ACADEMIC PRESS

Available online at www.sciencedirect.com

SCIENCE DIRECT®

Cladistics 19 (2003) 254–260

Cladistics

www.elsevier.com/locate/yclad

Iterative pass optimization of sequence data

Ward C. Wheeler*

Division of Invertebrate Zoology, American Museum of Natural History, Central Park West, 79th St., New York, NY 10024-5192, USA

Accepted 2 April 2003

Abstract

The problem of determining the minimum-cost hypothetical ancestral sequences for a given cladogram is known to be NP-complete. This "tree alignment" problem has motivated the considerable effort placed in multiple sequence alignment procedures. Wheeler in 1996 proposed a heuristic method, direct optimization, to calculate cladogram costs without the intervention of multiple sequence alignment. This method, though more efficient in time and more effective in cladogram length than many alignment-based procedures, greedily optimizes nodes based on descendent information only. In their proposal of an exact multiple alignment solution, Sankoff et al. in 1976 described a heuristic procedure—the iterative improvement method—to create alignments at internal nodes by solving a series of median problems. The combination of a three-sequence direct optimization with iterative improvement and a branch-length-based cladogram cost procedure, provides an algorithm that frequently results in superior (i.e., lower) cladogram costs. This iterative pass optimization is both computation and memory intensive, but economies can be made to reduce this burden. An example in arthropod systematics is discussed.

Systematists struggle with sequence homology. Traditionally, some form of multiple sequence alignment is performed, perhaps manually, to create the putative homology statements that many phylogenetic analysis programs require. Wheeler (1996) proposed a procedure (optimization alignment or direct optimization; DO) to move the alignment problem to one of cladogram optimization. In DO, hypothetical ancestral sequences are created through comparison of descendent sequences and insertion–deletion (indel) events are treated as another form of transformation in a series of potential transformation events. This requires precise statements concerning the relative cost of different types of transformations and the explicit inclusion of indel events in the calculation of cladogram cost. DO generally yields more parsimonious (lower cost) cladograms than those based on multiple alignment (Giribet et al., 2002). The algorithm is not exact, however, and the general problem is known to be NP-complete (Wang and Jiang, 1994). DO provides an upper bound on the true minimum cost.

In their examination of the multiple sequence alignment problem, Sankoff et al. (1976) proposed an exact solution through the use of n-dimensional string matching, given a known scheme of phylogenetic relationships. For n sequences of length m that method requires storage on the order m^n and would require $2^n - 1$ calculations at each cell in the n-dimensional structure. To be used to identify an optimal tree, furthermore, this procedure would have to be repeated for each possible tree. Clearly, this is intractable for even the smallest of data sets. Sankoff et al. proposed a heuristic procedure, however, based on iteratively solving the median problems at each internal node (Fig. 1). The median problem is basically a three-sequence (i.e., three-dimensional) version of the dynamic programming Needleman–

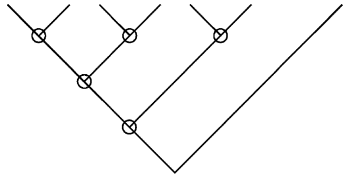

Fig. 1. Cladogram of seven taxa. The internal nodes are circled.

* Fax: 1-212-769-5233.
E-mail address: wheeler@amnh.org.

0748-3007/03/$ - see front matter © 2003 The Willi Hennig Society. Published by Elsevier Science (USA). All rights reserved.
doi:10.1016/S0748-3007(03)00047-1

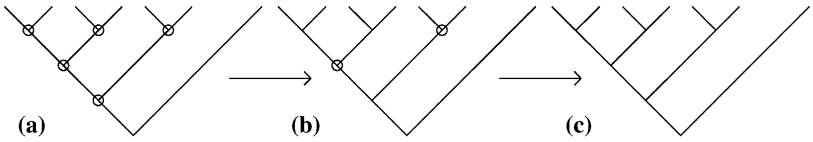

Fig. 2. Iterative improvement algorithm. The internal nodes (circled) are repeatedly determined (a to c) until a round of calculation results in no changes in the hypothetical ancestral sequences postulated.

Wunsch algorithm (Needleman and Wunsch, 1970). Each internal nodal sequence alignment is recalculated until a pass over the entire tree results in no novel changes and stability is achieved (iterative improvement; Fig. 2). The stable nodes are then used to generate a multiple sequence alignment. As part of this alignment process, "protosequences" akin to hypothetical ancestral sequences are also created (by majority of the three terminal states or left ambiguous).

A similar median problem solution can be created using a three-sequence version of DO that will then explicitly generate hypothetical ancestral sequences in a generalized version of Wheeler's (1996, 2002) method. The same iterative improvement algorithm can then be applied to the nodes, initialized with DO-derived hypothetical ancestral sequences (akin to the use of median genomes in the breakpoint analysis of Sankoff and Blanchette (1997)). The stable inferred sequences that are used to calculate the overall cladogram cost given these internal sequences do not contain ambiguities or "gaps" as they may in Sankoff et al.'s (1976) procedure. An implementation of the Sankoff et al. procedure based on the combination of a three-sequence DO with iterative improvement (i.e., iterative pass optimization), though more elaborate and time consuming than standard DO, can result in less costly cladograms.

Method

There are four segments to the procedure: (1) initialization, (2) median problem, (3) iterative improvement with incremental optimization, and (4) cladogram cost calculation.

Initialization

There are several possible ways to initialize internal (hypothetical ancestral) sequences such that the median optimization can begin. A simple method might be to use the topologically nearest terminal (i.e., leaf) sequence, but this performs poorly in my experience (at least using POY; Wheeler et al., 2002). Two other, superior options are to use the preliminary or the final hypothetical ancestral sequences generated by DO. In practice, initialization with the final set of hypothetical ancestral sequences is more time consuming and provides little, if any, extra efficiency in the iterative improvement phase.

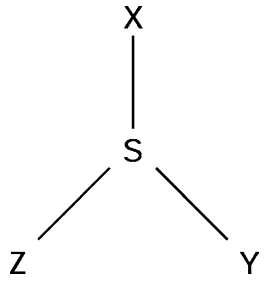

Fig. 3. The median sequence (S) of an internal node is determined by minimizing the sum cost to each of the adjacent nodal sequences (X, Y, and Z).

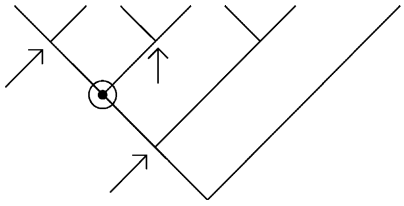

Fig. 4. If a hypothetical ancestral sequence (circled) changes when reoptimized during iterative improvement, the three adjacent nodes (arrows) must be reoptimized also.

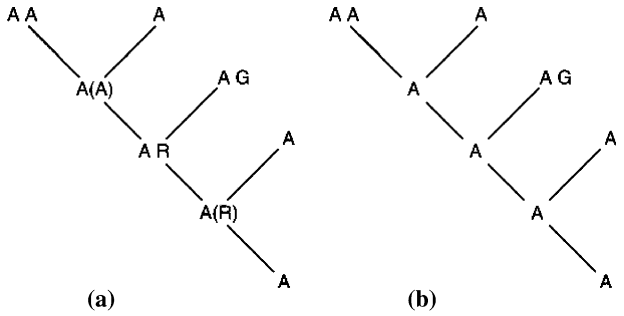

Fig. 5. Optimization of cladogram using direct optimization (a) Wheeler, 1996) and iterative pass optimization (b). The parentheses in a represent ambiguous optimization with respect to sequence length. This signifies an equally costly scenario during the downpass of DO.

Median problem and direct optimization

Sankoff and Cedergren (1973) defined a minimum mutation alignment method for three sequences. This method is basically a three-dimensional extension of the dynamic programming string match algorithm of Needleman and Wunsch (1970). At its core is the determination of a median sequence (S, see Fig. 3) among all

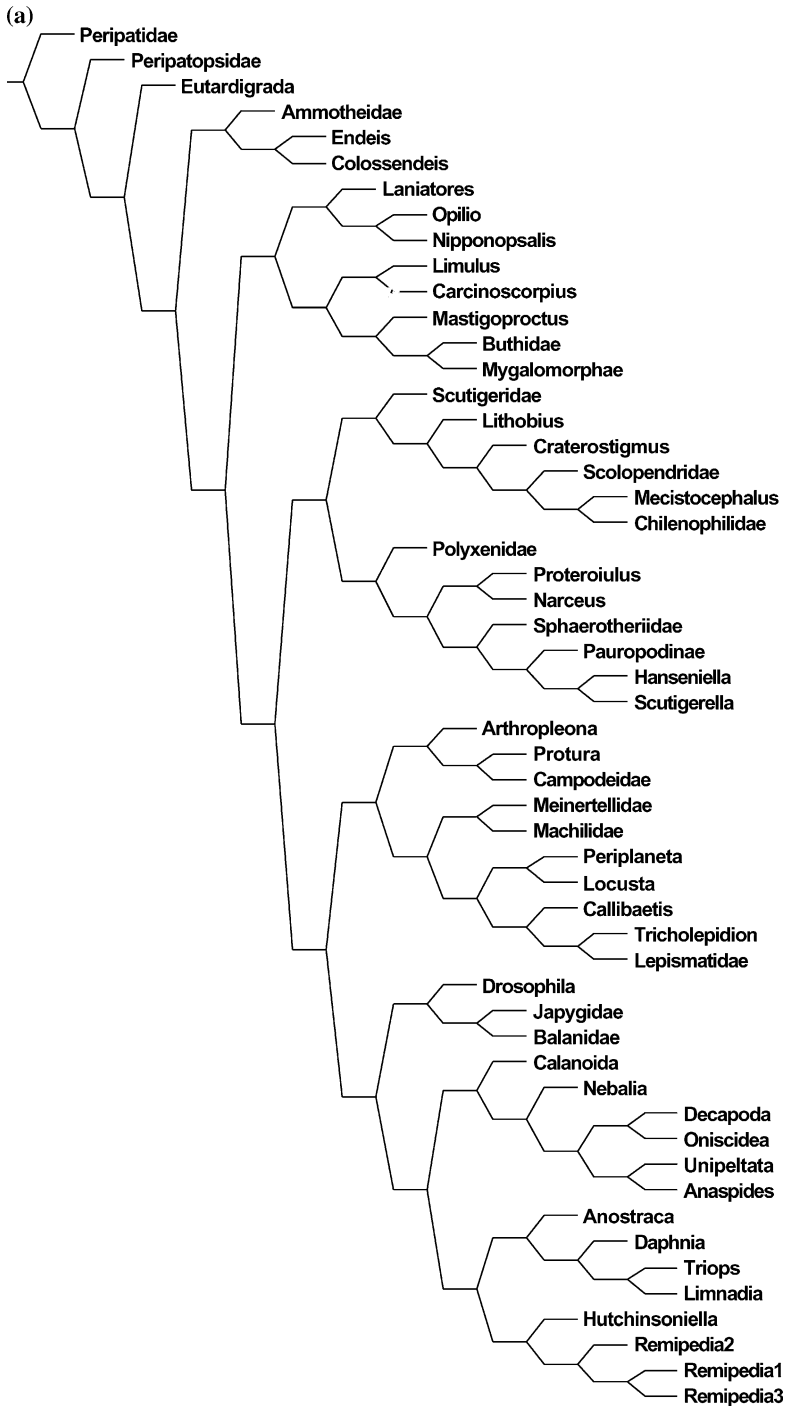

Fig. 6. Arthropod cladogram of Giribet et al. (2001). This cladogram weights all changes equally for a total cost of 27,393 steps for DO (a) and 27,198 steps for iterative pass (b).

sequences that, given a transformation cost matrix among A, C, G, T, and "gap," satisfies

$$\min(d(S, X) + d(S, Y) + d(S, Z)).$$

In the general (*n*-taxon) case, the cost function used to calculate the median sequence (and multiple alignment) is a "known" or at least prespecified phylogeny (Sankoff et al., 1976). Here, the multiple alignment is neither desired nor performed and the median sequence ("protosequence" of Sankoff et al.) is constructed by the three-dimensional version of DO (Wheeler, 1996, 2002),

```
for i = 0 to length X + 1 do
  for j = 0 to length Y + 1 do
    for k = 0 to length Z + 1 do
      M_cost(i, j, k) = min{
        M_cost(i − 1, j − 1, k − 1) + Cost(X_i, Y_j, Z_k);
```

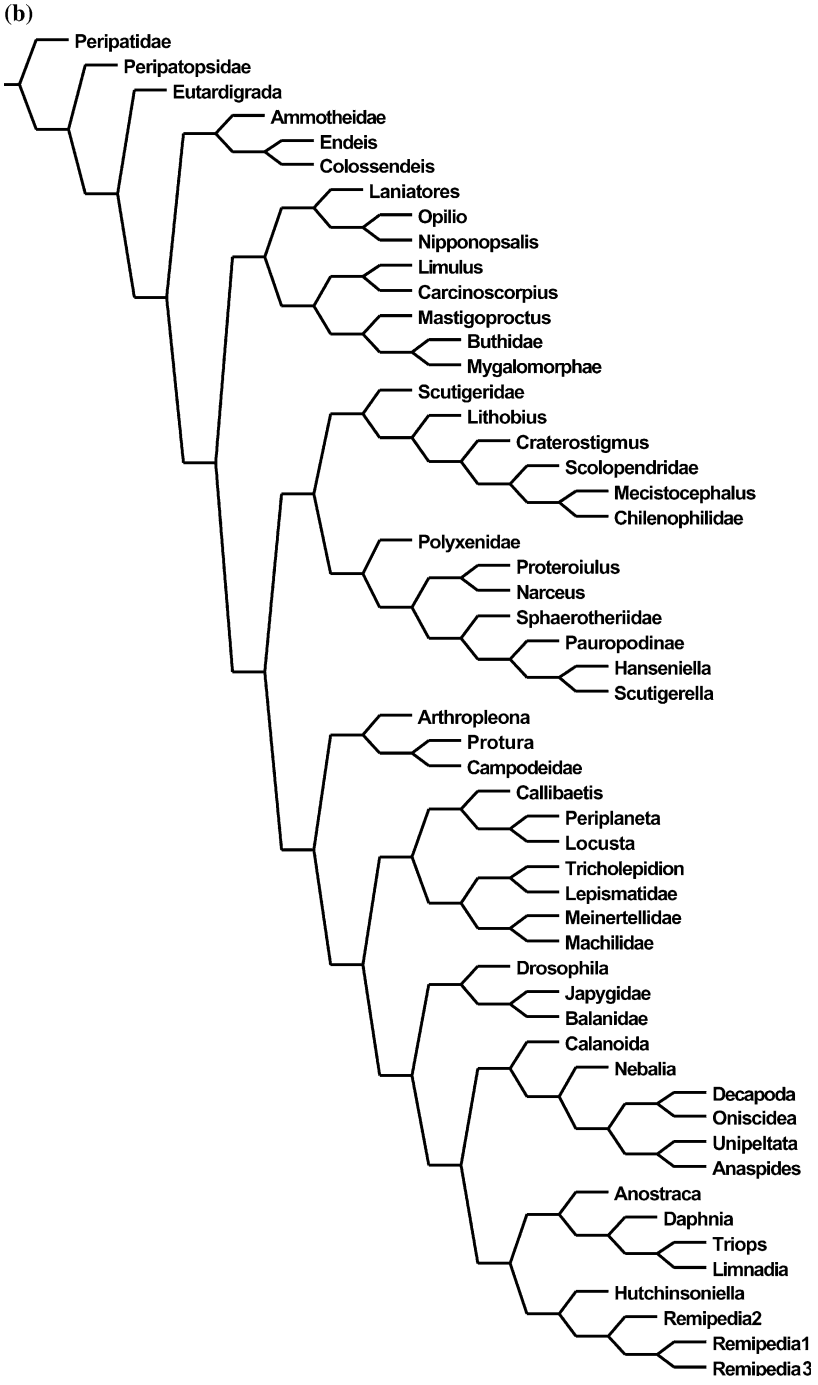

Fig. 6. (*continued*)

M_cost(i − 1, j − 1, k) + Cost(X_i, Y_j, gap);
M_cost(i − 1, j, k − 1) + Cost(X_i, gap, Z_k);
M_cost(i − 1, j, k) + Cost(X_i, gap, gap);
M_cost(i, j − 1, k − 1) + Cost(gap, Y_j, Z_k);
M_cost(i, j − 1, k) + Cost(gap, Y_j, gap);
M_cost(i, j, k − 1) + Cost(gap, gap, Z_k);
}
M_direction(i, j, k) = source cell to M_cost(i, j, k);
done
done
done,

where M_direction is used for the traceback (not required, but saves on calculations) and where cost (a, b, c) is the median cost among states a, b, and c (A, C, G, T, gap, or any combination of the five) for s (the median state) such that

$$s = \min(s \text{ in } A,\ C,\ G,\ T,\ \text{gap}) \min(d(s,a) + d(s,b) + d(s,c)).$$

During the traceback when the hypothetical ancestral sequences are created, "gaps" in median sequences are removed (s = "–"), and ambiguously determined median sequences are resolved to single nucleotides. This resolution can be accomplished in different ways, such as arbitrarily, randomly, to "accelerate" or "delay" transformation. Since the median problems are solved iteratively and this is a heuristic procedure, this choice may affect the final cladogram length calculations. In DO and other optimization procedures, these ambiguities are required to correctly calculate cladogram cost during the downpass, yet in this procedure this will cause incorrect cladogram cost calculation (underestimates) since the overall cladogram cost is derived from summing branch costs, as is explained below.

Iterative improvement

A heuristic solution for the *n*-taxon case can be constructed by solving the median problem above over the entire cladogram (*n*-2 internal nodes or vertices; Fig. 1). In each case, the median optimization would use the three adjacent nodes (whether terminal or DO initialized internal) to calculate the unique hypothetical ancestral sequence for that internal node. Since any change in the reconstructed sequence of a node will potentially affect those nodes adjacent to it, this process must be repeated until all the internal nodal sequences are stable. This is the iterative improvement method of Sankoff et al. (1976). Sankoff et al. (1976) reported that in their experience no more than five iterative rounds usually were required.

A form of incremental attribute evaluation (reviewed by Hudson (1991) and Gladstein (1997)) can be used to reduce the number of nodes calculated during each round of the iterative improvement. Since each internal node is directly connected to only three other nodes, a change in its sequence will require revisiting those three (or fewer if one or two are terminal) nodes (Fig. 4).

In my experience, rarely are more than three iterative passes required to attain stability when DO is used to initialize the internal nodes. Furthermore, the incremental attribute evaluation progressively reduces the number of median problems required by successive iterations such that usually fewer than $2n$ (n = number of terminals) median optimizations are required. This is, of course, a far cry from the average of two or so nodal optimizations required when incremental optimization is applied to character data (Gladstein, 1997), but it is manageable.

Cladogram cost calculation

The fourth and final step is cladogram cost calculation based on the optimized hypothetical ancestral sequences. This is accomplished quite simply by summing the pairwise cost of each ancestor–descendant path over the $(2n - 3)$ branches of the tree. This manner of calculation requires the specificity of hypothetical ancestral sequences discussed above. If these (inferred, not observed) sequences were to contain ambiguities (to reflect legitimate multiple equally costly nucleotide states or uncertainty in length), the cost calculation would be erroneously low. The cost can be further refined by adding a step that calculates the cladogram cost using the homologies implicit in the cladogram and the dynamic programming procedure of Sankoff and Rousseau (1975; Wheeler, in press; "exact" option in POY).

Results

An example

Consider five simple sequences related by a pectinate cladogram (Fig. 5). When optimized under DO with indel cost 2 and base substitution cost 1, the cladogram cost is 5 weighted steps. From the downpass states (Fig. 5a), it is clear that the greediness of the algorithm has resulted in over counting cladogram cost. Even though it is obviously more efficient to posit two insertions (A and G in five and three), DO presumes an extra nucleotide change in comparison to that required by independent insertions.

When iterative pass optimization is applied, the cladogram cost is 4 weighted steps, correctly minimizing the weighted events. This is because the first round of median state calculations converts the DO downpass preliminary states to single "A"s (Fig. 5b) immediately. These are stable to the next round of median state optimizations and the cladogram cost is calculated correctly.

An empirical example comes from the arthropod data set of Giribet et al. (2001). These data contain information from 54 taxa of both anatomical and nonsequence molecular variation (303 characters) and molecular (eight loci) origin. The original cladogram cost, with indels, nucleotide changes, and morphological transformations weighted equally was 27,393 steps (Fig. 6a). This cladogram, when diagnosed with iterative pass optimization yields a cladogram cost of 27,207 steps. Branch swapping (TBR) on this cladogram using iterative pass optimization and the "exact" option resulted in a further reduction in cladogram cost to 27,198 steps. This made several topological changes but kept the Tetraconata of the original analysis (Fig. 6b).

Shortcuts, speedups, time, and memory

That reduction of 0.69% in number of steps came at the premium of a 29-fold increase in execution time (419.5 s versus 14.62 s on a 500-mhz PIII) and a 9-fold increase in memory utilization (288 MB versus 31 MB). Both the execution time and the storage requirements of this op-

timization procedure are much greater than those of DO. In essence, both are proportional to the cube of the lengths of sequences as opposed to their square. These requirements can be reduced tremendously by taking advantage of several properties of the data at hand.

First, given that there are only five states (A, C, G, T, and "gap"), all triplet combinations of states and gaps in the three sequences can be precalculated to reduce the execution time of the three-dimensional string match in both the cost phase and the determination of hypothetical ancestral sequences. This requires that 31^3 (29,791) costs and outcomes be calculated and stored, certainly not an onerous burden. This reduces the individual base median problem to a simple look-up.

Second, when sequences are similar in length, the algorithm of Ukkonen (1985) can be used to reap great economies. The general Needleman and Wunsch (1970) algorithm implicitly examines all possible correspondences among the three input sequences, even those with absurdly large numbers of indels. Given that systematically related sequences are extremely unlikely to require such extensive "gapping," a temporary maximum gap size can be set with only those scenarios within a box whose sides are defined by the maximum gap size being examined (which can be an extremely reduced space; Fig. 7). After the optimization is performed, if the number of indels implicit in the resultant hypothetical ancestral sequence is greater than the current maximum gaps size, the gap size is increased (doubled) and the process repeated until the maximum gaps size is no longer limiting. This can reduce storage and execution times by orders of magnitude.

Third, and most obviously, given the strong dependence of both execution time and storage on sequence length, splitting up sequences on the basis of primer areas or other landmarks will be of great benefit. If long sequences were divided into "n" equal segments, speedups of the order of n^2 may be realized. Care is needed, however, or such splitting may result in constraining the homology search aspect of the procedure. This can be checked, however, by rediagnosing a cladogram based on chopped up sequences with the sequence in its entirety. If the splitting has not affected homology and has only aided in accelerating the search, cladogram rediagnosis will return the same cost in both cases. This could also be used as a heuristic search technique, with segmented sequences used for initial searches, which would then be refined using the contiguous data.

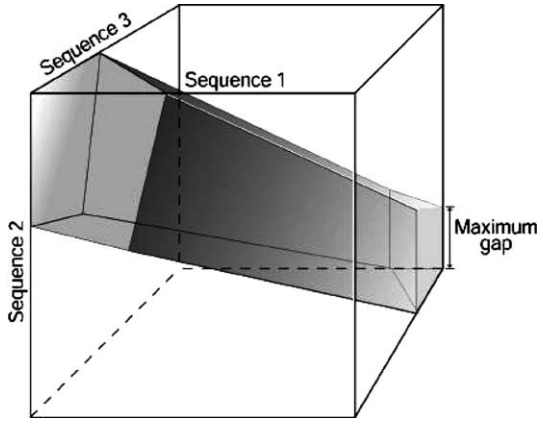

Fig. 7. The Ukkonen (1985) string-match shortcut applied to three sequences simultaneously. The reduction in volume defined by the maximum gap yields the speed-up. The maximum gap is expanded until it does not limit the determination of the hypothetical ancestral sequences from the three input sequences.

Discussion

Although proposed originally for multiple sequence alignment, the median optimization of Sankoff and Cedergren (1973) has been used most in the optimization of genomic rearrangement data. When coupled with the incremental attribute optimization and the initialization values of DO with its rationale for the emphasis on cladogram cost estimation, the resultant iterative pass optimization is an extremely effective means of determining parsimonious cladogram costs. Especially when coupled with full dynamic programming cost determination (Sankoff and Rousseau, 1975), this procedure can optimize cladograms more efficiently (i.e., with lower cost) and potentially ameliorate some of the root-based greediness of the DO algorithm. The extreme computational cost of the operation may reduce its heuristic use to a refinement step in a real search procedure, but the reduction in cladogram cost is impossible to ignore.

Acknowledgments

I acknowledge the help of Cyrille D'Haese, Jan De Laet, Gonzalo Giribet, Daniel Janies, Taran Grant, and Leo Smith in critiquing the manuscript and Steven Thurston for the good art work. Two anonymous reviewers also provided comments and suggestions which improved the manuscript considerably. Grant support has been generously supported by NSF Systematic Biology and NASA Fundamental Space Biology Program.

References

Giribet, G., Wheeler, W.C., Muona, J., 2002. DNA multiple sequence alignments. In: Desalle, R., Giribet, G., Wheeler, W. (Eds.), Molecular Systematics and Evolution: Theory and Practice. Birkhäuser, Basel, pp. 107–114.

Giribet, G., Edgecombe, G.D., Wheeler, W.C., 2001. Arthropod phylogeny based on eight molecular loci and morphology. Nature 413, 157–161.

Gladstein, D.S., 1997. Efficient incremental character optimization. Cladistics 13, 21–26.

Hudson, S.E., 1991. Incremental attribute evaluation: a flexible method for lazy update. ACM Trans. Program. Lang. Syst. 13, 315–341.

Needleman, S.B., Wunsch, C.D., 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. J. Mol. Biol. 48, 443–453.

Sankoff, D., Blanchette, M., 1997. The median problem for breakpoints in comparative genomics. Computing and Combinatorics, 3rd Ann. Int. Conf. COCOON 97 1276, 251–263.

Sankoff, D.D., Cedergren, R.J., 1973. A test for nucleotide-sequence homology. J. Mol. Biol. 77, 159–164.

Sankoff, D.D., Rousseau, P., 1975. Locating the vertices of a Steiner tree in arbitrary space. Math. Progam. 9, 240–246.

Sankoff, D., Cedergren, R.J., Lapalme, G., 1976. Frequency of insertion–deletion, transversion, and transition in the evolutions of 5S ribosomal RNA. J. Mol. Evol. 7, 133–139.

Ukkonen, E., 1985. Finding approximate patterns in strings. J. Algorithms 6, 132–137.

Wang, L., Jiang, T., 1994. On the complexity of multiple sequence alignment. J. Comput. Biol. 1, 337–348.

Wheeler, W.C., 1996. Optimization alignment: the end of multiple sequence alignment in phylogenetics? Cladistics 12, 1–9.

Wheeler, W.C., 2002. Optimization alignment: down, up, error, and improvements. In: Desalle, R., Giribet, G., Wheeler, W. (Eds.), Techniques in Molecular Systematics and Evolution. Birkhäuser, Basel, pp. 55–69.

Wheeler, W.C., Implied alignment: a synapomorphy-based multiple sequence alignment method and its use in cladogram search. Cladistics, in press.

Wheeler, W.C., Gladstein, D.S., Laet, J.D., 2002. POY. Version 3.0. ftp.amnh.org/pub/molecular/poy. Documentation by Daniel Janies and Ward Wheeler.