PUZZLE/PPUZZLE: Using Parallel Computing for Maximum Likelihood Phylogenetic Analysis

Heiko A. Schmidt[1], **Korbinian Strimmer**[2], **Arndt von Haeseler**[3], **and Martin Vingron**[1]

[1] Deutsches Krebsforschungszentrum, Theoretische Bioinformatik, Im Neuenheimer Feld 280, D-69120 Heidelberg, Germany, {H.Schmidt|M.Vingron}@DKFZ-Heidelberg.de
[2] MIPS, Max-Planck-Institut für Biochemie, Am Klopferspitz 18a, D-82152 Martinsried/München, Germany strimmer@mips.biochem.mpg.de
[3] Max-Planck-Institut für Evolutionäre Anthropologie, Evolutionäre Genetik, Inselstr. 22, D-04103 Leipzig, Germany, arndt@eva.mpg.de

Phylogenetic reconstruction is used in several fields of genetic sequence analysis. It is applied not only for inferring relationships between species or within gene or protein families. It is also used for classifying organisms, e.g., viruses, and for reconstructing duplication events and their chronological order in genomes. Hence phylogenetic reconstruction plays an important role in understanding development of sequences and genomes.

There are several methods for inferring phylogenetic trees from genetic sequence data. Approaches using maximum likelihood (ML) based techniques belong to the most respected. Since Felsenstein (1981) popularized this approach several further methods have been published (e.g., Strimmer and von Haeseler 1996; Olsen *et al.* 1994).

The need for vast computational resources to compute ML trees remains problematic. This is largely due to the number of possible trees growing exponentially with the number of sequences used. To cope with this problem, several heuristics have been applied to reduce the number of trees, which have to be evaluated.

Here we use the quartet puzzling algorithm (Strimmer and von Haeseler, 1996), which first computes all $\binom{N}{4}$ quartet trees for N input sequences. Then, in a so-called puzzling step these quartet trees are composed into an overall tree. Since this step highly depends on the order of sequences added to the puzzle tree, many trees with different permutations of the input order are constructed. In the final step a strict majority rule consensus tree is built from these puzzle trees.

Since the construction of the quartet trees as well as the construction of the puzzle trees are independent of each other we parallelized these steps applying master/worker concepts. For the parallelization we used the Message Passing Interface (MPI, Snir *et al.* 1996). Since MPI has become the *de facto* standard in parallel computing using message passing, implementations of the MPI libraries are available on almost all parallel platforms from massively parallel computers to clusters of workstations (COW).

With the parallelization we tried to cope especially with the problems occuring on COWs like different speed or load of the connected workstations. This allows people who do not have access to supercomputing resources to use parallel software in their research.

Tests on different parallel platforms show that our implementation scales well on heterogeneous COWs as well as parallel computers like IBM SP2 or HP S-Class (SMP).

With the parallel version of PUZZLE we were able to reconstruct trees from more than 100 sequences in reasonable time on a cluster consisting of different SUN workstations with usual daytime workload. These parallel features will be part of the regular PUZZLE package.

Acknowledgments

This work is supported by *Deutsche Forschungsgemeinschaft* under grants no. VI 160/3-1 and Ha 1628/4-1.

References

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

Olsen, G. J., Matsuda, H., Hagstrom, R. and Overbeek, R. (1994) fastDNAml: A tool for construction of phylogenetic trees of DNA sequences using maximum likelihood. *CABIOS*, **10**, 41–48.

Snir, M., Otto, S. W., Huss-Lederman, S., Walker, D. W. and Dongarra, J. (1996) *MPI: The Complete Reference.* The MIT Press, Cambridge, Massachusetts.

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