



## A NOVEL METHOD FOR ECONOMICAL DIAGNOSIS OF CLADOGRAMS UNDER SANKOFF OPTIMIZATION

Ward C. Wheeler<sup>1,3</sup> and Kevin Nixon<sup>2</sup>

<sup>1</sup> *Department of Invertebrates, American Museum of Natural History, Central Park West at 79th Street, New York, New York 10024-5192 USA and*  
<sup>2</sup> *L. H. Bailey Hortorium, Cornell University, Ithaca, New York 14853, U.S.A.*

*Received for publication 20 November 1993; accepted 25 June 1994*

**Abstract**—A method is proposed to recode certain phylogenetic cost matrices in order to lessen the computational burden involved in the search for parsimonious cladograms. Multistate characters with complex costs among character states are converted to unordered multistate and binary variables with differential weights. These new variables can be optimized efficiently. The method has applications in cases where character states are distributed hierarchically and symmetrically. Cases of “inapplicable” states in morphology, and transition–transversion weighting in molecular sequence data are discussed in this light.

### Introduction

Multistate characters whose transformation costs among states are specified by a matrix of values—matrix characters (Sankoff and Rousseau, 1975) are computationally expensive to optimize. While binary, additive multistate and non-additive multistate characters can be optimized via simple bit-wise operations, the matrix characters require much more elaborate and time-consuming procedures (Sankoff and Rousseau, 1975; Williams and Fitch, 1988). Searches which take minutes for non-additive (unordered) characters can require hours or days when a more specific cost regime is postulated. Anyone who has used the “step matrix” options in PAUP (Swofford, 1993) is aware of this. Here, we propose a method by which matrix characters can be recoded and optimized economically in certain special cases. Fortunately, these cases include some of the most common uses of step matrices—“inapplicable” data, transition-transversion ratios and insertion–deletion event costs.

### “Hierarchical” Characters

The states of characters can be thought of as hierarchically or non-hierarchically arranged. Hierarchical characters are those whose states can be organized into a series of non-overlapping nested sets each of which contains several members (Fig. 1; these set designations need not be unique). These are the characters that can be recoded for advantages in computational speed. The basis of this hierarchical pattern is the organization of the various transformations into classes. Two commonly encountered situations in which characters states are treated as hierarchically arranged (in addition to simple additive—ordered—characters) are in the use of characters which may be inapplicable in some taxa due to primitive or derived loss and the analysis of nucleotide sequence data under certain evolutionary models.

<sup>3</sup> To whom correspondence should be addressed.

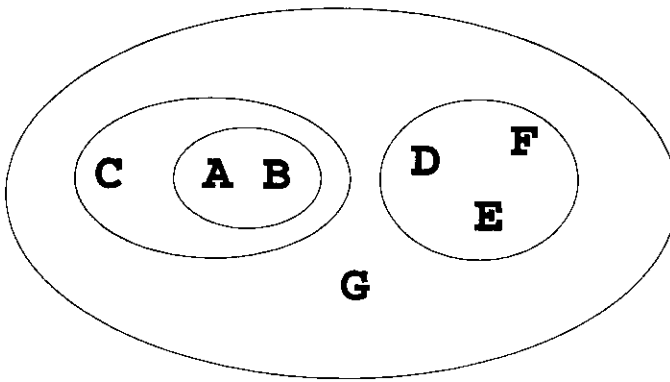


Fig. 1. A Venn-diagram scenario for hierarchically distributed characters states. Note that set size is not limited to two members.

### Inapplicable Characters

The problem of "inapplicable" data has recently received added attention (Nixon and Davis, 1991; Platnick et al., 1991). The problem, briefly stated, is that the inapplicable character states are not missing data but are treated this way in most analyses. The inapplicable state is observed, fixed and differentiated from any of the other observed states, it is just difficult to relate to other states. The coding method described here can accommodate the distinction between missing and inapplicable characters, and hopefully address some of the problems inherent in these character types.

A character which varies within the ingroup (hence is informative at resolving the relationships) may be absent in the outgroup. Wing venation patterns in insects are such a situation. The pattern of veins in the wings of insects vary tremendously. The sister taxon to winged insects does not possess wings, hence venation characters are "inapplicable". Currently, the outstate absence of wings would be treated as missing information even though clearly this is inappropriate. One way around this is to specify elaborate transformations in a matrix format among all the states (no wings, wings with veins unbranched, wings with veins branched one way, and wings with veins branched second way) incurring all the computational overhead implicit in this type of analysis. The coding method proposed here would simply recode this single character into several characters (depending on the exact relationship among states).

### The Method

If four states were observed in the ingroup: (1) unmodified; (2) modified-a; (3) modified-b and (4) modified-c, and out-taxa existed without any recognizable state at all, several distinctions can be made among the character states. The first distinction is between inapplicable and applicable states. The second is between modified and unmodified states, and the third is among modified "a", "b" and "c" (Fig. 2). Instead of erecting a matrix of cost values with many elements, three characters would be required in this recoding (if modifications "a", "b" and "c" are treated without ordering).

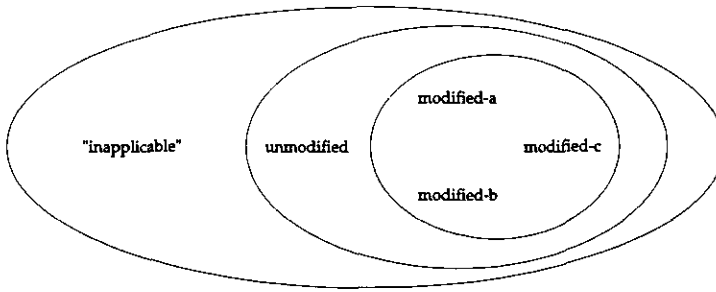


Fig. 2. A diagrammatic representation of the relationships among "inapplicable", "unmodified", "modified-a", "modified-b" and "modified-c" character states.

A more complex scenario is presented in the case of nucleic acid sequences where there are five states (four nucleotide bases and a gap character) with many types of transformation which may be treated differentially. These five states yield 10 possible symmetrical transformations (where  $\text{cost}_{ij} = \text{cost}_{ji}$ ) where  $n$  is the number of states. One class of transformation might be all those changes which occur between sequence gaps and nucleotides. There are then two types of characters: gaps and nucleotides. Furthermore, among these nucleotides there are two classes of bases: purines and pyrimidines—another class distinction. Finally, within purines there are two elements: Adenine and Guanine, and within pyrimidines: Thymine and Cytosine. A Venn-diagram representation (Fig. 3) shows the non-intersecting sets created by these distinctions among transformations. The transformation between these sets/elements are the classes of transformations. In this example, four transformation classes are present: (1) insertion-deletion events; (2) purine-pyrimidine-transversion events; (3) transitions between purines (Adenine and Guanine); and (4) transitions between pyrimidines (Thymine and Cytosine). This is equivalent to a matrix of transformation costs with each of the transformation values assuming one of the four costs specified for each type of transformation (Fig. 4). In this case, other transformation classes are possible (three GC versus two AT hydrogen bonds, for example) and alternate transformation distinctions can be made.

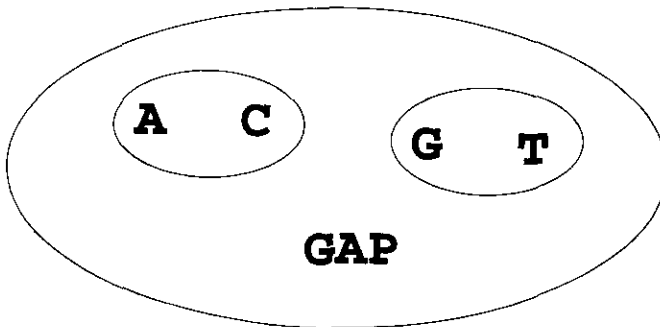


Fig. 3. A diagrammatic representation of the relationships among the five states of nucleic acid sequence characters.

	A	C	G	T	'-'
C	V				
G	I <sub>1</sub>	V			
T	V	I <sub>2</sub>	V		
'-'	D	D	D	D	

Fig. 4. The four types of character transformation which can be recoded for nucleic acid sequence characters.  $I_1$  is the transformation cost of purine (A-G) transitions.  $I_2$  is the cost of pyrimidine (C-T) transitions. V is the cost of transversions (purine-pyrimidine). D is the cost of an insertion-deletion event (this may be derived from sequence alignment).

### Recoding the Characters

The basis of the improvement in computational speed is the recoding of the matrix characters into a series of binary and non-additive multistate characters. The transformation classes form the structure of the recoding. Basically, each transformation type is represented by a binary character. These characters are then weighted in order to apply the appropriate cost to that variety of character change. In the case of molecular sequence data where insertion-deletion, transition-transversion and specific transition costs are specified, each of the original characters is recorded into five new variables (Fig. 5).

The first of these five variables is a non-additive multistate variable with five states—one for each of the four nucleotide and gap variables. This variable is assigned the base weight, usually one (1). The second variable keeps track of the insertion-deletion events. Each of the nucleotides is coded a "0" and gaps as "1". The weight assigned to this variable is more involved than for the base (first) variable. This is due to the fact that any transformation between gaps and nucleotides has already been counted once in the base variable. Hence, the insertion-deletion variable is given the weight of the cost associated with insertion-deletion events minus the base cost:

$$\begin{aligned} \text{Index variable weight} &= \text{insertion deletion transformation cost} \\ &\quad - \text{base transformation cost.} \end{aligned}$$

The same logic to avoid overcounting is applied to the following class variables. The third new variable codes for transversion events. Purines (A and G) are coded as "0", pyrimidines (C and T) as "1" and gaps as "?" (missing). The weight assigned

	Base (B)	InDel (ID)	Ti-Tv (TT)	A-G (AG)	C-T (CT)
A	0	0	0	0	?
C	1	0	1	?	0
G	2	0	0	1	?
T	3	0	1	?	1
GAP	4	1	?	?	?
Weight	B	ID-B	TT-B	AG-B	CT-B

Fig. 5. The recoding-reweighting scheme proposed in the text. Each of the four nucleotides and sequence gap has a unique coding. The five new characters are weighted based on the specified transformation costs. The base cost B could be set equal to that of the purine transition (AG). When this is done, the weight of the purine transition character becomes zero and the character can be ignored.

	A	C	G	T
C	3			
G	1	3		
T	3	2	3	
'-'	6	6	6	6

Original Sequence	Recoded Characters		
AAC	0000?	0000?	101?0
ACG	0000?	101?0	2001?
CCG	101?0	101?0	2001?
CT- (gap)	101?0	301?1	41???
 weights	 15201	 15201	 15201

Fig. 6. An example of the recoding–reweighting procedure. The matrix specifies transformation costs among the nucleotide (and gap) variables. The recoding procedure is applied to the sequences yielding a new data matrix with appropriate weights.

to this variable is the transversion cost minus the base weight (again because of counting in the base variable):

$$\text{Transversion variable weight} = \text{transversion cost} - \text{base transformation cost.}$$

The final two variables are the transitions from between purines (A and G) and pyrimidines (C and T). In the case of purine transitions, Adenine becomes “0”, Guanine “1” and pyrimidines and gaps “?”. The opposite occurs with pyrimidines with C and T denoted by “0” and “1” and the rest as “?”. Both of these variables are weighted as the transformation cost between purines or pyrimidines less the base weight (example in Fig. 6; in practice, one of the transitions can be assigned the base weight then that variable assumes a zero cost and is ignored).

These new variables can then be optimized as non-additive multistate (the “base” variable) and simple binary (the rest). It should be noted that these weights may assume negative values. As an example, if gap changes were to be assigned zero weight, the index recoded variable (the second) would be given a weight of  $-B$ . This would result in the gap transformations being counted in the first, base variable and then removed by the negatively weighted index variable.

TABLE 1  
Heuristic search timings for “step matrix” and recoded characters

Number of taxa	Length of tree	Number of trees	Rearrangements	Recorded time (s)	“Step matrix” time (s)
13	191	2	1 448	5.1	7.1
17	287	10	17 112	27	84
21	328	10	32 680	50	181
25	375	16	88 221	125	494
29	432	16	139 121	195	839
58 <sup>1</sup>	432	72	3 273 512	3448	15 558

<sup>1</sup> The 58 taxon data set is simply a doubling of the 29-taxon set.

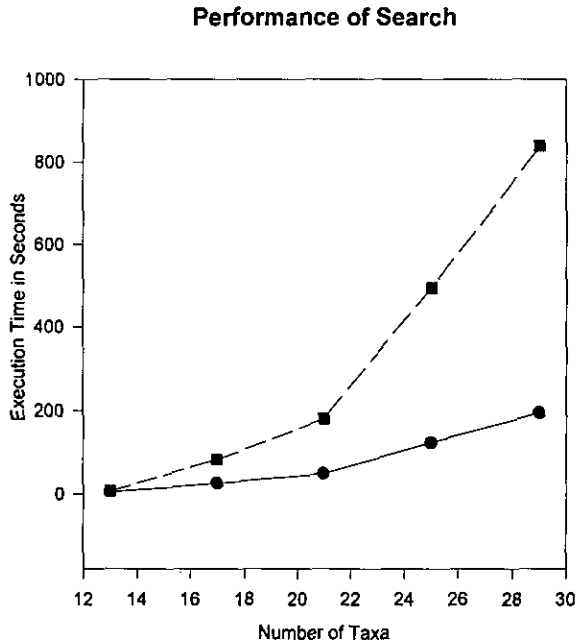


Fig. 7. A plot of the search times for both "step matrix" (dashed line and boxes) and the recoding method proposed here (solid line and circles). Search parameters are described in the text.

In the worst case, the effort expended in a search would be increased by a factor of four (when the base weight is assigned one of the transition variable weights). Most situations, however, would entail a much smaller premium. This is due to the likelihood that many of the recoded variables will be uninformative (autapomorphic or invariant) hence do not require repetitive optimization. A test data set of 29 hemipteran insect taxa (Wheeler et al., 1993) was subjected to this procedure. The data consisted of 31 morphological characters two of which were ordered multistate and 669 aligned nucleotides from nuclear 18S rDNA. Results from searches using PAUP ver. 3.1.1 (Swofford, 1993) on a MAC IICI, using the "closest" addition sequence with "hold" set to 5 and "TBR" branch swapping are summarized in Table 1 and Fig. 7. In each of the searches both coding methods yielded identical results not only in the length and number of trees but also in the number of rearrangements performed. Since the recoding method presented here does not alter tree length in any way, this is to be expected.

### Conclusions

The simple method for recoding matrix characters presented here cannot be used in all cases. However, when the character states are distributed hierarchically and symmetrically, this form of coding can lessen computation times considerably over "matrix" or "general" parsimony optimizations. The only effect of the recoding procedure is to accelerate the search for parsimonious solutions. The length, number and topologies of these solutions are unchanged. This coding system can accommodate the vast majority of cases in which these Sankoff characters are required, especially in the analysis of molecular sequence data.

### Acknowledgements

We would like to acknowledge Ranhy Bang, James Carpenter, Rob DeSalle, Steven Farris, John Gatesy, Pablo Goloboff, Cheryl Hayashi, Paul Vrana and Michael Whiting for discussion of these ideas.

### REFERENCES

- NIXON, K. C. AND J. I. DAVIS. 1991. Polymorphic taxa, missing values, and cladistic analysis. *Cladistics* 7:233–241.
- PLATNICK, N. I., C. E. GRISWOLD AND J. A. CODDINGTON. 1991. On missing entries in cladistic analysis. *Cladistics* 7:337–343.
- SANKOFF, D. D. and P. ROUSSEAU. 1975. Locating the vertices of a Steiner tree in arbitrary space. *Math. Prog.* 9:240–246.
- SWOFFORD, D. L. 1993. PAUP, Version 3.1.1, program and documentation. Champaign, Illinois.
- WHEELER, W. C., R. BANG AND R. T. SCHUH. 1993. Cladistic relationships among higher groups of Heteroptera: congruence between morphological and molecular data sets. *Ent. Scand.*, 24:121–138.
- WILLIAMS, P. L. and W. M. FITCH. 1988. Finding the minimal change in a given tree. *In*: B. Fernholm, K. Bremer, L. Brundin, H. Jörnvall, L. Rutberg and H.-E. Wanntorp (eds). *The Hierarchy of Life*. Elsevier, Amsterdam, pp. 453–470.

