Maximum likelihood methods in molecular phylogeny

An Introduction

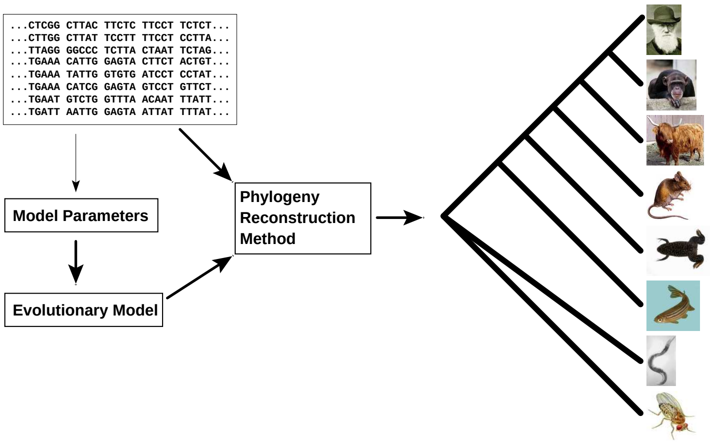
Heiko A. Schmidt

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

September 2008

Notes

Recap: Phylogenetic Reconstruction



```
...CTCGG CTTAC TTCTC TTCCT TCTCT...
...CTTGG CTTAT TCCTT TTCCT CCTTA...
...TTAGG GGCCC TCTTA CTAAT TCTAG...
...TGAAA CATTG GAGTA CTTCT ACTGT...
...TGAAA TATTG GTGTG ATCCT CCTAT...
...TGAAA CATCG GAGTA GTCCT GTTCT...
...TGAAT GTCTG GTTTA ACAAT TTATT...
...TGATT AATTG GAGTA ATTAT TTTAT...
```

Model Parameters

Evolutionary Model

Phylogeny Reconstruction Method

Notes

Main Types of Phylogenetic Methods

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

Notes

Introductory Example: ML on Coin Tossing

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

We take out one coin an toss 20 times:

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

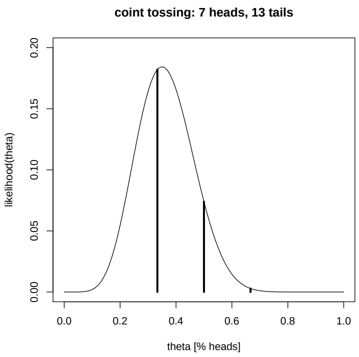
$$\begin{array}{rcl} \textbf{Probability} & & \textbf{Likelihood} \\ p(k \text{ heads in } n \text{ tosses}|\theta) & \equiv & L(\theta|k \text{ heads in } n \text{ tosses}) \\ & = & \binom{n}{k}\theta^k(1-\theta)^{n-k} \\ & & \text{(here binomial distribution)} \end{array}$$

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."*

Notes

Introductory Example: ML on Coin Tossing (Estimates)



Three coin case

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

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

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:

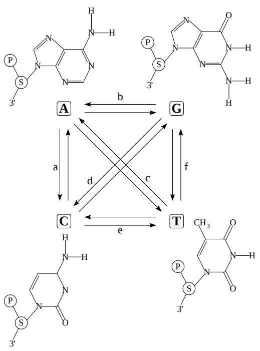
- evolutionary model
- its parameters
- tree topology
- its branch lengths

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

Notes

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

Notes

Modeling Evolution: Assumptions

- Evolution is usually modeled as a

  stationary, time-reversible Markov process.

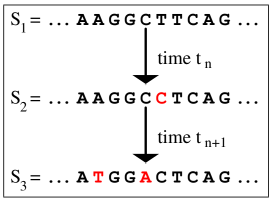
- What does that mean?

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 = ... AAGGCTTCAG ...

time t_n

S_2 = ... AAGGCCTCAG ...

time t_{n+1}

S_3 = ... ATGGACTCAG ...

Notes

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

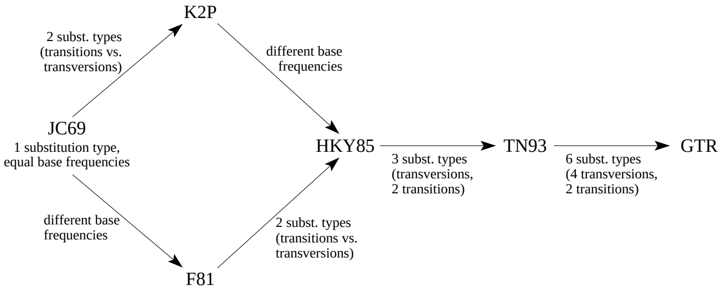
$$\pi_i \cdot P_{ij}(t) = P_{ji}(t) \cdot \pi_j$$

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

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

Notes

DNA models



K2P

2 subst. types (transitions vs. transversions)

different base frequencies

JC69
1 substitution type, equal base frequencies

HKY85

3 subst. types (transversions, 2 transitions)

TN93

6 subst. types (4 transversions, 2 transitions)

GTR

different base frequencies

2 subst. types (transitions vs. transversions)

F81

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

Notes

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) → database searching

Notes

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

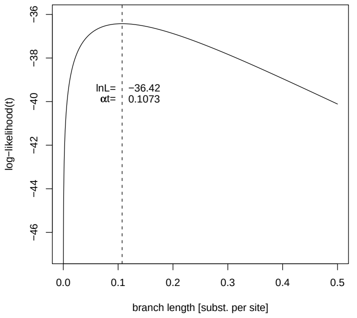
The Likelihood of sequence s evolving to s' in time t:

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

Likelihood surface for two sequences under JC69:

GATCCTGAGAGAAATAAAC $= s$
GGTCCTGACAGAAATAAAC $= s'$

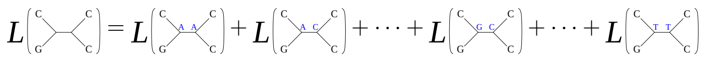
Note: we do not compute the probability of the distance t but that of the data $D = \{s, s'\}$.



Notes

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:

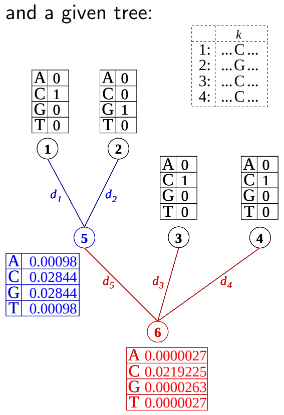


for every column in the alignment. . . but there is a fast algorithm.

Notes

Likelihoods of Trees (Single alignment column, given tree)

For a single alignment column and a given tree:



Likelihoods of nucleotides i at inner nodes:

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

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

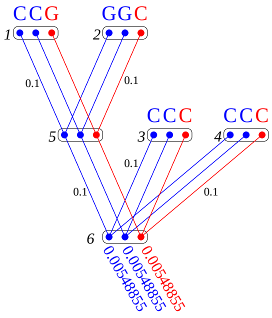
Site-Likelihood of an alignment column k:

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

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

Notes

Likelihoods of Trees (multiple columns)



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

$$\mathcal{L}(T) = \prod_{k=1}^{m} L^{(k)} \quad = \quad 0.005489^2 \cdot 0.005489$$
$$= \quad 0.0000001653381$$

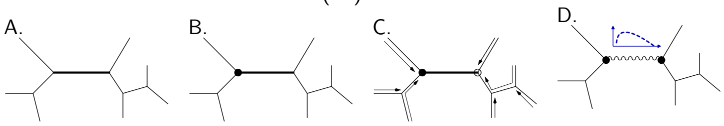
or the log-likelihood

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

Notes

Adjusting Branch Lengths Step-By-Step

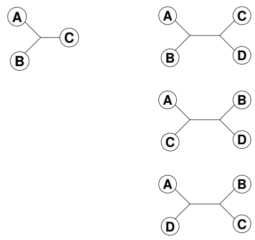
To compute optimal branch lengths do the following. Initialize the branch lengths.
Choose a branch (A.). Move the virtual root to an adjacent node (B.). Compute all partial likelihoods recursively (C.). Adjust the branch length to maximize the likelihood value (D.).

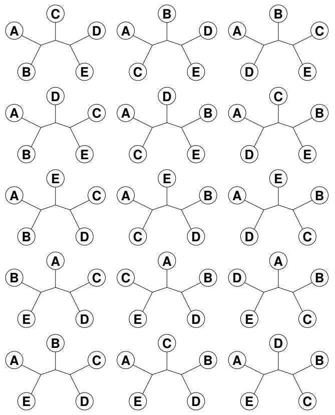


Repeat this for every branch until no better likelihood is gained.

Notes

Number of Trees to Examine. . .





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

Notes

Finding the ML Tree

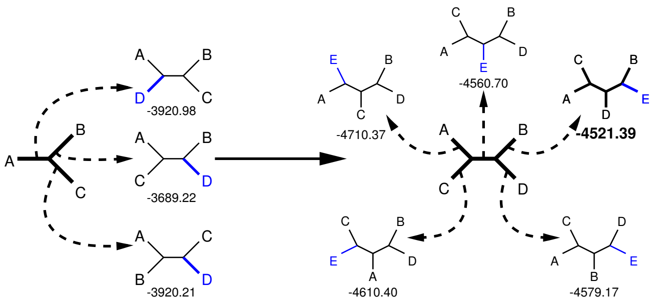
Exhaustive Search: guarantees to find the optimal tree, because all trees are evaluated, but not feasible for more than 10-12 taxa.

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.

Notes

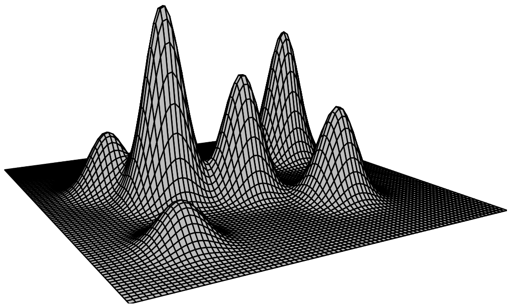
Build up a tree: Stepwise Insertion



Notes

Local Maxima

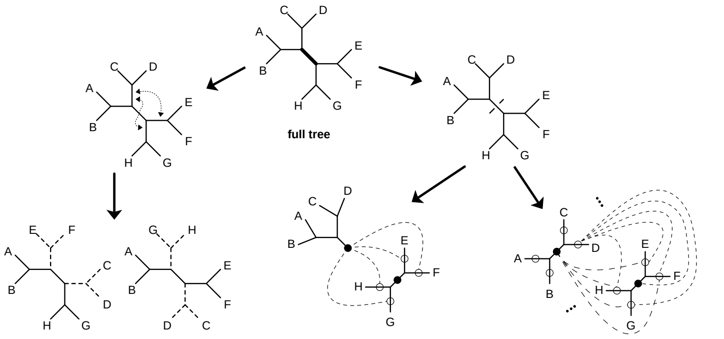
What if we have multiple maxima in the likelihood surface?



Tree rearrangements to escape local maxima.

Notes

Tree Rearrangements: Scanning a Tree's Neighborhood



Nearest Neighbor Interchange (NNI)
O(n) NNI trees

Subtree Pruning + Regrafting (SPR)
O(n^2) SPR trees

Tree−Bisection + Reconnection (TBR)
O(n^3) TBR trees

Notes

Search Strategy of IQPNNI

Concept: BioNJ tree + randomizations + fastNNIs

1. Start with (fast) BioNJ tree.
2. Do fastNNIs to optimize trees, i.e., evaluate all NNIs simultaneously and then accept all best ones which are non-conflicting. (during first round, almost identical to original PhyML).
3. Remove randomly a certain amount of taxa and re-insert them by a fast and rough quartet-based method. (some plausible 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 long enough,
numerically optimized ML computation,
offers codon models for reconstructing trees.

Notes

Search Strategy of PhyML 3.0

Concept: BioNJ tree + pre-screened SPR-neighborhood + fastNNIs

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

Pro: compared to the original PhyML, less prone to get stuck on local optima by using the the SPR-neighborhood now,
applies aLRT to check for branch support.

More: More information by Olivier Gascuel.

Notes

(Additional) ML-based programs:

- IQPNNI
- PhyML
- RAxML
- GARLI
- TREE-PUZZLE
- dnaml, proml (PHYLIP)
- fastDNAml
- MetaPiga
- SSA
- nucml, protml (MOLPHY)
- `http://evolution.genetics.washington.edu/phylip/software.html`

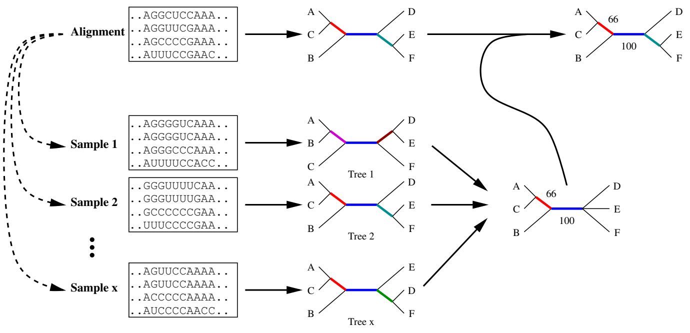
Notes

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?

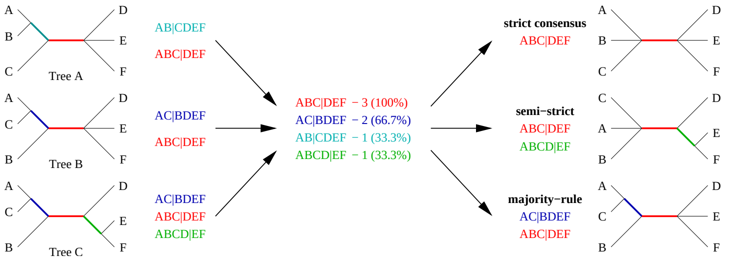
Notes

Assessing Confidence: The Bootstrap



Notes

Summarizing Trees: Consensus Methods



AB|CDEF
ABC|DEF

AC|BDEF
ABC|DEF

AC|BDEF
ABC|DEF
ABCD|EF

ABC|DEF − 3 (100%)
AC|BDEF − 2 (66.7%)
AB|CDEF − 1 (33.3%)
ABCD|EF − 1 (33.3%)

strict consensus
ABC|DEF

semi−strict
ABC|DEF
ABCD|EF

majority−rule
AC|BDEF
ABC|DEF

Notes

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 Majority 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, places uncertain taxa at the root of subtrees (hard to interpret)

Notes

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.

Notes

Methods to Assess 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).
  - Bootstrapping alignment columns + consensus.
  - Jackknifing alignment columns + consensus.
  - Trees from Bayesian MCMC sampling + consensus.
  - Evaluating the likelihoods of the different resolutions at each branch with aLRT (approximate LRT)
    More information by Olivier Gscuel!

Notes

Overview over Likelihood-based Analyses

- Comparing hypothesis with Likelihood-Ratio-Test (=LRT)
  - different models of evolution (ProtTest, MrModeltest, ModelTest - uses also other comparisons)
  - testing molecular clock assumption and root position (TREE-PUZZLE)
  - Checking alternative branching patterns with aLRT
- Determining model parameters (almost all methods estimation parameters)
- Testing for phylogenetic content (TREE-PUZZLE)
- Comparing/testing likelihoods of different tree topologies with Kishino-Hasegawa test, Shimodaira-Hasegawa test (TREE-PUZZLE), SOWH-test, AU test, ELW, ...
- Constructing confidence sets on posterior likelihoods (MrBayes)

Notes

Exercises:

the exercises can be found at

`http://www.cibiv.at/~hschmidt/VEME/ML`

Notes