PhyNav: A Novel Approach to Reconstruct Large Phylogenies

Le Sy Vinh[1], Heiko A. Schmidt[1], and Arndt von Haeseler[1,2]

[1] NIC, Forschungszentrum Jülich, D-52425 Jülich, Germany
[2] Bioinformatik, HHU Düsseldorf, D-40225 Düsseldorf, Germany

Abstract. A novel method, PhyNav, is introduced to reconstruct the evolutionary relationship among contemporary species based on their genetic data. The key idea is the definition of so-called minimal k-distance subsets which has fewer sequences but contains most relevant phylogenetic information from the whole dataset. For this reduced subset the subtree is created faster and serves as a scaffold to construct the full tree. Because many minimal subsets exist the procedure is repeated several times and the best tree with respect to some optimality criterion is considered as the inferred phylogenetic tree. PhyNav gives encouraging results compared to other programs on both simulated and real datasets.

A program to reconstruct phylogenetic trees based on DNA or amino acid based is available (`http://www.bi.uni-duesseldorf.de/software/phynav/`).

1 Introduction

One objective in phylogenetic analysis is the reconstruction of the evolutionary relationship among contemporary species based on their genetic information. The relationship is described by an unrooted bifurcating tree on which the leaves represent contemporary species and the internal nodes represent speciation events. The total number of unrooted bifurcating trees with $n \geq 3$ leaves is $\prod_{i=3}^{n} (2i-5)$ (cf. Felsenstein (1978)). This number increases rapidly with n. For $n = 55$ sequences the number of trees exceeds the estimate of 10^{81} atoms in the known universe.

Commonly used tree reconstruction methods are classified into three groups: (1) Minimum evolution methods (e.g., Rzhetsky and Nei (1993)), (2) maximum parsimony methods (e.g., Fitch (1971)), and (3) maximum likelihood methods (e.g., Felsenstein (1981)). Among these the maximum likelihood (ML) methods are statistically well founded and tend to give better results. An overview is given in Swofford et al. (1996) and Felsenstein (2003).

Here, we propose a new heuristic tree search strategy, which reduces the computational burden. Details how to construct the subsets are given in section 2. In section 3 we will describe PhyNav algorithm and how it elucidates the landscape of possible optimal trees. The algorithm is then applied to simulated as well as biological data (Section 4). Finally the results as well as some possible changes and improvements are discussed.

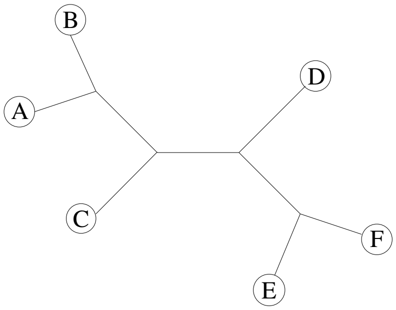

Fig. 1. An unrooted bifurcating tree of 6 species

2 Minimal *k*-distance subsets

First, we introduce the concept of *k-distance representatives*. A sequence s is said to be a k-distance representative for a sequence s' in a tree T if and only if their topological distance $d(s, s')$ in T, that is the number of branches on the path from s to s', is smaller or equal to $k \geq 0$. The smaller the value of k is the better a sequence s represents sequence s', and vice versa.

The k-distance representative sequence concept is now used to introduce minimal k-distance subsets. A subset S_k of sequences is called a minimal k-distance subset of an n-sequence set S iff the following two conditions hold:

1. For each sequence $s \in S$, there exists a sequence $s' \in S_k$ such that the sequence s' is a k-distance representative for the sequence s.
2. If we remove any sequence s' from S_k, S_k will violate the first condition. That means, the subset cannot be reduced any further.

The idea behind minimal k-distance subsets is that the phylogenetic information in the sequence subset S_k represents phylogenetic information from the whole dataset. $k = 3$ is a good choice according to our experience. This value retains enough phylogenetic information and connects distances that are far away.

A sequence $s \notin S_k$ is then called a *remaining sequence*. The set $\overline{S_k} := S \backslash S_k$ of all such sequences, which remain to be added to S_k to obtain the full set S, is called *remaining set*.

Since $|S_k| \leq |S|$, the subtree T_k from subset S_k can usually be constructed in less time than the full tree. This subtree is used as a scaffold to build a full tree containing all sequences by adding all sequences $r \in \overline{S_k}$.

For example, the sequences A and B in the tree in Figure 1 are 2-distance representative of each other, as are E and F. The sequence subsets $\{A, C, D, E\}$, $\{A, C, D, F\}$, $\{B, C, D, E\}$ and $\{B, C, D, F\}$ are minimal 2-distance representative subsets of the full set $\{A, B, C, D, E, F\}$.

3 The PhyNav algorithm

The Navigator algorithm is a three-step procedure: (1) the *Initial step*, (2) the *Navigator step*, and (3) the *Disembarking step*. We can use the algorithm with any objective function, e.g. maximum parsimony, maximum likelihood, to create a list of possible optimal trees. According to this objective function the best tree found is taken as the inferred phylogeny. In the PhyNav program we use the maximum likelihood principle.

The *Initial step* employs some fast tree reconstruction method to create an initial tree. In particular, PhyNav uses the BIONJ algorithm (Gascuel (1997)) with the pairwise evolutionary distances and a fast nearest neighbor interchange (NNI) operation as described by Guindon and Gascuel (2003) to create the initial tree. This tree is then called the current best tree and denoted as T_{best}. T_{best} is used to construct the k-distance subsets.

Next, the *Navigator step* finds a minimal k-distance subset S_k and constructs the corresponding subtree T_k. Note, that there exist many minimal k-distance subsets. One S_k can be determined in time of $O(n^2)$ (details are left out due to limited space). From the minimal k-distance subset S_k the corresponding subtree T_k could be created by several tree reconstruction methods. PhyNav used the subtree T_{sub} of T_{best} induced by the leaves in S_k. This subtree is subsequently optimized using NNI operations.

The last step, the *Disembarking step*, constructs the whole tree T based on the scaffold T_k using the k-distance information. PhyNav applies a simple method to insert the remaining sequences into the scaffold tree as follows. First we assign T by T_k. Each remaining sequence $r \in \overline{S_k}$ is inserted into an external branch e of T such that the corresponding leaf s_e adjacent to e is a k-distance representative for r. If there are more than one external branches possible one is selected randomly. To compensate for incorrect placements the full tree is optimized using again the fast NNI operation mentioned above. We replace the T_{best} by T if T has a better score.

It cannot be guaranteed that T_k determined in the *Navigator step* is the optimal tree for S_k due to the use of heuristics. Even if T_k is the best tree it does not guarantee that the tree T will be the optimal full tree. Hence, the *Navigator* and *Disembarking steps* are repeated several times. The number of repetitions can be adjusted. The best tree T_{best} is considered the final phylogenetic n-tree.

4 The efficiency of PhyNav

To measure the accuracy and the time-efficiency of PhyNav we reconstructed phylogenetic trees from simulated as well as biological datasets. The results are compared to the results of other programs, in particular, Weighbor (Bruno et al. (2000); version 1.2) and PHYML (Guindon and Gascuel (2003); version 2.1).

Computing times were measured on a Linux PC Cluster with 2.0 GHz CPU and 512MB RAM.

4.1 Simulated datasets

Analysis

To evaluate the accuracy we performed simulations. To simulate realistic datasets we performed the simulations on a tree topology reconstructed from a real dataset. To that end an elongation factor (EF-1α) dataset with 43 species was used. The dataset as well as the tree was obtained from TreeBase (`http://www.treebase.org`, study accession number S606, matrix accession number M932). The branch lengths of the tree topology were inferred using the TREE-PUZZLE package (Strimmer and von Haeseler (1996), Schmidt et al. (2002); version 5.1).

Based on that tree topology datasets were simulated using Seq-Gen (Rambaut and Grassly (1997); version 1.2.6) assuming the Kimura 2-parameter model with an transition:transversion ratio of 2.0 (Kimura (1980)). 1,000 datasets each were simulated with sequence lengths of 700 and 1000 bp.

The trees for simulated datasets were reconstructed using PhyNav, Weighbor (Bruno et al. (2000); version 1.2) and PHYML (Guindon and Gascuel (2003); version 2.1).

All programs were run with default options. The evolutionary model and its parameters were set to the simulation parameters. The PhyNav options were set to 5 repetitions and $k = 3$.

The results of the tree reconstructions were compared using two different methods. First the percentage of correctly reconstructed tree topologies was derived for each program and sequence length. To measure the variability of the results for each program, the Robinson-Foulds distance (Robinson and Foulds (1981)) was computed from each tree to the 'true tree' and the average was taken for each program and sequence length. The Robinson-Foulds distance counts the number of splits (bipartitions) in the two trees, which occur in only one of the trees. If the trees are identical their distance is zero.

Results for the simulated datasets

Tables 2(a) and 2(b) display the results for PhyNav, PHYML, and Weighbor. Both tables show that the faster Weighbor program (Table 2(c)) is out-performed by both PHYML and PhyNav. PHYML and PhyNav perform similarly well, both in the percentage of correctly reconstructed trees as well as their average Robinson-Foulds distance to the 'true tree'. However, PhyNav shows slightly better values for all analyses, and hence gives better results.

Table 1. Results for the simulated datasets: (a) percentage of correctly reconstructed trees, (b) average Robinson-Foulds distance between the 'true tree' and the reconstructed trees, and (c) average runtime of tree reconstruction.

	Weighbor	PHYML	PhyNav
700 bp	2.4	12.3	13.1
1000 bp	9.6	33.7	33.9

(a) Percentage of correct trees.

	Weighbor	PHYML	PhyNav
700 bp	7.57	4.09	3.96
1000 bp	4.62	2.11	2.07

(b) Robinson-Foulds distance.

	Weighbor	PHYML	PhyNav
700 bp	3.0s	6.9s	51.8s
1000 bp	3.6s	9.0s	65.8s

(c) Average runtime.

4.2 Biological datasets

Analysis

The PhyNav algorithm was applied to large biological datasets to test its efficiency on real datasets. Three datasets have been obtained from the PANDIT database (`http://www.ebi.ac.uk/goldman-srv/pandit/`; Whelan et al. (2003)). The first dataset consists of 76 Glyceraldehyde 3-phosphate dehydrogenase sequences with an alignment length of 633 bp (PF00044), the second of 105 sequences from the ATP synthase alpha/beta family (1821 bp, PF00006), and the last of 193 sequences with Calporin homology with an alignment of 465 bp (PF00307).

Since the true tree is usually not known for real datasets, the Robinson-Foulds distance cannot be used to measure the efficiency of algorithms. Therefore the likelihood value of the reconstructed trees is used to compare the methods.

Since Weighbor does not use likelihoods we only compare PHYML and PhyNav from the methods above. Note, that Weighbor already was outperformed in the simulation study (cf. 4.1). Additionally we wanted to use METAPIGA, another method for large datasets based on a genetic algorithm (Lemmon and Milinkovitch (2002)). Unfortunately the program crashed on

Table 2. Results from the biological datasets of 76 Glyceraldehyde 3-phosphate dehydrogenase sequences, of ATP synthase alpha/beta (105 seqs.), and of 193 Calporin homologs: (a) Log-likelihood values of the best reconstructed trees and (b) Runtimes of tree reconstruction consumed by the different methods. The PhyNav column presents the runtime of a single repetition.

sequences	length	PHYML	PhyNav
76	633 bp	-32133	-32094
105	1821 bp	-88975	-88632
193	465 bp	-64919	-64794

(a) Log-likelihood values.

sequences	length	PHYML	PhyNav		
			runtime	repetitions	(single repetition)
76	633 bp	40s	2529s	70	36.1s
105	1821 bp	117s	14413s	100	144.1s
193	465 bp	101s	22306s	200	111.5s

(b) Runtimes.

all three datasets. Thus, only PHYML and PhyNav were used for comparison.

Results for the biological datasets

As explained above we use the likelihood values of the reconstructed trees to compare the efficiency of the two programs. According to the maximum likelihood framework (cf. for example Felsenstein (1981)) the tree with the higher likelihood value represents the more likely tree.

The log-likelihood values are given in Table 3(a). These results show that PhyNav always find a tree with a higher likelihood. The increase of the likelihood ranged from 39 up to 343 orders of magnitude.

However, as Table 3(b) shows, the price to pay for better likelihood trees is an increase in computing time. Each single repetition in the algorithm has a time consumption comparable to the one run of PHYML.

5 Discussion and Conclusion

In this paper, a new search strategy to find the optimal large phylogenies is proposed. Starting from an initial tree the PhyNav method uses easy heuristics to reduce the number of sequences, reconstruct scaffold trees, and adding again the remaining sequences. During these steps the constructed trees were optimized using fast NNI operations.

The suggested method produced better results on all dataset compared to Weighbor and PHYML, two tree building programs to analyze large datasets. The tradeoff for better accuracy is of course the runtime. While Weighbor outperformed PHYML and PhyNav with respected to the runtime on the simulated datasets, PHYML is 7.5-fold faster than PhyNav. However, spending more time might be well acceptable, because the quality of the results increases.

On the biological datasets PhyNav showed much longer runtimes compared to PHYML. Nevertheless, the increase of the likelihoods ranging from 39 up to 343 orders of magnitude might well justify that this effort is worthwhile, since it is still far from the time consumptions demanded by classical ML methods like DNAML (Felsenstein (1993)).

The mechanism to add the remaining sequences of $\overline{S_k}$ to T_k cannot be expected to give the most accurate results. However, our way is simple but performs efficiently, especially since the NNI operations seem to well recover unfortunate placements during the construction of the full trees T. Additionally, it might be worth trying other algorithms like Quartet Puzzling of Strimmer and von Haeseler (1996) to add the remaining sequences.

References

BRUNO, W. J., SOCCI., N. D., and HALPERN, A. L. (2000): Weighted Neighbor Joining: A Likelihood Based-Approach to Distance-Based Phylogeny Reconstruction. *J. Mol. Evol.*, *17*, 189–197.

FELSENSTEIN, J. (1978): The number of evolutionary trees. *Syst. Zool.*, *27*, 27–33.

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

FELSENSTEIN, J. (1993): *PHYLIP (Phylogeny Inference Package) version 3.5c.* Department of Genetics, University of Washington, Seattle, distributed by the author.

FELSENSTEIN, J. (2003): *Inferring Phylogenies.* Sinauer Associates, Sunderland, Massachusetts.

FITCH, W. M. (1971): Toward defining the course of evolution: Minimum change for a specific tree topology. *Syst. Zool.*, *20*, 406–416.

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 GASCUEL, O. (2003): a Simple, Fast and Accurate Algorithm to Estimate Large Phylogenies by Maximum Likelihood. *sys. biology.*

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.

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

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

ROBINSON, D. R. and FOULDS, L. R. (1981): Comparison of phylogenetic trees. *Mathematical Biosciences*, *53*, 131–147.

RZHETSKY, A. and NEI, M. (1993): Theoretical Foundation of the Minimum-Evolution Method of Phylogenetic Inference. *Mol. Biol. Evol.*, *10*, 1073–1095.

SCHMIDT, H. A., STRIMMER, K., VINGRON, M., and VON HAESELER, A. (2002): TREE-PUZZLE: Maximum likelihood phylogenetic analysis using quartets and parallel computing. *Bioinformatics*, *18*, 502–504.

STRIMMER, K. and VON HAESELER, A. (1996): Quartet Puzzling: A Quartet Maximum–Likelihood Method for Reconstructing Tree Topologies. *Mol. Biol. Evol.*, *13*, 964–969.

SWOFFORD, D. L., OLSEN, G. J., WADDELL, P. J., and HILLIS, D. M. (1996): Phylogeny Reconstruction. In: D. M. Hillis, C. Moritz, and B. K. Mable (eds.), *Molecular Systematics*, Sinauer Associates, Sunderland, Massachusetts, 407–514, 2nd edn.

WHELAN, S., DE BAKKER, P. I. W., and GOLDMAN, N. (2003): Pandit: a database of protein and associated nucleotide domains with inferred trees. *Bioinformatics*, *19*, 1556–1563.