Stockholms
universitet

Prior Distributions on Phylogenetic Trees

Magnus Johansson

Masteruppsats i matematisk statistik
Master Thesis in Mathematical Statistics

Masteruppsats 2011:4
Matematisk statistik
Juni 2011

www.math.su.se

Matematisk statistik
Matematiska institutionen
Stockholms universitet
106 91 Stockholm

Stockholm University

Mathematical Statistics
Stockholm University
Master Thesis **2011:4**
http://www.math.su.se

Prior Distributions on Phylogenetic Trees

Magnus Johansson*

Juni 2011

Abstract

In Bayesian inference of phylogenetics the MCMC simulation method is used to estimate the support of phylogenetic trees. MCMC simulation is very useful for estimating the support of the most probable trees, but gives little information about the great majority of the possible trees. Bayesian inference has the ability to incorporate previous knowledge into a new analysis, specified in terms of a prior distribution. In this thesis I investigate four different methods for evaluating the result of an MCMC simulation in order to estimate the support of all possible trees, thus enabling the result to be used as a prior distribution. Two of the methods, the WIB and the BM method produced reasonable results, whereas the other two, the WAB and the SCM method did not produce reasonable results. I will discuss suggestions on how these methods could be improved in the future.

*Postal address: Mathematical Statistics, Stockholm University, SE-106 91, Sweden. E-mail:magnus_ltd@hotmail.com . Supervisor: Sebastian Höhna.

Contents

1 Introduction

1.1 Motivation

Bayesian statistics (see section 2.3.2 for details) involves specifying prior knowledge in advance and in terms of a probability distribution. Sometimes you want to say little in advance, in the phylogenetic application this could for example be done by giving equal weights to all trees. Sometimes you want to say more, this could for example be done by incorporating the results of a previous study. The support of a tree in Bayessian phylogenetic inference is most frequently computed by using the MCMC simulation algorithm (see section 2.4 for details), in which case the support is measured by the sample frequency. But, in most cases only the very best, i.e. most likely, trees will ever be sampled at least once, even if a long chain of simulations is run. This is due to the fact that the distribution of trees is often highly peaked, so the great majority of the possible trees will not be sampled at all. By only looking at the sample frequencies of the MCMC simulation we cannot say much about the trees that were not sampled. But, in order to assign a proper probability distribution we cannot allow any possible outcome to have zero probability. Hence, if we want to use the result of a previous analysis in another study we need a way of estimating the support of every possible tree.

1.2 Objective

The objective of this thesis is to compare known methods for evaluating the result of an MCMC simulation in order to estimate the support of each possible tree topology, as well as develop new improved methods and compare them with the known ones. Also, to give suggestions on which method to use under certain circumstances.

2 Background

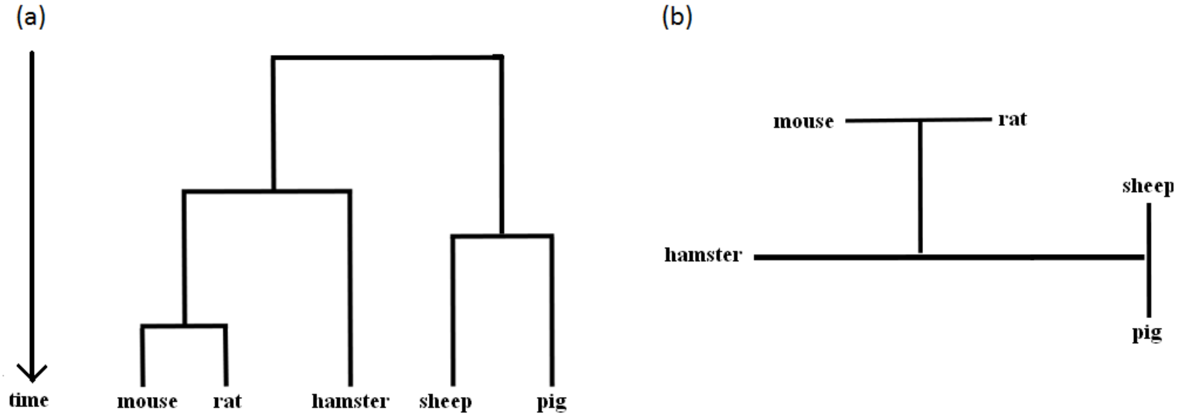
2.1 Phylogenetic inference

Phylogenetics is a field within biology which studies how organisms (plants, animals, etc.) are related to one another through the course of evolution. New species evolve from older ones and the mapping and understanding of their ancestry is the central task of phylogenetics. The relationship between species can be represented in the form of an evolutionary tree, a graph showing how the species with common ancestors are related. Phylogenetic trees are constructed using molecular sequencing data and morphological data. To construct a phylogenetic tree accurately involves using highly advanced mathematical models. Constructing and evaluating these trees are a part of phylogenetic inference. In this thesis I will focus on one relatively new approach to this problem, namely the Bayesian approach. This method is explained in more detail in section 2.3.2.

2.2 Phylogenetic trees

A phylogenetic tree is a graph where any two vertices are connected by exactly one simple path, i.e. it has no loops. If the tree only has one distinguished internal vertex, while the others are of degree of at least three, then the tree is said to be *rooted.* If the degree is of no more than three for any other internal vertex it is said to be *bifurcating.* If there is no distinguishable internal vertex the tree is said to be *unrooted.* For a more detailed description, see Semple and Steel [1]. Unlike rooted trees, unrooted trees have no natural way of representing time. Although this thesis mainly concerns the topology of trees, and not e.g. at which point in time a new species has evolved, I will only study rooted trees. I will also restrict myself to study bifurcating trees; since for most models of evolution, DNA mutations occur at random with a small rate and two mutations are assumed not to occur at the same time. Figure 1(a) is an example of a rooted tree with 5 taxa and Figure 1(b) an unrooted version (out of several possible) of the same tree. A phylogenetic tree, rooted and unrooted, is sometimes also referred to as a phylogeny, an evolutionary tree, or a cladogram. I will simply refer to them as trees.

Figure 1: Example of a rooted bifurcating tree (a) and unrooted verison (b) of the same phylogenetic tree (out of several possible) with 5 taxa. Notice that in (a) the horizontal axis measures the time of the branches and when the split occurred; this cannot be done in (b).



There are some useful formulas for calculating the number of possible tree topologies for n taxa (organisms). Felsenstein [2] shows that for rooted bifurcating trees, there are

$$T_{rooted}(n) = \frac{(2n-3)!}{2^{(n-2)} \cdot (n-2)!} \tag{1}$$

and for unrooted trees there are

$$T_{unrooted}(n) = \frac{(2n-5)!}{2^{(n-3)} \cdot (n-3)!} \tag{2}$$

different possible trees.

The number of possible trees thus increases at an exponential rate as the number of species increases. A classic example to illustrate the rapid increase of tree topologies is that there are roughly 10^{96} different rooted trees for 60 taxa. This can be compared with the estimated number of protons in the universe which is roughly 10^{89}. Hence, the sheer number of possible trees will lead to computational difficulties if not handled with care.

2.3 Methods for performing phylogenetic inference

There are several mathematical statistical methods for performing phylogenetic inference. In this section I will discuss some of them. The main focus lies on Bayesian inference of phylogenetics, which is the method I will use in this thesis. Nonetheless it might be interesting to give a brief account of some other methods and their strengths and weaknesses.

2.3.1 Maximum Likelihood (ML) in phylogenetic inference

The well-known Maximum Likelihood (ML) method was first developed in the early 20th centuary by R. A. Fisher. In the phylogenetic case ML uses mathematical statistical models of varying complexity to reconstruct the evolutionary trees, early works such as Felsenstein (1981) [20] shows how the ML can be calculated. In ML a hypothesis is judged based on how well it predicts the observed data. The tree with the highest likelihood, i.e. the tree which has the highest probability of producing the observed data, is preferred. The field of ML is well-developed and has a wide range of inference and model-fitting tools (see Whelan, S., Lio, P and Goldman [11], for a complete review of ML in phylogenetics). The main problem with the ML method in phylogenetic inference is the fact that the model has many parameters, and thus its complexity leads to extremely tedious calculations. Finding the maximum of the likelihood function involves searching through a multidimensional parameter space; current techniques that perform this task (e.g. Newton-Raphson) are not guaranteed to find the maximum. Some other problems are that it is difficult giving confidence intervals and specifying uncertainty.

2.3.2 Bayesian inference of phylogenetics

Bayesian inference of phylogenetics is based on the famous Bayes's theorem. In its simplest form, Bayes's theorem states (with the phylogenetic application in mind)

$$Pr(Tree|Data) = \frac{Pr(Data|Tree) \cdot Pr(Tree)}{Pr(Data)}. \tag{3}$$

The left hand side of equation (3) is known as the *Posterior Probability.* The term $Pr(Data|Tree)$ is the probability that the data is true given the tree, and is referred to as the *likelihood.* The term $Pr(Tree)$ is known as the *prior distribution.* The prior distribution incorporates our previous knowledge and must always be specified in advance whenever performing a Bayesian analysis. The posterior probability should be interpreted as the probability that the tree is correct given the prior, the data and the model of evolution. A poorly chosen prior distribution or model can lead to inaccurate conclusions about data, one must always keep this in mind when interpreting the posterior probability.

The prior distribution is what distinguishes Bayessian statistics from other major schools of statistics (e.g. maximum likelihood). The prior is both a strength and a weakness of the Bayesian analysis. In the case where previous knowledge is well established it is a strength to be able to incorporate this in the analysis. As for phylogenetic inference, only few groups of species are completely unstudied in terms of their evolutionary history, so systematists have in most cases some prior knowledge of the tree topology [5]. In the case where no previous knowledge exists, it is not always easy to specify this in the model. In this case we are interested in specifying a prior that has as little information as possible, often a uniform prior giving the same weight to all possible outcomes is used. This is often referred to as a flat prior or a non-informative prior, also the term minimum informal is used. Specifying a non-informative prior is a very delicate task but will not be further investigated in this thesis, for more information see Picket and Randle [18][19].

The posterior probability may seem easy to calculate explicitly, but this is most of the time not the case. It involves summation over all trees, then for each tree, integration over all possible values of the parameters, which usually involves branch lengths and substitution model parameter values and others. This is, with the exception of the most trivial cases, impossible to do analytically. The solution is to use an appropriate simulation method. The MCMC simulation algorithm is an excellent tool for this purpose, which I will solely use throughout this thesis. This will be discussed in more detail in section 2.4. For a more detailed description of Bayessian inference of phylogenetics see, Huelsenbeck et. al. [3][21], Holder and Lewis [4] and Alfaro and Holder [5].

2.3.3 Other methods of performing phylogenetic inference

- **Neighbor-joining.** The Neighbor-joining method compresses sequence on DNA into a distance matrix that represents an estimate of the evolutionary distance between sequence. This compression implies loss of information and reliable estimates of pairwise distances can be hard to obtain for divergent sequences. This is a relatively fast method which performs well when the divergence between sequences is low. For more information see Saitou, N and Nei, M [6] and Studier, J.A. and Keppler, K.J [7].

- **Parsimony.** Parsimony inference maps the history of a gene sequences onto a tree. A score is then assigned based on the minimum possible number of mutations that can produce the tree. Afterwards the trees are evaluated according to their scores. The parsimony method is also a rather fast method. The main disadvantage is that it can perform poorly if there is substantial variation in branch lengths. Also, it should be noted that the parsimony method is statistically not sound. For more information see: Farms, J.S. [8], Fittch, W.M. [9] and Kluge, A.G. and Farris, J.S [10]

2.4 Markov Chain Monte Carlo (MCMC) simulation

Computing the posterior probability is the aim of a Bayesian analysis. However, this is almost impossible (with the exception of the most trivial cases) to do analytically in the phylogenetic application. Fortunately, there are several ways of approximating this value. MCMC simulation is a class of algorithms which is most commonly used for this matter. The MCMC algorithm is a way of simulating a Markov chain with the desired posterior distribution as its equilibrium distribution and then using the frequency of a particular tree as an estimate of the marginal posterior probability of that tree.

The MCMC simulation algorithm takes a large amount of steps (thousands or even millions) that form a chain with the Markov property, i.e. each new step only depends on the previous one. One of the most frequently used MCMC-algorithms is the Metropolis-Hastings algorithm [22]. In the Metropolis-Hastings algorithm we start with a tree τ and data X. For each step, a proposal mechanisms suggests a new tree τ' somewhere in the parameter space. The proposal mechanism must satisfy the following the conditions: (1) it must be stochastic, (2) every possible tree must be accessible by repetition of the proposal mechanism and (3) the chain must be aperiodic. The newly proposed step is usually quite similar to the previous one, as it is usually generated by randomly changing a few of the parameters of the current one. The new tree τ' is then accepted (i.e. sampled) with probability

$$\alpha = \min\Big(1, \frac{f(\tau'|X)}{f(\tau|X)} \times \frac{f(\tau|\tau')}{f(\tau'|\tau)}\Big),$$

where α is known as the acceptance ratio.

If the move is accepted, τ' will become the next step in the chain. If the move is not accepted, τ will become the next step in the chain. Notice that $f(\tau|X)$ is the posterior probability, which is actually what we want to estimate. The brilliance of the Metropolis-Hastings algorithm is the fact that by the use of Bayes's theorem the acceptance ratio α can be calculated without having to calculate the posterior probability:

$$\alpha = \min\Big(1, \frac{f(\tau'|X)}{f(\tau|X)} \times \frac{f(\tau|\tau')}{f(\tau'|\tau)}\Big) = \min\Big(1, \frac{f(X|\tau')f(\tau')/f(X)}{f(X|\tau)f(\tau)/f(X)} \times \frac{f(\tau|\tau')}{f(\tau'|\tau)}\Big).$$

In the above equation, $f(X)$ cancels out, thus after some rearrangements we end up with:

$$\alpha = \min\Big(1, \frac{f(X|\tau')}{f(X|\tau)} \times \frac{f(\tau')}{f(\tau)} \times \frac{f(\tau|\tau')}{f(\tau'|\tau)}\Big).$$

Hence, we end up with three ratios; the likelihood ratio, $f(X|\tau')/f(X|\tau)$; the prior, ratio $f(\tau')/f(\tau)$ and the proposal, ratio $f(\tau|\tau')/f(\tau'|\tau)$. All of these ratios can easily be calculated [21].

Preferably, one would like all of the simulations to be independent of one another. The MCMC does not produce independent samples, however, one usually discards the first 10-50% of the simulations as a burn-in period, due to the fact that the first step is randomly chosen and does not necessarily represent a good estimate. It is also common practice to only sample from every n:th iteration (10th, 100th or so), thus reducing the dependency. The remaining samples will behave more like independent simulations, though it is important to keep in mind that they are not.

By repeating this procedure for a long time, the chain tends to stay in a region of high posterior probability, thus giving an accurate estimation of the posterior probability. The convergence of the MCMC algorithm is an interesting topic on its own. How can we be sure that enough simulations have been run in order to produce an accurate estimate of the posterior probability? This is still debated, but it will not be a part of this thesis. For more information about the MCMC algorithm in the phylogenetic context, see Holder and Lewis [4], Yang [12].

3 Methods

In this section I will discuss four methods for calculating the score of any given tree provided we have the result of an MCMC simulation. The scores can then be normalized in order to reflect a true probability, i.e. by dividing each score with the total sum of all the scores.

The Bayesian model for phylogenetic inference involves several different parameters, for example topology, branch lengths and substitution model parameters. Each parameter needs to have a prior specified in advance. The methods discussed in this thesis only involves the topology of the tree, and how to use the information from a previous analysis in order to specify the prior on the topology. When referring to a *tree* in this section I always mean the *topology* of the tree.

3.1 Basic definitions

The following definitions will be useful in explaining the methods and will be used from now on.

Definition 1. ***Bipartition.*** *A bipartition, also known as a clade, is a binary character (vector of 0's and 1's) which divides a set of taxa into two groups. A '1' represents that the taxon is present and a '0' represents that the taxon is absent.*

For example, the bipartition $\{0011\}$ divides four taxa into two groups of which taxa 3 and 4 are present and taxa 1 and 2 are absent.

Definition 2. ***Matrix representation of a tree.*** *In matrix representation, each row represents a bipartition and each column represents a taxon. Each tree topology then has a unique matrix representation up to the ordering of the rows. For a tree consisting of n taxa, it requires $n-1$ bipartitions to fully describe the tree, this results in the tree being represented as an $(n-1) \times n$-matrix.*

For example, the matrix representation of the tree in Figure 1 (a) is

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

The first column represents the first taxa, ”mouse” in this case, the second column represents the second taxa, ”rat”, etc. The last row may seem unnecessary, but it is useful when distinguishing certain subtrees from the whole tree. In some cases the trivial bipartition (i.e. the one consisting of only 1’s) can be omitted, however, in those cases I will say so.

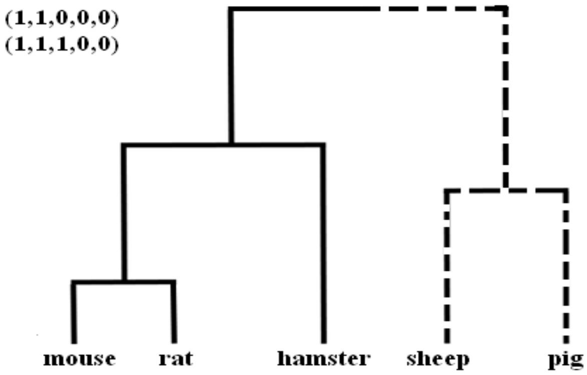
Definition 3. ***Newick representation of a tree.*** *In Newick representation, a tree is represented by enclosing each subtree within a bracket.*

The tree in Figure 1 (a) is written as (((mouse,rat),hamster),(sheep,pig)) in Newick representation. This way of describing a tree is convenient to use when describing a tree in written text.

Definition 4. ***Subtree.*** *A subtree is a set of bipartitions that fully describes a complete branch of the tree.*

Figure 2 is an example of a subtree consisting of 3 taxa from a tree consisting of 5 taxa.

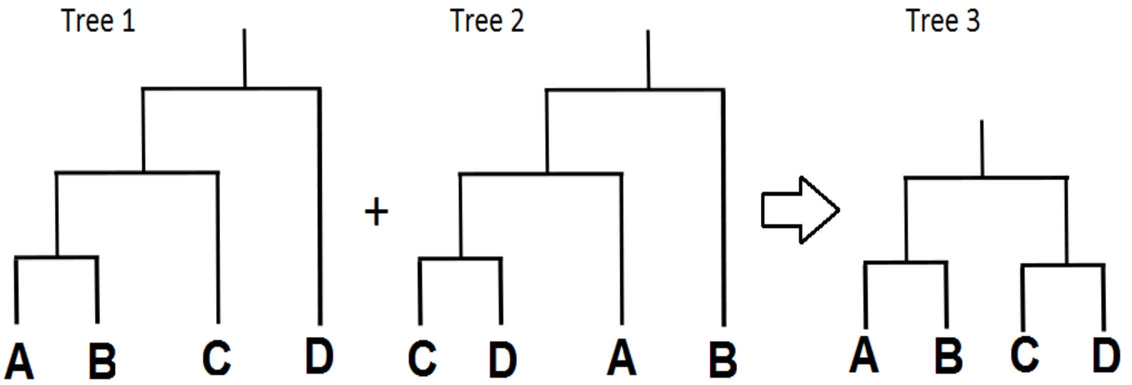
Figure 2: Example of a subtree of three taxa from a tree with five taxa. ((mouse,rat),hamster) is subtree of the whole tree.



3.2 The Bipartition Multiplication (BM) Method

As mentioned earlier, the MCMC algorithm is very unlikely to sample each tree at least once. The MCMC algorithm is also very unlikely to sample each bipartition at least once. However, by combining those bipartitions that actually were sampled from the MCMC simulation we can construct more trees than only those trees that were sampled. Consider Figure 3 as an example. Suppose that only Tree 1 and Tree 2 were sampled from the MCMC simulation. The subtrees (A,B) from Tree 1 and (C,D) from Tree 2 can then be combined to construct Tree 3. Even though tree 3 was

Figure 3: Combining two subtrees to form a new tree. The subtree (A,B) of Tree 1 and (C,D) of Tree 2 can be combined to construct Tree 3. This is the idea behind the BM method.



not sampled, we still have some information its subtrees. Intuitively, Tree 3 should be more probable than a tree which had none of its bipartitions sampled.

The procedure described above is the fundamental idea behind the BM method. The problem is how to deal with all bipartitions that were not sampled in the MCMC simulation. We suggest to assign all of those bipartitions that were not sampled with some small common frequency, say ϵ, which should be at least less than as though if the bipartitions were sampled once. This will be discussed in more detail in section 5.1.

Now, suppose we run an MCMC simulation for a data set with n taxa, and the bipartitions $b_1, ..., b_k$ are sampled with respective frequencies $s_1, ..., s_k$; we also assign a common frequency ϵ to all those bipartitions that were not sampled. Suppose tree τ_i is constructed by combining bipartitions $b^1_{\tau_i}, ..., b^{(n-1)}_{\tau_i}$, with respective sample frequencies $s^1_{\tau_i}, ..., s^{(n-1)}_{\tau_i}$, also suppose that we order the bipartitions so that the first $m \leq n-1$ bipartitions of τ_i were sampled in the MCMC simulation. The bipartition multiplication score S_{BM} of τ_i is then

$$S_{BM}(\tau_i) = \prod_{j=1}^{m} s^j_{\tau_i} \cdot \epsilon^{n-1-m}. \tag{4}$$

As mentioned earlier, most of the time all possible bipartitions will not be sampled, hence all trees will not have a unique score. The trees will be grouped into classes depending on what bipartitions they consist of and how many non-sampled bipartitons there are required to form a full tree. The number of classes will increase at a rate depending on the number of sampled bipartitions. No explicit formula can be given for how many classes there will be since it depends on which bipartitions that were actually sampled. All the trees that consists only of bipartitions that were not sampled will be grouped together and have the same score. The SCM-method (section 3.5) will try to deal with this problem.

3.3 The Weighted Independent Binary (WIB) Method

The WIB method was first introduced in the paper Bayesian Supertrees, Ronquist et al. [13]. Rather than focusing on all sampled bipartitions, we pick a reference tree and only focus on the bipartitions of that particular tree. The tree with highest sample frequency from the MCMC simulation is often used as a reference, but any sampled tree can be used. Based upon the reference tree, the whole tree space will be partitioned into several classes. Each tree is then assigned a score, based upon which class it belongs to.

For n taxa it requires $n-2$ bipartitions to fully describe the tree (we ignore the trivial bipartition here, i.e. the one consisting of only 1's). For each bipartition, b_i, we will compute a *WIB factor* $r_i > 0$. The WIB-factors will have the property that a tree consistent with bipartition b_i is r_i times more likely than a tree inconsistent with b_i. The WIB factors will also work multiplicatively. The distance between any tree and the reference tree will be measured based upon how many bipartitions they have in common.

For n taxa the tree space will be partitioned into 2^{n-2} classes. This is quite easy to understand if we look at a tree with 4 taxa, hence consisting of 2 bipartitions, say b_1 and b_2. The classes will then be those trees consistent with b_1 and b_2 simultaneously; b_1 but not b_2; b_2 but not b_1 and those consistent with neither b_1 nor b_2. The partitioning of a 4-taxa tree is summarized in Table 1.

Table 1: Partitioning of a 4-taxa tree with 2 bipartitions with respecitve WIB-factors r_1 and r_2.

Consistent with			
Partition 1	Partition 2	Predicted relative probability	Number of trees
yes	yes	$r_1 r_2$	1
yes	no	r_1	2
no	yes	r_2	2
no	no	1	10

In order to understand how to calculate the WIB factors for a general set of taxa, let us first have a look at the most basic example. Suppose we have a single WIB-factor r associated with the bipartition b. Let s be the corresponding bipartition frequency and $|T(b^+)|$ and$|T(b^-)|$ be the number of trees consistent and inconsistent with b respectively. Then we can use the bipartition odds to get the following formula:

$$\frac{s}{1-s} = \frac{r|T(b^+)|}{|T(b^-)|}, \tag{5}$$

and after some rearrangement

$$r = \frac{|T(b^-)|}{|T(b^+)|}\frac{s}{1-s}. \tag{6}$$

Hence, for a single bipartition we can easily find the corresponding WIB-factor from equation (6). In the same way, this can be further extended for a tree consisting of 4 taxa, i.e. two bipartitions. We then need to solve the system of equations (7)-(8).

$$\frac{s_1}{1-s_1} = \frac{r_1 r_2 |T(b_1^+, b_2^+)| + r_1 |T(b_1^+, b_2^-)|}{r_2 |T(b_1^-, b_2^+)| + |T(b_1^-, b_2^-)|} \tag{7}$$

$$\frac{s_2}{1-s_2} = \frac{r_1 r_2 |T(b_1^+, b_2^+)| + r_2 |T(b_1^-, b_2^+)|}{r_1 |T(b_1^+, b_2^-)| + |T(b_1^-, b_2^-)|}. \tag{8}$$

In order to generalize this system of equations for n taxa we need some simplifying notations. Let c be a matrix where each column represents a bipartition and each row c_i defines a set of trees by specifying whether or not they are consistent (1) or inconsistent(0) with a certain bipartition. Hence, $c_{ij} = 1$ when the trees in the set c_i are consistent with bipartition b_j. The c-matrix from the 4-taxa example is

$$\begin{pmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}.$$

The first row represents the set of trees consistent with b_1 and b_2, the second row the set of trees consistent with b_1 but inconsistent with b_2, and so on.

Let $N(c_i)$ be the number of trees in each set c_i (an algorithm for computing this number is shown in section 3.3.1). The system of equations to solve for k bipartitions, hence k WIB factors, is then

$$\frac{s_1}{1-s_1} = \frac{\sum_{i,c_{i1}=1} \left(N(c_i) \prod_j r_j^{c_{ij}}\right)}{\sum_{i,c_{i1}=0} \left(N(c_i) \prod_j r_j^{c_{ij}}\right)} \tag{9}$$

$$\vdots$$

$$\frac{s_k}{1-s_k} = \frac{\sum_{i,c_{ik}=1} \left(N(c_i) \prod_j r_j^{c_{ij}}\right)}{\sum_{i,c_{ik}=0} \left(N(c_i) \prod_j r_j^{c_{ij}}\right)}. \tag{10}$$

This is a non-linear system of equations, thus it needs to be solved numerically, for example by using the Newton-Raphson or the secant method. Notice that, if a bipartition is sampled in every iteration of the MCMC simulation, the bipartition odds, $s_i/(1-s_i)$, is not well-defined, i.e. division by zero. One way of dealing with this is to estimate the bipartition odds as though if one iteration from the MCMC simulation did not contain that particular bipartition. This is of course a problem but it should be of less concern as the number of iterations gets larger.

3.3.1 An algorithm for computing the number of trees consistent with a set of bipartitions.

This algorithm computes the number of rooted bifurcating trees that are simultaneously consistent with a certain combination of bipartitions.

Let B be a set of bipartitions, i.e. a matrix with each row representing a bipartition b_i. Suppose we have k bipartitions and n taxa, then B is a $k \times n$ matrix. The bipartitions must not be inconsistent with each other, in which case there are of course no tree consistent with that particular set of bipartitions. Let $T(n)$ be the number of rooted bifurcating trees for n taxa. This can be calculated as follows

$$T(n) = \prod_{i=2}^{n} (2i-3). \tag{11}$$

Another way of computing the number of rooted bifurcating trees was given in equation (1), however, equaiton (11) works better intuitively here (compare with equation (12)).

The number of rooted bifurcating trees consistent with B can be calculated using the following algorithm:

1. Construct a matrix containing all bipartitions and the whole tree as an additional row, i.e. a row of n 1's. Add k columns of all 0's. Think of the added columns as "artificial" taxa.

2. For each bipartition, search all the other bipartitions and check if it is contained, i.e. all its present taxa are also present in the other bipartition. If so, set all the taxa in the containing bipartition which are present in both bipartitions to 0 and set the first "available" artificial taxon to 1 on the row of the containing bipartition, i.e. all the k added columns must not contain more than one 1. After repeating this procedure for all bipartitions, denote the resulting matrix B^*.

3. Let $s(j)$ be the number of 1's of row j of B^*. Let $t(B)$ be the number of rooted bifurcating trees consistent with B, then $t(B)$ can be calculated by

$$t(B) = \prod_{j=1}^{k+1} \prod_{i=2}^{s(j)} (2i-3). \tag{12}$$

The following example illustrates the algorithm described above. Suppose we have 5 taxa and we want to find out how many trees are consistent with the two bipartitions $\{1,1,0,0,0\}$ and $\{1,1,1,0,0\}$ simultaneously. In the first step we construct the following matrix:

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}.$$

After performing the operations described in step 2 of the algorithm, the new matrix, B^*, will look like this:

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}.$$

Hence, $s(1) = 2$, $s(2) = 2$ and $s(3) = 3$. By inserting these values into equation (6) we get $t(B) = 3$. Hence, there are exactly 3 five-taxa trees which are simultaneously consistent with the bipartitions $\{1,1,0,0,0\}$ and $\{1,1,1,0,0\}$.

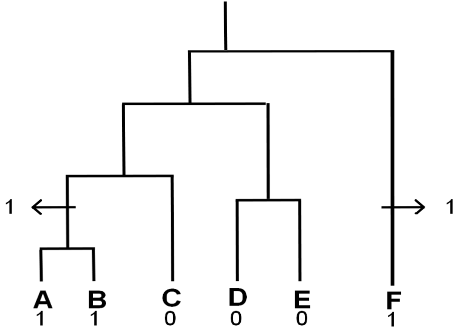
3.4 The Weighted Additive Binary (WAB) Method

The WAB method was also first introduced in the paper Bayessian Supertrees, Ronquist et al. [13]. The same fundamental idea as for the WIB method lies behind the WAB method. Based upon a reference tree, the tree space will be partitioned into several classes, a tree is then scored based upon which class it belongs to. The difference between the WIB and the WAB method lies in the way we partition the tree space. In the WIB method we used the presence/absence of a bipartition as a measure of distance, now we will introduce a new concept, the parsimony length, denoted $l(b_i, c_j)$. We will then partition the tree space based upon the parsimony length between the trees.

The WAB method will have the same number of multiplicative factors as the WIB method, but the factors will now have more than just two levels (present or absent for the WIB method). This method will result in more classes for trees consisting of 6 or more taxa, since for trees of 5 or fewer taxa the parsimony length can only be 1 or 2. The number of classes will depend on the topology of the reference tree, hence no explicit formula can be given. However, this method turns out to only be reliable for unrooted trees, we will by the end of this section see why this is the case. Nonetheless I will describe the method here since it is an interesting way of partitioning the tree space, and, as shown in [13], more efficient than the WIB method for unrooted trees. The WAB method may, with the aid of further development, lead to a more efficient way of partitioning the tree space even for rooted trees. This will be discussed in more detail in section 5.3.

The parsimony length is the least number of changes needed to perform on a tree in order to get the bipartition we want to measure. If a bipartition splits the tree into two subsets with n_0 and n_1 taxa in each bipartition, then the parsimony length of that tree can be no larger that $\min(n_0, n_1)$ on any tree. For example, suppose we have the tree depicted in Figure 4 and we want to calculate the parsimony length of that tree and the bipartition $\{1, 1, 0, 0, 0, 1\}$. At each vertex we can choose between 1 and 0, and we want to make as few changes as possible. In this case the least number of changes is 2, namely those indicated by the arrows in Figure 4.

Figure 4: Example of a calculating the parsimony length of the bipartition {1,1,0,0,0,1} on the tree of a 6 taxa seen in the figure. As indicated by the arrows, the parsimony length is 2 since two changes are needed.



In order to set up a similar system of equations as for the WIB method, but to find the WAB-factors, we need a few notations. Let $g(b_i)$ be the maximum parsimony length of the bipartition b_i on any tree. Again we want to construct a matrix, call it c^*, that partitions the tree space into several classes. For the WAB method each element $c^*_{ij} \in (0, 1, ..., (g(b_j) - 1))$. Each row in c^* should then be a unique combination of these numbers. To find which class a particular tree τ belongs to, find the numbers $c^*_{ij} = g(b_j) - l(b_j, \tau)$. Then match these to the proper row in the c^*-matrix. We also need to know the number of trees in each class, again let it be $N(c^*_i)$. For k taxa, the system of equations to solve in order to find the WAB-factors r_1,...,r_k is

$$\frac{s_1}{1-s_1} = \frac{\sum_{i,c_{i1}\geq 1} (N(c_i) \prod_j r_j^{c_{ij}})}{\sum_{i,c_{i1}=0} (N(c_i) \prod_j r_j^{c_{ij}})} \tag{13}$$

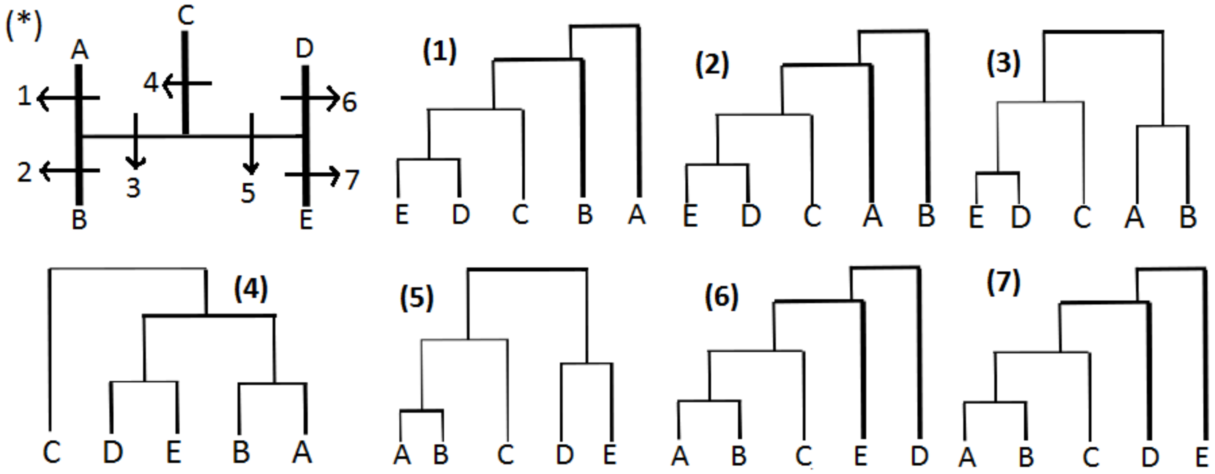
$$\vdots$$

$$\frac{s_k}{1-s_k} = \frac{\sum_{i,c_{ik}\geq 1} (N(c_i) \prod_j r_j^{c_{ij}})}{\sum_{i,c_{ik}=0} (N(c_i) \prod_j r_j^{c_{ij}})}. \tag{14}$$

The WAB method unfortunately only works well for unrooted trees. The problem with the WAB method for rooted trees is the parsimony length. This has to do with the fact that an unrooted tree requires one less bipartition than a rooted tree to be characterized. The parsimony length does not seem to distinguish where the rooting of the tree has occurred, hence each rooted version of the same unrooted tree will be put in the same class. For an unrooted tree of n taxa, there are $(2n - 3)$ ways of rooting it [2]. Hence, each class will consist of at least $(2n - 3)$ different trees. Since this is true for all trees, it also means that the best trees (i.e. trees with highest sample frequency) will be grouped together in the same class with several other trees, and, since a common score is given to all trees of the same class, the score cannot reflect the true probability distribution. I cannot give a formal proof of why the parsimony length does not distinguish where the rooting of the tree has occurred, however, by looking at an example this might become a bit more clear.

Let us consider Figure 5. Suppose that the tree labeled (1) had the highest sample frequency in an MCMC simulation and we want to use this tree as a reference tree. The tree labeled (*) is the unrooted version of tree (1), this can be rooted in 7 different ways, as indicated by the arrows labeled 1,..,7. Each rooting (1,...,7) results in the corresponding rooted tree (labeled (1),...,(7)) in Figure 5. Tree (1) consists of the bipartitions $b_1 = \{0, 0, 0, 1, 1\}$, $b_2 = \{0, 0, 1, 1, 1\}$ and $b_3 = \{0, 1, 1, 1, 1\}$. In order to find which class the trees (2),..,(7) belongs to, we want to calculate the parsimony length between those trees and the bipartitions b_1,b_2 and b_3. Note that, the parsimony length between a bipartition and a tree can only be 1 if either, (a) the bipartition itself is in the tree, (b) the inverse of the bipartition (i.e. changing all 0's to 1's and all 1's to 0's) is in the tree or, (c) the bipartition has $n - 1$ taxa present (with the total number of taxa being n). This is the case for the bipartitions b_1,b_2 and b_3 on all of the trees (2),...,(7), hence they would all be put in the same class. This is not a good partitioning, for example, tree (1) and tree (7) have no bipartitions in common, yet they would be put in the same class and have the same score, but the true likelihood of tree (7) should be much smaller than that of tree (1). With this being said, I will not continue to work with the WAB method in the analysis section, instead I will discuss it in section 5.3.

Figure 5: An unrooted tree (*) with 5 taxa (A,B,...,E) can be rooted in 7 ways, as indicated by the arrows numbered 1-7. Each rooting of (*) produces the unrooted tree (1),...,(7). The parsimony score for each bipartition of the trees (1),...,(7), on any other tree of (1),...,(7) is always 1. Hence the WAB method would put each of the trees (1),...,(7) in the same class.



3.5 The Similarity Coefficient Matrix (SCM) Method

All of the previous methods (BM, WIB, WAB) partitioned the tree space into a number of classes, then scored a tree based on which class it belongs to. The SCM method uses a different approach and will assign a score to each individual tree, in fact, the SCM method is somewhat similar to the neighbor-joining method [6][7]. The SCM method constructs a symmetric similarity coefficient matrix M. For n taxa M will be of size $n \times n$ where each row and column represents a taxon. In each spot M_{ij} there will be a coefficient determining how ”similar” taxa i and j are (M_{ii} will be empty, since it is not needed). The more similar, i.e. the closer their most recent common ancestor is between them, the greater the coefficient should be. Once M is constructed, the score is computed by multiplying the mean value of similarities for each possible subtree, including the sub-tree containing the whole tree. In other words, we average the row and column of the M matrix for each subtree and then merge the row and column together. Let us first look at an example on how to compute the score given a similarity coefficient matrix M. Suppose we have 4 taxa, say A,B,C and D, and the following SC-matrix

$$\begin{pmatrix} - & x_1 & x_2 & x_3 \\ x_1 & - & x_4 & x_5 \\ x_2 & x_4 & - & x_6 \\ x_3 & x_5 & x_6 & - \end{pmatrix}$$

Row 1 represents the first taxon, A, row 2 represents the second taxon, B, and so on. Suppose we want to score the tree $\mathcal{T} = (((A, B), C), D)$. The first subtree is

(A, B), its similarity coefficient is in row 1 and column 2 (and row 2 column 1), i.e.

$$sim(A, B) = x_1.$$

The next subtree is $((A, B), C)$, its similarity is

$$sim((A, B), C) = \frac{sim(A, C)}{2} + \frac{sim(B, C)}{2} = \frac{x_2 + x_4}{2},$$

and the corresponding merged similarity coefficient matrix (with row 1 now representing taxa A and B together, row 2 represent taxon C and row 3 represents taxon D) is

$$\begin{pmatrix} - & \frac{x_2+x_4}{2} & \frac{x_3+x_5}{2} \\ \frac{x_2+x_4}{2} & - & x_6 \\ \frac{x_3+x_5}{2} & x_6 & - \end{pmatrix}.$$

The final subtree is $(((A, B), C), D)$, i.e. the whole tree itself. Its similarity is then

$$sim((((A, B), C), D)) = \frac{sim((A, B), D)}{2} + \frac{sim(C, D)}{2} =$$

$$\frac{sim(A, D)}{4} + \frac{sim(B, D)}{4} + \frac{sim(C, D)}{2} = \frac{x_3 + x_5}{4} + \frac{x_6}{2},$$

and the corresponding merged similarity coefficient matrix is (with row 1 now representing taxa (AB) and C together and row 2 representing taxon D)

$$\begin{pmatrix} - & \frac{x_3+x_5}{4} + \frac{x_6}{2} \\ \frac{x_3+x_5}{4} + \frac{x_6}{2} & - \end{pmatrix}.$$

Thus, the SCM score for τ is

$$S_{SCM}(\tau) = x_1 \cdot \frac{x_2 + x_4}{2} \cdot (\frac{x_3 + x_5}{4} + \frac{x_6}{2}).$$

Now that we have seen how to compute the score of a tree, given the similarity coefficient matrix M, let us move on with how to calculate M, given the result of an MCMC simulation. For n taxa there are $n(n-1)/2$ similarity coefficients in the SC-matrix. Also notice that for any tree τ, $S_{SCM}(\tau)$ is some function of all similariry coefficients. Now suppose we run an MCMC simulation and the trees $\tau_1, \tau_2, ..., \tau_k$ are sampled at least once with the corresponding sample frequencies $s_1, s_2, ...s_k$. Choose the best $n(n-1)/2$ trees, if $k < n(n-1)/2$, then pick k trees. Now, we want to find those similarity coefficients $x_1, ..., x_{n(n-1)/2}$ such that the expression

$$(S_{SCM}(\tau_1) - s_1)^2 + (S_{SCM}(\tau_2) - s_2)^2 + ... + (S_{SCM}(\tau_{n(n-1)/2)}) - s_{n(n-1)/2})^2 \tag{15}$$

is minimized. Notice that the problem of minimizing (15) could also be expressed in the terms of the corresponding system of non-linear equations. In other words, we want to "fit" the matrix M to the best trees. Once we have the similarity coefficient matrix M, we can use M to compute the score of all possible trees.

The SCM method however, turned out to only work for artificial cases and not for real data, nonetheless I will include it in the analysis part (section 4.1) and further discuss it in section 5.3.

4 Analysis

In this section we will investigate how well the methods described in section 3 perform on real data. I have chosen two data sets containing 6 and 12 taxa respectively. By using equation (1) we can compute the number of possible tree topologies. For 6 taxa there are 945 unique tree topologies, thus we can check how well the methods perform on each tree. For 12 taxa there are 13749310575 unique tree topologies, thus investigating how well the methods perform on each tree is not computationally feasible for this thesis. Instead I will choose 1000 trees at random and investigate how well the methods perform on those trees.

4.1 Analysis of all possible tree topologies for a data set of 6 taxa

The data set which I used for this analysis is the "BGlobin" data set from the MrBayes package [14]. The data set consists of nucleotide sequences for 17 taxa, of which i removed 11 in order to get a data set consisting of 6 taxa. Also, I reduced the number of nucleotides to 130 for each taxa, in order to get some more variation in the simulations, otherwise the best tree is sampled some 99.9% of the time which would make comparing the methods more difficult. The MCMC simulation was performed in BEAST phylogenetics program [15] using default settings with 5,000,000 iterations and sampling from every 10th iteration, the first 10% were discarded as a burn-in period. The MCMC simulation resulted in a total of 24 unique trees and 18 unique bipartitions being sampled.

In order to evaluate the methods I calculated the marginal likelihood by using harmonic mean for each individual tree (by fixing the tree topology). This was done using default settings in the BEAST software [15]. However, this way of computing the marginal likelihood is not very reliable, see section 5.2 for further discussion. Also, I looked at the sample frequency (from the MCMC simulation) of those trees that actually were sampled at least once in order to evaluate how well the methods perform on those particular trees.

The WIB method produced 16 unique classes (as expected, see section 3.3) and the BM method produced 136 unique classes. Figure 6-8 show the plots of minus the logarithm of the normalized marginal likelihood versus minus the logarithm of the normalized scores for the WIB-, BM-, and SCM-method respectively. By normalized, I mean that each score has been divided by the sum of all scores, hence the scores from each possible method are on the same scale. The red line shows the predicted probabilities, i.e. the line which we want the methods to predict as close as possible. First of all, notice the scale on the axis of Figure 6. The WIB method underestimates the probability of the least likely trees by a huge amount, in relative terms. Also notice that the predicted probability for trees in the same class is varriyng a lot for both the WIB (Figure 6) and BM (Figure 7) method, especially those far away from the reference tree. However, one should note that the probabilities are extremely small in absolute terms. The SCM method (Figure 8) seems to fail miserably in predicting the probabilities, no clear pattern can be seen from the plot alone. In general, we can see that both the WIB (Figure 6) and the BM (Figure 7) methods are fairly consistent with the ordering of the trees, but the BM method seems to perform much better than the WIB method.

Figure 6: Minus the logarithm of the normalized marginal likelihood plotted against minus the logarithm of the normalized WIB scores for all 945 possible trees for the 6 taxa data set BGlobin. The red line shows the predicted scores, which is what we want the method to predict as close as possible.

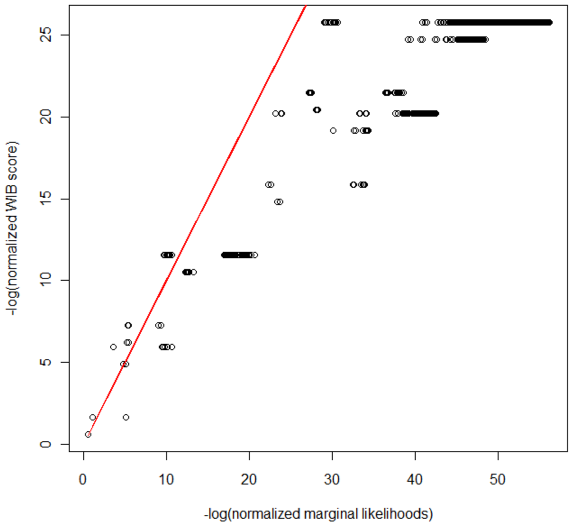


Figure 7: Minus the logarithm of the normalized marginal likelihood plotted against minus the logarithm of the normalized BM scores for all 945 possible trees for the 6 taxa data set BGlobin. The red line shows the predicted scores, which is what we want the method to predict as close as possible.

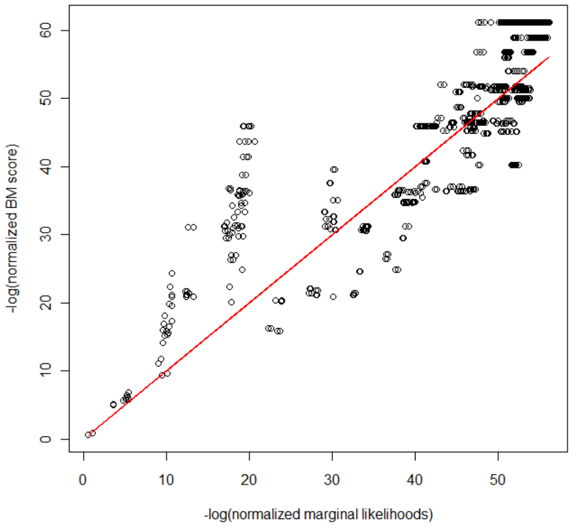
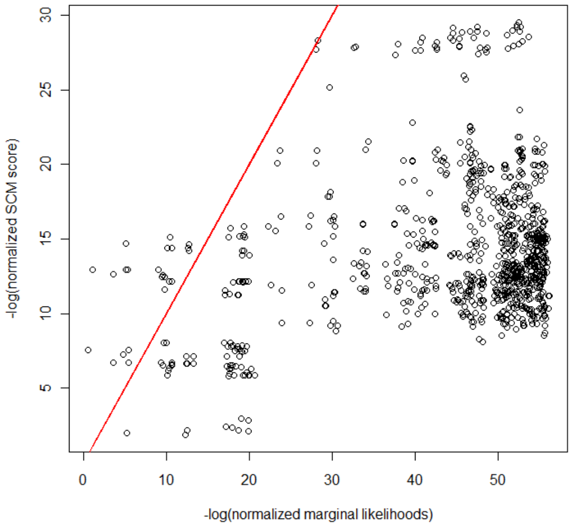


Figure 8: Minus the logarithm of the normalized marginal likelihood plotted against minus the logarithm of the normalized SCM scores for all 945 possible trees for the 6 taxa data set BGlobin. The red line shows the predicted score, which is what we want the method to predict as close as possible.



To measure how well the methods predict the true probabilities (assumed to be the normalized marginal likelihoods here) I calculated the sum of squares for each method against the marginal likelihoods. For example, for the WIB method, with normalized scores S_1^{WIB},...,S_{945}^{WIB} the sum of squares, SS_{WIB}, against the normalized marginal likelihoods S_1^{ML},...,S_{945}^{ML} is:

$$SS_{WIB} = \sum_{i=1}^{945} (S_i^{ML} - S_i^{WIB})^2 \tag{16}$$

The sums of squares were also calculated for the BM and SCM method in the same way as for the WIB method by using equation (16) but with the BM and the SCM scores respectively. Table 2 shows the results of the sum of squares for the various methods. From Table 2 we see that the BM method performs better than the WIB and that the SCM is clearly the worst performing method, as also indicated by Figure 6-8. There is no other way, at least that I know of, than the marginal likelihood (but there are several ways to calculate the marginal likelihood, see section 5.2 for more details) to find out how well the methods performs on the trees that were not sampled.

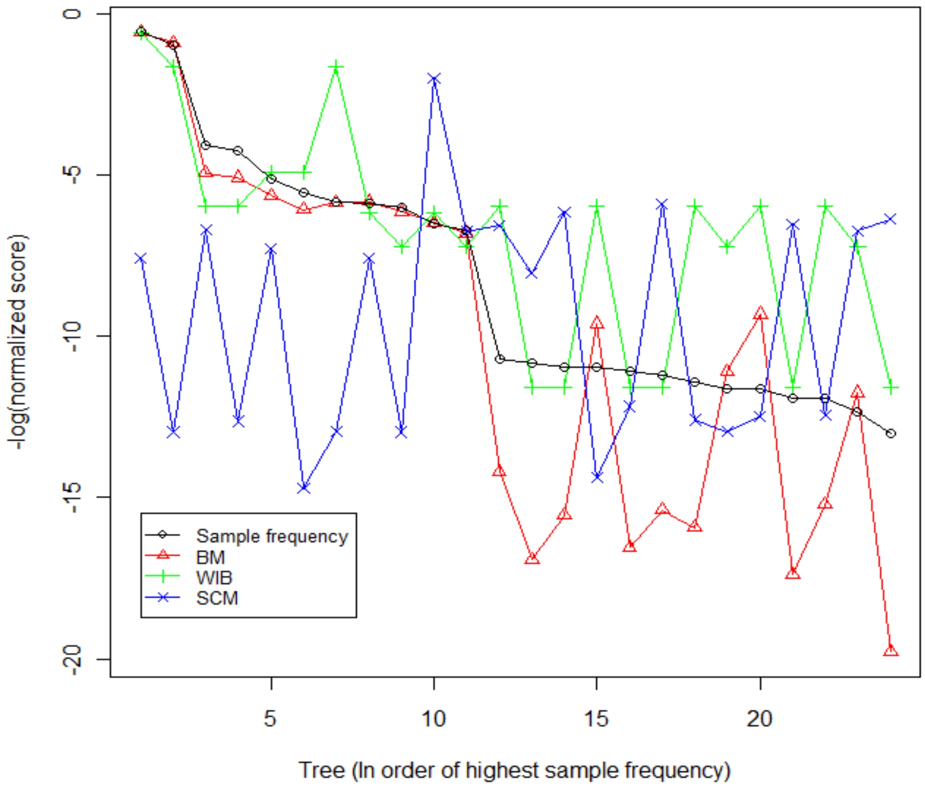
Another way of evaluating the methods is by looking at how well they perform on those 24 trees that were sampled in the MCMC simulation. In this case we do not need to rely on the marginal likelihoods, instead we can look directly at the sample frequencies of the MCMC simulation. Figure 9 shows the normalized scores for the various methods plotted together with the MCMC sample frequencies. The black circled line are the MCMC sample frequencies, i.e. the line which we want the scoring methods to predict. As discussed earlier, the sample frequencies from

Table 2: Sum of squares of various the scoring methods against the marginal likelihoods for the 6 taxa data set BGlobin. See equation (16) on how these numbers were calculated.

Method	**Sum of squares**
BM	0.0085851
WIB	0.0539842
SCM	0.5452235

the MCMC simulation are good estimates of the trees that were sampled, but, we know that every possible tree should have a probability greater than zero (even if the probability is very small for most trees), hence the sample frequency estimates are too high. However, the overestimate of the MCMC sample frequency is likely to be very small, due to the fact that the distribution of the trees is so highly skewed, hence if a method predicts the MCMC sample frequencies well, it indicates that the method performs well.

Figure 9: The normalized scores for the various methods for a dataset of 6 taxa, also the sample frequency from the MCMC simulation is shown. Notice that we are looking at minus the logarithm of the probabilities in order to get a better illustration.



I also calculated the marginal likelihoods twice, using 500,000 iterations the first time and 2 million iterations the second time and sampled from every 10th iteration in both cases. This was done in order to see how well the marginal likelihoods predicts the MCMC sample frequencies, the results are plotted in Figure 10 together with the sample frequencies.

Figure 10: The normalized marginal likelihoods for 500,000 and 2 million iterations respectively calculated for the 6 taxa data set BGlobin. Also the sample frequencies are plotted.

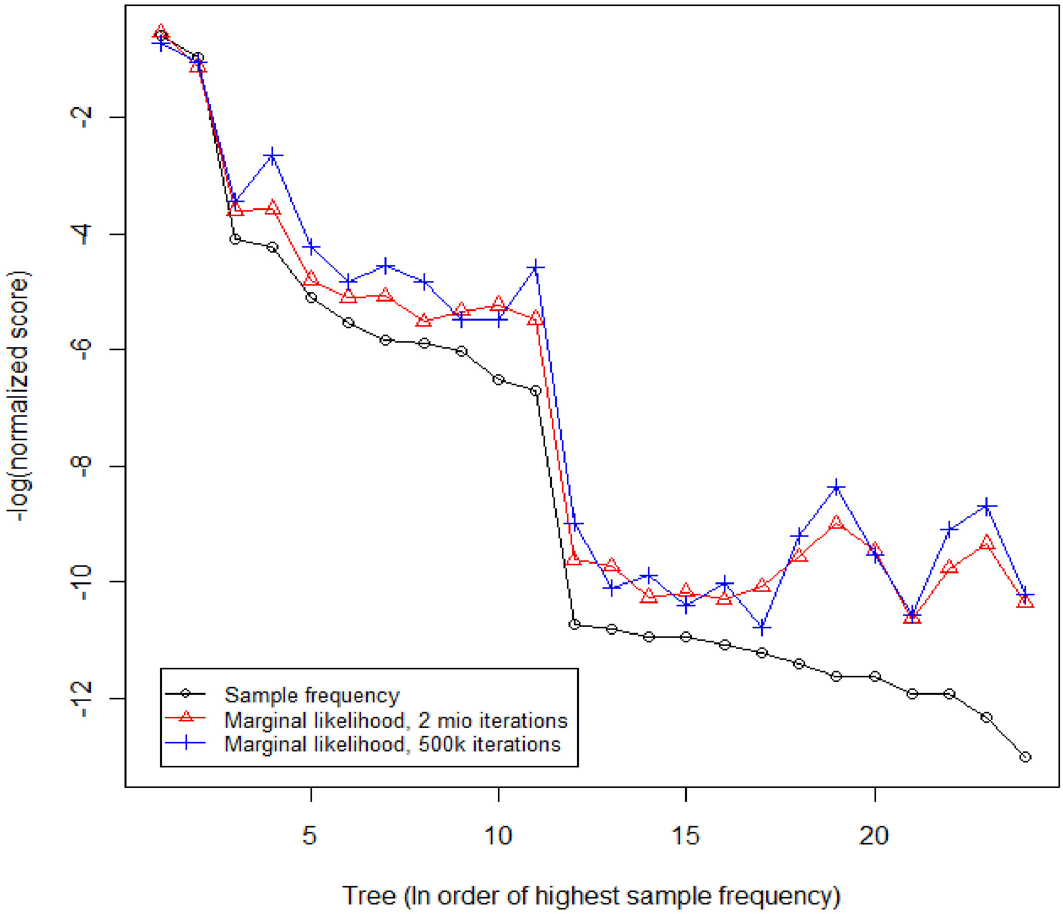


Table 3 shows the sum of squares of the methods versus the sampled frequencies (calculated the same way as in equation (16) but with sample frequencies instead of marginal likelihoods) for the 24 trees that were sampled. Here we see that the BM method outperforms the marginal likelihood using harmonic mean, again a warning that it might not be extremely reliable. The WIB methods seems to perform quite good too, but again is outperformed by the BM method.

Table 3: Sum of squares for the normalized scoring methods and the normalized marginal likelihoods against the 24 different sample frequencies for the 6 taxa data set BGlobin.

Method	**Sum of squares**
BM	0.0007392
WIB	0.0729910
SCM	0.4826861
Marginal likelihood with 2 million iterations	0.0044311
Marginal likelihood with 500k iterations	0.0103098

The actual predicted probability by the various methods for the first 10 trees is shown in Table 4. First of all, note that the WIB method has to put one tree in the best class (the class containing the reference tree), then two trees in the second to 4th best class. The second best class for the WIB method consists of tree 2 and tree 7, hence they have the same score, but tree 7 has a sample frequency far lower than tree 2. This is a weakness of the WIB method that isn't shared by the BM method, which produces results very similar to the sample frequencies. The SCM method, as noted before, clearly is not performing the way we hoped for. After analyzing the 6 taxa data set we can conclude that the WIB and the BM method are the two methods of interest for further analysis.

Table 4: Predicted probability for the various methods on the best 10 trees.

Tree	Sample Freq.	Marg. Likel. 2mio	Marg. Likel.500k	WIB	BM	SCM
1	0.5637	0.5820	0.4895	0.5588	0.5622	0.000500
2	0.3839	0.3226	0.3479	0.1963	0.4076	0.000020
3	0.0168	0.0279	0.0317	0.0026	0.0072	0.001200
4	0.0146	0.0281	0.0696	0.0026	0.0062	0.000030
5	0.0061	0.0082	0.0146	0.0074	0.0036	0.000700
6	0.0039	0.0061	0.0081	0.0074	0.0023	0.000004
7	0.0029	0.0062	0.0105	0.1963	0.0029	0.000020
8	0.0028	0.0040	0.0081	0.0021	0.0029	0.000500
9	0.0024	0.0048	0.0042	0.0007	0.0021	0.000002
10	0.0015	0.0053	0.0042	0.0021	0.0015	0.134000

4.2 Analysis of 1000 random trees for a data set of 12 taxa

In the previous section we looked at a rather small data set, but were able to see how well the methods performed on the whole tree space. The SCM method did not perform well, and will not be further investigated in this section. Instead let us focus on the WIB and the BM method to see how well they cope with a larger data set.

For the analysis of a 12 taxa data set I again used the nucleotide data set "BGlobin" from MrBayes [14] and reduced it to 12 taxa down from 17, but this time without any reduction in the length of the nucleotide sequences. 1000 trees were randomly choosen, the marginal likelihood for each tree was then calculated using 1,000,000 simulations and sampling from every 100th iteration using BEAST phylogenetics software [15] with default settings. The MCMC simulation resulted in the trees being distributed into 27 unique classes for the WIB method and 61 unique classes for the BM method. Figure 11 and 12 show minus the logarithm of the normalized marginal likelihood vs minus the logarithm of the normalized score for the BM and WIB method, the red line shows the predicted probabilities. All the methods picked a different tree as the best (i.e. highest score). I used the best one for the Marginal Likelihood as a reference tree and then normalized the scoring methods for both the BM and WIB method according to this tree, this means that we are looking at the relative probabilities and not absolute as in section 4.1. Since we are looking at the relative probilities, they may be greater than 1, hence minus the logarithm is would be smaller than 0. This is the case for some trees in the WIB method, as seen in figure 11.

Figure 11: Minus the logarithm of the normalized WIB scores plotted against minus the logarithm of the normalized marginal likelihoods for 1000 random tree on the "BGlobin" data set with 12 taxa. The red line shows the predicted scores, which is what we want the method to predict as close as possible

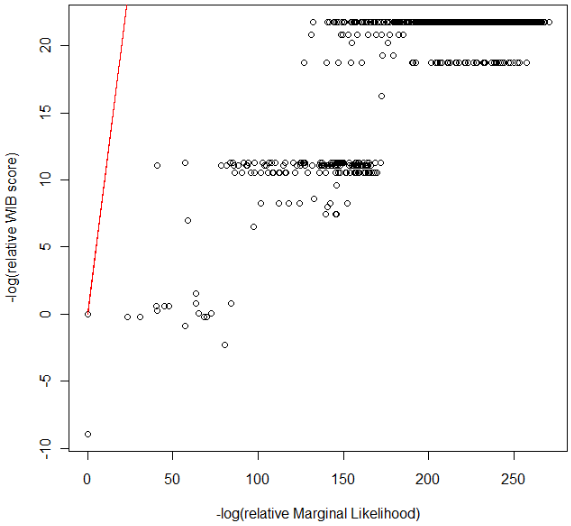
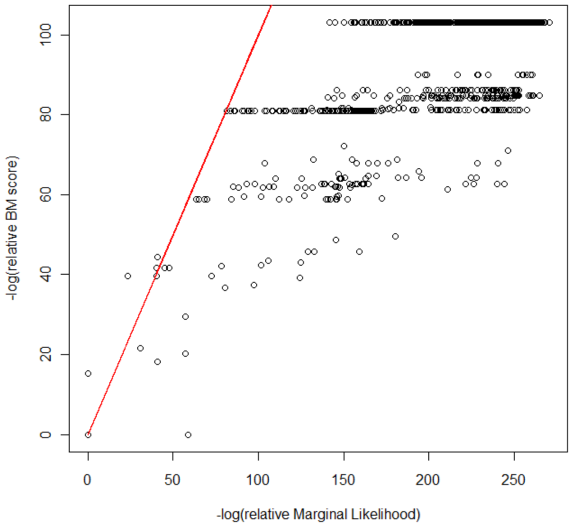


Figure 12: Minus the logarithm of the normalized BM scores plotted against minus the logarithm normalized marginal likelihoods for 1000 random tree on the "BGlobin" data set with 12 taxa. The red line shows the predicted scores, which is what we want the method to predict as close as possible



We see from the plots that the WIB method (Figure 11) again groups the trees into big classes when they are far from the reference tree whereas the BM methods (Figure 12) spreads them a bit more finely. Notice that the WIB method underestimates the probabilities by quite a lot (i.e. they are far away from the predicted probabilities as seen by the red line), the BM method also underestimates the probabilities, but not as much as the WIB but still by very much in relative terms. However, notice that we are dealing with very small numbers, so the difference in absolute terms is not very big.

The sum of squares (calculated in the same way as in equation (16)) is 2.403231 for the BM method and 61151871 for the WIB method. Again, the BM method performs better than the WIB. In general, the WIB and the BM method does not seem to be able to perform as well for the 12 taxa data set as for the 6 daxa data set. The fact that we are using the marginal likelihood calculated using harmonic mean as a measure of support could possible play a role here, since the method is flawed, see section 5.2 for further discussion.

5 Discussion

As we saw in section 4 and as explained in section 3.4, the WAB and the SCM method did not perform well, improvements for them will be discussed in section 5.3. The WIB and the BM method produced reasonble results, I will discuss them in more detail in section 5.1. The problem with evaluating the methods is that I used the marginal likelihood calculated using harmonic mean. Two quite recent papers [16] and [17] discuss two other methods that are much better for calculating the marginal likelihood, but more computational intensive and more difficult to implement. Due to time constraints I will not implement these methods for calculating the marginal likelihood, however I will briefly discuss them in section 5.2.

5.1 Evaluation of the methods

The results of the analysis clearly indicate that the BM method performs better than the WIB method. The BM method predicts the probabilities of the sampled trees (from the MCMC simulation) better than the WIB method and also, the BM method predicts the non-sampled trees better than the WIB method. The sample frequency is a good approximation of the trees that actually were sampled, but we know that it overestimates the "true" probabilities. As for the non-sampled trees we used the marginal likelihoods calculated using harmonic mean, which are not very reliable but still gives some indication of whether we are close to the "true" answer. However, one should notice a major difference in how the methods work here. The BM method uses more parameters, bipartitions in this case, than the WIB method; and also, the number of parameters used by the BM method depends on the number of bipartitions that were sampled from the MCMC simulation, whereas for the WIB method, the number of parameters is a fixed number (2^{n-2} for a data set of n taxa). In statistics, fewer parameters is preferred for the model in order to give more generality, however, as long as we do not suffer in the prediction. If we use too many parameters, the model will predict the data well, but it will not capture what lies behind the data (think of having n observations in the plane, then we can always fit a polynomial of degree n-1 to get a perfect match; the model will be excellent in predicting the past but will have little to say about the future). Hence, the question is, does the strength in the BM method lie in the fact that it uses so many parameters? And if so, how does this effect its usefulness in a situation where we want to use the BM method in order to incorporate the result of a previous MCMC simulation into a new analysis, which was the object of this thesis? These questions needs to be further investigated, but it will not be done in this thesis. Instead, here I will settled for claiming that the BM method predicts both the sampled the non-sampled trees of an MCMC simulation better than the WIB method.

An important flaw of the WIB method is the fact that it has to put at least two trees in each class except for the class containing the reference tree. This turned out not to produce good predictions for the 2nd and 7th best tree for the 6 taxa analysis (see Table 4). The BM method does not have this flaw, since every tree that is sampled at least once in the MCMC simulation will be put in a class of its own. Both the BM and the WIB method however have the problem of having a large class containing all of the worst trees with little or no information.

Another problem with the BM method that I haven't discussed previously is what to do with those bipartitions that weren't sampled. In section 3.2 I simply mentioned that you should put this value to some small ϵ. In the 6 taxa analysis performed in section 4.1 I put $\epsilon = 0.1/\#iterations$ and this seemed to produce reasonable results, see Figure 7. In the 12 taxa analysis I had much fewer iterations so I lowered the value to $\epsilon = 0.004/\#iterations$, in order to keep it at roughly the same rate as in the 6 taxa case. However, here it seems as if this value was too high, i.e. ϵ should have a smaller value (see Figure 12). But then again, we cannot know the marginal likelihoods in advance and adjust after them for a real situation. My suggestion is to keep ϵ low, in general, the fewer iterations the lower ϵ should be, and of course it should always be lower than 1 over total number of iterations. It is important to notice that the impact of choosing ϵ will only effect the very worst trees, whose probability is already very low.

5.2 Marginal Likelihood as a measure of support

In section 4, we used the marginal likelihoods computed using harmonic mean as a measure of support for the non-sampled trees from the MCMC simulation. As we clearly saw in Table 4, there are some issues using this method. It does not seem to produce results that are as accurate as we would like them to be. The time it takes to calculate the marginal likelihood is fairly large, and depends on the number of iterations as well as how often one chooses to sample from them. Due to time constraints, and the fact that I had to calculate these numbers for 945 and 1000 trees respectively I settled for the given numbers in section 4 (the time it took to calculate the marginal likelihoods used in section 4 was about a week in total). However, in order to see how the impact of iteration length and sample frequency effects the results, I picked one random tree and calculated the marginal likelihood with different size on the iteration length and different sample frequencies. The marginal likelihood was then calculated 10 times for the tree, with the sample frequency and iteration length fixed, using MCMC simulation in BEAST phylogenetic program [15] with default settings and fixed tree topology. The data is the same nucleotide sequence data as in section 4.1, the results are presented in Table 5.

Table 5: Testing the impact of iteration length and sample frequency for a randomly chosen tree from the 6 taxa dataset ”BGlobin” (same as in section 4.1). The marginal likelihood was calculated 10 times for the tree for each combination of iteration number and sample frequency.

iterations	**Sample freq.**	**mean**	**std. deviation**
0.5 million	1/10	-550.2118	0.45580283
0.5 million	1/100	-550.2374	0.35040478
0.5 million	1/1000	-550.3483	0.56835361
5 million	1/10	-550.1172	0.14279995
5 million	1/100	-550.1011	0.18255961
5 million	1/1000	-550.1615	0.23243634
15 million	1/10	-550.1338	0.07072064
15 million	1/100	-550.1513	0.09246035
15 million	1/1000	-550.1980	0.12992859

In Table 5 we see that more iterations are preferred and the marginal likelihood approximation does not seem to stabilize, even for 15 million iterations. Also, it is preferred to run a longer chain and sample less frequently than running a shorter chain and sample more frequently but still end up with the same number of samples. As indicated by the standard deviation, there is quite some difference between the different simulations of the same fixed settings. This will impact the results of an analysis when using marginal likelihoods as a measure of support. Of course, this is by no means any rigorous analysis but we clearly see that one of the issues from using the harmonic mean is that there is some variation in the results unless we run a very long chain. However, this is not the only problem, in Xie et. el [17], they give a proof that the expected value of the harmonic mean is greater than the true marginal likelihood, hence we should be very careful with using this method to evaluate the methods.

Two quite recent papers (from 2006 and 2010), Lartillot et. al. [16] and Xie et. al.[17] discusses two other methods for calculating the marginal likelihood. In Lartillot et. al. [16] they use something called *thermodynamic integration* and in Xie et. al. [17] they also compare thermodynamic integration to yet another method called *stepping stone*. I will not provide the details here, however one should notice that the thermodynamic integration method is not straightforward. It is "theoretically quite involved, requires additional code-writing for sampling along paths in the space of distributions and furthermore is computationally more intensive" [16]. In fact, it is "considerably more computationally costly than harmonic mean" [17]. The stepping stone method is however slightly less computational intensive and involves some elements of the thermodynamical integration method. It would certainly be interesting to compare the methods discussed in section 3 using these newer and more accurate methods. As of this date, there is no available software that can easily calculate these values and due to time constraints I will not be able to set up such a script.

5.3 Suggestions for future improvements

The general idea behind the WIB method was to use some kind of measure to partition the tree space, in this case it was whether or not a bipartition was consistent with a tree. The WAB method took the idea one step further by introducing the parsimony length. The parsimony length however turned out only to work for unrooted trees. This naturally leads to one question: Is there some other measure that can be used for rooted trees in order to partition the tree space more finely than the WIB method and get a more accurate prediction on the tree probabilities? Unfortunately I have not found such a measure, but it would be very interesting to investigate this further. My general feeling is that it should be possible to find such a measure. This would lead to a natural improvement of the WIB method for rooted trees, in the same way as the WAB method is a natural improvement of the WIB method for unrooted trees.

As for the SCM method, it has not been evaluated in any paper previously. It however turned out to fail miserably in predicting the tree probabilities. The idea however is interesting in itself. The major advantage of this method, if it would have worked, is that it does not group up trees into classes. During my time working with this method I found out that when the tree scores were created by using the SCM method, I was able to return exactly those similarity coefficients that created the scores. This means that, at least in some cases the method does actually work. However, for real data, I was not able to make it work any of the dozen or so cases which I tried. However, the idea behind the method seems plausible and perhaps with some further investigation on calculating the score or possibly expanding the SC matrix it could be possible to make it work.

For the BM method, it could be interesting to further evaluate the impact of choosing ϵ, i.e. the frequency of the non-sampled trees. However this is most easily done in a situation where we can calculate the marginal likelihoods accurately, since for very unlikely tree these values are so small so the harmonic mean method seems most unreliable. With the aid of the stepping stone method or possibly the thermodynamic integration method for calculating marginal likelihood it will be possible to accurately test the impact of choosing ϵ and find a more deterministic way of telling how to chose ϵ.

5.4 Conclusions

What method should we then pick when facing the task of incorporating the knowledge of a previous study into a new one? Based upon the results of the analysis, the choice of method between the four ones (BM, WIB, WAB, SCM) discussed in this thesis obviously stands between the WIB and the BM method. The WAB method is superior to the WIB method for unrooted trees but is not working for rooted trees [13]. The SCM method has some fundamental flaws and without further improvements, it is not suitable for the task at hand. It's hard to tell whether the improved accuracy of the BM method makes up for the fact that it uses more parameters. The risk of choosing a model with too many parameters is that we end up explaining the data rather than what causes the data and fails to take into account the random variations behind the data. The WIB method in this case is a bit more conservative in the number of parameters, but lacks in prediction, especially for trees far away from the reference tree (see Figure 6). The BM method captures surprisingly lots of information, even far away from the best tree (see Figure 7). Due to the fundamental differences in the way the BM and WIB methods are defined, it is not possible to claim one method superior to the other. Ultimately, it is up to the researcher to decide if the improved accuracy of the BM method makes up for the increased number of parameters, and whether or not it implies loss of generality.

6 Acknowledgements

I would like to take this opportunity to thank my supervisor Sebastian Höhna for all the support, useful meetings and interesting discussions.

References

[1] Charles Semple and Mike Steel. A supertree method for rooted trees. Discrete Applied Mathematics: 147-158 (2000).

[2] Joseph Felsenstein. The number of evolutionary trees. Syst. Zool. 27:2733 (1978).

[3] John P. Huelsenbeck, Fredrik Ronquist, Rasmus Nielsen and Jonathan P. Bollback. Bayesian inference of Phylogeny and its impact on evolutionary biology. Science 14: Vol. 294 no. 5550 pp. 2310-2314 (December 2001)

[4] M. Holder and P.O. Lewis. Phylogeny estimation: Traditional and Bayesian approaches, Nature Reviews Genetics 4, 275-284 (April 2003).

[5] Michael E. Alfaro and Mark T. Holder. The posterior and the prior in Bayesian Phylogenetics. Annual Review of Ecology, Evolution, and Systematics Vol. 37: 19-42 (December 2006)

[6] Saitou, N and Nei, M. The neighbor-joining method: a new method for reconstructing phylogenetic trees. Mol. Biol. Evol. 4, 406-425 (1987).

[7] Studier, J.A. and Keppler, K.J. A note on neighbor-joining algorithm of Saitou and Nei. Mol. Biol. Evol. 5, 729-731 (1988).

[8] Farms, J.S. Methods for computing Wagner trees. Syst. Zool. 19, 83-92(1970).

[9] Fittch, W.M. Toward defining the course of evolution: minimal change of a specific tree topology. Syst. Zool. 20, 406-416 (1971).

[10] Kluge, A.G. and Farris, J.S. Quantitative phyletics and the evolution of anurans. Syst Zool. 18, 1-32 (1969).

[11] Whelan, S., Lio, P and Goldman, N. Molecular phylogenetics: state-of-art methods for looking into the past. Trens Genet. 17, 262-272 (2001)

[12] Yang, Ziheng. Bayesian inference in molecular phylogenetics, 2005

[13] Fredrik Ronquist, John P. Huelsenbeck and Tom Britton. Phylogenetic supertrees: combining information to reveal the tree of life, chapter 9: Bayessian supertrees. pp 193-225 Computational Biology, volume 3. (2004)

[14] MrBayes:Bayesian Inference of Phylogeny, http://mrbayes.csit.fsu.edu/

[15] BEAST phylogenetics, http://beast.bio.ed.ac.uk/

[16] Lartillot, N and Philippe, H. Computing Bayes Factors Using Thermodynamic Integration, Systematic Biology, Volume55, Issue2, Pp. 195-207 (2006)

[17] Xie, W; Lewis, P O; Fan, Y; Kuo, L; Chen, M-H. Improving Marginal Likelihood Estimation for Bayesian Phylogenetic Model Selection. Systematic Biology Volume60, Issue2 Pp. 150-160 (2011)

[18] K. M. Picket and C.P. Randle Strange bayes indeed: uniform topological priors imply non-uniform clade priors.Molecular Phylogenetics and Evolution Volume 34, Issue 1, January 2005, Pages 203-211

[19] K. M. Picket and C.P. Randle. Are nonuniform clade priors important in Bayessian phylogenetic analysis? A response to Bradley et al. Systematic Biology, vol. 55, No. 1, 147-151, (2006)

[20] Felsenstein, J. Evolutionary trees from DNA sequences: a maximum likelihood approach. J. Mol: Evol. 17, 368-376 (1981)

[21] J.P. Huelsenbeck, B Lagret, R.E. Miller, F. Ronquist. Potential Applications and Pitfalls of Bayesian Inference of Phylogeny. Systematic Biology, 51:5, 673-688 (2002)

[22] W.K. Hastings, Monte Carlo sampling methods using Markov chains and their applications. Biometrika 57:97-109 (1970).