Performance of Supertree Methods on Various Dataset Decompositions

Bernard M.E. Moret* Usman Roshan† Tandy Warnow†
Tiffani L. Williams*

Abstract

Many phylogenetic reconstruction methods attempt to solve hard optimization problems, such as Maximum Parsimony (MP) and Maximum Likelihood (ML); consequently, these methods are limited severely by the number of taxa that they can handle in a reasonable time frame. A divide-and-conquer strategy, based upon dividing a dataset into overlapping subsets, constructing trees on these subsets, and then merging them into a tree on the full dataset, is one promising approach to overcoming the inherent computational difficulties in large-scale phylogenetic reconstruction. Such a method combines a careful data decomposition with the general approach of supertree methods, which assemble a single large tree from a collection of overlapping subtrees. In this paper, we compare a standard supertree method, matrix parsimony (MRP), to a supertree method (strict consensus merger, or SCM) designed to work in tandem with our dataset decomposition method (a disk-covering method, or DCM), on both random decompositions and DCM-produced decompositions. We examine the performance of these methods with respect to maximum parsimony scores, topological accuracy, and running time. We find that our DCM+SCM method outperforms the others with respect to all these criteria.

Contact: usman@cs.utexas.edu
Web page: http://www.cs.utexas.edu/users/usman/supertree
Keywords: disk covering methods, maximum parsimony, matrix parsimony, supertree

1 Introduction

Many phylogenetic reconstruction methods attempt to solve difficult optimization problems such as Maximum Parsimony (MP) and Maximum Likelihood (ML) (Felsenstein, 1981; Hillis *et al.*, 1996; Foulds & Graham, 1982; Steel, 1994), with the result that a biologically acceptable phylogenetic analysis can take years to complete on only a few hundred taxa (see (Chase *et al.*, 1993; Rice *et al.*, 1997) for a well known example). The computational requirements of large-scale phylogenetics have motivated systematists to develop alternative techniques for reconstructing evolutionary trees,

*Department of Computer Science, U. of New Mexico, moret,tlw@cs.unm.edu
†Department of Computer Science, U. of Texas at Austin, usman,tandy@cs.utexas.edu

such as so-called "supertree methods." Supertree methods combine smaller, overlapping subtrees into a larger tree; they can thus use existing, published reconstructions on which the community agrees as well as combine the outcomes of reconstruction on decompositions of a large dataset. The most popular supertree method is Matrix Representation Parsimony (MRP) (Baum, 1992; Ragan, 1992), which has been used in a number of phylogenetic studies; Bininda-Emonds and colleagues examined the topological accuracy of reconstructions made with several variants of MRP in simulations (Bininda-Emonds, 2002; Bininda-Emonds & Sanderson, 2001), while Page compared it to a different method (based on mincuts) (Page, 2002). Beyond these three studies, little has been done to investigate the relative performance of supertree methods in an experimental setting.

In this study, we use two different supertree methods, MRP and a method, Strict Consensus Merger (SCM) devised specifically to complement the DCM-2 (Disk-Covering Method) dataset decomposition technique (Huson *et al.*, 1999b). As a control method, we use a random decomposition into overlapping subsets. We thus obtain three different methods: DCM2+SCM, DCM2+MRP, and Random+MRP (we cannot combine Random with SCM, since the latter relies on specific features of the DCM2 decomposition). We compare these methods on both real and simulated datasets that range from 100 to more than $1,500$ taxa. Our study shows that DCM2+SCM outperformed the other two methods in terms of running time, topological accuracy, and maximum parsimony scores.

The rest of the paper is organized as follows. We describe our supertree algorithms in Section 2. We explain our experimental design, datasets, and model trees, in Section 3. Finally, we discuss our results in Section 4.

2 Divide-and-Conquer Phylogenetic Methods

Each of the divide-and-conquer methods we study use four steps to construct a supertree from a given dataset:

- Step 1: Decompose the dataset into smaller, overlapping subsets.
- Step 2: Construct phylogenetic trees on the subsets using the desired "base" phylogenetic reconstruction method.
- Step 3: Merge the subtrees into a single (not necessarily fully resolved) tree on the entire dataset.
- Step 4: Refine the resultant tree to produce a binary tree.

Steps 2 and 4 are the same in all of our algorithms: we use a fast heuristic search for maximum parsimony (MP) as the "base method" to construct the subtrees, and the same heuristic search for MP to refine the merged supertree into a binary (i.e., bifurcating) tree. Thus, our methods differ only in terms of how they perform Steps 1 and 3.

2.1 Data Decomposition

DCM2 decomposition The Disk-Covering Method (DCM) (Huson *et al.*, 1999a,b; Warnow *et al.*, 2001; Nakhleh *et al.*, 2001) is a family of meta-methods for phylogenetic reconstruction,

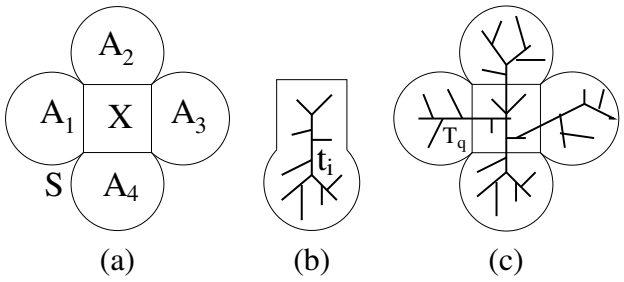


Figure 1: The three steps in Phase I of DCM2: (a) compute a clique separator X for S (relative to the triangulation of $G(d,q)$), producing subproblems $A_1 \cup X, A_2 \cup X, \ldots, A_r \cup X$; (b) compute a tree t_i for each subproblem $A_i \cup X$ using the chosen base method; and (c) merge the computed subtrees to obtain supertree T_q.

operating in conjunction with a "base method" such as maximum parsimony or neighbor-joining. In (Huson *et al.*, 1999b) we described a DCM, called DCM2, for speeding up searches for Maximum Parsimony trees; we use that same DCM here. The input to DCM2 is a set $S = \{s_1, \ldots, s_n\}$ of n aligned biomolecular sequences, a matrix d containing an estimate of their interleaf distances, and a particular $q \in \{d_{ij}\}$. DCM2 then computes a threshold graph $G(d,q)$, as follows: the vertices of $G(d,q)$ are the taxa, $s_1, s_2, \ldots, s_n$, and the edges of $G(d,q)$ are those pairs (s_i, s_j) obeying $d_{i,j} \leq q$. A greedy heuristic is used to triangulate this graph so as to minimize the weight of the heaviest edge added; let G^* denote the triangulation of $G(d,q)$. We then compute a clique separator X of G^*, which minimizes the maximum size of any set defined as $X \cup A_i$, where A_i is one of the components of $G^* - X$ (Figure 1). Once we have this separator, we compute trees on each subproblem (the sets $X \cup A_i$), then merge the trees into a single tree with SCM. DCM2 requires a particular threshold value, q, which influences the size of the subproblems examined (larger values tend to give larger subproblems for DCM2). We looked at two thresholds: d_0, the smallest threshold for which the threshold graph is connected, and d_4, which is the 5th threshold of evenly spaced values between d_0 and $d_9 = max\{d_{ij}\}$. In this paper we present the results for DCM2 at d_0, however our web page contains additional data at d_4.

Random decomposition Our RANDOM method decomposes the dataset into random overlapping subsets, using three parameters: the number x of subproblems, the desired minimum size y of each subproblem, and the desired minimum size z of the setwise intersection of all subsets. Let $n = |S|$ be the number of taxa to be distributed among the subsets. The x subsets are populated as follows. First, z taxa are randomly selected, and each of these z taxa are placed in each of the subsets. For each subset, we then randomly select an additional $(y - z)$ taxa from the remaining $(n - z)$ taxa. Finally, if any taxa have not been placed in any particular subset, we then add these taxa randomly to subsets.

2.2 Merging Subtrees

Matrix representation parsimony The MRP approach encodes a set $\mathcal{T}$ of trees as binary characters with missing values (i.e., "partial binary characters") as follows. Let S denote the full set of taxa, and let T be one of the trees in the set $\mathcal{T}$; thus T has leaf set $S_0 \subset S$. Let e be an arbitrary edge in T. The deletion of the edge e from T induces a bipartition π_e on S_0 into A and B. Now

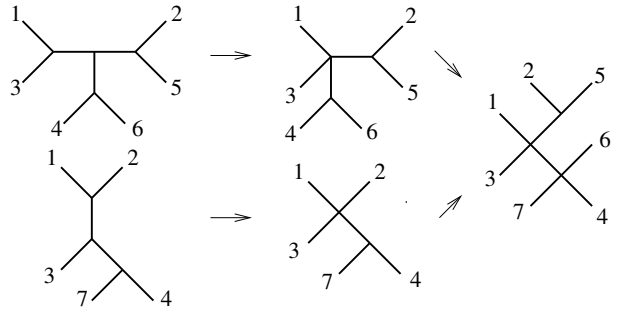


Figure 2: Merging two trees together, by first transforming them (through edge contractions) so that they induce the same subtrees on their shared leaves.

define a character c_e on all of S by extending π_e with $c_e(s) = 1$ for $s \in A$, $c_e(s) = 2$ for $s \in B$, and $c_e(s) =?$ for $s \notin S_0$. The set $C(\mathcal{T}) = \{c_e \mid \exists T \in \mathcal{T},\ e \in E(T)\}$ is thus a set of partial binary characters that encodes all topological constraints of the trees in $\mathcal{T}$. Furthermore, if a supertree exists which exactly satisfies all the constraints in the set $\mathcal{T}$, then it will have optimal parsimony score with respect to $C(\mathcal{T})$. Hence, the supertree will be the maximum parsimony solution for S on this set of characters.

Strict consensus merger The Strict Consensus Merger (SCM) method takes two trees t and t' on possibly different leaf sets, identifies the set of leaves X that they share, and modifies t and t' through a minimal set of edge contractions, so that they induce the same subtree on X—called the "backbone." Once they are modified in this way, the two trees can be merged: see Figure 2. A collection of more than two trees is merged sequentially, in arbitrary order (the order is irrelevant). The SCM of two trees may not be unique: it may be the case that some piece of each tree attaches onto the same edge of the backbone—a *collision*. When a collision occurs, we allow any attachments to take place so that the resultant tree agrees with the input trees (after the original edge contraction).

2.3 Phylogenetic Base Method and OTR

For the base method, we use a fast heuristic search implemented in `PAUP*4.0` for MP in which we use one random sequence addition order, and do TBR branch-swapping saving the best tree found until we reach a local optima. The Optimal Tree Refinement (OTR) problem is NP-hard, so we again use a fast heuristic search implemented in `PAUP*4.0`. We pass the unresolved supertree as a constraint tree to PAUP* and receive a (usually) binary tree refining the constraint tree, one that tends to optimize the parsimony score. We used the following PAUP*4.0 (Swofford, 1996) command:

```
constraints c1 (monophyly) = <constraint_tree>;
set criterion=parsimony maxtrees=1 increase=no;
hsearch start=stepwise addseq=random swap=tbr hold=1 nreps=1
        constraints=c1 enforce=yes;
```

3 Experiments

3.1 Overview

We had two conjectures to test: (i) that careful decomposition of the dataset is crucial to the success of supertree methods and that the DCM methods offer such a careful decomposition; and (ii) that the strict consensus merger developed as part of DCM is superior to MRP as a supertree assembly tool. We tested these conjectures on real and simulated datasets.

3.2 Biological and Simulated Datasets

Biological datasets We obtained 10 biological datasets from various sources, listed below with the number of sequences, their lengths, and the maximum p-distance (normalized Hamming distance) between any two sequences in the set.

1. A set of 218 aligned small subunit (SSU) ribosomal RNA sequences (4182 characters) for a phylogenetically representative set of prokaryotes (Maidak *et al.*, 2001) (max p-distance = 0.415).
2. A set 328 ITS sequences (946 characters) from the flowering plant Asteracaeae obtained from the Gutell Lab at the Institute for Cellular and Molecular Biology, The University of Texas at Austin (max p-distance = 0.524).
3. A set of 369 aligned 16S rRNA sequences (2726 characters) for four different classes of Bacteria: Aquificae, Thermotogae, Deinococcus-Thermus, and Nitrospira (Maidak *et al.*, 2001) (max p-distance = 0.314).
4. A set of 388 aligned 16S rRNA sequences (2497 characters) for three different classes of Bacteria: Planctomycetes, Chlamydiae, and Verrucomicrobia (Maidak *et al.*, 2001) (max p-distance = 0.456).
5. A set of 475 aligned small subunit Eukaryotic rRNA sequences (3277 characters) (Wuyts *et al.*, 2002) (max p-distance = 0.575).
6. A set of 556 aligned 16S rRNA sequences (2402 characters) for the Spirochaetes class of Bacteria (Maidak *et al.*, 2001) (max p-distance = 0.31).
7. A set of 567 "three gene: rbcL, atpB, and 18s" aligned sequences (2497 characters) of mostly angiosperms and and a carefully selected group of gymnosperms available from `www.cladistics.com/tree3GeneLink.htm` (max p-distance = 0.317).
8. A set of 590 aligned small subunit Eukaryotic rRNA sequences (1962 characters) (Wuyts *et al.*, 2002) (max p-distance = 0.382).
9. A set of 695 aligned 16S rRNA sequences (2550 characters) for the Cyanobacteria class of Bacteria (Maidak *et al.*, 2001) (max p-distance = 0.219).
10. A set of 778 aligned small subunit Mitochondrial rRNA sequences (1836 characters) (Wuyts *et al.*, 2002) (max p-distance = 0.656).

DCM-based methods require a distance matrix to compute the threshold graph—although the distance matrix does not play any role beyond that step. We used the Kimura 2-parameter plus Gamma (K2P+γ) distance-correction formula (Kimura, 1980; Yang, 1993) to compute a distance matrix on each biological dataset and we (arbitrarily) set the parameters $\kappa = 2$ and $\alpha = 1$. Because the only effect is on the threshold graph, we do not need the model to fit the data particularly well—and yet it is possible that a better fit would lead to better reconstructions. In that sense, our results for the DCM-based reconstructions are suboptimal.

Biologically-based model trees: We selected model trees that were based upon phylogenetic reconstructions of large biological datasets, so that we could explore topological accuracy as well as maximum parsimony scores and running time. We simulated evolution down these model trees under the K2P+γ model ($\alpha = 1, \kappa = 2$), and used corrected distances in each of our divide-and-conquer methods.

- Archaea 107 tree: Our first biological model tree was obtained from (Maidak *et al.*, 2001), which was constructed using Weighbor from RNA sequences (avg. branch length = 0.143)
- rbc*L* 500 tree: The second biological model tree was constructed using a parsimony analysis of a collection of 500 rbc*L* gene (DNA) sequences by (Rice *et al.*, 1997) (avg. branch length = 0.278).
- Mitochondria 1503 tree: Lastly, we used a 1503 taxa tree obtained from (Maidak *et al.*, 2001) (avg. branch length = 0.227).

Random birth death trees We generated 400 taxa model trees under the birth-death distribution so that we could explore topological accuracy in addition to maximum parsimony scores. We scaled the height of each model tree to five different heights (low to high): 0.1, 0.25, 0.5, 1, and 2, and multiplied the lengths of the tree edges by a random variable to deviate them from ultrametricity. We generated DNA sequences of length 2000 on these trees using the Kimura-2-Parameter plus Gamma model of sequence evolution ($\alpha = 1, \kappa = 2$).

3.3 Implementation and Platforms

Our DCM implementations are a combination of C++ (which uses LEDA 4.3) and Perl scripts; they were originally written by Daniel Huson and further expanded by us. RANDOM is also a combination of C++ and Perl scripts and was written by us. To run the MP heuristics used in FASTMP, MRP and OTR, we used `PAUP*4.0`. Our experiments were run on modest Pentium 500 MHz machines under Debian Linux.

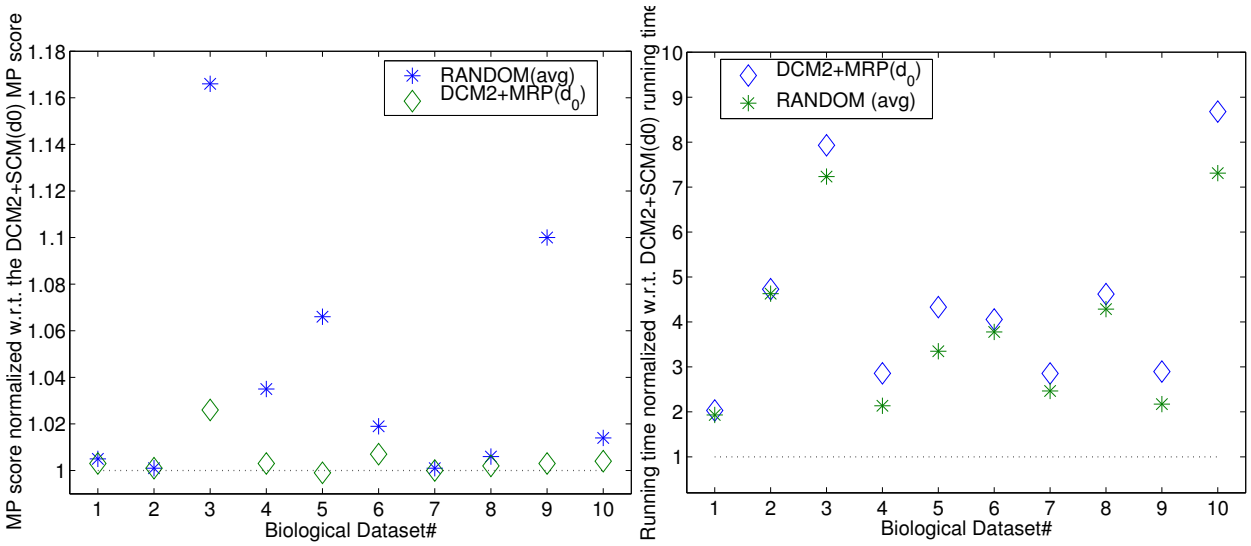
4 Results

We now examine the relative performance of DCM-based methods to RANDOM-based methods. In our experiments, the supertree method used by RANDOM was always MRP, so in this experiment we limit our attention to those methods using MRP for supertree reconstruction. We include $DCM2 + MPR(d_0)$ to represent a *DCM*2 analysis using *MRP*, and to enable us to investigate the

improvement obtained by using SCM instead of MRP in assembling a supertree. We also defined the parameters for RANDOM so as to produce the same number of datasets, of approximately the correct sizes and overlap as in $DCM2(d_0)$ on all the data we examined (SCM and MRP do not affect the dataset decomposition).

We ran a medium heuristic search for MRP using TBR search, with ten random sequence addition orders, and saving the 100 best trees, on the DCM-produced subtrees, measured the running times, and used those times as limits on the time spent by the same heuristic search for MRP on the subtrees obtained on the random subsets produced by RANDOM; however, for the simulated data on random birth death trees we did not set such a time limit.

Due to space limitations we cannot show all our experimental results; see our web page for the full set of figures and tables. All of our studies, both on real and simulated data, showed that DCM2+SCM outperformed the other methods (DCM2+MRP or RANDOM+MRP) at both thresholds, with respect to topological accuracy (on simulated data), maximum parsimony scores, or running time. This observation held on all the datasets we examined, whether real or simulated. The simulation study also showed that topological accuracy was also better for DCM2+SCM than for RANDOM+MRP and DCM2+MRP, at either threshold examined.

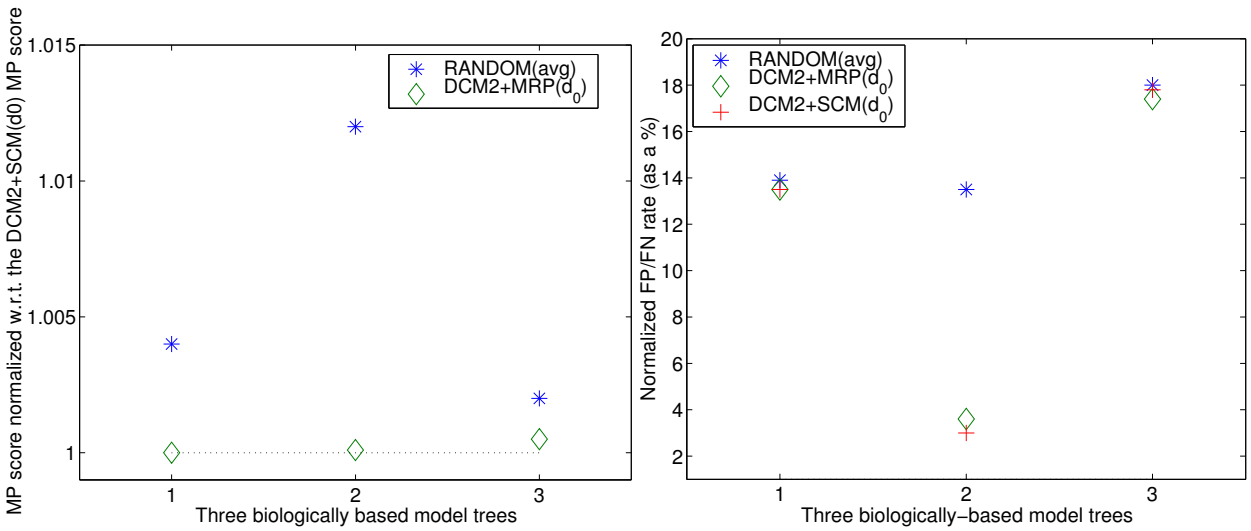


(a) Comparison of MP scores *(b) Comparison of running times*

Figure 3: MP scores and running time comparisons for DCM-based methods and RANDOM (averaged over 20 runs), normalized with respect to the DCM2+SCM(d_0) MP scores on each of the ten biological datasets.

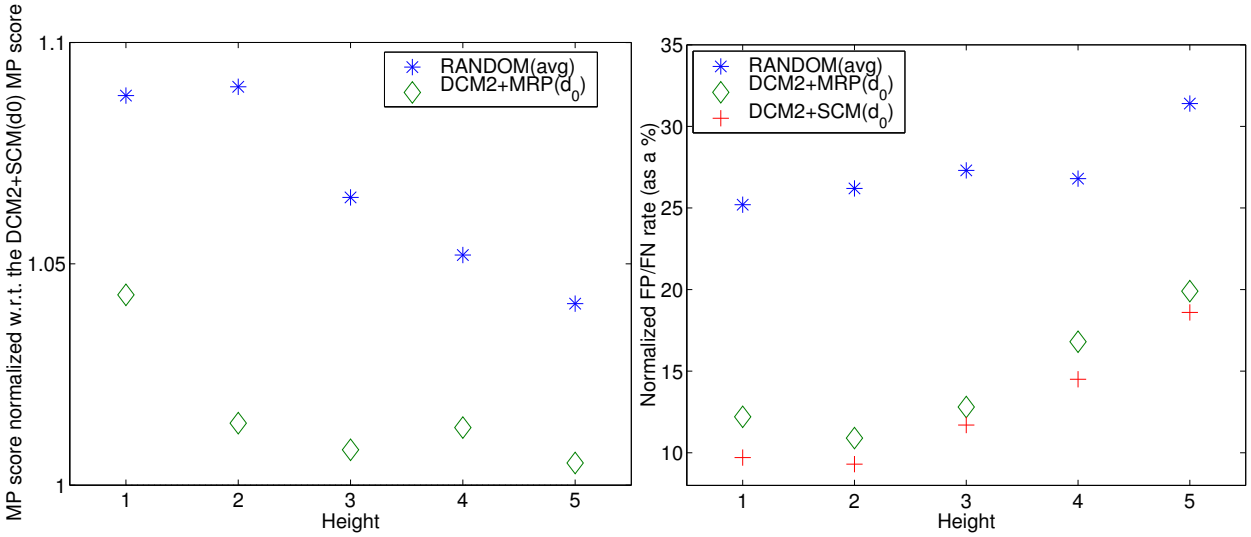
Figures 3, 4, and 5 are suggestive of some relative performance improvements of the various DCM2 methods over RANDOM+MRP, at least for dataset decompositions that are similar to those obtained by DCM2 at threshold d_0. First, at this threshold, when we consider MP scores, $DCM2+SCM$ outperforms $DCM2+MRP$ and $RANDOM+MRP$ has the worst performance. This suggests that DCM2 style decompositions may be better than RANDOM decompositions, and that when applicable, *SCM* may be better than MRP (SCM cannot be used on arbitrarily defined subproblems, since it works from a triangulated graph.) Furthermore, the running time of $DCM2+SCM$ is less than that of the other methods (see Figure 3).

Note that MRP is time consuming, when compared to the SCM phase. The time gained, on the order of hours on many of these datasets, could also be used to conduct a more thorough parsimony



(a) Comparison of MP scores , normalized w.r.t. DCM2+SCM(d_0) *(b) Comparison of FP/FN rates*

Figure 4: Comparison of MP scores and FP/FN rates of DCM-based methods and RANDOM (averaged over 20 runs). Data generated from three biological trees are shown.



(a) Comparison of MP scores , normalized w.r.t. DCM2+SCM(d_0) *(b) Comparison of FP/FN rates*

Figure 5: Comparison of MP scores and FP/FN rates of DCM-based methods and RANDOM (averaged over 10 runs). Data generated from random birth death trees with 400 taxa, sequence length 2000, deviation 4, and various heights are shown.

search in the subtrees or in the OTR phase of DCM2+SCM—we did not run such an equal-time comparison, but it is to be expected that it would widen the gap in parsimony scores between the OTR tree returned from DCM2+SCM and the other two.

5 Summary and Conclusions

We set out to understand the potential for supertree methods with respect to improving the speed of maximum parsimony searches, and to learn, in particular, if methods such as our Disk-Covering Methods, or "DCMs" (which have a particular decomposition strategy, and then combine subtrees in a particular way) were able to outperform a random decomposition into subproblems followed by the MRP method for combining subtrees. As we conjectured, we observed that there were clear advantages to be obtained from the use of our DCMs. In all our datasets, both real and synthetic, DCM2+SCM outperformed RANDOM+MRP and DCM2+MRP, with respect to maximum parsimony scores and running time; topological accuracy was also improved by using DCM2+SCM, as our simulation studies showed.

Our experiments are preliminary in that we have examined only a small part of the parameter space of phylogenetic trees; however, these studies show the potential for divide-and-conquer methods to be improved by the use of novel dataset decomposition strategies, and simple subtree merger techniques.

Several questions present themselves for further research. First, how well do divide-and-conquer strategies compare to global searches for maximum parsimony (in which the search is applied to the full dataset)? Secondly, the improvement obtained in going from the smallest possible threshold (d_0) to a larger threshold (d_4) (see web page) suggests that alternate decomposition techniques may still be found, so that better divide-and-conquer strategies may still be found. And since dataset decompositions affect the quality of supertree methods, how should biologists define subproblems? Clearly, random subsets are not how biologists will define their subsets, and so studies, such as this one, will need to be redone with biologically driven definitions of subproblems.

Finally, we conclude by observing that we studied the performance of supertree methods with respect to maximum parsimony scores, and this study should be redone with respect to other criteria, such as maximum likelihood, so that we can observe more general trends about the performance of supertree methods.

6 Acknowledgments

This work was supported by the National Science Foundation under grants ACI 00-81404 (Moret), DEB 01-20709 (Moret and Warnow), EIA 01-13095 (Moret), EIA 01-13654 (Warnow), EIA 01-21377 (Moret), and EIA 01-21680 (Warnow), by the David and Lucile Packard Foundation (Warnow), and by an Alfred P. Sloan Foundation Postdoctoral Fellowship in Computational Molecular Biology, DOE grant DE-FG03-02ER63426 (Williams).

References

Baum, B. (1992). Combining trees as a way of combining data sets for phylogenetic inference, and the desirability for combining phylogenetic trees. *Taxon*, **41**, 3–10.

Bininda-Emonds, O. R. P. (2002). MRP Supertree Construction in the Consensus Setting. *To appear in DIMACS*.

Bininda-Emonds, O. R. P. & Sanderson, M. J. (2001). Assessment of the accuracy of matrix representation with parsimony analysis supertree construction. *Syste. Biol*, **50**, 565–579.

Chase, M., Soltis, D., Olmstead, R., Morgan, D., Les, D., Mishler, B., Duvall, M., Price, R., Hillis, H., Qiu, Y.-L., Kron, A., Retig, J., Conti, E., Palmer, J., Manhart, J., Sytsma, K., Michaels, H., Kress, W., Karol, K., Clark, W., Hedrn, M., Gaut, B., Jansen, R., Kim, K.-J., Wimpe, C., Smith, J., Furnier, G., Strauss, S., Xiang, Q.-Y., Plunkett, G., Soltis, P., Swensen, S., Williams, S., Gadek, P., Quinn, C., Eguiarte, L., Golenberg, E., Jr, G., Graham, S., Barett, S., Dayananadan, S. & Albert, V. (1993). Phylogenetics of sees plants: An analysis of nucleotide sequences from the plastid gene *rbcL*. *Annals of the Missouri Botanical Garden*, **80**, 528–580.

Felsenstein, J. (1981). Evolutionary trees from DNA sequences: A maximum likelihood approach. *Journal of Molecular Evolution*, **17**, 368–376.

Foulds, L. R. & Graham, R. L. (1982). The Steiner problem in phylogeny is NP-complete. *Advances in Applied Mathematics*, **3**, 43–49.

Hillis, D., Moritz, C. & Mable, B. (1996). Molecular Systematics. Sinauer Pub., Boston.

Huson, D., Nettles, S. & Warnow, T. (1999a). Disk-covering, a fast-converging method for phylogenetic tree reconstruction. *Comput. Biol.*, **6**, 369–386.

Huson, D., Vawter, L. & Warnow, T. (1999b). Solving large scale phylogenetic problems using DCM2. In *ISMB 99*. pp. 118–129.

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.

Maidak, B., Cole, J., Lilburn, T., Jr, C. P., Saxman, P., Farris, R., Garrity, G., Olsen, G., Schmidt, T. & Tiedje, J. (2001). The RDP-II (Ribosomal Database Project). *Nucleic Acids Res*, **29(1)**, 173–4.

Nakhleh, L., Roshan, U., St. John, K., Sun, J. & Warnow, T. (2001). Designing fast converging phylogenetic methods. In *Proc. 9th Int'l Conf. on Intelligent Systems for Molecular Biology (ISMB01)*, volume 17 of *Bioinformatics*. Oxford U. Press, pp. S190–S198.

Page, R. (2002). Modified mincut supertrees. In Guigo, R. & Gusfield, D., (eds.) *Proc. 2nd Int'l Workshop Algorithms in Bioinformatics (WABI'02)*. Springer-Verlag, pp. 537–551.

Ragan, M. (1992). Phylogenetic inference based on matrix representation of trees. *Mol. Phylogenet. Evol.*, **1**, 53–58.

Rice, K., Donoghue, M. & Olmstead, R. (1997). Analyzing large datasets: *rbcL* 500 revisited. *Systematic Biology*, **46(3)**, 554–563.

Steel, M. A. (1994). The maximum likelihood point for a phylogenetic tree is not unique. *Systematic Biology*, **43**, 560–564.

Swofford, D. L. (1996). PAUP*: Phylogenetic analysis using parsimony (and other methods). Sinauer Associates, Underland, Massachusetts, Version 4.0.

Warnow, T., Moret, B. & John, K. S. (2001). Absolute convergence: true trees from short sequences. *Proceedings of ACM-SIAM Symposium on Discrete Algorithms (SODA 01)*, 186–195.

Wuyts, J., de Peer, Y. V., Winkelmans, T. & Wachter, R. D. (2002). The European database on small subunit ribosomal RNA. *Nucleic Acids Res*, **30**, 183–185.

Yang, Z. (1993). Maximum likelihood estimation of phylogeny from DNA sequences when substitution rates differ over sites. *Mol. Biol. Evol.*, **10**, 1396–1401.