PAMM · Proc. Appl. Math. Mech. **7**, 2120007–2120008 (2007) / **DOI** 10.1002/pamm.200700391

Numerical methods for inferring phylogenies

Felipe Albrecht * and **Nelson Borges** **

Department of Computer and Systems Engineering, Military Institute of Engineering (IME), Rio de Janeiro, Brazil.

Molecular phylogenies describe the study of evolution relationships between living beings, protein and genetic sequences and other molecular taxons. In recent years there has been increased interest in producing large and accurate phylogenies trees using some numerical computational methods approach. One class of molecular phylogenies is the distance matrices method, with the matrix whose entries represent the distinctions between the taxons under study. To construct large and accurate trees for a large number of taxons, the computational processing increases and parallel computing can be helpful. So far we have tested some methods to solve the linear systems, being driven to important conclusions, like which is the best tested method, and having also reached some questions about communication.

1 Introduction

Molecular phylogenies describe the study of evolution relationships between living beings, protein and genetic sequences and other molecular taxons. Inferring phylogenies from molecular data may be carried out through the point analysis of similarities and differences in the studied sequences. One class of molecular phylogenies is the distance matrices method, with a matrix whose entries represent the distinctions between the taxons under study. According to Felsenstein [1], the distance matrices methods may be shortly described as: compute a distance estimate for each pair of species and then search a tree that better approaches this distance set. This procedure discards all data originated from any combination bearing a high amount of characters and thus reduce the data matrix to a matrix associated to pairs of distances.

In recent years there has been increased interest in producing large and accurate phylogenies trees using some numerical computational methods approaches. To construct large and accurate trees for a large number of taxons, the computational processing increases and parallel computing can be helpful [2, 3]. We work with the distance matrix methods for inferring phylogenies by taking a large number of taxons. These methods have been introduced by Cavalli-Sforza and Edwards [4] and by Fitch and Margoliash [5]. The methods introduced by Cavalli-Sforza use a set of linear equations where the number of equations and their variables grow up with the number of taxons, linearly. Here, we use parallel computing to perform the time processing to solve the numerical block linear systems associated to the length of the branches of the phylogenetic trees.

2 Motivation

The least squares method is not suitable to get from the distance matrices the sides length for each branch of the considered trees. For a given tree topology, the solution of the corresponding system of linear algebraic equations, see [4], becomes intractable with direct methods if we are interested in a high number of taxons.

The matrix associated to the least-squares method, being symmetric and positive-definite, allows the use of the conjugated gradient method (CGM). We know that in the worst case convergence is guaranteed in n steps, being n the linear system order. Since each iteration for the CGM requires $n^2 + O(n)$ multiplications and divisions, we deduce that in this case we need $n^3 + O(n^2)$ operations, which is the same amount needed for a complete inversion of the matrix with the LL^t decomposition.

In order to improve convergence for CGM, we get hold of the matrix profile. It turns out that all matrices that appear are diagonal block-matrices, being all blocks also symmetric and positive-definite. We can then use the Preconditioned Conjugate Gradient method with block Jacobi as a preconditioner – this amounts to incomplete block decomposition. It is known that Jacobi method has a rather slow convergence but, when employed as a preconditioner, overall convergence is speeded up. Besides, parallelization of block-Jacobi preconditioned conjugate gradient method (BPCGM) may be obtained very efficiently.

3 The method

Given T, a tree without root and having a set of taxons L, our aim is to find a set of distances between these taxons in such a way that these distances turn out to be the best approximations – in the least-squares sense – to the observed distances.

The measure given by the least-squares method has the expression: $Q = \sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} w_{ij}(D_{ij} - d_{ij})^2$. In this equation, d_{ij} denotes the sum of the length of the branches connecting taxons i to j, D_{ij} means the distances given by the distance matrix while w_{ij} stands for the weights, here taken equal to unity. Also Q means the sum of the squares of the differences between

* E-mail: felipe.albrecht@gmail.com
** E-mail: nborges@ime.eb.br

D_{ij}, the distances given by the tree entries, and d_{ij}, the inferred distances. The least-squares method aims to minimize the value for Q. This minimum is reached first taking the derivatives of Q with respect to the branches length, then making the resulting equation all equal to zero.

The first step consists in generating the matrix X to represent the position the taxons occupy in the tree in a matrix representation. Here each column represents a side in the tree, while each row represents a pair of taxons. If the path between two taxons i and j passes through branch k, we must put label 1 on line ij and column k, otherwise we label this position with 0. This coefficient is denoted by $x_{ij,k}$. The coefficients $d_{i,j}$ denote the sums of the branches length. And this equation may be written in matrix form $(X^tX)v = X^t.d$, where X is the matrix which represents the position of all taxons on the tree and d is the distance vector between the taxons.

Hence, the linear system is written in the form $Av = b$ where A is a symmetric positive definite nxn matrix, $b = X^td$ is a vector associated to the independent term with dimension n and v is the unknown vector with dimension n. To solve this linear system we could invert the matrix A and then use $A^{-1} = (LL^t)^{-1} = (L^t)^{-1}L^{-1}$, where L is Cholesky decomposition lower triangular matrix (nxn). This process becomes costly when n gets large, reaching a unsuitable threshold.

Being the matrix A positive definite, we can apply the conjugate gradient method in its parallel version. Besides, the matrix A exhibits a block profile that assures that the block-Jacobi iterative method converges when applied to solve the linear system. As claimed before, our aim is to use iterative block-Jacobi as a preconditioner for the preconditioned conjugate gradient method (BPCGM).

Given the nxn matrix A, shown in our linear system $Av = b$, it must be partitioned in block matrices, with diagonal blocks B_k, where diagonal $(B_k) = d_k$, $d_k = a_{ii}$, for some $i = 1, 2, 3, ...n$. This way we get m (n_kxn_k) blocks B_k, $\sum_{k=0}^{m-1} n_k = n$.

First, each block B_k, $k = 0, 1, ..., m-1$, which turns out to be symmetric and positive definite, is inverted with Cholesky decomposition. The matrix A is preconditioned by matrix C (nxn) given by block matrices B_k, $k = 0, 1, .., m-1$. In this case, we work the linear system $C^{-1}Av^* = C^{-1}b$ for the conjugate gradient method. The processing for BPCGM is then shared between the m blocks B_k, $k = 0, 1, ..., m-1$ distributed among the p processors with $rank(p_j) = j, j = 0, 1, ..., p-1$, where p denotes the number of processes. We take $r_k = (m_k \mod p)$ and the process of $rank(p_j) \equiv r_k$ carries out the processing which corresponds to the m_k block sub-vector. We have a process which assigns the tasks globally (master), and the slave processes that execute the parts of BPCGM, always restricted to the values assigned to their task share.

The additional cost for the BPCGM is the construction of the sub-vector sequences $g_k = B_k^{-1}r_k$, where $r_k = A_kv_k - b_k$, A_k (n_kxn_k) are the blocks in the matrix A and as v_k and b_k the corresponding sub-vectors, for $k = 0, 1, .., m-1$.

4 First Results

We first used the sequential versions of block-Jacobi method (JBM), conjugate gradient and preconditioned conjugate gradient with block-Jacobi as preconditioner. It turns out that convergence is much better for the latter, although its speed-up with respect to the second one is rather small. It seems encouraging to use a parallel version of BPCGM. Nevertheless we must take into account its rather high data transmission rate as well as the additional cost to invert the block matrices B_k required for the algorithm start-up. Anyway, we can claim that as long as inversion of the matrix A is intractable, the parallel version of BPCGM is available for the rescue.

References

[1] J. Felsenstein, Inferring phylogenies (Sinauer Associates, Massachusetts, 2003).
[2] T. Keane, T. Naughton, S. Travers, J. McInerney, and G. McCormak, Bioinformatics **21(7)**, 969–974 (2005).
[3] A. Stamatakis, Distributed and parallel algorithms and systems
for inference of huge phylogenetic trees bases on the maximum likelihood method, Ph.d. on computer science, Technischen Universität München, München, 2004.
[4] L. Cavalli-Sforza and A. Edwards, Am. J. Human Genetics **19**(3), 233–257 (1967).
[5] W. M. Fitch and E. Margoliash, Science **760**(157), 279 (1967).