

Faster Reconstruction of Binary Near-Perfect Phylogenies

Kedar Dhamdhere ^{*}, Srinath Sridhar ^{*}, Guy E. Blelloch, ¹ Eran Halperin, ² R. Ravi, ³ Russell Schwartz ⁴

Keywords: phylogenetic trees, parsimony, near-perfect phylogeny

1 Introduction.

One of the classical problems of computational biology is to reconstruct an evolutionary tree for a set of taxa based on character data. Parsimony is one of the widely used metrics to solve the problem. It has been known to be particularly useful in the case when the evolutionary tree reconstructed is over a short period of time.

We borrow some of the following definitions and notations from [3]. The input to the problem is generally represented by a matrix I where rows R are string of states corresponding to a taxa. The columns $C = \{1, \dots, m\}$ are referred to as characters. The set of states corresponding to any character c is denoted by \mathcal{A}_c , therefore every taxon $s \in \mathcal{A}_1 \times \dots \times \mathcal{A}_m$. In a phylogenetic tree, each vertex v corresponds to a taxon and has an associated label $l(v) \in \mathcal{A}_1 \times \dots \times \mathcal{A}_m$.

Definition 1: A *phylogeny* for a set of n taxa R is a tree $T(V, E)$ with the following properties:

1. if a taxon $s \in R$ then $s \in l(V(T))$
2. for all $(u, v) \in E(T)$, $H(l(u), l(v)) = 1$ where H is the hamming distance

Definition 2: The *length* of a phylogeny T , $length(T) = |E(T)|$.

Definition 3: The penalty of a phylogeny T is defined as

$$penalty(T) = length(T) - \sum_{c \in C} (|\mathcal{A}_c| - 1)$$

Minimizing the length of a phylogeny is the problem of finding the most parsimonious tree. If the input I has the property that $penalty(T) = 0$ for the most parsimonious tree, then T is called a perfect phylogeny. Reconstructing a perfect phylogeny was proved to be NP-hard independently by Bodlaender et al. [2] and Steel [5]. This led researchers to work on fixed parameter versions of the problem (for e.g, [1], [4]). Lagergren and Fernandez-Baca, considered the problem of reconstructing near-perfect phylogenies [3] The assumption of a 'near'-perfect phylogeny is that $penalty(T)$ is small for the most parsimonious tree. Their algorithm runs in time $nm^{O(q)}2^{O(q^2r^2)}$, where r is the number of states per character, q is the penalty, n is the number of taxa and m is the number of characters.

^{*}Equally contributing authors

¹Computer Science Department, Carnegie Mellon University.

E-mail: { kedar, srinath, blelloch }@cs.cmu.edu

²International Computer Science Institute, Berkeley E-mail: heran@icsi.berkeley.edu

³Tepper School of Business, Carnegie Mellon University. E-mail: ravi@cmu.edu

⁴Department of Biological Sciences, Carnegie Mellon University.

E-mail: russells@andrew.cmu.edu

2 Results and Discussion

In our work, we consider an important special case of the problem when $r = |\mathcal{A}_c| = 2$ for all c . The case when $r = 2$ is primarily important because Single Nucleotide Polymorphisms (SNPs) are bi-allelic. We can therefore use the algorithm for reconstructing trees where the taxa are DNA sequences and the characters are SNP markers. We present a novel method for reconstructing near-perfect phylogenetic trees from binary character input. We show that if the penalty of the most parsimonious phylogeny is bounded by q , then we can reconstruct the phylogenetic tree in time $q^{O(q)}nm^2$. Our algorithm is almost entirely self-contained and its understanding requires only some fundamental theorems on phylogenetic trees. Although some existential proofs are hard, the algorithm itself is not very complicated to implement. We also expect the algorithm to perform significantly better than the above theoretical worst case bound.

In summary, the algorithm for reconstructing a perfect phylogeny for binary characters is computationally efficient but impractical in most real settings. For example, recurrently mutating bases violate the perfect phylogeny assumption. Near-perfect phylogenies however, can be expected to be more tolerant of real data sets. We believe that our method could lead to the first practical phylogenetic tree reconstruction algorithm that is both computationally feasible and robust to real biological data sets.

References

- [1] R. Agarwala and D. Fernandez-Baca. A Polynomial-Time Algorithm for the Perfect Phylogeny Problem when the Number of Character States is Fixed. In: *SIAM Journal on Computing*, 23 (1994). pp. 1216-1224.
- [2] H. Bodlaender, M. Fellows and T. Warnow. Two Strikes Against Perfect Phylogeny. In *proc. 19th International Colloquium on Automata, Languages and Programming, LNCS*, (1992). pp 273-283.
- [3] D. Fernandez-Baca and J. Lagergren. A Polynomial-Time Algorithm for Near-Perfect Phylogeny. In: *SIAM Journal on Computing*, 32 (2003). pp. 1115-1127.
- [4] S. Kannan and T. Warnow. A Fast Algorithm for the Computation and Enumeration of Perfect Phylogenies. In *SIAM Journal on Computing*, 26 (1997). pp 1749-1763.
- [5] M. A. Steel. The Complexity of Reconstructing Trees from Qualitative Characters and Subtrees. In *J. Classification*, 9 (1992). pp 91-116.