Consensus Methods Using Phylogenetic Databases

Mahesh M. Kulkarni and Bernard M.E. Moret
Department of Computer Science
University of New Mexico
Albuquerque, NM 87131
Email: {mahesh,moret}@cs.unm.edu

Abstract— **With the increasing use and size of phylogenies, the output of reconstruction programs must be stored for future reference, in which case post-tree analyses such as consensus must be run from a database. We set out to determine whether such analyses can be run at a reasonable cost; we chose consensus (which summarizes the information from many trees into a single tree) because of its general applicability and because it creates a severe demand on the database by requiring examination of every edge of every tree.**
Methodology: **We preprocess the data (trees) to create tables that support consensus computations, using our own extensions to the PhyloDB schema of Nakhleh *et al.* For each of the three consensus methods (strict, majority, and greedy), we compare the database computation with the memory-resident computation using the Phylip consensus programs. We use a large selection of datasets of varying sizes (up to 1,000 trees of up to 1,500 taxa each) and of varying degrees of commonality.**
Results: **The computations from the database are very practical: they often run faster, and never run more than 5 times slower, than the computations in main memory using Phylip. The additional storage costs are easily handled by any database system, while the preprocessing costs remain reasonable. Thus suitable preprocessing of phylogenetic data allows post-tree analyses to be run directly from the database at much the same cost as current memory-resident analyses.**

I. Overview

A phylogeny represents the evolutionary history of a group of organisms. Phylogenies are reconstructed from many types of data through a variety of methods, mostly using from molecular data using maximum parsimony (MP), maximum likelihood, or Bayesian (MCMC) estimation. MP and MCMC methods generate many trees: summarizing such collections of trees is done by taking their *consensus*.

Consensus methods identify bipartitions that appear in most of the trees. Bipartitions present in *all* trees define the *strict* consensus; bipartitions present in a *majority* of the trees define the *majority* consensus; finally, the *greedy* consensus considers bipartitions ordered by decreasing frequency of appearance in the input set of trees [1]. All three consensus can be computed efficiently, but require much memory, since each tree may contribute $O(n)$ new bipartitions. Consensus analysis is the most common type of *post-tree analysis*. Currently such analyses are conducted on ephemeral data: trees are kept in main memory only for the duration of an interactive session.

Large phylogenies are being reconstructed (the largest reported has over 10,000 taxa [2]) and larger ones are being targeted, up to the "Tree of Life," the phylogeny of all 10^7–10^8 organisms on the planet. Reconstruction at these scales is very time-consuming and sure to be a one-time affair; the trees obtained after months of computation must be stored in a database, just as one records steps in a lab notebook.

Nakhleh *et al.* [3] proposed PhyloDB, a schema that stores phylogenies in a structured manner. We extend PhyloDB and use the paradigm of preprocessing, storing, and querying to a much larger extent than used so far in any area of bioinformatics, by preprocessing existing database elements (trees) to construct new database elements (bipartitions), in order to support analyses with simple database queries on the new tables. Our experiments indicate that the time required to compute any consensus from the database is usually less (and never more than 5 times larger) than the time needed in a memory-resident approach. Moreover, our approach becomes more effective as the size of the problem increases; indeed, for very large problems, using a database becomes mandatory.

II. Phylogenetic Databases

The phylogenetic database, TreeBASE [4], uses a simple relational schema to store trees in the Newick string format (see [5]), which does not support efficient data retrieval *within* the trees. The PhyloDB schema [3] was designed to store phylogenetic data for purposes of analysis; it includes internal relations such as Edge, Tree_Taxon, etc. Phylogenetic trees stored using this schema can be efficiently queried on database systems that support transitive closure queries. We extend PhyloDB to support efficient handling of bipartitions.

III. Handling Bipartitions

We generate bipartitions from trees in the database by traversing (using transitive closure) the edge defining a bipartition to collect the group of taxa in the subtree; we set the bit corresponding to each taxon in this subgroup to 1. The resulting bit patterns are stored in the *Partition* table, which can efficiently support the following queries.

1) Find all distinct bipartitions for a given set of trees and list all trees containing them.
2) List all bipartitions and their frequency of appearance in each tree where this frequency is greater than or equal to some threshold.
3) List all distinct bipartitions sorted by the frequency of appearance of each bipartition in a given set of trees.
4) Given a model phylogeny T, find bipartitions from a tree that are not present in T (*false negatives*) or bipartitions from T that are not present in any tree (*false positives*).

The first three queries can be used to compute the strict, majority, or greedy consensus, respectively, while the last query is needed to assess the quality of reconstructions.

The greedy consensus requires testing each candidate bipartition for compatibility with already chosen bipartitions. For that test, we use the following characterization [1], which can be implemented with bit-pattern logical operations in the database: two groups of taxa are compatible *iff* one is a subset of the other or they are disjoint.

IV. Experimental Design and Results

We ran all three consensus algorithms using Phylip in main memory and our approach from the database. Our datasets were parameterized by the number of trees, for which we used 100, 500, and $1,000$, and by the number of taxa, for which we used 100, 500, $1,000$, $1,500$, and $2,000$.

We generate a set of binary trees in 3 steps. We use the `r8s` code [6] to generate a rooted, leaf-labelled binary tree under a Yule process. From this one tree, we generate a small collection of distinct trees by applying a random sequence of *Tree Bisection and Reconnection* operations. Finally, from each of the resulting trees, we generate a cluster of closely related trees by applying a random sequence of *Nearest-Neighbor Interchange* operations in restricted areas of the trees—creating the "islands" of phylogenetic analysis [7]. The collection of all resulting trees forms one instance.

The strict consensus algorithm on m trees of n taxa each takes $O(mn)$ time using Day's algorithm [8]. However, Phylip does not implement Day's algorithm, but treats strict consensus as a special case of majority consensus, so that both run in $O(n^2m)$ time. (The bipartition of every one of the $\Theta(mn)$ edges in the input must be computed.) The running time of the greedy consensus algorithm is $O(n^2m + n^2p)$ where p is the total number of distinct bipartitions present in each dataset—the algorithm may have to test every distinct bipartition for compatibility with already selected ones.

To compute the consensus from the database, we run a single query asking for the list of all bipartitions with a given property. The database server selects only those bipartitions with that property and returns them in serial order through the cursor mechanism, thereby enabling us to process only one bipartition at a time and saving us a lot of memory.

Fig. 1 shows the running times of the Phylip and PhyloDB implementations of the strict consensus. The running times of the greedy consensus (Fig. 2) increases more sharply with the numbers of trees and taxa, because p then tends to increase beyond m or n. Fig. 2(a) shows that, as number of trees increase, the overall ratio decreases due to the slow increase in p: as we increase number of trees, the probability of getting a bipartition not contained among the processed trees decreases.

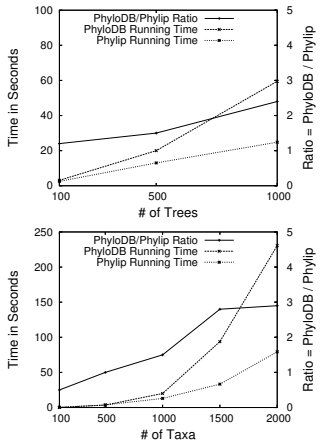

Fig. 1. Running times of Phylip and PhyloDB for the strict consensus for 1000 taxa (top) and 500 trees (bottom).

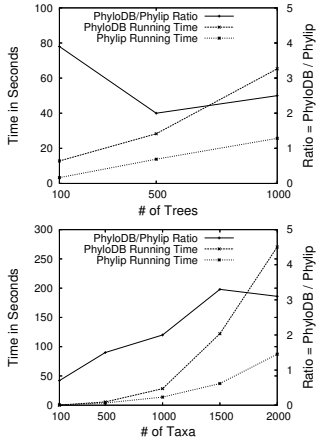

Fig. 2. Running times of Phylip and PhyloDB for the greedy consensus for 1000 taxa (top) and 500 trees (bottom).

Acknowledgements

This work is supported by the NSF under grants DEB 01-20709, IIS 01-21377, and EF 03-31654, by the NIH under grant 2R01GM056120-05A1, and by IBM Corporation under a contract from DARPA.

References

[1] D. Bryant, "A classification of consensus methods for phylogenetics," in *Bioconsensus*, ser. DIMACS Series in Discrete Mathematics and Theoretical Computer Science. AMS Press, New York, 2002, vol. 61, pp. 163–184.

[2] U. Roshan, B. Moret, T. Williams, and T. Warnow, "Rec-I-DCM3: A fast algorithmic technique for reconstructing large phylogenetic trees," in *Proc. 3rd IEEE Computational Systems Bioinformatics Conf. CSB'04*. IEEE Press, Piscataway, NJ, 2004, pp. 98–109.

[3] L. Nakhleh, D. Miranker, F. Barbancon, W. Piel, and M. Donoghue, "Requirements of phylogenetic databases," in *Proc. 3rd IEEE Symp. on Bioinformatics and Bioengineering BIBE'03*, 2003, pp. 141–148.

[4] M. Sanderson, M. Donoghue, W. Piel, and T. Eriksson, "TreeBASE: A prototype database of phylogenetic analyses and an interactive tool for browsing the phylogeny of life," *Amer. J. Bot.*, vol. 81, no. 6, p. 183, 1994.

[5] D. Swofford, G. Olsen, P. Waddell, and D. Hillis, "Phylogenetic inference," in *Molecular Systematics*, D. Hillis, B. Mable, and C. Moritz, Eds. Sinauer Assoc., Sunderland, MA, 1996, pp. 407–514.

[6] M. Sanderson, "r8s: inferring absolute rates of evolution and divergence times in the absence of a molecular clock," *Bioinformatics*, vol. 19, pp. 301–302, 2003.

[7] D. Maddison, "The discovery and importance of multiple islands of most-parsimonious trees," *Syst. Zool.*, vol. 40, no. 3, pp. 315–328, 1991.

[8] W. Day, "Optimal algorithms for comparing trees with labeled leaves," *J. Classification*, vol. 2, no. 1, pp. 7–28, 1985.