Available online at www.sciencedirect.com
SCIENCE DIRECT®

Computational Biology and Chemistry

Computational Biology and Chemistry 29 (2005) 196–203

www.elsevier.com/locate/compbiolchem

Integer linear programming as a tool for constructing trees from quartet data☆

Jan Weyer-Menkhoff[a,*], Claudine Devauchelle[b], Alex Grossmann[b], Stefan Grünewald[c]

[a] *Universität Göttingen, Biologische Fakultät, Institut für Mikrobiologie und Genetik, Abt. Bioinformatik Goldschmidtstr. 1, D-37073 Göttingen, Germany*
[b] *Laboratoire Génome et Informatique (LGI), Genopole-Evry, 523 place des Terrasses, 91000 Evry, France*
[c] *Allan Wilson Centre for Molecular Ecology and Evolution, Department of Mathematics and Statistics, University of Canterbury, Private Bag 4800, Christchurch, New Zealand*

Received 14 January 2005; accepted 6 April 2005

Abstract

The task of the quartet puzzling problem is to find a best-fitting binary X-tree for a finite n-set from confidence values for the $3\binom{n}{4}$ binary trees with exactly four leaves from X, its *fitness* being measured by the sum of the confidence values of all “induced” four-leaves subtrees. We describe a method for finding an exact solution of this problem by integer linear programming. Similar procedures can also be used for finding, e.g. best-fitting “circular” networks.

A crucial problem in this context is, of course, how to obtain the input confidence values for the quartet trees. We propose to use inner products of rate-matrix diagonals calculated for pairs of taxa and present the trees resulting from applying our approach to two data sets of up to 36 mitochondrial sequences of mammals including an outgroup.

Keywords: Weighted quartet; Integer linear programming; Observed rate matrix; Mammals’ mitochondrial evolution; Phylogeny

1. Introduction

Most methods to reconstruct phylogenetic trees, networks, or other structures, use either a distance matrix (e.g. neighbour joining) or a full sequence alignment (e.g. maximum likelihood and maximum parsimony) as their input (Saitou and Nei, 1987; Felsenstein, 1981; Fitch, 1971; Farris, 1970). While reducing the data to pairwise distances might cause the loss of some signals that can only be obtained by considering individual residues, working with the full alignment often makes it necessary to solve optimization problems which are not feasible for many taxa. A possible compromise between these approaches is to create residue-based trees for small subsets of the set of taxa of interest and then to combine the results to find a big tree. Since four taxa are needed to obtain different possible tree topologies, it is natural to consider all subsets with four elements (quadruples) of the set of taxa.

For any four taxa a, b, c, d from a finite set X of investigated taxa, there exist exactly three binary trees with leaf set $\{a, b, c, d\}$ which will be called *quartet trees* and which will be symbolised by $ab|cd$, $ac|bd$, $ad|bc$. The most straightforward idea for a quartet method is to use some tool to calculate the best fitting quartet tree for every quadruple of X and then to construct an X-tree, i.e. an unrooted binary tree with leaves labelled by X, that contains all optimal quartet trees as its restriction to the corresponding quadruple. Unfortunately, such a tree does not exist in general. Moreover, it turns out that, for real data, quartet methods that do not allow non-optimal quartet trees tend to produce trees with very few internal edges.

☆ This paper is based on ideas that were developed and worked out jointly with Andreas Dress (Max Planck Institute for Mathematics in the Sciences, Leipzig, Germany).

* Corresponding author. Tel.: +49 551 3914627; fax: +49 551 3914929.
E-mail addresses: jan@gobics.de (J. Weyer-Menkhoff), devauchelle@genopole.cnrs.fr (C. Devauchelle), grossman@genopole.cnrs.fr (A. Grossmann), s.grunewald@math.canterbury.ac.nz (S. Grünewald).

1476-9271/$ – see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compbiolchem.2005.04.001

Assuming that we accept that a good X-tree contains some non-optimal quartet trees, it is sensible to introduce a measure of quality. This way, we can measure how much worse a non-optimal quartet tree is compared to the respective optimal one. More precisely, we start with a function that maps every possible quartet tree q to a confidence value $w(q)$ which represents how much one thinks that q represents the true family relationship. Of course, we would prefer to accept a non-optimal quartet tree which has an almost equal confidence value as the optimal one, rather than to accept a non-optimal quartet tree with a confidence value significantly different to the one of the optimal quartet tree.

More formally, we are interested in solving the *quartet puzzling problem*: given a confidence value for every possible quartet tree on X, find a binary X-tree T such that the sum $w(T)$ of the confidence values of all quartet trees which are restrictions of T is maximal.

It has been shown in (Steel, 1992) that, for a given collection $\mathcal{Q}$ of quartet trees, it is NP-hard to decide if an X-tree exists which contains all quartet trees in $\mathcal{Q}$ as restrictions. Hence, we cannot expect that there is a polynomial algorithm to find an optimal tree. Some heuristics have been developed to construct a tree T for which $w(T)$ is large but not necessarily optimal. The most widely used method of this kind is Tree Puzzle (Strimmer and von Haeseler, 1996; Strimmer et al., 1997) which produces many binary trees and then applies a consensus method to obtain the not necessarily binary Tree-Puzzle tree. Other approaches are the "Geometric Algorithm" in (Ben-Dor et al., 1998) and a weighted version of AddQuart (Berry and Gascuel, 2000).

An exact method to solve the quartet puzzling problem is also presented in (Ben-Dor et al., 1998). That approach uses dynamic programming and manages to solve problems with up to 20 taxa.

Also an important approach for solving quartet problems is split decomposition (Bandelt and Dress, 1993, 1992; Dress et al., 1996b) with visualisation by the program Splitstree (jsplits) (Dress et al., 1996a; Huson, 1998; Huson and Bryant, 2005).

In this paper, we reformulate the problem as an integer linear programming (ILP) problem. The number of variables and constraints increases very rapidly with the number of taxa. The standard ILP tools became insufficient for families containing more than about 17 taxa. However, a collaboration with the Operational Research and Optimization Group of the Department of Mathematics and Statistics at the University of Edinburgh, especially with Ken McKinnon, and the Edinburgh Parallel Computing Centre has led to the development of an algorithm that made it possible to solve the problem for up to 36 taxa. Instead of considering all constraints at once, the algorithm adds only a small, randomly-chosen fraction of violated constraints to the solver. For more details, see (Weyer-Menkhoff, 2003).

Moreover, by slightly changing the constraints, we can solve the corresponding problem for other phylogenetic structures like cyclic split systems (Bandelt and Dress, 1992).

The quartet methods described above require confidence values for the possible quartets, and they are independent of the way of obtaining those confidence values. For example, Tree-Puzzle uses posterior likelihoods, and an other option would be parsimony scores.

In this paper, we also introduce a new method for calculating confidence values: we use the negative scalar product of diagonals of *observed rate matrices*. In (Devauchelle et al., 2001), an observed rate matrix is associated to each pair of taxa in a multiple alignment. It is defined as the matrix-valued logarithm of the corresponding observed Markov matrix. The idea of analysing observed rate matrices is that for each two taxa a matrix is calculated which consists of 20×20 entries. Each of these entries (especially the diagonal elements) expresses a genetic difference between the two taxa. As each noise event might effect some but not all of these "clocks", the weight of such errors is reduced in the scalar product.

We have used the data of twelve genes of the mitochondria of 20 and of 36 taxa (mammals and outgroup). We derived binary X-trees which are close to a previously published tree. Without asserting that we have found the correct tree of mammals, we conclude that the method of deriving confidence values via rate matrices as well as the method of solving the quartet puzzle problem with integer linear programming give promising results and that they should, independently and combined, be developed further to obtain a tool of phylogenetic analysis.

2. Preliminaries on X-trees and their generalisations, split systems and quartet systems

In this section, we will recall and introduce definitions of three equivalent concepts to express the same phylogenetic information: as binary X-tree, as a compatible collection of X-splits, or as a Colonius–Schulze quartet system (introduced later). The fact that in a lot of cases not all aspects of the evolution of investigated taxa can be expressed by binary X-trees (as well as by collections of compatible X-splits or by Colonius–Schulze quartet systems) has led to a big effort to generalise the three concepts by weakening conditions while keeping the equivalence (or at least an injective relation) between at least two of the three concepts.

We will also recall some of these generalisations.

Most important in this section for understanding the other sections will be Theorem 1, which explains the relation between binary X trees and sets of quartet trees satisfying certain conditions. Our approach enables to find an element which explains best given quartet confidence data and which is taken out of a class which can be more general than the class of binary X trees. In order to understand also these generalisations, the whole Section 2 should be read.

A *partial split* S of a finite set X is an unordered pair $\{A, B\}$ of two disjoint and non-empty subsets A, B of X. It is also