Phylogenetics

mILD: a tool for constructing and analyzing matrices of pairwise phylogenetic character incongruence tests

Paul J. Planet\textsuperscript{1} and Indra Neil Sarkar\textsuperscript{1,2,*}

\textsuperscript{1}Division of Invertebrate Zoology and \textsuperscript{2}Division of Library Services, American Museum of Natural History, New York, NY, USA

Received on August 2, 2005; revised on October 7, 2005; accepted on October 24, 2005
Advance Access publication November 2, 2005

ABSTRACT

Summary: Pairwise comparisons of disagreement in phylogenetic datasets offer a powerful tool for isolating historical incongruence for closer analysis. Statistically significant phylogenetic character incongruence may reflect important differences in evolutionary history, such as horizontal gene transfer. Such testing can also be used to specify possible combinations of datasets for further phylogenetic analysis. The process of comparing multiple datasets can be very time consuming, and it is sometimes unclear how to combine data partitions given the observed patterns of incongruence. Here we present an application that automates the process of making pairwise comparisons between large numbers of phylogenetic datasets using the Incongruence Length Difference (ILD) test. The application also implements strategies for data combination based on the patterns of incongruence observed in pairwise comparisons.

Availability: The application is freely available as a Perl script that interacts with the command-line version of PAUP\textsuperscript{*}.

Contact: sarkar@amnh.org

Supplementary information: http://www.GenomeCurator.org/mILD/

1 INTRODUCTION

In phylogenetic analyses, disagreement (incongruence) among the datasets or even partitions of the same dataset can be explained by real differences in evolutionary history or by the effects of misleading phylogenetic signal (noise or homoplasy) which may overwhelm valid signal as a result of inadequate sampling, experimental error or the effects of separating data into partitions. Identifying significant incongruence is a crucial first step in further phylogenetic analysis. The process of comparing multiple datasets can be very time consuming, and it is sometimes unclear how to combine data partitions given the observed patterns of incongruence. Here we present an application that automates the process of making pairwise comparisons between large numbers of phylogenetic datasets using the Incongruence Length Difference (ILD) test. The application also implements strategies for data combination based on the patterns of incongruence observed in pairwise comparisons.

Availability: The application is freely available as a Perl script that interacts with the command-line version of PAUP\textsuperscript{*}.

Contact: sarkar@amnh.org

Supplementary information: http://www.GenomeCurator.org/mILD/

\begin{equation}
\delta_{ILD} = L_{c} - \sum_{i=1}^{n} L_{i},
\end{equation}

where \(L_{c}\) is the number of steps in the most parsimonious tree found when all datasets (partitions) are analyzed in a combined analysis, and \(L_{i}\) is the number of steps of the most parsimonious tree found for data partition \(i\) out of a total of \(n\) partitions. \(\delta_{ILD}\) is given a \(P\)-value by comparing it with a distribution of randomly generated \(\delta_{ILD}\) values that are calculated from partitions, equal in size to the original, generated by random resampling of characters (columns in a phylogenetic matrix) from all partitions.

The ILD test assesses if the degree of incongruence seen among datasets (partitions) is simply because of partitioning them for separate analysis by indicating when the conflict between datasets is not significantly greater than the conflict within each dataset.

When more than two datasets (partitions) exist, as is often the case in genome-scale datasets, the ILD can be computed as a global test of overall character incongruence amongst all partitions, but this calculation cannot identify specifically which partitions are incongruent. Incongruent partitions can be identified by sequentially testing each partition against a combination of all others (Baker et al., 1997; Brown et al., 2002; Escobar-Paramo et al., 2004). However, this technique may also fail to identify incongruence when validly discordant phylogenetic signal is distributed over more than one partition. A more attractive, but laborious, solution is to test each partition against every other partition with the goal of combining data based on the patterns of congruence in a matrix of all pairwise comparisons (Baker et al., 1997; Lecointre et al., 1998, 2005; Planet et al., 2003). The two major impediments to this procedure are (1) that the number of tests increases exponentially as partitions are added, making manual implementation tedious when datasets are large, and prohibitive for genome-scale datasets, and, (2) in practice, multiple pairwise comparisons often result in asymmetries that confound unambiguous combination of partitions (Baker et al., 1997; Planet et al., 2003). To be symmetrical, all partitions should be congruent with all other partitions in the combination. For example, if partition A is congruent with B, and B is congruent with C, but A is not congruent with C then the combination ABC is not symmetrical. It is difficult to choose between the overlapping combinations AB and BC.

Two solutions have been proposed to account for the patterns of incongruence in pairwise character incongruence matrices. Lecointre (Lecointre et al., 2005) suggested eliminating incongruence by deleting individual sequences that cause incongruence in each partition. This is accomplished by sequentially removing (jackknifing) taxa from each partition until partitions are found to be congruent. Offending sequences are then deleted from a final ‘careful’ simultaneous analysis of all partitions. We refer to this as the taxon jackknife strategy. A second solution is to perform multiple rounds of pairwise tests, choosing combinations of

© The Author 2005. Published by Oxford University Press. All rights reserved. For Permissions, please email: journals.permissions@oxfordjournals.org
symmetrically congruent partitions for inclusion as single partitions in the next round (Planet et al., 2003). This process, which we refer to as the ‘snowball’ technique, can be repeated until all partitions have been combined or are found to be incongruent.

With many partitions, a large amount of incongruence and a high degree of asymmetry, both the snowball and taxon jackknife techniques are time intensive. We present an application that automates the procedure of making and analyzing matrices of ILD comparisons (mILD). mILD takes individual dataset/partition files and performs all pairwise comparisons by interacting with the unix version of PAUP. Users can then choose to implement either the taxon jackknife or snowball strategies for data combination. The goal of this application is to make data combination tools widely available and tractable for large genome-scale analysis.

2 METHODS AND DESCRIPTION OF ALGORITHMS

Each dataset (partition) is input into mILD as a single file of aligned sequences in FASTA format. mILD then concatenates sequences in each file by matching taxon names, and outputs a combined nexus file in which each input file is represented as a partition. Interacting with the UNIX version of PAUP, mILD carries out all pairwise comparisons between partitions using PAUP’s version of the ILD test—the partition homogeneity test—and outputs a matrix of ILD P values for each comparison.

The user can choose between the snowball and taxon jackknife strategies to further test the combinability of partitions. The user can also select heuristic or exhaustive snowball strategies. The algorithm for the latter is as follows:

1. Based on the values in the incongruence matrix find all symmetrically congruent combinations of partitions.
2. Choose one of the symmetric combinations. Currently, mILD allows the user to select either a random combination or the combination with (a) the most characters (b) the most partitions.
3. The combination chosen in step 2 is then represented as a single partition in another round of all pairwise tests. The process then returns to step 1.

This process repeats until all partitions are either combined or incongruent. In the exhaustive snowball strategy, all symmetrically congruent combinations are included as single partitions (along with all the original partitions) in the next round of testing. The exhaustible snowball technique ends when no new combinations are identified.

The user can also choose the taxon jackknife technique. Based on the preliminary incongruence matrix, for each incongruent pair of taxa mILD performs multiple rounds of ILD testing, excluding each taxon in turn. Each taxon that results in a loss of incongruence during the ILD test is flagged and reported.

For both data combination strategies, the choice of P-value threshold for determining incongruence can be set by the user. In previous analyses, we have found a P-value of 0.01 to be a conservative but reliable estimate. In addition, the user can choose to apply a strict Bonferroni correction.

3 CONSIDERATIONS AND EXPANSIONS

Recent critiques of the ILD test suggest that it performs poorly when partitions differ substantially in level of noise, overall evolutionary rate, rate heterogeneity among sites, number of informative sites or size (Dolphin et al., 2000; Yoder et al., 2001; Barker et al., 2002; Darlu et al., 2002; Downow et al., 2002). However, the ILD remains a powerful test when incongruence is caused by differing evolutionary histories under certain conditions (i.e. large numbers of informative sites and low to intermediate among-site rate variation) (Darlu et al., 2002). Most studies have suggested that the ILD test is generally more susceptible to type I errors (false rejection of congruence), than type II errors (failure to reject congruence when datasets are incongruent) (Hipp et al., 2004). Thus, in the right circumstances, the ILD test can be a sensitive starting point for identifying potentially incongruent partitions for further analysis (Hipp et al., 2004). We suggest that results from mILD analyses should be carefully scrutinized and tested with other follow-up tests of incongruence. Although our implementation of mILD is designed to interact with PAUP, it is designed to be easily modified to use other phylogenetic applications. Future versions will include other incongruence metrics, and tests of the validity of ILD P values such as those proposed by Dolphin et al., (2000).

4 CONCLUSION

As genomic datasets become increasingly available, new tools are required that combine rigorous phylogenetic analysis with high-throughput, automated data curation. mILD allows large numbers of phylogenetic datasets (e.g. gene alignments) to be tested for phylogenetic congruence in a statistical framework, and then automates techniques for combining the data based on patterns of incongruence. We intend to expand this application to include other tests of incongruence and strategies for data combination.

ACKNOWLEDGEMENTS

We thank David H. Figurski, Rob DeSalle and E. Kurt Lienau for the helpful discussion. P.J.P. is supported by NIH grant 5R01 GM062351-02; I.N.S. by NSF grants IIS-0214229 & DBI-0421604 and the Lewis B. & Dorothy Cullman Program for Molecular Systematics.

Conflict of Interest: none declared.

REFERENCES