

A Distributed Algorithm for Phylogenetics Inference

Felipe Fernandes Albrecht¹, Jomi Fred Hübner², and Alberto M. R. Dávila³

¹ Instituto Militar de Engenharia, Seção de Engenharia de Computação

² FURB, Departamento de Sistemas e Computação

³ Fundação Oswaldo Cruz, Instituto Oswaldo Cruz, Departamento de Bioquímica e Biologia Molecular

Abstract. The phylogenetics analysis that uses numerical taxonomy has a distance matrix, with the distances between the taxons. One of the numerical taxonomy techniques is the least squares. The least squares phylogenetics technique has an objective function that represents the inferred tree quality. This paper proposes a distribution of the method defined by Felsenstein, called: an alternating least squares method approach to inferring phylogenies from pairwise distances. This distribution aim the reduction of the execution time. The method proposed by Felsenstein has a execution time delayed when the set of taxons is very big. The proposal distributes the generated trees in the work processes and eliminates low quality trees. With this distribution, it is obtained a gain in the 50% in the execution time for a set containing 80 taxons, however, resulting a little reduction in the quality of the inferred trees.

1 Introduction

Molecular phylogenetics is the study of the evolutionary relations between different taxons, being they, the DNA, RNA, or proteic sequences. One technique used in molecular phylogenetics is the numerical taxonomy, where a distance matrix with the distances between the taxons is used. The techniques of the molecular phylogenetic inference employing numeric taxonomy are diverse: [2–5]. One of them, the least squares technique [4], has an objective function that represents the inferred tree quality. This function, shown in the equation 1, calculates the difference between the tree taxons distances and the input matrix taxons distances.

$$Q = \sum_{i=1}^n \sum_{j=1}^n w_{ij} (D_{ij} - d_{ij})^2 \quad (1)$$

To compute the tree branch lengths, Cavalli-Sforza[4] and Felsenstein[6] describes a method that uses a set of linear equations, but Felsenstein[1] says that this method needs a tree with a specific topology and the resolution of the linear equations can be a complicated and an expensive task.

Felsenstein [1] shows an alternative least squares method. This method infers the tree by steps, each one adding a new taxon in the tree, searching among all possible positions and choosing one that has the lower least squares. After choosing the best position and calculating the branch lengths, some optimizations are executed to decrease the tree least squares. A new set of trees is generated using the optimized tree of the previous step, and again it is chosen the tree with the lower least squares to perform the optimization. The iterations are repeated until all taxons of the input matrix are in the tree.

Analyzing this method, it is clear that it works like a search algorithm [7]. Where a data set is initially generated and is after a search among this set is performed to find the best data. In the case of the least squares method, the data set is all possible trees and the best data is the tree with the lower least square. The Figure 1 shows that at each algorithm step, a set of trees is created and the tree with lower last square is chosen. This selected tree is therefore used as the starting point for the trees created at the next step.

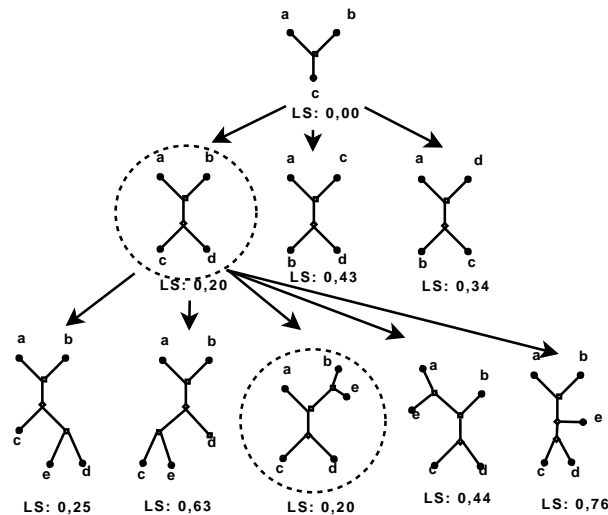


Fig. 1. Searching for the tree with lower least squares.

This algorithm can be distributed among several processes, where each process would be responsible for a tree set. Each process add new taxon in each tree of its set and in this way, creates new trees. Also, each process analysis its generated trees and eliminates those that has the worse least squares.

2 The Proposed Algorithm

The proposed algorithm is divided into two kinds of process: the scheduler and the workers. The scheduler creates the initial trees set and assigns a disjoint sub-

set of the set for each worker process. For this assignment, the scheduler creates all possible taxons triples and calculates the distances between the taxons that form it. Hence the scheduler gets the triples that has the lowers total distances and allocates a set of them for the workers. The size of the allocate is an user parameter.

For each algorithm iteration, the worker processes generate a new tree for each position of each tree where it is possible to insert a new taxon. After the generation of each tree, the worker calculates its least squares. When all the new trees was generated and its least squares calculated, the worker calculates the average \bar{x} and standard deviation σ of these least squares values. Each worker processes use these values to eliminate trees that have least square greater than the threshold defined as " $i.\bar{x} + j.\sigma$ ", where i and j values are users parameters. This way, at each iteration a tree set is generated by each worker process and some trees are eliminated. The reason for this elimination is that the successors of these trees unlikely will obtain the better final least squares.

After the new trees have been generated and the worst trees of each process removed, the worker processes sends to the scheduler a descriptor containing the trees identification and theirs least-squares. Hence, the scheduler calculates the average and standard deviation of all remained trees and sends a message to the process that created the tree informing that the tree has not a good least squares value accordingly to all trees and must be removed. The elimination process is done thus in two stages to not allow the exaggerated grow of the number of the trees and to avoid the exponential growth in the number of trees. These trees will be the base for the creation of new trees in the following iteration.

3 Implementation and Results

The implemented software, called `dleastquares`⁴, was written in C language and for inter-process communication, the MPI standard with its implementation LAM [8] was used. To test the implementation performance, eight distances matrix with hypothetical distances was created. `Dleastquares` and `kitch`, from PHYLIP [9] package, were evaluated and their executions times measured. The `kitch` software was executed at an Intel Pentium 4 3Ghz with 1 Gigabyte and the `dleastquares` at a cluster containing five of these computers. The `dleastquares` was executed with the following options: 4 initial triples by process and the minimum of 20 and maximum of 40 taxon by iteration.

The `dleastquares` and the `kitch` was executed 8 times and the matrix sizes varies from 10 taxons to 80 taxons. It is shown in figure 3 the execution time. Note that for a matrix with less than 50 taxons, the `kitch` performance is better. Otherwise, with matrices with more than 50 taxons, the `dleastquares` performance outperforms the `kitch`. With a matrix with 80 taxons, the `dleastquares` shows a time gain of 50%.

This distribution approach obtained a gain of 50% for a matrix with 80 taxons. However, a reduction in the quality of the inferred trees was produced

⁴ this software is freely available in <http://sourceforge.net/projects/distphylo>

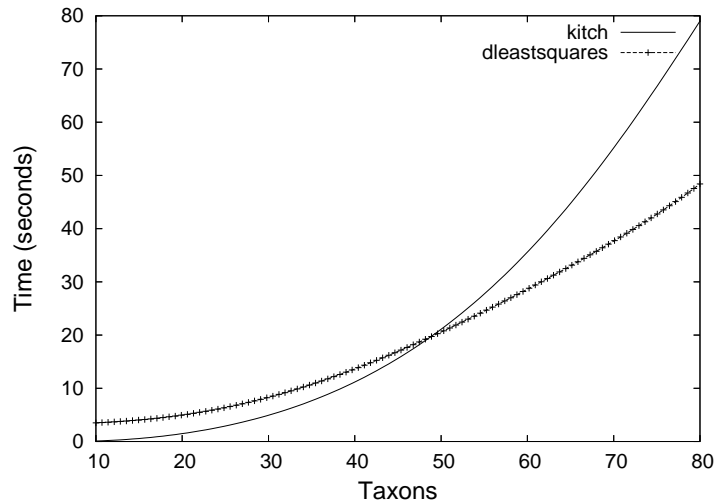


Fig. 2. Searching for the tree with lower least-squares.

because it is not done an exhausting search in all possible trees and all possible optimizations.

By this algorithm, it is possible to infer phylogenetic trees specifying parameters, like the analysed trees quantity at each iteration and the threshold for the tree elimination. Even not returning the best tree, the topology of the inferred tree shows similarity with trees inferred by others software.

References

- [1] Felsenstein, J: *An Alternating Least Squares Approach to Inferring Phylogenies from Pairwise Distances*. Systematic Biology, **46**: 101–111. 1997.
- [2] Sokal, R. R and P. H. A. Sneath: *Numerical Taxonomy*. W. H. Freeman, San Francisco, 1963.
- [3] Saitou, N. and M. Nei: *The neighbor-joining method: A new method for reconstructing phylogenetic trees*. Molecular Biology and Evolution **4**: 406–425. 1987.
- [4] Cavalli-Sforza LL, Edwards AWF: *Phylogenetic analysis: Models and estimation procedures*. Am J Hum Genet **19**: 233–257. 1967.
- [5] Fitch, W. M and E. Margolia: *Construction of phylogenetic trees*. Science **155**: 279–284. 1967.
- [6] Felsenstein, J: *Inferring Phylogenies*. Sinauer, Washington, 2004.
- [7] Thomas H. Cormen and Charles E. Leiserson and Ronald L. Rivest and Clifford Stein: *Introduction to Algorithms*. The MIT Press, Cambridge, MA, 2001.
- [8] Burns, Greg and Daoud, Raja and Vaigl James. Lam: an open cluster environment for MPI. In: *SUPERCOMPUTING SYMPOSIUM'94*. 42–386. Toronto: University of Toronto 1994.
- [9] Felsenstein, J: **PHYLIP (phylogeny inference package)**, version 3.6. Washington, 2005. <http://evolution.genetics.washington.edu/phylip/getme.html>.