Syst. Biol. 54(1):146–157, 2005
Copyright © Society of Systematic Biologists
ISSN: 1063-5157 print / 1076-836X online
DOI: 10.1080/10635150590905984

Missing the Forest for the Trees: Phylogenetic Compression and Its Implications for Inferring Complex Evolutionary Histories

CÉCILE ANÉ[1,2] AND MICHAEL J. SANDERSON[1]

[1] *Section of Evolution and Ecology, University of California, Davis, California 95616, USA; E-mail: mjsanderson@ucdavis.edu (M.J.S.)*
[2] *Current address: Department of Statistics, University of Wisconsin–Madison, Medical Science Center, 1300 University Ave., Madison, WI 53 706, USA; E-mail: ane@cs.wisc.edu*

Abstract.— Phylogenetic tree reconstruction is difficult in the presence of lateral gene transfer and other processes generating conflicting signals. We develop a new approach to this problem using ideas borrowed from algorithmic information theory. It selects the hypothesis that simultaneously minimizes the descriptive complexity of the tree(s) plus the data when encoded using those tree(s). In practice this is the hypothesis that can compress the data the most. We show not only that phylogenetic compression is an efficient method for encoding most phylogenetic data sets and is more efficient than compression schemes designed for single sequences, but also that it provides a clear information theoretic rule for determining when a collection of conflicting trees is a better explanation of the data than a single tree. By casting the parsimony problem in this more general framework, we also conclude that the so-called total-evidence tree—the tree constructed from all the data simultaneously—is not always the most economical explanation of the data. [Compression; information; Kolmogorov complexity; phylogenetics; total evidence.]

Recombination, lateral gene transfer, hybridization, and other biological processes generate conflict between phylogenetic trees constructed from different loci or different partitions of a sequence data set. Lateral gene transfer in bacteria provides many examples (Kurland et al., 2003; Lerat et al., 2003), and transfers between mitochondrial genomes of distantly related plants are now well documented (Bergthorsson et al., 2003). Hybridization and introgression is another source of conflict, as suggested by numerous disagreements between trees based on nuclear and organellar genes (Cronn and Wendel, 2003; Doyle et al., 2003). Conflicts can also emerge when different partitions have the same phylogenetic history but very different patterns of molecular evolution, causing biased inferences in one or more partitions (Rokas et al., 2003). This can arise, for example, if one partition is subject to long-branch attraction but another is not (e.g., Sanderson et al., 2000). No conceptual issue has generated more discussion in phylogenetics in the last decade than the treatment of these conflicts (Bull et al., 1993; Cunningham, 1997a, 1997b; de Queiroz et al., 1995; Farris et al., 1995; Huelsenbeck et al., 1996; Thornton and DeSalle, 2000).

The problem can be reduced to deciding when a collection of trees—a "forest"—is a better explanation for evolutionary relationships among a set of sequences than is a single tree. In this article we present a new framework for addressing this issue based on algorithmic information theory (Li and Vitanyi, 1997). An important tool in that field is an elegant formulation of parsimony as a form of data compression, a tool general enough to apply equally well to forests and to a single tree. This perspective places the question of "forest versus tree" within a unified inferential setting. It also provides a deterministic decision rule for choosing between these two hypotheses that does not depend on randomization tests such as the widely used incongruence length difference (ILD) test (Farris et al., 1995). Finally, it settles the long-standing phylogenetic controversy over whether the maximum parsimony tree based on the entire data set, the "total evidence" (Kluge, 1989), is always preferable to separate analyses of subsets of the data.

Maximum parsimony (MP) finds the tree for a given data matrix for which the sum across characters of the number of evolutionary changes required on that tree is minimized. Initially introduced as a heuristic approximation to likelihood methods (Edwards and Cavalli-Sforza, 1964), MP eventually would become the method of choice for phylogeneticists, until the more recent ascendancy of model-based inference methods beginning in the early 1990s (Felsenstein, 2001). In a widely cited paper, Farris (1979) proposed two rationales for constructing the MP tree: it is the most economical evolutionary explanation of the data, and it is the most efficient summary of its information content—the summary that requires literally the fewest symbols. Later workers (Kluge, 1989; Kluge and Wolf, 1993; Nixon and Carpenter, 1996) took up this view, arguing that subdividing data sets and constructing separate phylogenetic trees is inappropriate, because trees from partitions of any data set will tend to be suboptimal by the parsimony criterion relative to the entire data set. This spawned the so-called total evidence view of the problem (Kitching et al., 1998), which argues that "pooling the data and determining the most parsimonious solution for all the data . . . maximizes information content" (Nixon and Carpenter, 1996). Here we show that this statement cannot be universally true in complex data sets with conflicting signals.

Algorithmic information theory provides a very general framework for assessing the information content of hypotheses. It rests on the concept of Kolmogorov complexity, defined as the length of the shortest computer program that faithfully describes an object (Li and Vitanyi, 1997). A complex object requires a long program; a simple object with much regularity does not. A long random string of digits is complex because it cannot be coded as a computer program any shorter than the trivial program that just prints out all its digits it cannot be compressed. The irrational number, π, has an infinite number of digits but is not very complex, because a short computer program can be written that calculates this long string to any desired accuracy it can be compressed dramatically.

This idea can be extended to hypothesis selection via the minimum description length principle (Li and Vitanyi, 1997). The best hypothesis is the one that minimizes the length of the description of the hypothesis plus the length of the description of the data when encoded or compressed with the help of that hypothesis. Scientific inference based on some version of this idea has been applied to many biological problems, including sequence alignment (Allison et al., 1999, 2000; Allison and Yee, 1990), phylogeny reconstruction (Cheeseman and Kanefsky, 1993; Li et al., 2001; Milosavljevic et al., 1990; Otu and Sayood, 2003; Ren et al., 1995), and comparison of classifications (Day, 1983), though not to collections of trees constructed from separate data sets. In this framework, a hypothesis is either a tree or a collection of trees and the best hypothesis is the one that permits the maximum compression of that hypothesis plus the sequence alignment.

Materials and Methods

Preliminaries

Assume the data set, D, consists of n nucleotide sequences on the state set $\{a, c, g, t\}$, each sequence consisting of m sites. These sequences may be associated with a tree, T, with n leaves. A parsimony score, $L = L(T, D)$, for the data set with respect to tree T can be determined by standard algorithms (Semple and Steel, 2003). Write $\lg(x)$ for $\lceil \log_2(x) \rceil$, where $\lceil x \rceil$ is the smallest integer larger than x. We frequently need an efficient coding scheme for integers of arbitrary size. The size in bits of an encoding of integer, k, will be denoted des(k) (see Appendix A).

The raw unencoded form of these sequences requires 2 bits per nucleotide or a total of $2nm$ bits. This can be reduced using a consensus sequence and encoding the remaining sequences as differences from this consensus—essentially using a "star" phylogeny. With no missing data this coding length is $4m + L(2 + \lg(n))$, where $L = L(T^*, D)$, and T^* is the star phylogeny. This description length is shorter than the unencoded data whenever

$$\frac{L}{m} \leq \frac{2n-4}{\lg(n)+2} \sim \frac{2n}{\lg(n)}$$

Note, however, that L will tend to be quite large when there is phylogenetic signal in the data. Characters that evolve only once on the true tree will be regarded as parallelisms on T^*, adding length to L, making this star-tree encoding relatively inefficient.

Compression Using a Most-Parsimonious Tree

For phylogenetically related sequences, the information contained at any site in an alignment is partly redundant, because changes in sequence occur only occasionally in its evolutionary history. Many lineages simply retain a conserved site. Presumably, this observation motivated Farris's (1979, 1980) claim that the parsimony tree permits the most efficient summary of the data possible. He sketched the outlines of a coding scheme but did not develop it in sufficient detail to derive an expression for its description length. Details of our coding schemes are described in Appendices A to C; here we summarize the results.

The data are compressed with respect to a binary tree, T. If necessary, any tree can be made binary ("resolved") by adding zero-length branches (highly unresolved trees, not surprisingly, will be compressed better using the previous "star-phylogeny" scheme). The size in bits of the compressed file based on T has two parts. The first part is the number of bits required to encode T, which is $2n - 4$ for the shape, plus $n \lg n$ for the ordering of the taxon labels. The second part is the number of bits required to encode the matrix relative to T, which is accomplished by first encoding the sequence of one of the taxa, which takes $2m$ bits, then encoding the branch where each of L substitutions occur, and finally adding delimiters and an end of file signal:

$$4m + (2 + \lg(2n-3))L + \lg(2n-3)$$

The total file size is thus

$$2n - 4 + n \lg n + 4m + (2 + \lg(2n-3))L + \lg(2n-3) \tag{1}$$

This result is valid for any tree, but the size is smallest when T is the MP tree, because that tree minimizes L. Put another way, this proves that among all binary trees representing relationships among these taxa, the MP tree provides the most efficient (shortest) encoding of the data using the coding method outlined in Appendix A. We note that finding the MP tree exactly can require exponential running times. Polynomial time heuristics such as those available in standard tree reconstruction packages may terminate with suboptimal solutions, and hence compression efficiency will often be improved in direct proportion to the quality of the MP search strategy.

Figure 1 shows the length of codes constructed using Equation 1 as a function of the number of taxa and the amount of homoplasy in the data. Rates of evolution are monotonically increasing with L, and the amount of homoplasy is roughly proportional to L/m (L/m is related to the reciprocal of the "consistency index"). Compression is much more effective for large trees than small ones and for lower rates of evolution (lower homoplasy). In fact, it is more efficient to use phylogenetic compression than to keep the data uncompressed when

$$\frac{L}{m} \leq \frac{2n}{\log_2(n)}, \text{ when } m \text{ and } n \text{ are large}$$

Note that the ratio

$$\frac{L}{m} \Big/ \frac{n}{\log_2(n)}$$

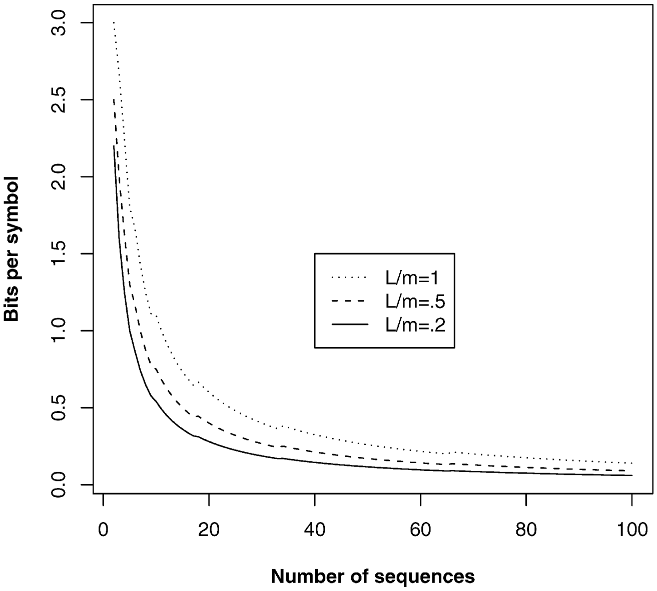

FIGURE 1. Phylogenetic compression rate based on Equation 1. The quantity L/m is the ratio of the inferred most parsimonious number of changes on the tree to the number of sites.

is close to the number of bits used per nucleotide. Only if homoplasy is very high and/or the trees are small is it worse to compress the data using the MP tree than to leave it uncompressed, due to the "overhead" of describing the tree.

Compression Using a Forest

A data set with phylogenetic signal can be compressed using the most parsimonious tree. However, it can sometimes be compressed even further by reference to a collection of trees rather than a single tree. Consider a partition of the sequence alignment, D, into l subsets of characters, the ith labeled D_i, each with m_i characters. This corresponds to a forest, $\mathcal{F}$, of l binary trees, the ith labeled T_i. Define the "total evidence" length as $L^{(TE)} = L(T, D)$, where T is the tree constructed from the entire data set, D, and the length of the forest as $L^{\text{forest}} = \sum_i L(T_i, D_i)$. Let the incongruence between the total evidence tree and the forest of individual trees be $\Delta L = L^{(TE)} - L^{\text{forest}}$. This "incongruence length" is the number of extra evolutionary steps required to fit the individual data sets in the partition to the overall total evidence tree (Farris et al., 1995).

If each of the trees in $\mathcal{F}$ is "sufficiently" different from one another, the best strategy is to separately compress each data subset with respect to its own MP tree. This can be achieved with a code of size

$$\begin{aligned} &\text{des}(l) + l(2n - 4 + n\,\text{lg}(n)) + l\,\text{lg}(2n-3) + 4m \\ &\quad + (2 + \text{lg}(2n-3))L^{\text{forest}} \qquad (2) \end{aligned}$$

This length is also shorter than uncompressed data under the same conditions as outlined for Equation 1 above.

The more interesting question is when is Equation 2 shorter than Equation 1; that is, when is the forest compression scheme better than the total evidence compression scheme? The answer is surprisingly simple, at least when n is large enough. Whenever

$$\Delta L \geq (l-1)n,$$

it is more efficient to use the forest to encode the data. The conventional method for assessing whether ΔL is large enough is to compare it to a null distribution generated by randomizing the data matrix (the ILD test; Farris et al., 1994, 1995). Here the cutoff is determined by the information content directly, with no reference to a hypothetical null distribution. The exact cutoff for any n is given in Appendix B.

If the trees are similar to one another it can be more efficient still to code each tree as a small number of rearrangements from one of the trees. For example, we choose a reference tree, say tree one, and calculate a sequence of intermediate trees that can be obtained by nearest-neighbor interchange (NNI) operations (Li et al., 1996). Each rearrangement can be described by specifying the branch around which the interchange occurs and which of the two possible rearrangements is chosen (see Appendix A). Let k_i be the NNI distance between tree T_i and T_1 and let the mean NNI distance be $\bar{k} = \frac{1}{l-1}\sum_{i=2}^{l} k_i$. Compression can now be achieved with length

$$\begin{aligned} &\text{des}(l) + 2n - 4 + n\,\text{lg}(n) + \bar{k}(l-1)(\text{lg}(n-3) + 1) \\ &\quad + \text{des}(k_2) + \cdots + \text{des}(k_l) + l\,\text{lg}(2n-3) \\ &\quad + 4m + (2 + \text{lg}(2n-3))L^{\text{forest}} \qquad (3) \end{aligned}$$

This is shorter than the single-tree description length in Equation 1 when

$$\Delta L \geq (l-1)\bar{k}$$

approximately, for large n. Appendix B gives the exact expression.

Choosing between the Forest and the Tree

The forest compression scheme is preferred when

$$\begin{aligned} \Delta L &\geq (l-1)n, \text{ or} \\ &\geq (l-1)\bar{k} \qquad (4) \end{aligned}$$

approximately for large n. The entity inferred when Equation 4 is satisfied is an entire *forest*, $\mathcal{F}$. It should not be regarded as a set of equally good *alternative* (single) explanations. In fact, by the conventional measure of tree quality in that latter context, the maximum parsimony score, the trees in $\mathcal{F}$ may score unequally with respect to the entire data set. Even more surprisingly, the

$$
\begin{array}{l} \\ a \\ b \\ c \\ d \end{array}
\underbrace{\begin{array}{ccc} \overbrace{\hphantom{\text{AAAAAAA}}}^{\text{7 times}} & \overbrace{\hphantom{\text{AAAA}}}^{\text{4 times}} & \overbrace{\hphantom{\text{AA}\ldots}}^{r\text{ times}} \\ \text{AAAAAAA} & \text{AAAA} & \text{AA}\ldots \\ \text{AAAAAAA} & \text{CCCC} & \text{AA}\ldots \\ \text{CCCCCCC} & \text{CCCC} & \text{AA}\ldots \\ \text{CCCCCCC} & \text{AAAA} & \text{AA}\ldots \end{array}}_{\text{submatrix } D_1}
\underbrace{\begin{array}{ccc} \overbrace{\hphantom{\text{AAAAAAA}}}^{\text{7 times}} & \overbrace{\hphantom{\text{AAAA}}}^{\text{4 times}} & \overbrace{\hphantom{\text{AA}\ldots}}^{r\text{ times}} \\ \text{AAAAAAA} & \text{AAAA} & \text{AA}\ldots \\ \text{CCCCCCC} & \text{CCCC} & \text{AA}\ldots \\ \text{AAAAAAA} & \text{CCCC} & \text{AA}\ldots \\ \text{CCCCCCC} & \text{AAAA} & \text{AA}\ldots \end{array}}_{\text{submatrix } D_2}
$$

FIGURE 2. Example discussed in text in which it is more efficient to code a data matrix with a forest of two trees than the single MP tree from the combined matrix.

total evidence tree T can have a better score than any of the trees in $\mathcal{F}$; nonetheless, $\mathcal{F}$ is preferred.

Figure 2 is a contrived data set with some of these surprising properties. It has two partitions, D_1 and D_2, each of which includes four informative sites favoring one tree, seven informative sites favoring a different tree, and r invariant sites. However, the trees favored in the two partitions are different, and the total evidence tree is different from either of those (see Page, 1996, for a similar example and a useful visualization). Let T be the total evidence tree, $(ad)(bc)$, T_1 be the MP tree favored by partition D_1, $(ab)(cd)$, and T_2 be the MP tree favored by D_2, $(ac)(bd)$.

The parsimony scores for the partitions are $L(T_1, D_1) = L(T_2, D_2) = 15$, which are each better than the score for the total evidence tree on either of these partitions, $L(T, D_1) = L(T, D_2) = 18$, and are also better than the scores of these trees on the opposite partition: $L(T_1, D_2) = L(T_2, D_1) = 22$. This confirms that the MP tree for each partition is indeed better than any other tree for that partition.

In addition, the score of trees T_1 and T_2 with respect to the whole matrix, D, is worse than the score of the total evidence tree with respect to D: $L(T_1, D) = L(T_2, D) = 37$, whereas $L(T, D) = 36$.

Now, the extra steps entailed by the incongruence, $\Delta L = L(T, D) - [L(T_1, D_1) + L(T_2, D_2)] = 36 - (15 + 15) = 6$, satisfy Equation 4 above, and the forest consisting of T_1 and T_2 is preferred over the total evidence tree T in our compression scheme. This is a striking result: a solution composed entirely of "suboptimal" trees—according to the conventional parsimony criterion—is preferred over the more parsimonious total evidence tree. The conclusion also holds using the exact cutoff values provided in Appendix B.

DATA SETS

Compression Efficiency in a Large Sample of Phylogenetic Data Sets

We compared compression efficiency in 638 nucleotide sequence data sets for protein coding genes in green plants. Details of how these data were obtained from GenBank are described elsewhere (Sanderson et al., 2003: data available at http://ginger.ucdavis.edu). Data sets ranged in size from 4 to 1079 taxa and 72 to 3375 characters. At one extreme a few data sets contain identical or nearly identical sequences; at the other extreme data sets contain sequences that are diverged by up to 40% at the amino acid level from other sequences in that data set.

Available DNA sequence compression programs rarely compress single sequences to less than about 1.6 bits per character (Chen et al., 2002). For comparative purposes, we examined the compression efficiency of a standard DNA sequence compression program, GenCompress (Chen et al., 2002), on the same data sets we compressed with our phylogenetic method. GenCompress uses a type of Lempel-Ziv coding (Cover and Thomas, 1991), tailored to DNA sequence data sets, and looks for approximate substring matches in the file. Both compression schemes are increasingly effective as the number of taxa increases. Both take advantage of the evolutionary redundancy of the similar sequences in phylogenetic data sets (Fig. 3A). With phylogenetic compression, data sets with 100 taxa can typically be compressed 90% to 95%. Large data sets were almost always more efficiently compressed via phylogenetic

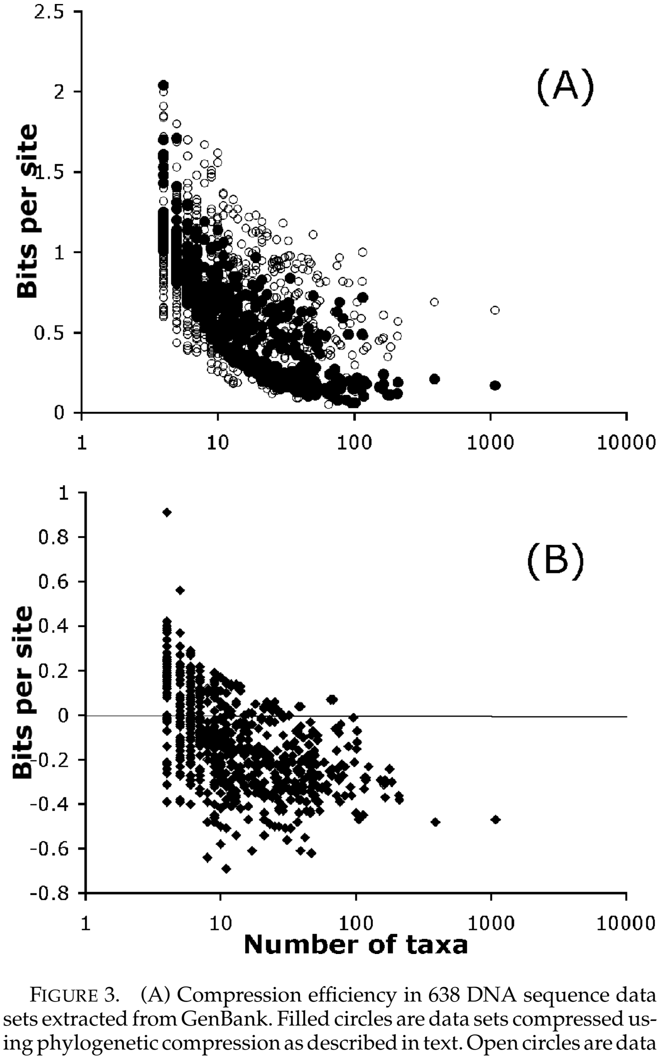

FIGURE 3. (A) Compression efficiency in 638 DNA sequence data sets extracted from GenBank. Filled circles are data sets compressed using phylogenetic compression as described in text. Open circles are data sets compressed using GenCompress (nonphylogenetic) compression. (B) Relative efficiency of these compression procedures. Vertical axis is the compression level obtained for phylogenetic compression minus compression level for GenCompress for a given data set. Values below 0 indicate better compression efficiency for phylogenetic compression.

compression than GenCompress (Fig. 3B). The most compressible data set, consisting of 102 taxa, 2155 sites, and a parsimony length of 278 could be compressed to 0.06 bits/site and generated a file less than half the size of the corresponding GenCompress file. As expected, the poorest performance occurred in small trees with high homoplasy. The worst was a data set of four taxa, 960 sites, and a tree length of 799 steps, which is an extraordinarily high level of homoplasy (nearly one change per site). Phylogenetic compression actually produced a file slightly larger than 2 bits/site for this data set, whereas GenCompress compressed these data twice as much. Although compression methods aimed at single DNA sequences usually do not achieve lower than 1.6 bits/site on single sequences, they perform better than this on data sets with collections of sequences with phylogenetic structure to them. Nonetheless, explicit phylogenetic compression is usually more efficient. It might be expected that GenCompress would perform better on phylogenetic data matrices that are transposed. Transposing rows and columns would, for example, turn constant characters (columns) into rows of constant blocks of identical letters. However, at least in all but the largest data sets (where the program simply would not terminate) most of the data sets actually were compressed *less* by GenCompress when transposed.

A Case Study of Conflicting Signals

Sanderson et al. (2000) analyzed phylogenetic relationships in 19 land plant species for two plastid genes, *psaA* and *psbB*, using parsimony and maximum likelihood (see also Magallón and Sanderson, 2002). An ILD test (Farris et al., 1995) for data set heterogeneity suggested congruent signals between genes but not between codon positions. Parsimony analysis of the two genes separately each yielded one most parsimonious tree, the "*psaA*-tree" and "*psbB*-tree," respectively, which are only slightly different from each other. Third codon positions from both genes combined yielded one most parsimonious tree (the "3-tree"), whereas first and second positions from both genes combined yielded four equally parsimonious alternative MP trees (the "12-tree"), all quite different from the 3-tree. The total evidence (TE) tree was the same as the 3-tree. Figure 4 shows a schematic of the various trees resulting from these analyses with their pairwise NNI distances. These were calculated using the program COMPONENT (Page, 1993). Recent discussions of phylogenetic relationships in seed plants have focused on the dispute between these two basic trees, which disagree strongly about (among other things) the position of the Gnetales, an enigmatic clade of seed plants whose relationships have long been the subject of controversy (Donoghue and Doyle, 2000). Rydin et al. (2002) reiterated older findings that the TE tree and 3-tree, which support the placement of Gnetales as the sister group of the remaining seed plants, is the best estimate of seed plant phylogeny. Other workers argue that the 12-tree is closer to the truth, basing this in part on extensive use of maximum likelihood methods (Aris-Brosou, 2003; Magallon and Sanderson, 2002; Sanderson et al., 2000), which generally favor that result. High rates of substitution in the 3rd position data make long-branch attraction a possible explanation for these differences (Sanderson et al., 2000).

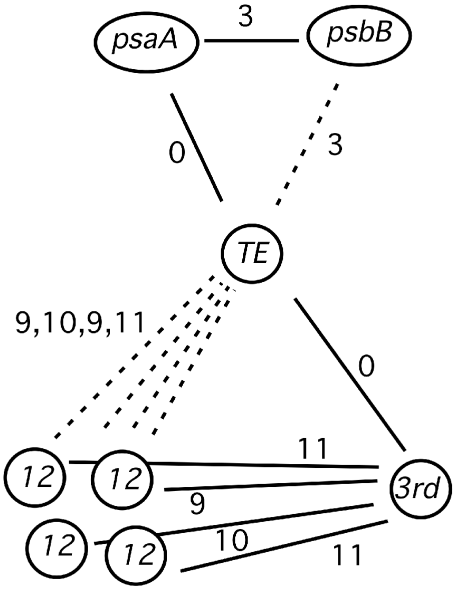

FIGURE 4. Schematic diagram showing trees resulting from parsimony analyses of combined total evidence data described in Table 1. Trees from the total evidence data set and the two different partitioning schemes are shown. One partition divides the data set into two genes (*psaA* versus *psbB*); the other divides it into two classes of codon positions (first and second [12] versus third [3]). Integers on internodes indicate the NNI distance between trees found in parsimony analyses (Li et al., 1996).

We excluded any sites having missing or ambiguous data, because the compression scheme does not allow more than four character states (but see Appendix C). The data compressed with respect to the total evidence tree required 50,083 bits (see Table 1). The data compressed with respect to the two-gene partition is larger than this: 50,197 bits when the trees are coded separately, 50,086

TABLE 1. Phylogenetic compression analysis of photosystem gene data sets. All data sets have 19 taxa. Missing or ambiguous sites were removed from alignments. Number of nucleotides in combined matrix is 66,291 (number of bits = 132,582).

Data partition/coding scheme	Number of characters	Length of MP tree or forest	Length of compressed file in bits
Total evidence	3489	4499	50083
12 partition	2325	885	16515
3 partition	1164	3581	33439
12 + 3 forest (separate tree coding)	3489	4466	49957
12 + 3 forest (NNI coding)	3489	4466 ($\Delta L = 33$)	49880
psaA partition	2130	2751	30663
psbB partition	1359	1745	19531
psaA+psbB forest (separate tree coding)	3489	4496	50197
psaA+psbB forest (NNI coding)	3489	4496 ($\Delta L = 3$)	50086

bits when they are coded using NNI distances. However, the data compressed with respect to the codon partition trees is smaller than the total evidence case: 49,957 bits using separate trees or 49,880 bits with NNI compression. Thus, compression is improved by using the codon partition, but degraded by using the gene partition. These results are consistent with the approximate predictions of Equation 4. For the total evidence tree to be favored, the incongruence length difference, ΔL, should be less than (approximately) the smaller of n, the number of taxa, 19 (or 17.25 exactly), or k, the NNI distance between the two trees (3 for the gene partition, 9 for the codon partition, approximately, or 3.375 and 7.625, respectively, for the exact cutoff). For the gene partition, $\Delta L = 3$, and the total evidence hypothesis is favored, but for the codon partition $\Delta L = 33$, so the forest is favored. These results agree with the conventional ILD test of homogeneity, which says that the two genes are not significantly different ($P = 0.77$) but the two codon partitions are ($P = 0.01$).

The "best" representation of the data is therefore a forest of two trees derived from the separate codon positions. The two trees do not have the same MP score with respect to the entire data matrix. Indeed, the 3-tree is more parsimonious at 4499 steps compared to the 12-tree, which has 4611 steps—112 steps longer. Nonetheless, the most economical explanation of *all the data* is a joint hypothesis that retains both trees.

GenCompress (also with ambiguous and missing sites removed) only compressed the alignment to 90,128 bits, almost twice as large as the phylogenetically compressed version.

Discussion

Identifying Conflicting Phylogenetic Signals or Non-Treelike History

Discovery of conflict between phylogenetic data sets provides some of the most compelling evidence for conflicting evolutionary histories. This conflict may be a difference in the branching relationships of the subsets of the data involved, or a difference in the molecular substitution processes tracking the same phylogeny, or both. Application of algorithmic information theory provides a new tool to diagnose these conflicts. Roughly speaking, whenever $\Delta L > \min(n, \bar{k})$ for two data sets, the data support the added complexity entailed by a hypothesis that there is a conflicting history for the sequences.

In the example of the plastid photosystem data, it is unlikely that the true phylogenetic tree for the first two codon positions is different from the tree for the third codon positions. No biological mechanism is known that would allow for such an outcome, yet our compression scheme leads unambiguously to an inference of heterogeneity. This is an example in which the substitution process is clearly heterogeneous but biology suggests that the tree per se is not. We interpret this to imply that tree inference methods would do well to consider different models for the different partitions. Such heterogeneous models might well then lead to an inference of a single tree, as was found when maximum likelihood was applied to the two codon partitions separately (Magallón and Sanderson, 2002). There are parallel generalizations of parsimony methods that allow different "models" (weighting schemes or methods that count site patterns differently than conventional parsimony methods; Willson, 1999).

Relationship to Other Tests for Conflict

Information theory provides a solution to one conundrum of the conventional parsimony approach to incongruence between data sets. Two data sets are incongruent whenever $\Delta L > 0$, which is almost always the case for any two subsets of a data set. Nonetheless, no one has seriously argued that data sets should be regarded as conflicting any time $\Delta L > 0$. Instead, the ILD randomization test is often used to establish a null distribution on ΔL, and incongruence is accepted only when ΔL is far enough out on the tail of this null distribution. The relative merits of this test have been scrutinized extensively (Barker and Lutzoni, 2002; Darlu and Lecointre, 2002; Dolphin et al., 2000; Hipp et al., 2004; Yoder et al., 2001). Our equation (4) is an alternative condition. It, too, establishes a cutoff value for ΔL, but it does not rely on null models. Instead it suggests that the natural penalty for added complexity of a hypothesis of two trees should be the cost of describing that additional tree. Formal measures of complexity based on compression allow description of the data to be placed on the same footing as descriptions of the tree, thus making it possible to evaluate the overall simplicity of a hypothesis involving multiple evolutionary histories. We have undertaken a small simulation study of the four-taxon case, which indicates that the level and power of the ILD test and our compression test are quite similar when using the separate trees encoding and for large data sets for the NNI encoding (results not shown). However, the ILD test is more conservative than the compression test, which is somewhat too liberal, for small data sets.

The computational advantages of the compression approach may be important with the increasing size of concatenated data matrices drawn from whole genome analyses (Lerat et al., 2003; Rokas et al., 2003) or broad surveys (Bapteste et al., 2002; Murphy et al., 2001). Generalizations of the ILD test or statistical tests like it are not obvious, although a promising Bayesian approach has recently been reported (Vogl et al., 2003). The calculations outlined in Equations 1 to 4 require the construction of parsimony trees, but these need to be done only once per partition, whereas randomization tests require large numbers of replicate searches. This means many combinations of subsets might be examined in the time it takes one typical ILD run.

Implications for the "Total Evidence" School of Inference

An intense debate has surrounded the argument that the maximum parsimony tree is the most faithful reflection of any data set, as opposed to several trees entailed by subsets of the data (Nixon and Carpenter, 1996).

Critics of this view have largely assailed it on biological grounds: that disparate evolutionary processes ("process partitions") sometimes underlie a single data set, and combination of these signals into a "total evidence" analysis will thus obscure their distinct histories (Bull et al., 1993). Our argument against the total evidence view is more direct. When simplicity, complexity, and information are cast in a sufficiently general framework, the most economical hypothesis associated with a given data set may well entail multiple conflicting trees, any one of which may be "suboptimal" by the conventional parsimony criterion. In our view, this defeats the fairly abstract but nonetheless compelling argument that the maximum parsimony tree is always the vehicle providing the most efficient summary of a data matrix.

Relationship to Work on Recombination

Hein (1990, 1993) comes close in spirit to the present work in using a parsimony criterion to identify a collection of trees implied by an alignment when recombination occurs between sequences. His method respects the order of sites in a sequence and chooses a tree for every site by minimizing an optimality criterion having two elements: the parsimony score of possible trees at that site and the rearrangement distance between trees at neighboring sites. This penalizes both substitutions and recombination events that are either too numerous among sites or too complex between any two neighboring sites. Different weights can be assigned to substitution events versus recombination events. Hein's method imposes additional structure on the problem, which is appropriate for the special case of recombination, but this is not generally required by our approach and could be removed from his. Perhaps more importantly, by casting our inference problem in terms of data compression and Kolmogorov complexity, we "remove" the issue of relative weights between different kinds of events, which naturally arises both in Hein's approach and other gene-tree-parsimony type methods (Page and Charleston, 1997). Some might argue that this merely gives equal weight to substitution and recombination in that problem, but it is worth noting that the compression approach does not stipulate that the source of data set conflict need be recombination or anything else.

Complexity of a Phylogenetic Data Set and its Evolutionary Implications

A description of the information content or complexity of a phylogenetic data set by the length of its most efficient encoding provides a novel kind of summary of the evolutionary history of a clade. It combines information about sequence divergence with speciation and extinction patterns. For example, a low complexity, highly compressible data set can arise when rates of sequence evolution are very slow (sequences are highly conserved), when most speciation events in the tree are very recent (high redundancy), and/or when extinction has selectively removed large clades, rather than removed lineages at random across the tree. A high complexity, minimally compressible data set arises when rates of evolution are high and/or the tree is nearly starlike.

Oddly enough, low complexity is ideal for tree-based prediction. One of the triumphs of phylogenetic biology has been the realization that a phylogenetic tree permits prediction about character states not yet observed in taxa whose relationships are known. This works best in areas of the tree with high redundancy or low information content. In other words, high character state diversity (high complexity) is not ideal for prediction. This idea finds formal support in Fano's inequality (Cover and Thomas, 1991), which provides bounds on the error associated with prediction in the presence of different levels of information. Although this is derived using probabilistic notions of entropy, the idea should hold for algorithmic measures of information content.

Entropy

If sites in a sequence were independent and identically distributed (i.i.d.), then the optimal compression rate of an infinite sequence would be the entropy of a random nucleotide ("nucleotide entropy"). This is computed from the base frequencies and is exactly 2 bits/nucleotide when base frequencies are equal. The optimal compression rate for real DNA sequences is actually less than the nucleotide entropy because sites are not i.i.d. Existing compression programs take advantage of repeated patterns to compress close to or below the entropy. For example, in the sequence data from the combined photosystem data set, the nucleotide entropy is 1.36 bits/nucleotide, and the GenCompress program achieves almost exactly this compression rate. Phylogenetic compression improves on this considerably, however, to about 0.76 bits/nucleotide. This is possible because our compression scheme takes advantage of the fact that a sequence alignment is *not* i.i.d. An alignment can be regarded as a set of nucleotide "patterns" (or "characters"), each of which corresponds to the combination of nucleotides at a given site for all taxa. The ideal compression rate of an infinite sequence of i.i.d. *patterns* is the entropy of a random pattern ("pattern entropy"). Our compression rate, in terms of bits per pattern, aims at being close to this pattern entropy. As the sequences are phylogenetically dependent, this entropy is much below $2n$, yielding an optimal compression rate much below $2n$ bits/pattern; i.e., 2 bits/nucleotide. The pattern entropy for the photosystem data implies a lower bound on compression of about 0.20 bits/nucleotide. Our compression comes closer than others to this lower bound, but still could be improved. One way to do this would be to compress the reference sequence used in our compression scheme (Appendix A). This would then account for dependence within a single sequence as well as across the tree.

Compression Optimality and Robustness of Model Selection

Choosing between a forest and a tree is a problem in model selection (Burnham and Anderson, 1998). Our model selection scheme is based on the idea of compression. For any given data set, it is possible that a different compression scheme might lead to a different choice of models. If different schemes favor different models, it would make sense to favor the model associated with the best compression scheme.

In an ideal setting, we would like to have an optimal compression scheme, having the lowest rate of all compression schemes on all data sets. If characters are i.i.d., then it is known that the pattern entropy is a lower bound on the compression rate of any (instantaneous or prefix-free) code (Cover and Thomas, 1991). Moreover, this lower bound can be achieved asymptotically with schemes called Huffman codes, for example (Cover and Thomas, 1991). For our code, we can show that its asymptotic compression rate is

$$4 + (2 + \lg(2n - 3))E\{L\}$$

where $E\{L\}$ is the average parsimony score of a random site pattern on the given tree. It can also be shown that this quantity is larger than the pattern entropy. Our code is thus not optimal, at least with respect to infinitely long sequences of i.i.d. patterns. A Huffman code would do better on very long sequences, even though it would require a heavy overhead to define a translation table at the beginning of the file. This is further evidence that the "total evidence" tree is not the most economical explanation of the data, when they are made of very long sequences.

On the other hand, we always deal with finite sequences, and the overhead due to the tree or the translation table definition is important. Our code may well perform better than a Huffman code on most data sets of a fixed size. No code has optimal performance on all sequences of all size (Li and Vitanyi, 1997), and no code has optimal performance (on average) on sequences drawn from any distribution and of any size. Making the assumption that the distribution of the data has some relationship to a tree, we built a code aimed at phylogenetically structured data.

Optimality of the code is a desirable property for the purposes of model selection. However, it is important to achieve a balance between the tree coding (overhead due to the model) and the matrix coding (information remaining in the data once the model is known). An excellent tree coding combined with a poor matrix coding will tend to favor complex models, whereas a poor tree coding combined with an excellent matrix coding will favor too simple models. For the purpose of model selection, the balance in model/data coding may be a more desirable property than overall optimality.

Extensions and Generalizations

The methods described here can be extended in several directions. Appendix C describes procedures for including missing data (as well as alignment gaps) and increasing the alphabet size to handle amino acid data or other data with more than four states. Generalization to nonbinary trees is also fairly straightforward.

A more interesting extension is to consider subsets of data that do not share the same taxon set. This is related to the problem of constructing phylogenetic supertrees (Bininda-Emonds et al., 2002), which are trees assembled from smaller trees so as to minimize any conflict between overlapping taxa. A given data set might be best compressed by associating subsets of the data with proper subtrees rather than trees having all the taxa.

This raises the much more general problem of *finding* the partition that globally minimizes the length of the code among all such partitions. This is obviously a hard problem because even if all subsets have the same taxa, the number of subsets is exponential in the number of characters in the matrix. Finding the MP tree for any element of any partition is already an NP hard problem, so finding it for all elements of all possible partitions is unlikely to be easy. However, it might be possible to impose some biologically relevant constraints on the kinds of partitions to be examined.

Finally, for some data it is conceivable that a graph rather than a tree would provide a better compression scheme. Consensus networks (Holland et al., 2004) have been used to summarize and visualize information about conflicts. However, graphs require more symbols to encode than trees do, and then yet more symbols are required to encode the character state changes associated with a graph, so presumably only data sets with extraordinarily high levels of conflict would benefit from this. Homoplasy can often be reduced by adding enough reticulations, but this seems like it offers too ad hoc a strategy for explaining away homoplasy—perhaps best avoided as a general method of tree inference. However, an information compression approach imposes a rather stiff penalty that the data must overcome before they support the added complexity of a reticulation hypothesis. Only if the savings in description length because of reduced homoplasy exceeds the extra complexity of handling graphs will such an evolutionary inference be warranted.

ACKNOWLEDGMENTS

We thank J. G. Burleigh, O. Eulenstein, and Raul Piaggio for discussion of these issues, and Mike Steel, Ming Li, and an anonymous reviewer for helpful comments.

REFERENCES

Allison, L., D. Powell, and T. I. Dix. 1999. Compression and approximate matching. Comp. J. 42:1–10.

Allison, L., L. Stern, T. Edgoose, and T. I. Dix. 2000. Sequence complexity for biological sequence analysis. Computers and Chemistry 24:43–55.

Allison, L., and C. N. Yee. 1990. Minimum message length encoding and the comparison of macromolecules. Bull. Math. Biol. 52:431–453.

Aris-Brosou, S. 2003. Least and most powerful tests to elucidate the origin of the seed plants in the presence of conflicting signals under misspecified models. Syst. Biol. 52:781–793.

Bapteste, E., H. Brinkmann, J. A. Lee, D. V. Moore, C. W. Sensen, P. Gordon, L. Durufle, T. Gaasterland, P. Lopez, M. Muller, and H. Philippe. 2002. The analysis of 100 genes supports the grouping of three highly divergent amoebae: Dictyostelium, Entamoeba, and Mastigamoeba. Proc. Natl. Acad. Sci. USA 99:1414–1419.

Barker, F. K., and F. M. Lutzoni. 2002. Utility of the incongruence length difference test. Syst. Biol. 51:625–637.

Bergthorsson, U., K. L. Adams, B. Thomason, and J. D. Palmer. 2003. Widespread horizontal transfer of mitochondrial genes in flowering plants. Nature 424:197–201.

Bininda-Emonds, O. R. P., J. Gittleman, and M. Steel. 2002. The (super)tree of life: procedures, problems, and prospects. Ann. Rev. Ecol. Syst. 33:265–290.

Bull, J. J., J. P. Huelsenbeck, C. W. Cunningham, D. L. Swofford, and P. J. Waddell. 1993. Partitioning and combining data in phylogenetic analysis. Syst. Biol. 42:384–397.

Burnham, K. P., and D. R. Anderson. 1998. Model selection and inference. Springer, New York.

Cheeseman, P., and B. Kanefsky. 1993. The reconstruction of evolutionary trees using minimal description length. Pages 91–100 *in* Advances in computer methods for systematic biology: Artificial intelligence, databases, computer vision (R. Fortuner, ed.). Johns Hopkins Press, Baltimore.

Chen, X., M. Li, B. Ma, and J. Tromp. 2002. DNACompress: Fast and effective DNA sequence compression. Bioinformatics (Oxford) 18:1696–1698.

Cover, T. M., and J. A. Thomas. 1991. Elements of information theory. John Wiley & Sons, New York.

Cronn, R., and J. F. Wendel. 2003. Cryptic trysts, genomic mergers, and plant speciation. New Phytologist 161:133–142.

Cunningham, C. W. 1997a. Can three incongruence tests predict when data should be combined? Mol. Biol. Evol. 14:733–740.

Cunningham, C. W. 1997b. Is congruence between data partitions a reliable predictor of phylogenetic accuracy? Empirically testing an iterative procedure for choosing among phylogenetic methods. Syst. Biol. 46:464–478.

Darlu, P., and G. Lecointre. 2002. When does the incongruence length difference test fail? Mol. Biol. Evol. 19:432.

Day, W. H. E. 1983. The role of complexity in comparing classifications. Math. Biosci. 66:97–114.

de Queiroz, A., M. J. Donoghue, and J. Kim. 1995. Separate versus combined analysis of phylogenetic evidence. Ann. Rev. Ecol. Syst. 26:657–681.

Dolphin, K., R. Belshaw, C. D. L. Orme, and D. L. J. Quicke. 2000. Noise and incongruence: Interpreting results of the incongruence length difference test. Mol. Phylogenet. Evol. 17:401–406.

Donoghue, M. J., and J. A. Doyle. 2000. Seed plant phylogeny: demise of the anthophyte hypothesis? Curr. Biol. 10:R106–R109.

Doyle, J. J., J. L. Doyle, J. T. Rauscher, and A. H. D. Brown. 2003. Diploid and polyploid reticulate evolution throughout the history of the perennial soybeans (Glycine subgenus Glycine). New Phytologist 161:121–132.

Edwards, A. W. F., and L. L. Cavalli-Sforza. 1964. Reconstruction of evolutionary trees. Pages 67-76 *in* Phenetic and phylogenetic classification (V. H. Heywood and J. McNeill, eds.). Systematics Association Publication, London.

Farris, J. S. 1979. The information content of the phylogenetic system. Syst. Zool. 28:483–519.

Farris, J. S. 1980. THe efficient diagnoses of the phylogenetic system. Syst. Zool. 29:386–401.

Farris, J. S., M. Kallersjo, A. G. Kluge, and C. Bult. 1994. Testing significance of incongruence. Cladistics 10:315–319.

Farris, J. S., M. Kallersjo, A. G. Kluge, and C. Bult. 1995. Constructing a significance test for incongruence. Syst. Biol. 44:570–572.

Felsenstein, J. 2001. The troubled growth of statistical phylogenetics. Syst. Biol. 50:465–467.

Hein, J. 1990. Reconstructing evolution of sequences subject to recombination using parsimony. Math. Biosci. 98:185–200.

Hein, J. 1993. A heuristic methdo to reconstruct the history of sequences subject to recombination. J. Mol. Evol. 36:396–405.

Hipp, A. L., J. C. Hall, and K. J. Sytsma. 2004. Congruence versus accuracy: Revisiting the incongruence length difference test. Syst. Biol. 53:81–89.

Holland, B. R., K. T. Huber, V. Moulton, and P. J. Lockhart. 2004. Using consensus networks to visualize contradictory evidence for species phylogeny. Mol. Biol. Evol. 21:1459–1461.

Huelsenbeck, J. P., J. J. Bull, and C. W. Cunningham. 1996. Combining data in phylogenetic analysis. Trends Ecol. Evol. 11:152–158.

Kitching, I. J., P. L. Forey, C. J. Humphries, and D. Williams. 1998. Cladistics: The theory and practice of parsimony analysis, 2nd edition. Oxford University Press, New York.

Kluge, A. G. 1989. A concern for evidence and a phylogenetic hypothesis of relationships among Epicrates (Boidae, Serpentes). Syst. Zool. 38:7–25.

Kluge, A. G., and A. J. Wolf. 1993. Cladistics: What's in a word? Cladistics 9:183–199.

Kurland, C. G., B. Canback, and O. G. Berg. 2003. Horizontal gene transfer: A critical view. Proc. Natl. Acad. Sci. USA 100:9658–9662.

Lerat, E., V. Daubin, and A. Moran. 2003. From gene trees to organismal phylogeny in prokaryotes: The case of the gamma-Proteobactera. PLoS Biol. 1:1–9.

Li, M., J. Badger, X. Chen, S. Kwong, P. Kearney, and H. Y. Zhang. 2001. An information based distance and its application to whole mitochondrial genome phylogeny. Bioinformatics (Oxford) 17:149–154.

Li, M., J. Tromp, and L. Zhang. 1996. On the nearest neighbor interchange distance between evolutionary trees. J. Theor. Biol. 182:463–467.

Li, M., and P. Vitanyi. 1997. An introduction to Kolmogorov complexity and its applications, 2nd edition. Springer-Verlag, New York.

Magallon, S., and M. J. Sanderson. 2002. Relationships among seed plants inferred from highly conserved genes: Sorting conflicting phylogenetic signals among ancient lineages. Am. J. Bot. 89:1991–2006.

Milosavljevic, A., D. Haussler, and J. Jurka. 1990. Clustering of macromolecular sequences by minimal length encoding. Pages 90–94 *in* Artificial intelligence and molecular biology: Working notes, Stanford University.

Murphy, W. J., E. Eizirik, W. E. Johnson, Y. P. Zhang, O. A. Ryder, and S. J. O'Brien. 2001. Molecular phylogenetics and the origins of placental mammals. Nature 409:614–618.

Nixon, K. C., and J. M. Carpenter. 1996. On simultaneous analysis. Cladistics 12:221–241.

Otu, H. H., and K. Sayood. 2003. A new sequence distance measure for phylogenetic tree construction. Bioinformatics (Oxford) 19:2122–2130.

Page, R. D. M. 1993. COMPONENT user's manual (version 2.0). Trustees of The Natural History Museum, London.

Page, R. D. M. 1996. On consensus, confidence and "total" evidence. Cladistics 12:83–92.

Page, R. D. M., and M. A. Charleston. 1997. From gene to organismal phylogeny: Reconciled trees and the gene tree/species tree problem. Mol. Phylogenet. Evol. 7:231–240.

Ren, F., H. Tanaka, and T. Gojobori. 1995. Construction of molecular evolutionary phylogenetic trees from DNA sequences based on minimum complexity principle. Computer methods and programs in biomedicine 46:121–130.

Rokas, A., B. Williams, N. King, and S. Carroll. 2003. Genome-scale approaches to resolving incongruence in molecular phylogenies. Nature 425:798–804.

Rydin, C., M. Kallersjo, and E. M. Friis. 2002. Seed plant relationships and the systematic position of Gnetales based on nuclear and chloroplast DNA: Conflicting data, rooting problems and the monophyly of conifers. Int. J. Plant Sci. 163:197–214.

Sanderson, M. J., A. C. Driskell, R. H. Ree, O. Eulenstein, and S. Langley. 2003. Obtaining maximal concatenated phylogenetic data sets from large sequence databases. Mol. Biol. Evol. 20:1036–1042.

Sanderson, M. J., M. F. Wojciechowski, J. M. Hu, T. S. Khan, and S. G. Brady. 2000. Error, bias, and long-branch attraction in data for two chloroplast photosystem genes in seed plants. Mol. Biol. Evol. 17:782–797.

Semple, C., and M. Steel. 2003. Phylogenetics. Oxford University Press, New York.

Thornton, J. W., and R. DeSalle. 2000. A new method to localize and test the significance of incongruence: detecting domain shuffling in the nuclear receptor subfamily. Syst. Biol. 49:183–201.

Vogl, C., J. Badger, P. Kearney, M. Li, M. Clegg, and T. Jian. 2003. Probabilistic analysis indicates discordant gene trees in chloroplast evolution. J. Mol. Evol. 56:330–340.

Willson, S. J. 1999. A higher order parsimony method to reduce long-branch attraction. Mol. Biol. Evol. 16:694–705.

Yoder, A. D., J. A. Irwin, and B. A. Payseur. 2001. Failure of the ILD to determine data combinability for slow loris phylogeny. Syst. Biol. 50:408–424.

First submitted 2 April 2004; reviews returned 27 May 2004; final acceptance 10 July 2004
Associate Editor: Mike Steel

Appendix A: Coding Scheme Description

This appendix is devoted to the coding schemes' detailed description and the derivation of Equations 1, 2, and 3 giving the lengths of the various compressed data sets.

The data matrix is coded with binary characters. Recall that a naive method would be to encode each nucleotide one after another, using 2 bits for each. A matrix with n sequences and m characters would be coded in $2nm$ bits. Though very simple, this naive method is still a compression scheme as standard text files use 1 byte (or 8 bits) for each symbol. In our experience, when run on such files the Unix command "compress" does not even reduce them to less than 2 bits per nucleotide.

Our compression schemes are intended to exploit sequence similarities to reduce the code length to less than $2nm$. Three schemes are presented here. The first one uses the "total evidence" MP tree. Only binary trees are considered. If an MP tree is not binary then it is arbitrarily resolved, as this operation does not change its parsimony score. The second scheme allows for partitioning the data into several consecutive submatrices. Associated MP trees are described separately. The third coding scheme is similar to the previous one except that MP trees are described by reference to already described trees. In each scheme, the resulting compressed file is organized in two parts: an introductory part contains the model description (MP tree[s], number of such trees if necessary) and the main part contains the matrix itself. Methods will be illustrated for the matrix shown in Figure 5. It is divided into 2 submatrices of size $m_1 = 4$, $m_2 = 2$ whose binary MP trees are as shown in Figure 6. Sequence labels a, b, c, and d (Fig. 5) are used to keep track of the sequence order in the matrix, but we do not intend to code them. This task may be done separately.

Before going further, let us recall some notation. Let lg denote the function $\log_2$ in base 2, rounded to the nearest larger or equal integer: $\lg(k) = \lceil \log_2(k) \rceil$. It is the smallest number of bits needed to code letters from an alphabet of size k with a fixed length code. When needed, the encoding of unbounded length integers will use a logarithmic ramp, and the length of the description of k will be denoted by des(k). More specifically, we have $\mathrm{des}(k) = \lg(k) + \lg(\lg(k)) + \cdots + 2 + 1$ whenever k is not a power of 2 (see Li and Vitanyi [1997] for more detail).

A.1: Matrix Encoding

Let us first assume that the model has been encoded, i.e., that the number of trees and the tree(s) are known, along with the root(s) and the edge ordering. In our example (Fig. 5), the model encoding would give, say, the two trees of Figure 6 both rooted at sequence a. It would not give the size $m_1 = 4$ and $m_2 = 2$ of the submatrices, though. However, as will be seen later, each (sub)matrix description ends with an end-of-matrix symbol, which is classified as an end-of-file symbol if there is a single matrix or for the last submatrix. Submatrices can then be safely concatenated. We now describe the coding procedure for one (sub)matrix. This procedure is shared by the three coding schemes.

$$\begin{array}{c|cc} a & \mathrm{A\,C\,C\,A} & \mathrm{A\,A} \\ b & \mathrm{A\,T\,T\,T} & \mathrm{A\,A} \\ c & \mathrm{A\,C\,C\,T} & \mathrm{G\,G} \\ d & \mathrm{A\,T\,T\,C} & \mathrm{G\,G} \\ & \underbrace{\phantom{\mathrm{A\,C\,C\,A}}}_{D_1} & \underbrace{\phantom{\mathrm{A\,A}}}_{D_2} \end{array}$$

FIGURE 5. Data matrix used to illustrate the coding scheme.

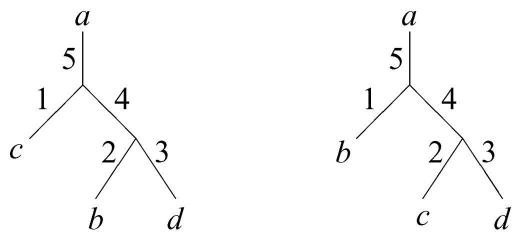

FIGURE 6. Most parsimonious trees for matrices in Figure 5. On the left is the MP tree of the left submatrix. On the right is the MP tree of the right submatrix. Edges are numbered according to a postorder tree traversal.

A pattern can be described by the state at the root and the substitutions that occurred along the tree. Coding the nucleotide at the root just takes 2 bits. For example, A may be translated to 00, C to 01, G to 10, and T to 11. Then a substitution is described by the new nucleotide replacing the "old" one and by the edge where it occurred. We need 2 bits to code the new nucleotide and $\lg(2n - 3)$ bits for the edge, as the binary tree has exactly $2n - 3$ edges. As we know the new nucleotide is different from the ancestral one, we will use this piece of information in the coding scheme. It will be useful later. More precisely, instead of coding the new nucleotide, we code the change of nucleotide, described by the difference between the new state and the ancestral one. It requires 2 bits as well, but the sequence 00, which codes for "no difference," would not be used. The edge number is described after the nucleotide change.

In the matrix of Figure 5, sites 1 to 4 are treated with respect to the first tree (on the left), whereas the second tree is used for sites 5 and 6. Encoding site 1 just requires us to code the ancestor A (00) and no substitution at all. Coding site 2 requires us to code the ancestor C (01) and one substitution from C to T on edge number 4 (coded by 100). T should be coded by 11; therefore, the difference between T and C is $11 - 01 = 10$. So the code for site 2 would be: 01 10 100. The fourth site is more complicated, as it needs 2 substitutions. It starts with an A (00) at the root. On edge number 5 (coded by 101), this A is replaced by a T (11). This substitution is coded by $11 - 00 = 11$. On edge number 3 (011) this T is replaced by a C, and this is coded by $01 - 11 = -10 = 10$ modulus 4. The whole code for site 4 would be 00 11 101 10 011. Because sites do not necessarily share the same number of substitutions, they do not share the same coding length either. Therefore, a signal is needed to separate them. This can be done by adding a 00 at the end of each site description. This 00 follows an edge number or an initial root state, where a substitution would be expected. At this decoding state, 00 means "no change in nucleotide," which is instead understood as "end of the current pattern description." In our example, the first site would be coded by 00 00, site 2 by 0110100 00, and so on. These coding words can be concatenated and form the final encoded matrix description: 0000 011010000 011010000 00111011001100.

To be complete, the code needs an end-of-matrix symbol. As is described, the last symbol of the matrix is the no-nucleotide-change symbol 00 signaling the end of a site description. This last 00 is actually replaced by a fake parsimony step description starting with, say 01, and ending with the description of an edge that does not exist, say $0 \ldots 0$ ($\lg(2n - 3)$ times). Indeed, there is an odd number $2n - 3$ of edges, and all $\lg(2n - 3)$-bit-long symbols are not used. In our example, 000 does not code for any edge. It is used as an end-of-matrix symbol.

The length of the matrix description is uniquely determined by the number m of sites (m_j in case of one submatrix) and the number of substitutions, which is the parsimony score L(or $L(T_j, D_j)$ in case of one submatrix). The root sequence is coded with $2m$ bits. The 00s separating the sites also take $2m$ bits. The remaining bits are due to substitutions. Each one needs $2 + \lg(2n - 3)$ bits. The end-of-matrix signal adds $\lg(2n - 3)$ bits. Summing up these quantities leads to a total of

$$4m + (2 + \lg(2n - 3))L + \lg(2n - 3) \text{ bits.}$$

A.2: Model Encoding

The introductory part of the compressed file is still to be defined. Its purpose is to give the model, including the tree(s) and possibly the matrix structure. It therefore differs from one coding scheme to another.

A.2.1: Single tree coding scheme.—With the simplest coding scheme just a single tree has to be described. At first, labels are ignored. The tree shape is encoded using the parenthesis description derived from the widely used Newick tree format: parentheses are kept and the rest (labels, commas) is removed. Then a left parenthesis is coded by 1 and a right one by 0. For example, the left tree in Figure 6 can be written as (a, c, (b, d)) in Newick format. Only (()) would be retained and finally coded by 1100. Notice that this is enough information to recover the tree shape because the tree is known to be binary.

Such a description is self-delimited and does not require the a priori knowledge of the number of taxa. This is because there are as many right parentheses as there are left parentheses, and because the right parenthesis closing the first left one is precisely the last parenthesis. A counter can be set to 0 at the beginning of the description and incremented by $+1$ (respectively -1) when a left (respectively right) parenthesis is encountered. The tree description is over when the counter returns to 0. The number of pairs of parentheses is the number of internal nodes in T, which is $n-2$. It follows that the number of taxa n is known at this point, and that the tree shape description exactly takes $2n-4$ bits.

Once the tree shape is known, sequences need to be mapped on the tips. Tips of T may be numbered in an implicit order, related to the order in which internal nodes appear in the tree. On the other hand, taxa are numbered in the order in which they appear in the matrix. For each i, let $j(i)$ be the number of the taxon that should be mapped on tip number i. Then the sequence $j(1), \dots j(n)$ is encoded in base 2. As there are n taxa we may use $\lg(n)$ bits for each one of them. Let us go back to the left tree of Figure 6. The decoder knows the shape: (()) and wants to recover the Newick description: (?,?,(?,?)) with the taxa numbers instead of question marks. The sequence to be encoded is then 1,3,2,4. As we want to spend only 2 bits for each number, we start numbering taxa from 0 and get the new sequence 0,2,1,3 coded by 00 10 01 11. The number of bits used in this step is $n\lg(n)$. Adding this to the previous step gives a total of

$$2n - 4 + n\lg n \text{ bits.}$$

Adding this length to the previous matrix description length gives the total length claimed in Equation 1. Now that the tree encoding is complete, the root is arbitrarily set to the first tip. Edges of T are also numbered by the postorder tree traversal, for matrix encoding purposes (section A.1).

A.2.2: Forest coding scheme with separately described trees.—When the model is a forest with an arbitrary number of trees, this number l of trees is first encoded with des(l) bits. For example, $l = 2$ is encoded by 10 0 of length des(2) = 3. Then each tree is coded just as described in the previous section. Trees are described separately, each description having the same length, $2n-4+n\lg n$ bits. Matrix descriptions follow. They are naturally separated by the previous end-of-file signals, that now tell where the matrix is divided into submatrices. As the number l of submatrices has been given before, the lth end-of-file signal is the true one. The total size of the compressed file is now

$$\mathrm{des}(l) + \underbrace{l(2n-4+n\lg(n))}_{\text{trees}} + l\lg(2n-3)$$
$$+ \underbrace{4m + (2+\lg(2n-3))L^{\text{forest}}}_{\text{matrix}} \text{ bits,}$$

which is the length given in Equation 2.

A.2.3: Forest coding scheme with trees described by rearrangement.—The last coding scheme also uses a forest. The number l of trees is first encoded, the first tree is described just as before, but each following tree is then described by its differences with respect to the first tree.

The description of a binary tree T' using a binary tree T is now described. We assume that T is rooted at one of its tips and that its edges are ordered. Tree T' will inherit T's root and T's edge ordering. Recall that the first tree T_1 was rooted to the first encountered tip, and its edges were numbered. We assume that the NNI (nearest neighbor interchange) distance k between T' and T has been computed, and that an optimal set of k NNI operations has been determined. Tree T' is then described by the encoding of the integer k followed by the descriptions of the k optimal NNI. In order to explain how an interchange is encoded, let us recall how it is defined. Two NNI operations may be done with respect to a given edge e. Let a, b, c, and d be the edges adjacent to e, such that a leads to the root, b is next to a and c has a lower edge number than d, as illustrated in Figure 7.

Two interchanges may occur around e, leading either to tree T_c or to T_d, edge b being swapped with either edge c or edge d. Both operations are defined by the edge e and an extra bit of information. We may use the bit 0 (respectively 1) for the operation leading to T_c (respectively T_d). There are $n-3$ internal edges, so coding e takes $\lg(n-3)$ bits, and each NNI operation takes $\lg(n-3)+1$ bits. The description of T' takes then a total of des(k) + $k(\lg(n-3)+1)$ bits. Recall the trees in Figure 6. Describing the second one from the first one would first require 1 bit (0) to encode $k = 1$. As there are only $n-3 = 1$ internal edges, there is no choice about the NNI edge. No bit (no information) is required for that. In the first tree, the edge sister to the root leads to c and needs to be swapped with the edge having thus the smallest number (the one leading to b). This NNI operation thus receives code 0. The total second tree description is then 00. With this coding scheme, the second tree gets a different edge ordering from what is shown in Figure 5. The branch from c (respectively b) inherits number 1 (respectively 2) from the first tree.

If k_i is the NNI distance between trees T_i and T_1 and $\bar{k} = (\sum_{i=2}^{l} k_i)/(l-1)$ is the average distance then the file is encoded with

$$\mathrm{des}(l) + \underbrace{2n-4+n\lg(n)}_{\text{first tree}}$$
$$+ \underbrace{\bar{k}(l-1)(\lg(n-3)+1) - \mathrm{des}(k_2) + \cdots + \mathrm{des}(k_l)}_{\text{next trees}}$$
$$+ l\lg(2n-3) + \underbrace{4m + (2+\lg(2n-3))L^{\text{forest}}}_{\text{matrix}} \text{ bits,}$$

which is the length given in Equation 3.

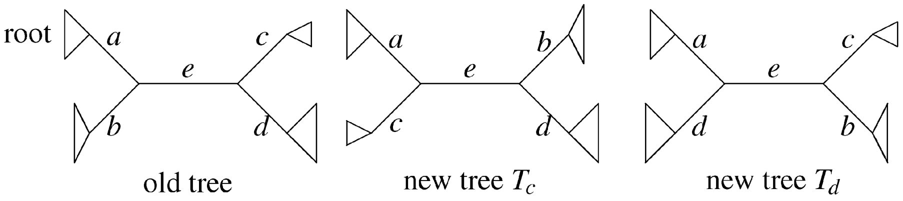

FIGURE 7. The two possible nearest neighbor interchange (NNI) operations around edge e. Edge c must have a lower edge number than d (see Fig. 6).

APPENDIX B: MODEL COMPARISON

The compression length in Equation 2 (several separately described trees) is shorter than the compression length in Equation 1 (one tree) as soon as $L^{(TE)} - L^{\text{forest}}$ is greater than the cutoff value

$$\Delta L^{\text{sep}}_{\max} = (l-1)\frac{2n - 4 + n\lg(n) + \lg(2n-3)}{2 + \lg(2n-3)} + \frac{\text{des}(l)}{2 + \lg(2n-3)}.$$

The first term on the right is less than or equal to $(l-1)(n+1)$ and is actually equivalent to $(l-1)n$ when n is large. As the second term tends to zero, it follows that for large n,

$$\Delta L^{\text{sep}}_{\max} \sim (l-1)n.$$

The compression length in Equation 3 (forest and trees described by reference to the first one) is shorter than the compression length in Equation 1 (one tree) as soon as $L^{(TE)} - L^{\text{forest}}$ is large enough. The cutoff value is now

$$\Delta L^{\text{ref}}_{\max} = (l-1)\frac{\bar{k}(\lg(n-3)+1) + \lg(2n-3)}{2 + \lg(2n-3)} + \frac{\text{des}(l) + \sum_{i=2}^{l}\text{des}(k_i)}{2 + \lg(2n-3)}$$

Recall that $\bar{k}$ is the average NNI distance between the first tree and the others. Here again the first term on the right is always $\leq (l-1)(\bar{k}+1)$. Moreover, the second term is negligible with respect to the first one since all k_i are bounded by $n\log(n) + O(n)$ (Li et al., 1996) and $\text{des}(k_i) = O(\lg(n))$. For large n we now have

$$\Delta L^{\text{ref}}_{\max} \sim (l-1)\bar{k}$$

APPENDIX C: EXTENDING THE ALPHABET TO GAPS OR OTHER SYMBOLS

As a general rule, aligned sequences contain gaps, ambiguous or missing data, or might code for amino acid data. Let us consider an alphabet with c letters. It could be $\{A, C, G, T, \text{—}\}$ for instance, with $c = 5$. This section describes an adaptation of the coding scheme to this new alphabet. The resulting model selection criterion is derived.

The naive encoding with a fixed-length code would use $b = \lg(c)$ bits for each letter, yielding a file of bnm bits. With DNA sequences, gaps, and/or missing data, we have $b = 3$. With 20 amino acids we have $b = 5$.

C.1: Coding Scheme

There is no difference in the model (tree(s), matrix structure), so that the only part that differs is the matrix encoding. When describing a site, gaps (—) or missing data (?) are treated just like other standard symbols. It differs from usual phylogenetic analyses in that a step is needed here to go from a letter to a "?," for instance. The parsimony score L may therefore be higher here than in usual analyses. The widely used PAUP program optionally allows users to count these gaps or "?" as necessitating extra steps.

The description of a given site or pattern includes the same elements as before. It starts with the "ancestral" state, using b bits instead of 2. Then for each parsimony step, the new state is described by encoding the change of state instead of the new state itself, using again b bits instead of 2. The edge description still takes $\lg(2n-3)$ bits, as does the end-of-file symbol. The pattern description finally ends with the signal $0\ldots0$ (b times). The total compressed matrix length is then

$$2bm + (b + \lg(2n-3))L + \lg(2n-3),$$

resulting in a compressed file length of $2n - 4 + n\lg(n)\lg(2n-3) + 2bm + (b + \lg(2n-3))L$ when using a single tree. When this is below bnm, compression is efficient. If m is large, the homoplasy L/m needs to fall below $b(n-2)/(b + \lg(2n-3)) \sim bn/\lg(n)$ for the compression to be efficient.

C.2: Model Selection

As before, the model selection can be based on $L^{(TE)} - L^{\text{forest}}$. If it is greater than $\Delta L_{\max}$, then the forest is preferred. The cutoff value $\Delta L_{\max}$ is now

$$(l-1)\frac{2n - 4 + n\lg(n) + \lg(2n-3)}{b + \lg(2n-3)} + \frac{\text{des}(l)}{b + \lg(2n-3)}$$

or

$$(l-1)\frac{\bar{k}(\lg(n-3)+1) + \lg(2n-3)}{b + \lg(2n-3)} + \frac{\text{des}(l) + \sum_{i=2}^{l}\text{des}(k_i)}{b + \lg(2n-3)}$$

depending on the chosen scheme. Interestingly, the asymptotic behavior (when n is large) is as before: $(l-1)n$ or $(l-1)\bar{k}$. It does not depend on the number of symbols in the alphabet.