The Phylogenetic Handbook

A Practical Approach to DNA and Protein Phylogeny

Edited by
Philippe Lemey, Marco Salemi, and Anne-Mieke Vandamme

Contents

12

Testing Tree Topologies

Heiko A. Schmidt

THEORY

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 to get support values on branches (Chapter 5). Furthermore, other methods that generate or sample sets of plausible trees are used to get support values like in Bayesian MCMC sampling (Chapter 7) or quartet puzzling (Chapter 6).

distance chpt

Bayes chpt

ML chpt

Various approaches have been devised to determine a best-suited ***evolutionary model*** (Chapter 10). Such approaches are often based on the ***maximum likelihood*** values gained 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 and 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 and Buckley, 2004, and Chapter 10 for details).

modeltest chpt, do they explain LTR, Akaike...

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 *et al.*, 2000), we will restrict ourselves to review a number of common tests for which 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, 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 and Castellan, 1988, p. 7). This null hypothesis has to be precise, since the test of significance is based on its rejection (cf. 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 errors possible when testing the null hypothesis H_0. First, the null hypothesis is rejected when it is 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., Zharkikh and Li, 1995; Siegel and Castellan, 1988).

Given a ***sample 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 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 equally or more extreme (with respect to the null hypothesis) than the observed value μ_0 and can only be determined after the test (Goodman, 1999). If the null hypothesis is rejected the p-value is also necessarily less than α.

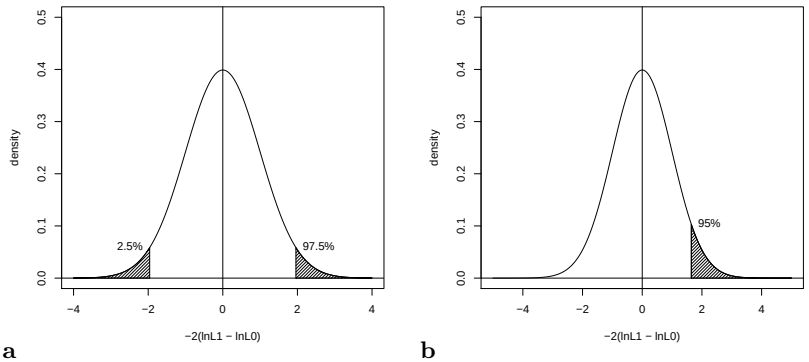

Fig. 12.1. Normal distributions for (a) a two-sided and (b) a one-sided test. An observed value μ_0 is (a) significantly different from the expectation μ if it is in one of the two shaded tails covering each 2.5% of the surface below the curve on either side or (b) significantly larger if it is above 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 $\alpha = 0.05$.

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 there 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 gaining the best maximum liklihood 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, that means, 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

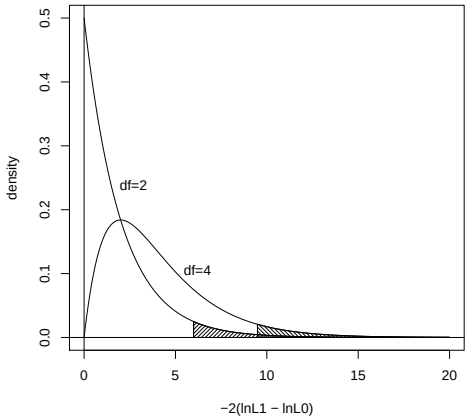

Fig. 12.2. χ^2 distribution for 2 and 4 degrees of freedom. An LRT is assumed significant, i.e., the more parameter-rich has a significantly higher likelihood L_1, if the Δ value is in the shaded 95% quantile.

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

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 Eq. 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.)

Selection chpt, does it still do LRT?

Although this methodology is straightforward for nested models, it is not generally 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 and Crandall, 1997).

If the tested models are not nested, their 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

$$\delta = \ln \frac{L_a}{L_b} = \ln L_a - \ln L_b = l_a - l_b \tag{12.2}$$

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 dataset D as described in Section 6.3.

To judge whether the obtained likelihoods are significantly different we need information about what the 'real' distribution of likelihood differences looks like.

In the ideal case one woule 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 by tossing a coin or roll dice a couple of additional times. Usually, we only have a limited dataset, the alignment where each column is usually regarded an independent sample from the 'true' process of evolution, to determine the desired distribution.

A common way to determine such distributions from limited datasets if further samples from the original process cannot be obtained are bootstrap re-sampling methods (Efron, 1979; Efron and 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 required that the values computed via the bootstrap have 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 (cf. **?**). 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 $\widetilde{l_x^{(i)}} = l_x^{(i)} - \bar{l}_x$. From the centered log-likelihoods $\widetilde{l}_x^{(i)}$ the likelihood ratios $\delta^{(i)}$ are computed between pairs of trees according to Eq. 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 (Eq. 6.12) or the log-likelihood as the sum of all site-log-likelihoods (Eq. 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 origi-

nal alignment D which then serve as input for Monte Carlo simulation performed by sequence generation programs such as seq-gen (Rambaut and 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 (cf. 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 exists 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 to refer to 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 relevant pairs of trees depend on the respective null hypothesis H_0 (rf. 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:

(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 $\delta^{(i)} = \tilde{l}_a^{(i)} - \tilde{l}_b^{(i)}$.

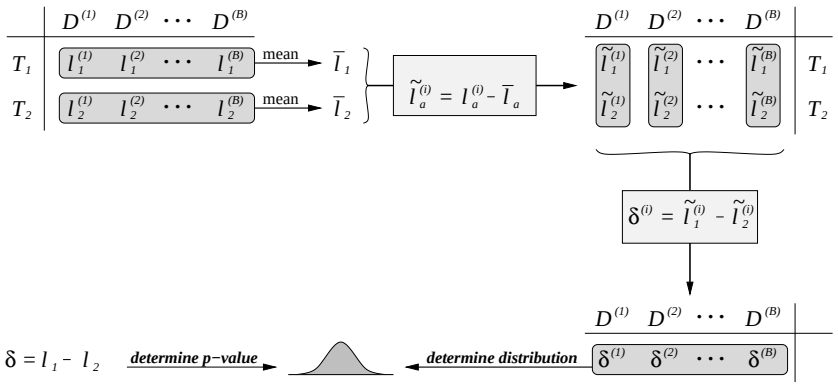

Fig. 12.3. Sketch of the Kishino-Hasegawa test. For both trees T_1 and T_2 the log-likelihoods l_a^i are computed for each bootstrap sample $D^{(i)}$. The log-likelihoods are subsequently 'centered' by the trees mean log-likelihood $\bar{l}_a$ across the bootstrap samples. The log-likelihood difference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

(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.

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 like 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 sub-optimal trees are equally supported or significantly worse than the best tree, or 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 test. However, a the KH test has often been applied this way (see

Goldman *et al.*, 2000; Shimodaira and Hasegawa, 1999, for extensive discussion).

The only way to adjust the KH test to this scenario to some extent 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:

(i) Estimate the log-likelihood values l_{ML} and l_x for all trees $T_x \in \mathcal{T}$. Compute $\delta_x = l_{ML} - l_x$.

(ii) Generate many ($B \geq 1000$) bootstrap samples i and 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 $\tilde{l_{ML}^{(i)}}$.

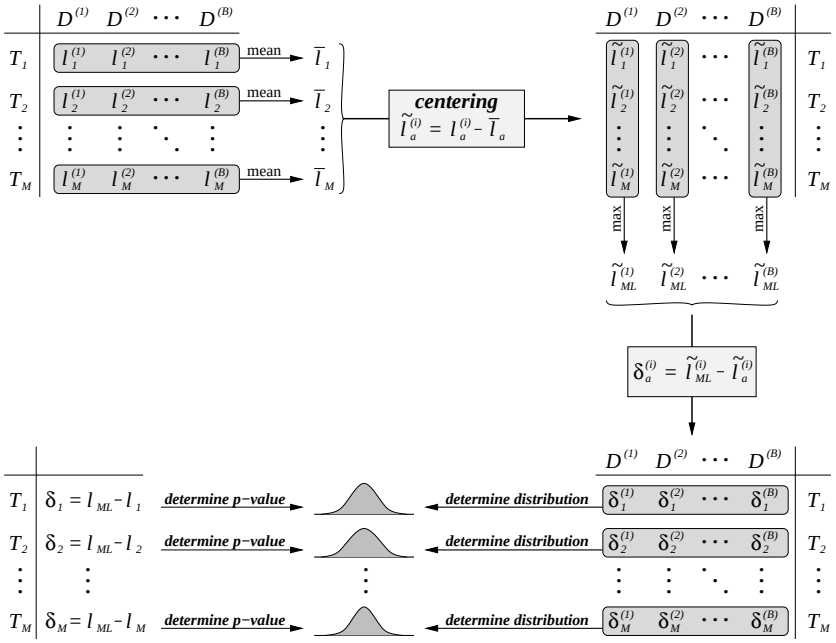

Fig. 12.4. Sketch of the Shimodaira-Hasegawa test. Log-likelihoods l_a^i are computed for each tree T_a and each bootstrap sample $D^{(i)}$ and subsequently 'centered' by the trees mean log-likelihood $\bar{l}_a$ across the bootstrap samples. For each bootstrap sample $D^{(i)}$ the tree with maximal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

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; Westfall and 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 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 the weighting the likelihood ratio depending on its variance, the tests are less conservative. Although this compensates for some of the above-mentioned conservative behaviour 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 couterparts.

12.5.6 The approximately unbiased 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 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, Olsen, Waddell and Hillis (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 interpreted as a 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 generated distribution of $\delta^{(i)}$ values. To this end, empirically count 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 and Grassly, 1997).

PAUP chpt

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 dataset D (with length N) drawn from the true distribution of the evolutionary process.

mention confidence sets of trees earlier

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 predefined threshold value C, typically 0.95. This view is related to significance testing (cf. 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:

(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 Eq. 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 $C = 0.95$.

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 and Rambaut, 2002). It is also independent from whether the true best tree is among

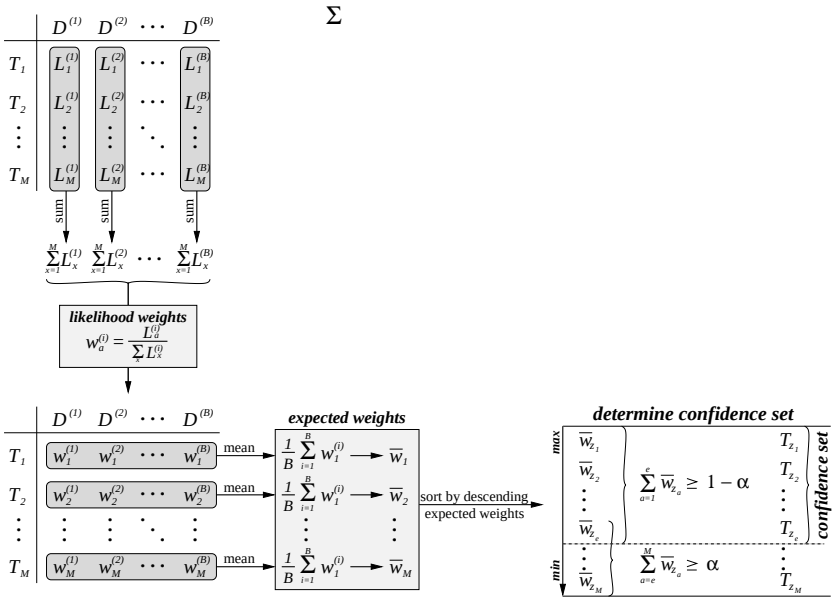

Fig. 12.5. Sketch of the confidence set generation from expected likelihood weights. From the bootstrap samples $D^{(1)} \dots D^{(B)}$ likelihoods $L_1^{(1)} \dots L_M^{(B)}$ are computed for each tree $T_1 \dots T_M$. Based on each bootstrap sample each likelihood $L_a^{(i)}$ is converted to a likelihood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

the input trees. Nevertheless, the simplifications made need long sequence datasets 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 datasets, 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 statistically be classified 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 be confirmed, 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.

12.8 Software packages

ModelTest chpt
clock chpt

While LRTs (cf. Chapters 10 and 11) for nested models are very abundant, only a few packages allow the execution of various tests like TREE-PUZZLE, PAUP*, or the CONSEL software package. More programs are available which only implement the KH and SH test. A detailed list of phylogenetic software which also might contain other test software or different tests can be found at Joe Felsenstein's web site, `http://evolution.genetics.washington.edu/PHYL`

Because such program packages are emerging at a rapid pace, the reader is advised to visit this web site and the software pages for updates.

12.9 Testing a set of trees with TREE-PUZZLE and CONSEL

In this example we will compare a set of 15 trees contained in `hivALN.15trees` based on the alignment in the `hivALN.phy` file. Both files are available from `http://www.thephylogenetichandbook.org`. We will use TREE-PUZZLE 5.3 (Schmidt *et al.*, 2002) to perform a number of tests (KH, SH, ELW) and to compute the site-log-likelihoods for the 15 trees. Subsequently, we will use the CONSEL (Shimodaira and Hasegawa, 2001) software to perform the AU test and a number of other tests. Both programs can be run under Linux/Unix, MacOS X, and Windows. CONSEL does need an input file with the site-log-likelihood values for the assessed trees, which we will also construct with TREE-PUZZLE. The tests implemented in both programs are based on the bootstrapping with the RELL method. Hence, no re-estimation of parameters or branch lengths will be performed. For the following we will assume a significance limit $\alpha = 0.05$.

ML chpt
ML chpt

The maximum likelihood analysis of the `hivALN.phy` dataset in Chapter 6 has shown that the relationships between five subtrees could not be resolved reliably (cf. Fig. 6.8b). Those subtrees were the outgroup (comprising HIV-1 group O and SIV sequences), from the HIV-1 group M a joined subtree of subtypes B and D, and the the three separate group M subtypes A, C, and G. Five disjoint subtrees can be related in a fully resolved tree in exactly 15 possible ways. These 15 possible trees are contained in `hivALN.15trees` and also sketched in Fig. 12.6. The branching orders within the subtrees were kept, since there was no contradiction between the two trees reconstructed with the two different methods (see Fig. 6.8).

ML chpt

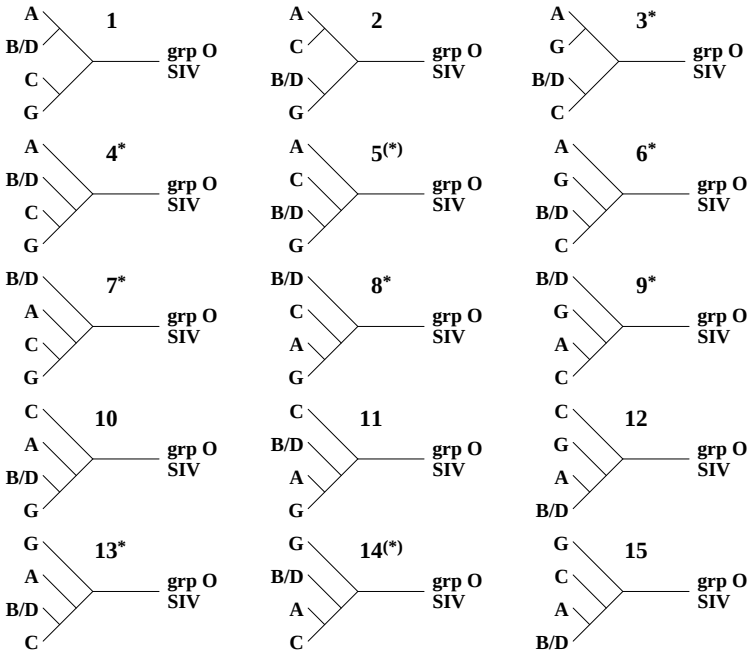

Fig. 12.6. The 15 possible tree topologies relating the four HIV-1 group M subtype-clades A, B/D, C, and G with the outgroup. The numbers give the order in the tree file. Tree 6 was the reconstructed ML tree from Fig. 6.8a. The stars * indicate the acceptance by the AU test. Those stars in brackets (*) had been excluded by the ELW method although excepted by the AU test.

12.9.1 Testing and obtaining site-likelihood with TREE-PUZZLE

The usage of TREE-PUZZLE has been explained in more detail in Chapter 6. Therefore, we will only highlight the relevant details and refer to the practical part of Chapter 6 for more information. ML chpt ML chpt

For analysis in the subsequent section we require TREE-PUZZLE to write the estimated side-log-likelihood values to a file for later use with CONSEL. To that end, we start TREE-PUZZLE with the command-line option `-wsl` (write site-likelihoods). It is recommended to execute TREE-PUZZLE with the following command from a terminal for this example:

```
puzzle -wsl hivALN.phy hivALN.15trees
```

where the first filename is the alignment and the second the tree file.

Then we will run TREE-PUZZLE with the following settings (changing options `k`, `x`, `w`, and `c`:

```
GENERAL OPTIONS
b                    Type of analysis? Tree reconstruction
k               Tree search procedure? Evaluate user defined trees
z   Compute clocklike branch lengths? No
e                 Parameter estimates? Approximate (faster)
x          Parameter estimation uses? Neighbor-joining tree

SUBSTITUTION PROCESS
d        Type of sequence input data? Auto: Nucleotides
h           Codon positions selected? Use all positions
m              Model of substitution? HKY (Hasegawa et al., 1985)
t  Transition/transversion parameter? Estimate from data set
f             Nucleotide frequencies? Estimate from data set

RATE HETEROGENEITY
w        Model of rate heterogeneity? Gamma distributed rates
a Gamma distribution parameter alpha? Estimate from data set
c    Number of Gamma rate categories? 4

Quit [q], confirm [y], or change [menu] settings: y
```

After typing 'y', TREE-PUZZLE will compute the maximum-likelihood values for all trees in `hivALN.15trees`, output the site-likelihood values to a file called `hivALN.15trees.sitelh`, and then proceeds with the implemented KH and SH tests and the ELW method, indicated by

```
[...]
Computing maximum likelihood branch lengths (without clock) for tree # 1
Computing maximum likelihood branch lengths (without clock) for tree # 2
Computing maximum likelihood branch lengths (without clock) for tree # 3
Computing maximum likelihood branch lengths (without clock) for tree # 4
Computing maximum likelihood branch lengths (without clock) for tree # 5
Computing maximum likelihood branch lengths (without clock) for tree # 6
Computing maximum likelihood branch lengths (without clock) for tree # 7
Computing maximum likelihood branch lengths (without clock) for tree # 8
Computing maximum likelihood branch lengths (without clock) for tree # 9
Computing maximum likelihood branch lengths (without clock) for tree # 10
Computing maximum likelihood branch lengths (without clock) for tree # 11
```

```
Computing maximum likelihood branch lengths (without clock) for tree # 12
Computing maximum likelihood branch lengths (without clock) for tree # 13
Computing maximum likelihood branch lengths (without clock) for tree # 14
Computing maximum likelihood branch lengths (without clock) for tree # 15
Performing single sided KH test.
Performing ELW test.
Performing SH test.

All results written to disk:
       Puzzle report file:         hivALN.15trees.puzzle
       Likelihood distances:       hivALN.15trees.dist
       Phylip tree file:           hivALN.15trees.tree
       Site likelihoods (PHYLIP):  hivALN.15trees.sitelh

The parameter estimation took 9.00 seconds (= 0.15 minutes = 0.00 hours)
The computation took 28.00 seconds (= 0.47 minutes = 0.01 hours)
     including input 28.00 seconds (= 0.47 minutes = 0.01 hours)
```

Note, that when analyzing tree files, the tree file's name `hivALN.15trees` is used as prefix for the output files. Besides other information, the resulting output file `hivALN.15trees.puzzle` shows the results of the ML analysis and the test results for the different topologies at the end of the file:

```
COMPARISON OF USER TREES (NO CLOCK)

Tree   log L   difference    S.E.    p-1sKH      p-SH     c-ELW    2sKH
--------------------------------------------------------------------------
 1  -17405.05    12.13      9.0392   0.0960 +  0.1870 +  0.0051 -     +
 2  -17405.90    12.99      8.9989   0.0780 +  0.1760 +  0.0027 -     +
 3  -17395.02     2.11      3.4895   0.2600 +  0.7860 +  0.1147 +     +
 4  -17401.24     8.33      8.2551   0.1580 +  0.3830 +  0.0704 +     +
 5  -17404.03    11.12      7.4308   0.0720 +  0.2290 +  0.0097 -     +
 6  -17392.91     0.00  <---- best   1.0000 +  1.0000 +  0.4437 +  best
 7  -17401.49     8.58      9.7675   0.1780 +  0.3760 +  0.0587 +     +
 8  -17396.14     3.22      6.8145   0.3160 +  0.7170 +  0.1770 +     +
 9  -17401.98     9.07      9.7895   0.1700 +  0.3400 +  0.0460 +     +
10  -17408.52    15.61      8.3014   0.0380 -  0.0780 +  0.0003 -     +
11  -17399.72     6.81      5.7552   0.1170 +  0.4840 +  0.0085 -     +
12  -17408.66    15.75      8.3151   0.0250 -  0.0740 +  0.0003 -     +
13  -17396.12     3.21      2.9334   0.1280 +  0.6930 +  0.0580 +     +
```

```
14  -17405.43    12.52      8.9404   0.0670 +  0.1910 +  0.0042 -     +
15  -17408.24    15.33      8.2263   0.0370 -  0.0860 +  0.0009 -     +
```

This table contains the trees in the same order as in the tree file. The columns show from the second column the log-likelihoods, the log-likelihood differences δ_x to the best tree, their standard error, the p-values of the one-sided Kishino-Hasegawa test (p-1sKH), the Shimodaira-Hasegawa test (p-SH) and the expected likelihood weights (c-ELW). The results of the two-sided KH test (2sKH) are only included for historical reason and should not be used. Due to the random generation of the bootstrap samples the p-values might differ slightly between repeated analyses. The '`+`' signs after the values indicate whether a method has chosen the corresponding tree for the confidence set or (equivalently) where the null hypothesis could not be rejected.

We already see that the three relevant columns show different sizes of their confidence sets (SH > KH > ELW), which reflects the different levels of conservativeness. The Shimodaira-Hasegawa test has even included all 15 trees, that means it could not reject a single null hypothesis.

12.9.2 Testing with CONSEL

The CONSEL program does not estimate the likelihoods for the trees itself. Hence we generated a site-likelihood file `hivALN.15trees.sitelh` with the `-wsl` option in TREE-PUZZLE in the previous section.

Note, that CONSEL version 0.1i does have problems with filenames that contain more than one dot ('.'). In such cases CONSEL discards the last file extension, but fails to attach its own ones, which in this example would lead to the loss of the contents of `hivALN.15trees`, being overwritten by the CONSEL output.

Therefore, rename the site-likelihood file accordingly, say to `hivALN.sitelh`. CONSEL generates the multiscaled bootstrap samples from TREE-PUZZLE `.sitelh` in CONSEL's own native `.rmt` format by running

```
makermt --puzzle hivALN.sitelh
```

While `makermt` performs the conversion and the bootstrapping it indicates its progress:

```
# $Id: makermt.c,v 1.14 2005/09/20 07:58:03 shimo Exp $
# seed:0 (MT19937 generator)
# K:10
```

```
# R:0.5 0.6 0.7 0.8 0.9 1 1.1 1.2 1.3 1.4
# B:10000 10000 10000 10000 10000 10000 10000 10000 10000 10000
# reading hivALN.sitelh
# M:15 N:2352
# writing hivALN.vt
# writing hivALN.rmt
# start generating total 100000 replicates for 15 items.........
# time elapsed for bootstrap t=15.89 sec
# exit normally
```

`K` indicates the number of differently scaled bootstrap samples, `R` lists the different alignment length scale factors $\frac{N}{N_r}$, and `B` the numbers of bootstrap samples for each N_r (cf. 12.5.6).

The command

```
consel hivALN
```

then performes the actual tests on the multiscaled bootstrap samples and writes the p-values to file, indicating as it proceeds

```
# $Id: consel.c,v 1.19 2004/11/11 08:14:09 shimo Exp $
# reading hivALN.rmt.........
# K:10
# R:0.5 0.599915 0.69983 0.799745 0.89966 1 1.09991 1.19983 1.29974 1.39966
# B:10000 10000 10000 10000 10000 10000 10000 10000 10000 10000
# M:15
# generate the identity association
# CM:15
# MC-TEST STARTS
# centering the replicates
# calculating kh-pvalue..............
# calculating mc-pvalue..............
# calculating the variances..............
# calculating weighted kh-pvalue..............
# calculating weighted mc-pvalue..............
# MC-TEST DONE
# calculate replicates of the statistics.........
# BP-TEST STARTS - DONE
# AU-TEST STARTS
# sorting the replicates.........
# calculating approximately unbiased p-values by MLE (fast) fitting..............
# time elapsed for AU test is t=0.05 sec
```

```
# ALPHA:0.05 0.1 0.5 0.9 0.95
# calculating confidence intervals..............
# AU-TEST DONE
# writing hivALN.pv
# writing hivALN.ci
# exit normally
```

To extract the resulting p-values from CONSEL's data files we run the command

```
catpv -s 1 hivALN
```

This generates a table of p-values in the following format:

```
# rank item    obs     au     np |     bp     pp     kh     sh    wkh    wsh |
#    1    1   12.1  0.045  0.001 |  0.001  4e-06  0.086  0.187  0.086  0.407 |
#    2    2   13.0  0.029  0.001 |  0.001  2e-06  0.074  0.157  0.074  0.383 |
#    3    3    2.1  0.450  0.095 |  0.092  0.101  0.264  0.798  0.264  0.769 |
#    4    4    8.3  0.234  0.061 |  0.061  2e-04  0.148  0.381  0.148  0.509 |
#    5    5   11.1  0.068  0.009 |  0.009  1e-05  0.068  0.219  0.068  0.305 |
#    6    6   -2.1  0.810  0.483 |  0.475  0.831  0.736  0.962  0.682  0.965 |
#    7    7    8.6  0.274  0.066 |  0.070  2e-04  0.185  0.362  0.185  0.571 |
#    8    8    3.2  0.498  0.188 |  0.190  0.033  0.318  0.721  0.318  0.746 |
#    9    9    9.1  0.266  0.068 |  0.065  1e-04  0.170  0.340  0.170  0.545 |
#   10   10   15.6  2e-04  2e-05 |      0  1e-07  0.032  0.068  0.032  0.206 |
#   11   11    6.8  0.043  0.003 |  0.003  0.001  0.116  0.478  0.089  0.364 |
#   12   12   15.7  0.015  1e-04 |  8e-05  1e-07  0.029  0.059  0.029  0.214 |
#   13   13    3.2  0.149  0.027 |  0.027  0.034  0.134  0.702  0.134  0.485 |
#   14   14   12.5  0.086  0.006 |  0.006  3e-06  0.081  0.178  0.081  0.375 |
#   15   15   15.3  0.007  3e-04 |  2e-04  2e-07  0.033  0.070  0.033  0.214 |
```

The `-s 1` option causes the trees to be output in the order of the tree file instead of their likelihood (the default). The output contains the index (`rank`, which would differ when ordered by likelihood), the tree number in the input (`item`), the log-likelihood difference δ_x to the best tree (`obs`), except for the best itself which shows the negative distance of the second best. Among other statistics and information values it prints the p-values of the AU test (`au`), KH test (`kh`), the SH test (`sh`), and weighted variants of the two (`wkh` and `wsh`). Due to the random generation of the bootstrap samples the p-values might differ slightly between repeated analyses. We assume the confidence sets based on a significance level $\alpha = 0.05$ are the largest for the SH test(s) and their

Table 12.1. *p-values estimated with TREE-PUZZLE and CONSEL. The '+' signs mean that a tree belongs to a confidence set or the null hypothesis could not be rejected.*

			TREE-PUZZLE			CONSEL		
	$\ln L$	δ	KH	SH	ELW	AU	KH	SH
1	-17405.05	12.13	0.096 +	0.187 +	0.0051	0.030	0.094 +	0.193 +
2	-17405.90	12.99	0.078 +	0.176 +	0.0027	0.044	0.076 +	0.159 +
3	-17395.02	2.11	0.260 +	0.786 +	0.1147 +	0.433 +	0.272 +	0.793 +
4	-17401.24	8.33	0.158 +	0.383 +	0.0704 +	0.235 +	0.164 +	0.384 +
5	-17404.03	11.12	0.072 +	0.229 +	0.0097	0.072 +	0.072 +	0.227 +
6	-17392.91	0.00	1.000 +	1.000 +	0.4437 +	0.826 +	0.728 +	0.960 +
7	-17401.49	8.58	0.178 +	0.376 +	0.0587 +	0.281 +	0.190 +	0.370 +
8	-17396.14	3.22	0.316 +	0.717 +	0.1770 +	0.475 +	0.319 +	0.721 +
9	-17401.98	9.07	0.170 +	0.340 +	0.0460 +	0.272 +	0.175 +	0.339 +
10	-17408.52	15.61	0.038	0.078 +	0.0003	0.005	0.034	0.072 +
11	-17399.72	6.81	0.117 +	0.484 +	0.0085	0.042	0.121 +	0.487 +
12	-17408.66	15.75	0.025	0.074 +	0.0003	0.040	0.034	0.064 +
13	-17396.12	3.21	0.128 +	0.693 +	0.0580 +	0.141 +	0.135 +	0.695 +
14	-17405.43	12.52	0.067 +	0.191 +	0.0042	0.074 +	0.082 +	0.177 +
15	-17408.24	15.33	0.037	0.086 +	0.0009	0.033	0.037	0.075 +

size is decreasing via the KH test(s) down to the AU test. While the p-values sometimes differ substantially between the weighted and unweighted versions of the SH and KH tests, the confidence sets remain the same.

12.10 Conclusions

Comparing the results of TREE-PUZZLE and the CONSEL program which are collected in Table 12.1 one can easily see that although the p-values of the tests computed by both programs differ slightly, the resulting confidence sets are the same. The differences of the p-values can be attributed to two reasons. First, the stochastic nature of the bootstrap samples causes fluctuations in the estimated values. Furthermore, rounding errors are introduced during the export of the site-log-likelihoods. Only TREE-PUZZLE itself has the advantage of the full precision, since the export to a text file for CONSEL does automatically decrease the precision of exported values.

Note, that although the 15 trees were constructed manually their selection was still *a priori*, since we know that tree 6 was the reconstructed ML tree (cf. Chapter 6). That tree remained the one with highest like- ML chpt

lihood even among the constructed trees in the analysis.

The KH test has been performed as a one-sided test, but still the assumptions implied lets one only use this test with caution. The SH tests nicely demonstrated its biasedness being the only test (together with its weighted variant WSH) not rejecting any of the trees.

The ELW and the AU test show very similar confidence sets, but the AU test is a bit more conservative. Still one has to keep in mind that the impact of model misspecification on ELW remains unclear.

The interpretation of these results are, that there is no tree to be clearly chosen over the others. But also the topological features of the accepted or rejected trees do not show an obvious trend. It seems clear, however, that the subtype C is not located close to the root, since all such trees (trees 10-12 in Fig. 12.6) have been rejected by ELW and the approximately unbiased AU test.

Furthermore, all those topologies are in the confidence sets of AU and ELW, where subtype A and G are located closely in the tree (Fig. 12.6, trees 3, 6, 8, 9, and 13) except if subtype C is close to the root (trees 11 and 12). This might reflect the possibly recombinant history of subtype G with subtype A being one of its parents as reported by Abecasis *et al.* (2005).

The tests showed that a number of rather different tree topologies could not be rejected. Other subsequent analyses have to be performed to clarify the possible influence of inter-subtype recombination (Chapter 12) or of other phenomena like noise, lateral gene transer, or directed selection which can also obscure the true phylogenetic history.

bootscan
chpt

Bibliography

Abecasis, A. B., Lemey, P., Vidal, N., de Oliveira, T., Peeters, M., Camacho, R., Shapiro, B., Rambaut, A. and Vandamme, A.-M. (2005) Recombination is confounding the early evolutionary history of hiv-1: subtype g is a circulating recombinant form. *J. Virol.*, **81**, in press.

Cox, D. R. (1961) Tests of separate families of hypotheses. In *Proceedings of the 4th Berkeley Symposium on Mathematical Statistics and Probability*, pp. 105–123, UCB Press, Berkeley, CA, USA.

Cox, D. R. (1962) Further results on tests of separate families of hypotheses. *J. R. Soc. Stat. B*, **24**, 406–424.

Efron, B. (1979) Bootstrap methods: Another look at the jackknife. *Ann. Stat.*, **7**, 1–26.

Efron, B., Halloran, E. and Holmes, S. (1996) Bootstrap confidence levels for phylogenetic trees. *Proc. Natl. Acad. Sci. USA*, **93**, 13429–13434.

Efron, B. and Tibshirani, R. (1994) *An Introduction to the Bootstrap*. Chapman and Hall, New York.

Felsenstein, J. (1985) Confidence limits on phylogenies: An approach using the bootstrap. *Evolution*, **39**, 783–791.

Fisher, R. A. (1971) *Design of Experiments*. 9th edition, Macmillan.

Goldman, N. (1993) Statistical tests of models of DNA substitution. *J. Mol. Evol.*, **36**, 182–198.

Goldman, N., Anderson, J. P. and Rodrigo, A. G. (2000) Likelihood-based tests of topologies in phylogenetics. *Syst. Biol.*, **49**, 652–670.

Goodman, S. N. (1999) Toward evidence-based medical statistics. 1: The P value fallacy. *Ann. Intern. Med.*, **130**, 1019–1021.

Hall, P. and Wilson, S. R. (1991) Two guidelines for bootstrap hypothesis testing. *Biometrics*, **47**, 757–762.

Huelsenbeck, J. P. and Crandall, K. A. (1997) Phylogeny estimation and

hypothesis testing using maximum likelihood. *Annu. Rev. Ecol. Syst.*, **28**, 437–466.

Johnson, J. B. and Omland, K. S. (2004) Model selection in ecology and evolution. *TREE*, **19**, 101–108.

Kishino, H. and Hasegawa, M. (1989) Evaluation of the maximum likelihood estimate of the evolutionary tree topologies from DNA sequence data, and the branching order in Hominoidea. *J. Mol. Evol.*, **29**, 170–179.

Kishino, H., Miyata, T. and Hasegawa, M. (1990) Maximum likelihood inference of protein phylogeny and the origin of chloroplasts. *J. Mol. Evol.*, **31**, 151–160.

Posada, D. and Buckley, T. (2004) Model selection and model averaging in phylogenetics: Advantages of akaike information criterion and bayesian approaches over likelihood ratio tests. *Syst. Biol.*, **53**, 793–808.

Rambaut, A. and Grassly, N. C. (1997) Seq-Gen: An application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees. *Comput. Appl. Biosci.*, **13**, 235–238.

Schmidt, H. A., Strimmer, K., Vingron, M. and von Haeseler, A. (2002) TREE-PUZZLE: Maximum likelihood phylogenetic analysis using quartets and parallel computing. *Bioinformatics*, **18**, 502–504.

Shimodaira, H. (2002) An approximately unbiased test of phylogenetic tree selection. *Syst. Biol.*, **51**, 492–508.

Shimodaira, H. and Hasegawa, M. (1999) Multiple comparisons of log-likelihoods with applications to phylogenetic inference. *Mol. Biol. Evol.*, **16**, 1114–1116.

Shimodaira, H. and Hasegawa, M. (2001) CONSEL: for assessing the confidence of phylogenetic tree selection. *Bioinformatics*, **17**, 1246–1247.

Siegel, S. and Castellan, Jr, N. J. (1988) *Nonparametric Statistics For The Behavioural Sciences*. 2nd edition, McGraw-Hill, Ney York.

Strimmer, K. and Rambaut, A. (2002) Inferring confidence sets of possibly misspecified gene trees. *Proc. R. Soc. Lond. B*, **269**, 137–142.

Swofford, D. L., Olsen, G. J., Waddell, P. J. and Hillis, D. M. (1996) Phylogeny reconstruction. In Hillis, D. M., Moritz, C. and Mable, B. K. (eds.), *Molecular Systematics*, 2nd edition, pp. 407–514, Sinauer Associates, Sunderland, Massachusetts.

Westfall, P. H. and Young, S. S. (1993) *Resampling-based multiple testing: Examples and methods for p-value adjustment*. John Wiley and Sons, New York, USA.

Yang, Z., Goldman, N. and Friday, A. E. (1995) Maximum likelihood trees from DNA sequences: a peculiar statistical estimation problem. *Syst. Biol.*, **44**, 384–399.

Zharkikh, A. and Li, W.-H. (1995) Estimation of confidence in phylogeny: The complete-and-partial bootstrap technique. *Mol. Phylogenet. Evol.*, **4**, 44–63.