Tree Containment in Multifurcating Phylogenies

Matthias Bentert[1] and Mathias Weller[2]

[1]Technical University of Berlin, Germany, matthias.bentert@tu-berlin.de
[2]LIGM, CNRS, Université Gustave Eiffel, Marne-la-Vallée, France mathias.weller@u-pem.fr

Filière visée: Informatique, Biotechnologie

1 Introduction

The reconstruction of phylogenetic trees is one of the major computationally intense tasks in bioinformatics. However, trees are inapt at representing inter-species gene-flow, which is an important factor in the evolution of bacteria (and, in large parts, of plants) leading to the use of *networks* instead. A basic building block of this reconstruction process is the Tree Containment problem, which asks whether a given evolutionary tree is "compatible" with a given network (can be "embedded" in the network). As this problem is well-researched for bifurcating trees and networks, it is a logical next step to try and generalize existing results to multifurcating inputs. Herein, high-degree nodes represent a lack of knowledge of the "true", bifurcating process. Thus, multifurcating nodes can be seen as compatible with any "binary realization".

2 Goal

The problem, called *Soft Tree Containment* (STC), is NP-hard, even on restricted classes of networks [1] for which the binary version is linear-time solvable [6, 9]. In this project, we want to attack STC (and the closely related (Soft) Cluster Containment problem) using parameterized algorithmics (see [4]). While standard parameters, like the number of horizontal transfer events ("reticulations") in the given network, can be considered low-hanging fruit, we are curious if stronger parameterization ("level" [3], "invisible tree components" [9], "scanwidth" [2], . . .) can provide a handle to solve STC on practically relevant instances. Such instances are publicly available and a contribution of independent interest could be to categorize them by the behavior of the proposed parameters. A third direction might be the search for efficient approximation or heuristic algorithms and the analysis of their proposed solution in practice. If a sufficiently interesting theoretical result can be obtained, implementation of the developed algorithm and publication in a algorithmic-focused conference or journal is desired. For further literature, see [5, 7, 8].

References

[1] M. Bentert, J. Malík, and M. Weller. Tree containment with soft polytomies. In *16th Scandinavian Symposium and Workshops on Algorithm Theory, SWAT 2018, June 18-20, 2018, Malmö, Sweden*, volume 101 of *LIPIcs*, pages 9:1–9:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.

[2] V. Berry, C. Scornavacca, and M. Weller. Scanning phylogenetic networks is np-hard. In *SOFSEM 2020: Theory and Practice of Computer Science*, pages 519–530, Cham, 2020. Springer International Publishing. ISBN 978-3-030-38919-2.

[3] C. Choy, J. Jansson, K. Sadakane, and W.-K. Sung. Computing the maximum agreement of phylogenetic networks. *Theoretical Computer Science*, 335(1):93–107, 2005. ISSN 0304-3975. Pattern Discovery in the Post Genome.

[4] R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity.* Texts in Computer Science. Springer, 2013.

[5] A. D. Gunawan, B. Lu, and L. Zhang. A program for verification of phylogenetic network models. *Bioinformatics*, 32(17): i503–i510, 2016.

[6] A. D. M. Gunawan. Solving the tree containment problem for reticulation-visible networks in linear time. In *Proceedings of the 5th International Conference on Algorithms for Computational Biology AlCoB*, volume 10849 of *Lecture Notes in Computer Science*, pages 24–36. Springer, 2018.

[7] I. A. Kanj, L. Nakhleh, C. Than, and G. Xia. Seeing the trees and their branches in the network is hard. *Theoretical Computer Science*, 401(1-3):153–164, 2008.

[8] L. Van Iersel, C. Semple, and M. Steel. Locating a tree in a phylogenetic network. *Information Processing Letters*, 110(23): 1037–1043, 2010.

[9] M. Weller. Linear-time tree containment in phylogenetic networks. In *Proceedings of the 16th International Conference on Comparative Genomics RECOMB-CG*, volume 11183 of *Lecture Notes in Computer Science*, pages 309–323. Springer, 2018.