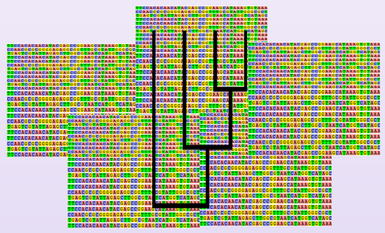
TreeScaper: identifying phylogenetic communities

Genevieve G. Mount[1], Diego Elias[1], David Morris[1], Jeremy Ash[2], Wen Huang[3], Melissa Marchand[4], Kyle A. Gallivan[4], James C. Wilgenbusch[5], Jeremy M. Brown[1]

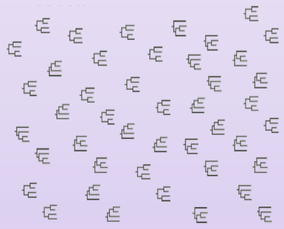
[1]Department of Biological Sciences and Museum of Natural Science, Louisiana State University, Baton Rouge, LA, [2]Bioinformatics Research Center, North Carolina State University, Raleigh, NC, [3]Department of Computational and Applied Mathematics, Rice University, St. Housten, TX, [4]Department of Mathematics, Florida State University, Tallahassee, FL, [5]Minnesota Supercomputing Institute, University of Minnesota, Minneapolis, MN

Common Problem

Genome-scale data results in hundreds to thousands of gene trees. **Summary tree** methods provide an easy to digest single answer, useful for downstream analyses. However, reducing thousands of trees to a single topology may **lose important information** and phylogenetic signal [1]



Should genome wide datasets be reduced to a single topology?



How do we parse the major phylogenetic signal from a large set of trees with no *a prior* knowledge of the system?

- Identifying and quantifying **conflicting signals** can be challenging[2].
- **Visual representation** of tree space is useful in identifying conflicting signals
- A more quantitative approach, such as **community detection**, may be able to identify communities when visual evidence is inconclusive.
- **Communities** of trees can be then summarized to understand the major topological conflicts within the tree set.

TreeScaper

TreeScaper allows users to explore tree sets and the information within both visually and quantitatively using **nonlinear dimensionality reduction** and **network based community detection**[3].

Visualization of tree space

- Pairwise tree distances are mapped onto lower dimensional space using nonlinear dimensionality reduction (NLDR), and then viewed in 2D or 3D. **Figure 1**
- TreeScaper implements multiple types of tree distance calculations such as Robinson-Foulds, matching, and subtree-prune-regraft to optimize exploration of the relationship between trees.

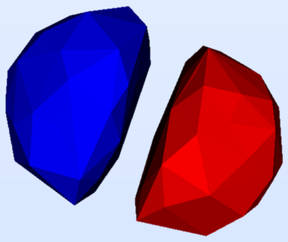


Fig 1. NLDR plot of two communities, graphed in 3D

Community detection using networks

- Networks consists of nodes and edges.
- Nodes in our networks represent either trees or bipartitions.
- Edges represent connections between nodes.
- Communities are identified as sets of nodes whose connections are denser than their connections to nodes external to the community.
- Phylogenetic communities can indicate distinct topological signals.

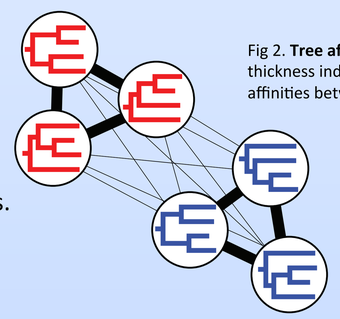


Fig 2. **Tree affinity** network. Line thickness indicates strength of affinities between topologies

TreeScaper uses two different types of networks to identify community structure:

Tree affinity

Matrix of pairwise **tree distances** are converted to affinities Topologies with high affinities (low tree to tree distances) are clustered together. Figure 2

Bipartition covariance

Bipartitions have positive covariances when they are found together more often than expected by chance. Figure 3

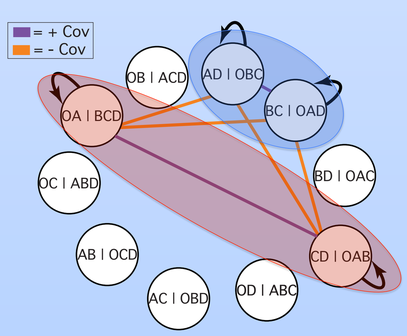


Fig 3. **Bipartition covariance** network. Line color indicates whether bipartitions are more or less likely to be found together than random chance.

Methods

Goal - Identify general patterns of when TreeScaper can accurately identify communities

TreeScaper - https://github.com/whuang08/TreeScaper/releases

Python Scripts - https://github.com/LizEve

Run TreeScaper

- Tree sets were **visualized** with **NLDR** using unweighted Robinson Foulds (RF) distances.
- **Community detection** was run using biparition covariance networks and tree affinity networks.
- Community detection was run using all available community detection models: No Null Model, Erdós-Rényi Model, Configuration Null Model, Constant Potts Model

Simulate tree sets

- Clusters of rooted trees were generated using random seed topologies. Additional trees for each cluster were generated by making NNI moves on the seed tree.
- Number of tips: 10 - 100
- Number of trees per cluster: 200
- Distance between seed trees: 1.0 normalized RF
- Distance from seed tree to edge of cluster: 0.043 - 1.0 normalized RF

Results

Comparing NLDR and community detection

- In some cases, visual inspection of NLDR plots is more accurate than community detection with affinity networks.
- **Figure 4** shows a case where affinity network methods do not identify two communities.
- In a tree sets with 10 tips, bipartition covariance networks can identify two distinct communities while NLDR plots are ambiguous. **Figure 6**

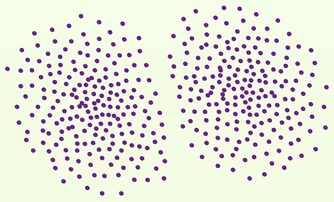


Fig 4. NLDR plot decidedly shows 2 communities.

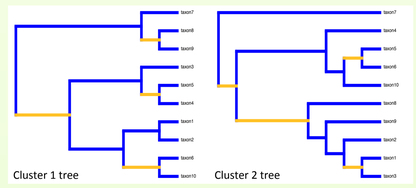


Fig 5. Most common bipartitions in each community are plotted on the two cluster seed trees.

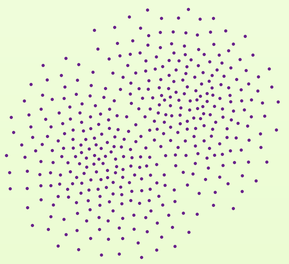


Fig 6. NLDR plot is ambiguous. Similarly affinity networks cannot detect accurate community structure in this case

Number of tips influence efficacy of network method

- Bipartition covariance networks appropriately identify two communities in tree sets with 10 to 25 tips when affinity networks did not. **Figure 5, 6, 7**
- Tree affinity networks were more reliable for trees sets with more tips. **Figure 7**

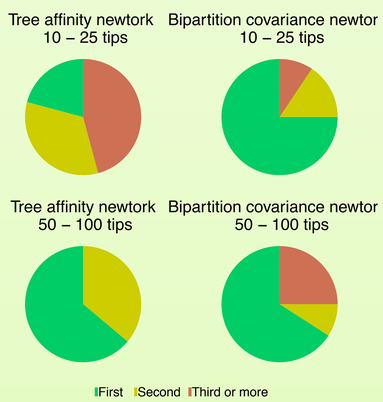


Fig 7. First indicates that the correct number of communities was the first number suggested by TreeScaper. Second indicates an alternate number was preferred.

Primary vs. secondary community structure

- Community detection identifies multiple potential levels of communities. As noted in the manual, the second number of communities detected is sometimes more appropriate than the first.
- We recommend having an a prior range of communities in mind while examining results, to determine if the second result is more reasonable.

Detection algorithms

- Constant Potts Model (CPM) accurately identified 2 communities most reliably, however in some cases other detection models identified 2 communities when CPM did not. Further exploration is warranted.

Future plans

These results are preliminary and include a few different tree set conditions. Future work will involve testing what methods in TreeScaper are most useful for detection phylogenetic signal under tree sets with different characteristics that mimic potential empirical datasets.

References:

1. Hillis DM, Heath TA, St John K. 2005. Analysis and visualization of tree space. Syst Biol 54(3):471–482.
2. Wilgenbusch JC, Huang W, Gallivan KA. 2017. Visualizing phylogenetic tree landscapes. BMC Bioinf. 18:85 DOI 10.1186/s12859-017-1479-1
3. Huang W, Zhou G,Marchand M, Ash JR, Morris D, Van Dooren P, Brown JM, Gallivan KA, Wilgenbusch J C. 2016 TreeScaper: Visualizing and Extracting Phylogenetic Signal from Sets of Trees. MBE. 33(12)3314-3316

Acknowledgements: This work was supported by NSF grant DBI-1262571 to JMB and DBI-1262476 to KAG and JW. Thank you to Jeremy Brown for putting up with me. Many thanks for Melissa and Wen for fixing bugs quickly. Subir Shakya for helping with python scripts.

We are interested in feedback!

How would you like use TreeScaper in your research?

What do you want to know about TreeScaper before using it?