Maximum Likelihood Methods in Phylogenetics

Heiko A. Schmidt

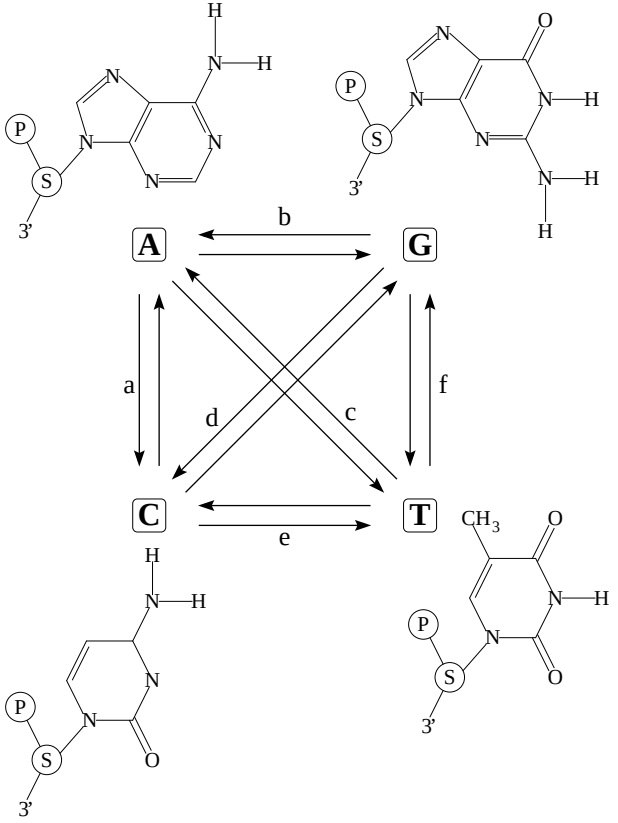
CIBIV - Center for Integrative Bioinformatics Vienna
Max F. Perutz Laboratories (MFPL)
Vienna, Austria
heiko.schmidt@univie.ac.at

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

Substitution Models

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



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

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

From Substitution rates to probabilities

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

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

(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.

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

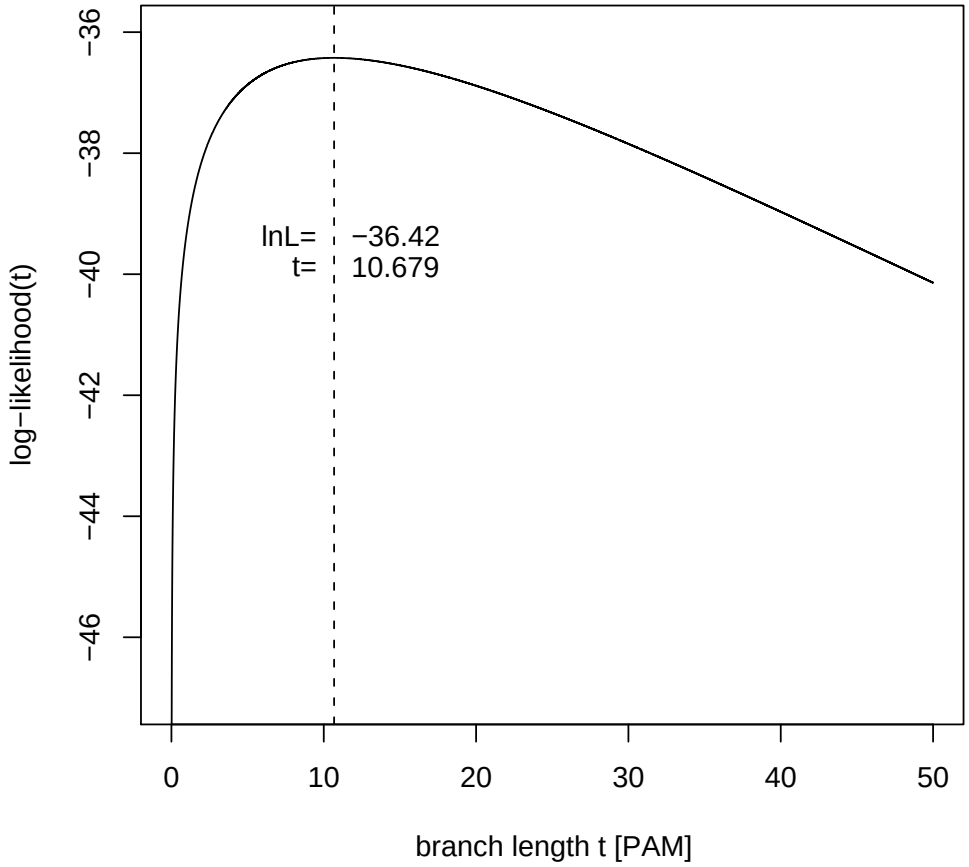
The Likelihood of sequence s evolving to s' in time t:

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

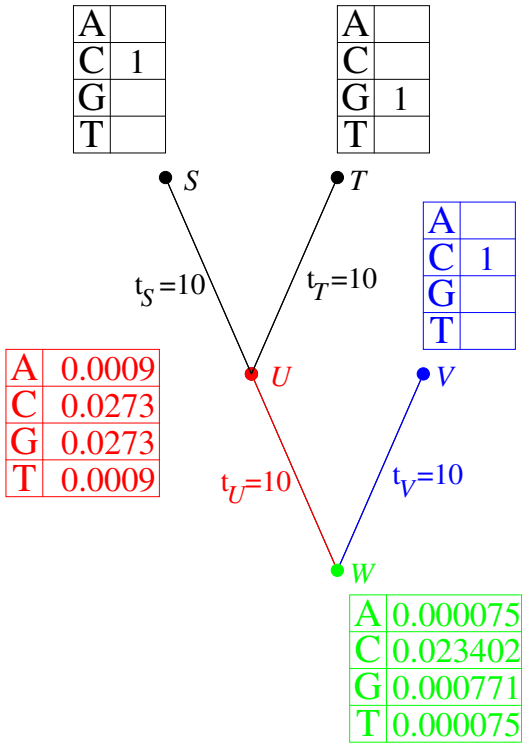
Likelihood surface for two sequences under JC69:

GATCCTGAGAGAAATAAAC
GGTCCTGACAGAAATAAAC

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



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



Likelihoods of nucleotides at inner nodes:

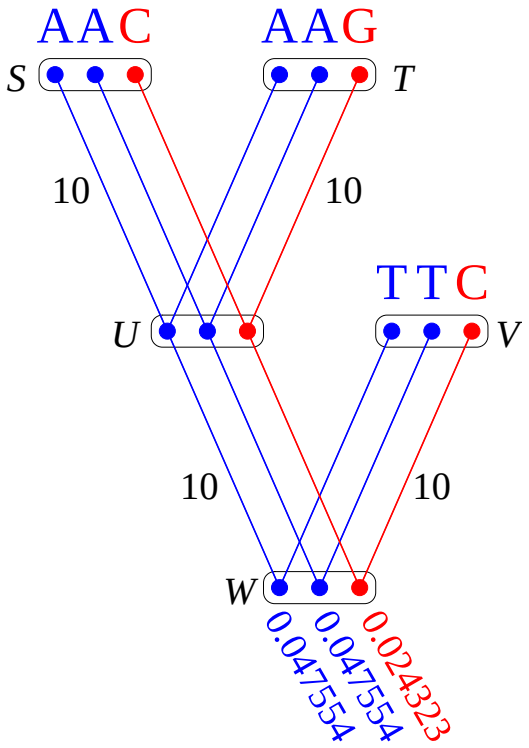
$$L_U(i) = [P_{iC}(10) \cdot L(C)] \cdot [P_{iG}(10) \cdot L(G)]$$

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

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

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

$$= 0.000055$$

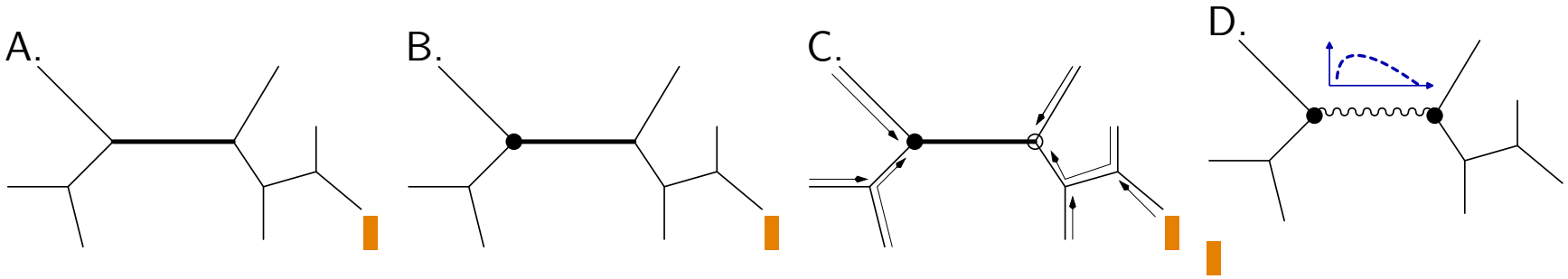
or the log-likelihood

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

Adjusting Branch Lengths Step-By-Step

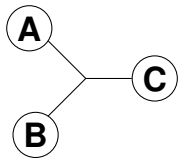
To compute optimal branch lengths do the following. Initialize the branch lengths.

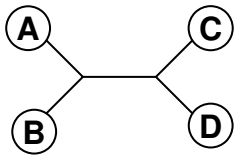
Choose a branch (A.). Move the virtual root to an adjacent node (B.). Compute all partial likelihoods recursively (C.). Adjust the branch length to maximize the likelihood value (D.).



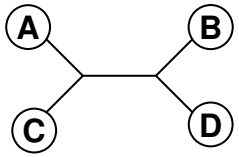
Repeat this for every branch until no better likelihood is gained.

Number of Trees to Examine. . .

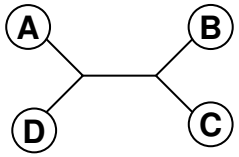




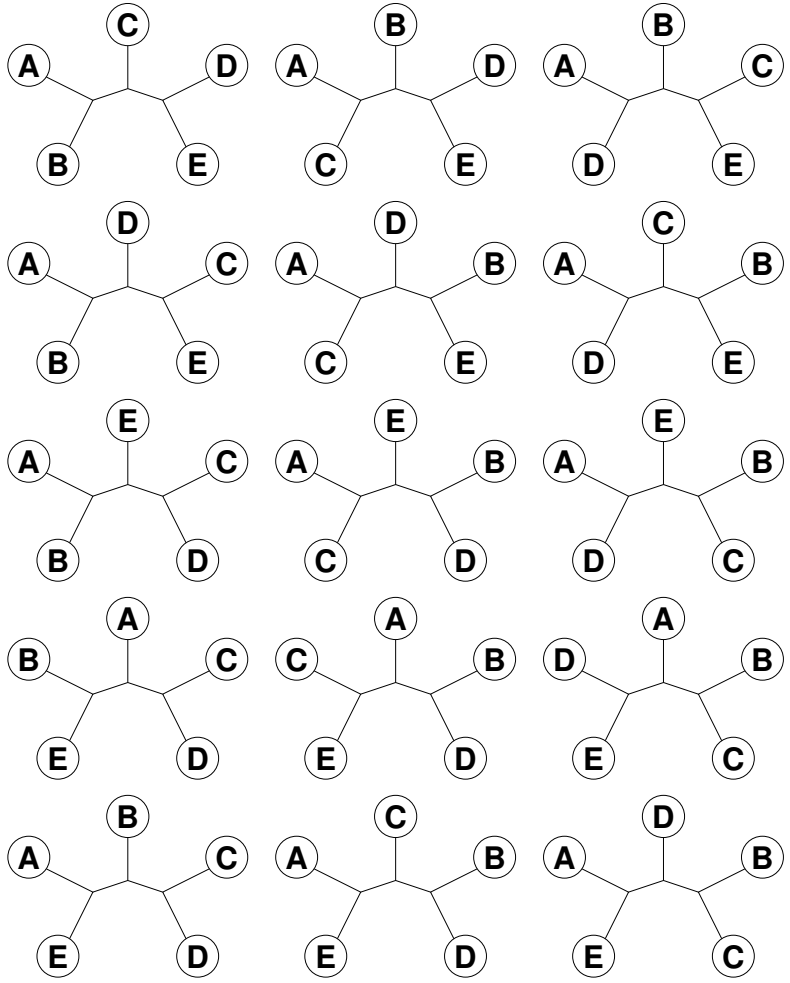
$L_{AB|CD}$



$L_{AC|BD}$



$L_{AD|BC}$



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

$$B(10) = 2027025$$

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

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

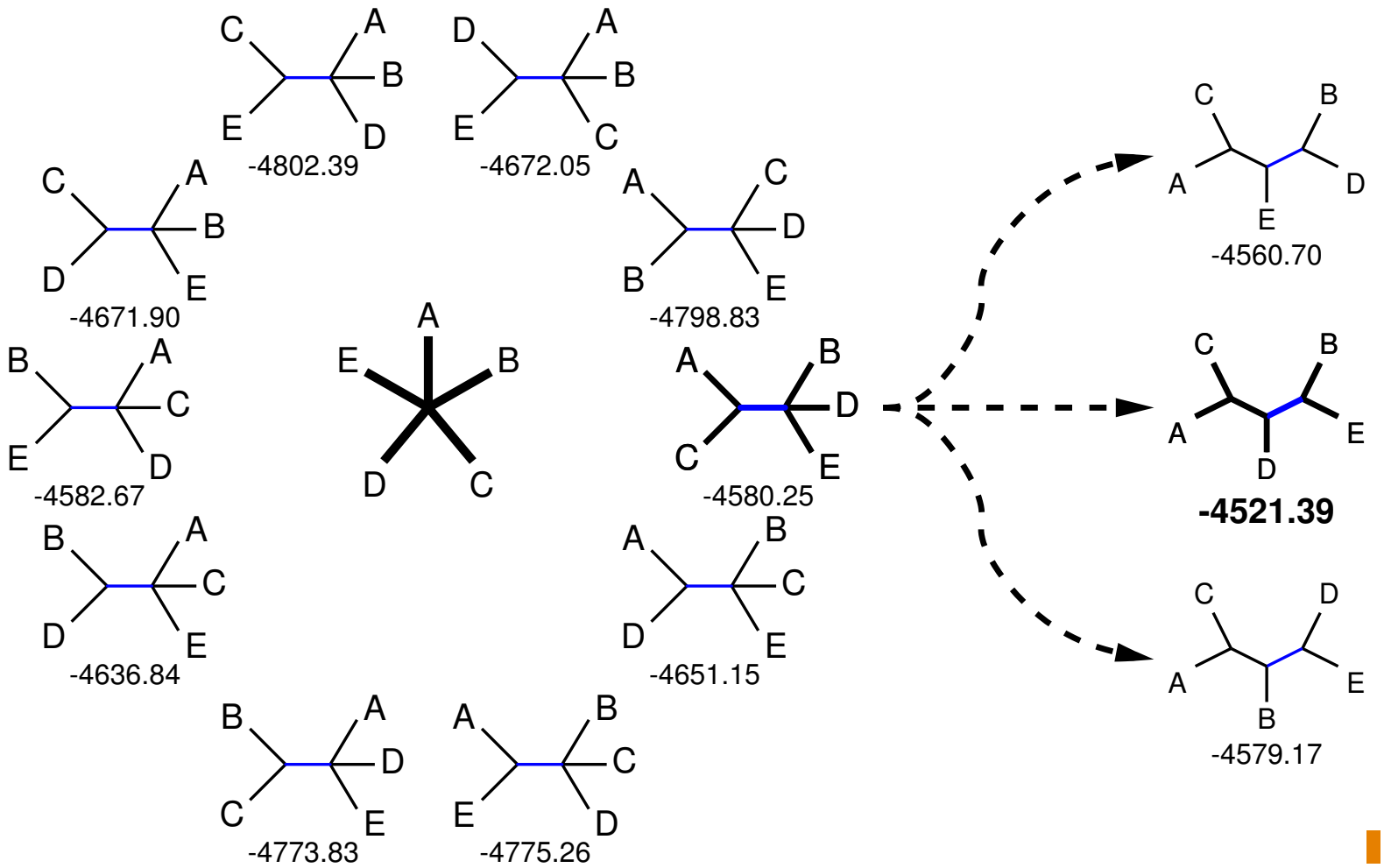
Finding the ML Tree

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

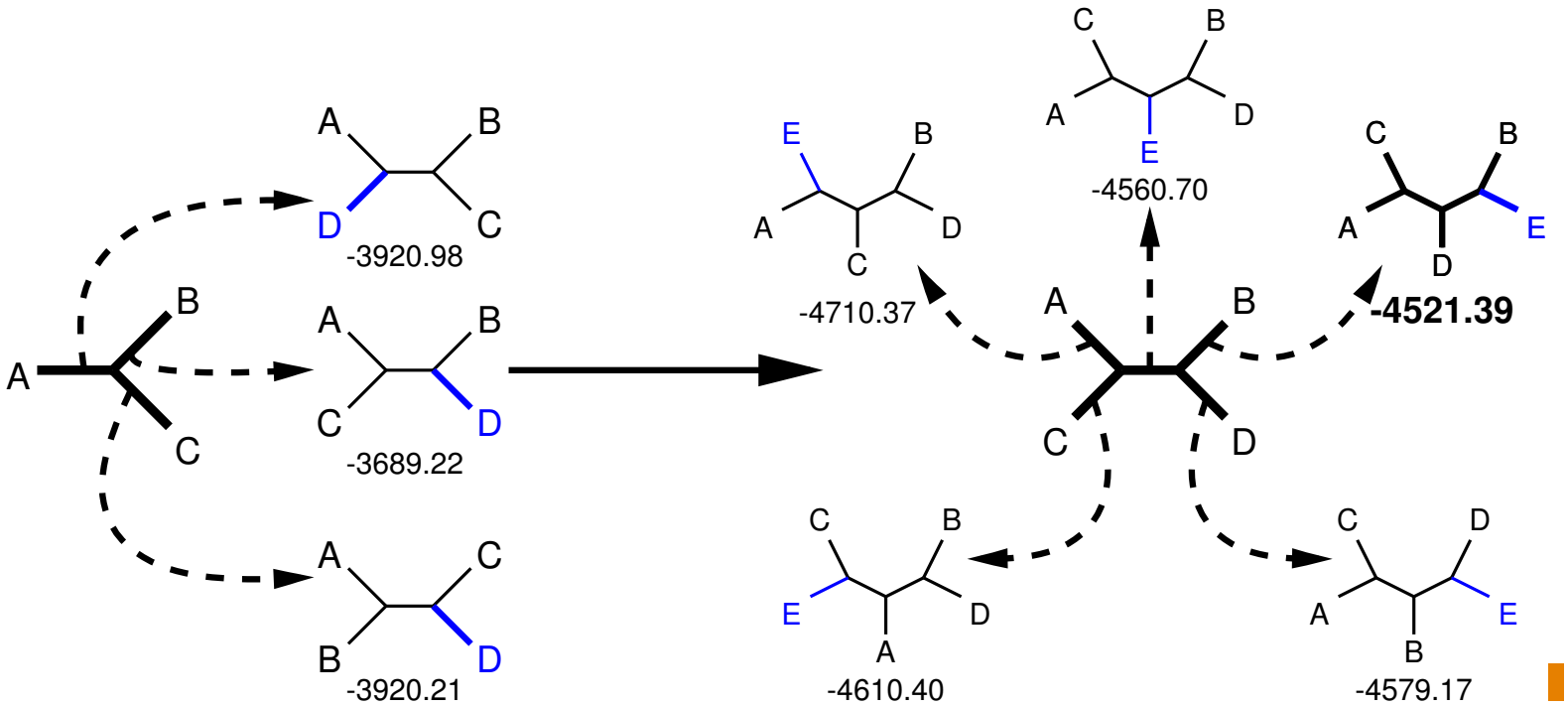
Branch and Bound: guarantees to find the optimal tree, without searching certain parts of the tree space – can run on more sequences, but often not for current-day datasets.

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

Build up a tree: Star Decomposition

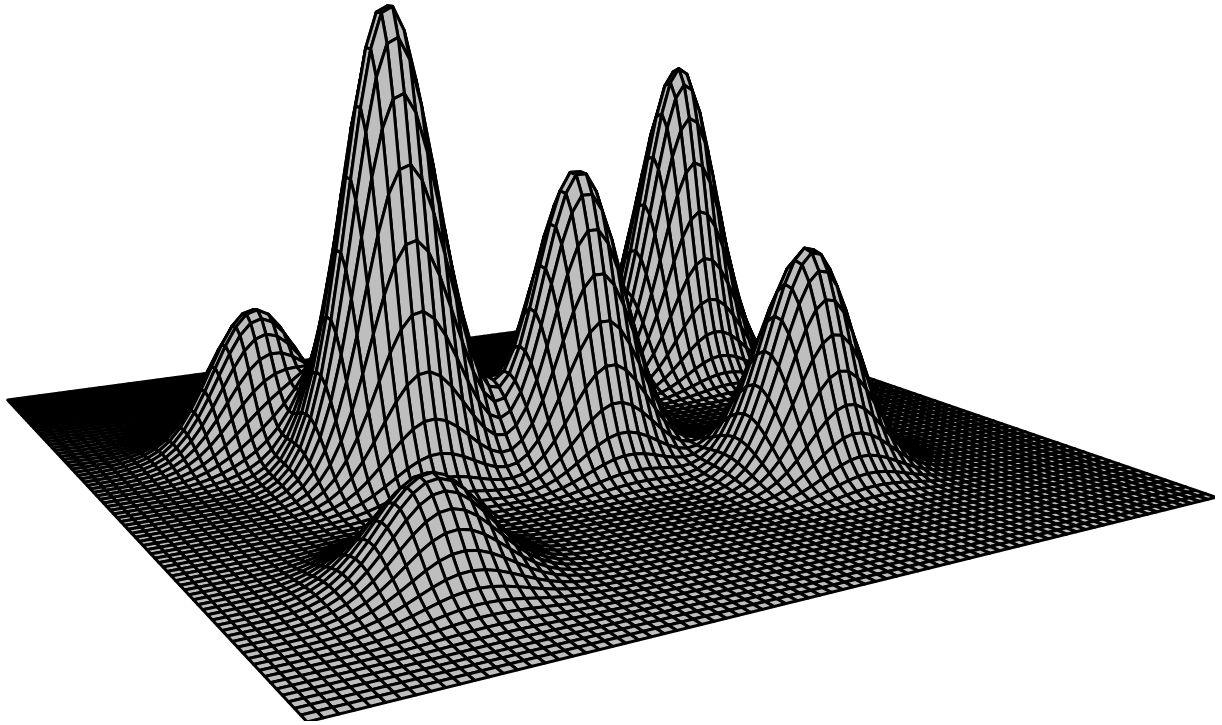


Build up a tree: Stepwise Insertion



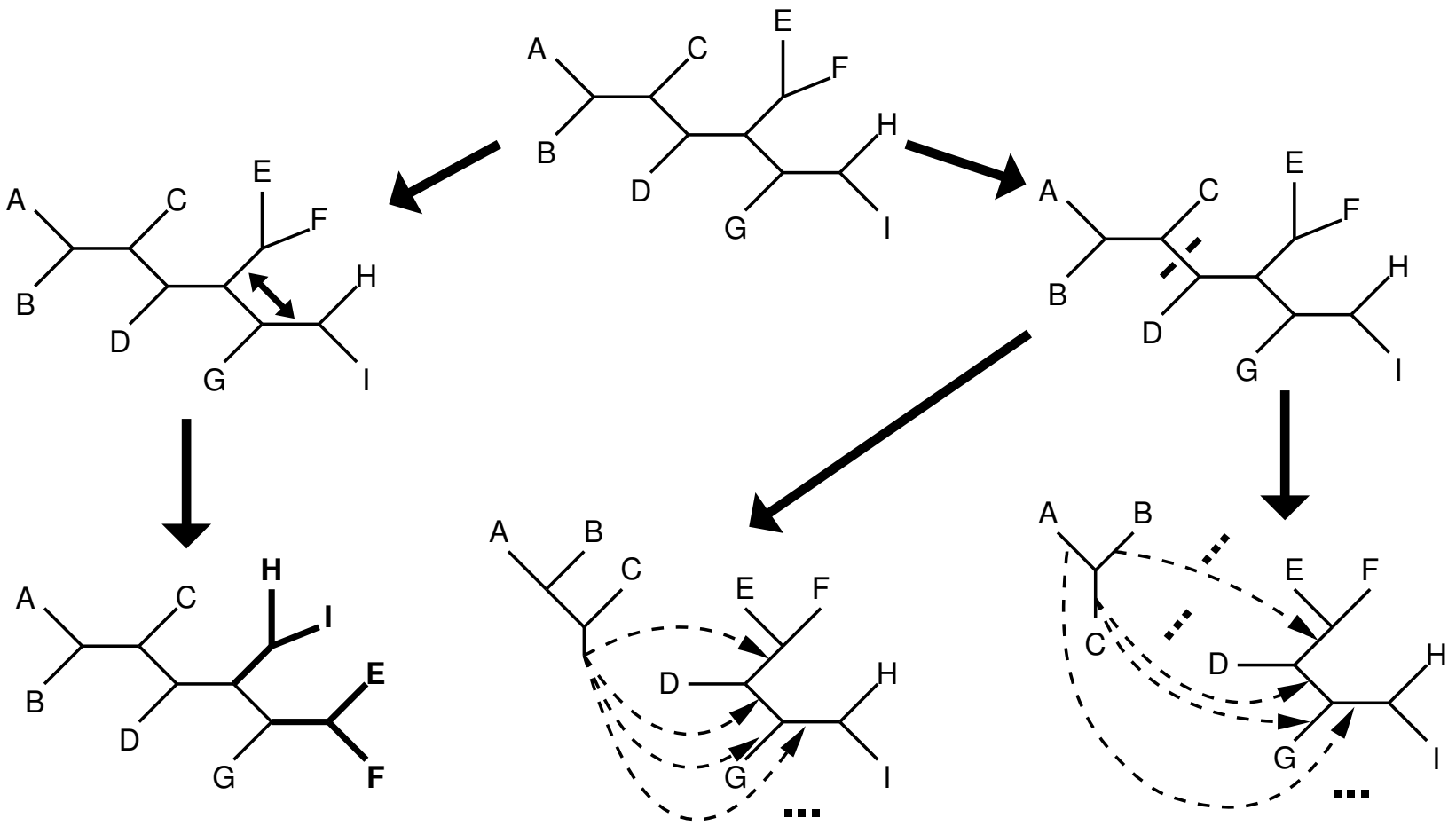
Local Maxima

What if we have multiple maxima in the likelihood surface?



Tree rearrangements to escape local maxima.

Tree Rearrangements



Nearest Neighbor Interchange

Possible NNI trees = O(n)

subtree pruning + regrafting

Possible SPR trees = O(n*n)

tree-bisection + reconnection

Possible TBR trees = O(n*n*n)

ML programs: DNAML (PHYLIP), fastDNAml

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

ML programs: MOLPHY

- Build tree with star decomposition
- after the last insertion optimize using SPR/global rearrangement (in DNAML; in fastDNAml user-adjustable gradually down to NNI)
- repeat rearrangements until no better tree found.

ML programs: (R)AxML family

- Descendant on fastDNAml, but . . .
- Starting with MP tree.
- Many smart algorithmic and numerical optimized ML computation.
- Uses *lazy rearrangements*, i.e., only the 3 insertion branches are optimized.
- (Several versions with slightly different algorithms, e.g., Simulated Annealing.)

ML programs: PHYML

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

ML programs: PHYML-SQP

- Start with BioNJ tree.
- Evaluate SQP by fast non-ML criterion to find best candidates.
- Evaluate the candidate(s) more rigorously with ML and fastNNI.
- Repeat until no better tree found anymore.

ML programs: IQPNNI

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 until stop criterion is met.

ML programs: Genetic Algorithms (GARLI, MetaPIGA)

- Start with some (random) tree.
- View tree topology, branch lengths, and model parameter as part of a 'genome'.
- Evolve the 'genome' by mutating (slightly changing) its parts.
- Accept or reject new tree topologies from a pool of suggested trees according to their likelihood.

ML programs: Simulated Annealing

- Start with some (random) tree.
- Start a '*hot chain*' to suggest tree topologies (being far away.
- Accept proposals according to their likelihood.
- Cool down the chain, until the suggestions end up in some (local) optimum.

Quartet Puzzling

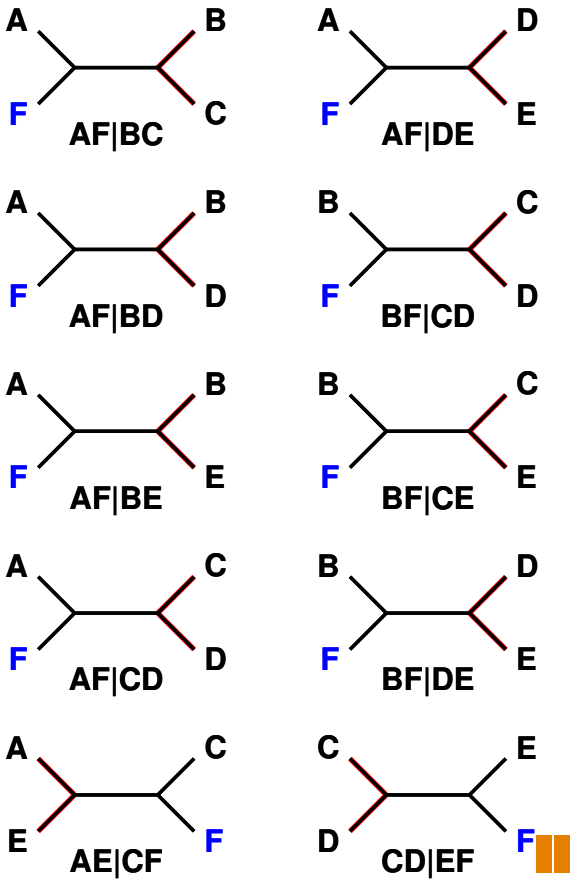
The Quartet Puzzling algorithm implemented in the TREE-PUZZLE program is a three step procedure:

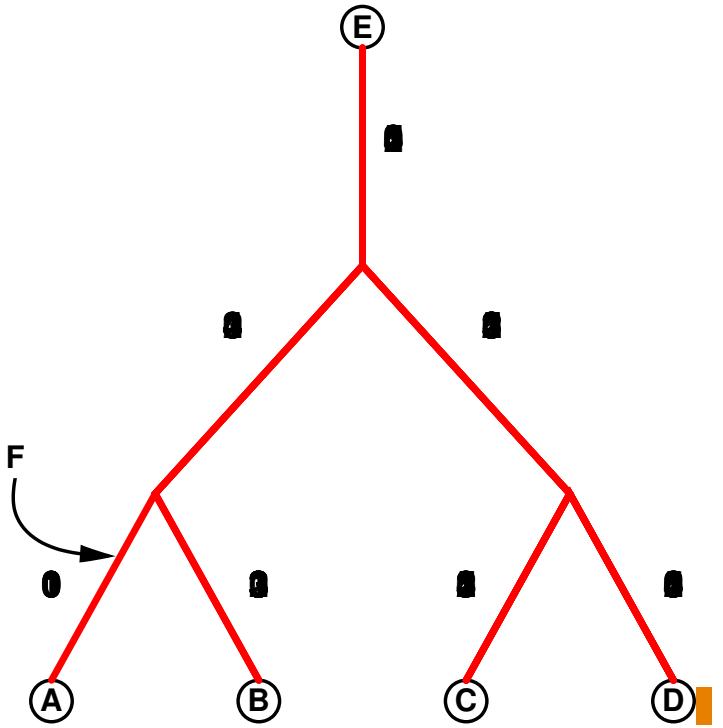
maximum-likelihood step: compute ML trees for all quartets of an alignment.

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.

Puzzling Step





Posterior Probabilities and Empirical Bayes

- 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:
  - Bootstrapping, Jackknifing alignment columns + consensus.
  - Randomizing input orders in stepwise insertions (TREE-PUZZLE).

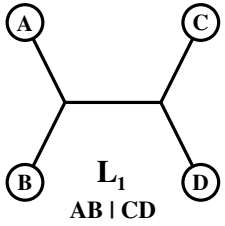
Posterior Probabilities and Empirical Bayes

- 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:

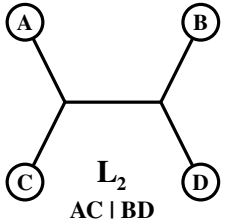
$$p_i = \frac{L_1}{\sum_n L_n}$$

- 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)

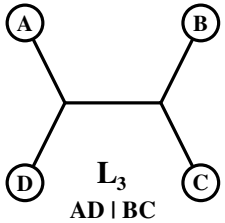
Plotting Posteriors: Likelihood Mapping



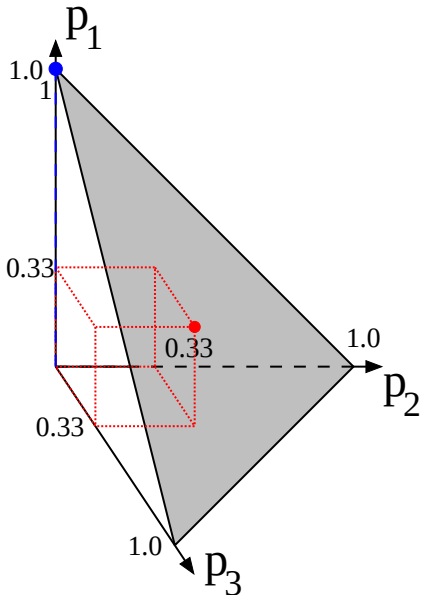
$$p_1 = \frac{L_1}{L_1 + L_2 + L_3}$$

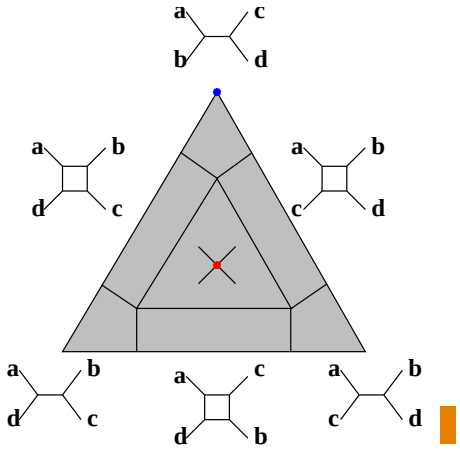


$$p_2 = \frac{L_2}{L_1 + L_2 + L_3}$$



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

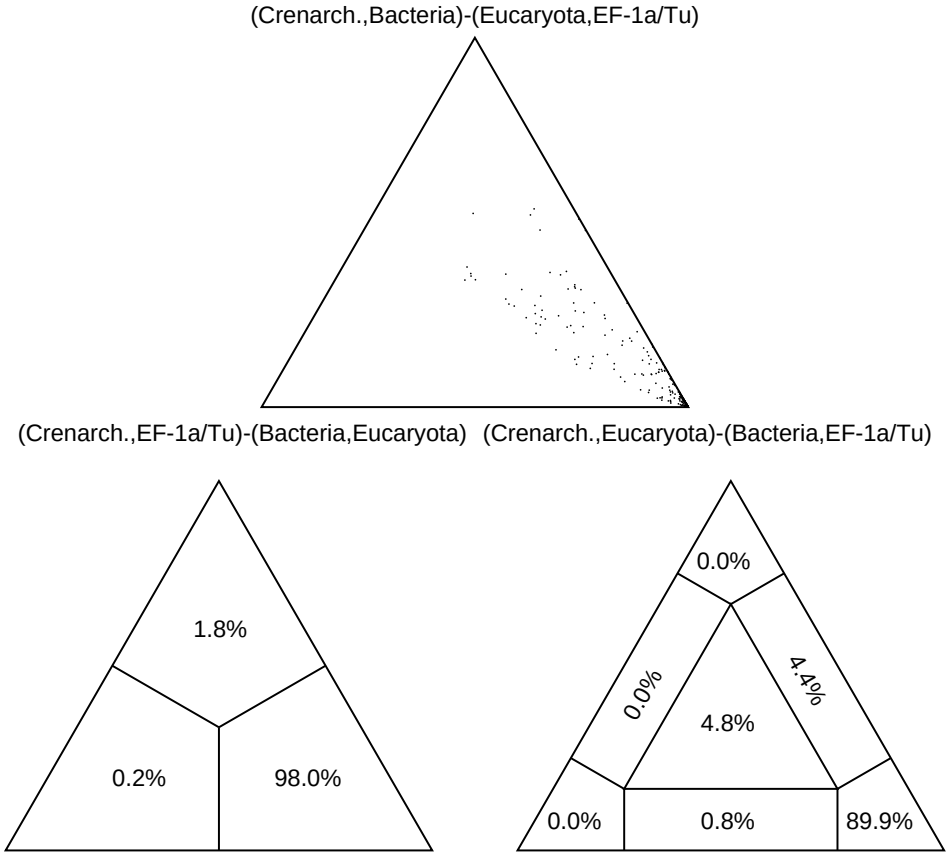




Since $p_1 + p_2 + p_3 = 1$, 3D points (p_1, p_2, p_3) fall into a triangular (simplex).

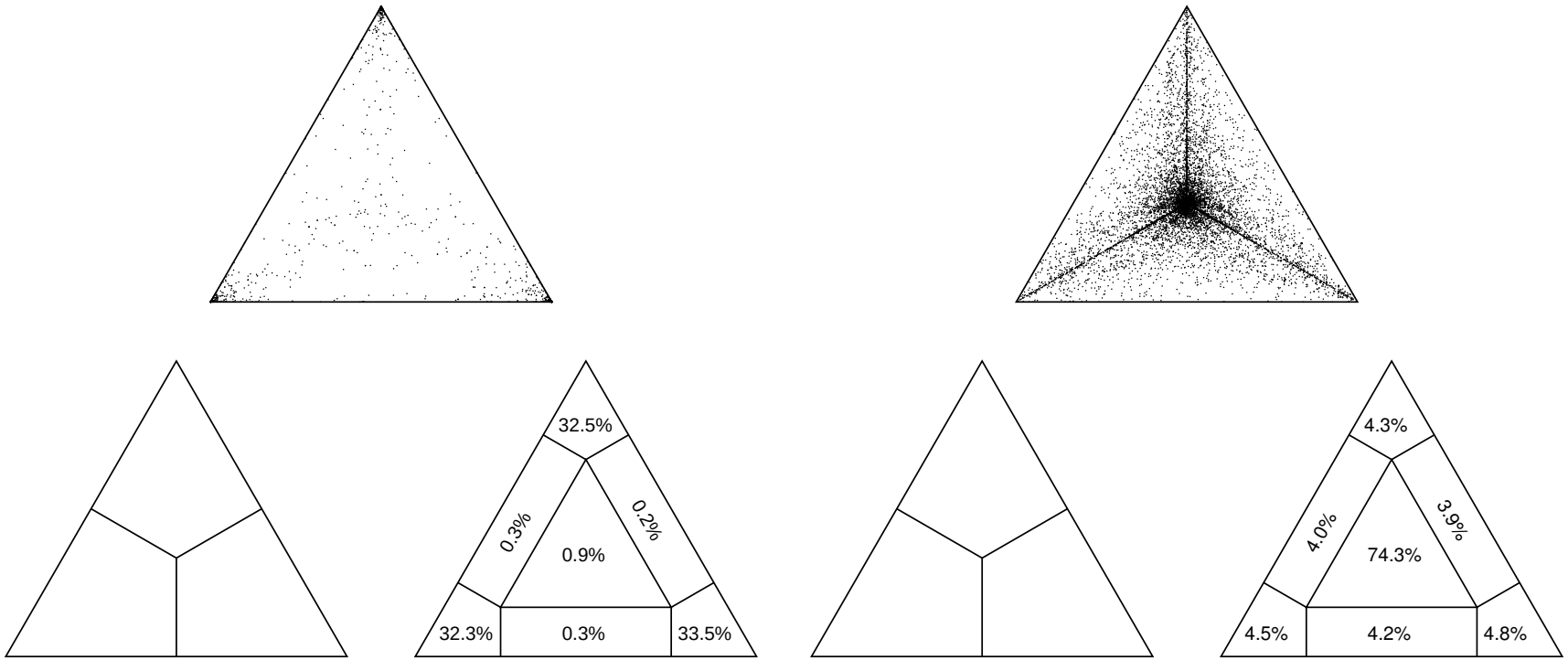
If we repeat this for all quartets (or a large random subset) in a dataset we can assess the amount of phylogenetic signal in the dataset.

Likelihood Mapping (Cluster Analysis)



The Simplex Plot can visualize the relationship among clusters.

Likelihood Mapping (Information Content)



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

LRT – Likelihood Ratio Test (1)

The Likelihood function offers a natural way of comparing nested evolutionary hypothesis using the Likelihood Ratio (LR) statistics:

$$\Delta = 2(\ln L_1 - \ln L_0)$$

L_1 maximum likelihood under the more parameter-rich, complex model (alternative hypothesis, H_A)

L_0 maximum likelihood under the less parameter-rich simple model (Null-hypothesis, H_0)

If the models are nested, i.e., H_0 is a special case of H_A and the Null-hypothesis (H0) is correct, Δ is asymptotically χ^2-distributed with the number of degrees of freedom equal to the difference in number of free parameters between the two models.

LRT – Likelihood Ratio Test (2)

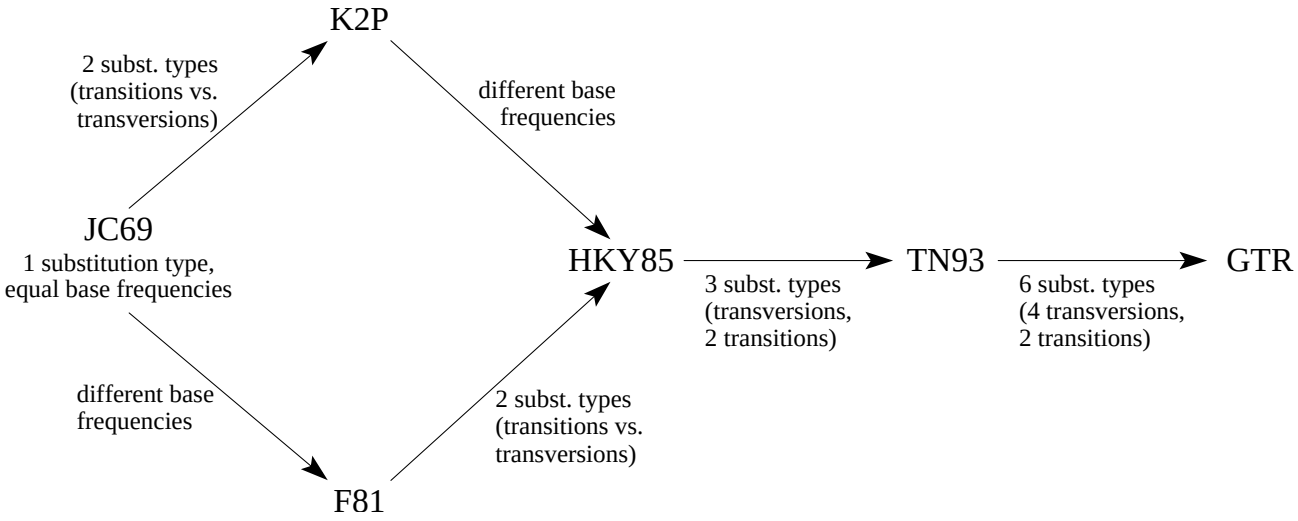
- If the LRT is significant (i.e., $p < 0.05$ or $p < 0.01$): the use of the additional parameters in the alternative model H_A increases the likelihood significantly.

- If Δ is close to zero, that is, $p > 0.05$: the alternative hypothesis H_A does not fit the data significantly better than H_0, that means using the additional parameters of H_A does not explain the data better.

- Only nested models can be tested:
  One model (H_0, Null-model, constraint model) is nested in another model (H_A, alternative, unconstraint model) if the model H_0 can be produced by restricting parameters in model H_A.

LRT – Typical cases of nested models

- Different levels of evolutionary models:



- *rate-homogeneous models* (H_0) are nested in *rate-heterogeneous models* (H_A)
- A tree assuming *molecular clock* (H_0) are nested its *non-clock* version (H_A)

How to Compare Tree Topologies

Since tree topologies are no normal parameters, they are generally not nested in each other. Hence, the χ^2 distribution cannot be used.

How do we get distributions of likelihood ratios for testing?

1. non-parametric bootstrap: resample columns from the alignments, optimize the given tree topologies to get distributions.

2. RELL (Rapid Estimation of Log-Likelihoods): resample directly from the site-likelihood and use these to compute likelihoods. (saves a lot of time of the optimization).

3. parametric bootstrap (Monte Carlo): from your best tree on a fixed parameters simulate datasets. Use those to re-evaluate the original trees.

How to Compare Tree Topologies: KH-test

Pairwise tests against of pairs of trees (Kishino-Hasegawa test, usually RELL).

H_0: The two trees explain the data equally well, i.e. $E(\ln \mathcal{L}_\infty - \ln \mathcal{L}_\in) = 0$.

H_A: The two trees do not explain the data equally well, i.e. $E(\ln \mathcal{L}_\infty - \ln \mathcal{L}_\in) \neq 0$.

Trees must not be taken *a priory* from an analysis of the same data.

Unfortunately, in most cases KH-test is used to compare the best tree to other trees, which ist incorrect (Goldman et al., 2000, Syst. Biol.).

How to Compare Tree Topologies: SH-test

Test designed to correctly test multiple user-defined trees (Shimodaira-Hasegawa test).The ML tree has to be always among the trees of the set!

H_0: Tree T_1 is the true tree.

H_A: Some other tree is the true tree.

Overview over Likelihood-based Analyses

- Comparing hypothesis with Likelihood-Ratio-Test (=LRT)
  - different models of evolution (ModelTest)
  - testing molecular clock assumption and root position (TREE-PUZZLE)
- Parameter estimation (TREE-PUZZLE, PAUP, ModelTest, . . .)
- Testing for phylogenetic content (TREE-PUZZLE)
- Comparing/testing different tree topologies with Kishino-Hasegawa test, Shimodaira-Hasegawa test (TREE-PUZZLE), SOWH-test, ELW
- Constructing confidence sets on posterior likelihoods (MrBayes)