IQPNNI: Moving Fast Through Tree Space and Stopping in Time

Le Sy Vinh† *and Arndt von Haeseler**†
*Heinrich-Heine Universität Düsseldorf, Germany; and †Forschungszentrum Jülich, Germany

An efficient tree reconstruction method (IQPNNI) is introduced to reconstruct a phylogenetic tree based on DNA or amino acid sequence data. Our approach combines various fast algorithms to generate a list of potential candidate trees. The key ingredient is the definition of so-called important quartets (IQs), which allow the computation of an intermediate tree in $O(n^2)$ time for n sequences. The resulting tree is then further optimized by applying the nearest neighbor interchange (NNI) operation. Subsequently a random fraction of the sequences is deleted from the best tree found so far. The deleted sequences are then re-inserted in the smaller tree using the important quartet puzzling (IQP) algorithm. These steps are repeated several times and the best tree, with respect to the likelihood criterion, is considered as the inferred phylogenetic tree. Moreover, we suggest a rule which indicates when to stop the search. Simulations show that IQPNNI gives a slightly better accuracy than other programs tested. Moreover, we applied the approach to 218 small subunit rRNA sequences and 500 rbcL sequences. We found trees with higher likelihood compared to the results by others. A program to reconstruct DNA or amino acid based phylogenetic trees is available online (http://www.bi.uni-duesseldorf.de/software/iqpnni).

Introduction

One objective in phylogenetic analysis is the reconstruction of the evolutionary relationship of species based on their genetic information. In recent years, the list of methods for building a phylogenetic tree from modern-day species has increased steadily (Swofford et al. 1996; Felsenstein 2004). New tree search heuristics have been developed to reduce the computational burden caused by increasing amounts of sequence data (e.g., Bruno, Socci, and Halpern 2000; Gascuel 1997; Huson, Nettles, and Warnow 1999; Lemmon and Milinkovitch 2002; Schmidt and von Haeseler 2002; Guindon and Gascuel 2003; Vos 2003). In addition, Monte Carlo methods have been employed to explore the space of possible tree topologies (Larget and Simon 1999; Ronquist and Huelsenbeck 2003). Finally, attempts to parallelize tree-reconstruction programs were introduced (Olsen et al. 1994; Charleston 2001; Brauer et al. 2002; Schmidt et al. 2002, 2003). Setting aside statistical issues, the recently published PHYML method (Guindon and Gascuel 2003) seems to be faster than other existing methods.

Another attempt to reduce computation time is the reconstruction of quartet trees, which are subsequently used to puzzle an overall tree (cf., Strimmer and von Haeseler 1996; Willson 1999; Ranwez and Gascuel 2001). However, the complexity of $O(n^4)$ prohibits an application of quartet methods to data with more than approximately 100 sequences, because it is necessary to evaluate all quartet trees. We propose the *Important Quartet Puzzling Method*, IQP, which uses only $O(n^2)$ quartets to construct a tree to overcome this computational drawback. The IQP approach is implemented as a part of an algorithm which efficiently elucidates the landscape of possible optimal trees. To this end, we introduce a search strategy that alternates between the nearest neighbor interchange (NNI) method and the sequential sequence-by-sequence assembly approach based on IQP. This combined strategy reduces the risk of getting stuck in a local optimum. For illustration, two recently analyzed data sets (Guindon and Gascuel 2003) are re-analyzed. Finally, we apply an estimator that is based on the time series of sights of better trees during the tree search to estimate when it becomes unlikely that a further search for a better tree will be successful.

Key words: sequence evolution, maximum likelihood, tree reconstruction, phylogenetics, quartet tree, nearest neighbor interchange, accuracy, rbcL, record value.

E-mail: haeseler@cs.uni-duesseldorf.de.

Mol. Biol. Evol. 21(8):1565–1571. 2004
doi:10.1093/molbev/msh176
Advance Access publication May 26, 2004

The Important Quartet Concept

In the following discussion, we introduce some notations: Consider a binary, rooted tree T with one distinguished node r, the *root* (see figure 1). A node, with one adjacent branch is called a *leaf* (typically a contemporary sequence/taxon), all other nodes are called *internal nodes*. For each leaf l, we compute the *distance* to the root $d(l, r)$ as the number of branches on the path from r to l in T (in computer science parlance the *depth*). Leaves l_1, l_2 with the same distance $d(l_1, r) = d(l_2, r) \equiv d$ are said to be on *level d*. For example, in figure 1 leaves a, b are on level 3, leaves g and h are on level 4, and so on. The distance $d(\cdot, r)$ induces a natural ordering of the leaves. A set of *k-representative leaves* $S_k(T)$ of T is simply a collection of pairs $(l, d(l, r))$ that includes at most k leaves with the smallest distances from the root, where possible ties are resolved randomly. We say the tree is represented by the k-representative leaf set. For example, the k-representative leaf set for $k = 4$ is $S_4(T) = \{(a, 3), (b, 3), (g, 4), (h, 4)\}$ in the current tree (fig. 1). For $k = 5$, the set is not unique; either of the leaves $(c, 5)$, $(d, 5)$, $(e, 5)$, or $(f, 5)$ can be added to $S_4(T)$ to form $S_5(T)$.

To motivate this abstract concept, think of $S_k(T)$ as a collection of contemporary sequences that are closest to the root (with respect to the defined distance). Thus, this collection resembles the ancestral sequence.

The notation of a k-representative leaf set generalizes to all rooted subtrees of the rooted tree T. Moreover, the computation of the corresponding sets can be done in linear time. Figure 1 shows that the representative leaf set of T can be computed from the representative leaf sets of the rooted subtrees T_1 and T_2. Similarly, the representative leaf sets of T_1 and T_2 can be obtained from $S_k(T_3)$, $S_k(T_4)$,

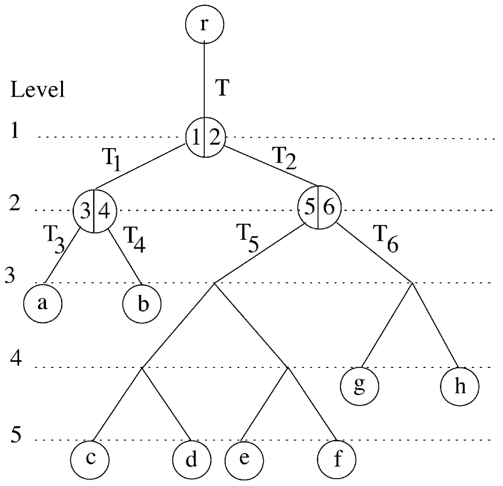

FIG. 1.—A binary tree T with root r. The horizontal lines indicate the levels of the leaves, $T_1, T_2, \ldots$ denote the rooted subtrees, where the root of T_i is indicated by the corresponding index i.

and $S_k(T_5)$, $S_k(T_6)$, respectively, and so on. The 4-representative leaf sets of subtrees T_5 and T_6 are $S_4(T_5) = \{(c, 3), (d, 3), (e, 3), (f, 3)\}$ and $S_4(T_6) = \{(g, 2), (h, 2)\}$. $S_4(T_2)$ is obtained from $S_4(T_5) \cup S_4(T_6)$ by increasing the corresponding distances by one, choosing the four leaves with the smallest distances, and breaking ties randomly. Thus, we obtain $S_4(T_2) = \{(g, 3), (h, 3), (c, 4), (d, 4)\}$.

Because the size of the k-representative leaf set of a rooted subtree T_i is $O(k)$, the cost of computing the representative leaf set from its child leaf sets is $O(k)$. Thus, for a rooted tree with n leaves the collection of all k-representative leaf sets is computed in $O(nk)$ time.

Now consider an unrooted tree T. Each internal node x splits T into three disjoint subtrees, which we then root with the original internal node x. Let T_1^x, T_2^x, and T_3^x denote the corresponding rooted subtrees with the same root x (see figure 2, left). For these rooted subtrees we compute $S_k(T_1^x)$, $S_k(T_2^x)$, and $S_k(T_3^x)$. As in the case of rooted trees the computation of all representative leaf sets for all rooted subtrees is efficiently possible. The collections of k-representative leaf sets are the building blocks for the IQP algorithm.

Now we are ready to introduce the concept of *important quartets*. In the original PUZZLE algorithm (Strimmer and von Haeseler 1996) a tree with n leaves was reconstructed sequentially by starting with a randomly chosen quartet tree into which the next sequence is inserted by evaluating all quartet trees that contain three sequences already present in the reconstructed tree and a fourth sequence, which needs to be inserted in this tree. If a tree with n leaves is reconstructed $O(n^4)$ quartets need to be evaluated.

We call a quartet $q = (t_1, t_2, t_3, y)$ an *important quartet* of an internal node x of an unrooted tree T if and only if

- the sequence y does not belong to the leaves of tree T, and
- the sequences t_1, t_2, and t_3 are elements of $S_k(T_1^x)$, $S_k(T_2^x)$, and $S_k(T_3^x)$, respectively.

Thus an important quartet consists of three representatives each from one of the three k-representative leaf sets derived from the internal node x and a sequence y, which needs to be inserted in the tree. By construction t_1, t_2, t_3 are close to x and close to each other; thus the reconstruction of a quartet tree for the quartet (t_1, t_2, t_3, y) is more likely to be accurate, because quartets with closely related sequences are less affected by the accumulation of evolutionary noise due to parallel and back substitutions.

More generally, a quartet q is an important quartet of an unrooted tree T if it is an important quartet of some internal node x of T.

Each internal node splits the tree into three rooted subtrees, and each of them is presented by at most k representative leaves. For a new sequence y and an internal node x, there are $O(k^3)$ important quartets. Because T has $O(n)$ internal nodes, $O(nk^3)$ important quartets are possible.

The IQP Algorithm

To put a new sequence y into a tree T, we first determine the important quartet set of the tree T by

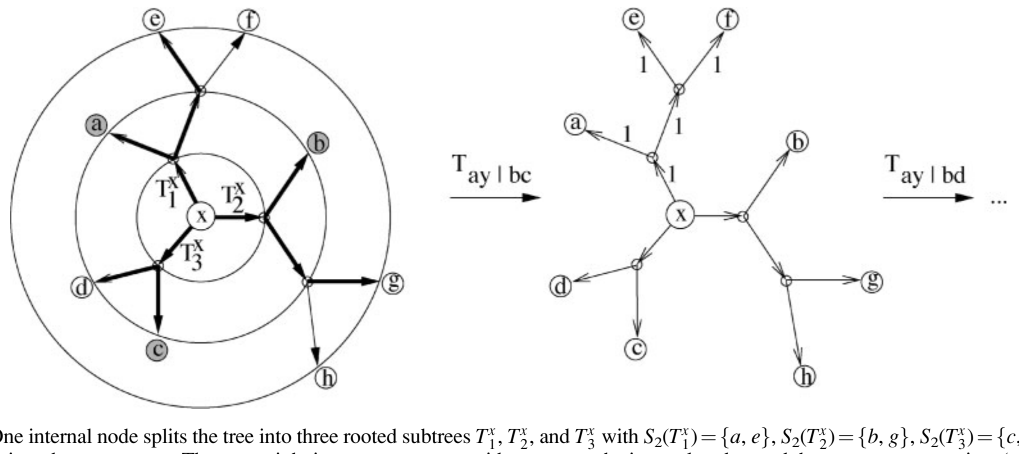

FIG. 2.—One internal node splits the tree into three rooted subtrees T_1^x, T_2^x, and T_3^x with $S_2(T_1^x) = \{a, e\}$, $S_2(T_2^x) = \{b, g\}$, $S_2(T_3^x) = \{c, d\}$. A new y is needed to insert into the current tree. There are eight important quartets with respect to the internal node x and the new sequence y, i.e., (y, a, b, c), (y, a, b, d), (y, a, g, c), (y, a, g, d), (y, e, b, c), (y, e, b, d), (y, e, g, c), and (y, e, g, d). The thick lines are the paths from the root x to representative leaves of three rooted subtrees. The gray circles indicate the representative leaves that are needed to compute the quartet tree (y, a, b, c). Because $T_{a,y|b,c}$ is reconstructed, all branches in the rooted subtree T_1^x receive a weight of 1.

specifying important quartets for all internal nodes of the tree T. Then for each important quartet $q = (t_1, t_2, t_3, y)$ the optimal tree topology (with respect to some objective function) is computed among the three unrooted topologies $T_{t_1,t_2|t_3,y}$, $T_{t_1,t_3|t_2,y}$, and $T_{t_1,y|t_2,t_3}$ (see fig. 3). In IQP, quartet trees are constructed using Neighbor Joining; these trees are the minimum evolution trees in case of four sequences (Saitou and Nei 1987). The estimated quartet trees are then used to place sequence y.

Figure 2 illustrates the procedure. To this end, consider quartet $q = (a, b, c, y)$ comprising the new sequence y, leaves a, c, and b are representatives of subtrees T_1^x, T_2^x, and T_3^x, respectively. Assume that, based on the sequences a, b, c, y, tree $T_{ay|bc}$ is reconstructed, then each branch of T_1^x gets score 1. Then we continue with the next quartet (a, b, d, y).

Repeating the above procedure for all important quartets of T and summing the resulting scores assigns a total score to each branch. Sequence y is inserted on the branch with the highest score. In case of ties, a branch with highest score is selected at random. Scores are computed in $O(nk^3)$ time by using a simple recursive procedure (Schmidt 2003, chapter 4).

In the following, k was set equal to four. Thus, if one wants to compute a phylogenetic tree following the TREE-PUZZLE procedure but using only important quartets, then one would need $O(n^2)$ computing time. However, we will not pursue this approach here, because simulations showed that the accuracy of this approach is not satisfying (data not shown).

Combining Tree-Reconstruction Methods

Here we suggest a combination of tree reconstruction methods to compute an optimal tree.

Combined-Algorithm (IQPNNI)

1. *initial step:* An initial tree is built applying BIONJ (Gascuel 1997). Then NNI is performed until no further improvement of the likelihood function is found (Guindon and Gascuel 2003). We call the resulting tree T_{best}, with log-likelihood ℓ_{best}.
2. *optimization step:* Delete each leaf with probability $0 < p_{del} < 1$ from T_{best}. Re-insert the collection of deleted leaves by applying the IQP. Optimize the resulting tree $T_{\text{intermediate}}$ using NNI.
3. *comparative step:* If the log-likelihood $\ell_{\text{intermediate}} > \ell_{\text{best}}$, then set $T_{\text{best}} \leftarrow T_{\text{intermediate}}$ and update the log-likelihood.
4. *stop criterion:* If the number of optimization steps is less than or equal to a pre-defined number of total optimization steps M, go to 2; otherwise, stop and output T_{best}.

This combined strategy has two main advantages. First, deleting and re-inserting some leaves in the optimization step helps us to escape from a local optimum when applying NNI. Moreover, deleting and re-inserting only a proportion of all leaves conserves parts of the optimized tree and therefore saves computing time.

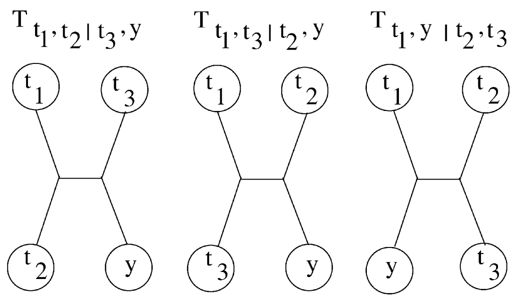

FIG. 3.—The three tree topologies of 4 sequences t_1, t_2, t_3, and y and their abbreviations as used in the text.

Accuracy

We tested the accuracy and computing time of IQPNNI, as well as other methods with simulated data and two real data sets. Computing time was measured on a 2.0 GHz PC with 512M RAM. The size of the representative leaf set was $k = 4$; the probability to delete a sequence was 0.3 for simulated data and 0.1 for the biological data.

The accuracy of IQPNNI was compared to Weighbor 1.2 (Bruno, Socci, and Halpern 2000), fastDNAml version 1.2 (Olsen et al. 1994), and PHYML version 2.0.1 (Guindon and Gascuel 2003). Weighbor is a distance-based method and is combined with DNADIST version 3.5 (Felsenstein 1993); the other programs are maximum likelihood methods (see Ranwez and Gascuel 2001 for a detailed reference about the performance of current quartet-based methods compared to other approaches). MetaPIGA (Lemmon and Milinkovitch 2002) was not included in the simulation study because it offers no version that runs in batch mode.

All methods were run with default options. However, the parameters of models of sequence evolution were not estimated but set identical to the simulated conditions.

Accuracy was measured as the percentage of cases where the inferred tree topology and the true tree topology are identical. Alternatively, we computed the distance between the inferred tree and the true tree according to Robinson and Foulds (1981)—i.e., the number of bipartitions present in one of the two trees but not the other.

Small Simulated Data

We generated randomly 3,000 trees with 30 taxa. Trees were drawn from the Yule-Harding distribution (Harding 1971). The branch lengths of trees were drawn from an exponential distribution with mean values equal to 0.03, 0.06, and 0.15 to accommodate for slow, medium, and high rates of evolution, respectively. Seq-Gen (Rambaut and Grassly 1997) was used to evolve sequences along the trees using the Kimura two-parameter model (Kimura 1980) with a transition/transversion ratio of 2.0, and a sequence length of 500 bp.

Tables 1 and 2 show that IQPNNI outperforms all other methods analyzed. However, the performance is, on average, only marginally better, i.e.—3.50 versus 3.54 for the Robinson and Foulds distance (table 2). Weighbor,

Table 1
The Percentage of Cases for Which the Inferred Tree and the True Tree Are Identical in Small Simulated Data Sets

	Weighbor	fastDNAml	PHYML	**IQPNNI**[a]
Slow rate	9.7	14.3	14.3	**14.7**
Medium rate	12.5	19.9	19.9	**20.4**
High rate	11.2	16.3	15.9	**16.5**
Average	11.1	16.8	16.7	**17.2**

[a] Parameter settings: $M = 20$, $p_{del} = 0.3$, and $k = 4$.

a distance-based method, displays the lowest performance in terms of accuracy.

It may be surprising that the probability to reconstruct the true tree is so small, an effect that is due to the short sequence length (500 bp). If we were to use sufficiently long sequences, then all methods would perform equally well with respect to accuracy. In this sense, reconstruction of phylogenetic trees is easy for simulated data. However, for biological data one typically has short sequences.

The computing time to estimate a phylogenetic tree with 30 sequences is not really an issue for the four programs tested. Weighbor is the fastest program; it needs on average 0.4 seconds to output a tree. PHYML is next (2.9 seconds), followed by IQPNNI (16.7 s), and fastDNAML (28.9 seconds). Because computing time is not an issue for small data sets, one should apply the program with highest accuracy.

Large Simulated Data

The accuracy of IQPNNI for large data sets was investigated on one random tree topology that was created as described above, but with 1,000 sequences of length 500, 1,000, and 2,000 base pairs. The mean branch length was set to 0.05. We compared Weighbor, PHYML, and IQPNNI. Unfortunately fastDNAml could not be applied to 1,000 sequence data sets, because the computing time was too long. Table 3 displays the Robinson and Foulds (1981) distance for the three tree reconstruction methods designed to deal with large numbers of taxa. The numbers in the table show the results of 10 simulation runs and the average performance for each method as a function of the alignment length. Not surprisingly, as the sequence length increases all methods get better, that is they reconstruct trees that are closer to the true tree. However, there is a substantial difference in the performance between Weighbor and PHYML or IQPNNI. Weighbor shows substantially reduced accuracy. The differences in performance between PHYML and IQPNNI are less pronounced.

IQPNNI is in 21 out of 30 simulations closer to the true tree; with PHYML, we observe 7 ties and two cases where PHYML is closer to the true tree. Ties occur for long sequences. Thus, IQPNNI shows a higher accuracy for short sequences. For longer sequences the differences between the two approaches disappear. We should note however, that the log-likelihood of the IQPNNI tree is typically higher than that of the corresponding PHYML tree. This increase in likelihood is associated with an increase in computing time. Table 4 displays the time necessary to evaluate one tree. Because we set the number of intermediate trees to $M = 100$ it took on average 4.5 h (500 bp), 7.2 h (1,000 bp), or 10.5 h (2,000 bp) for IQPNNI to run a simulation. On the other hand, because a large number of different trees were analyzed per run, we might put more confidence in the resulting tree. Note that $M = 100$ iterations do not allow for a thorough search in tree-space. For biological data sets the number of iterations must be larger.

Table 2
Average Robinson and Foulds Distance of Small Simulated Data Sets

	Weighbor	fastDNAml	PHYML	**IQPNNI**[a]
Slow rate	4.53	3.71	3.71	**3.64**
Medium rate	4.12	3.35	3.23	**3.22**
High rate	4.56	4.05	3.69	**3.64**
Average	4.40	3.70	3.54	**3.50**

[a] Parameter settings: $M = 20$, $p_{del} = 0.3$, and $k = 4$.

Table 3
Robinson and Foulds Distance for 30 Simulations of Data Sets with 1,000 Sequences

Simulation:	1	2	3	4	5	6	7	8	9	10	**Average**
500 bp											
Weighbor	258	238	196	242	260	234	244	264	202	226	**236.4**
PHYML	128	122	108	132	96	98	116	148	108	94	**115.0**
IQPNNI[a]	118	106	102	118	86	92	110	138	104	92	**106.6**
1,000 bp											
Weighbor	156	146	164	152	146	160	152	158	172	170	**157.6**
PHYML	86	56	76	76	76	64	76	80	70	80	**74.0**
IQPNNI[a]	88	56	72	72	72	62	70	80	66	76	**71.7**
2,000 bp											
Weighbor	94	108	120	124	106	102	112	92	100	104	**106.2**
PHYML	36	32	48	38	40	32	46	42	42	56	**41.2**
IQPNNI[a]	28	30	48	38	38	32	44	44	42	56	**39.0**

[a] Parameter settings: $M = 100$, $p_{del} = 0.3$, and $k = 4$.

Real Data

We applied IQPNNI to two large data sets, the ssu-rRNA alignment (218 species, 4,182 bp) and the rbcl-gene alignment (500 species, 1398 bp) recently analyzed (Guindon and Gascuel 2003), and we compared our results to PHYML (Guindon and Gascuel 2003) and MetaPIGA (Lemmon and Milinkovitch 2002). The parameter settings for IQPNNI are given in table 5. The

Table 4
Computing Times (in minutes) for 1,000 Sequences and Different Tree-Building Methods

	Weighbor	PHYML	**IQPNNI (for one intermediate tree)**[a]
500	190.0	6.5	**2.7**
1000	190.0	13.5	**4.3**
2000	172.0	19.0	**6.3**
Average	184.0	13.0	**4.4**

[a] Parameter settings: $M = 100$, $p_{del} = 0.3$, and $k = 4$.

Table 5
Best log Likelihoods and Computing Times for Three Tree Reconstruction Methods for Real Data

		log likelihood		
Gene	Number of Taxa	PHYML	MetaPIGA	**IQPNNI**[a]
ssu rRNA	218	−156,895	−156,715	**−156,604**
rbcl	500	−100,191	−100,080	**−100,011**
		Runtime (min)		
ssu rRNA	218	5.1	74.5	**379**
rbcl	500	7.5	158.5	**672**

NOTE.—Although we used the same data as Guindon and Gascuel (2003) our likelihood differs slightly from the ones published, a difference that is due to the new version 2.0.1 of PHYML (Guindon and Gascuel 2003). In addition, results of MetaPIGA (Lemmon and Milinkovitch 2002) depend on the random process.

[a]Parameter settings: $M = 300$, $p_{del} = 0.1$, and $k = 4$.

HKY model (Hasegawa, Kishino, and Yano 1985) was used for the DNA data and the transition transversion parameter was estimated from the data. The results are summarized in table 5.

In both cases IQPNNI found quite a lot of trees with higher likelihood values than the ones obtained with PHYML and MetaPIGA (see Figs 4 and 5). For the ssu-rRNA data, the best IQPNNI tree is 291 and 111 log-likelihood units higher than PHYML and MetaPIGA, respectively. For the rbcl-gene the best IQPNNI tree is 180 log-likelihood units larger than the PHYML tree and 69 log-likelihood units higher than the MetaPIGA tree. Thus, the increase in computation time (see table 5) is rewarded by a better maximum likelihood tree.

Stopping the Search

Figures 4 and 5 display for the ssu and rbcl DNA data sets the increase in log-likelihood as the number of iterations grows. During the first iterations we almost instantaneously observe a drastic increase in the likelihoods. After roughly 100 iterations (220 min or 300 min computing time) we are finding better trees compared to PHYML and MetaPIGA. The + signs above the lines labeled MetaPIGA and PHYML in figures 4 and 5 indicate the better trees that we discovered. As the search continues, the rate of discovering better trees decreases. However, as we continue our "guided tour through" tree space we keep on finding better trees.

Thus, as with MetaPIGA, we need a criterion to stop our quest for the best tree. Here we suggest applying an estimation method that is based on the time of occurrence (i.e., number of iterations) of better trees during our search. These time points are indicated by the jumps in the graph in Figures 4 and 5.

More precisely, let $L_1, L_2, \dots, L_j$ denote the log-likelihoods for the first j iterations, then the sequence $\tau(k)$ of record times (i.e., iteration number, when a better tree is found) is defined by

$$\tau(1) = 1, \quad \tau(k+1) = \min\{j | L_j > L_{\tau(k)}\}.$$

This sequence is used to estimate the point in time, τ_{stop}, at which to stop the search, i.e., the point at which it appears unlikely that a further search will lead to a better tree.

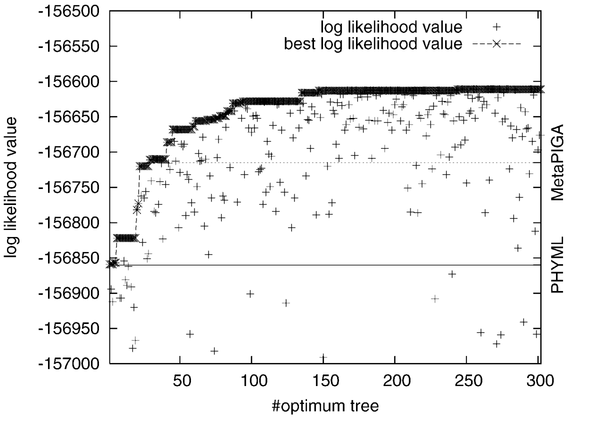

FIG. 4.—Exploring the likelihood surface of the suu-rRNA alignment of 218 species. The thick line shows the improvement in log likelihood during the IQNNI search. The plus signs represent trees generated according to the combined algorithm. IQPNNI generated 258 different trees; 219 of them had a higher likelihood than the best PHYML tree (horizontal line) (Guindon and Gascuel 2003), and 121 were better than MetaPIGA (horizontal dashed line) (Lemmon and Milinkovitch 2002).

Using the theory detailed in Cooke (1980) and Robert and Solow (2003), we estimate during the run of IQPNNI an upper 95% confidence limit $\tau_{95\%}$ of τ_{stop}. More precisely, consider a sequence of record times $\tau_1, \tau_2, \dots, \tau_k$, then we compute an upper $(1 - \alpha)100\%$ stopping time as

$$\tau_{(1-\alpha)100\%} = \tau_1 + \frac{\tau_1 - \tau_k}{\left(\frac{-\log(\alpha)}{k}\right)^{-\hat{\nu}} - 1},$$

where the shape parameter of the joint Weibull distribution ν is estimated as follows

$$\hat{\nu} = \frac{1}{k-1} \sum_{j=1}^{k-2} \log\left(\frac{\tau_1 - \tau_k}{\tau_1 - \tau_{j+1}}\right).$$

Once $\tau_{95\%}$ iterations have been carried out and a better tree has not been detected, the program will stop and output the

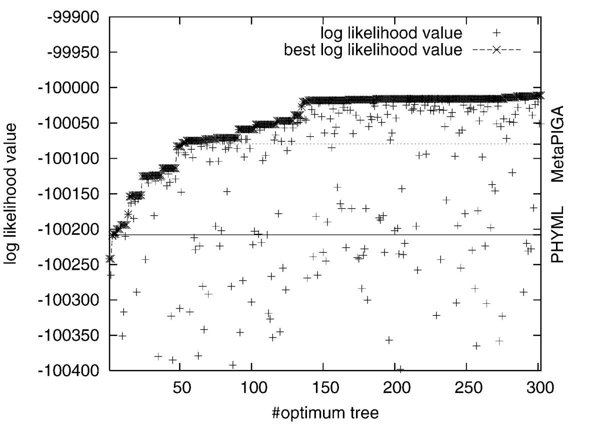

FIG. 5.—Exploring the likelihood surface of the rbcl alignment of 500 species (see figure 4). In all, 283 different trees were generated; 192 had a higher likelihood than the best PHYML tree (horizontal line) (Guindon and Gascuel 2003), and 125 showed larger log likelihoods as the best MetaPIGA tree (horizontal dashed line) (Lemmon and Milinkovitch 2002).

best tree found. We can conclude that we will not find a better tree with a probability of 95% during this search. Nevertheless, if a better tree is found before we hit $\tau_{95\%}$, we re-compute $\tau_{95\%}$ on the basis of the new record time added to the sequence $\tau(k)$.

This additional number of iterations to reach $\tau_{95\%}$ further increases the computation time; i.e for the 218-ssu rRNA about 8 hours were necessary and for the rbcl data we used a total of 15 h. But now we are in the position to know that, with 95% probability, we would not have found a better tree when extending the search even longer. If one is willing to spend more computation time, it is of course possible to compute $\tau_{99\%}$ or even higher upper bounds. One could also start a new run of IQPNNI.

Discussion

We have presented the IQP method to reconstruct trees from large sequence data. Our simulations and the analysis of biological data show that a combination of nearest neighbor interchange and IQP leads to trees that are either closer to the true tree, as defined by the simulations, or closer to trees with higher likelihoods than found so far for biological examples. This improved performance, however, is achieved by an increase in computing time as compared to other fast programs. However, the times needed are not unrealistic.

The algorithm presented here turns out to search the tree space efficiently for better trees. We have used the maximum likelihood criterion as an objective function; however, the algorithm is not restricted to maximum likelihood. Any tree reconstruction method with an objective function to minimize or maximize will work in the presented approach.

Although we have only presented results for $p_{del} = 0.3$ and $k = 4$ for simulated data, further studies with p_{del} ranging from 0.2 to 0.4 and k from 4 to 6 showed that the accuracy, based on the Robinson and Foulds distance, percentage of correctly reconstructed trees is not affected (data not show). Similarily, various combinations of k (4, 5, or 6) and p_{del}, ranging from 0.1 to 0.3 resulted in minor changes in the log likelihood, which varied from −156,715 to −156,604 ($k = 4$, $p_{del} = 0.1$) for the ssu_rRNA data set and from −100,058 to −100,011 ($k = 4$, $p_{del} = 0.1$) for the rbcl-gene alignment. We note that this observation does not allow any generalizations. In any real application one should run IQPNNI for different choices of k and p_{del}.

Our simulations indicate that IQPNNI shows a better performance than other tested methods in terms of being closer to the true tree. However, we have only considered a very narrow range of simulation. We have not taken into account the posibility of model violations, uncertainty of estimating the parameters of the model, and various other sources of uncertainty. Including all this is beyond the scope of the paper.

Because IQPNNI generates trees with similar high likelihoods, we have included an option in the program that allows output of a majority rule consensus tree of all the intermediate trees found during one search run of the tree space. We also output the frequencies of the groupings found in that tree. If one is interested only in the most frequent groupings found in the collection of trees with a high likelihood, then this option is helpful.

Finally, we are suggesting a statistic that provides a guide to stop the search for a better tree. This very simple and crude estimation procedure proves to be very useful, although it increases the computation time yet again. To get additional confidence in the reconstructed tree and its maximum likelihood value, one needs to repeat the search with several independent runs of our program. However, we are sure that we have only scratched the surface when it comes to applying statistical inference based on record values to problems of tree reconstruction. Further investigations into the performance of such methods are certainly necessary, but beyond the scope of this paper.

Supplementary Material

A computer program, called IQPNNI, is available and can be downloaded from Web site http://www.bi.uni-duesseldorf.de/software/iqpnni. The source code will be released soon and will be free to use for academic purposes. IQPNNI is written in C++ and can be run on all popular platforms, i.e., Windows, Linux. The program consists of over 40 classes, and some parts are taken from the TREE-PUZZLE package (http://www.tree-puzzle.de) and BIONJ (http://www.lirmm.fr/˜w3ifa/MAAS/BIONJ/BIONJ.html) software.

The ribosomal sequence alignment can be retrieved from http://rdp.cme.msu/download/SSU_rRNA/alignments/SSU_Prok_rep_phylip), and the rbcl alignment can be downloaded from (http://www.cis.upenn.edu/˜krice/treezilla/record.nex).

Acknowledgments

We express special thanks to Heiko Schmidt for his help. We thank Mark Wilkinson and an anonymous referee for helpful comments. We appreciate support from Stephane Guindon and Oliver Gascuel for providing an early manuscript of their paper as well as providing some data and their source code. We thank Arnold Janssen for helpful insights to extreme value theory. The assistance of Thomas Schlegel and other members of the Bioinformatics Institute are highly appreciated. Finally, we thank Jutta Buschbom for carefully reading the manuscript.

Literature Cited

Brauer, M. J., M. T. Holder, L. A. Dries, D. J. Zwickl, P. L. O. Lewis, and D. M. Hillis. 2002. Genetic algorithms and parallel processing in maximum-likelihood phylogeny inference. Mol. Biol. Evol. **19**:1717–1726.

Bruno, W. J., N. D. Socci., and A. L. Halpern. 2000. Weighted Neighbor Joining: a likelihood-based approach to distance-based phylogeny reconstruction. J. Mol. Evol. **17**:189–197.

Charleston, M. A. 2001. Hitch-hiking: a parallel heuristic search strategy, applied to the phylogeny problem. J. Comput. Biol. **8**:79–91.

Cooke, P. 1980. Optimal linear estimation of bounds of random variables. Biometrika **67**:257–258.

Felsenstein, J. 1993. PHYLIP (Phylogeny Inference Package) version 3.5c. Department of Genetics, University of Washington, Seattle. Distributed by the author.

———. 2004. Inferring Phylogenies. Sinauer Associates, Sunderland, Mass.

Gascuel, O. 1997. BIONJ: an improved version of the NJ algorithm based on a simple model of sequence data. Mol. Biol. Evol. **14**:685–695.

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

Harding, E. F. 1971. The probabilities of rooted tree-shapes generated by random bifurcation. Adv. Appl. Prob **3**:44–77.

Hasegawa, M., H. Kishino, and T.-A. Yano. 1985. Dating of the human–ape splitting by a molecular clock of mitochondrial DNA. J. Mol. Evol. **22**:160–174.

Huson, D. H., S. M. Nettles, and T. J. Warnow. 1999. Disk-covering, a fast-converging method for phylogenetic reconstruction. J. Comput. Biol. **6**:369–386.

Kimura, M. 1980. A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. J. Mol. Evol. **16**:111–120.

Larget, B. and D. L. Simon. 1999. Markov chain Monte Carlo alogorithms for the Bayesian of phylogenetic trees. Mol. Biol. Evol. **16**:750–759.

Lemmon, A. R., and M. C. Milinkovitch. 2002. The metapopulation genetic algorithm: an efficient solution for the problem of large phylogeny estimation. Proc. Natl. Acad. Sci. USA **99**:10516–10521.

Olsen, G. J., H. Matsuda, R. Hagstrom, and R. Overbeek. 1994. fastDNAml: a tool for construction of phylogenetic trees of DNA sequences using maximum likelihood. Comput. Appl. Biosci. **10**:41–48.

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

Ranwez, V., and O. Gascuel. 2001. Quartet-based phylogenetic inference: improvements and limits. Mol. Biol. Evol. **18**:1103–1116.

Robert, D. L., and A. R. Solow. 2003. When did the dodo become extinct? Nature **426**:245–245.

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

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

Saitou, N., and M. Nei. 1987. The Neighbor–Joining method: a new method for reconstructing phylogenetic trees. Mol. Biol. Evol. **4**:406–425.

Schmidt, H. A. 2003. Phylogenetic trees from large datasets. Ph.D. thesis, Universität Düsseldorf.

Schmidt, H. A., and A. von Haeseler. 2002. Quartet trees as a tool to reconstruct large trees from sequences. Pp. 379–388 *in* K. Jajuga, A. Sokolowski, and H.-H. Bock, eds., Data analysis, classification, and related methods. Studies in classification, data analysis, and knowledge organization. Springer-Verlag, Heidelberg, New York.

Schmidt, H. A., E. Petzold, M. Vingron, and A. von Haeseler. 2003. Molecular phylogenetics: parallelized parameter estimation and quartet puzzling. J. Parallel Distrib. Comput. **63**:719–727.

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

Strimmer, K. and A. von Haeseler. 1996. Quartet puzzling: a quartet maximum–likelihood method for reconstructing tree topologies. Mol. Biol. Evol. **13**:964–969.

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

Vos, R. 2003. Accelerated likelihood surface exploration: the likelihood ratchet. Syst. Biol **52**:368–373.

Willson, S. J. 1999. Building phylogenetic trees from quartets by using local inconsistency measures. Mol. Biol. Evol. **16**: 685–693.

Martin Embley, Associate Editor

Accepted May 25, 2004