COMPUTING LARGE PHYLOGENIES WITH STATISTICAL METHODS: PROBLEMS & SOLUTIONS

Stamatakis A.P., Ludwig T., Meier H.

Department of Computer Science, Technische Universität München
Department of Computer Science, Ruprecht-Karls Universität Heidelberg
e-mail: **stamatak@in.tum.de**
[*]Corresponding author

Keywords: evolution, phylogenetics, maximum likelihood, large phylogenies

Summary

The computation of ever larger as well as more accurate phylogenetic trees with the ultimate goal to compute the “tree of life” represents a major challenge in Bioinformatics. Statistical methods for phylogenetic analysis such as maximum likelihood or bayesian inference, have shown to be the most accurate methods for tree reconstruction. Unfortunately, the size of trees which can be computed in reasonable time is limited by the severe computational complexity induced by these statistical methods.
However, the field has witnessed great algorithmic advances over the last 3 years which enable inference of large phylogenetic trees containing 500-1000 sequences on a single CPU within a couple of hours using maximum likelihood programs such as RAxML and PHYML. An additional order of magnitude in terms of computable tree sizes can be obtained by parallelizing these new programs.
In this paper we briefly present the MPI-based parallel implementation of RAxML (Randomized Axelerated Maximum Likelihood), as a solution to compute large phylogenies. Within this context, we describe how parallel RAxML has been used to compute the –to the best of our knowledge- first maximum likelihood-based phylogenetic tree containing 10.000 taxa on an inexpensive LINUX PC-Cluster.
In addition, we address unresolved problems, which arise when computing large phylogenies for real-world sequence data consisting of more than 1.000 organisms with maximum likelihood, based on our experience with RAxML. Finally, we discuss potential algorithmic and technical enhancements of RAxML within the context of future work.
Availability: **wwwbode.in.tum.de/~stamatak**

Introduction

The inference of large phylogenetic trees based upon statistical models of nucleotide substitution is computationally intensive since the number of potential alternative tree topologies grows exponentially with the number of sequences and due to the high computational cost of the likelihood evaluation function for each individual topology. Although this has not been demonstrated to date, it is widely believed that maximum likelihood-based phylogenetic analysis is an NP-complete problem.
Therefore, progress in this field, in terms of gain in several orders of magnitude in conjunction with inexpensive hardware requirements, is rather achieved by algorithmic optimizations and introduction of new heuristics than by brute-force allocation of all available computational resources. E.g. a large and expensive grid of supercomputers has been used to conduct one of the most computationally intensive phylogenetic analyses to date

based on the relatively slow and old parallel fastDNAml (Stewart et al. 2001) code within the framework of the HPC challenge at the 2003 Supercomputing Conference (for details see **www.sc-conference.org/sc2003/tech_hpc.php**). Despite the unchallenged technical success the extreme computational effort could have been avoided by using more recent algorithms which execute approximately 50 times faster than fastDNAml and yield better results at the same time.

In a survey conducted by T.Williams et al. (2003) its has been demonstrated that MrBayes (Huelsenbeck et al., 2001a), an implementation of bayesian phylogenetic inference based on the Metropolis-Coupled Markov Chain Monte-Carlo technique, appears to be the currently fastest and most accurate program for phylogenetic inference. However, this survey is based entirely on simulated data, which can potentially generate misleading results.

More recently, Guidon et al. (2003) released a program called PHYML which is equally accurate and significantly faster than MrBayes and some of the most popular or efficient maximum likelihood programs like MetaPIGA (Lemmon et al. 2002), PAUP (Swofford et al. 2004), treepuzzle (Schmidt et al. 2001) and fastDNAml (Olsen et al. 1994).

Thus, PHYML and MrBayes represent the -to the best of our knowledge- currently fastest and most accurate phylogeny programs. In a recent paper (Stamatakis et al., 2004a) we describe the basic sequential implementation of RAxML (Randomized Axelerated Maximum Likelihood), which clearly outperforms PHYML and MrBayes on 9 large real world alignments containing 101 up to 1000 sequences both in terms of execution speed and final likelihood values, whereas it performs slightly worse on simulated data. Furthermore, in (Stamatakis et al., 2004) we also show that MrBayes fails to converge or converges significantly slower than RAxML and PHYML within reasonable time limits for some real world data sets. This result is not an argument against bayesian methods which are very useful and have experienced great impact (Huelsenbeck et al. 2001b) but for maximum likelihood methods which are still significantly faster and useful for verifying results of bayesian analyses.

Parallelization

The basic sequential algorithm of RAxML is outlined in (Stamatakis et al. 2004a). For parallelization we have chosen a coarse-grained approach which intends to minimze communication in order to allow for a http-based distributed implementation of RAxML as well (Stamatakis et al. 2004b).

The topology optimization process of RAxML is based upon a fast pre-evaluation of a large number of alternative topologies by application of the subtree rearrangement technique, which is also known as subtree pruning & re-grafting. The parallel code is based on a simple master-worker architecture, where the master maintains the currently best tree and distributes work by subtree IDs which are represented by simple integer values. Each subtree is then individually rearranged within the currently best tree by a worker. When a rearrangement step has been completed, i.e. all subtrees of the current tree have been rearranged, the best 20 (or # of workers, whichever is higher) trees obtained from this step are gathered by the master. The master then redistributes those 20 trees to the workers for branch length optimization and commences a new cycle of subtree rearrangements with the updated best tree. This process is repeated until no better tree is found.

However, the sequential algorithm contains a closely-coupled step: the subsequent application of topological improvements (Stamatakis et al., 2004a) which is difficult to parallelize. Thus, we have chosen to introduce some non-determinism in the parallel program to solve this problem. The non-determinism in the parallel program leads to a traversal of tree space on different paths for each individual program execution. As demonstrated by experimental results this non-determinism does not impose serious restrictions on program performance and partially leads to even superlinear speedup values. In **Figure 1** we plot the average speedup values for a 1.000 taxon alignment which has been extracted form the ARB small subunit ribonucleic acid database (Ludwig et al. 2004) over 4 parallel RAxML runs on 4, 8, 16, and 32 2.66 GHz Xeon processors respectively. Due to the non-determinism of the parallel code we provide two types of speedup values: "Fair" speedup indicates the point of time at which the parallel code detects a tree which shows a better likelihood value than the final topology of the sequential execution and "normal" speedup indicates the standard definition accounting for execution time until termination.

Computation of a 10.000-taxon phylogeny with RAxML

In order to conduct a large and meaningful phylogenetic analysis with RAxML we extracted an alignment comprising 10.000 sequences including organisms of the three domains Eukarya, Bacteria, and Archaea from the ARB database. The computation of the 10.000-taxon tree was conducted using the sequential, as well as the parallel version of RAxML. One of the advantages of RAxML consists in the randomized generation of parsimony starting trees. Thus, we computed 5 distinct randomized parsimony starting trees sequentially along with the first 3-4 rearrangement steps on a small cluster of Intel Xeon 2.4GHz processors at our institute. This phase required an average of 112.31 CPU hours per tree.

Thereafter, we executed several subsequent parallel runs (due to job run-time limitations of 24 hrs) starting with the sequential trees on either 32 or 64 processors on the 2.66GHz cluster mentioned above. The parallel computation required an average of approximately 1.600 accumulated CPU hours per tree. The best likelihood obtained for the 10.000 taxa was -949570.16 the worst -950047.78 and the average -949867.27.

PHYML reached a likelihood value of -959514.50 after 117.25 hrs on a 64-bit Itanium2 processor. Note, that the parsimony starting trees computed with RAxML showed likelihood values ranging between -954579.75 and -955308.00. The average time required for computing those starting trees on the Xeon processor was 10.99 hrs. Since bootstrapping is not feasible for this large data size and in order to gain some basic information about similarities among the 5 final trees we built a majority-rule consensus tree with consense (Jermiin et al. 1997). The consensus tree has 4777 bifurcating inner nodes which appear in all 5 trees, 1046 in 4, 1394 in 3, 1323 in 2, and 1153 in only 1 tree (average: 3.72). The results from this large phylogenetic analysis including all final trees as well as the consensus tree are available at: **wwwbode.cs.tum.edu/~stamatak**.

The final version of this paper will also include a biological analysis of the 10.000-taxon phylogeny.

Problems

Several new problems arise within the context of computation of large trees. An important observation is that memory consumption becomes critical, e.g. MrBayes and PHYML fail to execute for the 10.000-taxon alignment on a 32 bit processor with 4MB of main memory due to excessive memory requirements. Moreover, MrBayes could not be ported to a 64 bit Itanium2 processor whereas PHYML finally required 8.8MB of memory. In contrast to MrBayes and PHYML, RAxML required only approximately 800MB for the 10.000-taxon alignment. Thus, phylogeny programs for computation of large trees need to be designed for low memory consumption, since 64 bit architectures also induce a significant additional cost factor. Furthermore, consense is apparently not able to handle more than 5 10.000-taxon trees since it constantly exited with an error message when executed with more than 5 input trees.

Another important problem which is often underestimated is tree visualization which requires novel concepts for displaying large trees. Information obtained by phylogenetic analysis becomes valuable and can be interpreted only if appropriate tools are available. An initial visualization of the 10.000-taxon phylogeny with ATV (Zmasek et al. 2001) demonstrated that this standard tool is completely inadequate for viewing large trees. In fact, 2-D and 3-D hyperbolic tree viewers such as Walrus or Hypertree have been proposed as a solution (for details see **www.caida.org/tools/vizualisation/walrus** and **hypertree.sourceforge.net**) for large trees and graphs, which we did not find very helpful in the specific case however.

Finally, the assignment of confidence values to large trees remains problematic since execution of typically 100 or 1.000 distinct inferences to obtain bootstrap values or build consensus trees does not appear to be computationally feasible at present. In addition, MrBayes which directly yields confidence values is presently too slow and requires an excessive amount of memory for this tree sizes.

Thus, apart from the necessary improvements of associated tools phylogeny programs still require to become faster and yield at least equally good trees at the same time. In addition, they should incorporate more complex and exact models of evolution. Those two basic directions of research represent controversial targets due to an apparent trade-off between speed and quality. More sophisticated models, such as for example the General Time Reversible Model (GTR) of nucleotide substitution compared to HKY85 lead to significantly increased execution times.

Conclusion, Current & Future Work

Along with PHYML, RAxML currently represents one of the fastest and most accurate sequential phylogeny programs. In contrast to PHYML which is unfortunately only available as sequential program, we also provide a parallel MPI-based implementation of RAxML which has been used to conduct the –to the best of our knowledge- largest maximum likelihood analysis to date on a medium-sized PC cluster. Our program along with a benchmark set of best-known trees for real-world alignments, which have all been obtained by RAxML, is freely available at **wwwbode.in.tum.de/~stamatak**.

Currently, we are implementing model parameter optimization for the HKY85 and GTR models of nucleotide substitution in the new sequential version of RAxML, which will soon be released. Furthermore, we are working on a RAxML-based tool for splitting-up alignments into overlapping sub-alignments within the context of a divide-and-conquer supertree approach to phylogenetic inference.

Future work will cover the exploitation of the intrinsic fine-grained parallelism of RAxML on likelihood vector level by using Graphics Processor Units (GPUs) or other inexpensive hardware in a similar way as introduced by Krüger et al. (2003) for numerical simulations.

Finally, we will analyze the effect of the application of divide-and-conquer approaches and associated supertree methods to large maximum likelihood analyses in terms of final tree quality and execution times.

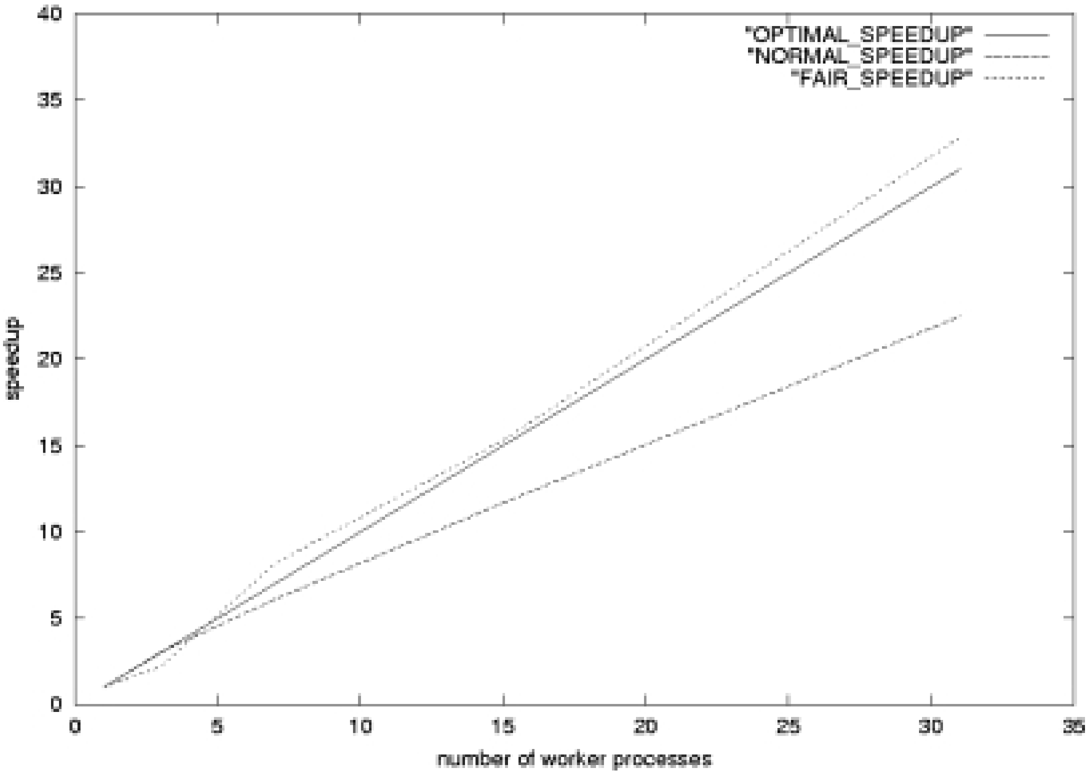

Figure 1: Normal and fair speedup values of parallel RAxML for a 1.000-taxon alignment on 4, 8, 16, and 32 Intel Xeon 2.66 GHz processors.

Acknowledgements

This work is sponsored under the project ID **ParBaum**, within the framework of the ``Competence Network for Technical, Scientific High Performance Computing in Bavaria'' (KONWIHR).

References

1. Guidon S., Gascuel O. (2003) A Simple, Fast, and Accurate Algorithm to Estimate Large Phylogenies by Maximum Likelihood. *Syst. Biol.* **52(5)**, 696-704.
2. Huelsenbeck J.P., Ronquist F., Nielsen R., Bollback J.P. (2001a) Bayesian Inference and its Impact on Evolutionary Biology. *Science* **294**, 2310-2314.
3. Huelsenbeck J.P., Ronquist F. (2001b) MrBayes: Bayesian Inference of Phylogenetic Trees. *Bioinformatics* **17(8)**, 754-755.
4. Jermiin L.S., Olsen G.J., Mengersen K.L. Easteal S. (1997) Majority-rule Consensus of Phylogenetic Trees Obtained by Maximum-Likelihood analysis. *Mol. Biol. Evol.* **14**, 1297-1302.
5. Krüger J., Westermann R. (2003) Linear Algebra Operators for GPU Implementations of Numerical Algorithms. *Proc. of SIGGRAPH2003.*

6. Lemmon A., Milinkovitch M. (2002) The Metapopulation Genetic Algorithm: An Efficient Solution for the Problem of Large Phylogeny Estimation. *Proc. Natl. Acad. Sci.* **99**, 10516-10521.
7. Ludwig W. et al., (2004) ARB : A Software Environment for Sequence Data. *Nucl. Acids Res.* **32(4)**, *1363-1371.*
8. Olsen G., Matsuda H., Hagstrom R., Overbeek R. (1994) fastDNAml: A Tool for Construction of Phylogenetic trees of DNA sequences using Maximum Likelihood. *Comput. Applic. Biosci.* **10**, 41-48.
9. PAUP project site: paup.csit.fsu.edu, visited March 2004.
10. Schmidt H.A., et al. (2002) TREE-PUZZLE: Maximum Likelihood Pylogenetic Analysis using Quartets and Parallel Computing. *Bioinformatics*, **18**, 502-504.
11. Stamatakis A.P., Ludwig T., Meier H. (2004a) New Fast and Accurate Heuristics for Inference of Large Phylogenetic Trees. To be published in *Proc. of IPDPS2004.* Preprint available on-line at: **wwwbode.in.tum.de/~stamatak/publications.html**
12. Stamatakis A.P., Ott M., Ludwig T., Meier H. (2004b) DRAxML@home: A Distributed Program for Computation of Large Phylogenetic Trees. To be published in *FGCS*.
13. Stewart C., et al. (2001) Parallel Implementation and Performance of fastDNAml – a Program for Maximum Likelihood Phylogenetic Inference. *Proc. of SC2001*.
14. Williams T.L, Moret B.M.E. (2003) An Investigation of Phylogenetic Likelihood Methods. *Proc. of BIBE2003*.
15. Zmasek C.M., Eddy M.R. (2001) ATV: Display and Manipulation of Annotated Phylogenetic Trees. *Bioinformatics* **17(4)**, 383-384.