Integer Linear Programming as a Tool for Constructing Trees from Quartet Data ⋆

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

[a] *Universität Göttingen, Institut für Mikrobiologie und Genetik, Abt. Bioinformatik*
Goldschmidtstr. 1, D-37073 Göttingen, Germany
jan@gobics.de

[b] *LGI (Laboratoire Génome et Informatique) - Genopole-Evry*
523 place des Terrasses, 91000 Evry, France
devauchelle@genopole.cnrs.fr, grossman@genopole.cnrs.fr

[c] *Allan Wilson Centre for Molecular Ecology and Evolution*
Department of Mathematics and Statistics
University of Canterbury
Private Bag 4800, Christchurch, New Zealand
s.grunewald@math.canterbury.ac.nz

[d] *corresponding author*

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.

Key words: weighted quartet, integer linear programming, observed rate matrix, Mammals' mitochondrial evolution, phylogeny

⋆ 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,

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.

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,

Germany).

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, cf. (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 called a *split* if $A \cup B = X$ holds. A partial split $\{A, B\}$ is called *trivial* if $\#A = 1$ or $\#B = 1$ holds.

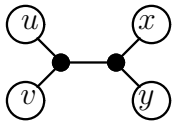

Fig. 1. The quartet tree $uv \mid xy$

Two splits S and S' are said to be *compatible* if there exist $A \in S$ and $A' \in S'$ with $A \cap A' = \emptyset$.

If we eliminate an edge from an X-tree T, the X-tree decomposes into two connected components. Thus, every edge e of an X-tree induces a unique split $S = S_e$ of X consisting of the set of elements of X which are leaves of one connected component and the set of elements of X which are leaves of the other connected component. Let

$$\mathfrak{S}(T) := \{S_e : e \text{ is an edge of } T\}$$

denote the collection of splits of X *displayed* by T (or the *split encoding* of T), i.e. the collection of all splits that are associated with T in this way.

As is easily seen and was noted already for example in (Buneman, 1971), any two splits in $\mathfrak{S}(T)$ are compatible. Conversely, given any set $\mathcal{S}$ of pairwise compatible splits of X containing all trivial splits on X, there exists exactly one X-tree T (up to canonical isomorphism) with $\mathfrak{S}(T) = \mathcal{S}$ which is therefore also called the *Buneman tree* associated with $\mathcal{S}$ and denoted by $\mathfrak{B}(\mathcal{S})$. Thus, $\mathfrak{S}(\mathfrak{B}(\mathcal{S})) = \mathcal{S}$ holds for every set $\mathcal{S}$ of pairwise compatible splits containing all trivial splits of X while $\mathfrak{B}(\mathfrak{S}(T))$ "=" T holds for every X-tree T. If and only if $\mathcal{S}$ is inclusion-maximal in the set of pairwise compatible collections of splits, $\mathfrak{B}(\mathcal{S})$ is binary.

How the Buneman tree can be constructed was already explained in 1971 in (Buneman, 1971).

A *circular split system* is a split system generated as follows:

Let $x_0, \ldots, x_n = x_0$ (in this order) be the vertices of a convex n-gon. Any pair of distinct edges $(x_i, x_{i+1}), (x_j, x_{j+1})$, where $i < j$, gives rise to a split $\{x_{i+1}, x_{i+2}, \ldots x_j\}, \{x_{j+1}, \ldots x_{i-1}, x_i\}$, that is, the split induced by any line crossing the edges (x_i, x_{i+1}) and (x_j, x_{j+1}).

The symbol $uv \mid xy$ denotes the quartet tree in Figure 1. Note that $uv \mid xy = vu \mid xy = xy \mid uv$ always holds. Given X, any collection of quartet trees on X will be called *quartet system on* X. The collection of all quartet trees on X will be denoted by $\mathfrak{S}_{2,2}(X)$.

Given a quartet system $\mathcal{Q}$ on X, we define a (partial) split $S = \{A, B\}$ of X

to be a (partial) $\mathcal{Q}$-*split* if $aa' \mid bb' \in \mathcal{Q}$ holds for all distinct $a, a' \in A$ and $b, b' \in B$.

We denote the set of all $\mathcal{Q}$-splits by $\mathfrak{S}(\mathcal{Q})$ and the collection of all partial $\mathcal{Q}$-splits by $\mathfrak{S}_{part}(\mathcal{Q})$. Conversely, given any collection $\mathcal{S}$ of X-splits, the quartet system $\mathfrak{Q}(\mathcal{S})$ of *quartet trees displayed by* $\mathcal{S}$ is defined by

$$\mathfrak{Q}(\mathcal{S}) := \Big\{ aa'|bb' \Big| \#\{a, a', b, b'\} = 4, \text{ and there exists a split } \{A, B\} \in \mathcal{S} \text{ with } a, a' \in A \text{ and } b, b' \in B \Big\}. \tag{1}$$

Note that $\mathfrak{Q}(\mathfrak{S}(\mathcal{Q})) \subseteq \mathfrak{Q}(\mathfrak{S}_{part}(\mathcal{Q})) = \mathcal{Q}$ holds for every quartet system $\mathcal{Q}$.

For an X-tree T, the quartet system $\mathfrak{Q}(\mathfrak{S}(T))$ is the collection of quartet trees which are restrictions of T.

Already in (Colonius and Schulze, 1977), conditions have been shown which enable to decide whether for a given quartet system $\mathcal{Q}$ a binary X-tree T exists for which $\mathcal{Q} = \mathfrak{Q}(\mathfrak{S}(T))$ holds: They have established a theorem that is essentially equivalent to:

Theorem 1 (Colonius and Schulze) *Given a finite set X of cardinality $n \geq 4$ and a quartet system $\mathcal{Q}$ on X, i.e. a subset $\mathcal{Q}$ of the set*

$$\mathfrak{S}_{2,2}(X) := \{ab|cd : a, b, c, d \in X, \#\{a, b, c, d\} = 4\}$$

of all $(2,2)$-splits of X, there exists a (necessarily unique) binary X-tree T with $\mathcal{Q} = \mathfrak{Q}(\mathfrak{S}(T))$ if and only if

$$\#(\mathcal{Q} \cap \{ab|cd, ac|bd, ad|bc\}) = 1 \tag{2}$$

holds for all a, b, c, d in X with $\#\{a, b, c, d\} = 4$, and

$$ab|cd, ab|de \in \mathcal{Q} \Rightarrow ab|ce \in \mathcal{Q} \tag{3}$$

as well as

$$ab|cd, bc|de \in \mathcal{Q} \Rightarrow ab|de \in \mathcal{Q} \tag{4}$$

holds for all a, b, c, d, e in X with $\#\{a, b, c, d, e\} = 5$.

A quartet system $\mathcal{Q}$ will be called a *simple cover* if

$$\#(\mathcal{Q} \cap \{ab|cd, ac|bd, ad|bc\}) = 1,$$

and a *double cover* if

$$\#\left(\mathcal{Q} \cap \{ab|cd, ac|bd, ad|bc\}\right) = 2$$

holds for all a, b, c, d with $\#\{a, b, c, d\} = 4$.

Further, $\mathcal{Q}$ will be called *telescopic* if Condition (4) from Theorem 1 holds for any five distinct elements a, b, c, d, e from X, and it will be called *transitive* if Condition (3) from Theorem 1 holds for any five distinct elements a, b, c, d, e as above.

A transitive telescopic simple cover will be called a Colonius-Schulze quartet system. So, Colonius-Schulze quartet systems on X correspond bijectively to binary X-trees and so to maximal compatible split systems on X.

Supplemented by (Weyer-Menkhoff, 2003), it is shown in (Bandelt and Dress, 1992) that $\mathcal{Q}$ is a transitive double cover if and only if $\mathfrak{S}(\mathcal{Q})$ is a circular split system.

3 Quartet Puzzle Problem and Integer Linear Programming

In this section, we reformulate the Quartet Puzzle Problem as integer linear programming problem.

We state the Quartet Puzzle Problem in a more general way: Given a collection $\mathbf{R}$ of quartet systems, find the element $\mathcal{Q} \in \mathbf{R}$ which maximises the sum $\sum_{q \in \mathcal{Q}} w(q)$ of the corresponding confidence values. If $\mathbf{R}$ is the collection of Colonius-Schulze quartet systems, this problem is equivalent to the problem to find the best-fitting binary X-tree.

An integer linear programming problem can be stated as follows:

Given linear functions $A : \mathbb{R}^m \to \mathbb{R}^n$ and $c : \mathbb{R}^m \to \mathbb{R}$ and a vector $b \in \mathbb{R}^n$, find $x \in \mathbb{Z}^m$ which satisfies:
(5)
The vector x maximises $c(x)$ in the set of x for which $A(x) \leq b$ holds.

(The condition $x \leq y$ denotes that $x_i \leq y_i$ holds for all positions i in the vectors. $\mathbf{0}$ denotes the zero vector, and $\mathbf{1}$ denotes the vector with 1 as every entry.)

The corresponding relaxed linear programming problem is to find $x \in \mathbb{R}^m$ which satisfies Condition (5).

For an introduction to linear programming problems, cf. (Saigal, 1995), for an introduction to integer linear programming problems, cf. (Schrijver, 1986).

We will show that if the collection $\mathbf{R}$ of allowed quartet systems consists of the Colonius-Schulze quartet systems on X, one can transform the problem into the following integer linear optimisation problem:

Problem 1 (Integer Linear Programming Problem) *Find* $x \in \mathbb{Z}^{\mathfrak{S}_{2,2}(X)}$ *which maximises* $\sum_{q \in \mathfrak{S}_{2,2}(X)} w(q) \cdot x_q$ *under the constraints that*

i)	$\mathbf{0} \leq x \leq \mathbf{1}$	
ii)	$x_{ab\|cd} + x_{ac\|bd} + x_{ad\|bc} = 1$	*for any distinct* $a, b, c, d \in X$
iii)	$x_{ab\|cd} + x_{ab\|de} - x_{ab\|ce} \leq 1$	*for any distinct* $a, b, c, d, e \in X$
iv)	$x_{ab\|cd} + x_{bc\|de} - x_{ab\|de} \leq 1$	*for any distinct* $a, b, c, d, e \in X$

hold.

Clearly, the problem is an integer linear programming problem in the form stated above: The objective function is linear, the "=" constraint can be transformed to a "$\leq$" and a "$\geq$" constraint and "$\geq$" constraints can be transformed to "$\leq$" constraints by multiplying with -1.

Any integral vector x satisfying Condition (i) has only components with 0 or 1 as value, and it corresponds to the quartet system $\mathcal{Q}_x$ which contains those quartet trees q of $\mathfrak{S}_{2,2}(X)$ for which x_q equals 1. So, this defines a 1-1 correspondence between the set of integral vectors satisfying Condition (i) and the set $2^{\mathfrak{S}_{2,2}(X)}$ of quartet systems on X.

An integral vector satisfying Condition (i) also satisfies Condition (ii) if and only if the corresponding quartet system $\mathcal{Q}_x$ is a simple cover. It satisfies Condition (iii) if and only if the corresponding quartet system $\mathcal{Q}_x$ is transitive: $x_{ab|cd} + x_{ab|de} = 2$ holds if and only if $ab \mid cd \in \mathcal{Q}_x$ and $ab \mid de \in \mathcal{Q}_x$ hold. In this case, $x_{ab|cd} + x_{ab|de} - x_{ab|ce} \leq 1$ holds if and only if $ab \mid ce \in \mathcal{Q}_x$ holds.

Similarly, an integral vector x satisfying Condition (i) also satisfies Condition (iv) if and only if the corresponding quartet system $\mathcal{Q}_x$ is telescopic.

It follows that x is a feasible solution of the integer linear programming problem above, if and only if the corresponding quartet system $\mathcal{Q}_x$ is a simple transitive telescopic cover, in other words, if and only if $\mathcal{Q}_x$ is Colonius Schulze.

As $x_q = 1$ holds for any $q \in \mathcal{Q}_x$ and as $x_q = 0$ holds for any $q \notin \mathcal{Q}_x$, $\sum_{q \in \mathfrak{S}_{2,2}(X)} w(q) \cdot x_q = \sum_{q \in \mathcal{Q}_x} w(q)$ holds, and x is an optimal solution of the integer linear programming problem if and only if $\mathcal{Q}_x$ is a solution of the quartet puzzle problem.

In the same way, one can see that $\mathcal{Q}_x$ is an optimal solution of the quartet puzzle problem for the collection $\mathbf{R}$ of allowed quartet systems being the collection of transitive double covers if and only if x solves the following integer linear programming problem:

Problem 2 (Integer Linear Programming Problem) *Find* $x \in \mathbb{Z}^{\mathfrak{S}_{2,2}(X)}$ *which maximises* $w(x)$ *under the constraints that*

i) $\mathbf{0} \leq x \leq \mathbf{1}$
ii) $x_{ab|cd} + x_{ac|bd} + x_{ad|bc} = 2$ *for any distinct* $a, b, c, d \in X$
iii) $x_{ab|cd} + x_{ab|de} - x_{ab|ce} \leq 1$ *for any distinct* $a, b, c, d, e \in X$

hold.

The following table shows the number of variables and the number of constraints which have to hold in addition to the constraints $x \geq 0$.

taxa	variables	constraints Prob. 1	constraints Prob. 2
n	$3 \cdot \binom{n}{4}$	$\binom{n}{4} + 30 \cdot \binom{n}{5}$	$\binom{n}{4} + 60 \cdot \binom{n}{5}$
5	15	35	65
6	45	195	375
7	105	665	1295
8	210	1750	3430
9	378	3906	7686
10	630	7770	15330
20	14535	469965	935085
30	82215	4302585	8577765
36	176715	11368665	22678425
40	274170	19831630	39571870

As described in (Weyer-Menkhoff, 2003), a way was found for solving such large problems for a number of taxa up to 36.

As we assume that NP-hard problems cannot be solved in polynomial time, and as Steel (1992) has shown that the quartet puzzle problem is NP-hard, we cannot expect that one can find a formulation of the linear programming problem with a polynomial number of constraints such that the solution of the relaxed linear programming problem is always integral.

For the real-data example given in Section 5, the solution of the relaxed linear programming problem was integral. As one can expect that for a good biologically meaningful confidence function, the quartet system which contains for any three corresponding quartet trees the one with the highest confidence differs only in a few quartet trees from the solution of the quartet puzzle problem, we hope that in practice a good confidence function yields an integral solution for the relaxed integer linear programming problem.

4 Observed Rate Matrices and Quartet Weights

It is important to find accurate confidence values for the quartet trees. For sufficiently long amino acid sequences, we suggest to use *observed rate matrices* which are introduced and investigated by Devauchelle et al. (2001).

The idea of observed rate matrices is the following: In a way that will be explained later, for any two taxa x, y under consideration, an observed rate matrix $L^{(x,y)}$ can be calculated. In (Devauchelle et al., 2001) the authors assert: "If one neglects fluctuations arising from the finite length of sequences, any continuous reversible Markov model with a single rate matrix Q over an arbitrary tree predicts that all the observed matrices L are multiples of Q." In other words, the observed rate matrices viewed as vectors in the 400 dimensional space would all point into the same direction.

In fact, as investigated in the same article, the vectors pointed only roughly into the same direction and one could see that directions belonging to pairs of the same phylogenetic group formed clouds.

For calculating a confidence value $w\,(ab \mid cd)$, we will use the negative scalar product of the diagonals of the observed rate matrices $L^{(a,b)}$ and $L^{(c,d)}$. As will be explained later, these values take advantage of both aspects of the observed rate Matrices: The length and the direction of the observed rate matrices.

It is not a disadvantage that these confidence values are usually negative if the quartet puzzling problem satisfies that an additive constant added to each of the confidence values will not change the solution. (The two problems considered in Section 3 satisfy this condition.) If you prefer to work with non-negative weights, just add a sufficiently large additive constant to the values.

The article (Devauchelle et al., 2001), starts with a natural Markov matrix $P^{(a,b)}$ associated to an aligned ordered pair (a, b) of sequences.

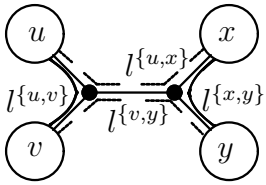

Fig. 2. If $uv \mid xy$ is the correct quartet tree, the lengths of $l^{\{u,v\}}$ and $l^{\{x,y\}}$ tend to be shorter than the lengths of $l^{\{u,x\}}$, $l^{\{u,y\}}$, $l^{\{v,x\}}$, $l^{\{v,y\}}$.

The *observed rate matrix* (if it exists) is defined as

$$L^{(x,y)} = \log P^{(x,y)} \tag{6}$$

where $\log P$ denotes the matrix-valued logarithm of a matrix P. If P is "close enough" to the identity matrix, this matrix logarithm exists.

For the matrix logarithm, the relation $\log P^\tau = \tau \log P$ holds for real numbers τ.

We symmetrise:

$$\overline{L}^{\{x,y\}} := \frac{1}{2}\left(L^{(x,y)} + L^{(y,x)}\right) \tag{7}$$

and let $l^{\{x,y\}}$ be the diagonal of $\overline{L}^{\{x,y\}}$.

As mentioned before, we suggest to use the negative scalar product $-\left\langle l^{\{u,v\}}, l^{\{x,y\}}\right\rangle$. as confidence value $w\left(uv \mid xy\right)$ for a quartet tree $uv \mid xy$.

There are two reasons:

First, if, for two matrices $P^{(x,y)}$ and $P^{(u,v)}$ the condition $\left(P^{(x,y)}\right)^\tau = P^{(u,v)}$ holds, $\tau \log P^{(x,y)} = \log P^{(u,v)}$ follows. So, we would expect that if the "time distance" between u and v is larger than the "time distance" between x and y that then the absolute value of each component of the vector $l^{\{u,v\}}$ is larger than the absolute value of the corresponding component in the vector $l^{\{x,y\}}$. So, consider the case that the quartet tree $uv \mid xy$ is the correct one as shown in Figure 2. If the lengths of the branches do not differ too much, one will in most cases observe that $l^{\{u,v\}}$ as well as $l^{\{x,y\}}$ is shorter than $l^{\{u,x\}}$, $l^{\{u,y\}}$, $l^{\{v,x\}}$, $l^{\{v,y\}}$. It follows that in most cases $w\left(uv \mid xy\right) = -\left\langle l^{\{u,v\}}, l^{\{x,y\}}\right\rangle$ is larger than $w\left(ux \mid vy\right) = -\left\langle l^{\{u,x\}}, l^{\{v,y\}}\right\rangle$ and $w\left(uy \mid vx\right) = -\left\langle l^{\{u,y\}}, l^{\{v,x\}}\right\rangle$.

The second reason for using these quartet weights is the following.

Assume that $uv \mid xy$ is the correct quartet tree. The "history" from u to v (or from v to u) is independent from the "history" from x to y while the "history" from u to x shares the middle edge with the "history" from v to y (see Figure 2). It follows that the rate matrices $L^{(u,x)}$ and $L^{(v,y)}$ – regarded as vectors in an 400 dimensional vector space – tend to point into the same direction while the rate matrices $L^{(u,v)}$ and $L^{(x,y)}$ tend to point into different directions.

This is the second reason why one would expect that $-\left\langle l^{\{u,v\}}, l^{\{x,y\}} \right\rangle$ is larger than $-\left\langle l^{\{u,x\}}, l^{\{v,y\}} \right\rangle$.

5 Validation with Biological Data

For checking that our methods produce reasonable results, we applied them to a set of 17 Mammals plus 3 taxa as outgroup. For the 20 taxa, we analysed 12 mtDNA encoded proteins (all except ND6). Manually produced alignments have been obtained from Trish McLenachan, Allan Wilson Centre, Massey University (personal communication). These alignments are close to those which were used by Penny et al. (1999) for the investigation of the mammalian evolution.

For each pair $\{a, b\}$ of elements of X, the diagonal $l^{\{a,b\}}$ of the observed rate matrix was calculated. For the confidence function $w : ab \mid cd \mapsto -\left\langle l^{\{a,b\}}, l^{\{c,d\}} \right\rangle$, the Quartet Puzzle Problem to find the best fitting Colonius-Schulze quartet system was solved by using the integer linear programming approach explained above. In order to calculate the Buneman tree corresponding to a quartet system, the package "phyloquart" by Berry (1999) was used. For comparison, we have also applied the phylogeny program "tree-puzzle" to the same alignments. (Cf. (Schmidt et al., 2003-2004, 2002).) The tree obtained by tree-puzzle has the same topology as the one published by Penny et al. (1999). The trees have been visualised with the "phylip" package by Felsenstein (1993) and they are shown in Figure 3.

The aim of this paper is not a careful investigation and discussion of the evolution of mammals. What we want to emphasise is, that the trees shown in Figure 3. are close enough to conclude that our methods produce reasonable results.

In order to show that the method can also be applied to a larger set of taxa, we

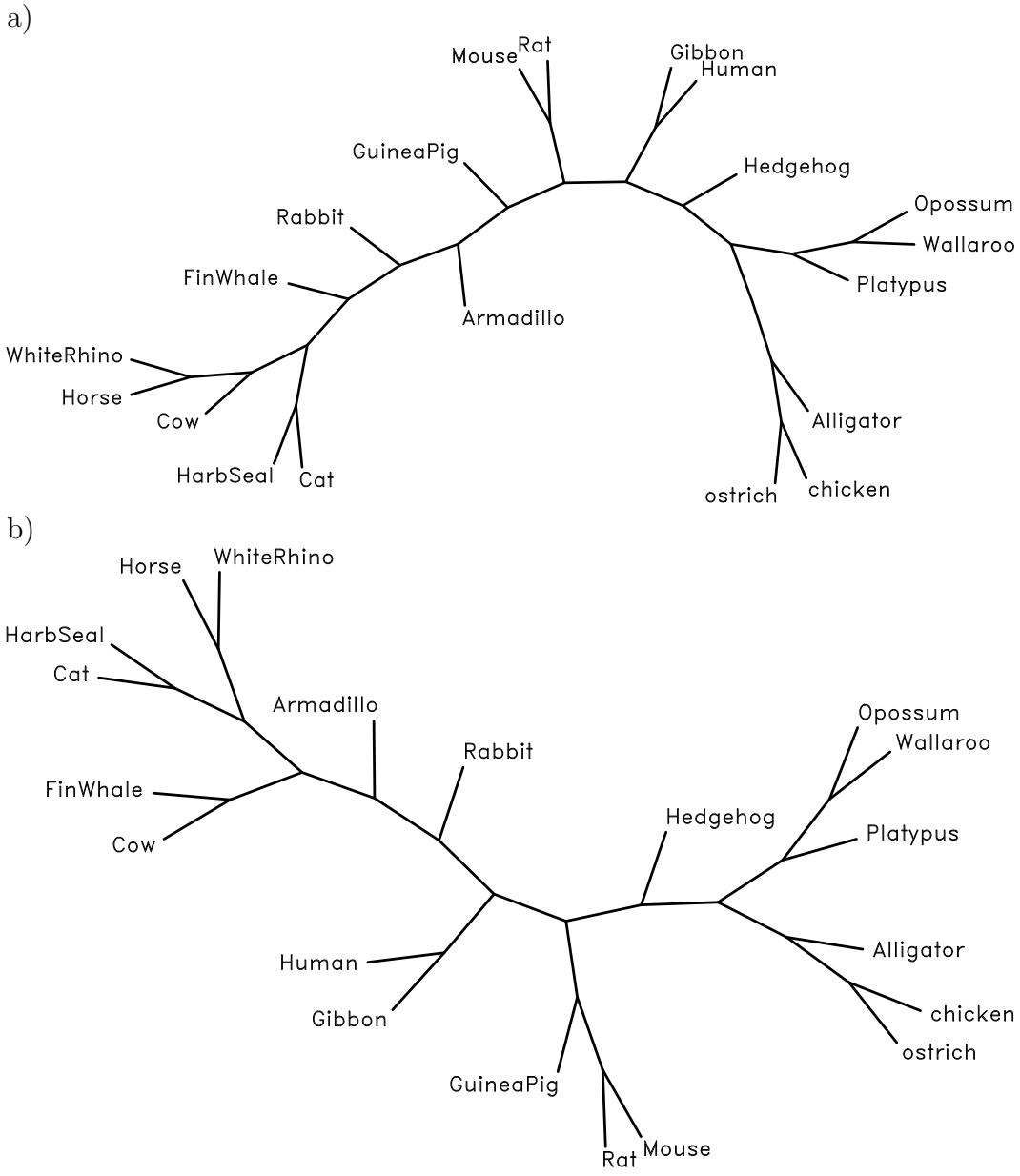

Fig. 3. Fig. 3a gives the topology of our solution, and Fig. 3b the topology obtained by Tree Puzzle, which is identical to the one given by Penny et al. (1999), if one removes the outgroups. The differences between Fig. 3a and Fig. 3b are: 1: In our solution, the whale is not grouped with the cow. 2: The guinea pig is not grouped with other rodents. 3: The rabbit is close to the primates. 4: The primates branched off earlier. The last three points are debated in Penny et al. (1999).

have also applied it to a set X of 36 taxa: 35 Mammals and the *Mustelus manazo* (shark) as outgroup: We analysed twelve mtDNA encoded proteins (all except ATP 8). The alignments have been obtained with CLUSTAL W (Thompson et al., 1994) using the default options and without manual adjustments. As above, we have used $w : ab \mid cd \mapsto -\left\langle l^{\{a,b\}}, l^{\{c,d\}} \right\rangle$ as confidence function

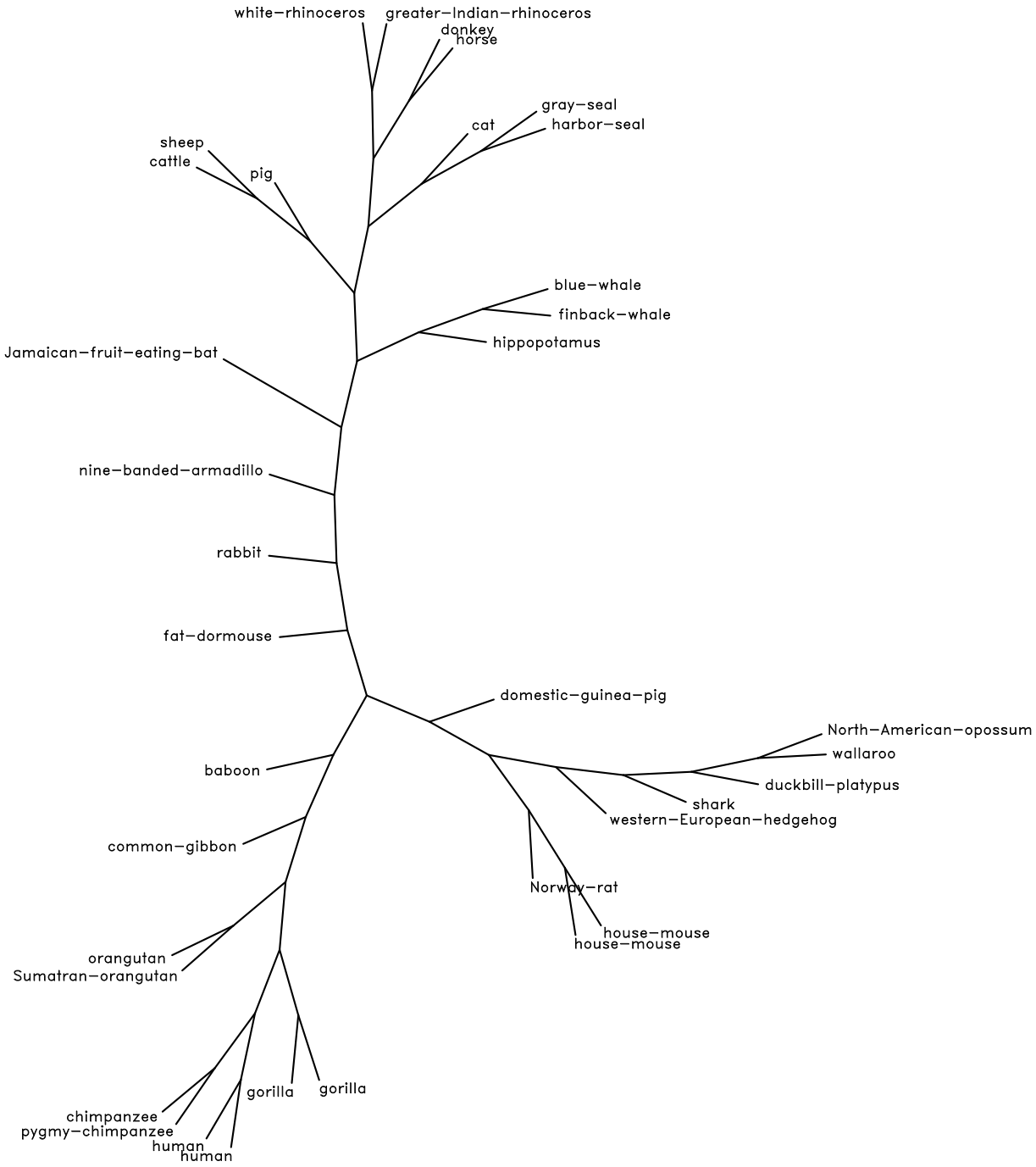

Fig. 4. Unrooted tree representing the solution of the Quartet Puzzle Problem searching for a Colonius Schulze quartet system with the weight function $w : ab \mid cd \mapsto -\left\langle l^{\{a,b\}}, l^{\{c,d\}} \right\rangle$ for an example of 36 taxa.

and found the best fitting Colonius-Schulze quartet system. The X-tree corresponding to the solution is shown in Figure 4. Its subtree on the 17 taxa used earlier has only two differences to the tree discussed in (Penny et al., 1999) (compare with Figure 3b): The placement of the guinea-pig and the placement

of the whale.

6 Conclusions

In this paper, we have reformulated the quartet puzzle problem as an integer linear programming problem, thus putting it into contact with a field that has been intensively studied for many decades. In addition, we have introduced a new confidence function, namely the negative scalar product of the diagonal of observed rate matrices.

We have applied these tools to mitochondrial data from mammals and reconstructed binary X-trees. They looked reasonable. So we conclude that the method to calculate confidence values from observed rate matrices and that the method to solve the quartet puzzle problem as integer linear programming problem is worth being developed further to a tool for phylogenetic analysis.

7 Acknowledgements

We thank Trish McLenachan for repreparing the aligned sequences used in (Penny et al., 1999). The work of Jan Weyer-Menkhoff was supported by the Graduate Program "Strukturbildungsprozesse" of the Deutsche Forschungsgemeinschaft at the University of Bielefeld, partly by the European Commission through grant number HPRI-CT-1999-00026 (the TRACS Programme at EPCC), and by DFG grant MO 1048/1-1. We also thank many people for their advice and suggestions, especially Burkhard Morgenstern and three anonymous reviewers.

References

Bandelt, H.-J., Dress, A., March 1992. A canonical decomposition theory for metrics on a finite set. Advances in Mathematics 92 (1), 47–105.

Bandelt, H.-J., Dress, A., 1993. A relational approach to split decomposition. Materialien / Universität Bielefeld, Forschungsschwerpunkt Mathematisierung 68.

Ben-Dor, A., Chor, B., Graur, D., Ophir, R., Pelleg, D., 1998. Constructing phylogenies from quartets: Elucidation of eutherian superordinal relationships. Journal of Computational Biology 5 (3), 377–390.

Berry, V., Feb 1999. Phyloquart 1.3 - a quartet phylogeny package. `http://www.lirmm.fr/~vberry/PHYLOQUART/phyloquart.html`

Berry, V., Gascuel, O., 2000. Inferring evolutionary trees with strong combinatorial evidence. Theoretical Computer Science 240 (2), 271–298.

Buneman, P., 1971. The recovery of trees from measures of dissimilarity. In: Hodson, F., Kendall, D., Tăutu, P. (Eds.), Proceedings of the Anglo-Romanian conference. The Royal Society of London and The Academy of the Socialist Republic of Romania, the University Press, Edinburgh, pp. 387–395.

Colonius, H., Schulze, H. H., 1977. Trees constructed from empirical relations. Braunschweiger Berichte aus dem Institut fuer Psychologie 1.

Devauchelle, C., Grossmann, A., Hénault, A., Holschneider, M., Monnerot, M., Risler, J. L., Torrésani, B., 2001. Rate matrices for analyzing large families of protein sequences. Journal of Computational Biology 8 (4), 381–399.

Dress, A., Huson, D., Moulton, V., 1996a. Analyzing and visualizing sequence and distance data using SPLITSTREE. Discrete Appl. Math. 71 (1-3), 95–109.

Dress, A., Moulton, V., Terhalle, W., 1996b. T-theory: An overview. Europ. J. Combinatorics 17, 161–175.

Farris, J., 1970. Methods for computing Wagner trees. Syst. Zool. 34, 21–24.

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

Felsenstein, J., 1993. Phylip (phylogeny inference package) version 3.5c. Distributed by the author, department of Genetics, University of Washington, Seattle.

Fitch, W. M., 1971. Toward defining the course of evolution: minimum change for a specific tree topology. Systematic Zoology 20, 406–416.

Huson, D. H., 1998. Splitstree: a program for analyzing and visualizing evolutionary data. Bioinformatics 14 (1), 68–73.

Huson, D. H., Bryant, D., 2005. Estimating phylogenetic trees and networks using splitstree4, in preparation. (jsplits) `http://www-ab.informatik.uni-tuebingen.de/software/jsplits/welcome_en.html`.

Penny, D., Hasegawa, M., Waddell, P. J., Hendy, M. D., 1999. Mammalian evolution: Timing and implications from using the logdeterminant transform for proteins of differing amino acid composition. Systematic Biology 48 (1), 76–93.

Saigal, R., 1995. Linear programming : a modern integrated analysis. Kluwer Academic Publishers, Boston.

Saitou, N., Nei, M., 1987. The neighbor-joining method: a new method for reconstructing phylogenetic trees. Molecular Biology and Evolution 4, 406–425.

Schmidt, H. A., Korbinian Strimmer, von Haeseler, A., 2003-2004. Tree-puzzle. `http://www.tree-puzzle.de`.

Schmidt, H. A., Strimmer, K., Vingron, M., von Haeseler, A., 2002. Tree-puzzle: maximum likelihood phylogenetic analysis using quartets and parallel computing. Bioinformatics 18, 502–504.

Schrijver, A., 1986. Theory of linear and integer programming. John Wiley & Sons.

Steel, M., 1992. The complexity of reconstructing trees from qualitive characters and subtrees. Journal of Classification 9, 91–116.

Strimmer, K., Goldman, N., von Haeseler, A., 1997. Bayesian probabilities and quartet puzzling. Mol. Biol. Evol. 14 (2), 210–211.

Strimmer, K., von Haeseler, A., 1996. Quartet puzzling: A quartet maximum-likelihood method for reconstructing tree topologies. Mol. Biol. Evol. 13 (7), 964–969.

Thompson, J., Higgins, D., Gibson, T., 1994. Clustal W: Improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position specific gap penalties and weight matrix choice. Nucl. Acids Res. 22, 4673–4680.

Weyer-Menkhoff, J., 2003. New quartet methods in phylogenetic combinatorics. Ph.D. thesis, Universität Bielefeld.