12

Testing tree topologies

THEORY

Heiko A. Schmidt

12.1 Introduction

Throughout the book a number of approaches have been exemplified to assess and compare various aspects of evolutionary trees and models.

To check the reliability of branches in a certain tree, one can use ***(non-parametric) bootstrapping*** or ***jackknifing***, combining alignment subsampling and consensus trees to get support values on branches (Chapter 5). Furthermore, other methods that generate or sample sets of plausible trees can be used to get support values, like Bayesian *MCMC* sampling (Chapter 7) or ***quartet puzzling*** (Chapter 6).

Various approaches have been devised to determine a best-suited ***evolutionary model*** (Chapter 10). Such approaches are often based on the ***maximum likelihood*** values obtained for the models in question. Different measures are applied like ***Akaike Information Criterion (AIC)***, ***Bayesian Infomation Criterion*** (***BIC***), *Akaike Weights*, and other model selection techniques (refer to Johnson & Omland, 2004, for review) to correct for the additional parameters in the more complex models. Such techniques are also implemented in programs like ModelTest to select the most useful model of evolution (see Posada & Buckley, 2004 and Chapter 10 for details).

In this chapter we will briefly review different techniques and tests to compare contradicting and, hence, non-nested tree topologies using their ***likelihood*** values. Since a large variation of testing approaches can be applied (see, e.g. Goldman

The Phylogenetic Handbook: a Practical Approach to Phylogenetic Analysis and Hypothesis Testing, Philippe Lemey, Marco Salemi, and Anne-Mieke Vandamme (eds.). Published by Cambridge University Press. © Cambridge University Press 2009.

et al., 2000), we will restrict ourselves to review a number of common tests for which easily accessible software implementations exist. We will briefly describe the different approaches, the hypotheses they test, and discuss possible problems and pitfalls.

12.2 Some definitions for distributions and testing

In the current context we are usually interested in whether the difference between two values, e.g. the likelihoods of two models or trees, are significantly different or could be explained by random effects.

To perform a test, we first have to state a *null hypothesis* H_0. The null hypothesis is the hypothesis of *no difference* and is usually the negation of the question we are interested in (Siegel & Castellan, 1988, p. 7). This null hypothesis has to be precise, since the test of significance is based on its rejection (Fisher, 1971). If the tested null hypothesis is rejected, the *alternative hypothesis* H_A is supported which typically reflects our question, like "are two likelihoods significantly different?"

There are two types of possible errors when testing the null hypothesis H_0. First, the null hypothesis is rejected when it is actually true (type I error). This result is also called a *false positive.* Second, an erroneous null hypothesis is failed to be rejected (type II error), also called *false negative.* The probability of a type I error is denoted by α. We set α to the largest probability of a type I error we are willing to accept; the *significance level,* typically $\alpha = 0.05$ (i.e. 5% error). This corresponds to a *confidence limit* of $(1 - \alpha) = 0.95$ or 95% (e.g. Siegel & Castellan, 1988; Zarkikh & Li, 1995).

Given a ***sampling distribution*** that reflects the probability of every possible sample value if drawn randomly, the null hypothesis (of no difference from expectation) can be tested. If the observed value μ_0 is in the region of rejection, i.e. outside the 95% *confidence interval* or *acceptance region,* the null hypothesis H_0 is rejected and the alternative hypothesis is supported. If that value falls inside the acceptance region, H_0 cannot be rejected at the chosen level of confidence (see Fig. 12.1a).

If we have prior knowledge about the direction of the effect of the alternative hypothesis, then a one-sided test is used. Note, that one-sided and two-sided tests do not differ in the size but in the location of the rejection region, i.e. in the one-sided test the region is entirely at one tail of the sample distribution (see Fig. 12.1b).

The significance level α has to be set in advance and determines the critical value below or above which the null hypothesis is rejected. The p-value, on the other hand, denotes the probability of obtaining a result equal or more extreme (with respect to the null hypothesis) than the observed value μ_0 and can only be

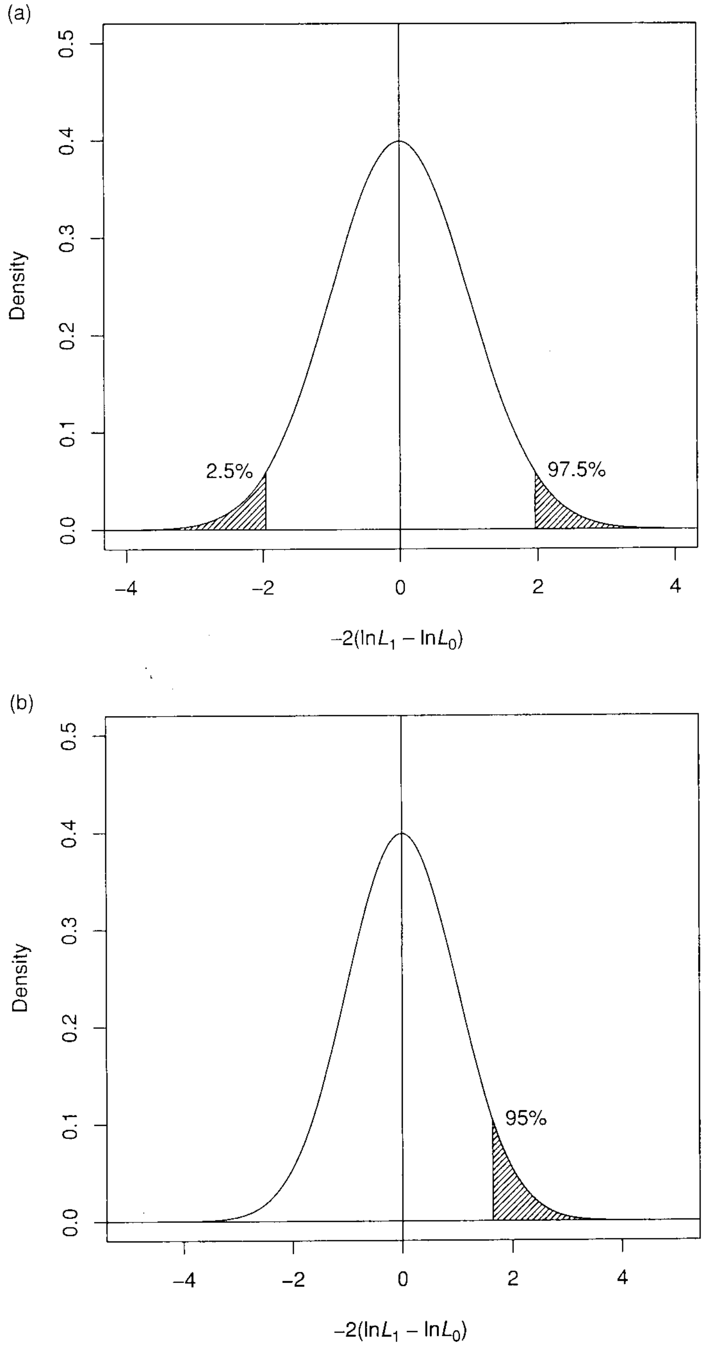

Fig. 12.1 Normal distributions for (a) a two-sided and (b) a one-sided test. An observed value μ_0 is (a) significantly different from the expectation μ if it is in one of the two shaded tails covering each 2.5% of the surface below the curve on either side or (b) significantly larger if it is above the upper 95% quantile. The null hypothesis cannot be rejected if μ_0 falls into the (unshaded) 95% confidence interval. These decisions depend on the significance level $\alpha = 0.05$.

determined after the test (Goodman, 1999). If the null hypothesis is rejected, the p-value is also necessarily less than α.

When testing tree topologies, there is a big difference whether the trees are selected *a priori* or *a posteriori*.

A priori means that the trees have been selected without knowledge about their support by the data or any optimizing analysis involved. Such trees might just be derived as logical alternative scenarios or, for example, from a ***Markov chain*** without prior knowledge about their likelihood values or probabilities. Hence, each of the trees of interest might be the one with the highest likelihood.

If the trees of interest are selected from an analysis to test whether, for example, the second, third, etc. tree is significantly worse than the best tree, that is the tree yielding the best maximum likelihood value called the *ML* tree (cf. Chapter 6), the trees are chosen *a posteriori*.

12.3 Likelihood ratio tests for nested models

If the two evolutionary models of interest are *nested*, meaning that the more parameter-rich model can be restricted to the simpler one by restricting its parameters, then ***likelihood ratio tests*** are straightforward to compare the likelihoods L_0 and L_1 of the two models based on a (single) tree. For convenience l_a denotes the log-likelihood $\ln L_a$ in the following. The ***likelihood ratio*** test (LRT) ***statistic***

$$\Delta = -2\ln\frac{L_0}{L_1} = 2(l_1 - l_0) \qquad (12.1)$$

follows approximately a χ^2 distribution for the respective degrees of freedom, that is, the number of additional parameters in the more parameter-rich model. L_1 is the likelihood of the alternative (more parameter-rich) model and L_0 that of the less parameter-rich null model. If their Δ value computed from (12.1) is located in the rejection area of the χ^2 distribution (the shaded area in Fig. 12.2) beyond the 95%-quantile (if a significance level of 5% is assumed), the null hypothesis is rejected and the alternative model is said to give a significantly higher likelihood L_1 compared to the null model. (Likelihood ratio tests of nested models are discussed in detail in Chapter 10.)

Although this methodology is straightforward for nested models, it is generally not applicable to compare different tree topologies. The problem with trees is that tree topologies cannot be interpreted as a single statistical parameter and, furthermore, it remains unclear how many parameters a tree represents with its possible groupings and branch lengths (Yang *et al.*, 1995; Huelsenbeck & Crandall, 1997).

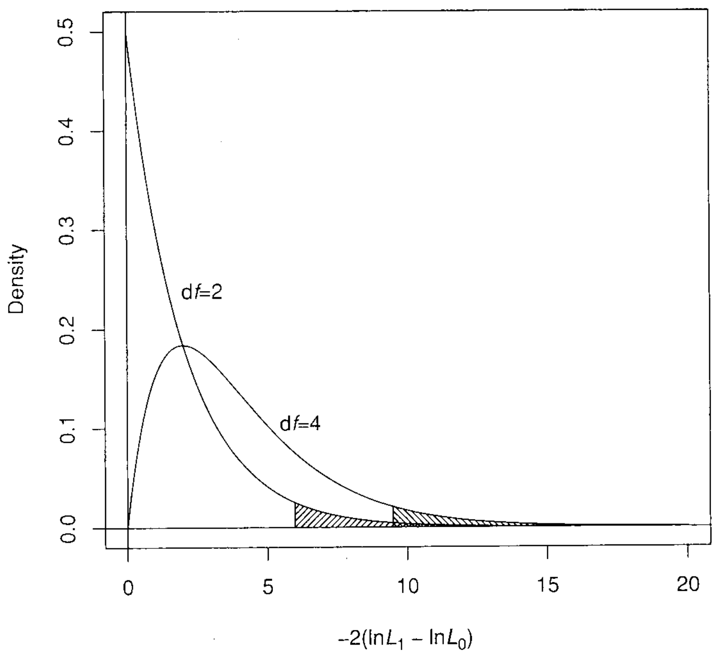

Fig. 12.2 χ^2 distribution for 2 and 4 degrees of freedom. An LRT is assumed significant, i.e. the more parameter-rich has a significantly higher likelihood L_1, if the Δ value is in the shaded 95% quantile.

If the tested models are not nested, the distribution of Δ is *Gaussian*, that means, according to a *normal distribution* (Cox, 1961, 1962). The shape of a normal distribution $\mathcal{N}(\mu, \sigma^2)$ is determined by its mean value μ and standard deviation σ.

Thus, the χ^2 distribution does not apply, and different steps must be taken to find the distribution that can be used to test the difference between two likelihoods.

12.4 How to get the distribution of likelihood ratios

Most of the methods we will be concerned with, will use a likelihood ratio statistic

$$\delta = \ln \frac{L_a}{L_b} = \ln L_a / \ln L_b = l_a - l_b \qquad (12.2)$$

to compare the difference of the log-likelihoods l_a and l_b for two trees T_a and T_b. These likelihoods are obtained by maximum likelihood optimization of model parameters, branch lengths, etc. on a given sequence data set D, as described in Section 6.3 (Chapter 6).

To judge whether the obtained likelihoods are significantly different, we need information on how the "real" distribution of likelihood differences under the null hypothesis looks like.

In the ideal case one would like to draw further samples from the process that generated our data. Unfortunately, we cannot re-run the process of evolution, as we would be able to by tossing a coin or roll dice a couple of additional times. Usually, we only have a limited data set, the alignment, where each column is usually regarded as an independent sample from the "true" process of evolution, to determine the desired distribution.

A common way to determine such distributions from limited data sets if further samples from the original process cannot be obtained are bootstrap re-sampling methods (Efron, 1979; Efron & Tibshirani, 1994; Goldman, 1993).

12.4.1 Non-parametric bootstrap

The ***non-parametric bootstrap*** has been mentioned in various chapters of this book. This bootstrap randomly re-samples columns from the alignment D with replacement to produce a number of pseudo-samples $D^{(i)}$ from the processes of evolution. In each of these pseudo-samples some columns might be included several times, while others have not been chosen at all (Felsenstein, 1985; Efron *et al.*, 1996). Each generated pseudo-sample $D^{(i)}$ is then used to compute the values of interest and to determine their distribution.

Here, based on each pseudo-sample alignment $D^{(i)}$ the maximum log-likelihood values $l_x^{(i)}$ of each tree T_x in the set of M trees $\mathcal{T}$ of interest are computed by complete optimization of branch lengths and model parameters.

For the use of bootstrap in a hypothesis testing scenario it is necessary for the values computed via the bootstrap to reflect the assumed null-distribution, although the pseudo-sample data might not. Several steps available to ensure this null-hypothesis conformity have been described (Hall & Wilson, 1991; Westfall & Young, 1993). The method of choice to adjust the log-likelihood values is the so-called *centering*, where each log-likelihood value $l_x^{(i)}$ of each tree T_x on pseudo-sample $D^{(i)}$ is shifted by the mean value $\bar{l}_x = \frac{1}{B}\sum_{i=1}^{B} l_x^{(i)}$ leading to a centered log-likelihood $\tilde{l}_x^{(i)} = l_x^{(i)} - \bar{l}_x$. From the centered log-likelihoods $\tilde{l}_x^{(i)}$ the log-likelihood differences $\delta^{(i)}$ are computed between pairs of trees according to (12.2) for each sample $D^{(i)}$. The obtained values $\delta^{(i)}$ are then used to infer the mean μ and the standard deviation σ of the respective normal (sample) distribution to test the observed ratio δ.

The re-optimization of the likelihood values for all the bootstrap samples is computationally very intense. Hence, Kishino *et al.* (1990) suggested ***resampling***

estimated log-likelihoods (***RELL***), a variant of the non-parametric bootstrap, that is computationally less demanding but not necessarily as accurate. We have seen in Chapter 6 that the likelihood values are computed by multiplying the *site-likelihoods* of each column D_j (6.12) or the log-likelihood as the sum of all site-log-likelihoods (6.18). Kishino *et al.* (1990) keep the site-log-likelihoods fixed and only "bootstrap" the pre-estimated site-log-likelihoods. This RELL method saves the time-consuming likelihood re-estimation, but it assumes some asymptotic conditions such as sufficiently large data and correctly specified models of evolution to produce valid results.

12.4.2 Parametric bootstrap

A different way to infer the distribution of δ is the ***parametric bootstrap*** (also called ***Monte Carlo simulation***). Here, the bootstrap samples are not drawn from the alignment but simulated along a tree with branch lengths and model parameters. That means, a tree with branch lengths and model parameters has to be inferred first from the original alignment D which then serve as input for Monte-Carlo simulation performed by sequence generation programs such as SEQ-GEN (Rambaut & Grassly, 1997). From the simulated bootstrap samples, one again estimates trees and their likelihoods which are, in turn, used to determine the distribution of δ.

Differently to non-parametric bootstrapping no adjustment step like centering (Section 12.4.1) is necessary, since the given tree, model, and parameters act as null-model according to which the bootstrap samples are generated by Monte-Carlo simulation.

For detailed descriptions of parametric bootstrap approaches refer to Goldman (1993) and Huelsenbeck and Crandall (1997).

12.5 Testing tree topologies

A large number of test variants exist by combining different approaches in the various steps of a test, like different bootstrap methods to generate the samples, the amount of optimization to compute the likelihoods, the choice of the trees of interest, or the assumptions made on the kind of normal distribution. To get an extensive overview on such variants, discussions about the possible ways, problems, and pitfalls of tree topology testing, we recommend Goldman *et al.* (2000) and Huelsenbeck and Crandall (1997) and references therein.

We only review a limited number of tests which are commonly used. To that end, we will mostly use the same notation as Goldman *et al.* (2000).

12.5.1 Tree tests – a general structure

First, the null hypothesis H_0 and the alternative hypothesis H_A have to be stated, since they determine the results of the test and also determine whether a test is applicable at all for the available data and the question a researcher is asking.

Second, testing trees with likelihoods follows a global structure:

(i) Compute the log-likelihood values l_x for all trees $T_x \in \mathcal{T}$ by fully optimizing all parameters. Also, all site-likelihoods are kept for bootstrapping.

(ii) Generate many ($B \geq 1000$) bootstrap samples $D^{(i)}$ ($i = 1 \ldots B$). Re-estimate the log-likelihood values $l_x^{(i)}$ (with optimization) for each tree T_x and each bootstrap sample $D^{(i)}$.

(iii) Adjust for each tree topology T_x all log-likelihoods $l_x^{(i)}$ to conform to the null hypothesis, if the bootstrap samples have been generated by non-parametric bootstrap or RELL. This is typically done by *centering* the log-likelihoods with the mean log-likelihood $\bar{l}_x^{(i)} = \frac{1}{B}\sum_{i=1}^{B} l_x^{(i)}$ across all bootstrap samples i:

$$\tilde{l}_x^{(i)} = l_x^{(i)} - \bar{l}_x^{(i)} \tag{12.3}$$

Refer to Hall and Wilson (1991) for more details on the necessity of centering.

(iv) Compute the log-likelihood differences $\delta^{(i)} = \tilde{l}_a^{(i)} - \tilde{l}_b^{(i)}$ between the relevant pair(s) of trees T_a and T_b. Use the $\delta^{(i)}$ values to determine their distribution.

Note, that the number and specification of the relevant pairs of trees depend on the respective null hypothesis H_0 (see following sections).

(v) Use the distribution of $\delta^{(i)}$ to test whether the null hypothesis is to be rejected. Obtain the p-value for the observed δ.

12.5.2 The original Kishino–Hasegawa (KH) test

Kishino and Hasegawa (1989) devised a test based on the RELL method to compare two *a priori* selected trees T_a and T_b, e.g. produced by a Markov chain.

The null and alternative hypotheses to be compared are (two-sided test):

H_0: The two trees are equally supported, i.e. the expected value $E[\delta] = \mu = 0$
H_A: The two trees are not supported equally, i.e. the expected value $E[\delta] = \mu \neq 0$

The KH test itself follows the following procedure (see also Fig. 12.3):

(i) Infer the log-likelihood values l_a and l_b for trees T_a and T_b. Compute $\delta = l_a^{(i)} - l_b^{(i)}$.

(ii) Generate many ($B \geq 1000$) bootstrap samples i and the respective log-likelihood values $l_a^{(i)}$ and $l_b^{(i)}$ with the RELL method.

(iii) Center the obtained likelihood values of each tree with the mean log-likelihood $\bar{l}_x^{(i)}$ across all samples i, as $\tilde{l}_x^{(i)} = l_x^{(i)} - \bar{l}_x^{(i)}$.

(iv) Determine the distribution of differences $\delta^{(i)} = \tilde{l}_a^{(i)} - \tilde{l}_b^{(i)}$.

(v) Use the distribution inferred from $\delta^{(i)}$ to test whether your trees are equally supported in a two-sided test. Obtain the p-value for the observed δ.

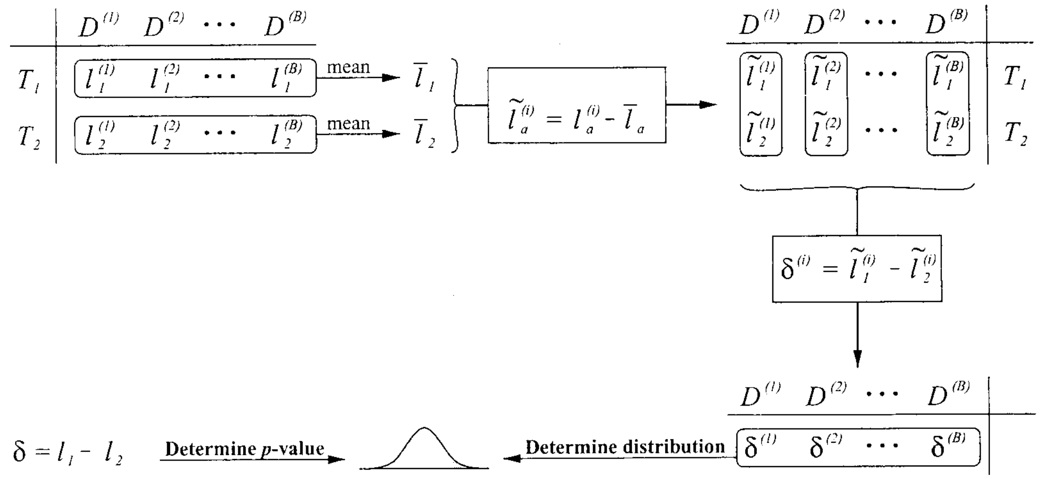
$D^{(1)}$ $D^{(2)}$ ··· $D^{(B)}$
T_1
T_2
$l_1^{(1)}$ $l_1^{(2)}$ ··· $l_1^{(B)}$
$l_2^{(1)}$ $l_2^{(2)}$ ··· $l_2^{(B)}$
mean
mean
$\bar{l}_1$
$\bar{l}_2$
$\tilde{l}_a^{(i)} = l_a^{(i)} - \bar{l}_a$
$\tilde{l}_1^{(1)}$ $\tilde{l}_1^{(2)}$ ··· $\tilde{l}_1^{(B)}$
$\tilde{l}_2^{(1)}$ $\tilde{l}_2^{(2)}$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

Fig. 12.3 Sketch of the Kishino–Hasegawa test. For both trees T_1 and T_2 the log-likelihoods l_a^i are computed for each bootstrap sample $D^{(i)}$. The log-likelihoods are subsequently "centered" by the trees mean log-likelihood $\bar{l}_a$ across the bootstrap samples. The log-likelihood difference $\delta_a^{(i)}$ between $\tilde{l}_1^{(i)}$ and $\tilde{l}_2^{(i)}$ is computed. Finally, the distribution of δ-values is determined and used to determine the p-value of the observed δ-value.

Since the two trees T_a and T_b are selected *a priori* both have an equal chance to gain the higher log-likelihood and, hence, δ might be positive or negative. Thus a two-sided test is applied as in Fig. 12.1a.

12.5.3 One-sided Kishino–Hasegawa test

Although the KH test was devised for *a priori* selected trees, in the majority of its applications it is (mis-)used to test whether sets of suboptimal trees are equally supported or significantly worse than the best tree, or to compare all trees against the one having the highest likelihood among all *a priori* selected trees.

The problem arises that, if T_a is the maximum likelihood tree or the tree with highest likelihood in the set and all trees in $\mathcal{T}$ are tested against this one tree T_{ML}, then $\delta = l_{ML} - l_b$ can hardly be negative. The above hypotheses are thus not tested properly by the original KH approach. However, the KH test has often been applied this way (see Goldman *et al.*, 2000 and Shimodaira & Hasagawa, 1999 for extensive discussion).

The only way to adjust the KH test to some extent to this scenario would be to use a one-sided test as indicated in Fig. 12.1b (cf. Goldman *et al.*, 2000). However, the null hypothesis $E[\delta] = 0$ might still be violated.

Many published conclusions based on wrongly applied KH tests might not be valid. Goldman *et al.* (2000) stated that the only possible adjustment to correct for this mistake might be to adjust the p-value to $p/2$. This has a similar effect as having performed a one-sided test instead.

12.5.4 Shimodaira–Hasegawa (SH) test

Shimodaira and Hasegawa (1999) devised a valid test to assess a set of *a posteriori* selected trees when the maximum likelihood tree is among the tested trees.

The null and alternative hypotheses tested by the Shimodaira–Hasegawa (SH) test look different in this case:

H_0: All trees $T_x \in \mathcal{T}$ (including the ML tree T_{ML}) are equally good explanations of the data.
H_A: Some or all trees $T_x \in \mathcal{T}$ are not equally good explanations of the data.

The test itself follows the following procedure (see also Fig. 12.4):

(i) Estimate the log-likelihood values l_{ML} and l_x for all trees $T_x \in \mathcal{T}$. Compute all $\delta_x = \ell_{ML} - l_x$.
(ii) Generate many ($B \geq 1000$) bootstrap samples i and compute the respective log-likelihood values $l_{ML}^{(i)}$ and $l_x^{(i)}$ (using the RELL method).
(iii) For each tree T_x, center the log-likelihood values with the mean log-likelihood $\bar{l}_x^{(i)}$ across all samples i, as $\tilde{l}_x^{(i)} = l_x^{(i)} - \bar{l}_x^{(i)}$.
(iv) For each bootstrap sample i, find the maximal log-likelihood $\tilde{l}_{ML}^{(i)}$ over all trees $T_x \in \mathcal{T}$ and compute the differences $\delta_x^{(i)} = \tilde{l}_{ML}^{(i)} - \tilde{l}_x^{(i)}$.
(v) For each tree T_x separately test whether the obtained δ_x value is in the rejection area beyond 95%. If so, reject the null hypothesis for T_x. If not, H_0 cannot be rejected. Obtain the p-value for the observed δ_x.

The one-sided test is appropriate here, since the log-likelihood $\tilde{l}_x^{(i)}$ of any tree T_x can only be smaller or equal to $\tilde{l}_{ML}^{(i)}$.

When applying the SH test, one has to keep in mind that the maximum likelihood tree is required to be among the tested trees, otherwise the estimated significance levels will be inaccurate (Goldman *et al.*, 2000 and Westfall & Young, 1993, p. 48).

Furthermore, it has been pointed out by Strimmer and Rambaut (2002) that the number of trees selected in the SH test is strongly correlated with the number of tested input trees, meaning, the more tree topologies are included in the test, the more trees are accepted. This conservative behavior makes the use of the SH test problematic for large sets of trees.

12.5.5 Weighted test variants

Shimodaira and Hasegawa (1999, comment 4) have suggested weighted variants of the SH and also the KH test, namely WSH and WKH, for cases where one wants

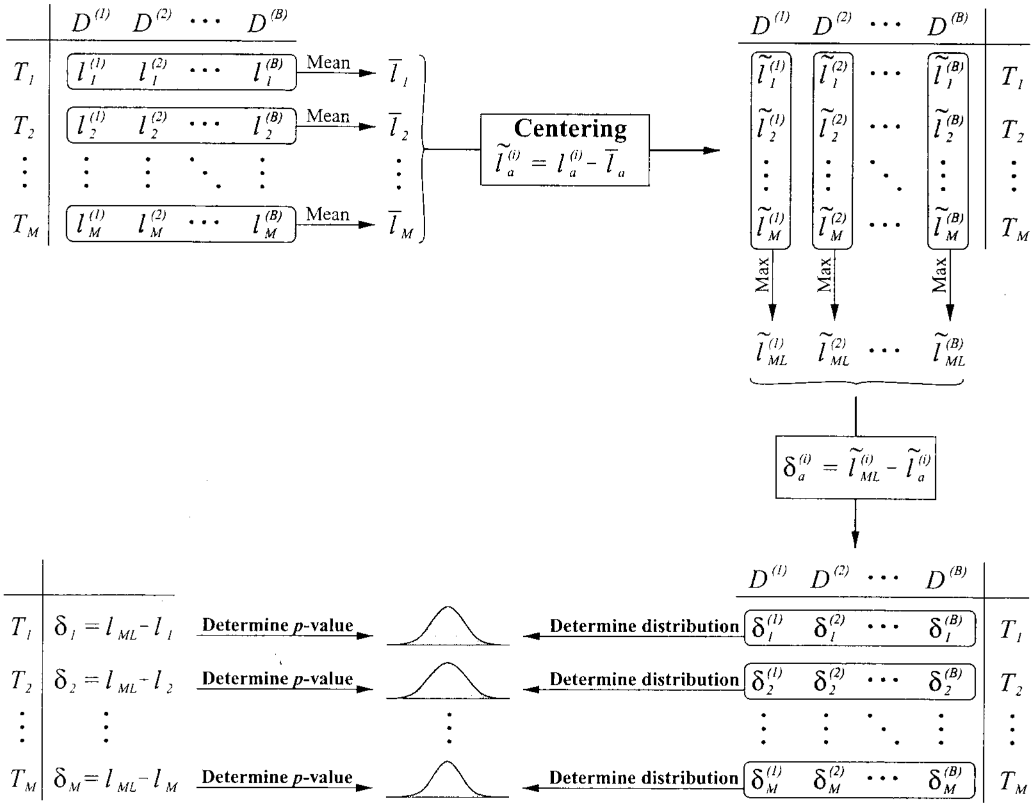
$D^{(1)}$ $D^{(2)}$ ... $D^{(B)}$
T_1
T_2
T_M
Mean
$\bar{l}_1$
$\bar{l}_2$
$\bar{l}_M$
Centering
$\tilde{l}_a^{(i)} = l_a^{(i)} - \bar{l}_a$
Max
$\tilde{l}_{ML}^{(1)}$ $\tilde{l}_{ML}^{(2)}$ ... $\tilde{l}_{ML}^{(B)}$
$\delta_a^{(i)} = \tilde{l}_{ML}^{(i)} - \tilde{l}_a^{(i)}$
$\delta_1 = l_{ML} - l_1$
$\delta_2 = l_{ML} - l_2$
$\delta_M = l_{ML} - l_M$
Determine p-value
Determine distribution

Fig. 12.4 Sketch of the Shimodaira–Hasegawa test. Log-likelihoods l_a^i are computed for each tree T_a and each bootstrap sample $D^{(i)}$ and subsequently "centered" by the trees mean log-likelihood $\bar{l}_a$ across the bootstrap samples. For each bootstrap sample $D^{(i)}$ the tree with maximal log-likelihood $\tilde{l}_{ML}^{(i)}$ is determined. Then log-likelihood difference $\delta_a^{(i)}$ between $\tilde{l}_{ML}^{(i)}$ and the corresponding $\tilde{l}_a^{(i)}$ is computed. Finally, the distribution of δ-values is determined and used to determine the p-value of the corresponding trees' observed δ-values.

to be less conservative. In these variants the likelihood ratio $l_a - l_b$ is weighted by the square root of its of variance $\sigma^2(l_a - l_b)$.

This is straightforward for the KH test in step (i) and (iv). In the SH test, however, all instances of $\delta_x = l_{ML} - l_x$ (step (i)) have to be substituted by

$$\delta_x = \max_{a \neq x} \left(\frac{\tilde{l}_a - \tilde{l}_x}{\sigma\left(\tilde{l}_a - \tilde{l}_x\right)} \right) \tag{12.4}$$

while $\delta_x^{(i)} = \tilde{l}_{ML}^{(i)} - \tilde{l}_x^{(i)}$ (step (iv)) is replaced by

$$\delta_x^{(i)} = \max_{a \neq x} \left(\frac{\tilde{l}_a^{(i)} - \tilde{l}_x^{(i)}}{\sigma\left(\tilde{l}_a^{(i)} - \tilde{l}_x^{(i)}\right)} \right) \tag{12.5}$$

Note, due to the weighting the maximal δ-value is not necessarily gained between the current tree and the ML tree, which is the case in the unweighted test. By weighting the likelihood ratio depending on its variance, the tests are less conservative. Although this compensates for some of the above-mentioned conservative behavior of the SH test, it does not completely correct for it (Shimodaira, 2002). Furthermore, both the WSH and WKH tests rely on the same assumptions like the presence or absence of the ML trees in the set of compared trees as their un-weighted counterparts.

12.5.6 The approximately unbiased (AU) test

Shimodaira (2002) explains the correlation of the number of input trees and the size of the confidence set returned by the SH test by the fact that the SH test is heavily biased. On the one hand, SH is very good in controlling its type I error, but it overestimates the selection bias and, thus, acts more conservative as the number of input trees grows.

Zharkikh and Li (1995) have suggested a method that is based on a complete, as well as a partial, bootstrap to enable the inference of the selection bias. Shimodaira (2002) later devised an **approximately unbiased** (**AU**) test based on a multiscale bootstrap to be able to better correct for the selection bias. The *multiscale bootstrap* works as follows.

From our input alignment D of length N, the multiscale bootstrap draws bootstrap replicates for a number of different lengths N_r. Some are smaller but also some are larger than the original sequence length N. For each length N_r, many bootstrap samples are drawn ($B \geq 10\,000$). The log-likelihood $l_x^{(i)*}$ obtained by the RELL method for the sequence length N_k are scaled with the factor N/N_k to the same virtual length N:

$$l_x^{(i,r)} = \frac{N}{N_k} l_x^{(i,r)*} \tag{12.6}$$

Using the results from the different sequence lengths N_r, the method is able to infer the unknown curvature of the selection bias needed for a proper correction. Thus, the AU test is approximately unbiased if an appropriate set of sequence lengths is used (see Shimodaira, 2002, for details).

According to Shimodaira (2002), the AU test tests for each tree $T_a \in \mathcal{T}$ the following null hypothesis.

$H_0(T_a)$: the expected value $E[l_a]$ of T_a is larger or equal to the expected values $E[l_x]$, for all $T_x \in \mathcal{T}$.

Although the AU test is not susceptible to the increase of trees, one has to be careful if many of the best trees are almost equally well supported – one might miss the true tree, since there is over-confidence in the wrong trees (Shimodaira, 2002). Furthermore, the method might be computationally infeasible if the tree set $\mathcal{T}$ contains several thousand trees. Shimodaira (2002) suggests a prefiltering with the KH test using a very conservative significance value (e.g. $\alpha = 0.001$) to reduce the tree set before applying the AU test.

12.5.7 Swofford–Olsen–Waddell–Hillis (SOWH) test

Swofford *et al.* (1996) suggested an approach (SOWH test) which – different from the above tests – applies parametric bootstrapping to compare the trees. The SOWH test tests the following hypotheses (cf. Goldman *et al.*, 2000):

H_0: The tree T_a is the true topology.
H_A: Some other topology is the true one.

To test each tree T_a from a set $\mathcal{T}$ the SOWH test proceeds as follows:

(i) Estimate the log-likelihood values l_{ML} and l_a and compute the test statistic $\delta = l_{ML} - l_a$.
(ii) Generate parametric bootstrap samples with Monte Carlo simulation along tree T_a with the ML parameters $\hat{\theta}_a$ estimated for tree T_a.
(iii) For each bootstrap sample, re-estimate the model parameters $\theta_a^{(i)}$ and the log-likelihood value $l_a^{(i)}$ for tree T_a (under the null hypothesis).
(iv) For each bootstrap sample, also re-estimate the model parameters $\theta_x^{(i)}$ and the log-likelihood value $l_x^{(i)}$ for all other trees $T_x \in \mathcal{T}$ to find the ML log-likelihood $l_{ML}^{(i)}$ for this bootstrap sample.
(v) Compute the difference values $\delta^{(i)} = l_{ML}^{(i)} - l_a^{(i)}$ which are interpreted as samples according to the distribution of δ under the null hypothesis H_0. Due to this assumption, no estimation of distribution parameters is performed.
(vi) Obtain the border of the rejection area directly from the generated distribution of $\delta^{(i)}$ values. To this end, empirically sum the $\delta^{(i)}$ values in ascending order until you have passed 95% of all $\delta^{(i)}$ values. Use this $\delta^{(i)}$ value as the critical value beyond which the null hypothesis is rejected.
(vii) Repeat this procedure for all different trees $T_a \in \mathcal{T}$.

The SOWH test utilizes the same test statistic δ as the KH and the SH test. Due to using the ML tree in the computation of δ, the assumption of $E[\delta] = 0$ would be inappropriate, however. Therefore, a one-sided test is used.

The repeated parametric bootstrap based on the respective T_a produces data conforming to the null hypothesis. Hence, no centering is necessary.

The main problem with tests based on parametric bootstrap is that they are computationally very demanding, often making extensive tests unfeasible. Furthermore, no straightforward implementation of the SOWH test seems available. Yet Goldman *et al.* (2000) give some advice at ***http://www.ebi.ac.uk/***

goldman/tests/ how to implement SOWH tests using PAUP* (see Chapter 8) and SEQ-GEN (Rambaut & Grassly, 1997).

12.6 Confidence sets based on likelihood weights

Strimmer and Rambaut (2002) approach the problem of comparing trees from a different perspective. Instead of significance testing, they devised a method that generates a **confidence set** of trees based on **expected likelihood weights** (ELW). They define a confidence set as the smallest subset of models – here trees – which together obtain a pre-defined probability C to be selected based on some random data set D (with length N) drawn from the true distribution of the evolutionary process. Note that this concept is related to ***credible sets*** of trees in Chapter 7.

Given the likelihoods L_x of each tree $T_x \in \mathcal{T}$, the likelihood weight w_a of a single tree T_a is computed as the fraction of the total likelihood summed over all trees $T_x \in \mathcal{T}$:

$$w_a = \frac{L_a}{\sum_x L_x} \tag{12.7}$$

with all likelihood weights w_a adding up to 1.0. One way of constructing a confidence set would be to collect all trees T_a in descending order of their weights until the sum of collected weights meets the pre-defined threshold value C, typically 0.95. This view is related to significance testing (see above) where the $1 - \alpha$ confidence region corresponds to the coverage of our confidence set by the cumulative level of confidence C.

To compute the precise selection probability, the *expected* likelihood weight $E[w_a]$, the true model has to be known which is hardly ever the case in reality. Hence, estimating the expected weights is based on a non-parametric bootstrap as in the previous sections (Fig. 12.5):

(i) Generate B bootstrap samples $D^{(i)}$. Estimate the corresponding likelihood values $L_x^{(i)}$ for each tree $T_x \in \mathcal{T}$ (e.g. with the RELL method).
(ii) Compute the likelihood weights $w_a(i)$ for all $T_a \in \mathcal{T}$ according to (12.7), within each bootstrap sample separately.
(iii) For each tree T_x derive its expected likelihood weight $E[w_a]$ by averaging over all bootstrap samples, assuming $E[w_a] \approx \overline{w}_a$:

$$\overline{w}_a = \frac{1}{B}\sum_{i=1}^{B} w_a^{(i)} \tag{12.8}$$

(iv) Construct the confidence set by selecting trees T_x in descending orders of their (inferred) expected weights $\overline{w}_x$ until their accumulated sum meets the pre-set level of confidence $C = 0.95$.

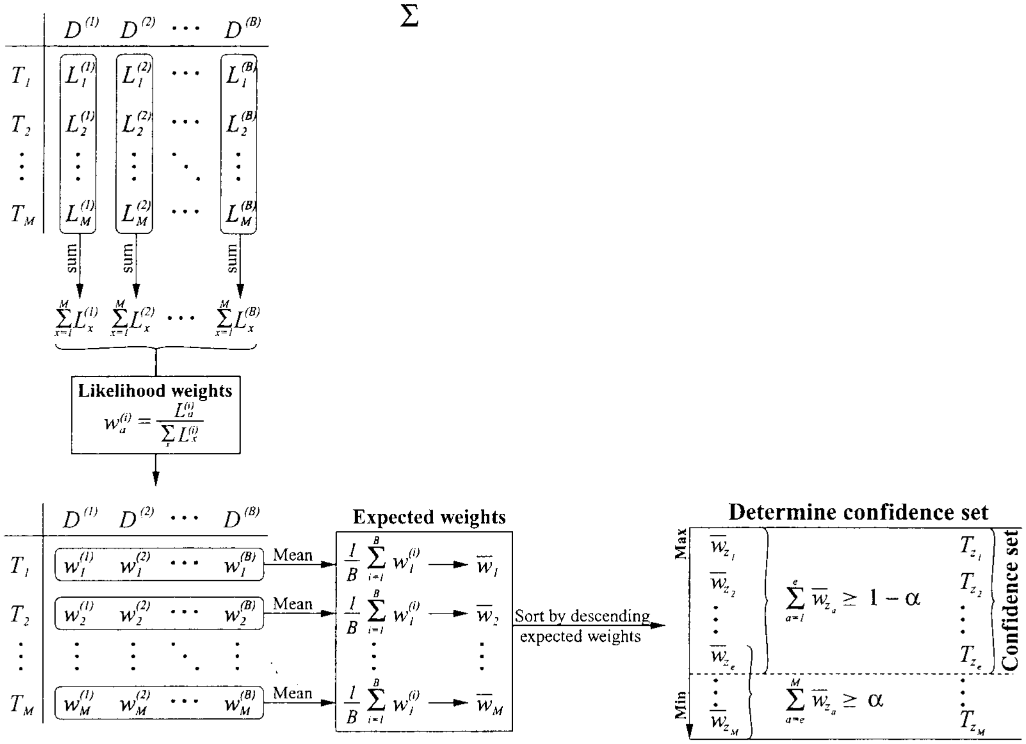
Σ
$D^{(1)}$ $D^{(2)}$ ··· $D^{(B)}$
T_1
T_2
T_M
$L_1^{(1)}$ $L_1^{(2)}$ ··· $L_1^{(B)}$
$L_2^{(1)}$ $L_2^{(2)}$ ··· $L_2^{(B)}$
$L_M^{(1)}$ $L_M^{(2)}$ ··· $L_M^{(B)}$
sum
$\sum_{x=1}^{M} L_x^{(1)}$ $\sum_{x=1}^{M} L_x^{(2)}$ ··· $\sum_{x=1}^{M} L_x^{(B)}$
Likelihood weights
$w_a^{(i)} = \frac{L_a^{(i)}}{\sum_x L_x^{(i)}}$
$w_1^{(1)}$ $w_1^{(2)}$ ··· $w_1^{(B)}$
$w_2^{(1)}$ $w_2^{(2)}$ ··· $w_2^{(B)}$
$w_M^{(1)}$ $w_M^{(2)}$ ··· $w_M^{(B)}$
Mean
Expected weights
$\frac{1}{B}\sum_{i=1}^{B} w_1^{(i)} \rightarrow \overline{w}_1$
$\frac{1}{B}\sum_{i=1}^{B} w_2^{(i)} \rightarrow \overline{w}_2$
$\frac{1}{B}\sum_{i=1}^{B} w_M^{(i)} \rightarrow \overline{w}_M$
Sort by descending expected weights
Determine confidence set
Max
Min
$\overline{w}_{z_1}$
$\overline{w}_{z_2}$
$\overline{w}_{z_e}$
$\overline{w}_{z_M}$
$\sum_{a=1}^{e} \overline{w}_{z_a} \geq 1-\alpha$
$\sum_{a=e}^{M} \overline{w}_{z_a} \geq \alpha$
T_{z_1}
T_{z_2}
T_{z_e}
T_{z_M}
Confidence set

Fig. 12.5 Sketch of the confidence set generation from expected likelihood weights. From the bootstrap samples $D^{(1)} \ldots D^{(B)}$ likelihoods $L_1^{(1)} \ldots L_M^{(B)}$ are computed for each tree $T_1 \ldots T_M$. Based on each bootstrap sample, each likelihood $L_a^{(i)}$ is converted to a likelihood weight $w_a^{(i)}$. From these weights, the expected likelihood weight $\overline{w}_a$ for each tree T_a across the corresponding bootstrap samples is computed. The trees are sorted by their expected likelihood weights $\overline{w}_a$. The confidence set collects trees in descending order such that the cumulative expected weights $\sum_{a=1}^{e} \overline{w}_{z_a}$ just contains the fraction $1 - \alpha$.

This method for selecting a confidence set seems not to be affected by the problem of SH, i.e. extending the constructed confidence set as more and more trees are added as input (Strimmer & Rambaut, 2002). It is also independent of whether or not the true best tree is among the input trees. Nevertheless, the simplifications made need long sequence data sets D to correct for possible model mis-specification, and large enough numbers of bootstrap samples to get valid estimates from the parametric bootstrap (especially if the RELL method is used). The impact of model mis-specification with data sets, however, remains unclear.

12.7 Conclusions

All methods we have examined above provide us with a kind of confidence set of trees, a subset from our input set $\mathcal{T}$. The trees within this confidence set cannot be classified statistically as significantly better, worse, or different (depending on the hypotheses tested) by the means of their likelihood values. That means, when two trees are selected for the confidence set, we cannot discuss their differences as significant even though their likelihoods might differ and their topologies might substantially contradict each other.

The trees in the confidence set are usually assumed to be close to the true tree. This conclusion is difficult to confirm, however, since the true tree might not be among those tested. Furthermore, model mis-specifications and violations of basic assumptions might render the test results invalid.

We have seen that it is of utmost importance to take into account the hypotheses and assumptions a test is based on. Knowing these limitations allows us to draw valid conclusions from tests we apply and, vice versa, to determine what tests are appropriate to answer certain questions we want to ask about our data.