22 Jan 2004

Identifying Character Non-Independence in Phylogenetic Data using Data Mining Techniques

Jennifer Leopold
Missouri University of Science and Technology, leopoldj@mst.edu

Anne M. Maglia
Missouri University of Science and Technology

M. Thakur

B. Patel

et. al. For a complete list of authors, see https://scholarsmine.mst.edu/comsci_facwork/258

Identifying Character Non-Independence in Phylogenetic Data Using Data Mining Techniques

Anne M. Maglia[1,3] **Jennifer L. Leopold**[2,4] **Venkat Ram Ghatti**[2,5]

[1]Department of Biological Sciences
[2]Department of Computer Science
University of Missouri-Rolla
Rolla, MO 65409, USA

[3]magliaa@umr.edu, [4]leopoldj@umr.edu, [5]vgk99@umr.edu

Abstract

Undiscovered relationships in a data set may confound analyses, particularly those that assume data independence. Such problems occur when characters used for phylogenetic analyses are not independent of one another. A main assumption of phylogenetic inference methods such as maximum likelihood and parsimony is that each character serves as an independent hypothesis of evolution. When this assumption is violated, the resulting phylogeny may not reflect true evolutionary history. Therefore, it is imperative that character non-independence be identified prior to phylogenetic analyses. To identify dependencies between phylogenetic characters, we applied three data mining techniques: 1) Bayesian networks, 2) decision tree induction, and 3) rule induction from coverings. We briefly discuss the main ideas behind each strategy, show how each technique performs on a small sample data set, and apply each method to an existing phylogenetic data set. We discuss the interestingness of the results of each method, and show that, although each method has its own strengths and weaknesses, rule induction from coverings presents the most useful solution for determining dependencies among phylogenetic data at this time.

Keywords: Data mining, character independence, phylogenetic data, machine learning.

1 Introduction

Undiscovered relationships of data in a data set may confound analyses, particularly those that assume data independence. In biological data, one such problem occurs when characters used for phylogenetic analyses are non-independent. A main assumption of phylogenetic inference methods such as maximum likelihood and parsimony is that each character serves as an independent hypothesis of evolution (Felsenstein, 1973; Kluge and Farris 1969). When this assumption is violated, correlated or non-independent characters are effectively overweighed in analyses (Chippendale and Wiens 1994), and the resulting phylogeny does not reflect the true evolutionary history. Therefore, it is imperative that character non-independence is identified prior to phylogenetic analyses.

There are several ways that characters or attributes can be non-independent. One attribute can depend upon another, or a set of attributes can be co-dependent. Attributes can also be correlated, wherein they are not dependent upon one another, but share a set of dependencies with other characteristics. In the context of phylogenetics, we expect characters that reflect homology (= similarity due to common ancestry) to share a set of dependencies that reflect the true evolutionary history of the group. This sort of dependency is referred to as *phylogenetic dependence* or *phylogenetic autocorrelation*. This type of dependency is the basis of all methods of phylogenetic analysis and is the expected demonstration of synapomorphy (= homologous characters that unite groups). However, if a set of non-independent characters reflects parallel or convergent events, their presence may lead to the wrong reconstruction of evolutionary history.

For example, let us suppose that a single evolutionary event gives rise to several seemingly unrelated characteristics. If those characteristics are each coded as an independent hypothesis of evolution (i.e., separate transformation series in the analyses), the resulting tree could be biased toward that evolutionary event. If that event was in fact merely a single convergence, the presence of several instances reflecting the event in the data set may outweigh the true homology in the data set, and thus, the analysis will not reflect the true evolutionary history of the group. As an oversimplified example, imagine the problems in resolving relationships that would result if one were to independently code all of the different morphological characteristics that a dolphin (a mammal) and a shark (a fish) share because they both have aquatic lifestyles (e.g., pectoral fins, anal fins, etc.). Recognizing character non-independence in a small morphological data set is difficult enough, let alone attempting to determine character non-independence in a large molecular data set (often with thousands of transformation series).

Although most systematists recognize the problems with character dependence (e.g., Maglia 1998; McKracken et al. 1999), few quantitative attempts have been made to identify non-independence of phylogenetic characters. Of those methods available, nearly all examine phylogenetic independence/autocorrelation of characters *after* phylogenetic analyses are conducted (e.g., Cheverud et al., 1985; Felsenstein, 1985; Maddison, 1990; Abouheif, 1999). However, to conduct these tests requires

a model phylogeny upon which to test hypotheses of independence, and thus, requires the assumption of independence of characters used to generate the phylogeny. Obviously, a better approach would be to test for character non-independence prior to conducting phylogenetic analyses.

O'Keefe and Wagner (2001) developed a "pre-tree" approach for visualizing suites of correlated characters using character compatibility. By calculating an association matrix between characters in a data set, and subsequently conducting eigenvector analyses on the matrix, they were able to identify characters with similar patterns of compatibility, and showed that some suites of characters were more correlated with one another than expected by chance alone.

Although the methods used by O'Keefe and Wagner provide an initial means to examine non-independence among phylogenetic characters, the results are limited in the understanding of relationships they provide. From these methods, it is only possible to identify characters that are correlated. Dependency or co-dependency relationships may exist in the data set that can not be determined using their methods. Furthermore, if dependencies could be determined, knowing the direction of the dependence relationships could be extremely helpful in furthering the understanding of the biological connections among the data in the data set. To understand the true nature of character non-independence in a data set, we must be able to identify correlation, dependency, and co-dependency.

Fortunately, the methodologies of data mining are dedicated to finding and describing structural patterns (such as dependency) in data (Witten and Frank). Therefore, applying additional data mining methods to the problem of non-independence in phylogenetic data may provide alternative and/or additional interpretations of the relationships of the characters in a data set. Because different problems yield to different techniques, it is never clear which techniques are suitable for a given situation (Han and Kamber). Therefore, to identify dependencies between phylogenetic characters, we applied three different data mining techniques: 1) Bayesian networks, 2) decision tree induction, and 3) rule induction from coverings. We briefly discuss the main ideas behind each strategy and show how each technique performs on a small sample data set. We then apply each method to the Wilkinson (1997) data set analyzed by O'Keefe and Wagner (2001) and compare our results to the results of O'Keefe and Wagner's (2001) statistical analysis. Finally, we comment on the interestingness of each method relative to the problem presented above.

2 Data

We analyzed several data sets to compare the various methods discussed here, including the Wilkinson (1997) data set reported in O'Keefe and Wagner (2001), a phylogenetic data set of Maglia (1998), and several small sample data sets found in Busse (67:table 3.10). For ease of discussion of the three methods, we will focus our initial comparisons on analyses of a simple fabricated data set shown in Table 1. Note that Characters B, C, and G have equivalent codings and could represent non-independency (as could Characters A, F, and J).

3 Description of Data Mining Methods

3.1 Bayesian Belief Networks

A Bayesian belief network is a graphical depiction of causal relationships between attributes. It is represented as a directed acyclic graph, where each node represents

	Characters									
taxa	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**
i	0	1	0	0	0	0	1	1	0	1
ii	0	0	1	1	1	0	0	1	0	1
iii	1	0	1	2	1	1	0	1	0	0
iv	1	0	1	2	2	1	0	1	1	0
v	1	1	0	2	2	1	1	0	1	0
vi	1	1	0	0	2	1	1	0	1	0

Table 1: Data set used in comparisons of three data mining methods

an attribute and each edge represents a probabilistic dependence between the two attributes (nodes) that are the endpoints of the edge. These dependencies are quantified using Bayes' theorem, which states:

$$P(H \mid X) = \frac{P(X \mid H)\, P(H)}{P(X)}$$

where P(H | X) is the probability of X given H. A node (representing a character from the data set) is considered to be conditionally independent of its nondescendant (attribute) nodes in the graph (Han and Kamber 2001). Thus, Bayesian networks should be appropriate for testing hypotheses of character dependencies in phylogenetic data sets.

We used *BK2* (http://biodi.sdsc.edu/bk2_home.html), a Bayesian network program developed by David Stockwell at the San Diego Supercomputer Center, to analyze the sample data set in Table 1. A subset of results is presented in Figure 1.

The network in Figure 1a resulted from running *BK2* with the maximum of parents per node set at 5, the search method set to all combinations, and characters specified in the following order: B,C,D,E,F,G,H,I,J,A (where A is the root). Figure 1b. resulted from an analysis with the same settings, except the order of the characters was: C,D,E,F,G,H,I,J,B,A (where A is the root).

The networks shown in Figure 1 provide some understanding of the dependency relationships in the sample character data. For example, in both networks, Characters B, C, and G have some dependency relationships among one another (i.e., B and C are dependent on G in Fig. 1a; C and G are dependent on B in Fig. 1b), an expected result given the codings in Table 1. Independence can also be ascertained from these networks—both networks show that Characters D and I are conditionally independent (given Character A) and

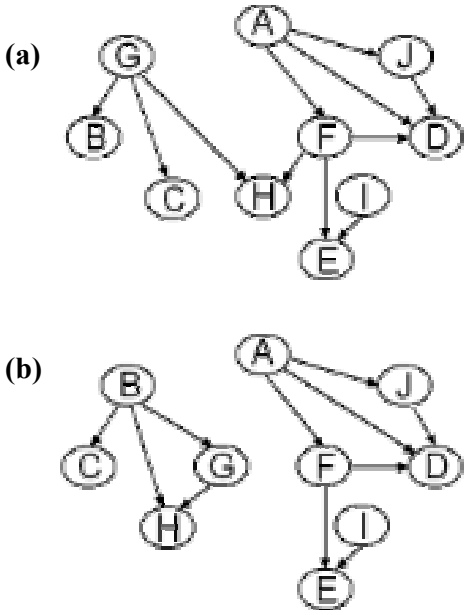

Figure 1: Bayesian networks of data set in Table 1. (See text for details)

that Characters C and H are conditionally independent (given A, G, and B). However, Figure 1a shows that Character H is *not* conditionally independent of A and F, whereas in Figure 1b H *is* conditionally independent of A and F.

Unfortunately, as shown by this example, the very nature of the Bayesian network model limits is usefulness for determining character dependencies independent of phylogenies. To build a Bayesian network, one must have some *a priori* knowledge of at least some of the relationships of the data and must specify a "starting point" (e.g., the root of the graph) from which all character relationships are built. Bayesian networks constructed from the same data set can yield different networks depending on the order that the characters are listed in the data set. This can occur even if the same attribute is always designated as the root, as is the case in Figure 1. For k attributes, there are k! permutations (i.e., orderings) of the attributes. Therefore, applying Bayesian networks to the problem of character non-independence necessitates constructing networks for every possible combination of characters.

Chickering (2002) proposed the concept of *equivalent* Bayesian networks to identify networks that may differ structurally, but still imply the same set of independence statements. One could imagine using this concept to apply a heuristic to reduce the number of perturbations necessary. Unfortunately, because different networks can be generated using the same root but with other characters in different order, these different networks are not always equivalent in their dependency relationships. Thus applying a heuristic could result in partial or even contradictory information.

3.2 Decision Tree Induction

Decision tree induction is a classic machine learning technique where one or more attributes are identified as "decisions" or "classifiers", and a flow-chart-like tree structure is generated to identify which combinations of attribute values result in which "decision" values. Each internal node in the decision tree denotes a test on an attribute, each branch represents an outcome of the test, and leaf nodes represent decisions or classes (Han and Kamber 2001). The particular decision tree algorithm utilized in our investigation is ID3 (Quinlan 1986), implemented in the well-known program C4.5 (Quinlan 1993).

The basic strategy of this algorithm is given in Han and Kamber (2001) as follows. The tree starts as a single node representing the entities in the data set. If the entities at this node all have the same value for the decision attribute(s), the node becomes a leaf. Otherwise, an entropy-based measure known as *information gain* is used as a heuristic for selecting the attribute that will best separate the samples into individual classes. The objective of this heuristic is to minimize the information needed to partition (i.e., classify) the entities in the data set. The selected attribute thus becomes a test at the current junction of the tree. A branch is created for each known value of this attribute, and the entities are partitioned accordingly.

The partitioning continues recursively until any one of the following conditions is satisfied: (i) all samples for a given node belong to the same class, (ii) there are no remaining attributes on which the entities can be further partitioned, or (iii) there are no entities for a branch corresponding to the assignment of a particular known value for that attribute. Conditions (ii) and (iii) may necessitate the application of a *majority voting* strategy whereby a node is made into a leaf and labeled with the decision value that occurs in the majority (but not necessarily *all*) of the entities partitioned at that node.

Rules for determining the values of the decision attributes can be formed from the paths from the root of the tree to the various leaf nodes. These rules also provide information about the dependencies between attributes and decisions.

We used Ross Quinlan's latest version of the C4.5 Algorithm, *See5* (available at the RuleQuest website: http://www.rulequest.com), to analyze the sample data set. We ran 10 analyses, each with a different character assuming the role of the decision class. For all analyses, the rule-sets option was chosen. Table 2 shows a summary of the results.

The results of the C4.5 analysis give some understanding of the dependency relationships of the data in Table 1. For example, when Character B is the decision class (i.e., the character of interest), C4.5 identified rules involving Character C (e.g., if C is 1, then B is 0). This result was expected given the codings in Table 1. Similarly, the analysis identified the dependency of Character G on Character B. However, note that this method did not identify rules that included the relationships G → C or C → G. (Similarly, it did not identify the relationships between J and F that are obviously present in Table 1). In this very small data set, it is easy to see the dependency relationship between C and G, but in a larger, more

realistic data set, this relationship and others would likely be lost.

decision	rules
A	Rule 1: J = 1 → A = 0 (0.75)
	Rule 2: J = 0 → A = 1 (0.83)
B	Rule 1: C = 1 → B = 0 (0.80)
	Rule 2: C = 0 → B = 1 (0.80)
C	Rule 1: B = 1 → C = 0 (0.80)
	Rule 2: B = 0 → C = 1 (0.80)
D	None
E	Rule 1: I = 0 → E = 1 (0.60)
	Rule 2: I = 1 → E = 2 (0.80)
F	Rule 1: A = 0 → F= 0 (0.75)
	Rule 2: A = 1 → F= 1 (0.83)
G	Rule 1: B = 0 → G = 0 (0.80)
	Rule 2: B = 1 → G = 1 (0.80)
H	Rule 1: B = 1 → H = 0 (0.60)
	Rule 2: B = 0 → H = 1 (0.80)
I	Rule 1: E = 1 → I = 0 (0.75)
	Rule 2: E = 2 → I =1 (0.800)
J	Rule 1: A = 1 → J = 0 (0.833)
	Rule 2: A = 0 → J = 1 (0.750)

Table 2. Results of C4.5 analysis of Table 1 showing rules generated for each decision attribute. Values in parentheses indicate error rate for rules.

The reason for the loss of information is that decision tree induction is constructed to optimize information gain, and selects a tree with the maximum number of classes at each node. Therefore, at a given node, a decision tree induction algorithm might not report other trees that could determine the decision attribute(s) in terms of other possible combinations of attributes (such as C → G in the sample data set). Thus, the amount of attribute-dependency information reported is limited.

A second concern with the use of decision tree induction is that because decision trees must select one attribute to split on first, a decision tree can be much larger than an equivalent set of rules (Witten and Frank 2000). This could be a problem in phylogenetic data sets because of their potential large size (e.g., thousands of characters).

A final area of concern is that the rules generated from a decision tree may not be "perfect" or "correct." In other words, the rule may pertain to only some of the rows (i.e., taxa in a phylogenetic data set), and an alternative rule may apply to other rows. For example, the rules for decision attribute E in Table 2 state that I = 0 → E = 1. However, examining the codings in Table 1, we see that it is also true that I = 0 → E = 0. This can occur for a number of reasons including the application of the majority voting strategy mentioned above, anomalies that can occur in the tree from outliers or noise in the data set, and the effects of various tree pruning techniques that may be applied to simplify the tree. Programs such as C4.5 report the error rate for the application of each rule to the given data set (such as the values in parentheses in Table 2) and thereby quantify the confidence with which each rule can be applied.

Because the rules generated are not 100% correct, applying such algorithms to phylogenetic data could result in the identification of dependencies that not apply to all of the taxa in the data set. Therefore, it is possible that algorithms such as C4.5 would report only one rule for character combinations that, in reality, show all possible combinations (e.g., 0 → 0, 0 → 1, 1 → 0, 1 → 1). Characters such as these can not be dependent because the fact that every combination of states is present proves that they are free to evolve independently from one another. Therefore, applying decision tree induction to inferring relationships of phylogenetic characters can also result in identification of false dependencies.

3.3 Rule Induction from Coverings

Decision tree algorithms such as those used in C4.5 are based on a divide-and-conquer approach, successively finding an attribute to split on that best separates the partitions of entities determined thus far. An alternative approach is to take each possible decision and determine a minimal set of attributes that can determine or "cover" all instances of it (Witten and Frank 2000).

RICO (Rule Induction from COverings; available at: http://web.umr.edu/~bioinf/biominer/) is a Java implementation of an algorithm given in Grzymala-Busse (1991) for finding all possible coverings for a given data set. The approach taken in this algorithm uses some of the concepts introduced by (Pawlak 1984) for rough sets, a classification scheme based on approximations of partitions of entities in a data set.

For this covering algorithm, if S is a set of attributes and R is a set of "decision" attributes, a covering P of R in S can be found if the following three conditions are satisfied:

(i) P is a subset of S;

(ii) (ii) R *depends* on P. That is, if a pair of entities x and y cannot be distinguished by means of attributes from P, then x and y also cannot be distinguished by means of attributes from R. If this is true, then entities x and y are said to be *indiscernible* by P (and, hence, R), denoted $x \sim_P y$. An *indiscernibility relation* ~P is such a partition over all entities in the data set;

(iii) (iii) P is minimal.

Condition (ii) is true if and only if an equivalent condition ≤, known as the *attribute dependency inequality*, holds for P* and R*, the partitions of all attributes and decisions generated by P and R, respectively, where, for a set of attributes A:

$$A^* = \pi_{a \in A} \sim \{a\}^*.$$

The inequality P* ≤ R* holds if and only if for each block B of P*, there exists a block B′ of R* such that B is a subset of B′.

Once a covering is determined, it is a straightforward process to induce rules from it. Although any single covering may be a basis for computing a rule set that describes the entire data set, it can be even more useful to identify *all* possible coverings. The more extensive rule set that results not only facilitates classification in terms of different combinations of attributes (an advantage when the values for some attributes may in practice be

more "expensive" for the scientist to obtain), but also defines a set of *essential attributes* (Grzymala-Busse 1991) for each decision; that is, attributes that occur in at least one covering for a decision, and thus can play some role in determining that decision. Similarly, knowing all coverings for a decision identifies *non-essential attributes*; attributes that are in no way involved with determining that decision. The concept of essential attributes can be further qualified as *highly useful attributes*; that is, essential attributes that are involved in a large number of coverings, and/or can be used to classify a large number of the entities in the data set. Of course, it would be up to the data expert to specify what is considered "large."

Finding all coverings can be computationally expensive since, in theory, each possible subset of attributes must be tested as a potential covering (unless that subset is a superset of a covering that has already been identified). For a data set of k attributes, there are 2^k different subsets. In a morphological data set, this may be 50-80 characters, but in the typical molecular data set this may be closer to 2,000 characters. For phylogenetic data sets, some constraints can be applied to the covering algorithm to reduce the execution time. For example, the cardinality of the candidate subsets could be limited to a small number (e.g., 3 or 4) because most systematists analyzing the character data will likely find it difficult to conceptualize combinations of many characters to determine the state of the character of interest. Furthermore, it is reasonable to limit the number of rules reported to only those that cover a certain number of entities in the data set. For example, a rule that only applies to one taxon is far less phylogenetically informative than a rule that applies to 75% of the taxa. Again, it is up to the expert to identify those limits.

Table 3 shows all coverings resulting from a *RICO* analysis of the data set in Table 1. Note that, as expected from the codings in Table 1, the coverings indicate that

decision	coverings
A	{F}, {J}, {E, D}, {H, D}, {I, D}
B	{C}, {G}, {H, D}, {H, E}
C	{B}, {G}, {H, D}, {H, E}
D	None
E	{I, B}, {I, C}, {I, D}, {I, G}
F	{A}, {J}, {E, D}, {H, D}, {I, D}
G	{B}, {C}, {H, D}, {H, E}
H	{B, A}, {C, A}, {G, A}, {E, B}, {F, B}, {I, B}, {J, B}, {E, C}, {F, C}, {I, C}, {J, C}, {G, E}, {G, F}, {I, G}, {J, G}
I	{E}
J	{A}, {F}, {E, D}, {H, D}, {I, D}

Table 3. All coverings resulting from *RICO* analysis of data in Table 1.

there are dependency relationships among A, J, and F (e.g., A is dependent upon F and A is dependent upon J; F is dependent upon A and F is dependent upon J, etc.), as well as B, C, and G. Interestingly, *RICO* identified all of the obvious potentially non-independent characters in the data set, but also identified several other additional dependencies (e.g., A is dependent upon a combination of E and D).

To further evaluate the dependency relationships among the characters, we examined the rules produced from the coverings in Table 3. To reduce the volume of data reported here (given that *RICO* identified 172 rules from the coverings in Table 3), we will discuss only those rules identified for Characters A as the decision attribute (Table 4). Note that the rules show a one-to-one dependency of Character A on Character F and on Character J. Although this information is also conveyed in the coverings in Table 3, examining the rules in Table 4 gives a more specific view of the relationships. In other words, we can say that with 100% accuracy, in this data set, if we know the state of Character J (e.g., 0), we can know the state of Character A (e.g., 1).

decision	rules
A	Rule set 1: F = 0 → A = 0
	F = 1 → A = 1
	Rule set 2: J = 0 → A = 1
	J = 1 → A = 0
	Rule set 3: E = 0 & D = 0 → A = 0
	E = 1 & D = 1 → A = 0
	E = 1 & D = 2 → A = 1
	E = 2 & D = 0 → A = 1
	E = 2 & D = 2 → A = 1
	Rule set 4: H = 0 & D = 0 → A = 1
	H = 0 & D = 2 → A = 1
	H = 1 & D = 0 → A = 0
	H = 1 & D = 1 → A = 0
	H = 1 & D = 2 → A = 1
	Rule set 5: I = 0 & D = 0 → A = 0
	I = 0 & D = 1 → A = 0
	I = 0 & D = 2 → A = 1
	I = 1 & D = 0 → A = 1
	I = 1 & D = 2 → A = 1

Table 4. Rules produced from the coverings of Character A as the decision attribute in the RICO analysis of the data in Table 1.

It is important to remember that *RICO* produces rules from all coverings, meaning that the combined set of rules describes the entire data set. Therefore, we can be confident that, unlike the rules produced by C4.5, we are not overlooking possible character combinations (thus resulting in misidentified dependencies). However, this can result in very large sets of rules (such as those in Table 4). Interestingly, *RICO* identified rules for Character A that include Characters E and D, H and D, and I and D. Note that some combinations of character states result in a similar state in the decision attribute (e.g., E = 0 & D = 0 → A = 0 and E = 1 & D = 1 → A = 0). Although *RICO* identified a dependency relationship among these characters, the fact that there are multiple combinations of characters associated with the same state in the decision attribute could indicate that rules such as these (with multiple combinations identified) may not necessarily reflect phylogenetic character non-independence. Rules such as those in Rule set 1 and Rule set 2 in Table 4 in which there is only one character state combination for each decision attribute state clearly

reflect character non-independence. However, the expert user must carefully examine rules with multiple combinations data to insure that all the rules produced do in fact represent phylogenetic character non-independence.

4 Performance on Real Data Set

In this section we compare the performance of each of the data mining methods above to the statistical analysis presented by O'Keefe and Wagner (2001). Unfortunately, as discussed by O'Keefe and Wagner (2001:672), their methods do not have the power to determine character independence on data sets of fewer than 20 characters. Therefore, rather than making comparisons using our sample data set (Table 1), we will compare the methods to the results obtained by them when they analyzed the phylogenetic data set of Wilkinson (1997). This data set was chosen because Wilkinson (1997) suspected that there were several suites of correlated characters in the data.

Here we present a simplified overview of the methods presented by O'Keefe and Wagner (2001); see their original text for a more thorough description of the methods. First, a dissimilarity matrix is generated using the following steps. A character-by-character pairwise compatibility matrix is created, wherein 1 indicates characters *i* and *j* are compatible (e.g., they do not show all possible character state combinations; O'Keefe and Wagner, 2001), and 0 indicates that they are incompatible. The resulting matrix is converted to a mutual compatibility matrix in which each value $m_{i,j}$ represents the number of characters with which both *i* and *j* are compatible. A dissimilarity matrix is then constructed a wherein $d_{i,j} = 1\text{-}m_{i,j}/(n\text{-}2)$ where n is the

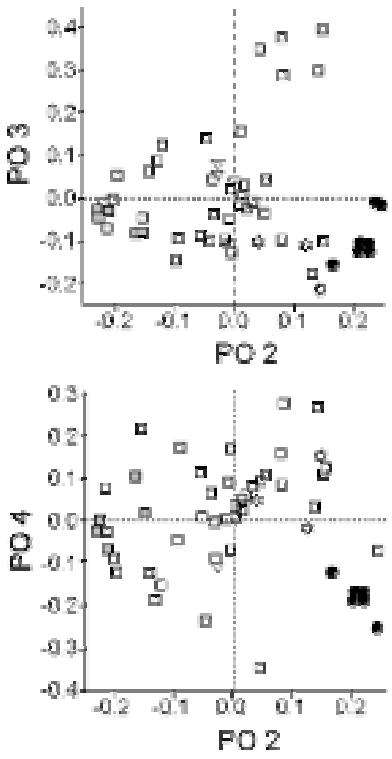

Figure 3. Results of Principle Cordinates analysis of Wilkinson's (1997) data set presented by O'Keefe and Wagner (2001). Shaded shapes identify correlated suites of characters. See text for further description. (Redrawn from O'Keefe and Wagner 2001.)

number of characters. The matrix is Gower transformed (Gower 1966), and decomposed for corresponding eigenvectors using principal coordinates analysis. The eigenvectors (PO) are plotted to reveal mutual compatibilities and separation of correlated characters, specifically those below the first PO. The results are compared to Monte Carlo simulations to determine if they are statistically significant from those expected at random. Figure 3 shows the results of O'Keefe and Wagner's (2001) analysis of Wilkinson's (1997) data set of 78 morphological characters. They were able to identify two separate suites of correlated characters, shown here in Figure 3 by the gray squares and black circles.

The largest correlated suite of characters O'Keefe and Wagner (2001) were able to identify included Characters E1.1, E1.3, E1.5, E1.6, E3, R43, and T57 (original character numbering). These include characters pertaining to specific muscles of the eye (E1.1, E1.3, E1.5, E1.6), the optic nerve (E3), the process of metamorphosis (T43), and the teeth (T57). The second largest partition they uncovered included Characters T4, T5, T6, T16, T31, and

E1.1	E1.3	E1.5	E1.6	E3	T43	T57
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
?	?	?	?	?	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
1	1	1	1	1	1	1
1	?	?	1	1	1	1
?	?	?	?	?	1	1
?	0	?	0	1	1	1
0	1	1	0	1	1	1
1	0	1	1	1	1	0
?	?	?	?	?	1	1
1	?	1	1	1	?	1
1	?	1	1	1	?	1
0	0	0	0	1	1	0
?	?	?	?	?	1	0
0	0	0	0	0	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
0	?	?	?	1	1	0
0	?	?	?	1	0	0

Table 4. Character codings for first suite of correlated characters reported in OKeefe and Wagner (2001). Question marks indicate missing data. See text for character descriptions.

T56. These are characters describing bones of the cranium (T4, T5, T6, T16) the cloaca (T31) and a cranial muscle (T56). These results were consistent with the suites of dependent characters identified by Wilkinson

(1997) in his original paper. Character codings for the first set of correlated characters are presented in Table 4.

4.1 Bayesian Belief Networks

Because of the size of Wilkinson's (1997) data set, it was not practical to run Bayesian analyses for all of the possible permutations of characters. Therefore, we ran a Bayesian analysis of the characters in Table 4 as well as three additional characters randomly chosen from Wilkinson's (1997) data set. The resulting network is depicted in Figure 2. The network indicates that there are several dependency relationships between the characters listed in Table 4. However, the randomly chosen characters (A1, H1.4, and T20a) cluster out apart from the other characters. This is consistent with the findings of

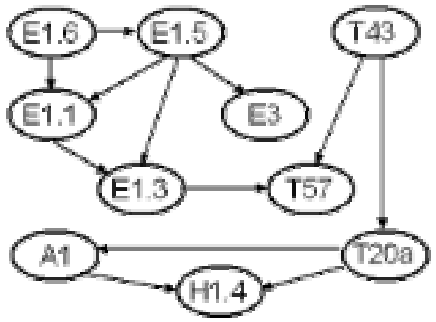

Figure 2. Sample Bayesian belief network of characters listed in Table 4 and three other randomly selected characters from Wilkinson (1997).

O'Keefe and Wagner (2001), in that the Bayesian network identifies dependency relationships among the correlated characters in Table 4, but not the non-correlated characters. Therefore, this would suggest that the Bayesian analysis can identify some relationships within the data set.

It is important to remember that the characters included in the analysis and the order of those characters will influence the resulting belief net and dependency relationships depicted. Therefore, it is impractical to use Bayesian networks to identify all of the dependency relationships in the data set. But, this method could be used to validate/refute relationships among characters that have been identified by an expert as potentially correlated or dependent.

4.2 Decision Tree Induction

We analyzed the entire data set of Wilkinson (1997) using *See5* (a C4.5 implementation). For the sake of brevity, we included only those rules relevant to the characters listed in Table 4. Note that for a few of the first suite of characters identified as correlated by O'Keefe and Wagner (2001), C4.5 identified dependency rules that include characters in that suite. For example, Character E1.5 was identified as being dependent upon Character E3 and Character E1.6 is dependent on Character E1.1. However, for the other characters, dependency relationships were found that did not correspond to those expected from O'Keefe and Wagner's (2001) results.

The results of the C4.5 analysis indicate relationships among the data that were not identified by the correlation analysis of O'Keefe and Wagner (2001). Interestingly, the C4.5 analysis identified dependency relationships among several characters of the eye (as identified by "E" in the character numbering). This seems to indicate that decision tree induction may be a useful tool in addition to the analysis of O'Keefe and Wagner (2001) in identifying additional hidden relationships in the data.

decision	rules
E1.1	Rule 1: E1.2 = 0 → E1.1 = 0 (0.864) Rule 2: E1.2 = 1 → E1.1 = 1 (0.900)
E1.3	Rule 1: E1.4 = 0 → E1.3 = 0 (0.846) Rule 2: E1.4 = 1 → E1.3 = 1 (0.857)
E1.5	Rule 1: E3 = 0 → E1.5 = 0 (0.875) Rule 2: E3 = 1 → E1.5 = 1 (0.833)
E1.6	Rule 1: E1.1 = 0 → E1.6 = 0 (0.900) Rule 2: E1.1 = 1 → E1.6 = 1 (0.910)
E3	Rule 1: T1 = 0 → E3 = 0 (0.857) Rule 2: T1 = 1 → E3 = 1 (0.889)
T43	Rule 1: T1 = 0 → T43 = 0 (0.875) Rule 2: T1 = 1 → T43 = 1 (0.895)
T57	Rule 1: A51 = 0 & A75 = 0 → T57 = 0 (0.867) Rule 2: A75 = 1 → T57 = 1 (0.900) Rule 3: A51 = 1 → T57 = 1 (0.900)

Table 5. Results of C4.5 analysis of Wilkinson's (1997) data set. Only data for the first suite of correlated characters (Table 4) identified by O'Keefe and Wagner (2001) are reported.

However, it is important to remember that applying decision tree induction in order to infer relationships of phylogenetic characters is problematic because of the issues discussed earlier (including loss of information and the identification of false relationships).

4.3 Rule Induction from Coverings

Because of the size of the Wilkinson (1997) data set, finding all coverings in the data set would have been computationally expensive. Therefore, we restricted the analysis to only report coverings containing three or fewer characters and rules that applied to at least three taxa. Despite these restrictions, *RICO* identified more than 575 coverings and more than 700 rules just for the characters listed in Table 4. Thus, it was necessary to further limit the results of the analyses reported here. To restrict the information presented (and thus allow for a manageable discussion of the results), we identified highly useful attributes (Table 6)—characters that were involved in at least five rules (for a given decision attribute). Although this number is completely arbitrary, it is quite small considering that *RICO* reported 100+ rules for most of the decision attributes. We chose to err on the side of overestimating dependency relationships (some of which could subsequently be ruled out by further examination), rather than to exclude true dependency relationships. (Note: It is important to remember that in some cases, *RICO* will report rules with different combinations of states resulting in the same state in the decision attribute, as discussed previously. Thus, although identifying highly useful attributes provides a

good "first pass" at determining dependency relationships, it is important to subsequently examine the rules in which the attributes were involved.)

For Character E1.3, there were only three rules identified, each of which included Characters E1.1, E1.4, and E1.5. Therefore, the values reported for E1.3 in Table 6 are all of the characters that E1.3 are dependent upon, not just highly useful characters as defined previously.

The *RICO* analysis identified several dependencies that are consistent with the findings of O'Keefe and Wagner (2001). For example, Character E1.1 was dependent on Characters E1.5 and E1.6. Similarly, E3 was found to be dependent on E1.1, E1.3, and E1.5. However, several other relationships were uncovered that were not found using the methods of O'Keefe and Wagner (2001). For example, the rules pertaining to Character E3 as the decision indicate that E3 also has a dependency relationship with additional characters (E1.2, E1.4, E4, O4, T1, T35, T50, T52).

decision	highly useful attributes
E1.1	E1.2, E1.5, E1.6, E6, A10, H1.4, H2, T46, T47
E1.3	E1.1, E1.4, E1.5
E1.5	E1.3, E1.6
E1.6	E1.1, E1.2, E1.3, E1.4, E1.5, A10, O3, O6, T53, T55, T56
E3	E1.1, E1.2, E1.3, E1.4, E1.5, E1.6, E4, O4, T1, T35, T50, T52
T43	E1.1, E1.4, A4, H1.4, H2, H5, O3, T15a, T20a, T28, T42, T46, T47, T50, T51, T57, T58
T57	E1.2, E1.3, H2, T28, T32, T53

Figure 6. Highly useful attributes identified by a *RICO* analysis of the Wilkinson (1997) data set.

As with the C4.5 analysis, the *RICO* analysis identified dependency relationships among several characters of the eye, but *RICO* also included characters pertaining to other parts of the body (e.g., muscles, cranial bones, etc.). This suggests that *RICO* may be a useful tool in addition to the analysis of O'Keefe and Wagner (2001) in identifying additional non-obvious relationships in the data. Because *RICO* identifies all relationships (and is only limited by the constraints that the user imposes), it is possible that all dependency relationships in the data set could be uncovered. Furthermore, the *RICO* analysis does not report false relationships as could C4.5.

5 Conclusions

5.1 Interestingness of the Methods

Based on the performance of the three methods on both the test data set and the Wilkinson (1997) data set, we are able to comment on the interestingness of the results reported by each method. Data patterns can be said to be interesting if: (1) they are easily understood (by humans), (2) they are valid on new data with a degree of certainty, (3) they are potentially useful, and (4) they are novel (Han and Kamber 2001). All of the methods presented herein are equal in the novelty of their results; however, they all vary to some degree relative to the first three criteria.

Inherent in the idea of a Bayesian belief network is a method of presenting dependencies in visual, easy-to-identify form. However, Bayesian analyses, when applied to the problem described herein, are limited with respect to the ordering of characters and the necessity of prior knowledge of relationships. Therefore, the patterns of dependencies inherent in phylogenetic data sets are not obvious through Bayesian analysis. Furthermore, because of the number of different perturbations required to uncover a complete picture of the dependency relationships, the validity of the method when applied to new data is suspect. Thus, the utility of the method is limited in this application, and the usefulness of the Bayesian networks here is poor.

The results of decision tree induction provide a more readily understandable pattern of dependencies in phylogenetic data sets. The reported rules give clear statements of dependency relationships from one attribute to the next. However, because those rules are not "perfect," their degree of certainty when applied to novel data (or even some portions of the existing data set), is suspect. Furthermore, because there is the possibility of reporting false dependency relationships, or missing some dependency relationships altogether, the usefulness of decision tree induction for understanding phylogenetic character non-independence is limited.

The results of the rule induction from covering analyses are somewhat difficult to understand because of the sheer number of rules reported. Furthermore, when multiple combinations lead to the same decision state, results become slightly obscure. But because *RICO* reports all coverings, and all rules are "perfect," it can be applied to new data with complete certainty. Furthermore, coverings can be used to identify all dependency relationships, and rules from coverings can be used to examine further the nature of those relationships. Overall, *RICO* provides the most useful results for the problem described herein.

5.2 Using *RICO* Results in Phylogenetic Analyses

Now that we have identified the most suitable method for determining character dependence in phylogenetic data, we want to caution the reader to use the information gained from *RICO* carefully. In typical phylogenetic practices, if one has reason to believe that a set of characters is non-independent, usually one of two steps is taken: (1) all but one of the non-independent characters is deleted (or all are combined into one character describing a single "character suite") or (2) a weighting scheme is invoked that results in each character having less importance in the phylogenetic analysis.

Neither of these steps should be taken solely on the basis of *RICO* results. The results of *RICO* analyses should be used as a way to identify *possible* character non-dependence problems in phylogenetic data sets—in other words, it provides hypotheses of character non-independence that should be tested by phylogenetic analysis. Because data mining techniques are based on

recognizing patterns in data, it is possible that some of the patterns recognized by *RICO* reflect homology (= true evolutionary history), and thus should be preserved in the data set.

The best possible scenario for utilizing *RICO* results would be to: 1) run the *RICO* analysis before any phylogenetic analysis; 2) run the phylogenetic analysis and plot the distribution of characters on the resulting tree; and 3) compare the *RICO* results to the resulting phylogeny. If there is a high degree of homoplasy (e.g., reversals, parallelisms) in the phylogenetic analysis, particularly at nodes where several characters identified as dependent by *RICO* show support, then it is likely that there is non-independence in the data set. The next step would be to run the analyses presented by O'Keefe and Wagner (2001), paying special attention to those sets of characters identified as non-independent by *RICO* (including homologies and homoplasies).

5.3 Application to Molecular Data Sets

One area that we have made little mention of thus far is in the application of these methods to genomic data. In this paper we concentrated our discussion of the various methods on morphological data sets for three reasons: (1) they are relatively small and easy to work with, (2) character non-independence is easier to conceptualize (and identify) in morphological data, and (3) very little is known about character non-independence in molecular phylogenetic data.

Certain characteristics of DNA indicate that character non-independence poses as much of a problem in molecular data sets as it does in morphological data. For example, some regions of DNA are highly conserved, suggesting that some areas evolve independently, whereas other regions evolve as a character "suite". Also, some DNA sequences are selected for because of their structural properties (e.g., preferential binding, molecular stability, etc.), suggesting that some nucleotide positions are dependent upon those around them. Thus, applying a method such as *RICO* to understand the inter-relationships of molecular data would be highly valuable.

An additional impetus to applying *RICO* to molecular data sets is that by determining all coverings in a data set, we are able to generate rules that can be used for predictions. If we could identify dependency relationships among loci, we could use known sequence information to estimate unknown sequences.

Unfortunately, because of the size of most molecular data sets, applying *RICO* and other data mining techniques to large data sets is currently intractable. However, we plan to explore the implementation of *RICO* using parallel and distributed processing, so that we can examine all coverings in large genomic data sets. Furthermore, we plan to explore the use of other data mining techniques alone and in combination with *RICO* to further examine the problem of phylogenetic character non-independence.

6 Acknowledgements

We thank Dan St. Clair for his discussions and helpful comments.

7 References

Cheverud, J. M., Dow, M. M., and W. Leutenegger. (1995): The quantitative assessment of phylogenetic constraints in comparative analyses: Sexual dimorphism in body weight among primates. *Evolution* **39**:1335-1351.

Chickering, D. (2002): Learning Equivalence Classes of Bayesian-Network Structures. *Journal of Machine Learning Research* **2**:445-498.

Felsenstein, J. (1973): Maximum-likelihood and minimal-steps methods for estimating evolutionary trees from data on discrete characters. *Systematic Zoology* **22**:240-249.

Felsenstein, J. (1985): Phylogenies and the comparative method. *American Naturalist* **125**:1-15.

Grzymala-Busse, J. (1991): *Managing Uncertainty in Expert Systems*, Boston: Kluwer Academic Publishers.

Han, J. and Kamber, M. (2001): *Data Mining Concepts and Techniques*. San Francisco: Morgan Kaufmann Publishers.

Kluge, A. G. and Farris, J. S. (1969): Quantitative phyletics and the evolution of anurans. *Systematic Zoology* **18**:1-32.

Maddison, W. P. (1990): A method for testing the correlated evolution of two binary characters: Are gains or losses concentrated on certain branches of a phylogenetic tree? *Evolution* **44:**539-557.

Maglia, A. M. (1998): Phylogenetic relationships of pelobatid frogs (Anura: Pelobatidae) based on osteological evidence. *Scientific Papers of the Natural History Museum of the University of Kansas.* **10**:1-19.

McCracken, K. G., Harshman, J., McClellan, D. A., and Afton, A. D. (1999): Data set incongruence and correlated character evolution: an example of functional convergence in the hind-limbs of stifftail diving ducks. *Systematic Biology* **48**:683-714.

O'Keefe, F. R. and Wagner, P. J. (2001): Inferring and testing hypotheses of cladistic character dependence by using character compatability. *Systematic Biology* **50**:657-675.

Pawlak, Z. (1984): Rough Classification. *International Journal of Man-Machine Studies* **20**: 469-483.

Quinlan, J. (1986): Induction of Decision Trees. *Machine Learning* **1**:81-106.

Quinlan, J. (1993): *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann Publishers.

Witten, I. and Frank, E. (2000): *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. San Francisco: Morgan Kaufmann Publishers.

Wilkinson, M. (1997): Characters, congruence, and quality: A study of neuroanatomical and traditional data in caecilian phylogeny. *Biological Reviews* **72**:423-470.