TreeScaper: Visualizing and extracting phylogenetic signal from sets of trees

Wen Huang  
*Rice University*

Guifang Zhou  
*Louisiana State University*

Melissa Marchand  
*Florida State University*

Jeremy R. Ash  
*NC State University*

David Morris  
*Louisiana State University*

*See next page for additional authors*

Follow this and additional works at: https://digitalcommons.lsu.edu/biosci_pubs

**Recommended Citation**


This Article is brought to you for free and open access by the Department of Biological Sciences at LSU Digital Commons. It has been accepted for inclusion in Faculty Publications by an authorized administrator of LSU Digital Commons. For more information, please contact ir@lsu.edu.
Authors
Wen Huang, Guifang Zhou, Melissa Marchand, Jeremy R. Ash, David Morris, Paul Van Dooren, Jeremy M. Brown, Kyle A. Gallivan, and Jim C. Wilgenbusch
TreeScaper: Visualizing and Extracting Phylogenetic Signal from Sets of Trees

Wen Huang,*,1 Guifang Zhou,2 Melissa Marchand,3 Jeremy R. Ash,4 David Morris,2 Paul Van Dooren,5 Jeremy M. Brown,2 Kyle A. Gallivan,3 and Jim C. Wilgenbusch6

1Department of Computational and Applied Mathematics, Rice University, St. Houston, TX
2Department of Biological Sciences and Museum of Natural Science, Louisiana State University, Baton Rouge, LA
3Department of Mathematics, Florida State University, Tallahassee, FL
4Bioinformatics Research Center, North Carolina State University, Raleigh, NC
5Department of Mathematical Engineering, ICTEAM, Université catholique de Louvain, Belgium, Germany
6Minnesota Supercomputing Institute, University of Minnesota, Minneapolis, MN

*Corresponding author: E-mail: huwst08@gmail.com.
Associate editor: Keith Crandall

Abstract

Modern phylogenomic analyses often result in large collections of phylogenetic trees representing uncertainty in individual gene trees, variation across genes, or both. Extracting phylogenetic signal from these tree sets can be challenging, as they are difficult to visualize, explore, and quantify. To overcome some of these challenges, we have developed TreeScaper, an application for tree set visualization as well as the identification of distinct phylogenetic signals. GUI and command-line versions of TreeScaper and a manual with tutorials can be downloaded from https://github.com/whuang08/TreeScaper/releases. TreeScaper is distributed under the GNU General Public License.

Key words: TreeScaper, visualization, community detection, phylogenetic trees.

Introduction

As phylogenetic analyses have matured to handle genome-scale data, two major sources of variation in phylogenetic trees have become increasingly important to consider. First, estimates of gene trees often have considerable uncertainty, given low evolutionary rates in conserved genomic regions targeted for phylogenetics. Second, phylogenetic estimates often vary considerably across genes due to both biological (e.g., coalescent variation) and non-biological (e.g., systematic error) causes. Accurate reconstructions of phylogenetic relationships and meaningful insights into genomic evolution must consider both sources of variation, and many phylogenetic analyses return a set of trees to represent such variation. Most methods currently used to analyze or summarize tree sets involve substantial loss of information, as the set is condensed into a point estimate or visualized without any formal or quantitative summary (Hillis et al. 2005). These constraints can impose significant limitations on our ability to extract phylogenetic information and draw biological conclusions. Recent advances have been made in methods for calculating distances between trees (e.g., SPR distances, Whidden et al. 2016), visualizing sets of trees (Hillis et al. 2005), and summarizing the variation in phylogenetic signal (Lewitus and Morlon 2015; Gori et al. 2016). However, existing software tools are focused on a subset of these tasks, despite their synergism in allowing users to explore and extract information. Here, we describe TreeScaper, a software tool that brings together much of this functionality and also provides new approaches to accomplishing these goals.

Overview of TreeScaper

TreeScaper allows users to accomplish many different tasks, including (i) computing pairwise distances between trees with a variety of different metrics, (ii) projecting and visualizing trees in low dimensional Euclidean space, (iii) estimating the intrinsic dimensionality of the space formed by the tree set, (iv) computing the covariance matrix of bipartition presence/absence across trees, and (v) finding communities of bipartitions or trees using state-of-the-art community detection methods. Many of these functions are not available in any other software implementation of which we are aware.

Below we provide an overview of two general tasks in TreeScaper: the visualization of tree sets in two or three dimensions using nonlinear dimensionality reduction (NLDR), and the detection and characterization of distinct phylogenetic signals within tree sets using community detection on graphs. Figure 1 highlights TreeScaper’s capabilities both to visualize treespace and detect distinct communities of trees. Because community detection operates on the original tree-to-tree distances, it may detect subtle relationships that NLDR does not reveal. For this dataset, both NLDR and community detection reveal a partitioning of the tree set that perfectly corresponds to the different genes from which the phylogenies were inferred.
Nonlinear Dimensionality Reduction

NLDR seeks to find low dimensional representations of a set of high dimensional data. TreeScaper begins by computing pairwise tree distances between trees using one of several metrics, such as Robinson-Foulds (Robinson and Foulds 1981), matching (Bogdanowicz and Giaro 2012), or subtree prune and regraft (SPR) distances (following Whidden et al. 2010). NLDR then looks for low dimensional points \( f(x_i) \) in the Euclidean space that minimize the distortions of pairwise distances. Specifically, given \( n \) phylogenetic trees \( t_1, t_2, \ldots, t_n \), the optimization problem is

\[
\min_{x_1, x_2, \ldots, x_n \in \mathbb{R}^d} f(d_{ij}; T_{ij}) \left( |d_y - T_{ij}| \right),
\]

where \( d_y \) is the Euclidean distance between \( x_i \) and \( x_j \), and \( T_{ij} \) is the tree distance between \( t_i \) and \( t_j \). Multiple cost functions \( f(d_{ij}; T_{ij}) \) are implemented in TreeScaper (e.g., normalized, Kruskal’s, and Sammon’s stress functions, as well as curvilinear component analysis (see Lee and Verleysen 2007, for details).

Community Detection Methods

A network has community structure if its nodes can be easily clustered into sets with dense, internal connections. In phylogenetic analysis, community structure can be used to identify distinct topological signals. TreeScaper uses two distinct network types to accomplish this: networks of trees or bipartitions. Tree networks employ edge weights based on tree affinities (a decreasing function of a user-specified tree distance). Bipartition networks use edge weights indicating whether certain bipartitions are found in the same trees more or less often than expected by chance (i.e., their covariance). Whereas tree distances have previously been used in conjunction with other methods for detecting distinct topological signals (Gori et al. 2016; Lewitus and Morlon 2015), bipartition covariances are unique to TreeScaper.

There are a number of different methods to detect communities (Blondel et al. 2008). Four models are included in TreeScaper: No Null Model (Newman and Girvan 2004), Configuration Null Model (Newman 2006), Erdős–Rényi Null Model (Reichardt and Bornholdt 2006), and the Constant Potts Model (CPM; Traag et al. 2011). The CPM belongs to a family of approaches that includes resolution-limit-free methods. This family can accommodate a mixture of positive and negative weights (Traag and Bruggeman 2009), which is important for community detection with bipartition covariance networks.

Conclusion

TreeScaper provides an integrated, lightweight platform for exploring the phylogenetic information in large sets of trees, through both a GUI and a command-line interface. By providing a multitude of related functions in a single package with an intuitive interface, TreeScaper facilitates adoption among new users and naturally lends itself to integration in larger analytical pipelines (e.g., Galaxy). We are actively integrating other tree-to-tree distances and graph partitioning (clustering) methods, as well as developing new approaches. TreeScaper’s existing architecture for handling and processing
phylogenetic data structures greatly facilitates the development of new methods.

Acknowledgments
The authors thank Chris Whidden for help with SPR distance calculations, as well as Michael Landis, and an anonymous reviewer for insightful comments that improved the manuscript. This work was supported by the US National Science Foundation under DBI-1262571 to JMB and DBI-1262476 to KAG and JW. This paper presents research results of the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization) under IAP VII/19, funded by the Interuniversity Attraction Poles Programme initiated by the Belgian Science Policy Office.

References