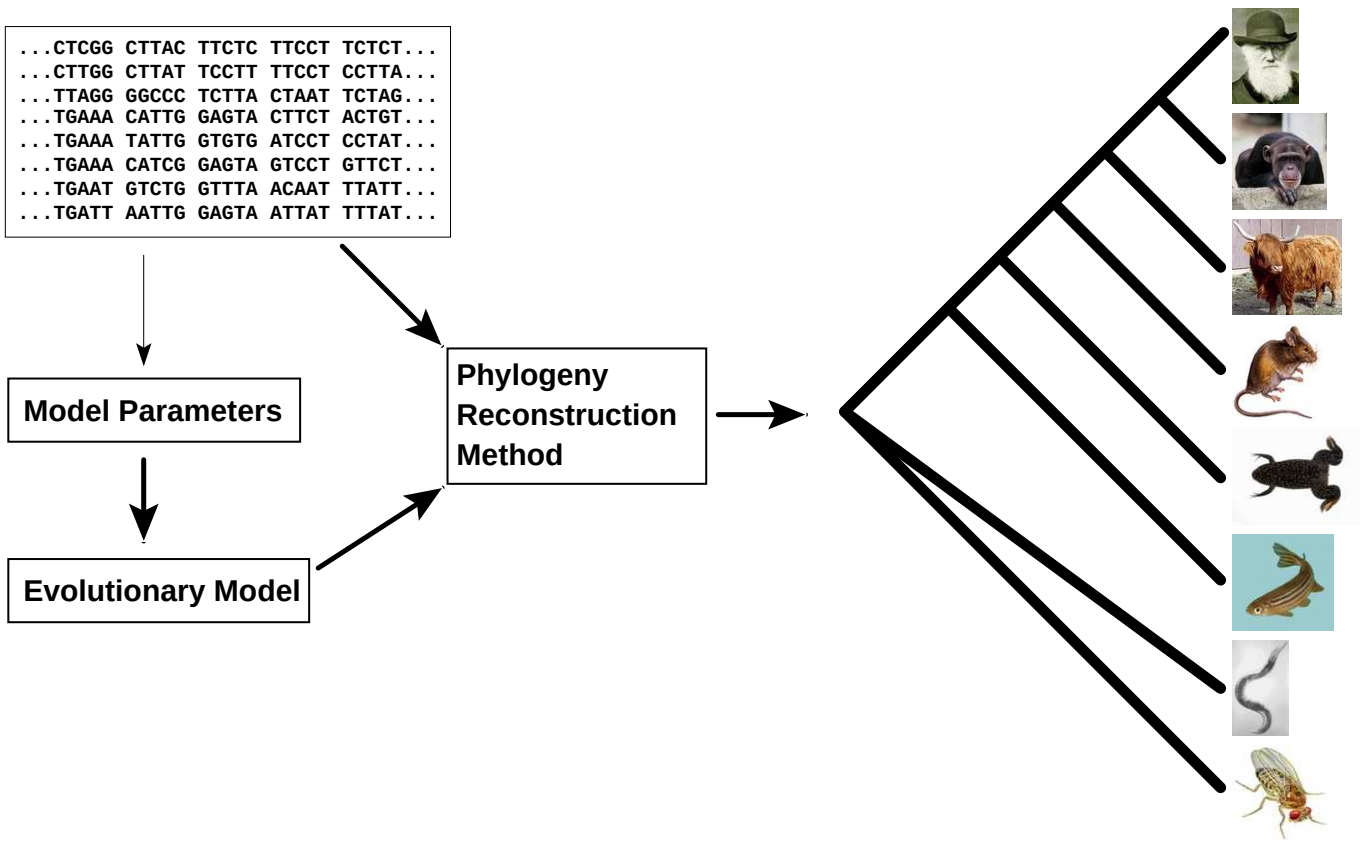
Applied Bioinformatics
Phylogeny Reconstruction

Heiko A. Schmidt

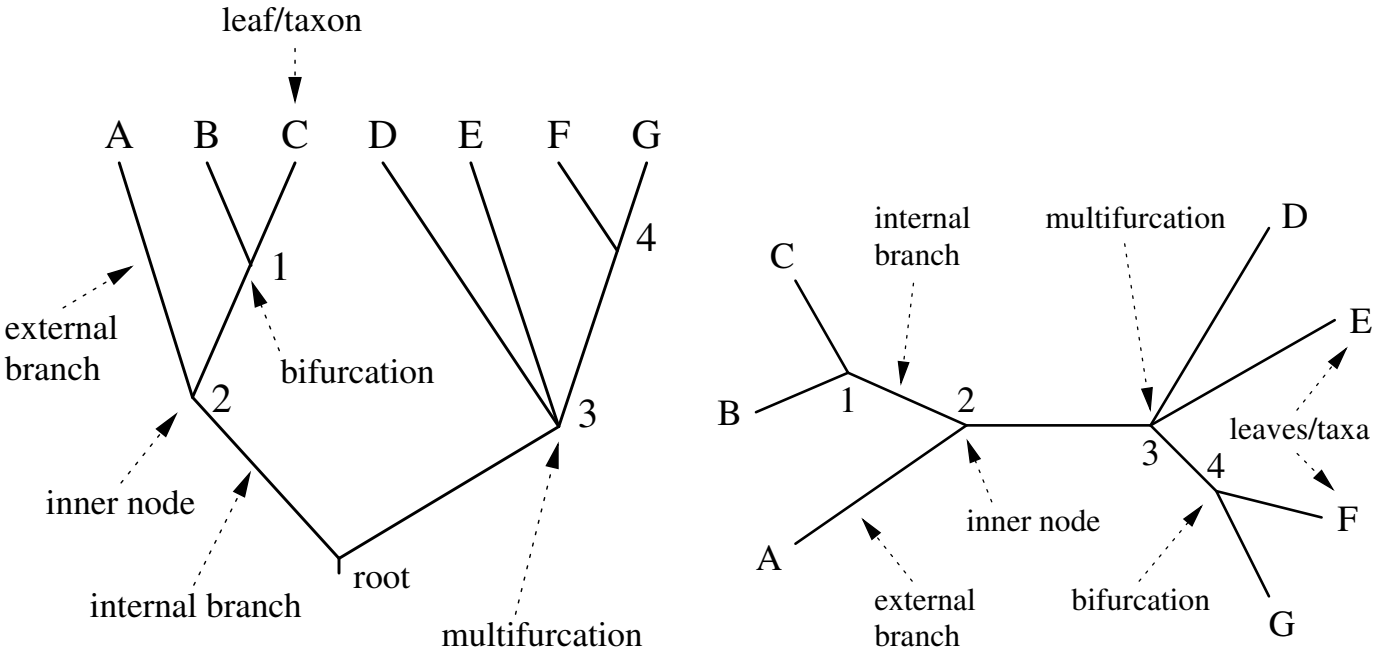
Center for Integrative Bioinformatics Vienna (CIBIV)
Max F. Perutz Laboratories (MFPL)
Vienna, Austria
heiko.schmidt@univie.ac.at

February 2008

Recap: Phylogenetic Reconstruction



Some Notation



Main Types of Phylogenetic Methods

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

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:

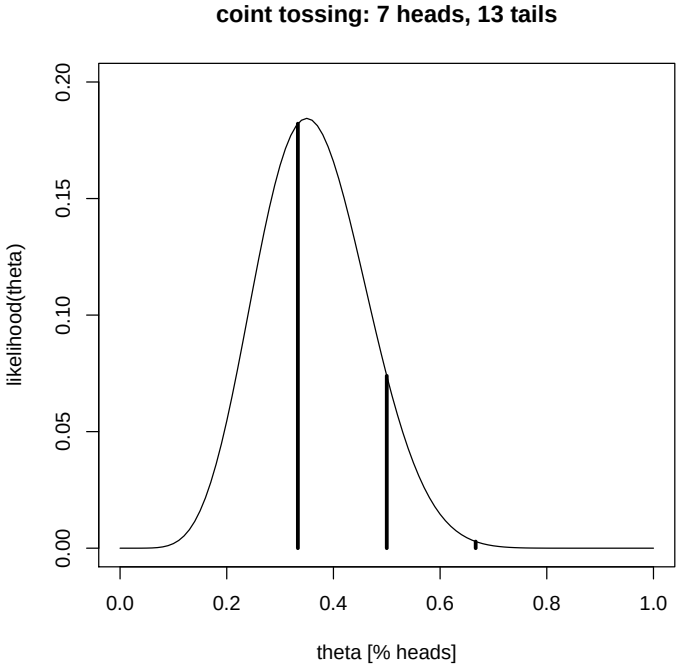
$$H, T, T, H, H, T, T, T, T, H, T, T, H, T, H, T, T, H, T, T$$

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

Introduction: ML on Coin Tossing (Estimate)



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

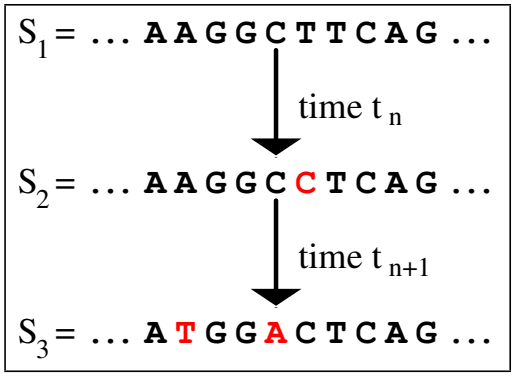
  stationary, time-reversible Markov process.

- 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.



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

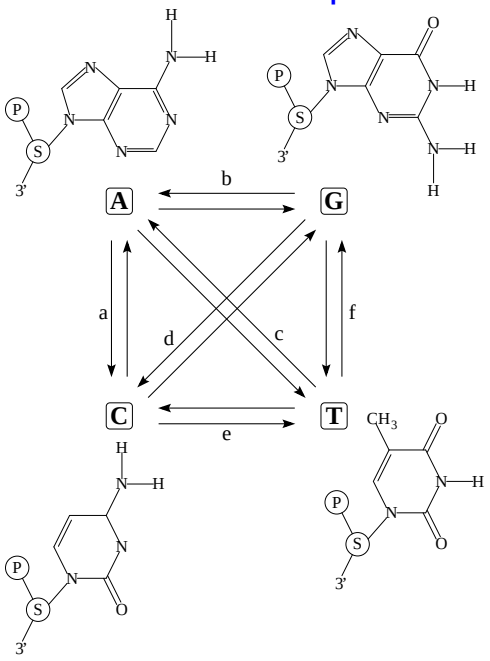
$$\pi_i \cdot P_{ij}(t) = P_{ji}(t) \cdot \pi_j$$

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:



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

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

From Substitution rates to probabilities

... R and Π are combined into the instantaneous rate matrix Q

$$Q = \begin{pmatrix} \bullet_A & a\pi_C & b\pi_G & c\pi_T \\ a\pi_A & \bullet_C & d\pi_G & e\pi_T \\ b\pi_A & d\pi_C & \bullet_G & f\pi_T \\ c\pi_A & e\pi_C & f\pi_G & \bullet_T \end{pmatrix} \quad \begin{aligned} \bullet_A &= -(a\pi_C + b\pi_G + c\pi_T) \\ \bullet_C &= -(a\pi_A + d\pi_G + e\pi_T) \\ \bullet_G &= -(b\pi_A + d\pi_C + f\pi_T) \\ \bullet_T &= -(c\pi_A + e\pi_C + f\pi_G) \end{aligned}$$

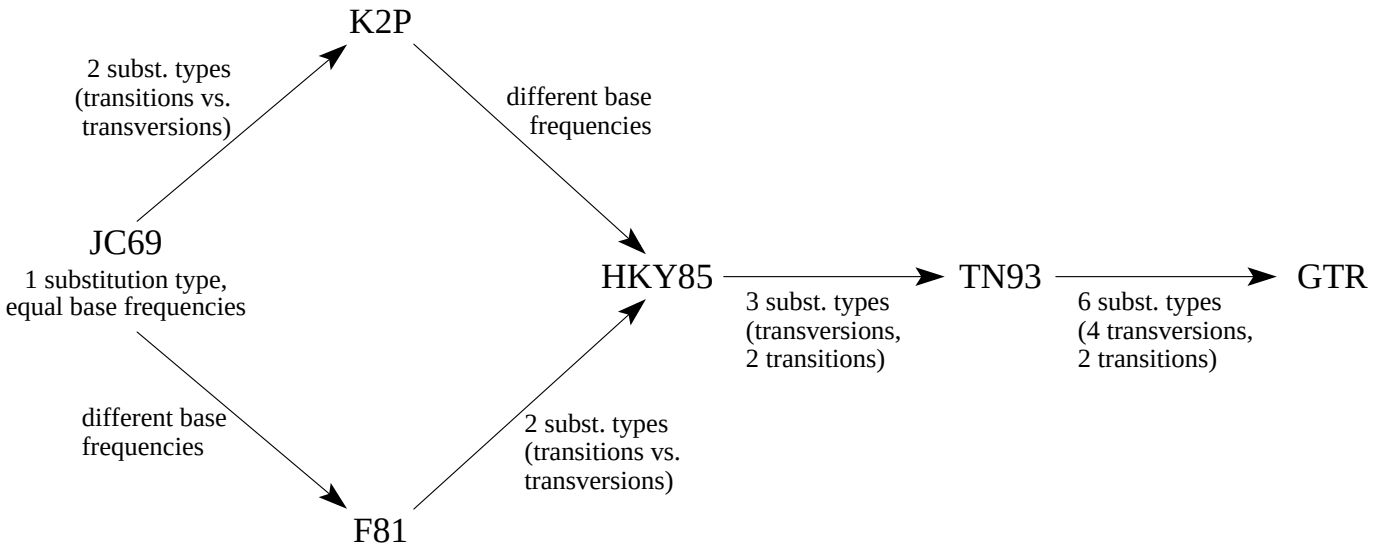
(where the row sums are zero).

Given now the instantaneous rate matrix Q, we can compute a substitution probability matrix P

$$P(t) = e^{Qt}$$

With this matrix P we can compute the probability $P_{ij}(t)$ of a change $i \longrightarrow j$ over a 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)
- Dayhoff (Dayhoff *et al.*, 1978)
- JTT (Jones *et al.*, 1992)
- mtREV (Adachi & Hasegawa, 1996)
- cpREV (Adachi *et al.*, 2000)
- VT (Müller & Vingron, 2000)
- WAG (Whelan & Goldman, 2000)
- ...
- BLOSUM 62 (Henikoff & Henikoff, 1992)

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

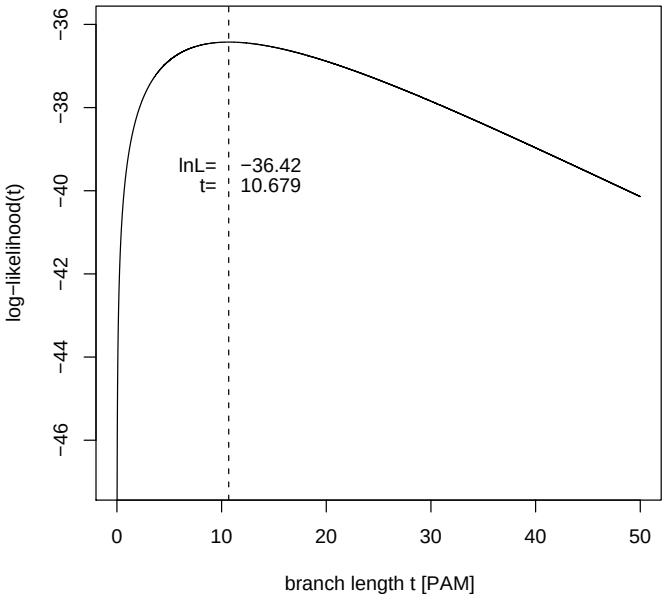
The Likelihood of sequence s evolving to s' in time t:

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

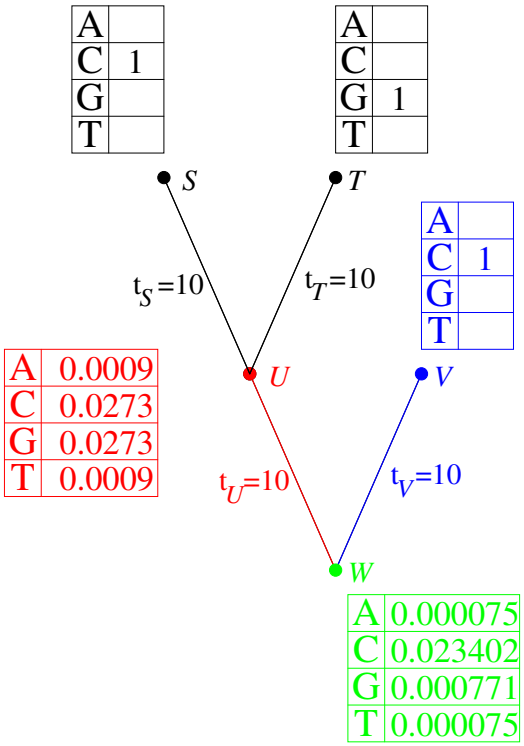
Likelihood surface for two sequences under JC69:

GATCCTGAGAGAAATAAAC
GGTCCTGACAGAAATAAAC

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



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



Likelihoods of nucleotides at inner nodes:

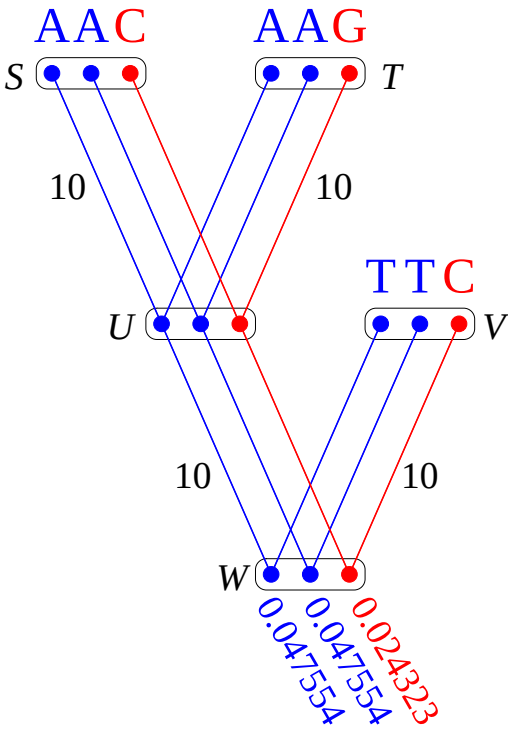
$$L_U(i) = [P_{iC}(10) \cdot L(C)] \cdot [P_{iG}(10) \cdot L(G)]$$

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

Site-Likelihood of an alignment column k:

$$L^{(k)} = \sum_{\substack{i= \\ ACGT}} \pi_i \cdot L_W(i) = 0.024323$$

Likelihoods of Trees (multiple columns)



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

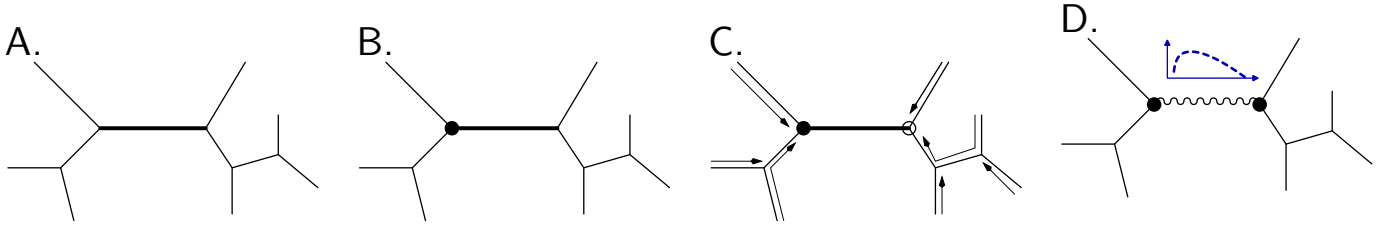
$$\mathcal{L}(T) = \prod_{k=1}^{m} L^{(k)} = 0.047554^2 \cdot 0.024323 = 0.000055$$

or the log-likelihood

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

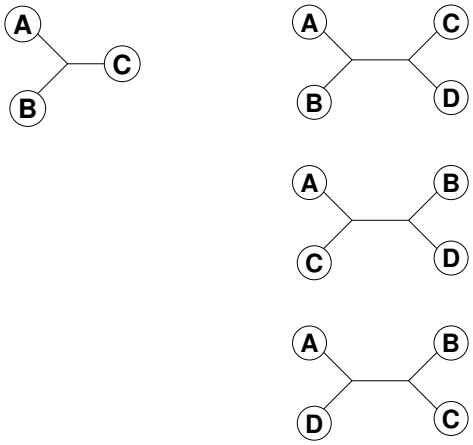
Adjusting Branch Lengths Step-By-Step

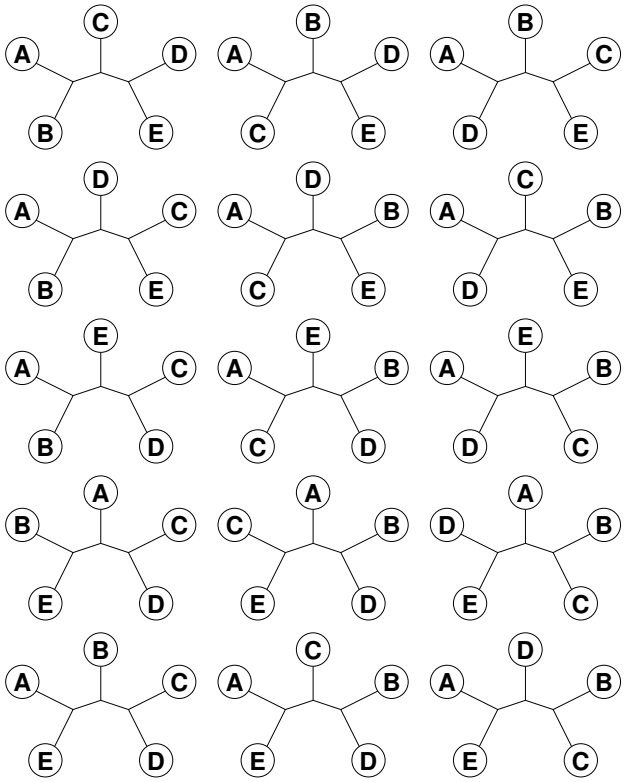
To compute optimal branch lengths do the following. Initialize the branch lengths.
Choose a branch (A.). Move the virtual root to an adjacent node (B.). Compute all partial likelihoods recursively (C.). Adjust the branch length to maximize the likelihood value (D.).



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

Number of Trees to Examine...





$$B(n) = \frac{(2n-5)!}{2^{n-3}(n-3)!}$$

$$B(10) = 2027025$$

$$B(55) = 2.98 \cdot 10^{84}$$

$$B(100) = 1.70 \cdot 10^{182}$$

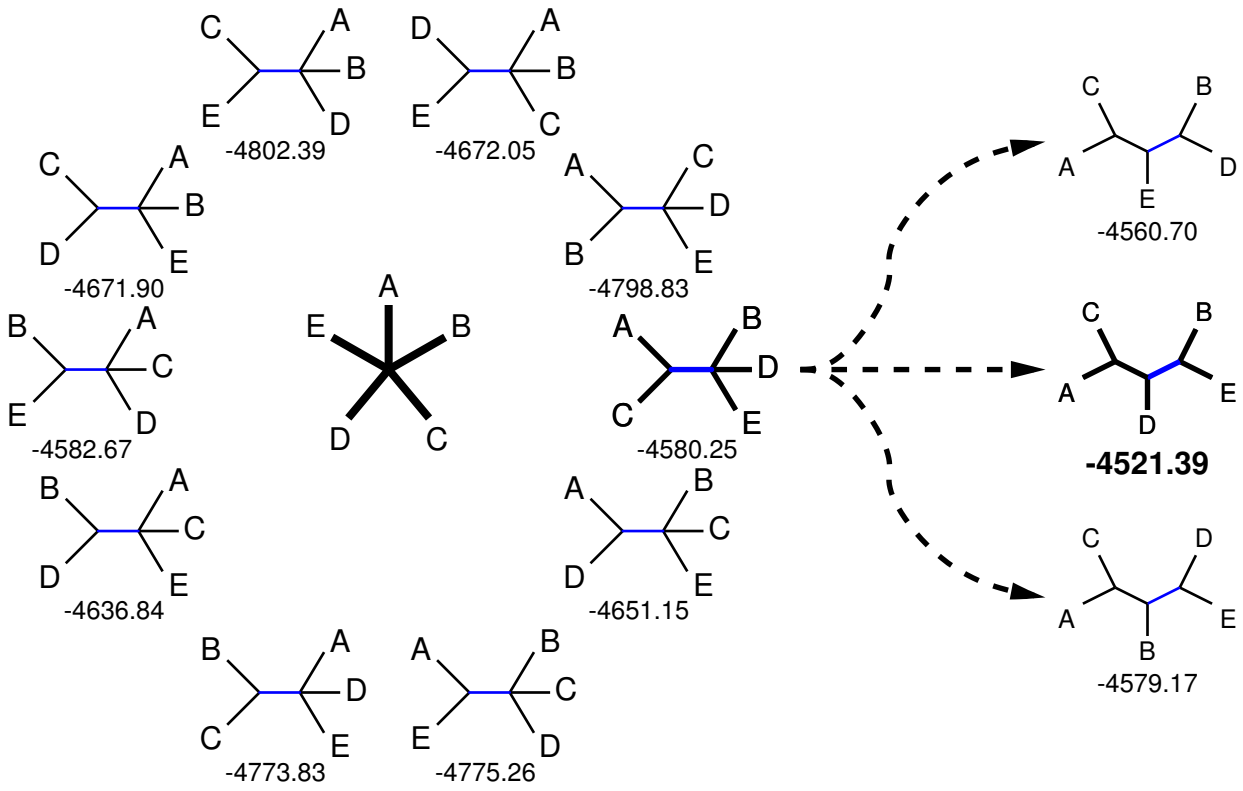
Finding the ML Tree

Exhaustive Search: guarantees to find the optimal tree, because all trees are evaluated, but not feasible for more than 10-12 taxa.

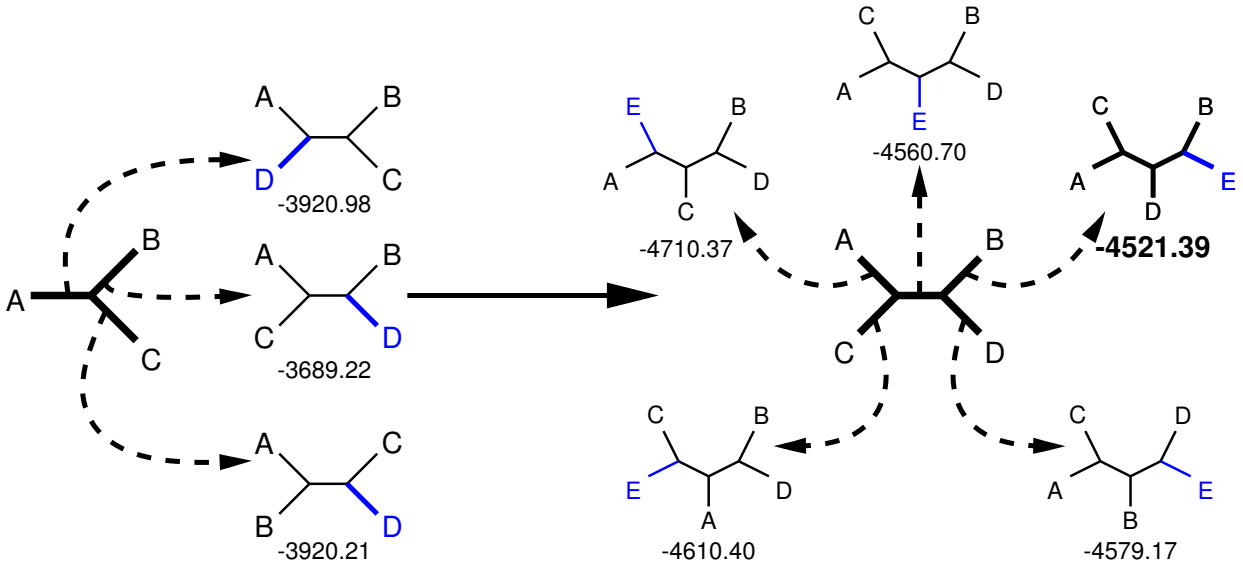
Branch and Bound: guarantees to find the optimal tree, without searching certain parts of the tree space – can run on more sequences, but often not for current-day datasets.

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

Build up a tree: Star Decomposition

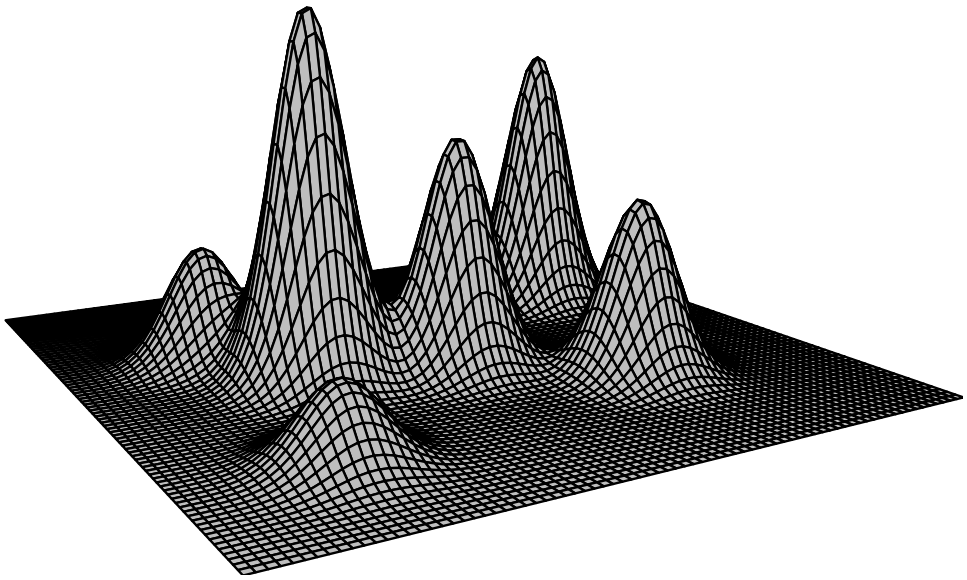


Build up a tree: Stepwise Insertion



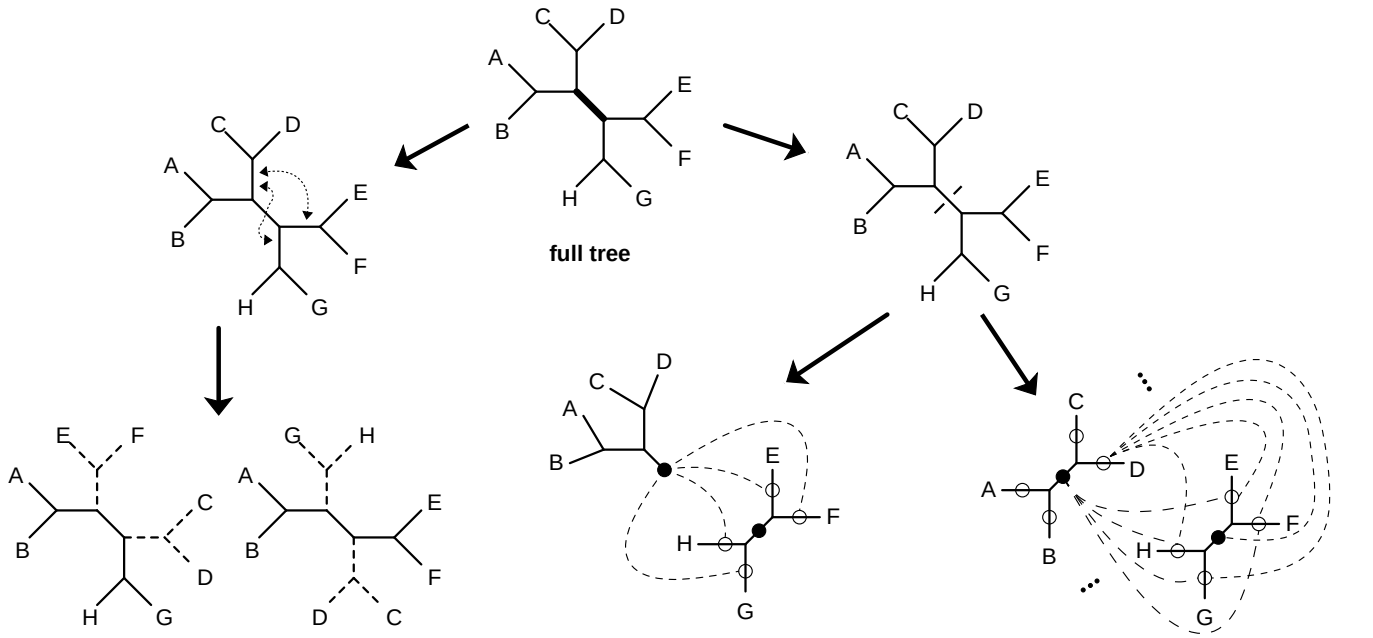
Local Maxima

What if we have multiple maxima in the likelihood surface?



Tree rearrangements to escape local maxima.

Tree Rearrangements: Scanning a Tree's Neighborhood



ML programs: DNAML (PHYLIP), fastDNAml

Concept: Stepwise insertion + NNI/SPR

1. Build tree with stepwise insertion
   (a) after each insertion optimize using NNI/local rearrangement (default, but user-adjustable gradually up to SPR; only fastDNAml)
   (b) repeat (a) rearrangements until no better tree found.
2. after the last insertion optimize using SPR/global rearrangement (in DNAML; in fastDNAml user-adjustable gradually down to NNI)
3. repeat (2) rearrangements until no better tree found.

Pro: Evaluating large neighborhood with SPR.

Con: Slow.

Note: To save time, in other methods steps (1) and (2) are usually substituted by swiftly computed trees (e.g., BioNJ).

ML programs: PHYML

Concept: BioNJ tree + fastNNI

1. Start with BioNJ tree.
2. Do fastNNIs to optimize trees, i.e., evaluate all NNIs simultaneously and then merge all best ones which are non-conflicting.
3. Repeat (1) until no better tree found anymore.

Pro: Fast

Con: Prone to get stuck on local optima due to NNI-only.

ML programs: IQPNNI

Concept: BioNJ tree + randomization + fastNNI

1. Start with BioNJ tree.
2. Do fastNNIs to optimize trees, i.e., evaluate all NNIs simultaneously and then accept all best ones which are non-conflicting. (after first round, identical to PHYML).
3. Remove randomly a certain amount of taxa and re-insert them by a fast and rough quartet-based method. (some randomization)
4. Repeat (2)-(3) until stop criterion is met.

Pro: Can evade local optima,
offers automatic stopping criterion,
hints when search didn't run enough,
numerically optimized ML computation,
offers codon models

Con: slower than PhyML/RAxML

ML programs:

- RAxML
- GARLI
- MetaPiga
- SSA
- TREE-PUZZLE
- MOLPHY
- `http://evolution.genetics.washington.edu/phylip/software.html`

How reliable is the reconstructed tree:

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

Quartet Puzzling

The Quartet Puzzling algorithm implemented in the TREE-PUZZLE program is a three step procedure:

maximum-likelihood step: compute ML trees for all quartets of an alignment.

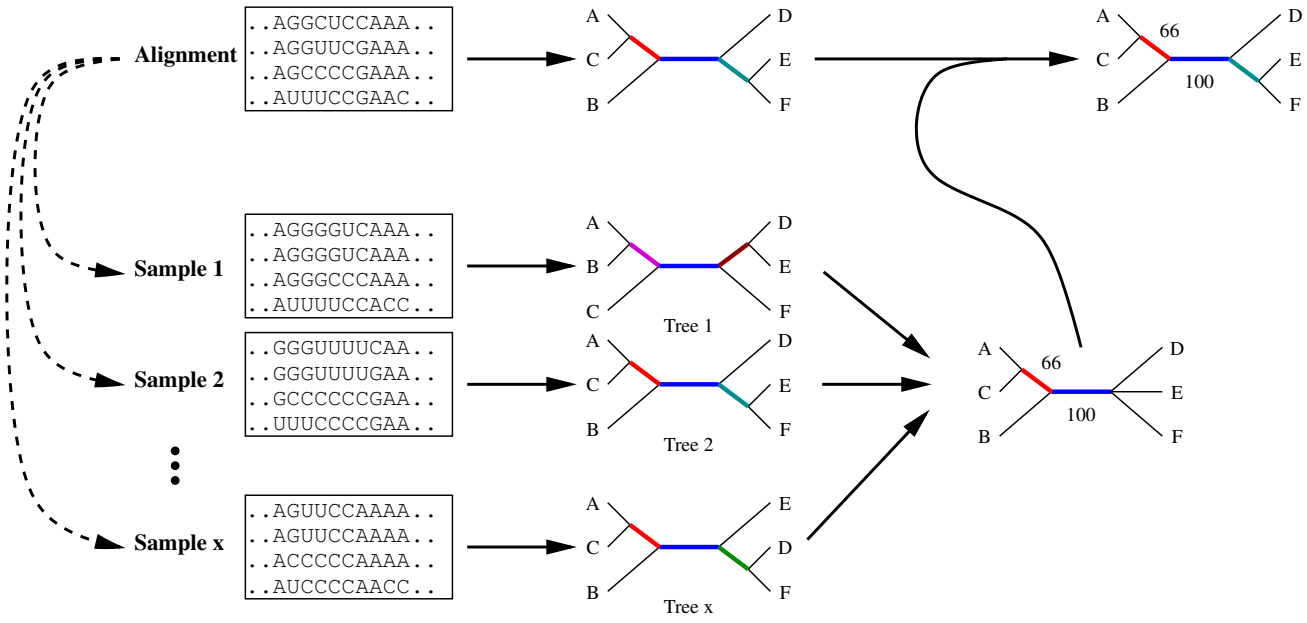
puzzling step: compose intermediate tree from quartet trees (this is done multiple times).

consensus step: construct a majority rule consensus tree from the intermediate trees and evaluate the branch lengths.

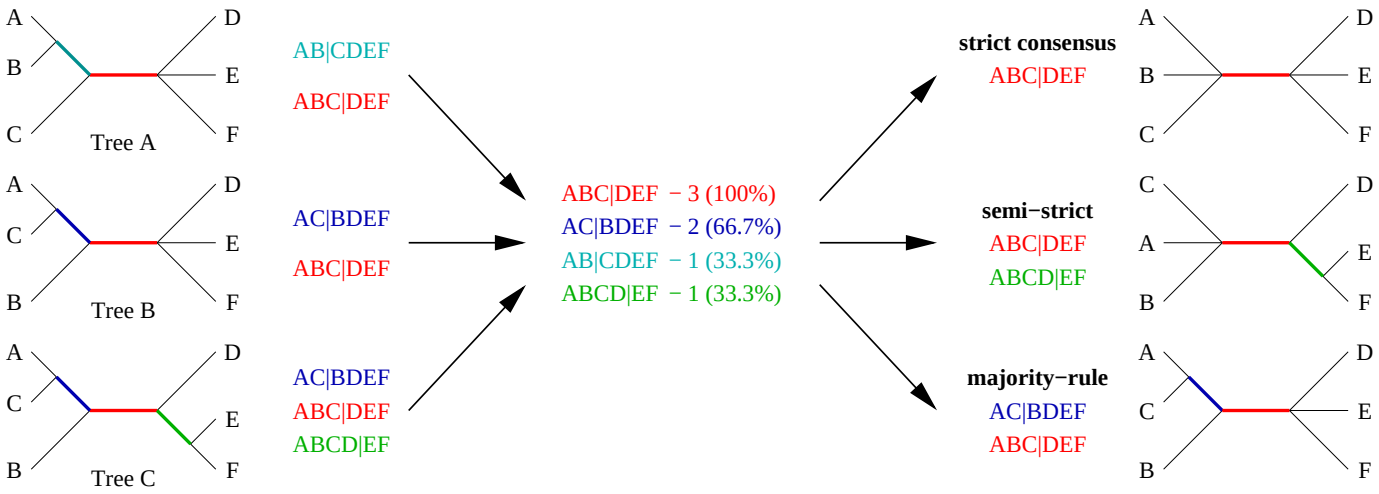
Branch Support

- We can now reconstruct ML trees, but how comparable are the likelihoods, how reliable the groupings?
- Branch reliability can be checked, support values computed using:
  - Randomizing input orders in stepwise insertions (TREE-PUZZLE).
  - Jackknifing alignment columns + consensus.
  - Bootstrapping alignment columns + consensus.
  - Trees from Bayesian MCMC sampling + consensus.

Estimating Confidence: The Bootstrap



Summarizing Trees: Consensus Methods



Summarizing Trees: Consensus Methods

- Majority-based: (Sorted splits added in descending order)
  - Strict consensus: all splits found in all trees
  - Semi-Strict consensus: all splits uncontradicted in all trees
  - Majority Rule Consensus M_ℓ: all splits found in more than fraction ℓ of the trees (typically $\ell = 0.5$).
  - Relative Majoroty Consensus: all splits even below 0.5 down to the first incongruence.
  - Majority Rule extended (MRe): incompatible splits are discarded and all added that are compatible with incorporated splits.
- Adams consensus: reflects common nestings

Posterior Probabilities and Empirical Bayes

- Problem: How different are likelihoods? Just from the value of likelihoods one often cannot tell whether they are significantly different.
- Normalization: Posterior probabilities are computed:

$$p_i = \frac{L_1}{\sum_n L_n}$$

- Usage:
  - Which sites along an alignment support a tree most?
  - Are there sites/partitions not supporting a tree?
  - Which model of evolution (e.g. dependent, independent) is supported by which site/partition? (PAML)
  - Is a site fast/medium/slowly evolving? (PAML, TREE-PUZZLE)
  - Constructing confidence sets on posterior tree likelihoods (MrBayes)

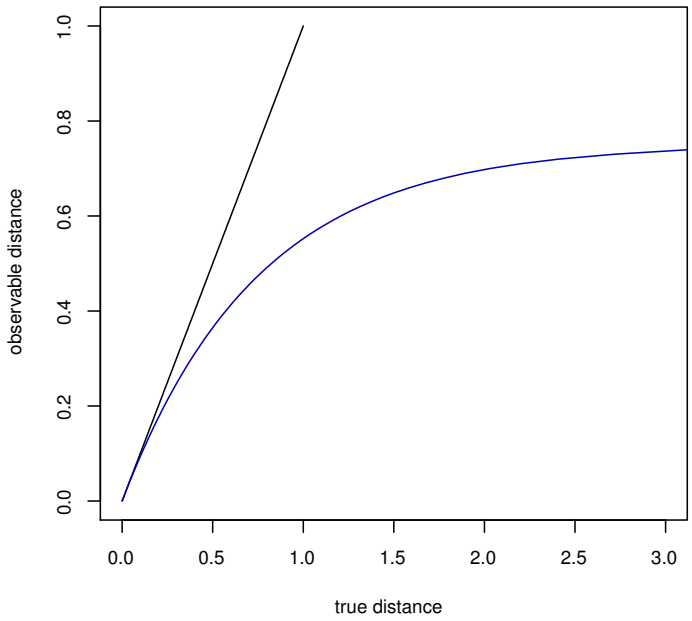
Phylogenetic Information

The information about the true tree, might be obscured or unextractable from an alignment due to

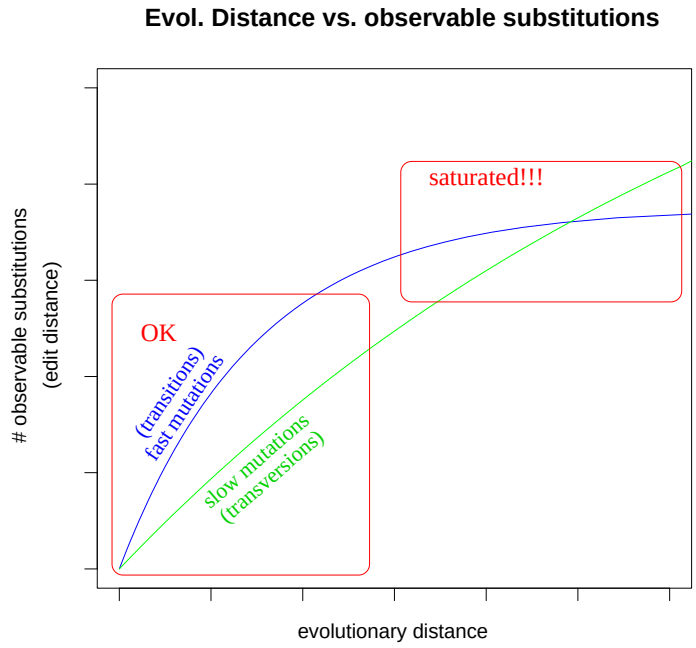
- too similar sequences
  (no differences $\rightarrow$ no information)
- sequences are to divergent
  (saturated sequences $\rightarrow$ information drowned in noise)

Are there ways to check for this?

Reminder: Jukes-Canter Correction for Multiple Hits



Plotting Substitutions vs. Distance for DNA



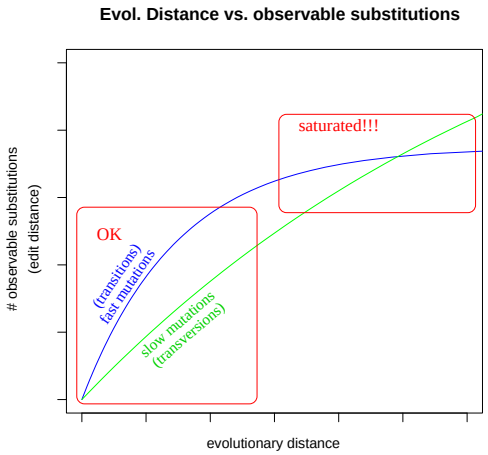
Transitions (ts) usually occur much more often the transversions (tv).
Thus, the ts-curve rises faster but reaches the plateau earlier.

The tv-curve can only 'overtake' the ts-curve if the latter is quite saturated!

Saturation Plots for DNA

Saturation Plots can be created as follows

- Take every pair of sequences
  - Count the number of observable substitutions (e.g., transitions, transversions)
  - Compute the distances of the sequence pair (e.g., with ML)
- . . . and plot the evolutionary distance (x-axis) against the observed substitutions (y-axis) for each class of mutations.



Software for DNA Saturation Plots

Saturation Plots can be created using

- **Windows:** DAMBE (Xia and Xie, 2001)

  → `Graphics` menu
  → `Transition and transversion vs. divergence`

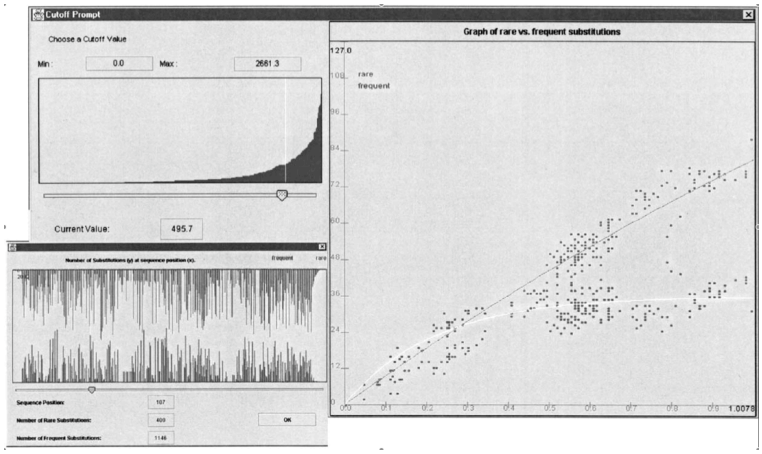
- **All OS:** TREE-PUZZLE (Schmidt *et al.*, 2001), plotting the data in `*.tstv` with a few lines in the `R` program (`www.r-project.org`):

```
tstvtab = read.table("ali.phy.tstv", header=T)  # read data
attach(tstv)                                    # use headers as names
pdf(file="tstv.pdf")                            # open PDF file
maxsubst=max(ts,tv)                             # find maximum
plot(distance,ts,col=2,ylab="observed substitutions",ylim=c(0,maxsubst))
points(distance,tv,col=3)                       # plot
dev.off()                                       # close PDF file
detach(tstvtab)                                 # release names
q()                                             # quit R program
```

Saturation Plots for AA (AsaturA, van de Peer et al. 2002)

The same can be done for amino acids, but

- There is not intuitive division into fast and slow substitutions,
- AsaturaA orders the the substitution types according to the probabilities in a substitution probability matrix (e.g., PAM, WAG).
- Then, the user has to set a cutoff between *fast* and *slow*.
  (But there are no guidelines for that choice.)
- Then the numbers of fast and slow substitutions are plotted against the distance accordingly.

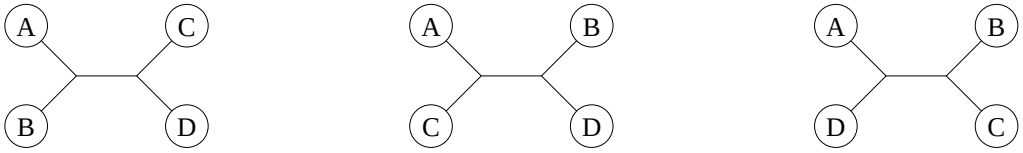


Likelihood Weights, Posterior Prob., and Empirical Bayes

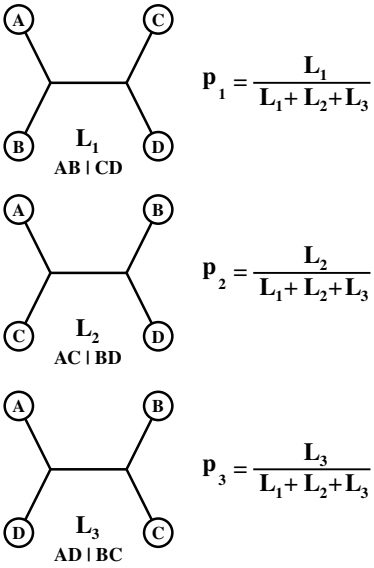
- We can compute a likelihood value for a tree based given an alignment and model... (cf. the *lecture on ML methods*).
- Problem: How different are the likelihoods?
  Just from the value of likelihoods one often cannot tell whether they are significantly different.
- Normalization: Posterior probabilities are computed:

$$p_i = \frac{L_i}{\sum_n L_n}$$

- We can use that on the three different quartet topologies to assess phylogenetic information in our data.



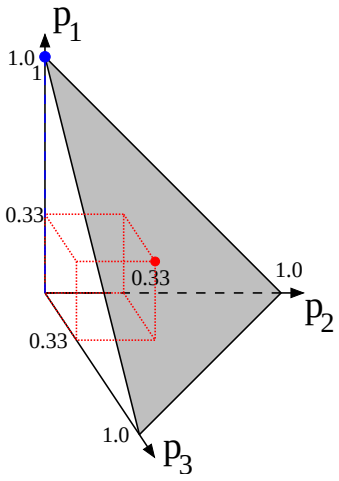
Plotting Posteriors: Likelihood Mapping

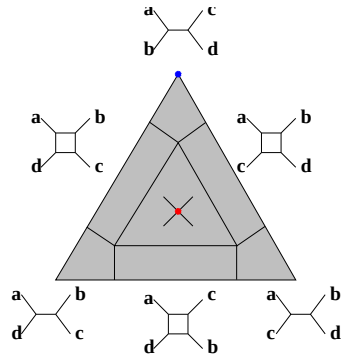


$$p_1 = \frac{L_1}{L_1 + L_2 + L_3}$$

$$p_2 = \frac{L_2}{L_1 + L_2 + L_3}$$

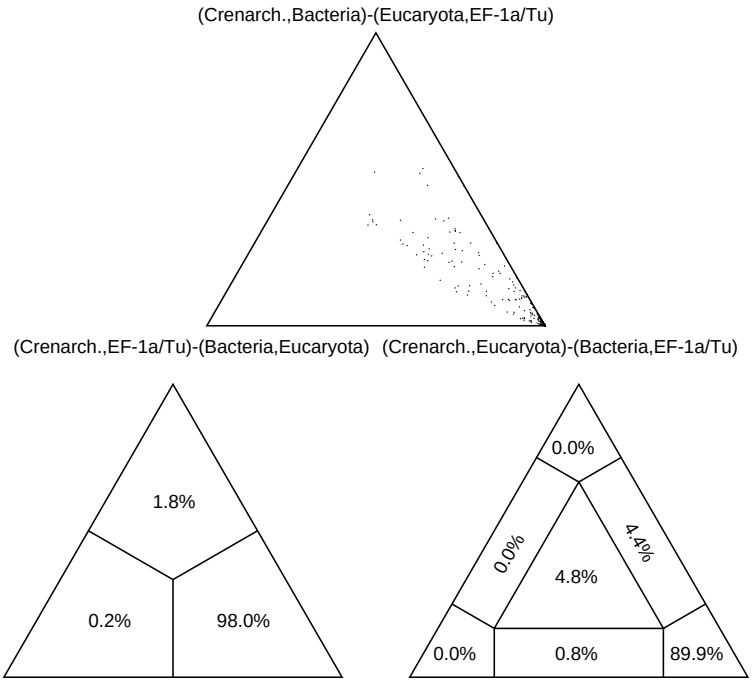
$$p_3 = \frac{L_3}{L_1 + L_2 + L_3}$$





Since $p_1 + p_2 + p_3 = 1$, 3D points (p_1, p_2, p_3) fall into a triangular (simplex).
If we repeat this for all quartets (or a large random subset) in a dataset we can assess the amount of phylogenetic signal in the dataset.

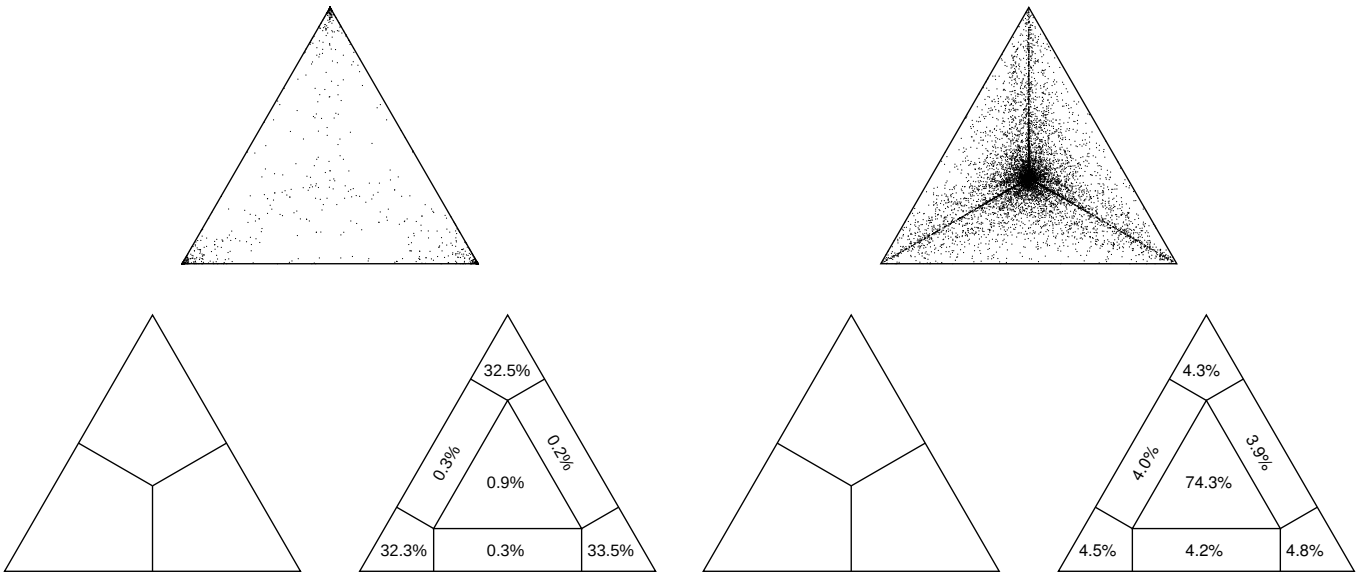
Likelihood Mapping (Cluster Analysis)



The Simplex Plot can visualize the relationship among (4) sets of taxa.

The taxa/sequences are assigned to four sets (A,B,C,D) one for each leaf of a quartet tree.

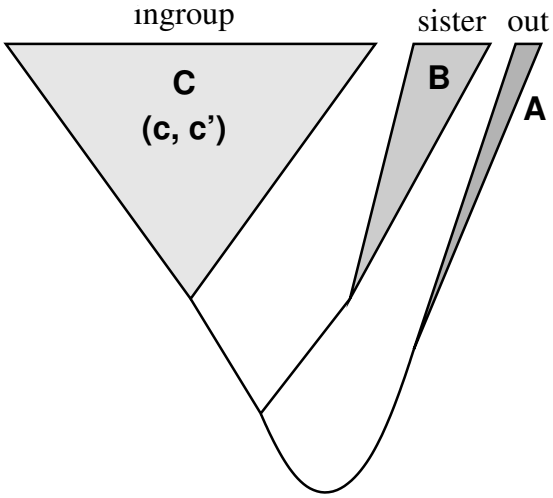
Likelihood Mapping (Information Content)



The Simplex Plot can also visualize the information content in an alignment.

By not assigning taxa to clusters, four are chosen randomly for each leaf. We have to add the percentages in the corners (resolved) or rectangles (partly resolved), respectively. Center means unresolved.

Likelihood Mapping to Validate Outgroups



- We can check the reliability of an outgroup by assigning taxa to three sets: C - the examined ingroup, B - an early sister group, and A - the outgroup.
- random quartets are drawn from the sets: two from C and one each from B and A.
- if not $a, b|c, c'$ (upper corner) is the support topology, A is not a good outgroup (or B is not a proper sister group).

Exercises:

the exercises can be found at

`http://www.cibiv.at/~ingo/applied_bioinf`