JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 17, Number 3, 2010

Pp. 337–354
DOI: 10.1089/cmb.2009.0179

How Many Bootstrap Replicates Are Necessary?

NICHOLAS D. PATTENGALE,[1] MASOUD ALIPOUR,[1] OLAF R.P. BININDA-EMONDS,[3]
BERNARD M.E. MORET,[2,4] and ALEXANDROS STAMATAKIS[5]

ABSTRACT

Phylogenetic bootstrapping (BS) is a standard technique for inferring confidence values on phylogenetic trees that is based on reconstructing many trees from minor variations of the input data, trees called replicates. BS is used with all phylogenetic reconstruction approaches, but we focus here on one of the most popular, maximum likelihood (ML). Because ML inference is so computationally demanding, it has proved too expensive to date to assess the impact of the number of replicates used in BS on the relative accuracy of the support values. For the same reason, a rather small number (typically 100) of BS replicates are computed in real-world studies. Stamatakis et al. recently introduced a BS algorithm that is 1 to 2 orders of magnitude faster than previous techniques, while yielding qualitatively comparable support values, making an experimental study possible. In this article, we propose *stopping criteria*—that is, thresholds computed at runtime to determine when enough replicates have been generated—and we report on the first large-scale experimental study to assess the effect of the number of replicates on the quality of support values, including the performance of our proposed criteria. We run our tests on 17 diverse real-world DNA—single-gene as well as multi-gene—datasets, which include 125–2,554 taxa. We find that our stopping criteria typically stop computations after 100–500 replicates (although the most conservative criterion may continue for several thousand replicates) while producing support values that correlate at better than 99.5% with the reference values on the best ML trees. Significantly, we also find that the stopping criteria can recommend very different numbers of replicates for different datasets of comparable sizes. Our results are thus twofold: (i) they give the first experimental assessment of the effect of the number of BS replicates on the quality of support values returned through BS, and (ii) they validate our proposals for stopping criteria. Practitioners will no longer have to enter a guess nor worry about the quality of support values; moreover, with most counts of replicates in the 100–500 range, robust BS under ML inference becomes computationally practical for most datasets. The complete test suite is available at http://lcbb.epfl.ch/BS.tar.bz2, and BS with our stopping criteria is included in the latest release of RAxML v7.2.5, available at http://wwwkramer.in.tum.de/exelixis/software.html.

[1]Department of Computer Science, University of New Mexico, Albuquerque, New Mexico.
[2]LCBB, EPFL, Lausanne, Switzerland.
[3]AG Systematik und Evolutionsbiologie, IBU, University of Oldenburg, Oldenburg, Germany.
[4]Swiss Institute of Bioinformatics, Lausanne, Switzerland.
[5]The Exelixis Lab, Department of Computer Science, Technische Universität München, Munich, Germany.

Key words: bootstopping, bootstrap, maximum likelihood, phylogenetic inference, stopping criterion, support value.

1. INTRODUCTION

PHYLOGENETIC TREES ARE USED TO REPRESENT the evolutionary histories of related organisms (as well, of course, as of any other units subject to evolutionary changes, from protein through genes and genomes to languages and ecologies). Most phylogenetic reconstructions for a collection of organisms take as input DNA or protein sequence alignments. These input sequences are placed at the leaves of the putative tree, and reconstruction proceeds by searching for an optimal internal branching structure for the tree. Due to the rapid and rapidly accelerating growth of sequence data in the last few years, reconstruction of trees with more than 1,000 leaves has become increasingly common, often using sequence data from many genes (so-called multi-gene or phylogenomic alignments). Such practice represents a major departure from the typical practice of the last 20 years, in which trees of 10–100 organisms were inferred from the sequence of a few simple ribosomal genes. Scaling up inference in terms of the number of organisms, the length and complexity of the sequence data, and the diameter (largest pairwise distance among the organisms) is a very challenging issue (Moret, 2007). The search space (all possible distinct branching structures) is notoriously large ($(2n-5)!! = (2n-5)\cdot(2n-7)\ldots 5\cdot 3\cdot 1$) (Edwards et al., 1963) and unstructured. Both maximum parsimony and maximum likelihood (ML) approaches are known to be NP-hard (Foulds and Graham, 1982; Roch, 2006), but both are preferred to the simpler distance methods, especially in the presence of more complex data or data with large diameters.

Significant progress has been achieved in the field of heuristic ML search algorithms with programs such as PHYML (Guindon and Gascuel, 2003), GARLI (Zwickl, 2006), Lea-Phy (Whelan, 2007), and RAxML (Stamatakis, 2006). However, there is still a major bottleneck in computing bootstrapping (BS) support values on these trees, which can require more than one month of sequential execution time for a likely insufficient number of 100 replicates (Soltis and Soltis, 2003) on a reasonably fast CPU. To date, it has proved infeasible to assess empirically the convergence properties of BS values, much less to evaluate means for dynamically deciding when a set of replicates is sufficiently large—at least on the size of trees where computing BS values is an issue.

Recently, Stamatakis et al. (2008) introduced a fast BS algorithm that yields a run time acceleration of one to two orders of magnitude compared to other current algorithms while returning qualitatively comparable support values. This improvement makes possible a large-scale experimental study on BS stopping criteria, the results of which are the topic of this article.

We propose two stopping criteria. Both split the set of replicates computed so far into two equal sets and compute statistics on the two sets. The frequency criterion (FC) is based on the observed frequencies of occurrences of distinct bipartitions; the more conservative weight criterion (WC) computes the consensus tree for each subset and scores their similarity. Both criteria can be computed efficiently, and so a stopping test can be run every so many replicates until stopping is indicated. We test these criteria and the general convergence properties of BS values on 17 diverse real-world DNA—single-gene as well as multi-gene—datasets that include 125–2,554 sequences. We find that our stopping criteria typically stop computations after 100–500 replicates (although the most conservative criterion may continue for several thousand replicates) while producing support values that correlate at better than 99.5% with the reference values on the best ML trees. Unsurprisingly, differences tend to occur mostly on branches with poor support—on branches with support values of at least 0.75, over 98% of the values returned after early stopping agree with the reference values to within 5%.

Our results show that the BS convergence speeds of empirical datasets are highly dataset-dependent, which means that bootstopping criteria can and should be deployed to determine convergence on a per alignment basis. The criteria help to conduct as many BS replicates as *necessary* for a given accuracy level and thus help to reduce the computational costs for phylogenetic analyses. Practitioners will no longer have to enter a guess nor worry about the quality of support values; moreover, with most counts of replicates in the 100–500 range, robust BS under ML inference becomes computationally practical for most datasets.

The remainder of this article is organized as follows: In Section 2, we review the BS concept and related work on stopping criteria for (mostly non-phylogenetic) BS procedures, including a brief overview of

convergence criteria for MrBayes (Ronquist and Huelsenbeck, 2003). In Section 3, we describe our family of stopping criteria. In Section 5, we describe our experimental study, give detailed results, and discuss their implications.

Over and above the preliminary version of this article (Pattengale et al., 2009), we have added the following content: The criteria are now fully implemented in the current release version 7.2.5 of RAxML, which required a large re-engineering effort. We have added an entirely new section (Section 4), which details our major undertaking to improve the runtime performance of our technique. Specifically, we discuss and assess the applicability of bipartition hashing (Pattengale et al., 2007) to bootstopping (Section 4.1) and present timing data showing the speed-up that RAxML has enjoyed via the application of these techniques (Section 4.3). These techniques have also been integrated with all other functions in RAxML that operate on bipartitions, such as a fast implementation of the Robinson-Foulds (RF) distance. We present new results on stability properties of our criteria (Section 5.3), namely that they appear tolerant to reordering BS replicates, as well as seemingly independent of the BS procedure (standard versus Stamatakis' rapid BS [Stamatakis et al., 2008]). Finally, in Section 5.5, we have also included a comparison to Hedges equation that demonstrates that the number of replicates required to achieve a certain accuracy level is indeed highly dataset-dependent

2. RELATED WORK ON BOOTSTOPPING CRITERIA

2.1. The phylogenetic bootstrap

Phylogenetic BS is a fairly straightforward application of the standard statistical (nonparametric) BS and was originally suggested by Felsenstein (1985) as a way to assign confidence values to edges/clades in phylogenetic trees. Phylogenetic BS proceeds by generating perturbed BS alignments, which are assembled by randomly drawing alignment columns from the original input alignment with replacement. The number of columns in the BS alignment is identical to the number of columns in the original alignment, but the column composition is different. Then, for each BS alignment, a tree is reconstructed independently. The procedure returns a collection of tree *replicates*. The replicates can then be used either to compute consensus trees of various flavors or to draw confidence values onto a reference tree, for example, the best-scoring ML tree. Each edge/branch in such a reference tree is then assigned a confidence value equal to the number of replicates in which it appears. The question we address in this article is: How many replicates must be generated in order to yield accurate confidence values? By accurate confidence values, we mean relative accuracy of support values (the "true" support values are unknown for empirical datasets) with respect to support values obtained by a very large number ($\geq$10,000 in our experiments) of reference replicates. The extent to which the question about the appropriate number of BS replicates has been answered in other applications of the (non-phylogenetic) BS is the subject of the following subsection.

2.2 General bootstopping criteria

Most of the literature addressing (whether theoretically or empirically) the issue of ensuring a sufficient number of replicates stems from the area of general statistics or econometrics. However, they are difficult to apply to phylogenetic BS due to the significantly higher computational and theoretical complexity of the estimator (Holmes, 2003). In addition, the problem is more complex since the number of entities (bipartitions) to which support values are assigned grows during the BS procedure—that is, adding more BS replicates increases the number of unique bipartitions. This is not commonly the case for other application areas of the general BS procedure and general bootstopping criteria that have recently been proposed (Guo and Peddada, 2008).

Standard textbooks on BS such as Davidson and Hinkley (2003) and Efron and Tibshirani (1993) suggest choosing a sufficiently large number, B, of BS replicates without addressing exact bounds for B. This does not represent a problem in most cases where the BS procedure is applied to simple statistical measures such as the mean or variance of univariate statistics. Efron and Tibshirani (1993) suggest that $B = 500$ is sufficient for the general standard BS method in most case. Manly (1997) proposes a simple approach to determine B *a priori*—that is, before conducting the BS analysis, based on a worst-case scenario by approximating the standard deviation of BS statistics. The analysis in Manly (1997) concludes that a general setting of $B = 200$ provides a relatively small error margin in BS estimation. This approximation

can only be applied to standard BS procedures, based on simple, univariate statistics. However, a larger number of BS replicates is required for other applications of the BS such as the computation of confidence intervals or tests of significance. Hall (1986) proposes a general method for stopping the BS in a percentile-*t* confidence interval. In the area of econometrics, Davidson and MacKinnon (2000) propose a two-step procedure to determine *B* for BS P-values based on the most powerful test. Andrews and Buchinsky (1997, 2000, 2001) propose and evaluate a general three-step algorithm to specify *B* in the BS procedure. Andrews and Buchinsky (2002) then further extend their algorithm to BS BCA intervals.

With respect to phylogenetics, Hedges (1992) suggests a method to specify *B a priori* for a given level of significance. In the approach of Hedges (1992), a bipartition is assumed to occur in BS replicates according to a binomial distribution, with binomial parameter *p* equal to its true support. As such, it is possible (under the binomial assumption) to calculate an upper bound *B* on the number of replicates needed to achieve a specified accuracy. Figure 1 shows the estimation of *B* for three accuracy thresholds.

This approach does not take into account the number of sequences and hence the number of potential alternative tree topologies, or the number of base-pairs or distinct patterns in the alignment. However, as underlined by our experimental results, important alignment-specific properties such as the "gappyness" (percentage of gaps) of the alignment, the quality of the alignment, and the respective phylogenetic signal strength greatly influence the estimator (the tree search algorithm) and hence the stability of BS replicates. We conclude that an adaptive stopping criterion that is computed on the fly at regular intervals during the actual BS search is best suited to take into account the particularities of real-world datasets and to determine a useful trade-off between accuracy and inference time. We are convinced that such trade-offs will become increasingly important for analysis on large phylogenomic datasets under computational resource constraints, as a recent collaborative study (Hejnol et al., 2009) with biologists already required 2,000,000 CPU hours on an IBM BlueGene/L supercomputer. Therefore, we assess our approach empirically, via a large number of computational experiments on diverse real datasets.

2.3. Bayesian convergence criteria and tools

There exists some work on convergence criteria and tools for Bayesian phylogenetic analyses, most probably because the convergence of the actual search as opposed to a sufficient number of BS replicates in ML represents a more serious methodological problem for Markov chain Monte Carlo (MCMC) in general and phylogenetic MCMC searches in particular (Mossel and Vigoda, 2006; Soltis et al., 2007; Stamatakis et al., 2004). Brooks and Gelman (1998) and Gelman and Rubin (1992) provide general frameworks to determine convergence of iterative simulations, with a focus on MCMC methods. MrBayes implements convergence diagnostics for multiple Metropolis-coupled MCMC chains that use the average standard deviation in partition frequency values across independent analyses. One potential drawback is that these statistics take into account all partition frequencies and not only the important, highly supported ones. In addition, there exist tools for graphical exploration of convergence such as Are We There Yet? (AWTY) (Nylander et al., 2007) to visualize convergence rates of posterior split probabilities and branch lengths or Tracer (Rambaut and Drummond, 2004) that analyzes time-series plots of substitution model parameters. AWTY also offers bivariate plots of split frequencies for trees obtained via independent chains. Note that both AWTY and Tracer require the user to visually inspect the respective output and determine whether the

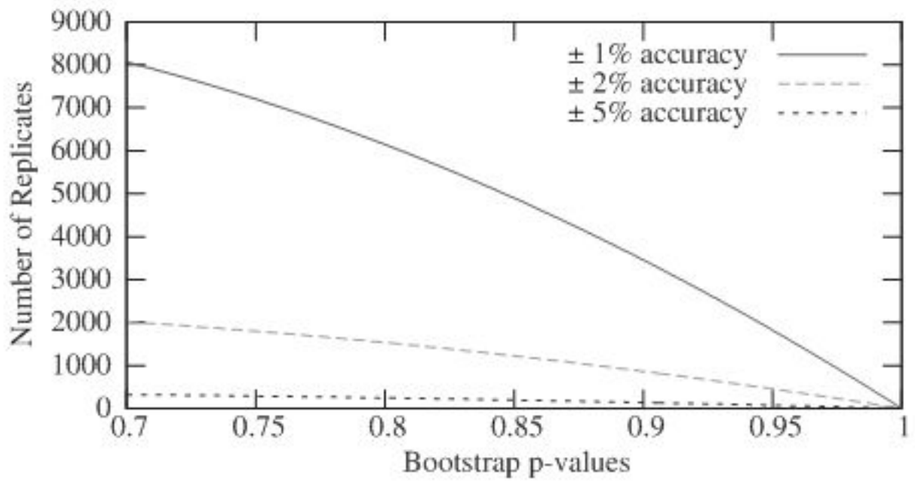


FIG. 1. Number of required replicates for various confidence intervals according to Hedges.

MCMC chains have converged. We are not aware of any computational experiments to assess the performance and accuracy of the above methods.

3 BOOTSTOPPING CRITERIA

In this section, we introduce stopping criteria for BS procedures, which we call "bootstopping" criteria. These are measures that are computed and used at run time, during the replicate inference phase, to decide when enough replicates have been computed. The FC is based upon Pearson's correlation coefficient, whereas the RF WC is based upon the (weighted) symmetric topological difference widely used in phylogenetics.

3.1. Terminology and definitions

A phylogenetic tree T is an unrooted binary tree; its leaves (also called tips) are labeled by the organism names of the input alignment, while its internal nodes represent hypothetical extinct common ancestors. Removing a branch between nodes a and b from a tree T disconnects the tree and creates two smaller trees, T_a and T_b. The trees T_a and T_b induce a *bipartition* (or split) of the set S of taxa (organism names at the leaves) of T into two disjoint taxon sets A and B ($A \cup B = S$). We denote such a bipartition as $A|B$. Thus, there exists a one-to-one correspondence between the bipartitions of S and the branches of T, so that each tree is uniquely characterized by the set of bipartitions it induces. If $|S| = n$, then any (unrooted) multifurcating phylogenetic tree for S has at most $2n - 3$. If the tree is fully bifurcating the number of bipartitions is exactly $2n - 3$, while the number of non-trivial bipartitions (i.e., splits at branches that do not lead to a tip) is $n - 3$.

The RF metric (sometimes referred to as symmetric difference) is a dissimilarity metric between two trees and counts the number of bipartitions that occur in one tree and not the other; i.e., the RF distance count is incremented by one for bipartitions that are unique to one of the two trees. The Weighted Robinson-Foulds (WRF) metric generalizes the RF metric by summing the weights of the bipartitions that contribute to the RF metric (and also, optionally, includes the sum of differences between the weights of shared bipartitions). Finally, consensus methods take a set of trees and return a single "summary" tree. The majority rule (MR) consensus method returns a tree containing only bipartitions that exist in greater than half the input trees. The extended majority rule (MRE) method (also known as *greedy consensus*) uses the MR consensus tree as a starting point and greedily adds bipartitions that occur in less than half the input trees by descending order of their frequency in the hopes (although not always possible) of obtaining a fully bifurcating (binary) tree.

3.2. Stopping criteria

The two criteria we present in the following are both based on the same underlying mechanism. Initially, the set of replicates to be tested for convergence is randomly split into two equal halves. Then, we compute statistics between the bipartition support values induced by these halves. If the difference between the splits of the replicates are small, this indicates that adding more replicates will not significantly change the bipartition composition of the replicate set. In addition, we compute the statistics not only for one but for 100 random splits of the replicate sets; i.e., we draw a sample from all possible random splits of the replicates by applying a permutation test.

3.2.1. Frequency criterion. The FC uses the bipartition frequencies of all replicates computed up to the point at which the test is conducted, for example, every 50 replicates (i.e., at 50, 100, 150, 200, … replicates). One major design goal is to devise stand-alone criteria that do not rely on a previously computed best-known ML tree for the original alignment. This is partially due to the rapid BS algorithm (and future extensions thereof) in RAxML that uses information gathered during the BS search to steer and accelerate the search for the best-scoring ML tree on the original alignment. Another important goal is to avoid a heavy dependency on the spacing (e.g., every 10, 20, or 50 replicates) of two successive steps of the test; i.e., we do not want to compute statistics that compare 20 with 30 replicates. Therefore, we have adopted a procedure that is in some sense similar to the aforementioned convergence tests for MCMC chains implemented in MrBayes. There are two main differences though: (i) we do not use the test to

determine convergence of the tree search itself, and (ii) we do not apply the test to only one single random or fixed split of the replicate tree set.

Our FC test works as follows: Assume that the test is conducted every 50 replicates (i.e., after the computation of 50, 100, 150, … BS replicates). This spacing of 50 has been chosen empirically, in order to achieve a reasonable computational trade-off between the cost of the test and the cost for computing replicates (future work will cover the development of adaptive spacing strategies). The empirical setting also fits the typical range of bootstopped tree topologies, which are 150–450 in our FC-based experiments, depending on the strength of the signal in the respective alignment. For the sake of simplicity, assume that we conduct the test for 50 replicates. At the top level of our procedure we perform a permutation test by randomly splitting up those 50 tress $p = 100$ times ($p = 100$ permutations) into disjoint sets s_1, s_2 of equal size with 25 trees each. The advantage of 100 random splits over a single random split or a fixed split into, for example, replicates with even and odd numbers, is that the curve is smoothed and depends to a far lesser degree on a by-chance favorable or unfavorable single split of the data.

In Figure 2, we depict the impact of using $p = 1$, 10, and 100 permutations on the FC and WC criteria (see Section 3.2.2) for a dataset with 500 sequences. As expected, the curve becomes smoother for larger p settings; a setting of $p = 10$ appears to be sufficient to smooth the curve and reduce the cost of the test. Though statistically more stable, the disadvantage of this approach is clearly the significantly increased computational cost of the test. Nonetheless, an initial, highly optimized at a technical level, yet algorithmically naïve implementation requires only 1 minute to conduct all 6 tests on 50, 100, … , 300 replicates on a 1,481 taxon dataset (2 minutes, 40 seconds for $p = 1000$ random splits), compared to roughly 27 hours for the computation of 300 rapid BS replicates.

For each of the aforementioned 100 random splits, we compute the support vectors v_1 for s_1 and v_2 for s_2 for all bipartitions b_{ALL} found in $s_1 \cup s_2$ (i.e., all bipartitions contained in the original 50 trees). Note that both vectors v_1, v_2 have length b_{ALL}. Given those two vectors for each permutation (random split) i, where, $i = 0, \ldots, 99$ we simply compute Pearson's correlation coefficient ρ_i on the vectors. Our procedure stops if there are at least 99 ρ_i with $\rho_i \geq 0.99$ (only one possible parameter setting). We henceforth denote the Pearson's threshold used as ρ_{FC}. A potential drawback of this method is that the support frequencies on the best-scoring tree or for all bipartitions found during the BS search might not follow a normal distribution. Nonetheless, the FC method appears to work reasonably well in practice (see Section 5). Another potential drawback is that the FC criterion is based on the bipartition frequencies of all bipartitions found. However, from a biological point of view, one is only interested in the "important" bipartitions (i.e., the bipartitions induced by the best-scoring ML tree or the bipartitions that form part of a strict, majority rule, or extended majority rule consensus tree). We address the design of a criterion that only takes into account important bipartitions in the next section. Nonetheless, the FC test can easily be extended in the future to take into

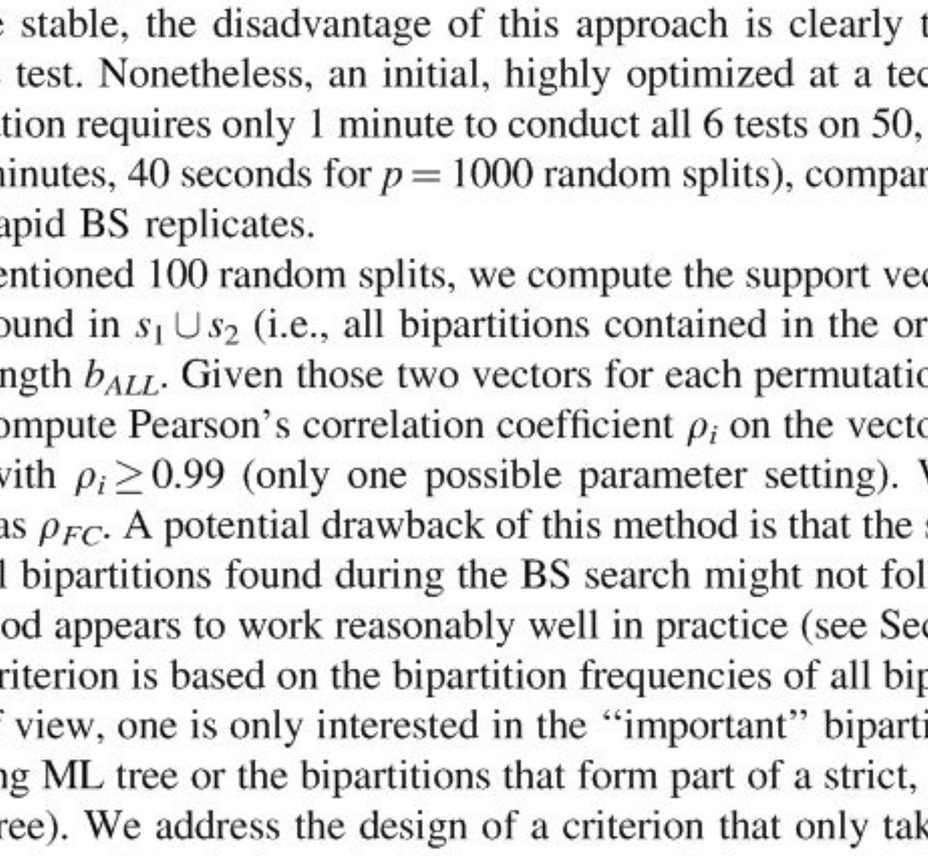


FIG. 2. Frequency criterion (FC; **top**) and weight criterion (WC; **bottom**) for various p settings on dataset 500.

account the important bipartitions by providing a user-defined best-scoring ML tree using either Pearson's correlation or, for example, the mean square error between corresponding bipartition support values.

3.2.2. Weighted Robinson-Foulds distance-based criterion. The WRF distance criterion is employed similarly to the FC criterion (i.e., every 50 trees and uses $p = 100$ permutations per test). Rather than computing a vector correlation, we compute the majority rules consensus trees for s_1 and s_2, and then assess the (dis)similarity between the two consensus trees. We then use the respective consensus trees, which only contain support values for "important" biologically relevant partitions, to calculate the WRF distance between the consensus tree $c(s_1)$ of tree set s_1 and the consensus tree $c(s_2)$ of tree set s_2.

As a distance measure and hence convergence criterion, we use the WRF. This weighted topological distance measure between consensus trees takes into account the support values and penalizes incongruent subtrees with low support to a lesser extent. When RF distances are significantly larger than their weighted counterparts (i.e., WRF), this indicates that the differences in the consensus trees are induced by subtrees with low support. When WRF $\approx$ RF, this means that the differences in the tree topologies under comparison are due to differently placed clades/subtrees with high support. From a biological perspective the WRF distance represents a more reasonable measure since systematists are typically interested in the phylogenetic position of subtrees with high support. In real-world studies, the typical empirical threshold is set to 75%; i.e., clades with a BS support of $\geq 75\%$ are usually considered to be monophyletic (Soltis and Soltis, 2003). As for the FC criterion, the WC stopping rule can be invoked with varying numbers of permutations and threshold settings. One might, for example, stop the BS procedure, if for $p = 99$ out of 100 permutations, the relative WRF between $c(s_1)$ and $c(s_2)$ is $\leq 5\%$. For reasons of consistency, we also denote the threshold parameter for WC as ρ_{WC}; a ρ_{WC} setting of 0.97 means that the BS search is stopped when p WRF distances are $\leq$ (1.0–0.97), or 3%.

4. IMPLEMENTATION CONSIDERATIONS

In the following, we address an important issue that had been completely omitted from the original article—in other words, efficiently implementing our criteria, which also entails several interesting algorithmic problems. In the following, we focus on implementation and performance of the WC criterion, which we consider to be the biologically more meaningful criterion. The algorithmic problems associated with the FC criterion are analogous.

4.1. Application of bipartition hashing

The efficient computation of our BS convergence criteria (see Section 3.2) is closely related to efficiently computing the RF metric (Robinson and Foulds, 1981) and handling bipartitions induced by a large collection of trees (as outlined in Section 3.1). The main computational challenge lies in the design of efficient methods to extract, maintain, and operate on lists that contain all non-trivial bipartitions (splits) induced by a collection of trees. Apart from computing the RF distances, such lists of bipartitions are also required for computing consensus trees (Jermiin et al., 1997) or implementing convergence assessment mechanisms for Bayesian inference programs (Nylander et al., 2007). While the theoretically optimal RF algorithm is well-described (Day, 1985), important technical details are often not considered and rarely assessed experimentally such as, for example, the choice of the hash function.

A bipartition of a tree T, $A|B$ can be represented by two presence/absence bit vectors v_A,v_B of length n, where every bit denotes the presence/absence of a taxon in the subtree to the left (T_a) and to the right (T_b) of the edge/branch that is being cut. Clearly, v_A is the bit-wise complement of v_B. Because of this property, it suffices to either store v_A or v_B. In order to ensure consistency of this choice between v_A and v_B and avoid computational overhead to check whether two bit-vectors are bit-wise complements of each other, one may chose to always store the bit-vector that contains (or does not contain) a specific taxon, for example, the first taxon in the input alignment. This is important, to ensure consistency among bipartitions extracted from two distinct trees, T_1,T_2, because a bipartition that is shared between the trees may be stored as v_A for T_1 and v_B for T_2.

Let us now consider how to efficiently extract bipartitions from an unrooted tree that is already stored in memory; i.e., we do not consider how to efficiently read in trees in the standard NEWICK format (see

http://evolution.genetics.washington.edu/phylip/newicktree.html) from file. The algorithm for efficient computation of the bipartitions at each inner branch is conceptually very similar to Felsenstein's pruning algorithm for computing the ML score on a tree (Felsenstein, 1981). It relies on a rooted view of the otherwise unrooted tree by the bipartition bit vectors, as well as on a cyclic organization of inner node pointers as used for ML computations (for details about this data structure organization, see Stamatakis and Ott, 2008). Initially, we will assign bit vectors of length n to all $2n$ - 2 nodes of the tree and initialize the bipartition vectors at the tips accordingly (i.e., just set the bit that corresponds to the respective taxon number).

Thereafter, we place a virtual root into the branch that leads to the first taxon in the input alignment and recursively compute all bipartition vectors bottom-up towards the virtual root via a depth-first traversal. Keep in mind that all inner bipartition vectors will be oriented towards the virtual root of the tree. Every time we compute the bipartition vector at an inner node that is connected to another inner node, we can directly store the bipartition in a hash table. This means that we are always storing only those bipartitions that do not contain the selected taxon and thereby ensure consistency. The complexity of this operation is $O(n^2)$, since we need to compute $n-3$ bipartition vectors, and the computation of each bipartition vector is a for loop over n bits. However, in practice, 32, 64, or even 128 (if SSE-vectorized code is used) bit vector entries can be computed in one CPU cycle, such that a more accurate approximation for the actual number of instructions is, for example, $(n \cdot (n/32)$.

Given this efficient method for extracting bipartitions from trees, we can now consider the appropriate data structure for storing these bipartitions. The usage of a hash table is straight-forward and represents an efficient choice. However, the question arises how to select a hash function for the hash key, which in our case is simply the bipartition vector. The usage of universal hash functions (Carter and Wegman, 1977) as advocated in some more theoretical papers (Sul and Williams, 2007; Sul et al., 2008; Amenta et al., 2003) is highly questionable: firstly, because the computation of a universal hash function given a bit vector of length n is slow, and secondly, universal hash functions only work well when hash keys are equally randomly distributed (Carter and Wegman, 1977), which is not very likely for hash keys that are induced by a hierarchical data structure such as a tree. Those two practical performance considerations have not been addressed in the aforementioned articles.

In contrast to this, we have experimentally assessed several highly tuned open-source hash functions that are nicely summarized at http://burtleburtle.net/bob/hash/doobs.html adopting an algorithmic engineering approach (Moret, 2002). In addition to this collection of hash functions, we also tested a phylogeny-specific hash key proposed by Pattengale et al. (2007). This method takes advantage of the tree structure and uses 32- or 64-bit integer values as hash keys instead of the entire bipartition vector. Initially, each taxon is initialized by a random unsigned 64-bit integer number. Then, the hash numbers for the bipartitions are also computed bottom up towards the virtual root by performing a bit-wise exclusive or on the respective child numbers (hash-keys). This procedure can be conveniently integrated into the depth-first traversal that is used to compute the bipartition vectors. Extensive tests on large collections of trees have revealed that this method slightly outperforms all other tested hash functions in terms of speed and generates the same amount of collisions that are resolved by chaining in the current RAxML implementation. The procedure is outlined in Figure 3.

For performing the splits of our permutation tests for FC and WC (see Sections 3.2.1 and 3.2.2), we also need to keep track of the trees that contain a bipartition that is stored in the hash table. For this, we deploy an additional presence/absence bit vector of length r, where r is the number of trees/replicates. Hence, if we add an entry to the hash table and the respective slot is already occupied, we initially need to compare the bipartition vector (or list of bipartition vectors) in that slot with the bipartition vector to be added. If it matches one of the stored bipartition vectors, we simply set the respective bit for the replicate number to 1; otherwise, we resolve by chaining.

4.3. Running time improvement

The initial implementation of our bootstopping criteria (Pattengale et al., 2009) did not perform bipartition hashing (see Section 4.1). Instead, a list of bipartitions were accumulated, such that determining whether a bipartition had been previously encountered required a (worst-case linear time) scan of the list. To this end, we have integrated and thoroughly assessed two alternative implementations that deploy bipartition hashing. The first implementation, which was written from scratch in RAxML, is based upon

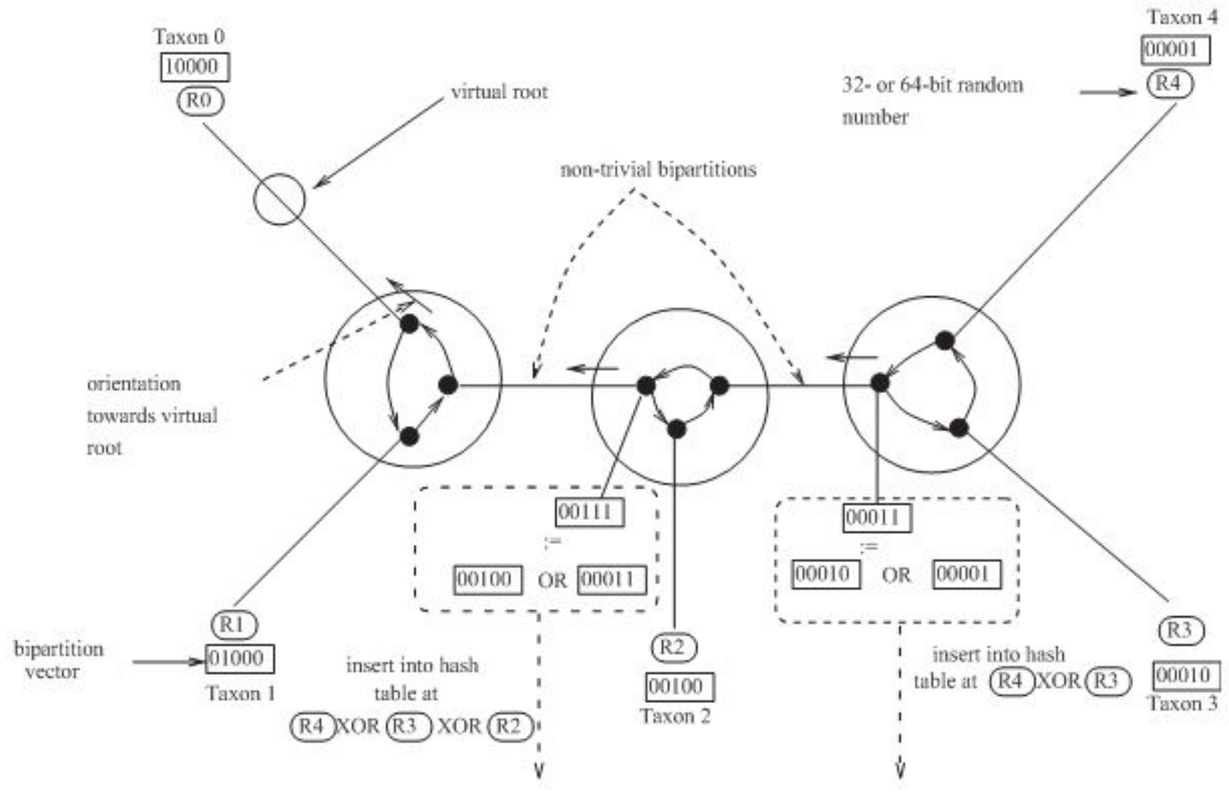


FIG. 3. Outline of the procedure to efficiently extract bipartitions and generate bipartition hash numbers on an unrooted binary tree.

hashing with chaining as described above. The second implementation invokes an appropriately adapted version of FastRF (Pattengale et al., 2007) that has been integrated into RAxML (currently unreleased). The latter implementation ignores collisions and thus is an inexact technique, but within a tolerable error. While operations on bipartitions of trees represent an interesting algorithmic problem, one must keep in mind that the respective execution times of the bootstop test are insignificant, compared to the actual replicate inference times under ML. Nonetheless, they may become a limiting factor for parallel scalability on massively parallel machines because of Amdahl's law. Within this context, speed as well as a potential future parallelization *are* important issues.

In Table 1, we report the speedup achieved by RAxML for the optimized bootstopping functions. The column *DATA* labels each data set and corresponds to its number of taxa (see Section 5.1). Column *CON-WC* indicates how many trees were processed for a setting of $\rho_{WC} = 0.03$. Finally, columns *(Pattengale et al., 2009) impl.*, *RAxML 7.2.5*, and *RAxML+FastRF* correspond to running times (in seconds) of the WC implementation for the preliminary version of this paper (Pattengale et al., 2009), the publicly available open-source version of RAxML 7.2.5, and the integration of FastRF (Pattengale et al., 2007) with RAxML, respectively.

TABLE 1. PERFORMANCE IMPROVEMENTS FOR THE BOOTSTOPPING FUNCTION IN RAXML

DATA	*CON-WC*	*(Pattengale et al., 2009) impl.*	*RAxML 7.2.5*	*RAxML + FastRF*
150	650	2.48	2.52	3.29
218	700	5.21	5.30	6.16[b]
500	400	3.70	3.42	4.28
994	300	4.84	4.25	3.97
1,481	450	38.71	34.56	37.58
2,000	600	90.20	76.78	85.69
2,554	500	64.80	52.42	52.51
4,114	100[a]	24.45	11.32	8.65
6,718	100[a]	122.18	26.92	17.00
7,764	100[a]	264.43	37.09	23.09
37,381	250[a]	dnf	1,700.29	453.86

Columns 3–5 are reported in seconds of CPU time.

[a]Bootstopping did not converge, and the indicated number of replicates reflects all that were available.

[b]Bootstopping for this sample actually converged after 550 trees (when $\rho_{WC} = 0.03$); however, we adjusted ρ_{WC} to 0.0297 (thereby requiring 700 replicates to stop) to enable a meaningfully comparable run time.

We observe that the new bipartition hashing approach has yielded a dramatic speed-up over the preliminary implementation, especially as the number of taxa grows. Further, if one is willing to sacrifice exactness (FastRF has a failure probability and is thus inexact), the third implementation is particularly desirable for datsets with huge number of taxa. For a discussion of the accuracy of the FastRF approach, see Pattengale et al. (2007).

5. EXPERIMENTAL SETUP AND RESULTS

5.1. Experimental setup

To test the performance and accuracy of FC and WC, we used 17 real-world DNA alignments containing 125–2,554 sequences. The number of distinct alignment patterns was 348–19,436. For the sake of simplicity, alignments will henceforth be referenced by the number of taxa as provided in Table 2. The experimental data spans a broad range of mostly hand-aligned sequences, including rbcL genes (500, 2,554), mammalian sequences (125, 1,288, 2,308), bacterial and archaeal sequences (714, 994, 1,481, 1,512, 1,604, 2,000), ITS sequences (354), fungal sequences (628, 1,908), and grasses (404). The 10,000 reference BS replicates on each dataset were inferred on two AMD-based Linux clusters with 128 and 144 CPUs, respectively. All result files and datasets used are available for download at http://lcbb.epfl.ch/BS.tar.bz2 We make this data available in the hope that it will be useful as a basis for further exploration of stopping criteria as well as general properties of BS.

Computational experiments were conducted as follows. For each dataset, we computed a minimum of 10,000 BS replicates using the Rapid Bootstrapping (RBS) (Stamatakis et al., 2008) algorithm implemented in RAxML. We then applied stand-alone bootstopping tests (either FC or WC) that take the set of 10,000 BS reference replicates as input and only execute the tests described in Section 3 without performing the actual BS search. Returned is a file containing the first k trees from the full set, where k is determined by the stopping criterion (FC or WC, along with appropriate parameter values). We refer to these first k trees as the "bootstopped" trees.

We then computed a number of (dis)similarity metrics between the reference replicates and the bootstopped replicates, including correlation coefficient, RF between MRE consensus trees of the two sets, and

TABLE 2. PERFORMANCE ANALYSIS OF FC ($p=99$, $\rho_{FC}=0.99$) VERSUS WC ($p=99$, $\rho_{WC}=0.97$) FOR THREE METRICS: NUMBER OF TREES TO CONVERGE, WRF BETWEEN MRE CONSENSUS TREES, AND CORRELATION COEFFICIENT

DATA	CON-FC	*CON-WC*	*WRF-FC*	*WRF-WC*	*P-FC*	*P-WC*	*# Patterns*
125	150	50	0	0	0.9997	0.9994	19,436
150	250	650	0.03	0.01	0.9984	0.9994	1,130
218	300	550	0.04	0.01	0.9977	0.9988	1,846
354	450	1200	0.03	0.01	0.9979	0.9992	348
404	250	700	0.04	0.01	0.9965	0.9988	7,429
500	200	400	0.03	0.01	0.9982	0.9991	1,193
628	250	450	0.03	0.01	0.9975	0.9987	1,033
714	200	400	0.03	0.02	0.9977	0.9989	1,231
994	150	300	0.04	0.02	0.9964	0.9974	3,363
1,288	200	400	0.03	0.02	0.9967	0.9985	1,132
1,481	300	450	0.04	0.02	0.9968	0.9979	1,241
1,512	250	350	0.03	0.02	0.9977	0.9983	1,576
1,604	250	600	0.04	0.02	0.9975	0.9990	1,275
1,908	200	400	0.03	0.02	0.9975	0.9987	1,209
2,000	300	600	0.03	0.01	0.9976	0.9989	1,251
2,308	150	200	0.03	0.02	0.9980	0.9985	1,184
2,554	200	500	0.03	0.01	0.9975	0.9991	1,232
1,102	238	482	0.03	0.01	0.9976	0.9987	2,771

Column # Patterns indicates the number of distinct column patterns in each alignment. The last line depicts the respective averages.

WRF between the MRE consensus trees of the two sets. Additionally, support values from the bootstopped and full replicate sets were drawn on the best-scoring ML tree and the resulting support values compared.

5.2. *Results for FC and WC methods*

In Table 2, we provide basic performance data for FC and WC. Column *DATA* lists the alignments, *CON-FC* the FC bootstop convergence number, and column *CON-WC* the WC bootstop convergence number. Columns *WRF-FC* and *WRF-WC* provide the WRF distance between the MRE consensus tree for the bootstopped trees and the MRE consensus tree induced by the reference replicates for FC and WC, respectively. Finally, columns *P-FC* and *P-WC* provide Pearson's correlation coefficient between support values from the bootstopped trees and the reference trees on the best-scoring ML tree for FC and WC, respectively.

We observe that WC tends to be more conservative (i.e., stops the BS search after more replicates, except for dataset 125). Dataset 125 is a particularly long phylogenomic alignment of mammals and exhibits a surprisingly low variability for the bipartitions it induces. The 10,000 reference replicates only induce a total of 195 distinct bipartitions, which is extremely low given that a single BS tree for this dataset induces $125 - 3 = 122$ nontrivial bipartitions. The WC method appears to capture this inherent stability of the BS trees sooner than FC, while the WRF to the MRE tree is 0 in both cases; i.e., the consensus trees for 50, 150, and 10,000 replicates are exactly identical. This also underlines our claim that our criteria help avoid needless computation (and needless energy expenditures, as large clusters tend to be power-hungry), in particular on such large and challenging phylogenomic datasets. Due to the general trend for WC to stop later, both *WC* metrics (P/WRF) are higher than the respective values for FC. For WC, a setting of $\rho_{WC} = 0.97$ always returns a bootstopped set with a WRF of $<2\%$ to the MRE consensus of the reference replicates. The results also clearly show that there is a significant alignment-dependent variability in the stopping numbers, as these range between 150 and 450 replicates for FC and between 50 and 1,200 for WC.

In Table 3, we provide additional metrics for the bootstopped trees. Columns μ_x and σ_x^2 provide the mean error and the mean squared error between support values induced by the $x = \{\text{FC,WC}\}$-bootstopped trees and by the reference trees on the best-scoring ML tree. Columns *SUPPLOSS-FC* and *SUPPLOSS-WC* quantify the deviations of support values in the best scoring ML tree.

In Figure 4, we graphically depict, for one dataset (1481), the convergence of FC versus WC. We plot the RF and WRF distances between the MRE consensus of the bootstopped trees and reference trees over

TABLE 3. PERFORMANCE ANALYSIS OF FC ($p = 99$, $\rho_{FC} = 0.99$) VERSUS. WC ($p = 99$, $\rho_{WC} = 0.97$) FOR THREE METRICS: MEAN ERROR, MEAN SQUARED ERROR, AND LOSS OF SUPPORT

DATA	μ-*FC*	σ^2-*FC*	μ-*WC*	σ^2-*WC*	*SUPPLOSS-FC*	*SUPPLOSS-WC*
125	0.303279	0.637530	0.483607	1.807108	0.001066	0.004672
150	1.544218	2.941922	1.074830	1.402564	0.009252	0.003605
218	1.865116	3.205062	1.297674	1.836971	0.005070	0.004674
354	1.364672	1.912598	0.886040	0.864506	0.002009	0.002835
404	2.553616	6.626178	1.384040	2.386179	0.012357	0.007170
500	1.792757	3.532503	1.239437	1.936634	0.010020	0.006841
628	2.030400	4.531876	1.398400	2.175677	0.013400	0.008408
714	2.129395	4.973412	1.424754	2.396237	0.010858	0.008833
994	2.498486	11.178353	2.068618	9.014464	0.013895	0.010575
1,288	2.477821	8.308652	1.700389	3.752257	0.013899	0.009864
1,481	1.845061	5.082219	1.496617	3.243223	0.008562	0.007287
1,512	1.762094	3.958643	1.552684	3.176317	0.008403	0.006289
1,604	1.898813	3.891073	1.229232	1.746953	0.008120	0.005721
1,908	1.961680	4.209030	1.377528	2.298479	0.009711	0.007113
2,000	1.773160	3.323105	1.184276	1.504350	0.008488	0.005020
2,308	1.951410	6.626706	1.703254	4.919317	0.010330	0.009681
2,554	2.063897	4.639194	1.248530	1.793192	0.011319	0.006370
1,102	1.871522	4.681062	1.338230	2.720849	0.009221	0.006762

The last line depicts the respective averages.

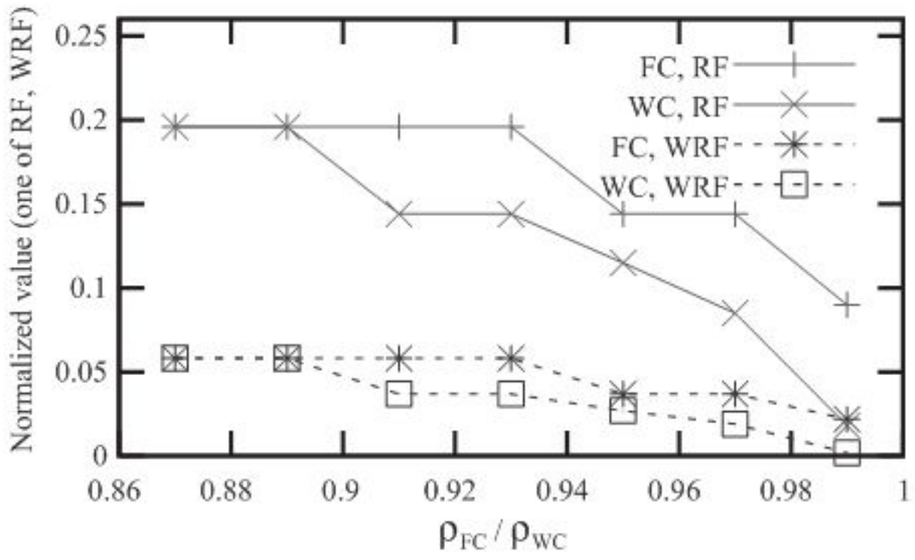


FIG. 4. Plot showing convergence of weight criterion (WC) over frequency criterion (FC) for various threshold settings (ρ_{FC} and ρ_{WC} respectively) on dataset 1418.

distinct settings (0.87, 0.88, … ,0.99) for ρ_{FC} and ρ_{WC}. For all but two datasets, we observed that WC yielded a better convergence (while it required almost 50% more replicates on average) toward replicate sets whose consensi are more congruent (i.e., have lower RF and WRF distances) with the full replicate sets, as a function of ρ. This favorable property is due to the fact that WC is exclusively based on the "important" bipartitions. Therefore, WC allows us to more precisely specify the desired degree of accuracy with respect to the biologically relevant information via an appropriate setting of ρ. As can be derived from Table 2, a setting of $\rho = 0.97$ for WC induces a WRF toward the reference dataset consensus that is $\leq 2\%$ in all cases for all of our datasets. Hence, the usage of a WC threshold will also be more meaningful, because it appears to be strongly correlated with the final WRF distance to the 10,000 reference replicates.

5.3. Robustness of criteria

To conclude that our criteria are robust, we investigated the sensitivity of our criteria to two factors: the ordering of BS replicates and the method used to create the BS replicates. In Table 4 (for WC, $\rho_{WC} = 0.03$) and Table 5 (for FC, $\rho_{FC} = 0.99$), we report on the results. For each full set of BS replicates, we generated 10 random permutations of the order of trees. We then applied our bootstop procedures again on each of the copies (permutations).

TABLE 4. DATA SUPPORTING THE ROBUSTNESS OF WC TO REORDERING REPLICATES AS WELL AS THE METHOD FOR GENERATING BOOTSTRAP REPLICATES

DATA	μ_ε	σ_ε	μ_μ	σ_μ	SBS_ε
125	7.9	2.39	3.86	1.63	
150	3.2	1.08	1.04	0.23	
218	2.9	0.54	0.90	0.14	3.0
404	3.7	0.64	1.03	0.18	
500	4.9	0.70	1.62	0.22	4.0
628	4.7	0.78	1.38	0.16	
354	3.0	0.63	0.88	0.16	5.0
714	5.5	0.92	1.96	0.23	6.0
994	7.2	0.75	2.71	0.56	
1,481	4.6	0.92	1.39	0.19	
2,000	4.9	0.70	1.23	0.13	
1,288	6.1	0.70	1.94	0.20	
1,604	5.0	0.77	1.30	0.08	
1,908	6.4	0.92	1.90	0.11	7.0
2,554	5.8	0.87	1.58	0.09	
2,308	9.0	1.61	3.12	0.33	
1,512	6.5	0.92	2.04	0.14	

The notable column is σ_ε, which indicates strong agreement across criterion applications while shuffling replicates.

TABLE 5. DATA SUPPORTING THE ROBUSTNESS OF FC TO REORDERING REPLICATES AS WELL AS THE METHOD FOR GENERATING BOOTSTRAP REPLICATES

DATA	μ_ε	σ_ε	μ_μ	σ_μ	SBS_ε
125	5.8	1.78	1.74	0.70	
150	5.5	1.75	2.98	0.49	
218	5.5	1.36	2.47	0.43	4.0
404	6.1	0.94	2.44	0.32	
500	7.4	1.36	3.45	0.42	5.0
628	6.8	1.33	2.95	0.27	
354	4.3	0.90	1.60	0.14	5.0
714	7.3	1.10	3.54	0.31	7.0
994	9.5	1.63	4.20	0.57	
1,481	6.1	1.22	2.21	0.21	
2,000	6.8	1.08	2.26	0.11	
1,288	8.4	1.43	3.41	0.19	
1,604	7.9	1.58	2.78	0.20	
1,908	9.2	1.33	3.32	0.17	7.0
2,554	9.1	1.87	3.44	0.32	
2,308	10.0	1.18	3.99	0.19	
1,512	7.3	1.10	2.90	0.20	

The notable column is σ_ε, which indicates strong agreement across criterion applications while shuffling replicates.

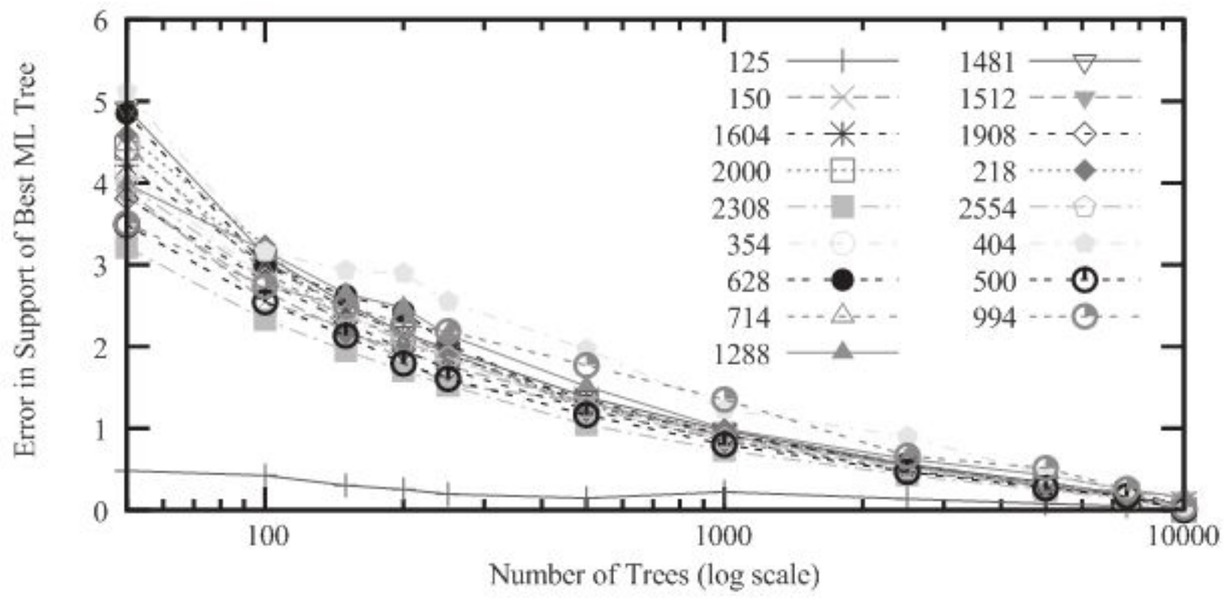
In each table, μ_ε refers to the mean of the worst support value error (for bipartitions from the best ML tree with support of $>75\%$) across 10 permutations. While the selected threshold settings for the stopping criteria yield certain accuracy errors, the standard deviation of the same quantity – σ_ε, is small, which underlines the robustness of our stopping criteria under permutations of the input replicates that we intended to demonstrate. We have also included (for completeness) the mean and standard deviation of the average error in support (again, of bipartitions from the best ML tree with support of $\geq 75\%$) as μ_μ and σ_μ.

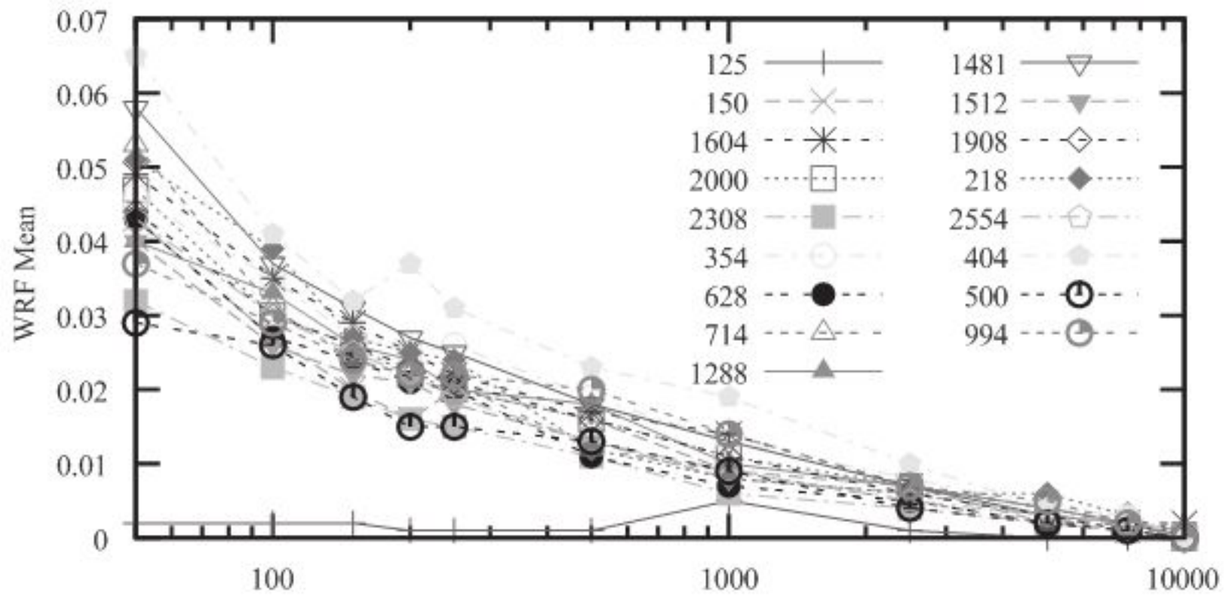
Regarding robustness to the method used to generate replicates, we generated standard BS replicates for five of our datasets and subsequently ran them through our bootstopping criteria. The results are also listed in Tables 4 and 5, under the column SBS_ε. Clearly, the error in bipartition support for the bootstopped set of standard BS sets agrees nicely with the rapid BS case.

5.4. *Convergence of data sets*

In addition to assessing our stopping criteria, we have also comprehensively assessed the inherent convergence properties of our replicate sets. Doing so has enabled us to understand a number of quantities that tend to reflect BS support and may help in the design of improved stopping criteria. We have plotted a number of (dis)similarity measures between a subset (i.e., the first m trees) and full replicate ($\geq 10,000$ trees) set. In Figure 5, we plot the RF and WRF (in the two lower plots) between the MRE consensus of each tree set restricted to the first m trees versus its respective full set of replicates ($\geq 10,000$ trees). This plot shows the differences in convergence speeds among datasets. In addition, it underlines that WRF introduces less noise than RF as replicates are added, so that WRF is a more reliable measure for convergence. An extreme example for this is dataset 354, a short (348 alignment columns) alignment of maple tree sequences from the ITS gene that is known to be hard to analyze (Grimm et al., 2006). A comparison between the development of RF and WRF over the number of trees for this alignment shows that there are many sequences with low support that are placed in different parts of the tree and essentially reflect unresolved nodes. The slight increase of distance metrics around 1,000 replicates and consecutive decrease observed for dataset 125 might be minor artifacts of the RAxML RBS algorithm.

Also in the upper part of Figure 5, we plot the development of the mean error between support values of m replicates and all replicates on the best-scoring tree. The three plots in Figure 5 clearly show that the development of WRF distances over the number of replicates is highly congruent to the development of the mean error on the best-scoring tree. Thus, WRF can be used as a criterion to determine convergence without an external reference tree. Accordingly, Figure 6 shows the development of WC and FC over





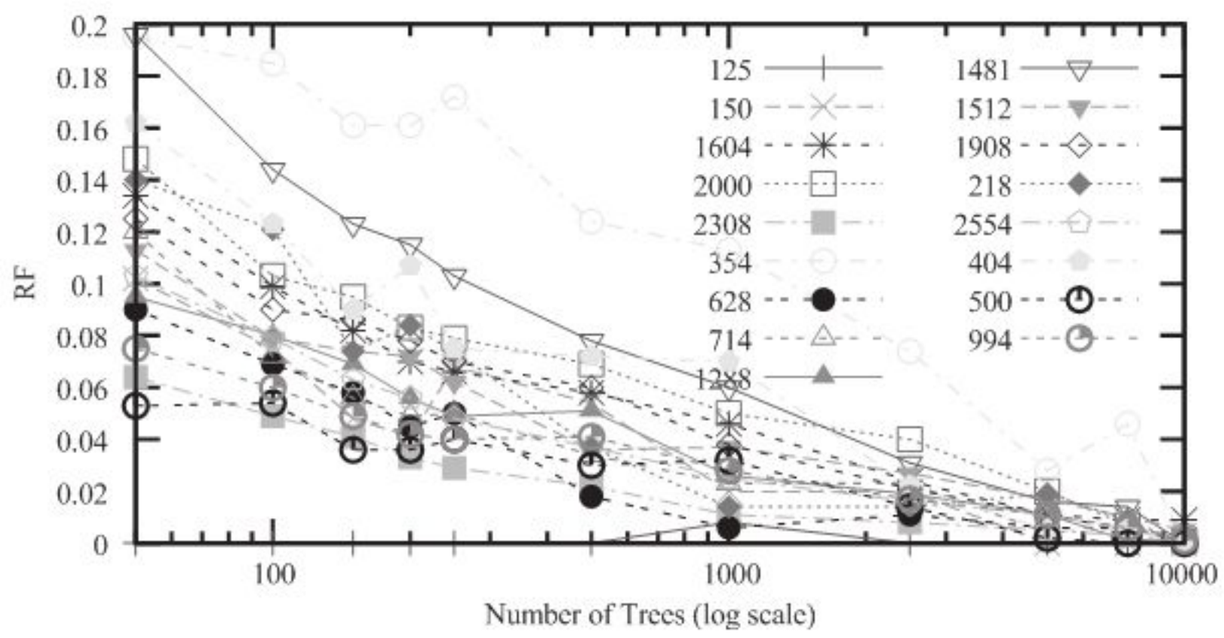


FIG. 5. Inherent convergence of replicate sets scored by error (**top**) in support of best maximum likelihood (ML) tree Weighted Robinson-Foulds (WRF) (**middle**) and Robinson-Foulds (RF) (**bottom**) distances between the first m trees and the entire (10,000 tree) set.

number of replicates, which as desired tracks nicely with Figure 5. Designing such a criterion has been a major goal of the phylogenetic community; WRF is the first good answer. Moreover, the plots can help to determine an appropriate threshold setting for ρ_{WC}, depending on the desired degree of accuracy.

Finally, in Figure 7, we plot the support values of FC/WC-bootstopped trees against the support values from the reference replicates on the best-scoring ML tree for dataset 628. The comparison clearly shows a decrease in deviations from the diagonal for the WC criterion.

5.5. Comparison to Hedges criterion

We also experimentally assess the accuracy of the formula proposed by Hedges, (1992), which is also covered briefly in Section 1 on real datasets. As already mentioned, it can be used to compute an upper

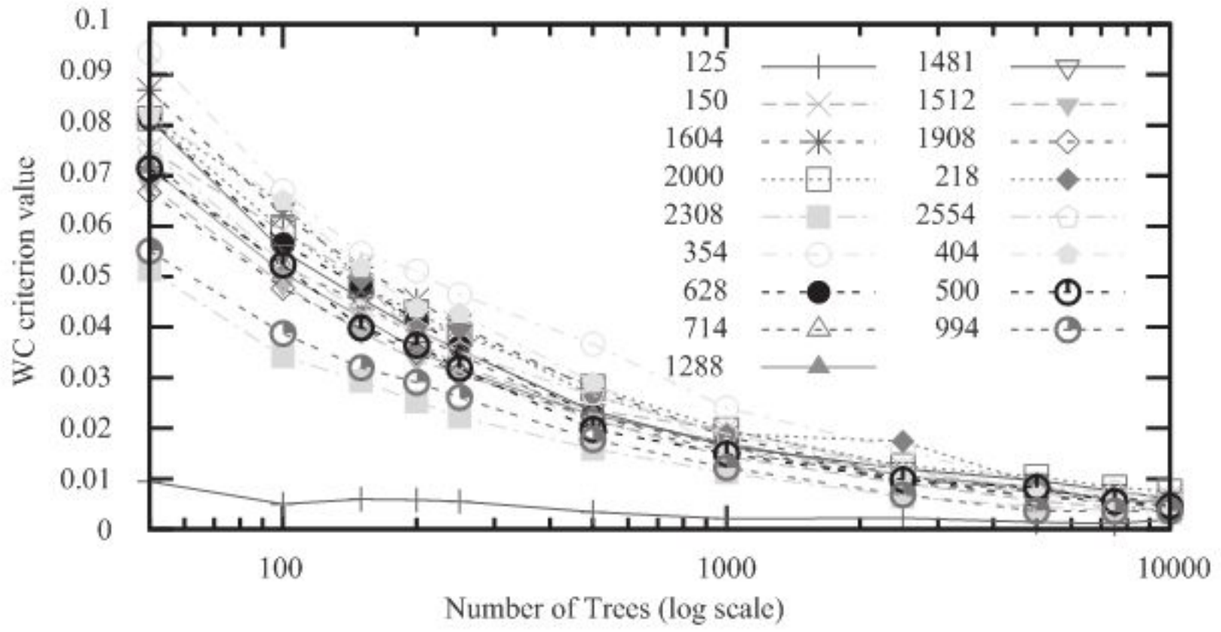


FIG. 6. Values of frequency criterion (FC) and weight criterion (WC) for tree subsets consisting of the first *m* trees.

bound for the number of replicates that are required to achieve a certain accuracy. In our experiments, we set the upper bound such that the theoretical error for support values of 75% (or greater) lies at ±2%. This upper bound is roughly 2000 replicates, as can be derived from Figure 1. We chose this threshold of accuracy because biologists typically employ this threshold when deciding whether a bipartition is supported or not. This empirical setting is also suggested by an in-depth study on real and simulated datasets (Hillis and Bull, 1993).

Therefore, we performed experiments in order to determine how many replicates were truly necessary (for our data) to meet the desired accuracy of ±2% on our datasets for all bipartitions supported by >75% by 10,000 replicates on the best-scoring ML tree. The results of our experiments (i.e., the number of replicates per datasets to achieve the desired accuracy) are as follows:

Dataset	125	1288	1481	150	1512	1604	1908	2000	218
# Replicates	950	dnf[1]	1400	1600	1500	1400	1650	1650	1200

Dataset	2308	2554	354	404	500	628	714	994
# Replicates	1550	1900	1550	1550	1700	1850	1200	1550

As such, we conclude that the estimate of Hedges (1992) provides a reasonable upper bound for the accuracy, meaning that it is not a gross overestimate. To our knowledge, this data represents the first empirical assessment of Hedges formula. Nonetheless, the number of replicates required is highly

[1]Dataset 1288 had not beat the threshold by 2000 replicates, but had by 2500.

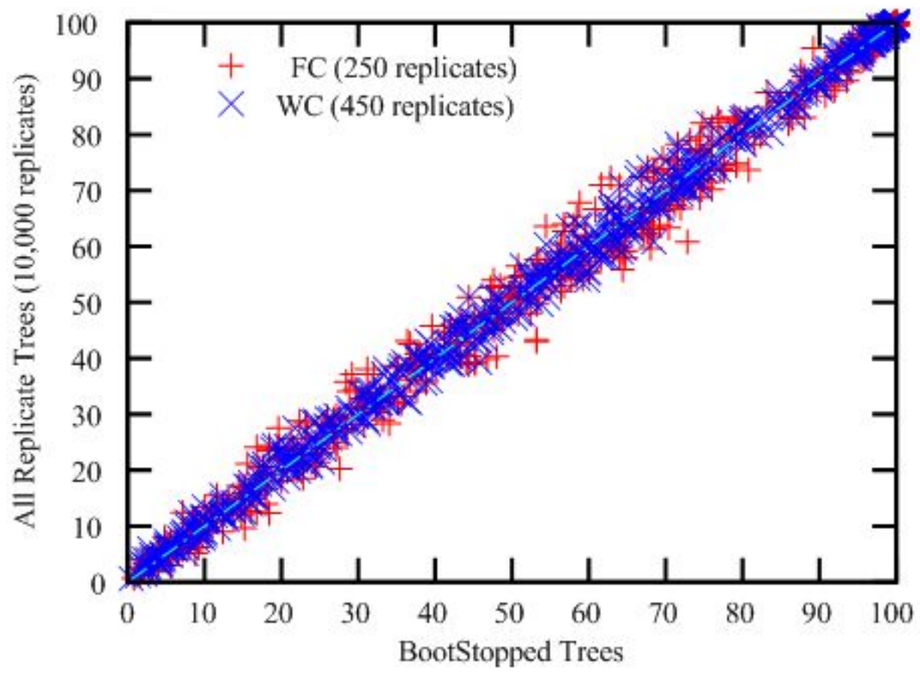


FIG. 7. Support values drawn on the best maximum likelihood (ML) tree for frequency criterion (FC; blue) and weight criterion (WC; red) versus full replicate set, for dataset 628.

dataset-dependent. Thus, given the above stopping numbers, there is a potential for computing far too many replicates and wasting 30% or more CPU hours when deploying the formula. Interestingly, dataset 1288 requires more replicates than indicated by Hedges (1992) formula to achieve the desired accuracy level.

6. CONCLUSION

We have conducted the first large-scale empirical ML-based study of the convergence properties of BS, using biological datasets that cover a wide range of input alignment sizes and a broad variety of organisms and genes. In addition, we have developed and assessed two bootstopping criteria that can be computed at run time and do not rely on externally provided reference trees to determine convergence. The criteria have been designed so as to capture a stopping point that provides sufficient accuracy for an unambiguous biological interpretation of the resulting consensus trees or best-known ML trees with support values. The correlation between bootstopped support values and support values from 10,000 reference trees exceeds 99.5% in all cases, while the relative weighted tree distance (used with the WC criterion) is smaller than the specified threshold value in all cases. We conclude that the WC criterion yields better performance and higher accuracy than FC, while it correlates very well with the mean error of support values on the best-scoring tree. We advocate the use of WC over FC because it only takes into account the BS support of "important" bipartitions which are subject to biological interpretation. We have also shown that the number of replicates required to achieve a certain level of accuracy is highly dataset-dependent for real data, so that, by using our criteria, an investigator need only compute as many replicates as necessary, thus avoiding the waste of scarce computational resources, in particular for future large-scale phylogenomic analyses. Finally, we have fully integrated the criteria into the current release of RAxML and provided a detailed description and study of implementation issues associated to the stopping functions. Our production level implementation yields speed-ups of the stopping function up to a factor of 7 on datasets with thousands of taxa.

Since the preliminary version of this paper, we have completed the full integration of the advanced hashing techniques into RAxML 7.2.5. We have parallelized (Aberer et al., submitted) the hash table operations using Pthreads and vectorize operations on bit vectors by using SSE3 instructions. Finally, we will devise ways to dynamically adapt the spacing of FC/WC criteria (which is currently fixed at 50) to the convergence speed of the BS replicates (i.e., use a more sparse spacing for the initial phase and a denser spacing for the later phase of the BS search).

ACKNOWLEDGMENTS

We would like to thank Derrick Zwickl and Bret Larget for useful discussions on this manuscript. We are also thankful to Andrew Rambaut for discussions on Tracer and AWTY. We would also like to thank the

following colleagues for providing real-world datasets: N. Poulakakis, U. Roshan, M. Gottschling, M. Göker, G. Grimm, C. Robertson, and N. Salamin. Part of this work was funded under the auspices of the Emmy Noether program by the German Science Foundation.

DISCLOSURE STATEMENT

No competing financial interests exist.

REFERENCES

Aberer, A., Pattengale, N., and Stamatakis, A. 2010. Parallel Computation of Phylogenetic Consensus Trees. Exelixis-RRDR-2010-1. TU Munich, January 2010. Submitted.

Amenta, N., Clarke, F., and St. John, K. 2003. A linear-time majority tree algorithm. *Lect. Notes Comput. Sci.* 216–227.

Andrews, D.W.K., and Buchinsky, M. 1997. *On the Number of Bootstrap Repetitions for Bootstrap Standard Errors, Confidence Intervals, and Tests. Cowles Foundation Paper 1141R.*

Andrews, D.W.K., and Buchinsky, M. 2000. A three-step method for choosing the number of bootstrap repetitions. *Econometrica* 68, 23–51.

Andrews, D.W.K., and Buchinsky, M. 2001. Evaluation of a three-step method for choosing the number of bootstrap repetitions. *J. Econometrics* 103, 345–386.

Andrews, D.W.K., and Buchinsky, M. 2002. On the number of bootstrap repetitions for BCA confidence intervals. *Econometric Theory* 18, 962–984.

Brooks, S.P., and Gelman, A. 1998. General methods for monitoring convergence of iterative simulations. *J. Comput. Graphical Statist.* 7, 434–455.

Carter, J.L., and Wegman, M.N. 1977. Universal classes of hash functions. *Proc. 9th Annu. ACM Symp. Theory Comput.* 106–112.

Davidson, A.C., and Hinkley, D.V. 2003. *Bootstrap Methods and Their Application*. Cambridge University Press, New York.

Davidson, R., and MacKinnon, J.G. 2000. Bootstrap tests: how many bootstraps? *Econometric Rev.* 19, 55–68.

Day, W.H.E. 1985. Optimal algorithms for comparing trees with labeled leaves. *J. Classif.* 2, 7–28.

Edwards, A.W.F., Cavalli-Sforza, L.L., Heywood, V.H., et al. 1963. Phenetic and phylogenetic classification. *System. Assoc. Public.* 6, 67–76.

Efron, B., and Tibshirani, R.J. 1993. *An Introduction to the Bootstrap*. Chapman & Hall, New York.

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

Felsenstein, J. 1981. Evolutionary trees from DNA sequences: a maximum likelihood approach. *J. Mol. Evol.* 17, 368–376.

Foulds, L.R., and Graham, R.L. 1982. The Steiner problem in phylogeny is NP-complete. *Adv. Appl. Math.* 3, 299.

Gelman, A., and Rubin, D.B. 1992. Inference from iterative simulation using multiple sequences. *Statist Sci.* 7, 457–511.

Grimm, G.W., Renner, S.S., Stamatakis, A., et al. 2006. A nuclear ribosomal DNA phylogeny of acer inferred with maximum likelihood, splits graphs, and motif analyses of 606 sequences. *Evol. Bioinform. Online* 2, 279–294.

Guindon, S., and Gascuel, O. 2003. A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood. *Syst. Biol.* 52, 696–704.

Guo, W., and Peddada, S. 2008. Adaptive choice of the number of bootstrap samples in large-scale multiple testing. *Statist Appl. Genet. Mol. Biol.* 7, 1.

Hall, P. 1986. On the number of bootstrap simulations required to construct a confidence interval. *Ann. Statist.* 14, 1453–1462.

Hedges, S.B. 1992. The number of replications needed for accurate estimation of the bootstrap P value in phylogenetic studies. *Mol. Biol. Evol.* 9, 366–369.

Hejnol, A., Obst, M., Stamatakis, A., et al. 2009. Assessing the root of bilaterian animals with scalable phylogenomic methods. In: *Proc. R. Soc.* B276, 4261–4270.

Hillis, D.M., and Bull, J.J. 1993. An empirical test of bootstrapping as a method for assessing confidence in phylogenetic analysis. *System. Biol.* 42, 182.

Holmes, S. 2003. Bootstrapping phylogenetic trees: theory and methods. *Statist. Sci.* 18, 241–255.

Jermiin, L.S., Olsen, G.J., Mengerson, K.L., et al. 1997. Majority-rule consensus of phylogenetic trees obtained by maximum-likelihood analysis. *Mol. Biol. Evol.* 14, 1296.

Manly, B.F.J. 1997. *Randomization, Bootstrap and Monte Carlo Methods in Biology*. CRC Press, Boca Raton, FL.

Moret, B.M.E. 2007. Large-scale phylogenetic reconstruction, 29–48. In: Brown, J.R., ed., *Comparative Genomics: Basic and Applied Research*, CRC Press/Taylor & Francis, Boca Raton, FL.

Moret, B.M.E. 2002. Towards a discipline of experimental algorithmics, 197. In: *Data Structures, Near Neighbor Searches, and Methodology: Fifth and Sixth DIMACS Implementation Challenges: Papers Related to the DIMACS Challenge on Dictionaries and Priority Queues (1995–1996) and the DIMACS Challenge on Near Neighbor Searches (1998–1999).*

Mossel, E., and Vigoda, E. 2006. Limitations of Markov chain Monte Carlo algorithms for Bayesian inference of phylogeny. *Ann. Appl. Probab.* 16, 2215–2234.

Nylander, J.A.A., Wilgenbusch, J.C., Warren, D.L., et al. 2007. AWTY (Are We There Yet?): a system for graphical exploration of MCMC convergence in Bayesian phylogenetics. *Bioinformatics* advance access August 30.

Pattengale, N.D., Gottlieb, E.J., and Moret, B.M.E. 2007. Efficiently computing the Robinson-Foulds metric. *J. Comput. Biol.* 14, 724–735.

Pattengale, N.D., Alipour, M., Bininda-Emonds, O.R.P., et al. 2009. How many bootstrap replicates are necessary? *Lect. Notes Comput. Sci.* 5541, 184–200.

Rambaut, A., and Drummond, A. 2004. *Tracer MCMC Trace Analysis Tool*, version 1.3.

Robinson, D.F., and Foulds, L.R. 1981. Comparison of phylogenetic trees. *Math. Biosci.*, 53, 131–147.

Roch, S. 2006. A short proof that phylogenetic tree reconstruction by maximum likelihood is hard. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 92–94.

Ronquist, F., and Huelsenbeck, J.P. 2003. MrBayes 3: Bayesian phylogenetic inference under mixed models. *Bioinformatics* 19, 1572–1574.

Soltis, D.E., and Soltis, P.S. 2003. Applying the bootstrap in phylogeny reconstruction. *Statist. Sci.* 18, 256–267.

Soltis, D.E., Gitzendanner, M.A., and Soltis, P.S. 2007. A 567-taxon data set for angiosperms: the challenges posed by Bayesian analyses of large data sets. *Int. J. Plant Sci.* 168, 137–157.

Stamatakis, A. 2006. RAxML-VI-HPC: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. *Bioinformatics* 22, 2688–2690.

Stamatakis, A., and Ott, M. 2008. Efficient computation of the phylogenetic likelihood function on multi-gene alignments and multi-core architectures. *Phil. Trans. R. Soc. B*, 363, 3977–3984.

Stamatakis, A., Meier, H., and Ludwig, T. 2004. New fast and accurate heuristics for inference of large phylogenetic trees. *Proc. IPDPS2004.*

Stamatakis, A., Hoover, P., and Rougemont. 2008. A rapid bootstrap algorithm for the RAxML web servers. *System Biol.* (in press).

Sul, S.J., and Williams, T.L. 2007. A randomized algorithm for comparing sets of phylogenetic trees. *Proc. 5th Asia-Pacific Bioinform. Conf.* 121.

Sul, S.J., Brammer, G., and Williams, T.L. 2008. Efficiently computing arbitrarily-sized Robinson-Foulds distance matrices. *Proc. 8th Int. Workshop Algorithms Bioinform.* 123–134.

Whelan, S. 2007. New approaches to phylogenetic tree search and their application to large numbers of protein alignments. *Syst. Biol.* 56, 727–740.

Zwickl, D. 2006. *Genetic Algorithm Approaches for the Phylogenetic Analysis of Large Biological Sequence Datasets under the Maximum Likelihood Criterion.* [Ph.D. dissertation]. University of Texas at Austin.

Address correspondence to:
Dr. Nicholas D. Pattengale
Department of Computer Science
University of New Mexico
Albuquerque, NM 87123

E-mail: nickp@cs.unm.edu