Evaluating the use of Monte Carlo simulation for statistically assessing topological congruence of phylogenetic trees

James G. DuBose[1*]

[1]Department of Biology, Emory University
* james.g.dubose@gmail.com

Abstract

Many questions in evolutionary biology involve assessing the topological congruence between phylogenetic trees. However, quantification of tree congruence provides a single measurement that is often difficult to contextualize and use to make inferences. One approach to statistically inform measurements of tree congruence is to generate null congruence distributions using Monte Carlo simulations, which can then be used for hypothesis testing. Although this approach has previously been used in empirical studies, its validity and effectiveness have yet to be tested. Here, I evaluate this method in combination with a variety of commonly used congruence metrics. I first explore how the number of simulations conducted influences the stability of null model generation. I then explore the error and power associated with different congruence metrics and tree sizes. These findings reveal nuances to this approach that are important for researchers to consider when using it to make evolutionary inferences. Finally, I introduce *manticore*, an R library that provides a simple interface for researchers to utilize this approach.

Introduction

Many areas of study in evolutionary biology often involve quantifying the topological similarity between phylogenetic trees. For example, investigations of co-evolution and co-diversification [1], convergent evolution [2], and horizontal gene transfer [3], are often prefaced by evaluating the congruence between phylogenetic trees that describe the system. Furthermore, new phylogenetic reconstruction methods are often evaluated by quantifying the congruence between reconstructed trees and a trees known to be true [4]. As a result, a variety of metrics have been developed for quantifying tree congruence and dissimilarity. However, such metrics only provide a single measurement of tree congruence that can be difficult to contextualize and use make statistically informed inferences from.

One approach to statistically inform measurements of tree congruence is to use Monte Carlo simulations to generate a null distribution, which can then be used for hypothesis testing. This approach has been previously used in several empirical studies, but its validity and effectiveness as a statistical test have not been evaluated [5,6]. Choice of congruence metric is likely one of the most important factors that influence the effectiveness of this approach, as different metrics capture different aspects of congruence [7–10]. The Robinson-Foulds distance is among the first and most widely used metrics for quantifying congruence between trees [11]. However, more recent advances have uncovered several undesirable characteristics of this metric and yielded alternative metrics with more suitable properties [4,7–10].

To help researchers contextualize quantifications of tree congruence, here I evaluate the use of Monte Carlo simulations for statistically assessing tree congruence across a variety of commonly used congruence metrics. Because this approach relies on randomization, I first explore how the number of simulations conducted influences the stability of null model generation. I then explore the error and power associated with different congruence metrics and tree sizes, which can be used to inform future applications of this method. Finally, I introduce *manticore*, an R library that provides a simple interface for researchers to utilize this approach.

Methods

Congruence metrics

The Robinson-Foulds (RF) distance, which has historically been the most widely used metric, measures the number of splits that are present in one tree but not the other [11]. This calculation has been criticized as too conservative, as it does not account for similar but not identical splits [4]. Therefore, a variety of approaches for addressing this issue have been developed, including the generalized Robinson-Foulds metric, Nye similarity (NS) [8], and the Jaccard-Robinson-Foulds distance (JRF) [9]. The matching split distance (MSD) take a different perspective by finding the matchings of splits between trees that minimizes dissimilarity [10]. More recently, information-theoretic approaches have been introduced to weight splits based on their information content, which has significantly improved quantification of tree congruence [7]. This has led to the development of new information-theoretic tree congruence metrics, such as the information-corrected Robinson-Foulds (ICRF), matching split information distance (MSID), mutual clustering information (MCI), and shared phylogenetic information (SPI) [7].

Monte Carlo algorithm for generating null congruence models

Generation of a null congruence model starts with the specification of a reference tree and a comparison tree, each with n leaves. A null congruence distribution is then generated by using Monte Carlo simulations to evaluate the congruence between i random trees with n leaves and the reference tree. The observed congruence between the reference tree and the comparison tree is then compared to the null distribution, and statistical significance of the observed congruence is then quantified as the proportion of random trees that showed greater than or equal congruence with the reference tree ($P(Null \geq Observed)$). This allows for testing the null hypothesis that the observed congruence is not greater than randomly expected. For convention, I hereafter refer to this method as a *random tree congruence test*. For methodological evaluations and implementation in the later described R library, I used the *rtopology* function from the *ape* R library (v5.7.1) to generate random trees [12] and the various functions from the *TreeDist* R library (v2.7.0) for quantifying congruence [13].

Random tree congruence test evaluations

To investigate the effect of simulation number of the stability of *P(Null ≥ Observed)* across independently simulated null distributions, I first simulated 10 random pairs of trees for trees with 5, 10, 20, 40, 60, 80, and 100 leaves. For each pair, I conducted 10 independent random tree congruence tests with 10^1, 10^2, 10^3, and 10^4 iterations. For each number of iterations, I then calculated the standard deviation in *P(Null ≥ Observed)* across replicate tests to evaluate the variation that can be expected with each number of iterations. I performed this procedure for each of the aforementioned congruence metric using the same pairs of initially generated trees.

To investigate the error and power associated with the random tree congruence test when using the various congruence metrics, I first simulated 1000 random pairs of trees for each of the previously described tree sizes. For each pair, I conducted a random tree congruence test using each congruence metric and 1000 iterations (which I found to be a sufficient number based on the previously described analysis) and recorded the *P(Null ≥ Observed)* values. For a statistical test with an acceptable error rate, the expected distribution of *P(Null ≥ Observed)* values for randomly generated data should follow an approximate uniform distribution, with *P(Null ≥ Observed)* values of 0.05 or less occurring at a relative frequency of 0.05, values of 0.1 or less occurring at a relative frequency of 0.1, etc. Therefore, the expected cumulative density function can be described as $F(P) = P$. For each metric and each tree size, I used Kolmogorov-Smirnov tests to compare the distribution of *P(Null ≥ Observed)* values to the expected distribution. I used the *ks_1samp* function from the *SciPy* Python library (v1.11.4) to perform all Kolmogorov-Smirnov tests [14].

To evaluate the power of random tree congruence tests when paired with the aforementioned congruence metrics, I used the previously described simulated data set to quantify the proportion of significant results (at an $\alpha = 0.05$) as a function of congruence. In order to compare congruence on a common scale, I first scaled congruence values to between 0 and 1 using min-max normalization. I then fit logistic models to describe the proportion of significant results ($\alpha = 0.05$) as a function of scaled congruence using the *Logit* function from the *statsmodels* Python library (v0.14.0) [15]. Note that in several cases, there were not enough significant results detected for proper model fitting, which are depicted as $y=0$ lines on their corresponding plots.

The manticore R library

To help researchers use the algorithm presented and investigated in this study, I have created *manticore*: *Monte cArlo simulaioN algoriThm for assessIng tree COngRuencE*. *Manticore* is an open-source R library that provides a simple interface for conducting random tree congruence tests (https://github.com/gabe-dubose/manticore). *Manticore* currently supports the congruence metrics described in this study but can also grow support additional metrics that may be developed as the study of tree congruence quantification advances.

Results

The effect of Monte Carlo simulation number on null distribution stability

The number of Monte Carlo simulations performed can affect the stability of the null distribution. Performing more simulations is expected to reduce the variation between independently generated null distributions, which gives greater statistical confidence. However, performing an excessive number of simulations can become computationally costly, and may not be practical for many applications (such as scanning a number of gene trees against species trees to evaluate horizontal gene transfer). Therefore, I examined how much variation in *P(Null ≥ Observed)* can be expected across four orders of magnitude in iterations. This showed that for each metric and across all tree sizes, 10^3 iterations were sufficient for reducing the standard deviation across independent evaluations to less than half of the typical α of 0.05 (Figure 1). Furthermore, using 10^4 iterations nearly eliminated the variation between independently conducted tests (Figure 1).

The effects of tree size and congruence metric on conformity to the expected error distribution

The distributions of *P(Null ≥ Observed)* values for the majority of tree sizes and congruence metrics did not significantly deviate from the expected error distribution ($D \leq 0.035$, $p \geq 0.08$) (Figure 2). However, there were several exceptions. The most striking (though least surprising) pattern was that all congruence metrics produced *P(Null ≥ Observed)* distributions that significantly deviated from expectations with trees of 5 leaves ($D \geq 0.493$, $p < 2e^{-16}$). Visual inspection of the empirical cumulative density functions revealed that this distribution was significantly biased towards higher values (Figure 2). This issue was alleviated for most metrics when the number of leaves was increased to 10. However, use of the ICRF, and MSD metrics still resulted in slight deviation from expectations for trees with 10 leaves ($D \geq 0.121$, $p < 2e^{-16}$) (Figure 2A, B, and G). Finally, use of the RF metric resulted in *P(Null ≥ Observed)* distributions that were significantly biased towards higher values for all tree sizes ($D \geq 0.629$, $p < 2e^{-16}$) (Figure 2A).

The effects of congruence metric and tree size on power

Consistent with previous analyses, random tree congruence tests failed to detect significance congruence between pairs of trees with 5 leaves, regardless of the metric used (Figure 3). For larger trees, the power of random tree congruence tests varied depending on the congruence metric used. Unsurprisingly, use of the RF metric failed to detect significant congruence between even the most congruent of trees for any tree size (Figure 3A). Use of JRF, MSD, and MSID resulted in more conservative statistical estimates, where only highly congruent pairs (scaled congruence ≥ ~0.8) were identified as significantly congruent at a frequencies of approximately 0.8 and greater (Figures 3C, G, H). In contrast, use of NS, MCI, and SPI facilitated greater power to detect subtle congruence, where comparatively less congruent trees (scaled congruence between > ~0.5) were identified as significantly congruent at a frequencies of 0.8 and greater (Figures 3E, F). Finally, use of ICRF generally failed to detect statistical significance in the most congruent pairs of trees (scaled congruence ≥ ~0.8) at frequency of 0.8 and greater (Figure 3B), suggesting its use results in limited power. Furthermore, use of ICRF

was unable to reliably detect significant congruence in pairs of trees with 10 leaves (Figure 3B).

Discussion

Evaluating the use of Monte Carlo simulations for statistically assessing tree congruence revealed several features of this approach that are useful to consider when applying it to data. First, the results depicted in Figure 1 may be helpful for determining the number of simulations to perform, where using 10^3 simulations is likely sufficient, particularly if dealing with a large set of trees. However, if the resulting *P(Null* $\geq$ *Observed)* value plus the expected variability associated with 10^3 iterations overlaps with α, re-conducting the test using 10^4 simulations would allow for more confidence in the result. The same can be said for use of 10^4 simulations, but is the issue is less likely given its significant reduction in variability across repeated null simulations.

In terms of properties as a statistical test, it is apparent that this approach is not suited to be used with the RF distance, as it failed to reliably identify significant congruence between even the most congruent of trees (Figure 2A and 2B). This finding is consistent with the overly conservative nature of the RF distance, which has been extensively quantified and discussed [4,7]. Given this, the following discussion pertains only to the other congruence metrics evaluated in this study.

A more intuitive finding was that this approach is also not suited for making inferences regarding smaller trees (n leaves = 5), regardless of the congruence metric used. Smaller trees have fewer unique topologies and therefore, more limited null distributions. However, this approach generally showed expected properties for trees with 10 leaves or greater when using most congruence metrics. Use of the ICRF distance was an exception, which still showed high degrees of type 2 error for trees with 10 leaves (Figures 2B and 3B). Use of ICRF also resulted in comparatively less power than other metrics, for both modestly and highly congruent trees (Figure 3B). More generally, use of different metrics resulted in varying degrees of power to detect more subtle congruence. JRF, MSD, and MSID resulted in more conservative estimates, where only the most congruent of trees were reliably identified as significant (Figure 3C, G, H). Alternatively, use of NS, MCI, and SPI resulted in greater power to detect more subtle congruence, where more modestly congruent tress were also reliably identified as significant (Figure 3D, E, F). In addition to the differences in how congruence is evaluated between metrics, these findings provide important context to consider when applying Monte Carlo simulations to statistically assess tree congruence.

Data and code availability

No data was generated for this manuscript. All code written for analyses (including the *manticore* R library) is available at https://github.com/gabe-dubose/manticore and has been archived under the DOI 10.5281/zenodo.14828441.

Competing interests

I have no competing interests.

Funding

I received no funding for this study.

References

1. Martínez-Aquino A. 2016 Phylogenetic framework for coevolutionary studies: a compass for exploring jungles of tangled trees. *Current Zoology* **62**, 393–403. (doi:10.1093/cz/zow018)

2. Partha R, Kowalczyk A, Clark NL, Chikina M. 2019 Robust Method for Detecting Convergent Shifts in Evolutionary Rates. *Molecular Biology and Evolution* **36**, 1817–1830. (doi:10.1093/molbev/msz107)

3. Ravenhall M, Škunca N, Lassalle F, Dessimoz C. 2015 Inferring Horizontal Gene Transfer. *PLoS Comput Biol* **11**, e1004095. (doi:10.1371/journal.pcbi.1004095)

4. Smith MR. 2022 Robust Analysis of Phylogenetic Tree Space. *Systematic Biology* **71**, 1255–1270. (doi:10.1093/sysbio/syab100)

5. Brooks AW, Kohl KD, Brucker RM, Van Opstal EJ, Bordenstein SR. 2016 Phylosymbiosis: Relationships and Functional Effects of Microbial Communities across Host Evolutionary History. *PLoS Biol* **14**, e2000225. (doi:10.1371/journal.pbio.2000225)

6. Trevelline BK, Sosa J, Hartup BK, Kohl KD. 2020 A bird's-eye view of phylosymbiosis: weak signatures of phylosymbiosis among all 15 species of cranes. *Proc. R. Soc. B.* **287**, 20192988. (doi:10.1098/rspb.2019.2988)

7. Smith MR. 2020 Information theoretic generalized Robinson–Foulds metrics for comparing phylogenetic trees. *Bioinformatics* **36**. (doi:10.1093/bioinformatics/btaa614)

8. Nye TMW, Liò P, Gilks WR. 2006 A novel algorithm and web-based tool for comparing two alternative phylogenetic trees. *Bioinformatics* **22**, 117–119. (doi:10.1093/bioinformatics/bti720)

9. Böcker S, Canzar S, Klau GW. 2013 The Generalized Robinson-Foulds Metric. In *Algorithms in Bioinformatics* (eds A Darling, J Stoye), pp. 156–169. Berlin, Heidelberg: Springer Berlin Heidelberg.

10. Bogdanowicz D, Giaro K. 2012 Matching Split Distance for Unrooted Binary Phylogenetic Trees. *IEEE/ACM Trans. Comput. Biol. and Bioinf.* **9**, 150–160. (doi:10.1109/TCBB.2011.48)

11. Robinson DF, Foulds LR. 1981 Comparison of phylogenetic trees. *Mathematical Biosciences* **53**, 131–147. (doi:10.1016/0025-5564(81)90043-2)

12. Paradis E, Schliep K. 2019 ape 5.0: an environment for modern phylogenetics and evolutionary analyses in R. *Bioinformatics* **35**, 526–528. (doi:10.1093/bioinformatics/bty633)

13. Smith MR. 2020 *TreeDist: Distances between Phylogenetic Trees. R package version 2.6.2*. (doi:10.5281/zenodo.3528124)

14. Virtanen P *et al.* 2020 SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* **17**, 261–272. (doi:10.1038/s41592-019-0686-2)

15. Seabold S, Perktold J. 2010 statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*,

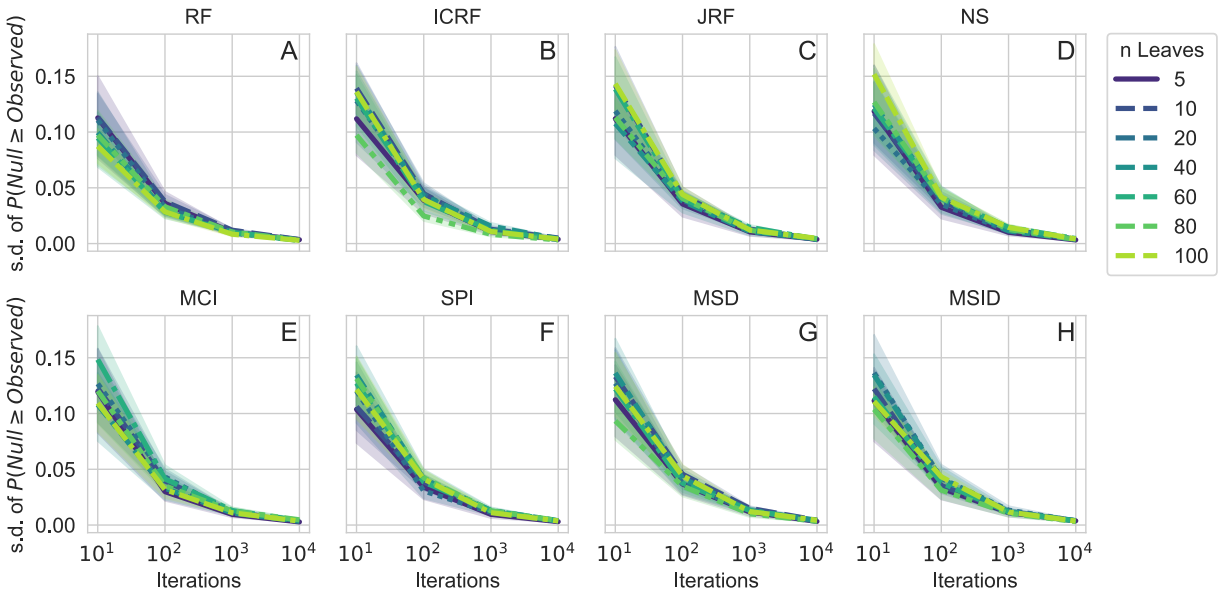

Figure 1. The effect of Monte Carlo simulation number on null distribution stability. In each panel, the y-axis represents the standard deviation across repeated null model simulations, with higher values indicating more variability. The x-axis represents the number of Monte Carlo simulations performed (iterations). Across all metrics and tree sizes, null model variability was reduced to $< \alpha/2$ (given the typical $\alpha = 0.05$) by 10^3 iterations.

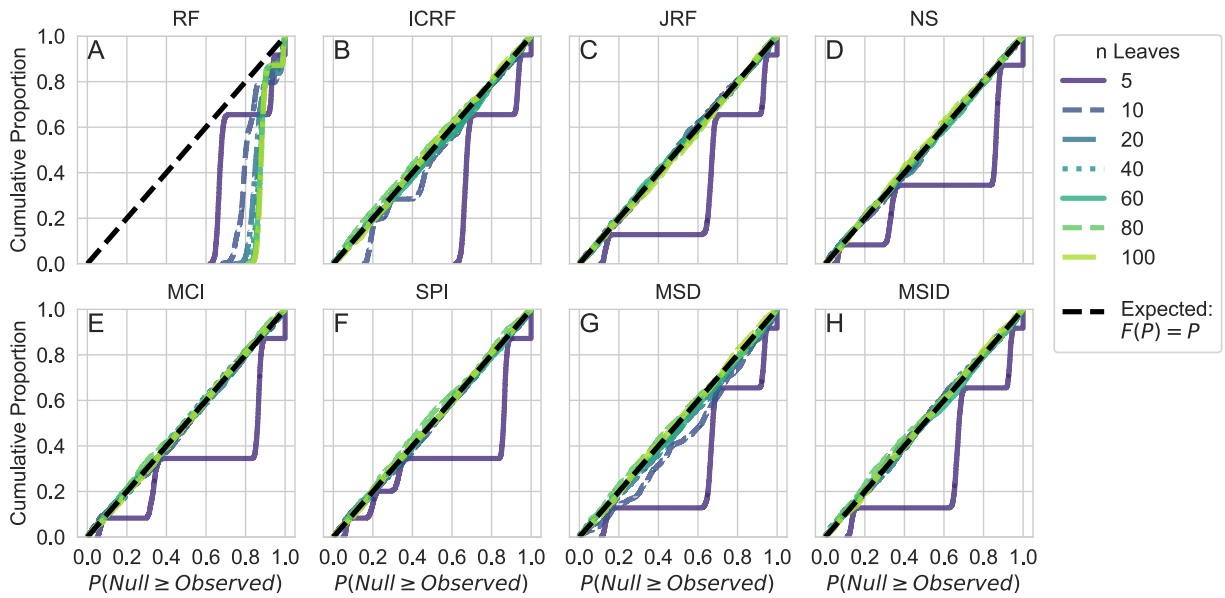

Figure 2. The effects of tree size and congruence metric on conformity to the expected error distribution. Each panel depicts the empirical cumulative density function of *P(Null ≥ Observed)* values computed for 1000 random pairs of trees. Here, the x-axis represents *P(Null ≥ Observed)* values, and the y-axis represents the cumulative proportion of values that are less than or equal to the corresponding x-axis value. The dashed black line in each panel depicts the expected (uniform) cumulative density function.

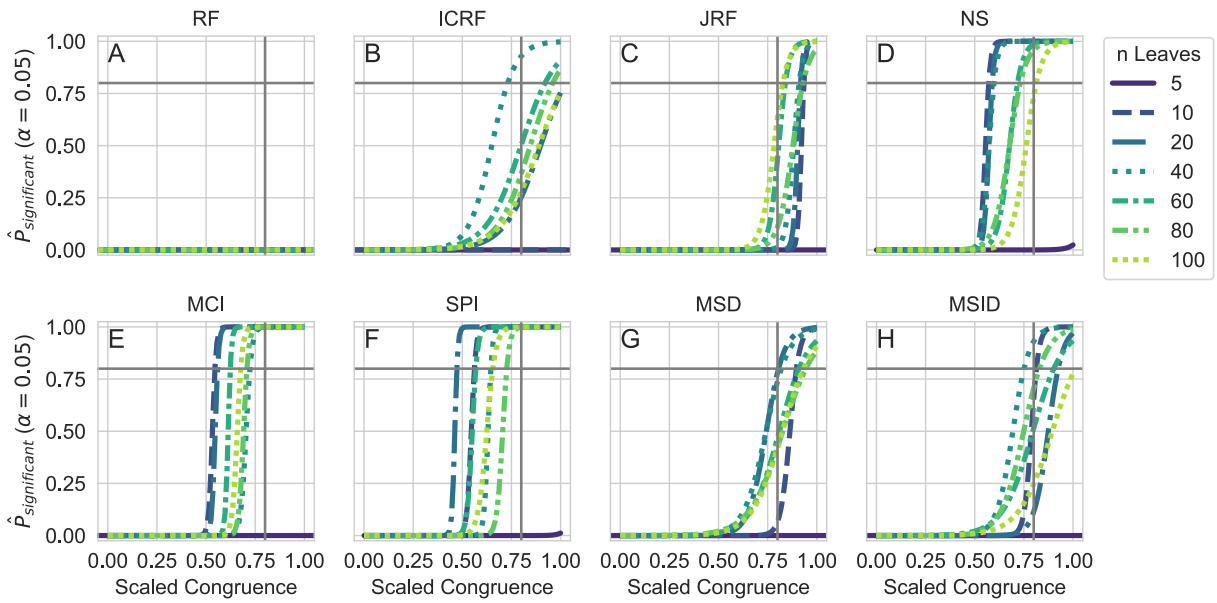

Figure 3. The effects of congruence metric and tree size on power. Each panel depicts the estimated proportion of significant tests (y-axis) as a function of scaled congruence (x-axis). Metrics that show a high proportion of significant results at lower scaled congruence values have greater power to detect more subtle congruence, while metrics that only reach a high proportion of significant results for the most highly congruent trees are more conservative. The darker gray lines in each panel indicate y=0.8 (high proportion significant) and x=0.8 (high congruence), and are meant serve as guides to help visualize the results.