Problematic character coding methods in morphology and their effects

M.D. Brazeau (Martin)

The LINNEAN SOCIETY of London BIOLOGICAL Journal of the Linnean Society

Biological Journal of the Linnean Society, 2011, **104**, 489–498. With 4 figures

REVIEW ARTICLE

Problematic character coding methods in morphology and their effects

MARTIN D. BRAZEAU*

Netherlands Centre for Biodiversity Naturalis, PO Box 9517, 2300 RA, Leiden, The Netherlands

Received 10 March 2011; revised 6 June 2011; accepted for publication 6 June 2011

The effects of different coding practices in morphological phylogenetic analysis are well documented. In many cases, we can determine that certain practices can be regarded as undesirable and should be avoided. Certain coding practices do not correctly translate the expected information to the cladistic algorithm. It may go unnoticed that expressions of character information in character lists, which may be entirely logical to any reader, do not necessarily reflect the mathematics employed by a phylogenetic algorithm. Despite a wealth of literature on coding procedures and documentation of these issues, problematic character coding practices are still common. A review is provided of different coding and character formulation practices, particularly relating to multistate character information that may either: (1) lead to a failure to capture grouping information implied in the character list; (2) cause problematic weighting or spuriously high certainty in particular optimizations; and (3) impose congruence artificially, by linking more than one variable character to a particular state. Each of these is reviewed and presented with a hypothetical example. Recommendations for avoiding these pitfalls are described in light of how parsimony algorithms work with character data. Character lists must be drawn up not only to present character variation logically, but also with consideration for how computer algorithms implement cladistic logic. The widespread use of problematic character coding procedures may account for some of the perceived problems with morphological data. Therefore, an exploration of the effects of these methods and standardization of methods should be a goal for the very near future. *Biological Journal of the Linnean Society*, 2011, **104**, 489–498.

ADDITIONAL KEYWORDS: cladistics – discrete characters – Fitch algorithm – parsimony.

INTRODUCTION

When are cladistic methods not cladistic? The most important features of cladistic analysis are the explicit criteria for distinguishing homologies (i.e. synapomorphies) from homoplasies and synapomorphies from symplesiomorphies. However, the ability of computer-based cladistic analyses to accomplish these tasks in the way intended by the user can be severely hampered by different methods of coding the same data, especially for multistate characters. Nevertheless, published character lists for morphological phylogenetic analyses demonstrate considerable variation in the methods used to represent multistate character information. This variation is often unexplained or unjustified, especially when a combination of methods is employed. However, investigations have shown that different coding practices can affect tree search results, as well as interpretations of character homology and transformation (Pleijel, 1995; Hawkins, Hughes & Scotland, 1997; Strong & Lipscomb, 1999; Forey & Kitching, 2000). Although debate exists as to which is the best method to apply, there are nevertheless practices that would seem to be universally undesirable. Here, a review is provided of some common problems in devising morphological characters and how they may either violate the assumptions of the analysis or fail to take these assumptions into account, even when they appear fine 'on paper'. The purpose is to alert not only morphologists, but also molecular systematists of the undesirable effects that certain coding practices may entail and to highlight possible ways of dealing with them. Both groups have an

*E-mail: martin.brazeau@gmail.com

interest in how or why their results differ both within and between data types used for phylogenetic analysis. It is hoped that this will help investigators better capture the grouping information available in their data at the same time as avoiding spurious conclusions.

It has been shown that certain approaches to multistate character information fail to capture significant grouping information in analysis (Hawkins *et al*., 1997), even when they logically represent the character data to the human reader. This is because not all methods of representing character data are mathematically equivalent when the phylogenetic algorithm is taken into consideration. A corollary of this is that considerable variation in results between studies may have little or nothing to do with the data themselves. It may instead have very much to do with how the data are atomized by the researcher and translated into variables in a two-dimensional matrix. It may even imply the need for a major re-evaluation of contemporary morphological datasets to assess the influence of spurious coding practices. It is hoped that the present review will provide a useful guide to investigators applying phylogenetic systematics to questions of species relationships and morphological homology. This review should also serve as a guide to editors and referees who must review character lists and data matrices as the problems highlighted here can be easily spotted. The issues raised in this paper concern what should be universally undesirable consequences: (1) failure to capture important grouping information; (2) unintentional weighting of characters; and (3) unintentional or unjustified imposition of congruence on characters. It is therefore hoped that attention to these problems may lead to some editorial policies on character lists, or move us towards a standardization of methods.

Terminology

Characters and character states

There exists some debate about the validity or importance of the distinction between characters and character states. There are cogent arguments both for why the distinction is important (de Pinna, 1991), and why there may be no difference at all (Patterson, 1982). The problem can be remedied by applying a distinction similar to that between apomorphy, synapomorphy, and symplesiomorphy (this may, in fact, be the same distinction). That is, the distinction is indeed important, although it is not an absolute one but is defined by relative context in a phylogenetic hierarchy. Any morphological character might be construed as a character state of another character, especially one to which it is subordinate. For example, hair is not only a character, but also a character state of the epidermis.

Compound character

Any character requiring two or more conditional qualifiers to specify it. For example: 'a tail that is both blue and scaly' is a compound character if treated as a single variable. It is, in fact, three variables; two of which exhibit a hierarchic dependency on the condition of having a tail.

Token

Any symbol or object which stands to represent an object or idea. In this case, integer or symbol values used to represent the morphological condition observed in an organism.

Transformation series

Because the term 'character' can have equivocal usage (Ghiselin, 1984), the term 'transformation series' is used here when referring specifically to the columns of a data matrix. The terms 'character state' and 'character values' are used interchangeably here.

Parsimony algorithms

Problematic character coding schemes arise when the investigator fails to account for how steps are counted on a tree. Most researchers applying cladistic methods will have familiarity with its theoretical underpinnings (something we first learn by making cladograms 'by hand'). Because we can read the text of a character list, we are able to mentally translate the information that we read into accurately counting steps on trees. We can see when states of a character share a relationship not shared with a third state. Mixing methods of coding will not necessarily lead to problems for human readers, so long as we understand the intentions of the character list's author (who is often identical to the reader). However, most contemporary cladistic studies are not done by hand, but by computer programs. Although character lists may have much detail in them, a parsimony program only receives a matrix of discrete symbols. A thorough and accessible review of the most common procedure for assigning ancestral states, counting steps, and selecting trees is given in the first two chapters of Felsenstein's (2004: 1–18) *Inferring Phylogenies* and Chapter 4 of the MACCLADE 4 manual (Maddison & Maddison, 2001). For completeness, important aspects of the generally used Fitch algorithm (Fitch, 1971) for unordered characters are reviewed here because they are worth keeping in mind when considering the examples below, or when assembling one's own character list.

For every tree topology examined during the tree search process, the optimality (parsimony score) of a tree is evaluated one transformation series (i.e. one matrix column) at a time, summing the steps required by all series for each tree. To begin the process, character state symbols (e.g. *A*, *C*, *T*, *G* or 0, 1, 2, 3, . . . *n*) of a particular character (i.e. position along a DNA sequence or a particular anatomical structure) are first assigned to the tips of the branch corresponding to the terminal taxon in which they are observed. The algorithm 'works its way down' from the branch tips (called a downpass or a postorder traversal), assigning a preliminary set of characters at each internal node by examining the states found at its descendant nodes. When the descendant nodes share the same state of a character, nearest common node is assigned that character. Where they do not share a common character, both characters are applied to the internal node and a step is added to the calculated length of the tree (i.e. the tree now implies at least one evolutionary change for this character). At this stage, the only thing the algorithm checks is whether or not the symbols at the two descendent nodes are the 'same token'. Final optimization for each node must wait for information from its local outgroup nodes. This requires another pass of a preorder (or uppass algorithm), which starts at the root and assigns to a node the set of characters shared by its nearest sister node and its ancestor that were calculated during the downpass. The notion of being unweighted or having symmetrical costs should now be evident because the algorithm does not 'care' which states it finds first at the descendent nodes, nor does it evaluate the additive difference between the numerical values of the numerical tokens. Again, unless the algorithm is informed otherwise (i.e. by ordering, weighting, or loading a step matrix, which will use a different algorithm), this has the same cost, regardless of which token may be found at either node: one if they are different; zero if they are the same.

This algorithm takes into account all the information from the branch tips and the nearest sister taxa of a particular node to estimate the synapomorphic set of characters at that node. It does so repeatedly and consistently, perhaps more consistently and objectively than a person doing the procedure by hand might. However, these algorithms present a potential series of problems for morphological data. They were conceived for use with molecular sequence data, and thus for a finite number of states and a linear array of (mostly) independent transformation series. The algorithm's design assumes that all information about variation within a character can be contained within a single transformation series. Because morphological data are inherently hierarchical, they may violate this assumption. To make these methods work for morphological data, we must somehow distribute the hierarchic information to multiple transformation series or try to compress it into a single transformation series. In either of these two approaches, there may well be a state token for each variable, and no difference will be apparent to the reader. However they will actually have very different computational consequences for cladistic algorithms, because (again, barring weighting or ordering) the transformation series and states within them are nonhierarchical and the algorithm actually assumes 'independence' of each transformation series.

PROBLEMATIC CHARACTER CODINGS

'Uninformative' states and 'pseudo-ordering'

The attempt to compress hierarchically related states into a single transformation series results in a compound character. When a compound character includes an absence state plus a series of 'present but variable' states, phylogenetic information can be lost to the tree search process. The problem arises from the tacit, but false, assumption that parsimony algorithms assign a relationship to non-0 tokens (i.e. 1, 2, 3 . . .) to the exclusion of the 0 token. However, the Fitch algorithm for unordered, unweighted characters is actually indifferent to the particular state symbols used. Hierarchic relationships between states in a transformation series can only be transmitted to the algorithm in the form of a model (either a step matrix or character tree) or ordering. Otherwise, all changes are symmetric.

The first case is that of an 'uninformative' state, which arises when any one state in a compound multistate character appears only once in its respective transformation series (Fig. 1). Thus, in a three-state unordered character, if state 2 appears only for one taxon, then that 2 imparts no 'grouping' information. It is also the case that PAUP* does not report such states as uninformative, and so it can easily be missed by investigators. The reason for the uninformativeness is the symmetry and equal weighting of all transformations in the transformation series (Fig. 1). Because a change from 0 to 1 is equal to a change from 0 to 2, or from 1 to 2 (or any inverse of these changes), it makes no difference in parsimony score whether 1 and 2 arise together (imply synapomorphy of their shared condition) or appear separately in a given tree: both situations require at least two steps.

The 'uninformative state' is a particular case of what can be called 'pseudo-ordering'. In the former, the state is only uninformative because we have no additional taxa exhibiting state 2. If we do add such

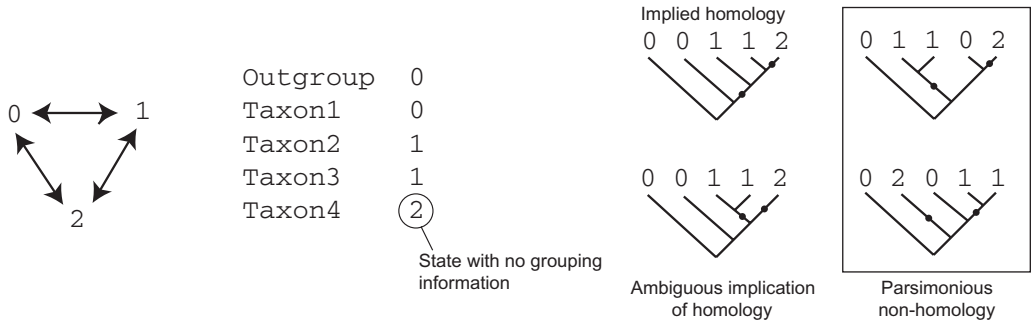

Figure 1. An unordered, equally weighted multistate character as commonly found in morphological data matrices and summarizing the problems of uninformative states. The middle section shows the character scored in a hypothetical matrix. The four trees to the right indicate four optimal and equally parsimonious character state mappings for this character formulation, with character states indicated at the branch tips. Note that the two trees on the far right do not require a synapomorphic relationship between states 1 and 2, and are thus implying homoplasy for whatever condition is shared between them (such as presence of the structure of which 1 and 2 are simply variants). If the multistate information were broken into two characters, the trees in the box on the right would require one additional step over the ones to the left.

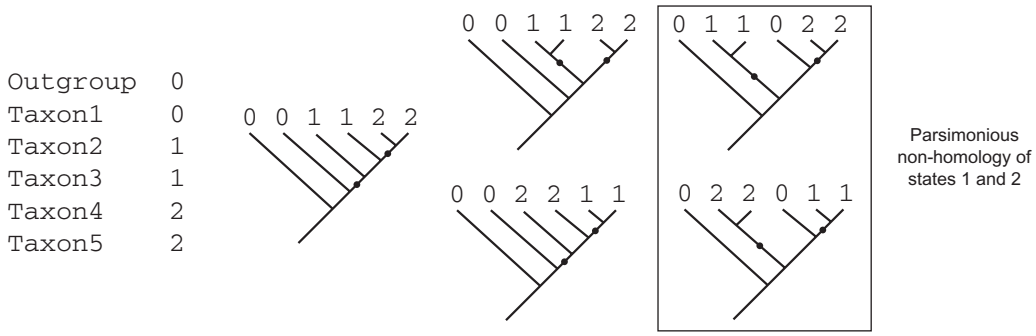

Figure 2. Equally parsimonious solutions with one taxon added to the character scheme of Fig. 1. As in Fig. 1, no special synapomorphic relationship between characters 1 and 2 is implied in the cladograms furthest to the right. Thus, if state 0 reflects an 'absence' state, and states 1 and 2 are two different conditions of the 'presence' state, then information about the 'presence' state is lost and non-homology of the condition is equally parsimonious as homology. As in Fig. 1, if the multistate information were broken into two characters, the trees in the box on the right would require one additional step over the ones to the left.

taxa, the state becomes informative, although only at the level of grouping taxa exhibiting state two. As before, all transformations are symmetric, unless they are ordered or weighted. That is, states 1 and 2 are each one step from 0 and from each other. No information is given to the algorithm to indicate that states 1 and 2 have a special relationship not shared with 0. Nevertheless, characters such as the following are common in data matrices:

Foramen in bone *X*: absent(0); present and oval (1); present and keyhole-shaped (2).

In such a formulation, information about the presence condition, in spite of shape, is lost (Fig. 2). Without ordering (likely unjustified for such a character), grouping information between states 1 and 2 is lost because the analysis treats all changes as symmetrical.

Recommended solution

Except for the number of taxa representing each state, there is no major difference between 'pseudo-ordering' and the 'uninformative' state problem. One simply has to account for two dichotomies: one between the presence and absence of the feature, and one between the two (or more) versions of the feature. An uninformative state can be remedied by ordering, although with some reservation. It will not affect tree shape, although it can create spurious transformational optimizations under some topologies. The best solution is to atomize the characters in such as way as to capture: (1) presence or absence of the feature and

(2) values (i.e. states) of the feature. Using contingent coding, the character list might appear as follows:

1. Foramen in bone *X*: absent (0); present (1).
2. Shape of foramen in bone X when present: oval (0); keyhole-shaped (1).

Bear in mind that the order of state tokens does not matter. For example, absent could be assigned 1, and present 0. As long as all species that are the same are coded the same.

Different procedures for dealing with this problem are discussed later in the text. The case of contingent (or reductive) coding, which is the method preferred here, the matrix would appear as follows:

Character:	1	2
Taxon 1	0	–
Taxon 2	0	–
Taxon 3	1	0
Taxon 4	1	0
Taxon 5	1	1
Taxon 6	1	1

With '–' standing in for 'logical inapplicability'. This can be substituted for a '?' but, for clarity, a minus sign tells the reader that the missing entry symbol is employed because of logical inapplicability rather than absence of data. The token will be treated the same way by the algorithm: as missing data.

Because non-applicable characters are treated as missing data, some spurious results can arise (Maddison, 1993; Strong & Lipscomb, 1999). Therefore, when analyzing such data, the software in use should be instructed to collapse zero-length branches (Coddington & Scharff, 1994; Strong & Lipscomb, 1999). This is done by default in NONA and TNT, although it has to be set manually in PAUP*.

Repeated absences

Repetition of the state 'absence' for a particular character also reflects a failure to acknowledge the symmetry of character changes. In the following fictional example, the absence of a bone appears as a character state three times

1. Bone *X*: absent (0); present (1)
2. Bone *X*: absent (0); oval (1); bilobate (2)
3. Bone *X*: absent (0); smooth (1); rugose (2).

This has two related weighting effects. In characters 2 and 3, states 1 and 2 can only appear if character 1 is in state 1. Thus, by having an absence state, presence is counted an additional time for each of character 2 and 3. Regardless of the shape or texture of bone *X*, the presence of the bone will always demand at least two extra steps. Thus, this is the same as giving state 1 a cost of three steps. Furthermore, the absence state is appearing three times. This means the loss of bone *X* can be counted as many as three times along the same branch without ever having to count its reappearance!

Recommended solution

Remove the additional appearances of the absence state. The condition of absence of a structure should appear only once in a data matrix.

Helpful rule of thumb

For all character states, consider how many times this exact same state could appear along a single branch, even if found in a different transformation series. If it is more than once, there is a problem with the character formulation.

Compound characters

Subjectivity and supposed lack of repeatability are common criticisms of morphological systematics. Perhaps aiming to counter this, the well-intentioned morphologist attempts to give very specific, detailed descriptions of their characters to increase the ease of identification. The problem is that such character descriptions can create compounds of conditions if the descriptive conditions are all treated as necessary conditions for identifying a trait. However, one must beware that each of such condition might legitimately be considered its own character. Otherwise, one imposes congruence or incongruence on traits where this does not necessarily have to be the case.

An example:

A dermal bone with contact to bone A and bone B and bears the pineal opening: absent (0); present (1).

Here, we can only presume that the qualities of being a dermal bone, contacting bone A and bone B, and having a pineal opening are important for identifying the structure in question. Furthermore, we can only presume that these qualities are important because their existence in combination is otherwise variable, and therefore might not always be found in combination. However, because they are variable, then their variations could independently supply phylogenetic information. For that reason, treating them as a compound may have theoretically dubious consequences: imposing congruence on certain instances of the qualities needed to identify a particular bone.

Complex phenotypes may have evolved gradually, perhaps in a stepwise fashion and character lists may need to leave the analysis open to discovering these patterns of character evolution. Using many qualities to identify a character may preclude discovering a more general relationship to other characters. Debates may quickly descend into fruitless

arguments over 'definitions' of characters, rather than over describable properties of organisms. It is therefore granted that this section may be the most theoretically controversial as it relates to the problem of similarity in testing homologies. Nevertheless, homologies need not reflect all-or-nothing similarity. Instead, the degree of similarity is proportional to the degree of relationship. Therefore, I consider any additional similarity criteria to be irrelevant to the problem of a hypothesis of homology once it has been properly cast as a conditional relationship (in agreement with Kluge, 2003). However, this position is not without its criticisms (Rieppel & Kearney, 2002, 2007). Regardless of these debates, it is worthwhile that authors of phylogenetic datasets critically assess these kinds of compound characters, especially where they require specifiers that could (or even do!) easily appear as their own independent character in a dataset.

Helpful rule of thumb

Consider how many descriptive terms are required to qualify a character. Consider whether they are not specific instances of more generally applicable characters. For example, as in the tail example used in the introduction (*sensu* Maddison, 1993), a tail that is described as being scaly could simply be a more general feature of an organism, more general than the property of having or not having a tail. Therefore, presence of a tail is one character while scaliness is another, and both could be given their own transformation series. Being a scaly tail does not refute homology with other characters conditional on being a tail.

CHARACTERS VERSUS THEIR TOKENS

Some of the problems described above may arise from the fact that symbols like '0' and '–' (as a 'minus' sign) have connotative associations with concepts such as absence or loss, which originate from usages in other (i.e. non-phylogenetic) contexts. A common thread among these problems is the implication that '0' is treated as inherently different from any nonzero state, perhaps even standing in for primitiveness. This is perhaps a result of the frequent use of '0' for the states of the root taxon as a matter of convention. The possibility arises that the convention sometimes gets mistaken (consciously or unconsciously) for a procedural necessity. The purpose of reviewing some details of phylogenetic counting algorithms earlier was to underscore the fact that the tokens only indicate an identity. This helps to evaluate not only how the above scoring procedures are problematic, but also how we choose among alternative ways of dealing with these problems, as described in the subsequent section.

It is important not to confuse '0' with a statement of primitiveness when drafting character lists and scoring matrices, especially for unordered characters. In theory, it makes no difference whether absence is represented by '–', '+', '0', or '1'. Assuming you have not ordered or weighted characters, you could swap all instances of '0' for '1' and vice versa and you will still obtain the same result as before. The only thing that matters is that taxa that share the same conditions are coded the same way. Because of the default assumptions of transformational symmetry, there is no difference between 0 scores and non-0 scores where the assessment of a transformation cost takes place. States 1, 2, 3, etc., are not derived states until a phylogenetic analysis has been conducted and returns that result, and the Fitch algorithm will not add any additional steps if each appears independently in the tree or together. Similarly, state 0 is not the 'plesiomorphic state' simply because it is labelled 0. 'Derived' and 'plesiomorphic' are properties that are exposed as the phylogenetic algorithm assigns optimal values of the characters to internal nodes. They can be represented by any token the investigator chooses. It is the investigator's responsibility to ensure that the tokens accurately reflect the character information in a truly symmetric manner because the parsimony algorithm (unless given specific commands otherwise) will treat character transformations symmetrically. The consequence of failing to account for this symmetry will be a loss of grouping information.

RECOMMENDATIONS AND CONCLUSIONS

Character-taxon matrices and their accompanying character lists should be viewed as formatted data, and not just a table of observations. That is, they should be constructed with an understanding of how that information will be interpreted by the algorithm that is receiving them. For many multistate characters, authors should consider how character state information is (or is not) distributed to other transformation series. The problem with '0' being used as a catch-all for anything that simply 'isn't 1' should be borne in mind when using binary characters (see discussion below). The use of multistate characters should be critically assessed, especially where they relate to compounds of a presence condition with other variable states. A state appearing only once is defensible if it is to avoid lumping non-equivalent conditions under a single catch-all alternative state. This would occur, for example, when there really is a trichotomous value (e.g. three different colour characters). However, it is problematic if it results in ignoring valid character identities contained in the character descriptions, and thus this recommendation

concerns only multistate characters where there is a hierarchic relationship between the conditions of states.

Approaches to multistate data

Admittedly, multistate characters introduce their myriad troubles such as adding missing data and pseudo-parsimonious optimizations (Maddison, 1993). There exists a diversity of views about the best approach to these problems. Solutions include contingent coding and the use of non-applicable states (Strong & Lipscomb, 1999), non-additive binary coding for all variables, and the use of Sankoff matrices on elaborate character compounds (Forey & Kitching, 2000). What should concern us most, however, is the ability of a coding procedure to permit the analysis to do what we expect it to do given the data we have to hand. Until more appropriate algorithms are implemented in available software, all the current approaches have their problems, and only some can be said to be better than others.

Non-additive binary coding

This method simply elects to have all states scored as either 0 or 1. The most extreme implementation is where 0 represents 'absence' and the 1-value represents the trait value. No multistate characters are scored because each trait value is given its own absence/presence character. As a consequence, all variables should be accounted for, provided that no compound characters are used. However, serious problems arise from this method, and a description of them helps to understand coding problems more generally. In agreement with Forey & Kitching (2000), non-additive binary coding is unjustifiable because of its failure to give logical character optimizations. However, another problem is that it may force or permit underestimates of the actual amount of variation in a character, and thus favour hypotheses of synapomorphy for what could as well be symplesiomorphy. To illustrate the problems, consider discrete character coding methods applied to molecular sequence data, which represent a multistate character problem. Non-additive binary coding makes the absence token (usually 0) correspond to a 'nonspecified other' variable (Hawkins, 2000). The '0' token becomes a catch-all for anything that isn't scored as '1'. To see the potential problem with this, consider applying the non-additive binary coding to molecular sequence data: we would formulate characters such as 'adenine at site 121: absent (0); present (1)'. The optimization problem can be quite clearly seen: 'not-adenine' could be optimized as a supporting synapomorphy of a node, regardless of whether the absence referred to a different base or an insertion or deletion at that site.

Another major problem of non-additive binary coding is its preference for making each value of a character its own synapomorphy when plesiomorphy may be an equally parsimonious alternative. In the case of DNA, we can see that our decisiveness about adenine as a synapomorphy of a clade will depend on the kinds of non-adenine bases in other species under analysis. If all non-adenine bases (i.e. cytosine, thymine, and guanine) are represented in the non-adenine taxa, the tree requires at least three steps, regardless of the tree topology. However, assuming a non-adenine root taxon, the non-additive binary version of this character only requires one step, for a monophyletic grouping of adenines (i.e. adenine as a single, unique, unchanged synapomorphy) requires one step. All other trees require two or more.

Figure 3 shows the analogous problem occurring in morphological data when non-additive binary coding is used. In this case, we force one state to act as a catch-all for anything that isn't the state with a clearly specified condition. Any given iteration of the counting procedure will always count fewer steps on any tree that places all instances of '1' in a clade to the exclusion of all species with '0'. This has the

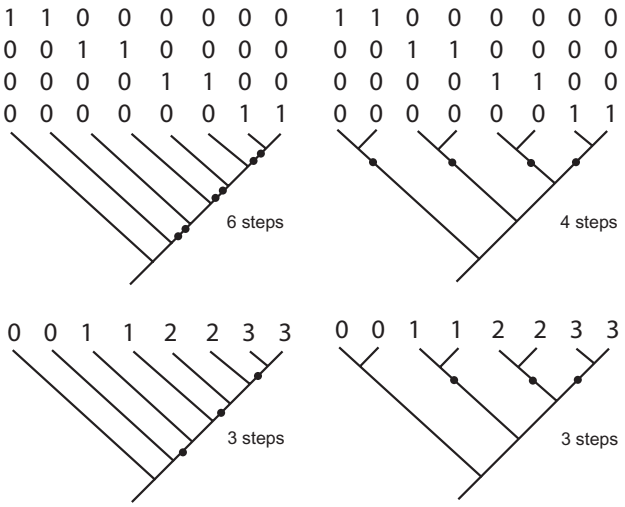

Figure 3. Comparison of the same observational data assumed to be conditions of the same feature or structure and are coded under non-additive binary coding (top) and multistate coding (bottom). Under non-additive binary coding, the pectinate tree on the left is two steps shorter than the tree in which the species that phenetically resemble each other are sister taxa. Under multistate coding, both topologies are of equal minimum length, and thus the coding procedure gives no preference to plesiomorphy over apomorphy interpretations of the characters. Note that, in the lower left tree, the exact placement of the changes would be ambiguous; the placement of the dots is only to facilitate counting.

effect of failing to distinguish apomorphy from plesiomorphy and will thus favour trees that make clades out of assemblages that could just as well be paraphyletic, given the same observations. Figure 3 compares non-additive and multistate coding for the same observation statements for two different topologies: a fully pectinate tree and one in which there is a clade for every shared state/character. Multistate coding is more tolerant of plesiomorphy and finds both topologies to be equally parsimonious. However, non-additive binary coding is biased in favour of apomorphy because an extra step is required to 'return' to zero (regardless of whether this 0 refers to anything that is similar to what it stood for lower in the tree). Notably, and perhaps not unexpectedly, this condition also resembles the UPGMA tree for these same data (not shown). This reflects a fundamental distinction between cladistics and phenetics: the distinction between plesiomorphy and apomorphy, which in the case of multistate coding is allowed to be more fairly decided by character congruence.

This may in turn explain why non-additive binary coding has been found to produce more fully resolved trees (Hawkins *et al*. 1997; Fig. 4), effectively by creating spuriously unequivocal character optimizations by lumping all alternatives that are 'not-1' into a single token, '0', when there may in fact be many different ways to be 'not-1' (just as there are four ways to be not-adenine, one of which involves not even being a base at all but, instead, the absence of the base position entirely, in the case of an indel). As a consequence, it reduces the number of equally parsimonious options, although typically by biasing against cases of possible plesiomorphy (see the 'doublet rule' of Maddison *et al.* 1984). As a result, the cladistic analysis may not be doing what cladistics is intended to do. A helpful way to approach binary characters is to consider whether 'both' states refer to a singular identifiable condition, is not simply anything that isn't state-X, or could itself be used to group two organisms if one were to actually observe them.

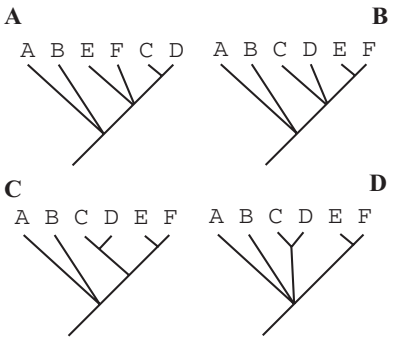

Figure 4. Four trees, adapted from Hawkins *et al*. (1997), showing distinct topologies obtained for the same data using different coding schemes. Trees A, B, and C result from contingent scoring. Trees A, B, C, and D all occur when the data is 'conventional' (i.e. corresponds to the 'pseudo-ordering' problem). Under non-additive binary coding, only tree C (the most fully resolved) results.

Contingent/reductive coding

An alternative to non-additive binary coding is to use either multistate or binary contingent coding schemes. These introduce non-applicable states (as in the examples above), treated as missing data, when there is no logical interpretation of the character for a given taxon. Numerical scores are only given to taxa that can logically be scored for a trait value contingent upon the presence of another character. This method has the benefit of allowing one to capture the grouping information implicit in the presence or absence of a feature, while simultaneously capturing the grouping information implicit in transformations between trait values. Unfortunately, non-applicable states are interpreted by modern parsimony algorithms as being identical to missing data, and are thus subject to the problems of 'pseudo-parsimony' (Maddison, 1993). Parsimony algorithms will optimize a contingent character state across parts of the tree where the subject character is absent (i.e. the 'red tail, blue tail' problem where tails may be absent in large parts of the tree, and nonhomologous instances of tail colour are nonetheless influencing each others' parsimony scores). Another problem is that contingent scoring can lead to spurious groupings by assigning a known state to a clade consisting of taxa sharing only the non-applicable state (Strong and Lipscomb, 1999). This can be circumvented by telling the algorithm to collapse zero-length branches, which is the default in TNT, although it must be set by the user in PAUP*.

Step matrices

Forey and Kitching (2000) recommend the use of Sankoff matrices and this may be a workable solution for some morphological datasets. The details of this procedure are beyond the scope of this contribution, and readers are encouraged to consult Forey & Kitching's work. In brief, however, it works as follows: a compound multistate character transformation series is produced that includes an absence state, plus a state for each of the possible trait combinations that can appear in the character when it is present. A step matrix is implemented to evaluate the path length in terms of the number of changes required to transform from one state combination to another. Step matrices are easily implemented in TNT (Goloboff, Farris & Nixon, 2008) or PAUP* (Swofford, 2003), although these often end up being very elaborate and may be very time-consuming to formulate. One has to be

cautious of weighting character transformations and proceed very carefully, ensuring that particular transformations are not given higher costs than they would normally have under any other coding practice. The problem with this approach is that it may weight the transformation from absence to presence more highly because presence can imply a series of two or more additional conditions (if one follows the procedure of Forey & Kitching). This is balanced by the fact that the other individual trait values no longer appear elsewhere in the primary data matrix but, instead, are implied within the step matrix. However, if some of the topologies in the search require multiple appearances of only one 'version' of the most inclusive character variable (that is, a transformation from 0 to 1, although no further transformations higher up in the clade), then the analysis may be biased against these. This would appear to depend on the weight of the forward transformation and the possible maximum number of times the character can appear in the tree. Further work exploring the effects of Sankoff matrices is required.

There appears to be no error-free method for approaching hierarchic morphologic data. However, in agreement with Strong & Lipscomb (1999), the best solution appears to be contingent coding (or 'reductive coding' in their terminology). More importantly, workers using cladistic methods with morphology should understand the mathematical consequences of their coding procedures when a phylogenetic algorithm is taken into account and interpret the results appropriately. Especially where parsimony is used, the behaviour of the algorithm is not a black box. It is very possible to understand what is going on 'inside' the functioning of a phylogenetic parsimony algorithm. It is therefore also possible to keep that process in mind when turning our descriptive data into a series of discrete variables that will be treated iteratively in phylogenetic analysis. Workers should therefore be explicit about the approaches they choose but certainly refrain from any methods that generate otherwise avoidable biases. Methods that should be precluded have been outlined here, and it is even hoped that some of these recommendations may even evolve into editorial standards.

Coding schemes or the nature of data?

Morphological data in systematics is often criticised in light of molecular results that obtain strong statistical support for a conflicting topology. It is not my intention to debate the merits of particular data types here because problems with methods may be as or more important than problems with data. Criticisms of morphology usually point to subjectivity, paucity, and incompleteness of data, convergence as a result of function, or other aspects that might affect the data itself. By contrast, very little attention has been paid by either side of this debate to the known or potential effects of different coding practices on morphological data. We know different strategies for the same observations will produce different trees. However, we are yet unaware of the extent to which variation across morphological phylogenetic topologies is attributable to discrepancies in coding practice. Was morphology really favouring a particular clade, or is a dubious coding practice behind the discrepancy? Answering the question will have important ramifications with respect to how we assess divergences between morphological and molecular datasets.

Furthermore, it is not uncommon to criticise morphology based on taxonomies that are not cladistically derived but, instead, are based on traditional, verbal character lists. Implicit in character lists is a form of reasoning that is similarly problematic to non-additive binary coding, or even phenetics. However, the monophyly of many traditional groups assumed to be supported by morphology can be questioned, simply in light of the fact that verbal character lists may be optimized equivocally when cladistic logic is applied consistently (Brazeau, 2009; Friedman & Brazeau, 2010). Traditional character lists, however, may tend to favour treating characters as synapomorphies over symplesiomorphies, leading to a higher tendency to consider paraphyletic assemblages monophyletic.

It remains very possible (even likely) that some discrepancies reflect artefacts of coding rather than data. However, this has never been deeply explored experimentally. What then for the divergence of some morphological and molecular trees? The default reaction is to assert problems with one or the other dataset. This appears to fail to acknowledge the most likely source of error: the human being doing the work. Only once a dataset becomes suspect, often times out of raw scepticism, are the methods examined. However, with morphological results, one should always examine the coding practices before concluding that the problem was with morphological data *per se*. The data may be fine; the problem might be that the computer was never told what all of the data are.

ACKNOWLEDGEMENTS

Motivation to write this note was provided by discussions about published data matrices with Torsten Liebrecht. Helpful comments were provided by Daniel Snitting and Matt Friedman. Leo Smith (Field Museum) and an anonymous referee are thanked for their comments that have greatly improved the manuscript. Much of this review was written while the author was a postdoctoral fellow at the Museum

für Naturkunde in Berlin. Johannes Müller is thanked for being a gracious host during this time, and for discussions on this and related topics. The author was supported by a Fonds québecois pour la recherche sur la nature et les technologies B3 post-doctoral fellowship.

REFERENCES

Brazeau MD. 2009. The braincase and jaws of a Devonian 'acanthodian' an the origin of modern gnathostomes. *Nature* **457:** 305–308.

Coddington J, Scharff N. 1994. Problems with zero-length branches. *Cladistics* **10:** 415–423.

Felsenstein J. 2004. *Inferring phylogenies*. Sunderland, MA: Sinauer Associates.

Fitch WM. 1971. Toward defining the course of evolution: minimum change for a specific tree topology. *Systematic Zoology* **20:** 406–416.

Forey P, Kitching I. 2000. Experiments in coding multistate characters. In: Scotland RW, Pennington RT, eds. *Homology and systematics: coding characters for phylogenetic analysis*. London: Taylor & Francis, 54–80.

Friedman M, Brazeau MD. 2010. A reappraisal of the origin and basal radiation of the Osteichthyes. *Journal of Vertebrate Paleontology* **30:** 36–56.

Ghiselin MT. 1984. 'Definition', 'character', and other equivocal terms. *Systematic Zoology* **33:** 104–110.

Goloboff PA, Farris JS, Nixon KC. 2008. TNT, a free program for phylogenetic analysis. *Cladistics* **24:** 774–786.

Hawkins JA. 2000. A survey of primary homology assessment: different botanists perceive and define characters in different ways. In: Scotland RW, Pennington RT, eds. *Homology and systematics: coding characters for phylogenetic analysis*. London: Taylor & Francis, 22–53.

Hawkins JA, Hughes CE, Scotland RW. 1997. Primary homology assessment, characters and character states. *Cladistics* **13:** 275–283.

Kluge AG. 2003. The repugnant and the mature in phylogenetic inference: atemporal similarity and historical identity. *Cladistics* **19:** 356–368.

Maddison WP. 1993. Missing data versus missing characters in phylogenetic analysis. *Systematic Biology* **42:** 576–581.

Maddison WP, Donoghue MJ, Maddison DR. 1984. Outgroup analysis and parsimony. *Systematic Zoology* **33:** 83–103.

Maddison DR, Maddison WP. 2001. *MacClade 4*, Version 4.02. Sunderland, MA: Sinauer Associates.

Patterson C. 1982. Morphological characters and homology. In Joysey KA, Friday AE, eds. *Problems of phylogenetic reconstruction*. Systematics Association Special Volume 2. London: Academic Press, 21–74.

de Pinna MCC. 1991. Concepts and tests of homology in the cladistic paradigm. *Cladistics* **7:** 367–394.

Pleijel F. 1995. On character coding for phylogeny reconstruction. *Cladistics* **11:** 309–315.

Rieppel O, Kearney M. 2002. Similarity. *Biological Journal of the Linnean Society* **75:** 59–82.

Rieppel O, Kearney M. 2007. The poverty of taxonomic characters. *Biology and Philosophy* **22:** 95–113.

Strong EE, Lipscomb D. 1999. Character coding and inapplicable data. *Cladistics* **15:** 363–371.

Swofford DL. 2003. *PAUP*. Phylogenetic analysis using parsimony (*and other methods),* Version 4. Sunderland, MA: Sinauer Associates.