Alignment characters, dynamic programming and heuristic solutions

Ward Wheeler

Department of Invertebrates, American Museum of Natural History, Central Park West @ 79th, Street, New York, NY 10024, USA

Summary

The method of direct optimization of nucleic acid sequences proposed by Wheeler (1996) is elaborated and explained in light of dynamic programming procedures. An exact solution to the problem of phylogenetic reconstruction of unequal-length sequences is described, and its impracticality demonstrated. A branch-and-bound procedure is elucidated to accelerate this process. Additionally, a series of heuristic solutions are defined for this general problem, allowing for both significant decrease in computational effort and integration with existing algorithmic economies. Finally, potential implications of this method are discussed in light of putative long-branch attraction problems.

Introduction

Phylogenetic reconstruction of molecular sequence data requires two types of transformational events: (i) nucleotide (or amino acid) substitution and (ii) nucleotide (or amino acid) insertion and deletion. Standard procedures of character optimization and diagnosis (Farris, 1970; Fitch, 1971; Sankoff and Rousseau, 1975; Sankoff and Cedergren, 1983) accommodate character state transformation easily. Insertion-deletion events, however, are not so simply explained. Normally, a sequence alignment procedure is performed to establish the putative homologies which are required for standard character analysis (Feng and Doolittle, 1987, 1990; Hein, 1989, 1990; Higgins and Sharp, 1988, 1989; Wheeler and Gladstein, 1992, 1994), and gaps are inserted and treated as a fifth state. A heuristic method has been proposed (Wheeler, 1996) to diagnose cladogram topologies directly without the intervening multiple-alignment step. This discussion seeks to place the direct optimization method within the context of dynamic programming and to present both exact and heuristic solutions to the problem.

R. DeSalle, B. Schierwater (eds) Molecular Approaches to Ecology and Evolution

The problem

Unlike many sources of information, molecular sequence data present not only variation in character state, but also in character number. That is, the number of characters presented by terminals may vary because sequences frequently differ in length. A cartoon of this situation is illustrated in Figure 1. Normally, the four terminal sequences (A, AA, AG and A), would undergo multiple alignment (Fig. 2), and some sort of dynamic programming (Sankoff and Rousseau, 1975; Sankoff and Cedergren, 1983) or short-cut procedure (Fitch, 1971) would be performed to diagnose the length or cost (in evolutionary steps) of any dendrogram (Fig. 3). Part of this process involves the insertion of placeholders – gaps ("-") to make the individual characters comparable. The sequence gaps are not observations, but the residue of insertion-deletion events required by the variation in sequence length. This introduces a certain epistemological inconsistency (Wheeler, 1996), treating what are in essence transformational events as equal to observations (such as A, G etc.).

The optimization procedure of Wheeler (1996) seeks to avoid this inconsistency and simplify the process by generalizing optimization to include insertion-deletion events. Direct optimization yields more intelligible and frequently more parsimonious results (Fig. 4).

Dynamic programming

An exact solution to the problem of sequence length variation can be achieved by recasting the diagnosis of cladograms from one of sequentially optimizing a series of simple characters (including gaps) to one of relating a single immensely (but not infinitely) complex character. In essence, all imaginable sequences (of all lengths) are possible states of this single character. The objective, then, is to create the most parsimonious character transformation series.

Within this framework, dynamic programming can be applied to determine the exact solution. As with the steps involved in optimizing Sankoff-type characters, the first issue is to define all the possible character states at each node. For the standard approach based on multiple alignment this would be simple – five states: A, C, G, T and gap. Here, however, the character correspondences and ancestral sequence length are unknown. The length of the hypothetical ancestral sequences is bounded by length zero at one extreme – no bases (sequences arise *de novo* repeatedly), and by the sum of the lengths of all the input sequences, since no parsimony-based operation could yielded anything longer. Since each of the four bases (with nucleic acids – gaps do not exist in real sequences after all) is possible at each position, the total number of possible states is: #states = $\Sigma 4^k$ where k is summed from 0 to the sum length of all input sequences. For three sequences of length four, there would be 22,369,621 possible states. In reality this number would be more tightly bound-

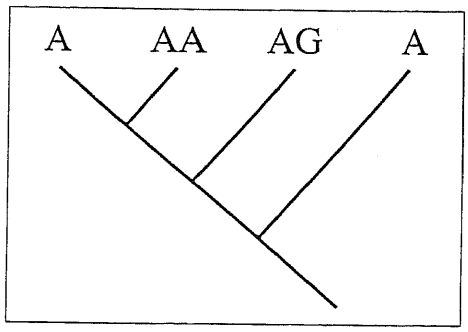

Figure 1. Four simple terminal sequences (A, AA, AG and A) related by cladogram.

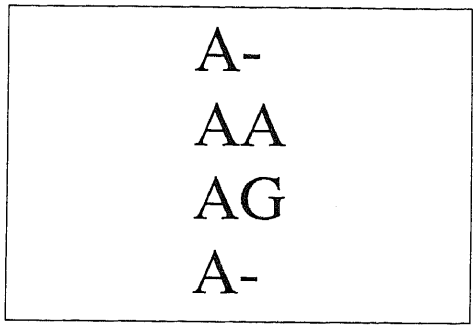

Figure 2. The minimum-cost multiple alignment for the four sequences of Figure 1.

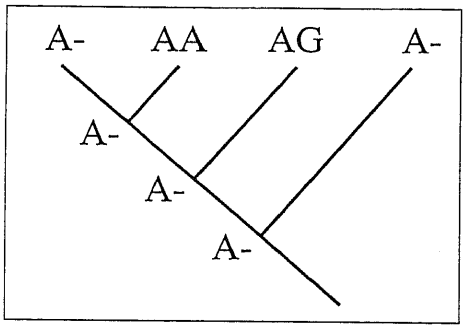

Figure 3. Standard method diagnosis of the cladogram and taxa of Figure 1 using the multiple alignment of Figure 2.

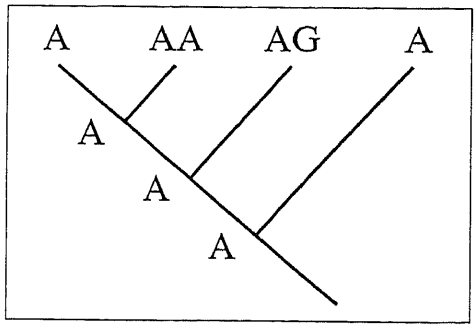

Figure 4. Diagnosis of taxa and cladogram of Figure 1 by the direct method proposed here.

ed. Since any insertion which would simultaneously occur in two descendent lineages would be transferred to the line leading to their ancestor, the minimum length of an ancestral sequence would be the lower of the two descendants and the maximum length would be their sum. Hence: #states = $\Sigma 4^k$ again, however, where k is summed from min (descendent 1, descendent 2) to the sum length of the two descendants (descendent 1 + descendent 2) sequences. For three sequences of length three, four and five there would be from 21,824 to 349,504. These are still large numbers, but orders of magnitude smaller than the exhaustive case.

Once all the possible cases are enumerated (and a suitable cost matrix relating all the possible states to each other has been specified), standard dynamic programming in a "down pass" will yield the most parsimonious cladogram length and ancestral state assignments. Unfortunately, this will take a long time. Each internal

node will require approximately twice the number of states squared operations, from each of the possible combinations of the two descendent states and each of the ancestral states (this will be smaller if the descendants are terminal taxa). Clearly, this is impracticable for all but the smallest cases.

Branch and bound

Although the exhaustive optimization described above is absurdly involved, the number of considered ancestral states and operations can be further limited. Since the cost of each path through each possible state is a monotonically increasing function (lengths cannot go down), an upper bound on cost can exclude the vast majority of possible states (Hendy and Penny, 1982). Longer nucleic acid strings require (in general) more indels, hence the longer character states (i.e. sequences) are likely to be excludable very early in the process.

The procedure outlined earlier (Wheeler, 1996) can act as such a bound since it yields upper-bound estimates of tree length. In the example of Figure 1, an upper bound of two insertion-deletion events and one base transformation (assuming insertion-deletion events are assigned greater cost than base substitution) would be postulated (Fig. 5). This would exclude ancestral state reconstructions of length greater than two. Any higher number would exceed the bounded cost. The number of possible character states would be limited to six (A, G, AA, GG, AG and GA). Although this would result in a further dramatic reduction in computational complexity, for real-world cases the reduction would likely be from an extremely absurd situation to one which is merely absurd. Most likely, we will be limited to heuristics.

Heuristics

Wheeler (1996) presented a procedure to improve the initial estimates of cladogram lengths. The optimization procedure is a straightforward generalization of nonadditive or unordered optimization (Farris, 1970; Fitch, 1971). The down pass optimization is depicted in Figure 6. In this case, there are five sequences of unequal lengths. Without prior knowledge of base correspondences (alignment), it is impossible to construct a hypothetical ancestor or determine how costly that operation is (in terms of transformations). Hence, correspondences (putative homologies) must be constructed as we go down the tree for the comparisons made at each node. In essence, all possible schemes of comparison must be examined for each node and that scheme which minimizes the number of minimum-cost union events (weighted by the cost of a base transformation) and insertions and deletions (weighted by the gap cost) is assigned to the node. In this way, the most efficient (i.e. lowest cost) hypothetical ancestor is constructed.

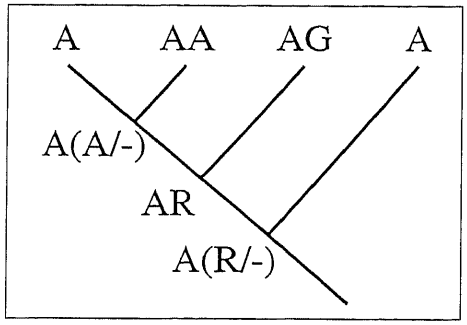

Figure 5. Down-pass optimization of Figure 1 using the method of Wheeler (1996) used as an upper bound for exact analysis.

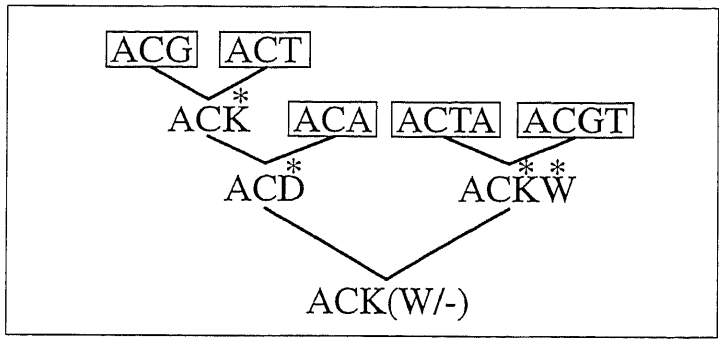

Figure 6. A more complex example of the direct-optimization procedure. Boxes surround terminal taxa, asterisks (*) denote base substitutions, parentheses insertion-deletion events.

As with non-additive analysis, the procedure begins at the top of the tree (or more specifically at an ancestral node of two terminal taxa) with the sequences ACG and ACT. The construction of the hypothetical ancestor can be broken down into two operations. The first can be thought of as an alignment step. The sequences are aligned to minimize the weighted cost of indels and base transformations as determined by union/intersection counts. This is performed with the proviso that if a gap is inserted in one sequence to correspond to a gap in the other, this is done at no cost (the sequences would have a nonempty intersection). Each possible alignment is considered (via dynamic programming) as in the Needleman and Wunsch (1970) procedure. In this example, the best alignment contains one base transformation and no gaps (Fig. 6). In the second operation, the hypothetical ancestor is constructed from this alignment by taking the union/intersection position by position along the sequence yielding ACK (K = T or G in IUPAC parlance). This hypothetical ancestral sequence is then compared with the next terminal ACA yielding another hypothetical ancestral sequence, ACD at the cost of another base transformation.

On the other side, a similar operation comparing ACTA with ACGT yields ACKW. Proceeding to the next node, ACK is compared with ACKW. Here, the alignment step requires no nucleotide transformations but does require an insertion-

deletion event, yielding an ancestral sequence with ambiguities in both base assignment and length. The first three bases are reconstructed as before ACK. The reconstruction of the fourth, however, is more complicated. This optimization would allows each of the three possibilities A, T, or GAP. This signifies that the position may contain either an A or a T or just not be there at all. The entire topology has been diagnosed at a cost of one insertion deletion events and four base transformations.

Greediness and shortcuts

In assigning states to hypothetical ancestral sequences, a method has been used of necessity which may introduce error in calculating the tree length. When proceeding down the tree, nucleotide assignments cases may occur in which a nucleotide base (say A) is faced with a corresponding ambiguity in its putative sister taxon as to whether or not a base exists (say G or GAP). How do we determine the condition of the ancestral sequence? If all transformations were equal (including indel events, transitions and transversions), the ancestral condition would be the union of the three states à la Fitch. However, this is rarely the case. More frequently investigators postulate that indels are less plausible that base substitutions, that is the cost of an insertion or deletion is greater than that of a nucleotide substitution. In this case, the ancestral condition would then be assigned "R" for the union of A and G and the GAP possibility excluded, taking the lower-cost transformation as yielding the ancestral condition. It may be more globally parsimonious that the indel ambiguity not be removed at this stage, but the procedure will not foresee this. Hence, the operations described here may overestimate tree length. Analogous reasoning holds for choices where transition-transversion bias is involved. Identical behavior could be observed using the optimization procedure described here with sequences of identical length and comparing the results with a dynamically programmed tree length (via a Sankoff step-matrix procedure).

Since we can not know the future (further down) sequences, optimization of unequal length sequences requires this myopia. As an aside, this method is in essence a weighted nonadditive optimization. Given that the various transformation weights are known, all optimization events, their costs, and results can be calculated before the actual tree search. In this way, weighted step matrix parsimony calculations can be accomplished at considerable savings in cost (in my experience the general weighting comes at a cost of a low – ~5% – fixed premium on execution time). As mentioned before, the procedure is local, globally more parsimonious may not be considered (or even rejected). As a result, any error in length should be an overestimate. For the case where the sequences are equal in length, the results can be adjusted by full dynamic programming of individual candidate topologies.

This direct optimization procedure frequently can be improved to get a better (i.e. lower) upper bound on length through rerooting the down-pass network. A

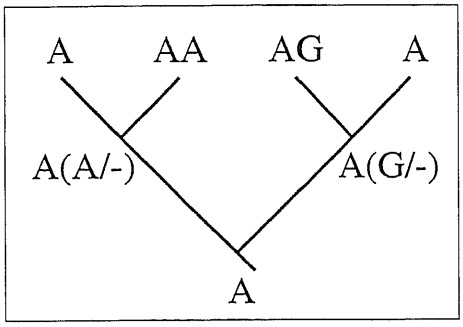

Figure 7. Rerooted version of the example of Figure 5.

"virtual" root is postulated, and each of the $2n - 3$ (for n terminal taxa) possible roots used to determine a down-pass cost and the minimum retained. The estimate of tree length based on the rooting of Figure 5 (two insertion-deletion events and a single base change) is reduced by moving the root (Fig. 7), resulting in a length of two insertion-deletion events. Some of the greediness of the down-pass algorithm can be circumvented this way.

In addition to this rerooting, which requires the examination of $n - 1$ nodes (n terminals), a second or up pass can be added to determine an estimate of the final states for each of the internal nodes (with the proviso that the final sequences not contain any gaps). Although this in itself will not effect the estimate of cladogram length, the determination of "final" states allows the use of other algorithmic speedups (Goloboff, 1996; Gladstein, 1997) which can dramatically improve the efficiency of tree searches.

The determination of final states can be achieved by traversing the tree in the direction opposite from that of the initial pass, away from the root (or virtual root – if rerooting has been performed). The final state set is defined as those states which minimize the total cost summed over the three paths from the node: the current node to each of its descendants and to the final states of its immediate ancestor. As with the initial ("down") pass, the results of all character combinations can be precalculated based on the matrix of transformation costs among each of the four nucleotide states and insertion-deletion costs.

These precalculations assume that only local taxon states matter, and introduce the errors of this heuristic procedure. For the initial pass, a two-dimensional matrix is required containing the resultant state and cost for each of the character combinations of A, C, G, T and "-", and ambiguities– 32×32. The second pass will require a second matrix of three dimensions describing all the possible interactions among the two descendant and one ancestral states.

Conclusions

The procedures outlined here allow the determination of a upper bound on cladogram length based on direct optimization of nucleotide sequences. This method is a simple extension of parsimony-based cladogram construction to include the origination and disposition of characters. Although this method is more elaborate and time-consuming that standard optimization procedures, the avoidance of multiple sequence alignment should result in both more efficient and parsimonious results. Furthermore, the nonindependence of gaps (i.e. insertion-deletion length), so tribulating to phylogenetic analysis based on the necessary assumption of character independence, can be accommodated seamlessly. In fact, any such model in which nucleotide changes interact (e.g. codon effects) can be integrated. Since the transformations are occurring within a single "character", or more specifically among complex character states, the independence of character vectors is not violated. Insertion-deletion costs can be complex, nonlinear functions of length, or codon bias added to analysis without violating this tenet of epistemological unity.

A final note concerns that pit of Dis, the "Felsenstein zone", (Felsenstein, 1978). The basic notion of nonhistorical, stochastically derived character matching based on random similarity. It is postulated that under certain conditions, the most parsimonious result will not be the correct one. Leaving aside the notion of "correctness" for the moment, it is worth noting that the size of this zone is inversely proportional to the number of character states expressed by the phylogenetic data. This is due to the requirement that the "bad" randomly similar character must overwhelm the "good" historically informative ones. S. Farris (personal communication) and others have noted that using characters in n-tuples (supersites) would reduce any long-branch problem by increasing the number of states – thereby decreasing the chance of random similarity. The methodology described here treats entire sequences as characters with huge numbers of possible character states. If the inverse relationship between the size of the Felsenstein zone and number of character states holds, the zone may be small indeed.

Acknowledgments

I would like to thank James Carpenter, Michael Whiting, David Gladstein and Rob DeSalle for many helpful comments and suggestions.

References

Farris, J. S. (1970) A method for computing Wagner trees. *Syst. Zool.* 34: 21–34.

Felsenstein, J. (1978) Cases in which parsimony or compatibility methods will be positively misleading. *Syst. Zool.* 27: 401–410.

Feng, D. and Doolittle, R. F. (1987) Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *J. Mol. Evol.* 25: 351–360.

Feng, D. and Doolittle, R. F. (1990) Progressive alignment and phylogenetic tree construction of protein sequences. *Meth. Enzomol.* 183: 375–387.

Fitch, W. M. (1971) Toward defining a course of evolution: minimum changes for a specific tree topology. *Syst. Zool.* 20: 406–416.

Gladstein, D. G. (1997) Incremental evaluation and the diagnosis of cladograms. *Cladistics* pp. 21–26.

Goloboff, P. A. (1996) NoNa Program and Documentation available from the author.

Hein, J. (1989) A new method that simultaneously aligns and reconstructs ancestral sequences for any number of homologous sequences, when a phylogeny is given. *Molec. Biol. Evol.* 6: 649–668.

Hein, J. (1990) Unified approach to alignment and phylogenies. *Meth. Enzomol.* 183: 626–644.

Hendy, M. D. and Penny, D. (1982) Branch and bound algorithms to determine minimal evolutionary trees. *Math. Biosci.* 59: 277–290.

Higgins, D. G. and Sharp, P. M. (1988) CLUSTAL: a package for performing multiple sequence alignment on a microcomputer. *Gene* 73: 237–244.

Higgins, D. G. and Sharp, P. M. (1989) Fast and sensitive multiple sequence alignments on a microcomputer. *CABIOS* 5: 151–153.

Needleman, S. B. and 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. D. and Rousseau, P. (1975) Locating the vertices of a Steiner tree in arbitrary space. *Math. Program.* 9: 240–246.

Sankoff, D. D. and Cedergren, R. J. (1983) Simultaneous comparison of three or more sequences related by a tree. *In*: Sankoff, D. and Kruskal, J. B. (eds) *Time Warps, String Edits, and Macromolecules: The Theory and Practise of Sequence Comparison*, Addison-Wesley, Reading, MA, pp. 253–264.

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

Wheeler, W. C. and Gladstein, D. G. (1992) Malign: A Multiple Sequence Alignment Program. New York, NY.

Wheeler, W. C. and Gladstein D. G. (1994) Malign: a multiple nucleic acid sequence alignment program. *J. Hered.* 85: 417.