Information Processing Letters 114 (2014) 13–18

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

Fishing for minimum evolution trees with Neighbor-Nets

Sarah Bastkowski [a,*], Andreas Spillner [b], Vincent Moulton [a]

[a] *School of Computing Sciences, University of East Anglia, Norwich, NR4 7TJ, UK*
[b] *Department of Mathematics and Computer Science, University of Greifswald, Germany*

ARTICLE INFO

Article history:
Received 18 February 2013
Received in revised form 8 October 2013
Accepted 14 October 2013
Available online 21 October 2013
Communicated by M. Chrobak

Keywords:
Algorithms
Phylogenetics
Minimum evolution
Dynamic programming
Phylogenetic network
Circular split system

ABSTRACT

In evolutionary biology, biologists commonly use a phylogenetic tree to represent the evolutionary history of some set of species. A common approach taken to construct such a tree is to search through the space of all possible phylogenetic trees on the set so as to find one that optimizes some score function, such as the minimum evolution criterion. However, this is hampered by the fact that the space of phylogenetic trees is extremely large in general. Interestingly, an alternative approach, which has received somewhat less attention in the literature, is to instead search for trees within some set of bipartitions or splits of the set of species in question. Here we consider the problem of searching through a set of splits that is circular. Such sets can, for example, be generated by the NeighborNet algorithm for constructing phylogenetic networks. More specifically, we present an $O(n^4)$ time algorithm for finding an optimal minimum evolution tree in a circular set of splits on a set of species of size n. In addition, using simulations, we compare the performance of this algorithm when applied to NeighborNet output with that of FastME, a leading method for searching for minimum evolution trees in tree space. We find that, even though a circular set of splits represents just a tiny fraction of the total number of possible splits of a set, the trees obtained from circular sets compare quite favorably with those obtained with FastME, suggesting that the approach could warrant further investigation.

1. Introduction

A *phylogenetic tree* on a given set of species X is a connected, acyclic graph such that its leaf set is X and all its non-leaf vertices have degree at least three [24]. Such trees are used by biologists to represent the evolutionary history of the species in X. An important problem in phylogenetics is to construct such trees, and various methods have been developed for this purpose [18]. A common approach to tackling this problem is to search through the space of phylogenetic trees, trying to find a tree (or trees) that optimize some score such as the minimum evolution criterion [23]. However, a straight-forward exhaustive search is hampered by the fact that the space of phylogenetic trees on X grows exponentially in $n = |X|$. Moreover, it has been shown that finding an optimal tree is NP-hard for many of the popular optimization criteria (see e.g. [6,8]).

Interestingly, there is an alternative approach to searching through tree space, which was studied quite early on in the development of phylogenetics (see e.g. [10,21]), and more recently in [3], but that has received somewhat less attention in the literature. In particular, instead of searching through the set of all possible trees on the set X, we look for trees within a collection of bipartitions or *splits* of X. The rationale behind this approach is that any phylogenetic tree induces a set of splits of X in which every split corresponds to a branch of the tree, and that this set of splits uniquely determines the tree (cf. [24]). Intriguingly, in [4] a dynamic programming framework is developed to search for trees in a given collection of splits of X, also called a *split system*. Although still requiring exponential time in general, this approach has the advantage that it can yield polynomial time algorithms when restricted to

[*] Corresponding author. Tel.: +44 7407385089.
E-mail addresses: s.bastkowski@uea.ac.uk (S. Bastkowski), andreas.spillner@uni-greifswald.de (A. Spillner), vincent.moulton@cmp.uea.ac.uk (V. Moulton).

0020-0190/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ipl.2013.10.003

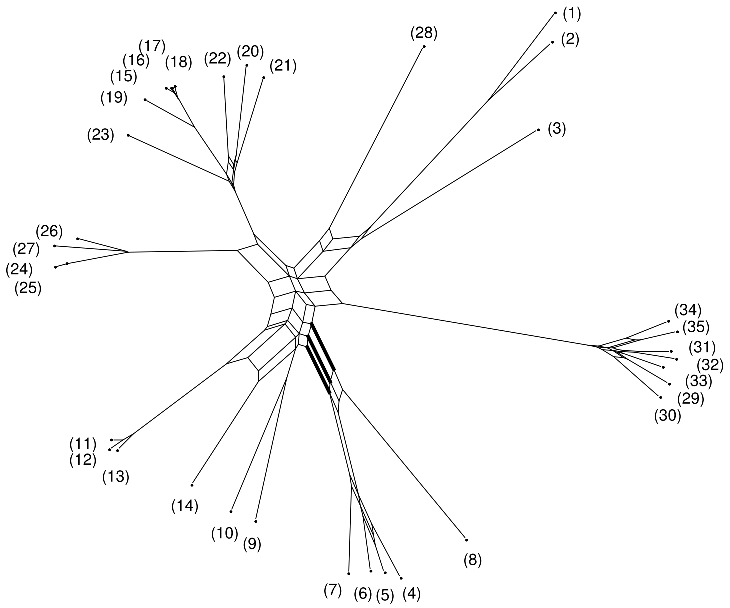

Fig. 1. A phylogenetic network generated with the NeighborNet algorithm displaying a circular split system for 35 tomato-infecting begomoviruses [22]. The group of three bold gray branches, for example, represents the split of $X = \{1, 2, \ldots, 35\}$ into the subsets $A = \{4, 5, 6, 7, 8\}$ and $B = X - A$.

split systems having size that is polynomial in $n = |X|$. It is therefore of interest to develop efficient algorithms to search for trees in special classes of split systems, as well as ways to generate split systems which capture salient information.

In this vein, here we develop an algorithm for searching for a tree that locally optimizes the minimum evolution criterion by searching in a *circular* split system. This is a special type of split system that can be generated, for example, by the NeighborNet algorithm [5] for constructing phylogenetic networks (see Fig. 1 for an example). In particular, we show that for a circular split system there is an $O(n^4)$ time algorithm for computing an optimal minimum evolution tree, which improves on the run time of $O(n^7)$ for the more general minimum evolution algorithm presented by Bryant in [4, Section 5.5]. We also present some simulations which indicate that minimum evolution trees in circular split systems generated by NeighborNet can compare favorably with those obtained by searching through the whole of tree space.

Before continuing, we note that, in view of the fact that split systems are often displayed by phylogenetic networks such as the one in Fig. 1, it might appear that the problem of searching for trees in split systems is closely related to the problem of finding optimal subtrees in phylogenetic networks. While some recent results on this latter problem can be found in [16,17], it is, in fact, quite different from the problem we study here since, for example, the minimum evolution tree in a circular split system generated by NeighborNet is not necessarily a subtree of the network used to display this split system.

The structure of the rest of this paper is as follows. After recalling some background material on the minimum evolution problem in the next section, in Section 3, we recall Bryant's dynamic programming algorithm for finding minimum evolution trees in a split system. We then describe our new algorithm in Section 4 and, in the following section, we present a short investigation into how the minimum evolution trees within split systems generated by NeighborNet and some related methods compare with those generated by FastME [11], one of the leading programs for finding minimum evolution trees by searching through tree space. We conclude with a discussion of some possible future directions in Section 6.

2. The minimum evolution problem

We begin by recalling some relevant terminology and notation (cf. also [24]). Let X be a finite, non-empty set, usually corresponding to some set of species or taxa. A *phylogenetic tree* (on X) is a connected, acyclic graph $T = (V, E)$ with leaf set X. Any non-leaf vertex of T is called an *internal vertex* of T, a branch incident to a leaf is called an *external branch* of T and a branch whose endpoints are both internal vertices is called an *internal branch* of T. In this paper, we consider only *binary* phylogenetic trees, that is, trees in which every interior vertex is incident to precisely three branches. Often the branches $e \in E$ of a phylogenetic tree $T = (V, E)$ are assigned a real number $\omega(e)$ known as the branch's *length*. The sum of the lengths of all branches in a phylogenetic tree T is called the *length* of T and denoted by $\ell_\omega(T)$. In addition, we denote the total length of the branches on the path connecting any two leaves x and y of T by $\ell_\omega(x, y)$.

When constructing phylogenetic trees, biologists often begin by computing a distance matrix D on X (estimated from, for example, molecular sequences) [18, Ch. 4], that is, a symmetric matrix D indexed by the set X which assigns the distance $D(x, y)$ between x and y to any pair x, y of elements in X and which is zero on the diagonal. Given