Maximum Likelihood trees and Likelihood-based Tree Topology Testing

Heiko A. Schmidt

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

September 2008

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

What is the Maximum Likelihood (ML) Approach?

Having the probabilistic process of evolution and its parameters, we could compute the probability of any outcoming sequence data.

$$\underset{\textbf{Probability}}{p(\text{Data} \mid \text{Parameter set } \theta)} \equiv \underset{\textbf{Likelihood}}{L(\text{Parameter set } \theta \mid \text{Data})}$$

But here we are interested in the process and parameters themselves.

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

Aim: The ML approach seaches for that parameter set θ for the process (i.e., evolution) which maximizes the probability of our given dataset.

Notes

Problem: parameter sets

Problem: In phylogenetic analysis, the parameter set θ comprises:

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

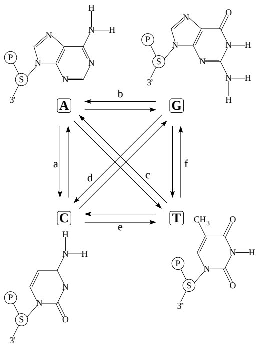
This makes ML a high dimensional optimization problem that usually cannot be solved in one go.

Hence, some parameters like the substitution model and the model parameters are often determined/set separately from the tree.

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:



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

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

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

Notes

Evolutionary Models

Besides a number of DNA substitution models like

- JC69, K2P, F81, HKY85, TN93, or GTR

there are a number of (specialized) protein models:

- 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

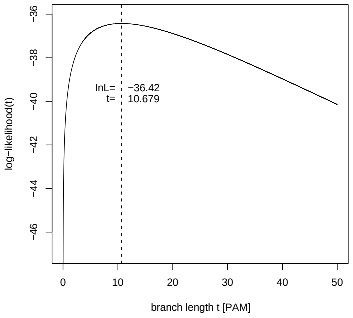
Computing ML Distances Using $\mathbf{P_{ij}(t)}$

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
GGTCCTGACAGAAATAAAC
```

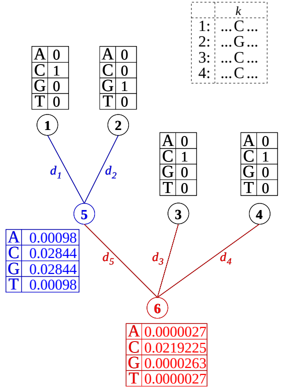


Note: we do not compute the probability of the distance t but that of the data $D = \{s, s'\}$.

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:

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

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

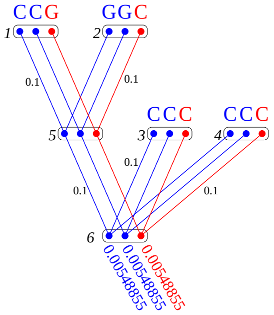
Site-Likelihood of an alignment column k:

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

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

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

$$= 0.0000001653381$$

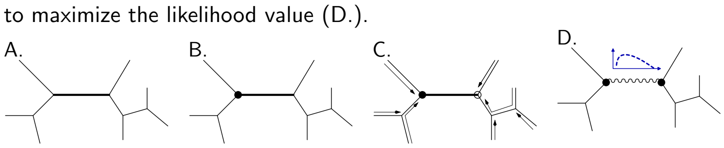
or the log-likelihood

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

Notes

Adjusting Branch Lengths Step-By-Step

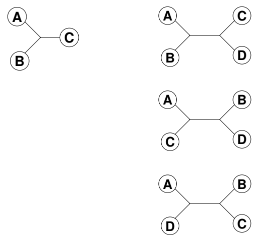
To compute optimal branch lengths do the following. Initialize the branch lengths.
Choose a branch (A.). Move the virtual root to an adjacent node (B.). Compute all partial likelihoods recursively (C.). Adjust the branch length to maximize the likelihood value (D.).

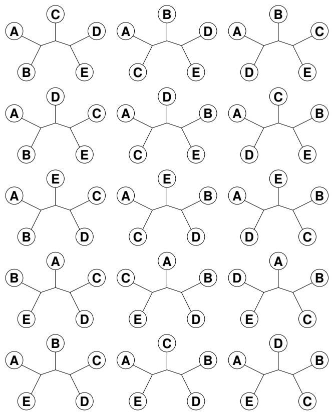


Repeat this for every branch until no better likelihood is gained. This is than the ML value of this tree given the Model and Data.

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

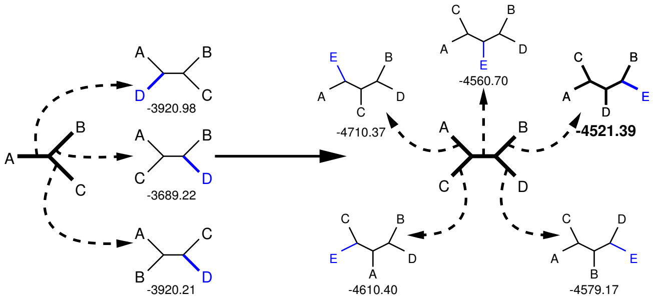
Exhaustive Search: guarantees to find the optimal tree, because all trees are evaluated, but not feasible for more than 10-12 taxa.

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

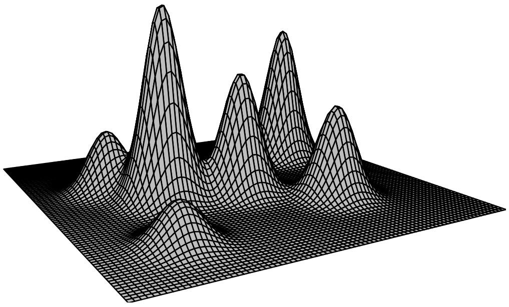
Build up a tree: Stepwise Insertion



Notes

Local Maxima

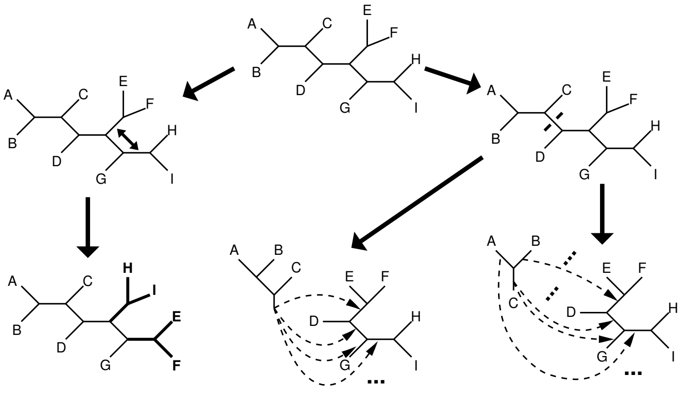
What if we have multiple maxima in the likelihood surface?



Tree rearrangements to escape local maxima.

Notes

Tree Rearrangements: Scanning a Tree's Neighborhood



Nearest Neighbor Interchange

Possible NNI trees = O(n)

subtree pruning + regrafting

Possible SPR trees = O(n*n)

tree-bisection + reconnection

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

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.

Notes

ML programs: Other strategies

Classical Strategies:

- IQPNNI
- PhyML
- RAxML
- dnaml, proml (PHYLIP)
- fastDNAml
- nucml, protml (MOLPHY)

Other Strategies:

- Quartet-based trees (TREE-PUZZLE, Qstar)
- Genetic Algorithms (GARLI, GAML, MetaPIGA)
- Simulated Annealing (SSA, RAxML-SA)
- ...

Note: The last two are also based on NNI/SPR/TBR.
For more programs see:
`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. Questions arise:
- How can we assess reliability for the subtree?
- Is one likelihood value significantly better/worse/different than another?

Notes

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 (e.g., TREE-PUZZLE).
  - Jackknifing alignment columns + consensus.
  - Bootstrapping alignment columns + consensus.
  - Trees from Bayesian MCMC sampling + consensus.
  - approximate LRT (aLRT) on the different branching patterns of four subtrees.

Notes

Are two evolutionary trees/models different?

Given sequence alignments and substitution models, we can reconstruct tree and compute their likelihoods.

But can we decide from the likelihood

- which is the best substitution model to use?

→ yes, using (hierarchical) LRTs and other model selection criteria

- which tree is better (in terms of their likelihoods)?

→ only if likelihoods are computed from identical datasets/taxa.

- whether two tree likelihoods are significantly different?

→ yes, for identical datasets/taxa

- whether one tree likelihood is significantly better/worse?

→ yes, for identical datasets/taxa

These questions can be assesed by hypothesis testing.

Notes

Hypothesis testing: prerequisistes

- What question do I want to answer?
  - Say should I use the JC model or the GTR model?
  - Or perhaps, Is tree T_a statistically better than tree T_b?
- It is important to note that you should know the Null hypothesis/hypotheses **before** you “collect” the data.

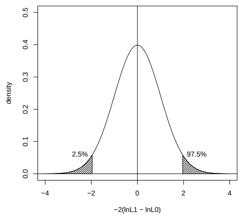
Notes

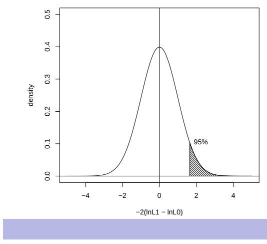
Testing Tree Topologies with LRT?

- **Only nested models** can be tested by ordinary LRT: 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.
- two differrent topologies are not nested.
- Thus, LRT cannot be used on different topologies, because the assumption of the χ^2 distribution does not fit.
- Hence, other (bootstrap-bases) methods have been devised to determine the distribution of log-likelihood differences for testing (e.g., KH or SH test).

Notes

Usual Null-Hypotheses:





First the **Null hypothesis** (H_0)has to be stated, for example:

- **top:** The two likelihood are not significantly different – i.e. their expected difference $E(\ln L_1 - \ln L_0) = 0$.
- **bottom:** The 2nd likelihood is not significantly worse – i.e. their expected difference $E(\ln L_1 - \ln L_0) \leq 0$.

If the observed value falls into the white area, the Null hypothesis cannot be rejected. If it falls into the grey area, this is interpreted as support for the alternative by rejecting the Null hypothesis.

Notes

Basic Idea:

1. Compute log-likelihood values $L_1, \ldots, L_N$ for your trees $T_1, \ldots, T_N$.
2. Draw bootstrap samples i from the alignment, re-estimate the log-likelihood values $L_x^{(i)}$ for each tree T_x and for each sample i.
3. Adjust the log-likelihoods with the mean by setting $\tilde{L}_x^{(i)} = L_x^{(i)} - \bar{L}_x^{(i)}$ (Centering) Centering is needed to correct make the bootstrap samples conforming to the Null model.
4. Use the differences between the $\tilde{L}_x^{(i)}$ to determine the distribution of differences $\delta^{(i)} = \tilde{L}_y^{(i)} - \tilde{L}_z^{(i)}$.
5. Use the distribution of $\delta^{(i)}$ to test your trees.

Notes

Time Saving: Resampling of Estimated Log-Likelihoods (RELL)

- The re-optimization to get the log-likelihood values $L_x^{(i)}$ is very time consuming.
- Hence, often the site-likelihoods are fixed.
- Instead of alignment columns, the bootstrap samples from the already estimated site-log-likelihoods, which are then added bootstrapped likelihood $L_x^{(i)}$.
- The resampling of estimated log-likelihoods (RELL) has been shown to be often sufficient to produce the distribution of log-likelihood differences.

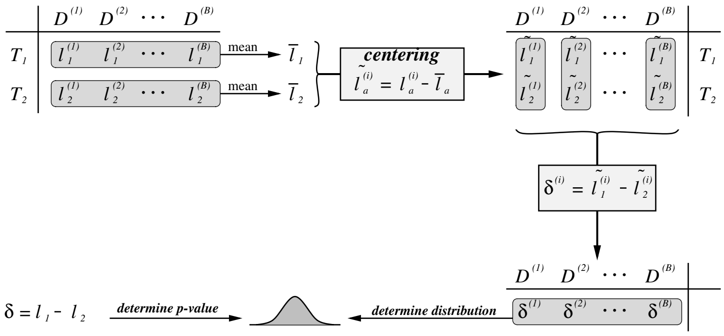
Notes

Original Kishino and Hasegawa test (KH test)

- This test was devised to test whether two *a priori* chosen trees (e.g., from a Markov Chain) are equally well supported by the dataset.
- H_0: the expected $\delta = L_1 - L_2 = 0$.
  H_A: the expected $\delta = L_1 - L_2 \neq 0$.
  KH assumes that the ML tree is not among the trees.

Notes

Kishino-Hasegawa test:



$D^{(1)}$ $D^{(2)}$ ⋯ $D^{(B)}$
T_1
T_2
$l_1^{(1)}$ $l_1^{(2)}$ ⋯ $l_1^{(B)}$
$l_2^{(1)}$ $l_2^{(2)}$ ⋯ $l_2^{(B)}$
mean
mean
$\bar{l}_1$
$\bar{l}_2$
centering
$\tilde{l}_a^{(i)} = l_a^{(i)} - \bar{l}_a$
$\tilde{l}_1^{(1)}$ $\tilde{l}_1^{(2)}$ ⋯ $\tilde{l}_1^{(B)}$
$\tilde{l}_2^{(1)}$ $\tilde{l}_2^{(2)}$ ⋯ $\tilde{l}_2^{(B)}$
$\delta^{(i)} = \tilde{l}_1^{(i)} - \tilde{l}_2^{(i)}$
$\delta^{(1)}$ $\delta^{(2)}$ ⋯ $\delta^{(B)}$
determine distribution
determine p-value
$\delta = l_1 - l_2$

Notes

Mis-use of the Kishino and Hasegawa test (KH test)

1. Often, instead two *a priory* chosen trees, one tree is tested against the ML tree T_{ML}.
2. That means, $\delta = L_{ML} - L_1$ can rarely be negative.
3. Hence, δ has to be tested in a single-sided regime.
4. H_0: the expected $\delta = L_1 - L_2 = 0$. H_A: the expected $\delta = L_1 - L_2 > 0$.

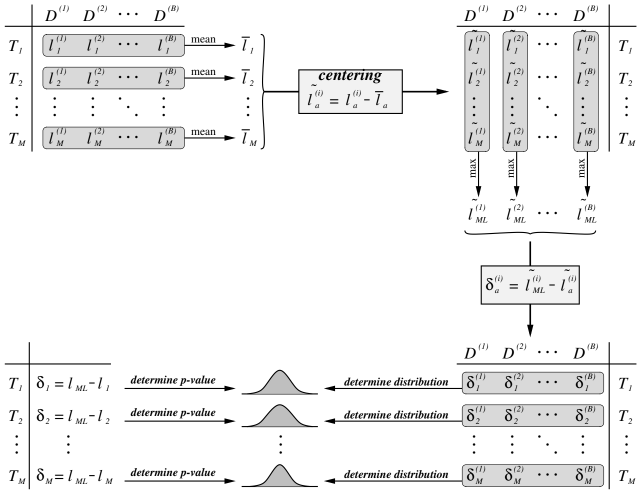
Notes

Multiple trees (Shimodaira and Hasegawa test - SH test)

- The SH test offers a correct way to test a set of trees, which may be chosen *a posteriory* after ML analysis.
- H_0: All trees including T_{ML} are equally supported. H_A: Some or all trees T_x are not equally well supported.
- The SH test assumes, that the ML tree T_{ML} is among the trees.

Notes

Shimodaira-Hasegawa test:



$D^{(1)}$ $D^{(2)}$ $\cdots$ $D^{(B)}$
T_1 $l_1^{(1)}$ $l_1^{(2)}$ $\cdots$ $l_1^{(B)}$ mean $\bar{l}_1$
T_2 $l_2^{(1)}$ $l_2^{(2)}$ $\cdots$ $l_2^{(B)}$ mean $\bar{l}_2$
T_M $l_M^{(1)}$ $l_M^{(2)}$ $\cdots$ $l_M^{(B)}$ mean $\bar{l}_M$
centering
$\tilde{l}_a^{(i)} = l_a^{(i)} - \bar{l}_a$
$\tilde{l}_1^{(1)}$ $\tilde{l}_2^{(1)}$ $\tilde{l}_M^{(1)}$ $\tilde{l}_1^{(2)}$ $\tilde{l}_M^{(B)}$
max
$\tilde{l}_{ML}^{(1)}$ $\tilde{l}_{ML}^{(2)}$ $\cdots$ $\tilde{l}_{ML}^{(B)}$
$\delta_a^{(i)} = \tilde{l}_{ML}^{(i)} - \tilde{l}_a^{(i)}$
$\delta_1 = l_{ML} - l_1$ determine p-value determine distribution $\delta_1^{(1)}$ $\delta_1^{(2)}$ $\cdots$ $\delta_1^{(B)}$
$\delta_2 = l_{ML} - l_2$ determine p-value determine distribution $\delta_2^{(1)}$ $\delta_2^{(2)}$ $\cdots$ $\delta_2^{(B)}$
$\delta_M = l_{ML} - l_M$ determine p-value determine distribution $\delta_M^{(1)}$ $\delta_M^{(2)}$ $\cdots$ $\delta_M^{(B)}$

Notes

The SH procedure:

1. Compute log-likelihood values and the differences $\delta_x = L_{ML} - L_x$ for your trees.
2. Draw bootstrap samples i from the alignment (with RELL) to gain log-likelihood values $L_x^{(i)}$ for each tree T_x.
3. Adjust the log-likelihoods with the mean over the samples i by setting $\tilde{L}_x^{(i)} = L_x^{(i)} - \bar{L}_x^{(i)}$ (Centering)
4. For each sample i, find $\tilde{L}_{ML}^{(i)}$ over all topologies T_x.
5. and compute $\delta_x^{(i)} = \tilde{L}_{ML}^{(i)} - \tilde{L}_x^{(i)}$.
6. For each tree T_x, test whether δ_x is a plausible sample from the distribution of $\delta_x^{(i)}$ (over all replicates i).
7. We use a single sided test, since $\tilde{L}_{ML}^{(i)} \geq \tilde{L}_x^{(i)}$.

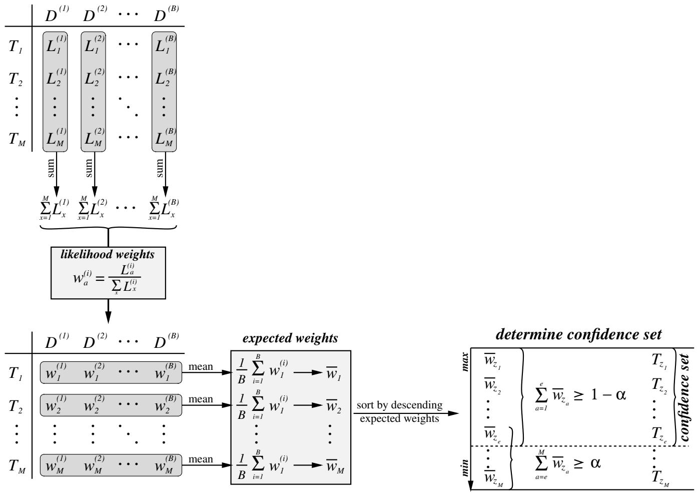
Notes

SH and the Approximately Unbiased Test (AU)

- The problem of the SH test is that it is very conservative (due to a selection bias)
- that means the more trees we add to the test the more trees will not be rejected.
- Shimodaira has suggested an approximately unbiased (AU) test which determine this bias by drawing bootstrap samples of different sizes (i.e. more and also less numbers of columns) to overcome that bias.
- Although the AU test has not the problem of conservativeness, it might have problems of artificial over-confidence if many best trees are equally good.

Notes

Confidence Sets from Expected Likelihood weights:


$D^{(1)}$ $D^{(2)}$ ⋯ $D^{(B)}$
T_1 T_2 ⋮ T_M
$L_1^{(1)}$ $L_2^{(1)}$ $L_M^{(1)}$
sum
$\sum_{x=1}^{M} L_x^{(1)}$ $\sum_{x=1}^{M} L_x^{(2)}$ ⋯ $\sum_{x=1}^{M} L_x^{(B)}$
likelihood weights
$w_a^{(i)} = \frac{L_a^{(i)}}{\sum L_x^{(i)}}$
$w_1^{(1)}$ $w_1^{(2)}$ ⋯ $w_1^{(B)}$
mean
expected weights
$\frac{1}{B}\sum_{i=1}^{B} w_1^{(i)} \rightarrow \overline{w}_1$
sort by descending expected weights
determine confidence set
max
min
$\sum_{a=1}^{c} \overline{w}_{z_a} \geq 1-\alpha$
$\sum_{a=c}^{M} \overline{w}_{z_a} \geq \alpha$
T_{z_1} T_{z_2} T_{z_c} T_{z_M}
confidence set

Notes

Pros and Cons of various tests:

- **Kishino-Hasegawa test (KH)** – usually mis-used if the trees are not chosen *a priori*.
- **Shimodaira-Hasegawa (SH)** – test for multiple trees, affected by selection error, i.e., gets more conservative with the number of trees.
- **Weighted Shimodaira-Hasegawa (wSH)** – SH test weighted with the variance of the likelihood difference (less conservative than SH).
- **Expected likelihood weights (ELW)** – less conservative than SH, but the impact of model mis-specification unclear.
- **Approximately unbiased test (AU)** – fixes the conservativeness issue of SH, but many similarly good trees can lead to artificial over-confidence.

Notes

Summary

- Likelihoods gives a strong statistical framework for hypothesis testing.
- Proper experimental design and proper use of tests is required.
- One should always be aware of the hypotheses a test assesses and should make sure that this answers the question asked.
- Testing tree topologies can be used to assess whether two competing hypotheses are really substantially different. If they are not, one cannot be prefered over the other.
- Unfortunately, there is not the final test (yet), so one should apply several of them and be aware of their hypotheses, assumptions, and pitfalls.

Notes

Exercises:

the exercises can be found at

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

Notes

Notes

Notes