Spectral Partitioning of Phylogenetic Data Sets Based on Compatibility

DUHONG CHEN,¹ J. GORDON BURLEIGH,² ³ AND DAVID FERNANDEZ-BACA¹

¹Department of Computer Science, Iowa State University, Ames, IA 50011, USA
²Section of Evolution and Ecology, University of California, Davis, CA 95616, USA
³NEStCent, Durham, NC 27705, USA; E-mail: jgb12@duke.edu

Abstract.—We describe two new methods to partition phylogenetic data sets of discrete characters based on pairwise compatibility. The partitioning methods make no assumptions regarding the phylogeny, model of evolution, or characteristics of the data. The methods first build a compatibility graph, in which each node represents a character in the data set. Edges in the compatibility graph may represent strict compatibility of characters or they may be weighted based on a fractional compatibility scoring procedure that measures how close the characters are to being compatible. Given the desired number of partitions, the partitioning methods then seek to cluster the characters with the highest average pairwise compatibility, so that characters in each cluster are more compatible with each other than they are with characters in the other cluster(s). Partitioning according to these criteria is computationally intractable (NP-hard); however, spectral methods can quickly provide high-quality solutions. We demonstrate that the spectral partitioning effectively identifies characters with different evolutionary histories in simulated data sets, and it is better at highlighting phylogenetic conflict within empirical data sets than previously used partitioning methods. [Compatibility; spectral methods; phylogeny; phylogenetic conflict; partitioning.]

The treatment of conflict within data sets is among the most debated questions in phylogenetics (e.g., Bull et al., 1993; Chippindale and Wiens, 1994; De Queiroz et al., 1995; Farris et al., 1994; Huelsenbeck et al., 1996; Cunningham, 1997; Thornton and DeSalle, 2000; Wilgenbusch and De Queiroz, 2000; Brandley et al., 2005), and it is an especially relevant concern with the prevalence of large data sets that combine data from many sources. For example, recent genomic-scale phylogenetic analyses have called attention to tremendous variation in the phylogenetic signal among loci (Bapteste et al., 2002; Rokas et al., 2003; Driskell et al., 2004). Phylogenetic conflict may be caused by heterogeneity in the process of evolution, such as differing relative branch lengths among characters across the phylogeny (e.g., Rokas, 2003). Alternately, characters in a data set may have different phylogenies due to processes such as hybridization, recombination, lineage sorting, or horizontal transfer (e.g., Doyle et al., 2003; Takahasi et al., 2001; Lerat et al., 2003). To understand variation in the processes of evolution within a data set and to consider the effects of conflicting phylogenetic signals within a data set, it is first necessary to identify and describe the phylogenetic conflict within a data set.

Perhaps the most common strategies to reveal phylogenetic conflict within a data set compare phylogenies from nonoverlapping sets of characters. For molecular data, partitioning characters by locus or genome, codon position, coding and noncoding regions, or stem and loop regions may reveal evidence of phylogenetic conflict (e.g., Brandley et al., 2005). These biologically defined partitioning strategies require some knowledge regarding the characteristics of the data that are likely to reveal conflict, and they may miss unexpected patterns of conflict within the data. Other partitioning methods seek to cluster characters based on their rates of evolution (Brinkmann and Philippe, 1999; Hirt et al., 1999; Burleigh and Mathews, 2004; Pisani, 2004) rather than conflicting phylogenetic signals. The methods of Brinkmann and Philippe (1999) and Burleigh and Mathews (2004) require some knowledge of the phylogeny, and Burleigh and Mathews’ (2004) method also assumes a model of evolution.

The goal of this paper is to describe a fast approach for partitioning phylogenetic data sets that explicitly seeks to highlight conflicting phylogenetic signals among characters within a data set. Additionally, we want partitioning methods that do not require assumptions regarding the phylogeny or process of evolution. We introduce two related partitioning methods that partition discrete phylogenetic characters based on pairwise compatibility. The methods first require building a compatibility graph for the data, in which each character is represented by a node and the edges or edge weights connecting nodes are based on pairwise compatibility. Next, given a specified number of partitions, the methods estimate the clusters of characters with the highest pairwise compatibility (or average edge weight), so that on average, characters in each cluster are more compatible with each other than with characters in the other cluster(s). Partitioning according to these criteria is computationally intractable (NP-hard; Shi and Malik, 2000) and thus requires heuristic solutions. We describe and implement spectral methods that can quickly estimate the optimal partitions based on pairwise compatibility. We demonstrate that spectral partitioning methods effectively identify characters with different phylogenies in simulated data sets, and they identify greater phylogenetic conflict than previously used partitioning methods in selected empirical data sets.

METHODS

Our partitioning methods use compatibility to cluster phylogenetic characters. A set of characters C is said to be compatible if it admits a perfect phylogeny, or in other words, if C can evolve on a single topology without homoplasy (Le Quesne, 1969). If C is compatible, all
characters in C must be pairwise compatible; however, if there are more than two character states and/or missing data, sets of pairwise compatible characters are not necessarily compatible (see Felsenstein, 2004). Our goal is to partition a set of discrete characters into two or more clusters so that characters in the same cluster are more compatible with each other than they are to characters in different clusters. We use strict compatibility, as well as a fractional compatibility scheme that measures the degree of compatibility between pairs of characters. The latter is useful in situations where strict pairwise compatibility is unlikely, such as when characters originate from large sets of taxa or the characters have fast rates of evolution. We cluster only parsimony-informative characters, because characters that are not parsimony informative are compatible with all other characters and will fit equally well into any cluster. The implementation of our partitioning methods removes the uninformative sites prior to partitioning and places them in a separate cluster.

The Pairwise Character Compatibility Graph

The pairwise character compatibility graph (PCCG) of a set of characters C, denoted by \( G_C \), is a complete edge-weighted graph whose vertex set is C (see Fig. 1). We consider two versions of the PCCG. In the 0-1, or strict compatibility, version, the weight of an edge \((c_1, c_2)\) in \( G_C \) is 1 if \( c_1 \) and \( c_2 \) are compatible and 0 otherwise. In the more general setting, or the fractional compatibility version, the weight of edge \((c_1, c_2)\) is a number between 0 and 1 that reflects the degree of compatibility between \( c_1 \) and \( c_2 \). Note that, in practice, we remove all edges whose weight is 0 from the PCCG.

To construct the 0-1 version of PCCG, we first need to test the compatibility of every pair of characters \( c_1 \) and \( c_2 \). To this end, define the state intersection graph (SIG) of \( c_1 \) and \( c_2 \) as the bipartite graph \( H(c_1, c_2) = (V_1, V_2, E) \), where \( V_1 \) and \( V_2 \) correspond to the states of character \( c_1 \) and \( c_2 \), respectively. More specifically, for \( i = 1 \) or 2, suppose \( c_i \) has character states 1 through \( r_i \), and let \( S_j \) denote the set of all taxa with state \( j \) on character \( i \). Then, \( V_i \) consists of vertices \( S_{1j}, \ldots, S_{r_j} \). There is an edge between \( S_{ip}(1 \leq p \leq r_1) \) and \( S_{iq}(1 \leq q \leq r_2) \) if and only if \( S_{ip} \) and \( S_{iq} \) have at least one taxon in common (see Fig. 1a). It is known that two characters \( c_1 \) and \( c_2 \) are compatible if and only if \( H(c_1, c_2) \) is acyclic (Estabrook and McMorris, 1977; Setubal and Meidanis, 1997; Fig. 1a). This can be tested in time linear in the size of \( H(c_1, c_2) \).

In the more general case of the PCCG, the weight of an edge is given by a fractional character compatibility score, denoted by \( \text{fcc}(c_1, c_2) \), which measures the degree of compatibility between \( c_1 \) and \( c_2 \) based on parsimony. The value of \( \text{fcc}(c_1, c_2) \) is given by

\[
\text{fcc}(c_1, c_2) = \frac{q - p}{q} = 1 - \frac{p}{q}
\]

where \( p \) is the minimum number of edges whose removal from \( H(c_1, c_2) \) yields an acyclic graph and \( q = r_1r_2 - (r_1 + r_2 - 1) \), where, as before, \( r_1 \) and \( r_2 \) are the number of states in \( c_1 \) and \( c_2 \), respectively. Note that \( q \) is precisely the number of edges that need to be removed from a complete bipartite graph with \( r_1 \) and \( r_2 \) vertices in each part in order to make it acyclic. Hence, \( q \) is an upper bound on \( p \), and thus, \( \text{fcc}(c_1, c_2) \) is a number between 0 and 1.

The following result, which is proved in the appendix, shows the relationship between \( \text{fcc}(c_1, c_2) \), parsimony, and compatibility. Let \( \text{pars}(c_1, c_2) \) be the parsimony score of characters \( c_1 \) and \( c_2 \); that is, the smallest number of state changes in a phylogeny for \( c_1 \) and \( c_2 \).

**Theorem 1:** Let \( p \) be the number of edges that need to be removed from \( H(c_1, c_2) \) to make it acyclic, for some pair of
characters $c_1$ and $c_2$ with $r_1$ and $r_2$ states, respectively. Then, $p = \text{pars} (c_1, c_2) - (r_1 + r_2 - 2)$.

Thus, if $c_1$ and $c_2$ are compatible, $p = 0$, and $f_{cc}(c_1, c_2) = 1$. On the other hand, the highest degree of incompatibility between $c_1$ and $c_2$ occurs when these characters jointly exhibit all possible combinations of states. In this case $H(c_1, c_2)$ is a complete bipartite graph. Thus, $p = q$ and $f_{cc}(c_1, c_2) = 0$. Theorem 1 was independently proven by Bruen and Bryant (2007), who refer to the value $p$ as the refined incompatibility score of characters $c_1$ and $c_2$ (see also Bruen et al., 2006, and Althaus and Naujoks, 2006).

Graph Partitioning

The goal of graph partitioning is to divide the vertices of a graph, which for our purposes will be a PCCG, into clusters subject to size or balance constraints while minimizing the weight of the interconnections among those clusters (see Fig. 1b). Formally, let $G = (V, E, w)$ be an edge-weighted undirected graph, where $V$ is the set of nodes, $E$ is the set of edges, and $w$ is the weight function having a number associated with each edge. A two-way partition $(V_1, V_2)$ of $V$, where $V_1 \cup V_2 = V$ and $V_1 \cap V_2 = \emptyset$, is called a cut. The sets of vertices $V_1$ and $V_2$ can be separated by removing the edges connecting them. The value of a cut $(V_1, V_2)$ is:

$$\text{cut}(V_1, V_2) = \sum_{u \in V_1, v \in V_2} w(u, v)$$

A minimum cut (or min-cut for short) is a cut that has minimum value among all cuts in $G$. While a min-cut in $G$ can be found in polynomial time (Hao and Orlin, 1992; Karger and Stein, 1996; Stoer and Wagner, 1997), the min-cut tends to be extremely unbalanced, in that one of the parts may contain very few nodes (Shi and Malik, 2000). These extraneous nodes may be viewed as outliers in the character partition. To make the partition more meaningful, it is useful to modify the scoring of a cut so that a low score implies some balance between the weights of the edges crossing the cut and the sizes of the resulting clusters. One way to do this was proposed by Shi and Malik (2000). They defined the normalized value of a cut $(V_1, V_2)$ as

$$N\text{cut}(V_1, V_2) = \text{cut}(V_1, V_2) \times \left( \frac{1}{c(V_1)} + \frac{1}{c(V_2)} \right)$$

where $c(V_1) = \sum_{u \in V_1, v \in V} w(u, v)$ and $c(V_2) = \sum_{u \in V_2, v \in V} w(u, v)$. The goal is to find a cut with minimum normalized value. For example, consider the 0-1 PCCG in Figure 1b, and the cut $(V_1, V_2)$ where $V_1 = \{c_1, c_2, c_3, c_4\}$ and $V_2 = \{c_5, c_6, c_7\}$. Then, $c(V_1) = 9, c(V_2) = 8$, and $\text{cut}(V_1, V_2) = 3$. Thus, $N\text{cut}(V_1, V_2) = 17/24$.

The notion of a two-way partitioning of a graph can be generalized to an arbitrary number of partitions. A multi-way cut of a graph $G = (V, E, w)$ is a partition $\Delta = \{V_1, V_2, \ldots, V_k\}$ of $V$ into the disjoint nonempty sets $V_1, V_2, \ldots, V_k$. The normalized value of a multi-way cut is defined as follows (Meila and Xu, 2003; Yu and Shi, 2003):

$$MN\text{cut}(\Delta) = \sum_{p=1}^{K} \sum_{q=p+1}^{K} N\text{cut}(V_p, V_q)$$

The problem of finding a $K$-way cut with minimum normalized value is NP-hard for $K \geq 2$ (Meila and Xu, 2003). Nevertheless, spectral clustering can be used to obtain partitions with a small normalized cut value (Ding et al., 2001; Shi and Malik, 2000; Yu and Shi, 2003). We outline the spectral method next.

Spectral Clustering.—The “spectral method” for clustering is actually a class of closely related techniques. These methods usually involve taking the top eigenvectors of some matrix based on similarity between objects and then using them to cluster objects. By “top” eigenvectors, we mean the ones corresponding to either the biggest or smallest eigenvalues, depending on the formulation. In our case, they correspond to the smallest eigenvalues.

The starting point is a matrix $W$ that represents the PCCG $G$ of a set of characters as follows. Let $n$ be the number of vertices in $G$. Then $W$ is an $n$ by $n$ matrix, where, for each pair of nodes $u, v$ in $G$, $W_{uv} = 0$ if there is no edge between $u$ and $v$ or if $u = v$; otherwise $W_{uv}$ is the weight of edge $(u, v)$. Let $d_u = \sum_{v \in V} W_{uv}$ denote the volume of the vertex $u$, and $D$ be the diagonal matrix with the $(u, u)$-th element having value $d_u$. The Laplacian matrix of $G$ is $L = I - D^{-1/2}WD^{-1/2}$, where $I$ is the identity matrix. Shi and Malik (2000) showed that the eigenvector corresponding to the second smallest eigenvalue of $L$ can be used to approximate the cut of $G$ with minimum normalized value.

Having the pairwise compatibility matrix of characters, we can apply a general multi-way spectral clustering algorithm to partition $C$. Here we use the algorithm described by Ng et al. (2002), whose robustness to noise has been demonstrated experimentally by Verma and Meila (2003).

Algorithm for Character Partitioning

Input: A set of characters $C$, and an integer $K$.
Output: A partition $\Delta = \{C_1, C_2, \ldots, C_K\}$ of $C$.
(1) Construct the PCCG of $C$ and its corresponding matrix $W$.
(2) Compute the Laplacian matrix $L = I - D^{-1/2}WD^{-1/2}$.
(3) Let $x_1, x_2, \ldots, x_K$ be eigenvectors corresponding to the $K$ smallest eigenvalues. Form the matrix $X = \{x_1, x_2, \ldots, x_K\}$ by stacking the eigenvectors in columns.
(4) Form the matrix $Y$ from $X$ by renormalizing each of $X$’s rows to have unit length (i.e., $Y_{ij} = \frac{x_{ij}}{\sqrt{\sum x_{ij}^2}}$).
(5) Treating each row of $Y$ as a point in $K$ dimensions, cluster those points using the $K$-means algorithm to obtain $K$ clusters (MacQueen, 1967).
A program that implements spectral partitioning for sets of characters is available at the following website: http://pilin.cs.iastate.edu/public/spectral. Our implementation uses the Lanczos algorithm (Cullum and Willoughby, 2002) to obtain the eigenvectors corresponding to the $K$ smallest eigenvalues of the Laplacian matrix of $G$.

**Simulation Experiments**

We first tested the ability of the spectral partitioning methods to detect sites with different phylogenies using simulated data sets. In each simulation, we obtained two random trees using the default parameters of the YULE.C simulate procedure from r8s, which produced random topologies with a root-tip length of 1.0 (Sanderson, 2003). Alignments of 500 base pairs (bp) in length were generated based on each random tree using the Monte Carlo simulation method in Seq-Gen (Rambaut and Grassly, 1997). Thus, each simulation generates a 1000-bp alignment; the first 500 bp evolving from the first tree, and the second 500 bp evolving from the second tree. The simulations used a Jukes-Cantor (1969) model of sequence evolution, which assumes equal nucleotide frequencies and equal frequencies of all substitutions, and rate variation among sites was modeled by a discrete gamma distribution with four rate categories and an $\alpha$ shape parameter value of 0.5 (Yang, 1994). We performed 10 simulation replicates under two different protocols that were designed to produce different overall levels of pairwise compatibility. In the first simulation protocol, 10 replicates were generated using 25-taxon trees with the branch lengths from r8s scaled by a factor of 0.1. In the second protocol, another 10 replicates were generated using 100-taxon trees using the r8s branch lengths. A program that creates simulated data sets following our protocols is available at http://pilin.cs.iastate.edu/public/spectral.

For each simulation replicate, we partitioned the simulated characters into two clusters using both the strict and fractional compatibility spectral partitioning methods. We examined how accurately each partitioning method clusters nucleotides based on their evolutionary history (the tree used to simulate the data). We also calculated the incongruence length difference (ILD) for each partitioned data set. The ILD is the difference in the lengths of the most parsimonious tree for the entire data set minus the sum of the lengths of the most parsimonious trees from each partition (Mickevich and Farris, 1981; Farris et al., 1994). The ILD measures the extent to which two data sets result in different trees, with a higher score indicating greater incongruence, or more conflict, among the data partitions.

**Empirical Examples**

We also examined the performance of the spectral partitioning methods using two empirical data sets in which partitioning has revealed evidence of conflicting phylogenetic signals. The first example is a seed plant data set from Sanderson et al. (2000) containing sequences from two plastid genes, psaA and psbB, from 19 taxa. This alignment contains 1165 parsimony-informative characters. In the original study, Sanderson et al. (2000) partitioned sites from the first and second codon positions from the third codon position sites, and separate parsimony analyses of the two partitions resulted in very different phylogenetic trees. The second example is a 13-locus, 25-taxon seed plant data set from Burleigh and Mathews, which contains 5600 parsimony-informative characters (2004). This study partitioned the data according to rate class assignments based on a discrete gamma distribution with eight categories. The sites in the two fastest rate classes (the RC78 sites) compose one partition, and the other sites (the RCslow sites) compose the other partition. Again, parsimony analyses of each partition support conflicting seed plant hypotheses (Burleigh and Mathews, 2004). We used both spectral partition methods on the parsimony informative characters from the two empirical data sets. We compared the ILD values from the spectral partitioning and the previously used partitioning methods. We also performed a partition homogeneity (or ILD) test to examine the significance of conflict among the partitions (Farris et al., 1994). For the ILD test, we examined only the parsimony informative sites, using 1000 randomly sampled partitions and a maximum parsimony heuristic with 10 random sequence addition replicates and TBR branch swapping as implemented in PAUP* (Swofford, 2002). We also compared the density, or the average pairwise compatibility among sites, of the different partitions. The density of a set of characters may be defined using either strict or fractional compatibility: The 0-1 density is simply the average pairwise compatibility from the 0-1, or strict compatibility, version of the PCCG. If the 0-1 density is 0.5, then on average each character is compatible with half of the other characters in the partition; if the 0-1 density is 1, then all characters are pairwise compatible, and the PCCG is a clique. For the similarity scoring version of the PCCG, the fractional compatibility density is the sum of all edge weights in a partition divided by the maximum possible total weight of the edges, which is $n(n - 1)/2$, where $n$ is the number of nodes (characters) in the partition. The fractional density is the average edge weight among characters in a partition.

**RESULTS**

**Simulation Experiments**

The simulation experiments demonstrate the ability of spectral partitioning methods to cluster sites based on their evolutionary history. The partitions largely corresponded to the characters' simulated topology (Tables 1, 2). In the 25-taxon simulations, the spectral partitioning using strict and fractional compatibility give similar results (Tables 1a, 2a). On average across all 10 replicates, 91.7% of the informative characters in a single strict compatibility partition come from one of the simulated topologies, and 92.1% of the informative characters in a single fractional compatibility partition come from one of the simulated topologies (Table 1a). The estimated parsimony scores obtained by analyzing each partition
Table 1. Results from partitioning simulated data sets. Each simulation replicate generated a 500-bp alignment for two randomly generated topologies. The results show the distribution of parsimony informative characters from each tree among partitions generated using spectral partitioning with strict compatibility and fractional compatibility.

<table>
<thead>
<tr>
<th>Replicate</th>
<th>Tree 1</th>
<th>Tree 2</th>
<th>Tree 1</th>
<th>Tree 2</th>
<th>Tree 1</th>
<th>Tree 2</th>
<th>Tree 1</th>
<th>Tree 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>124</td>
<td>6</td>
<td>19</td>
<td>112</td>
<td>124</td>
<td>6</td>
<td>19</td>
<td>112</td>
</tr>
<tr>
<td>2</td>
<td>151</td>
<td>13</td>
<td>16</td>
<td>149</td>
<td>152</td>
<td>11</td>
<td>15</td>
<td>151</td>
</tr>
<tr>
<td>3</td>
<td>145</td>
<td>10</td>
<td>12</td>
<td>142</td>
<td>151</td>
<td>6</td>
<td>16</td>
<td>146</td>
</tr>
<tr>
<td>4</td>
<td>146</td>
<td>6</td>
<td>16</td>
<td>151</td>
<td>141</td>
<td>7</td>
<td>21</td>
<td>150</td>
</tr>
<tr>
<td>5</td>
<td>104</td>
<td>15</td>
<td>17</td>
<td>130</td>
<td>108</td>
<td>19</td>
<td>13</td>
<td>126</td>
</tr>
<tr>
<td>6</td>
<td>135</td>
<td>16</td>
<td>16</td>
<td>123</td>
<td>135</td>
<td>14</td>
<td>16</td>
<td>125</td>
</tr>
<tr>
<td>7</td>
<td>144</td>
<td>14</td>
<td>7</td>
<td>134</td>
<td>146</td>
<td>13</td>
<td>5</td>
<td>135</td>
</tr>
<tr>
<td>8</td>
<td>145</td>
<td>10</td>
<td>9</td>
<td>144</td>
<td>143</td>
<td>8</td>
<td>11</td>
<td>146</td>
</tr>
<tr>
<td>9</td>
<td>129</td>
<td>3</td>
<td>6</td>
<td>146</td>
<td>129</td>
<td>6</td>
<td>16</td>
<td>145</td>
</tr>
<tr>
<td>10</td>
<td>134</td>
<td>24</td>
<td>8</td>
<td>128</td>
<td>134</td>
<td>2</td>
<td>8</td>
<td>128</td>
</tr>
</tbody>
</table>

Table 2. Parsimony scores resulting from different partitioning procedures. For each simulated data set, we calculated the cumulative parsimony scores that result from separate parsimony analyses of data from each partition. We compare three partitioning schemes: the true partition separating simulated data sets from the two random trees ("Tree Partition"), spectral partitioning using fractional compatibility, and spectral partitioning using strict compatibility. The parsimony scores in the column marked "No Partition" are based on a parsimony analysis of the unpartitioned data set.

<table>
<thead>
<tr>
<th>Replicate</th>
<th>Tree partition</th>
<th>Fractional compatibility</th>
<th>Strict compatibility</th>
<th>No partition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>697</td>
<td>701</td>
<td>704</td>
<td>919</td>
</tr>
<tr>
<td>2</td>
<td>1066</td>
<td>1071</td>
<td>1078</td>
<td>1402</td>
</tr>
<tr>
<td>3</td>
<td>858</td>
<td>858</td>
<td>862</td>
<td>1176</td>
</tr>
<tr>
<td>4</td>
<td>863</td>
<td>871</td>
<td>866</td>
<td>1126</td>
</tr>
<tr>
<td>5</td>
<td>739</td>
<td>746</td>
<td>746</td>
<td>932</td>
</tr>
<tr>
<td>6</td>
<td>829</td>
<td>829</td>
<td>834</td>
<td>1093</td>
</tr>
<tr>
<td>7</td>
<td>862</td>
<td>867</td>
<td>870</td>
<td>1148</td>
</tr>
<tr>
<td>8</td>
<td>907</td>
<td>909</td>
<td>913</td>
<td>1144</td>
</tr>
<tr>
<td>9</td>
<td>736</td>
<td>737</td>
<td>737</td>
<td>1003</td>
</tr>
<tr>
<td>10</td>
<td>829</td>
<td>832</td>
<td>833</td>
<td>1125</td>
</tr>
</tbody>
</table>

Table 2. Results from partitioning simulated data sets. Each simulation replicate generated a 500-bp alignment for two randomly generated topologies. The results show the distribution of parsimony informative characters from each tree among partitions generated using spectral partitioning with strict compatibility and fractional compatibility.

In the 100-taxon simulations with no branch length scaling, overall spectral partitioning using fractional compatibility outperforms spectral partitioning using strict compatibility (Tables 1b, 2b). On average across all 10 replicates, 85.3% of the informative characters in a single strict compatibility partition come from one simulated topology, and 97.3% of the informative characters in a single fractional compatibility partition come from one simulated topology (Table 1b). Also on average, the estimated parsimony scores obtained by analyzing each partition separately exceed the estimated parsimony scores obtained by analyzing the true partitions separately by 4.3% and 0.3% in the strict and fractional compatibility partitions respectively (Table 2b). The ILD for strict compatibility ranges from 88 to 4081 (average = 2922.5) and from 3384 to 4130 (average = 3703.6) for the fractional compatibility partitions (Table 2b). The spectral partitioning analyses of the simulated data sets took approximately 2 s CPU time for the 25-taxon data sets and 25 s CPU time for 100-taxon data sets under Linux on an Intel Pentium 4 3.0 GHz processor with 1 Gb memory.

Empirical Examples

In both the 2-locus data set of Sanderson et al. (2000) and the 13-locus data set of Burleigh and Mathews (2004), both spectral partitioning methods highlight more conflict within the data set than prior partitioning methods. In the psaA and psbB data set of Sanderson et al. (2000), both PCCG versions of the spectral partitioning produce partitions that reject the null hypothesis of no heterogeneity in the partition homogeneity test ($P < 0.001$ for both). The average 0-1 density and fractional density of characters using the two spectral partitioning methods are nearly equal, and they exceed the average densities from the codon partitioning method (Table 3a).

In the 13-locus data set of Burleigh and Mathews (2004), all three partitioning methods produce partitions that reject the null hypothesis of no heterogeneity in the partition heterogeneity test ($P < 0.001$ for all). Again, the average 0-1 density and fractional density of the clusters using the two spectral partitioning methods are nearly equal (Table 3b). Though the 0-1 and fractional density values are highest for the RCslow partition than separately exceed the estimated parsimony scores obtained by analyzing the true partitions separately by an average of 0.7% and 0.4% in the strict and fractional compatibility partitions, respectively (Table 2a). The ILD for strict compatibility ranges from 186 to 324 (average = 262.5) and from 186 to 331 (average = 264.7) for the fractional compatibility partitions (Table 2a).
Table 3. Density of partitions from empirical data sets. The 0-1 density ("0-1 Dens.") is the average pairwise compatibility of sites in a partition, and the fractional density ("Frac. dens.") is the average edge weight based on the fractional compatibility. We also present the average density of the clusters ("Average"), which is the sum of the density of each cluster multiplied by the proportion of edges in that cluster, $[(density_1 \times edges_1) + (density_2 \times edges_2)]/(edges_1 + edges_2)$. For each data set, the density is shown for the spectral partitions using strict compatibility ("Str. Comp.") and fractional compatibility ("Frac. Comp.") as well as for the previously used partitioning scheme.

<table>
<thead>
<tr>
<th>Partition</th>
<th>No. of chars.</th>
<th>0-1 Dens.</th>
<th>Frac. dens.</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. Sanderson et al. (2001) data.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Str. Comp. 1</td>
<td>682</td>
<td>0.386</td>
<td>0.639</td>
</tr>
<tr>
<td>Str. Comp. 2</td>
<td>483</td>
<td>0.577</td>
<td>0.766</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td>0.450</td>
<td>0.693</td>
</tr>
<tr>
<td>Frac. Comp. 1</td>
<td>630</td>
<td>0.365</td>
<td>0.624</td>
</tr>
<tr>
<td>Frac. Comp. 2</td>
<td>535</td>
<td>0.581</td>
<td>0.771</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td>0.455</td>
<td>0.703</td>
</tr>
<tr>
<td>codon12</td>
<td>289</td>
<td>0.389</td>
<td>0.621</td>
</tr>
<tr>
<td>codon3</td>
<td>876</td>
<td>0.409</td>
<td>0.660</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td>0.407</td>
<td>0.656</td>
</tr>
<tr>
<td>Str. Comp. 1</td>
<td>2140</td>
<td>0.672</td>
<td>0.823</td>
</tr>
<tr>
<td>Str. Comp. 2</td>
<td>3460</td>
<td>0.711</td>
<td>0.845</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td>0.700</td>
<td>0.839</td>
</tr>
<tr>
<td>Frac. Comp. 1</td>
<td>2011</td>
<td>0.693</td>
<td>0.841</td>
</tr>
<tr>
<td>Frac. Comp. 2</td>
<td>3589</td>
<td>0.700</td>
<td>0.837</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td>0.699</td>
<td>0.838</td>
</tr>
<tr>
<td>RCslow</td>
<td>2478</td>
<td>0.839</td>
<td>0.862</td>
</tr>
<tr>
<td>RC78</td>
<td>3122</td>
<td>0.475</td>
<td>0.769</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td>0.615</td>
<td>0.818</td>
</tr>
</tbody>
</table>

for any other partition, the average densities from the spectral methods exceed those from the rate class partitioning (Table 3b). All spectral partitioning analyses of empirical data were done under Linux on an Intel Pentium 4 3.0 GHz processor with 1 Gb memory. The spectral partitioning of the psaA and psbB data set of Sanderson et al. (2000) with 1165 parsimony-informative characters used approximately 33 s CPU-time, and the spectral partitioning of the 13-locus data set with 5600 parsimony-informative characters took just over 14 min (848 s) CPU-time.

**DISCUSSION**

This study describes how a fast and powerful partitioning method from computer science (e.g., Ding et al., 2001; Shi and Malik, 2000; Yu and Shi, 2003) can be used to cluster phylogenetic characters based on compatibility. Our simulation experiment demonstrates that partitioning characters based on our compatibility criterion performs well in identifying conflicting evolutionary histories within data sets. Furthermore, the empirical examples show that partitioning based on compatibility may help reveal previously undetected levels of phylogenetic conflict within data sets.

**Partitioning and Compatibility**

Compatibility and compatibility graphs have a long history in phylogenetics (see Meacham and Estabrook, 1985; Semple and Steel, 2003; Felsenstein, 2003). Although today compatibility criteria are seldom used to estimate phylogenies, compatibility has been used to weight characters for parsimony analyses (Penny and Hendy, 1985, 1986; Sharkey, 1989) and, more recently, to identify anomalous and fast-evolving characters in data sets (Meacham, 1994; Pisani, 2004). These methods are based on the notion that fast-evolving characters experience more homoplasy than slowly evolving characters and thus are pairwise compatible with fewer characters than slowly evolving characters.

The lack of pairwise compatibility among fast evolving characters suggests a potential weakness for partitioning data based on compatibility. If data sets contain characters from many taxa or only fast evolving characters, there may be little pairwise compatibility among characters, and the strict, or 0-1, compatibility graph may have a very low density. In such cases it may be difficult to partition characters based on strict compatibility. Spectral partitioning using fractional compatibility addresses this concern by relaxing the strict compatibility criterion. Characters are evaluated based on how close to they are to being compatible, and in theory, no two characters need be compatible for this method to work. If there is little pairwise compatibility among characters, we might expect spectral partitioning using fractional compatibility to perform better than spectral partitioning based on strict compatibility. We see this in the simulation experiment when spectral partitioning with fractional compatibility generally outperforms strict compatibility partitioning in the replicates with larger (100-taxon) trees and longer (unscaled) branch lengths (Table 1).

Unlike previous methods, the spectral partitioning methods use pairwise compatibility to estimate shared phylogenetic signal, not rates of evolution. They implicitly assume that incompatible characters are more likely to have conflicting phylogenetic signals than compatible characters. Perhaps most similar to our methods, Pisani (2004) used compatibility to identify the fastest evolving sites using a randomization test that compares a character’s observed overall pairwise compatibility with the overall pairwise compatibility that would be expected if the states of the character states were random. A justification of Pisani’s (2004) method is the idea that the fastest evolving sites are most likely to contribute to long-branch attraction effects, suggesting that differences in rate of evolution may produce major conflicts within the data set. Removing or otherwise down-weighting fast-evolving sites appears to be effective in some phylogenetic analyses (e.g., Philippe et al., 2000; Burleigh and Mathews, 2004; Dutilh et al., 2004; Pisani, 2004), but it is easy to imagine cases in which notable phylogenetic conflict exists among fast or slowly evolving sites. The spectral partitioning methods attempt to directly partition sites based on phylogenetic conflict without assumptions regarding the source of the conflict. However, this goal is not necessarily incompatible with Pisani’s (2004) method, and Pisani’s (2004) method may even enhance the performance of spectral partitioning. If a fast-evolving character is phylogenetically uninformative or
random, then it cannot be clustered in an informed manner based on compatibility. Thus, it may be useful to identify and remove such characters prior to spectral partitioning.

Performance of Spectral Partitioning

Our goal in partitioning phylogenetic data sets is to reveal conflicting evolutionary histories within the data set, whether due to different substitution processes or phylogenies. The spectral methods make it possible to estimate the optimal partitioning of a data set based on pairwise compatibility in a reasonable amount of time. Yet the effectiveness of our spectral methods in partitioning depends on how well the compatibility criterion reveals conflicting phylogenetic signals. In the simulated data sets, spectral partitioning using the compatibility criterion performs well in predicting which characters evolved from the same or different topologies (Table 1). Furthermore, if the ILD scores are indicative of the degree of conflict among partitions, spectral partitioning performs nearly as well at revealing conflict as knowing the actual underlying topologies. The combined parsimony score of the best spectral partitions never exceeds the estimate of the parsimony score of the true (tree) partition by more than 1.5%, and in one case, the estimate of parsimony score of the best spectral partition is better than the true (tree) partition (Table 1b). Because both topologies in the simulation study were generated randomly, the simulations represent an extreme case of phylogenetic conflict among sites in the alignment. We expect little pairwise compatibility among characters evolving in the different random trees, and therefore, we might expect spectral partitioning to perform well. Still, with any two topologies, at least some sites likely will be compatible with both topologies, and thus it may be extremely difficult to accurately partition every single character. We also caution against interpreting seemingly significant partitions as evidence of multiple phylogenies underlying the data. For example, in some data sets simulated from a single phylogeny with among-site rate variation, spectral partitioning may yield partitions that reject the ILD test (data not shown). This is consistent with the observation that rate variation among sites can produce apparent heterogeneity in the phylogenetic signal that is detected by the ILD test (e.g., Dolphin et al., 2000; Barker and Lutzoni, 2002).

In the empirical data sets, compatibility also appears to reveal higher levels of phylogenetic conflict than previously used partitioning methods. The much higher ILD scores from spectral partitioning coincide with higher average compatibility among sites in the spectral partitions. Though a phylogenetic analysis of the different codon position partitions in the psaA and psbB data set produces different phylogenetic trees (Sanderson et al., 2001), an ILD test does not detect significant heterogeneity in the codon partitions. However, the ILD test detects significant heterogeneity among the spectral partitions.

A $\chi^2$ test uncovers no significant correspondence between the codon partition and the spectral partitions (Snedecor and Cochran, 1989; Table 4). In the 13-locus data set, all partitions produce significant ILD tests, and a $\chi^2$ test demonstrates a significant correspondence between the rate class and spectral partitions (Table 4; $\chi^2 = 6.50$ for strict compatibility; $\chi^2 = 18.23$ fractional compatibility; $P < 0.01$ for both). The empirical examples demonstrate that previously used partitions will not necessarily fail to find conflict; however, they may not relate to the major conflicts in the data set.

Implementation of Spectral Partitioning

Although spectral partitioning based on compatibility appears to effectively highlight phylogenetic conflict within a data set, there are still questions regarding its performance and implementation. For example, the spectral partitioning methods do not require complete data sets (e.g., the data set of Burleigh and Mathews [2004] has ~35% missing data), but it is difficult to assess character compatibility with incomplete data. Thus, incomplete data may compromise the performance of the spectral partitioning methods. Also, spectral partitioning seeks to balance the size of partitions, and consequently, it may not perform well if phylogenetic conflicts are caused by a small percentage of the total characters.

An important issue in using spectral partitioning is to determine the optimal number of partitions. For simplicity, our examples use only two partitions; however, one can specify any number of partitions. There are numerous tests and methods to determine if data from different partitions represent significant phylogenetic conflict, some of which also can address the optimal number of partitions (e.g., Farris et al., 1994; Huelsenbeck and Bull, 1996; Mason-Gamer and Kellogg, 1996; Dolfini et al., 2000; Kauff and Lutzoni, 2002; Vogl et al., 2003; Ane and Sanderson, 2005; Brandley et al., 2005). In one example, Ane and Sanderson (2005) suggest an information theoretic approach to determine the optimal number of partitions. In this approach, the optimal partitioning scheme is the one that minimizes the descriptive complexity, or the compressed file size, of the data and the trees resulting from spectral partitioning.

### Table 4. Similarity of different partitioning schemes in empirical data sets

<table>
<thead>
<tr>
<th>Str. Comp.1</th>
<th>Str. Comp.2</th>
<th>Frac. Comp.1</th>
<th>Frac. Comp.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. Sanderson et al. (2001) data.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Frac. Comp. 1</td>
<td>578</td>
<td>52</td>
<td></td>
</tr>
<tr>
<td>Frac. Comp. 2</td>
<td>104</td>
<td>431</td>
<td></td>
</tr>
<tr>
<td>Codon12</td>
<td>174</td>
<td>115</td>
<td>148</td>
</tr>
<tr>
<td>Codon3</td>
<td>508</td>
<td>368</td>
<td>482</td>
</tr>
<tr>
<td>Frac. Comp. 1</td>
<td>1912</td>
<td>99</td>
<td></td>
</tr>
<tr>
<td>Frac. Comp. 2</td>
<td>228</td>
<td>3361</td>
<td></td>
</tr>
<tr>
<td>RCslow</td>
<td>993</td>
<td>1485</td>
<td>966</td>
</tr>
<tr>
<td>RC78</td>
<td>1147</td>
<td>1975</td>
<td>1045</td>
</tr>
</tbody>
</table>
from the partitions. The optimal number of partitions is approximately the largest number of partitions for which 
\[ \Delta L \geq (l - 1)n, \]
where \( \Delta L \) is the change in the sum of the lengths of the most parsimonious trees from each partition, \( l \) is the number of binary trees (or partitions), and \( n \) is the number of leaves in the binary tree (see Ané and Sanderson, 2004). In other words, one should add an additional partition if it reduces the sum of the lengths of the most parsimonious trees from each partition by a number that is at least as high as the number of leaves in the tree. This approach also demonstrates that our methods may provide good solutions for estimating the partition that globally minimizes the compression among all possible partitions.

There are many opinions on how one should treat conflict in phylogenetic analyses (e.g., Kluge, 1989; Bull et al., 1993; Chippendale and Wiens, 1994; De Queiroz et al., 1995; Huelsenbeck et al., 1996; Nixon and Carpenter, 1996; Brandley et al., 2005), and we do not attempt to resolve this issue. The ability of spectral partitioning to reveal previously undetected levels of conflict within a data set makes this discussion especially important. If the conflicting characters are interspersed within a linked region, it is highly unlikely that the conflict reflects different phylogenies. In such cases, it is popular to apply different substitution models to each partition (e.g., Nylander et al., 2004; Brandley et al., 2005). However, the spectral partitions are based on compatibility, not conflicts in substitution models or parameters. Therefore, although characters in different partitions may have different branch lengths, there is no reason to assume they will have different optimal substitution models or substitution parameter values.

The spectral partitioning method may be extended in several directions. Although we use spectral partitioning to cluster characters based on compatibility, the data could be clustered based on other characteristics, such as shared rates of evolution. Furthermore, in some cases it may be useful to incorporate biological constraints on the partitioning strategy. For example, we might not wish to combine characters from different loci into the same partition. Our examples demonstrate that spectral partitioning performs well in the absence of assumptions or regarding the characteristics of the data, and it likely can be improved by incorporating additional biological information.

ACKNOWLEDGEMENTS

We thank M. J. Sanderson and O. Eulenstein for discussion of spectral partitioning, and Davide Pisani, Barbara Holland, and Mike Steel for helpful comments on the manuscript. This study was funded by NSF grants EF-0334832 (D.C and D.E-B.) and 0431154 (G.G.B.) and by NESCent, NSF EF-0423641 (J.G.B.).

REFERENCES


Chippinda and Wiens, 1996; Nixon and Carpenter, 1996; Brandley et al., 2005), and we do not attempt to resolve this issue. The ability of spectral partitioning to reveal previously undetected levels of conflict within a data set makes this discussion especially important. If the conflicting characters are interspersed within a linked region, it is highly unlikely that the conflict reflects different phylogenies. In such cases, it is popular to apply different substitution models to each partition (e.g., Nylander et al., 2004; Brandley et al., 2005). However, the spectral partitions are based on compatibility, not conflicts in substitution models or parameters. Therefore, although characters in different partitions may have different branch lengths, there is no reason to assume they will have different optimal substitution models or substitution parameter values.

The spectral partitioning method may be extended in several directions. Although we use spectral partitioning to cluster characters based on compatibility, the data could be clustered based on other characteristics, such as shared rates of evolution. Furthermore, in some cases it may be useful to incorporate biological constraints on the partitioning strategy. For example, we might not wish to combine characters from different loci into the same partition. Our examples demonstrate that spectral partitioning performs well in the absence of assumptions or regarding the characteristics of the data, and it likely can be improved by incorporating additional biological information.


First submitted 21 December 2006; reviews returned 9 March 2007; final acceptance 2 May 2007

Associate Editor: Mike Steel

APPENDIX

This section contains the proof of Theorem 1, which is restated below.

**Theorem 1:** Let $p$ be the minimum number of edges that need to be removed from $H(c_1, c_2)$ to make it acyclic, for some pair of characters $c_1$ and $c_2$ with $r_1$ and $r_2$ states, respectively. Then, $p = \text{pars}(c_1, c_2) - (r_1 + r_2 - 2)$. To prove this result, we need two lemmas. We use the following notation. Let $G$ be a graph. Then, $V(G)$ and $E(G)$ denote the vertex and edge sets of $G$, and $\text{deg}(G)$ denote the vertex and edges of $G$.

**Lemma 1:** If $H(c_1, c_2)$ is connected, $\text{pars}(c_1, c_2) = |H(c_1, c_2)| - 1$.

**Proof:** We first show that $\text{pars}(c_1, c_2) \geq |H(c_1, c_2)| - 1$. Let $P$ be any phylogeny for characters $c_1$ and $c_2$. Since $P$ contains $H(c_1, c_2)$ distinct state pairs, and each state pair induces a unique subtree, there must be a total of at least $|H(c_1, c_2)| - 1$ character state changes for characters $c_1$ and $c_2$ in $P$; that is, $\text{pars}(c_1, c_2) \geq |H(c_1, c_2)| - 1$. Next, we argue that $\text{pars}(c_1, c_2) \leq |H(c_1, c_2)| - 1$. Since $H(c_1, c_2)$ is connected, it has a spanning tree $T$. Let $E' = |H(c_1, c_2)| - E(T)$. The edges in $T$ correspond to a subset of the taxa $S$ such that characters $c_1$ and $c_2$ restricted to $S$ are compatible. These characters therefore have a phylogenetic tree $P'$ with $\text{deg}(T) - 2 = e(T) - 1$ state changes.

We can extend subtree $P'$ to a tree $P$ containing all of the taxa—including those appearing in $E'$. As follows. For each edge $e < y < z$, we find a set of taxa in $P'$ such that either all of them have state pair $x < y$, or all of them have state pair $y > z$. These taxa must all be connected to some node $v$ of $P'$ whose parent is, say, $u$. We then replace edge $(u, v)$ in $T$ by edges $(u, w)$ and $(w, v)$, add an edge $(w, w')$, and connect all taxata with states $x < y < z$. Continuing in this fashion until all edges in $E'$ have been considered, we obtain a tree $P$ that requires at most $|E(T)| - 1 + |E'| = |H(c_1, c_2)| - 1$ state changes. Hence, $\text{pars}(c_1, c_2) \leq |H(c_1, c_2)| - 1$. 

First submitted 21 December 2006; reviews returned 9 March 2007; final acceptance 2 May 2007

Associate Editor: Mike Steel
Lemma 2: For any pair of characters \( c_1 \) and \( c_2 \), \( \text{pars}(c_1, c_2) = e[H(c_1, c_2)] + k - 2 \), where \( k \geq 1 \) is the number of connected components in \( H(c_1, c_2) \).

Proof: Let \( C_1, C_2, \ldots, C_k \) be the connected components of \( H(c_1, c_2) \). By Lemma 1, each \( C_i \) has a phylogeny with \( |E(C_i)| - 1 \) state changes and no fewer changes are possible.

Consider any two components \( C_i \) and \( C_j \), where \( 1 < i < j \leq k \). Any state pair \( <x_i, y_i> \) in \( C_i \) and \( <x_j, y_j> \) in \( C_j \) must satisfy \( x_i \neq x_j \) and \( y_i \neq y_j \) (otherwise, \( C_i \) and \( C_j \) are connected). Thus, two state changes are required in order to connect any two phylogenies from different connected components; and totally \( 2(k - 1) \) state changes are required for connecting the \( k \) phylogenies to form a tree for all the taxa. Hence,

\[
\text{pars}(c_1, c_2) \geq \sum_{i=1}^{k} |E(C_i)| - 1 + 2(k - 1) = \sum_{i=1}^{k} |E(C_i)| - k + 2k - 2 = e[H(c_1, c_2)] + k - 2.
\]

On the other hand, we can construct a phylogeny \( P \) of \( c_1 \) and \( c_2 \) with \( e[H(c_1, c_2)] + k - 2 \) state changes. First, we construct phylogenies for each of the \( C_i \).s. Then, we create a new root \( r \) that links to each root \( r_i \) of each such phylogeny, and assign \( r \) the same states as \( r_i \). Any edge \( \{r, r_i\} \) where \( 2 < i < k \), counts for 2 changes. Thus, the parsimony score of \( P \) is

\[
\sum_{i=1}^{k} |E(C_i)| - k + 2k - 2 = e[H(c_1, c_2)] + k - 2.
\]

Proof of Theorem 1: Let \( E' \) be a minimum set of edges whose removal from \( H(c_1, c_2) \) yields an acyclic subgraph; i.e., a spanning forest \( R \) of \( H(c_1, c_2) \). Then,

\[
e(R) = v(R) - k = r_1 + r_2 - k,
\]

where the first equality follows from the properties of forests, and the second is because \( v(R) = v[H(c_1, c_2)] = r_1 + r_2 \). Now,

\[
p = |E'|
\]

\[
e[H(c_1, c_2)] - e(R)
\]

\[
= \text{pars}(c_1, c_2) - k + 2 - (r_1 + r_2 - k) \quad \text{(by Lemma 2)}
\]

\[
= \text{pars}(c_1, c_2) - (r_1 + r_2 - 2).
\]