Available online at www.sciencedirect.com
SCIENCE DIRECT®
Cladistics 19 (2003) 348–355
Cladistics
www.elsevier.com/locate/yclad

Search-based optimization

Ward C. Wheeler*

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

Accepted 25 March 2003

Abstract

The problem of determining the minimum cost hypothetical ancestral sequences for a given cladogram is known to be NP-complete (Wang and Jiang, 1994). Traditionally, point estimations of hypothetical ancestral sequences have been used to gain heuristic, upper bounds on cladogram cost. These include procedures with such diverse approaches as non-additive optimization of multiple sequence alignment, direct optimization (Wheeler, 1996), and fixed-state character optimization (Wheeler, 1999). A method is proposed here which, by extending fixed-state character optimization, replaces the estimation process with a search. This form of optimization examines a diversity of potential state solutions for cost-efficient hypothetical ancestral sequences and can result in greatly more parsimonious cladograms. Additionally, such an approach can be applied to other NP-complete phylogenetic optimization problems such as genomic break-point analysis.

Introduction

Systematists have consciously grappled with the NP-completeness of phylogenetic tree searching for some time, but as yet have not directly addressed the same property of the optimization problem presented by sequence data. The problem of determining the internal nodal sequences, such that the overall cladogram cost is minimized for a given cladogram is known to be NP-complete (Wang and Jiang, 1994). This is easily understood when one contemplates the increasing number of possible sequences as the number of observed sequences increase. In principle, all possible sequences of lengths 0 to the sum of all terminals with all possible combinations of nucleotides may occur. Wang and Jiang (1994) discussed this in their proof of NP-completeness and Wheeler (1998) in terms of Direct Optimization (Wheeler, 1996).

The traditional approach to the estimation of cladogram costs has been one of constructing some sort of point estimate for hypothetical ancestral sequences and then using this to determine an upper-bound on cladogram cost. The coupled processes of multiple sequence alignment and separate phylogenetic reconstruction does this through establishing global, static homologies to deal with length variation and then using standard optimization techniques to estimate the internal node character states. Direct Optimization (DO; Wheeler, 1996) takes a more explicit approach to this estimation, by establishing cladogram-specific homology schemes in a preliminary pass and constructing ancestral sequences in a second (up) pass. Unsurprisingly, DO usually yields better upper bounds on cladogram cost than multiple alignment methods. A third method, Fixed-Character State optimization (FSO; Wheeler, 1999), estimates internal nodal sequences by requiring they be drawn from the set defined by the terminals. In general, this yields less satisfactory cladogram costs, but may have other desirable properties (Wheeler, 2001).

An exact solution to this problem is obvious, if laborious. As with cladogram searches, one could simply enumerate all possible sequences and try each of them in turn at each of the internal nodes. Such an approach would guarantee the optimal solution, but would be impractical or impossible in all but the simplest cases. It may be, however, that in the same way that we can arrive at quite satisfactory results examining a very small fraction of binary trees when we search for cladograms, that we may not need to examine every possible sequence to arrive at a satisfactory (if not guaranteed minimal) solution. We could then examine each of these

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

0748-3007/$ - see front matter © 2003 The Willi Hennig Society. Published by Elsevier Science (USA). All rights reserved.
doi:10.1016/S0748-3007(03)00052-5

candidate solutions and determine the minimum cladogram cost given the set of examined sequences. Noting that we are often satisfied (achieve usable, stable results) with examining $n^4$of the $\Pi\ 2i-5$ ($i=3$ to n for n terminals—a tiny fraction) cladograms with repeated random addition sequences and TBR branch swapping, we might have similar luck in finding useful solutions with sequence optimization. This is the crux of search-based optimization.

The method

Briefly, there are two steps to performing search-based optimization. The first is to define the set of possible ancestral sequences. This could be achieved in many ways; here we will accomplish this through the use of the final sequence states generated by DO. The second step is to evaluate this (potentially quite large) state set for a given cladogram via dynamic programming. One could start with a small state set, perform the analysis, and then enlarge the set of possible sequences repeating the procedure. This might be done until stability or exhaustion occurred (Fig. 1).

It is worthwhile to note here that the "character" of sequence data used in this form of analysis is the entire contiguous sequence segment. This may be a fragment, locus, or more extensive stretch of nucleic acid. This is the same character concept used in DO (Wheeler, 1996) and FSO (Wheeler, 1999).

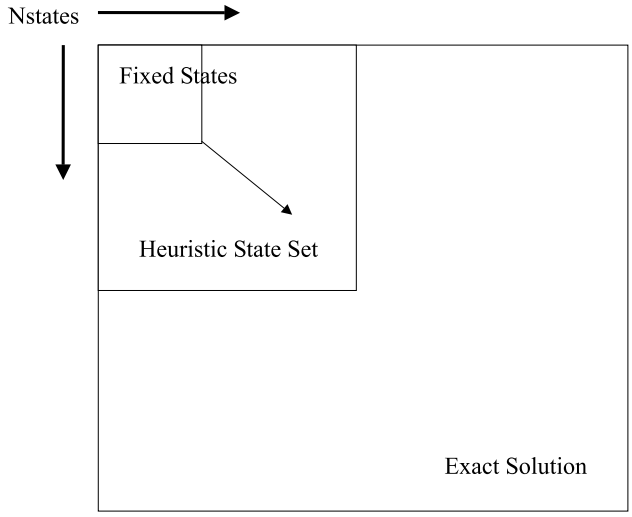

Fig. 1. Character state set for the general problem of sequence optimization. The relative sizes of the transformation cost matrices from sequence to sequence is shown by the squares. The smallest is that generated by Fixed State Optimization (Wheeler, 1999), the intermediate size represents the heuristic sequence set explored by the method presented here, and the entire square represents the exact solution derived form examining all possible sequences.

The first step, as mentioned above, is the generation of a set of potential ancestral sequences. Given that the computational cost of the dynamic programming step which follows will be dependent on the square of the size of this set, the collection should be as small, yet as inclusive of pertinent variation, as possible. As an example, even though sequences containing many "T"s are possible in ancestral sequences where the terminals exhibit only "G"s and "C"s, employment of such hypothetical ancestral sequences would be inefficient since they are very unlikely to be useful. Along these lines, one approach is to perform a series of random addition Wagner builds and reconstruct the ancestral sequences using DO. Since it is crucial that these sequences be specific (ambiguities can lead to underestimation of cladogram cost and non-metricity), when the up-pass optimization regime would allow multiple possible nucleotides, one is chosen at random and the optimization proceeds up the tree. In this way, non-ambiguous sequences are generated which are consistent with at least some schemes of phylogenetic relationship. These random additions are performed, and $n-1$ (for n terminals) potentially unique hypothetical ancestral sequences accumulated at each iteration.

The reason that ambiguities can lead to problems comes from the edit cost calculations. The edit cost between two sequences, one consisting of A's, C's, G's, and T's and a second of an equal number of N's, would be zero. This "N" sequence would also have a zero edit cost to any sequence of that same length. The non-metricity comes in when transformations between the specific sequences (ACGT's) could pass through an intermediate "N" sequence state. This would entail zero cost, whereas a direct transformation would have non-zero cost—a violation of the triangle inequality.

Once the set of potential states is created (and augmented with the observed sequences), the optimization of a cladogram or cladograms can proceed via dynamic programming (Sankoff and Rousseau, 1975), akin to a Sankoff-style matrix character with a large number of potential states. The time of this operation will be dependent both on the number of terminal taxa (linearly) and the number of potential states (quadratically). Large state sets will result in lengthy searches.

A synthetic example

Consider the sequences of Fig. 2, with 9 taxa and lengths of 2 to 12 nucleotides. Potential internal sequences were generated with POY (vers. 3.0; Wheeler et al., 2002) using the options "–notbr –maxtrees 1 –seed –1 –random 1 –printhypanc –diagnose." The default indel cost of 2 and base substitution cost of 1 were used as well. This caused POY to create a single random addition Wagner tree without any refinement such as SPR or

```
a
      1 AAATTT

b
      1 ATATATATAT

c
      1 TATAT

d
      1 ATAGATACTAAA

e
      1 TTTAAA

f
      1 TTGACATAGCA

g
      1 TA

h
      1 GGGACCC

i
      1 AAA
```

Fig. 2. "Bad" sequence data set.

TBR branch swapping. A cladogram was saved each time and a "random" optimization regime was used to create non-ambiguous hypothetical ancestral sequences. These hypothetical ancestral sequences were kept each time and duplications removed. The first 100 unique sequences are shown in Table 1.

Once defined, the potential sequence set can be applied to cladogram diagnosis and searches. Without any additional states specified, so that the sequence set is limited to observed sequences (fixed states), the shortest cladogram found had a cost of 44 weighted steps. With 100 additional random addition sequences, the minimum cost cladogram had a cost of 40 weighted steps. This minimum cost was achieved at 61 additional sequences (for a states set of 70 including the 9 observed) and remained at 40 through states set sizes of 2000.

Using CLUSTAL (Thompson et al., 1994, 1997), PHAST (Goloboff, 1996), and MALIGN (Wheeler and Gladstein, 1994, 1991–1998) by way of comparison, conventional multiple sequence alignment followed by standard phylogenetic analysis yielded less parsimonious results. CLUSTAL + PHAST resulted in a cladogram of cost 56, and MALIGN + PHAST at cost 43. DO (via POY) generated a cladogram of weighted cost 40. This difference was even more pronounced when a second sequence file was added which did not allow consistent accommodation of sequence length variation. This data set (Fig. 3) is a permutation of the sample data set of Fig. 2. Although each sequence file can yield a minimum cost cladogram of 40 steps, together the minimum for search-based optimization (with the same 100 random additional sequences) is 84. DO yielded a cladogram of 86 weighted steps, MALIGN + PHAST 106, and CLUSTAL + PHAST 120. Clearly, this form of optimization is yielding shorter cost cladograms than other methods. The cladogram cost with no additional states (Fixed-States cladogram) had a cost of 91 weighted steps (Fig. 4). The search-based optimization cladogram is 2% shorter than DO and over 20% shorter than the multiple alignment methods. These differences are quite startling given the small size of the test data set.

Table 1

AAA	TTATATATC
TATATA	ATATATATAT
GATATA	TTACATATCT
AGGTAGC	TTAGATATCT
TTAGATAGCA	GATATCT
ATAGATATAA	GATACAA
TATAAA	ATGATATAGAA
TTTAAA	TATAAT
TGATATA	TTGATATAA
TAGAAAGC	TTGACATAA
GGAAACC	TTGACAA
ATATATC	TAGACATAA
TTATATATCA	TAGACAA
ATATATATAA	TAAAAT
ATATAT	TGTTGAAA
TATAT	TGATATAAA
ATA	TTGATATTAAA
ATATATACTAAA	ATAGATACTAAA
ATAATAT	ATGATATATAA
TAATAT	GTAATAT
TATTAT	AAATTT
TTGACATAGAA	GATTT
TTAGATATAT	GGTTGAAC
GATATAT	AGATATAAC
TAAAA	TAGATATAACA
ATAGATATAT	TAATAA
TTATAT	TGGGATC
TGATAT	ATGATATAC
TAGAGATC	GATATAC
ATAGATATCA	AGAGATATAA
GATATAA	TGAGATATCA
GTTTATA	GATATCA
TATAA	ATATAA
ATATATATA	TTAAA
GGACAC	TTATT
GTAAAC	GAGAT
TAAAT	GGGAACC
AAAAT	ATAGATATC
TTTATAAC	ATGATATATCA
TTATATAGC	GATAA
ATATATATAC	GGGAAA
ATAGATACTAA	ATATATAAA
ATACATATAT	TAGATATTAAA
ATAAAA	AAATAT
GAGACAA	AATAT
ATGATACTAAA	GTGTAAC
ATGATATAAAA	AA
TATATAA	ATACATATAA
TGTATAA	TAA
TATATAT	TATA

```
a
      1 ATAGATACTAAA
b
      1 TA
c
      1 TTGACATAGCA
d
      1 AAA
e
      1 GGGACCC
f
      1 TTTAAA
g
      1 ATATATATAT
h
      1 TATAT
i
      1 AAATTT
```

Fig. 3. "Bad" sequence data set permuted in taxa such that the two versions cannot be simultaneously accorded optimal arrangement.

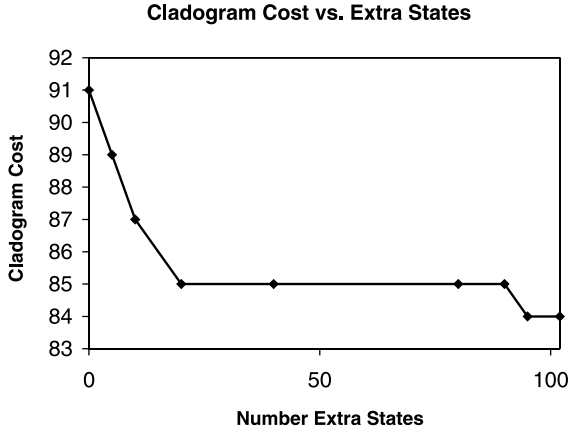

Fig. 4. Cladogram cost as a function of extra sequence states. As additional potential ancestral sequences are considered during cladogram search, overall cladogram cost decreases.

Real examples

A demonstration of search-based optimization applied to a real data set concerns arthropod and centipede relationships. In their multilocus molecular sequence + anatomy (and non-sequence molecular data) total evidence analysis of 54 arthropod taxa, Giribet et al. (2001) reported support for "Pancrustacea" a somewhat surprising grouping of insects and crustaceans. To do this, Giribet et al. presented 303 morphological characters and 8 nuclear and mitochondrial loci. They analyzed their data-using DO and their congruence-based sensitivity analysis favored a cladogram employing completely homogeneous weighting (indels = transversions = transitions) at 27,375 steps.

In order to prepare the set of potential ancestral sequences, a procedure similar to that used for the test case above was performed. For each locus, or fragment of a locus, 50 random addition Wagner trees were calculated using DO. No branch swapping was performed and sequence variation was optimized on the resulting cladogram and final hypothetical ancestral sequences reconstructed. To this were added the optimized best results from searches of the individual loci (or fragments) and a combined run. A total of 2756 hypothetical ancestral sequences were generated for each locus/fragment. There was often considerable redundancy and the number of unique potential sequences varied greatly (Table 2).

Table 2
Unique candidate hypothetical ancestral sequences for the arthropod data set (after Giribet et al., 2001)

Fragment	Number unique sequences
18s1	737
18s2	1038
18s3	2210
18s4	122
18s5	564
18s6	30
18s8	1647
18s9	263
18s10	32
18s11	763
18s12	139
18s13	172
18s14	23
18s15	372
18s17	30
18s18	1472
18s19	328
18s20	155
18s21	102
18s23	62
28s1	35
28s2	1253
28s4	1651
28s5	254
16s1	681
16s2	52
16s3	27
16s4	183
16s5	1727
16s6	1037
16s7	1809
16s8	1819
16s9	670
16s10	1529
U2	1021
H3	1809
EF1b	1999
Pol1	1908
Pol2	1212
Pol3	829
Pol5	828
COI	2232

Due to memory constraints, up to the first 550 of these unique sequences were chosen as the potential state set (in addition to the observed sequences). This is the set used for subsequent search-based optimization analyses.

When the cladogram of Giribet et al., was diagnosed, the original cladogram cost of 27,375 was recovered (Fig. 5), for DO, and 24,107 steps for the search-based approach (FSO yielded 26,969 steps). When a complete search was performed (using POY on 50 1GHz PIII processors in parallel) a cost of 23,408 steps was found (Fig. 6) a decrease in cladogram cost of 17%. A FSO search resulted in a cladogram of cost 26,066 (Fig. 7). Given that the FSO analysis yielded a less costly cladogram than the DO, ambiguities in the observed sequences (mainly terminal N's) may have artificially reduced cost in the FSO and the search-based cladograms. The additional decrement, however, from the search-based analysis over the FSO (11.8%) suggests that this is not the entire effect. The following example has no such potentially confounding factors.

A second example comes form Edgecombe et al.'s (2002) centipede data, which are very complete with few sequence ambiguities (N's). As with the previous

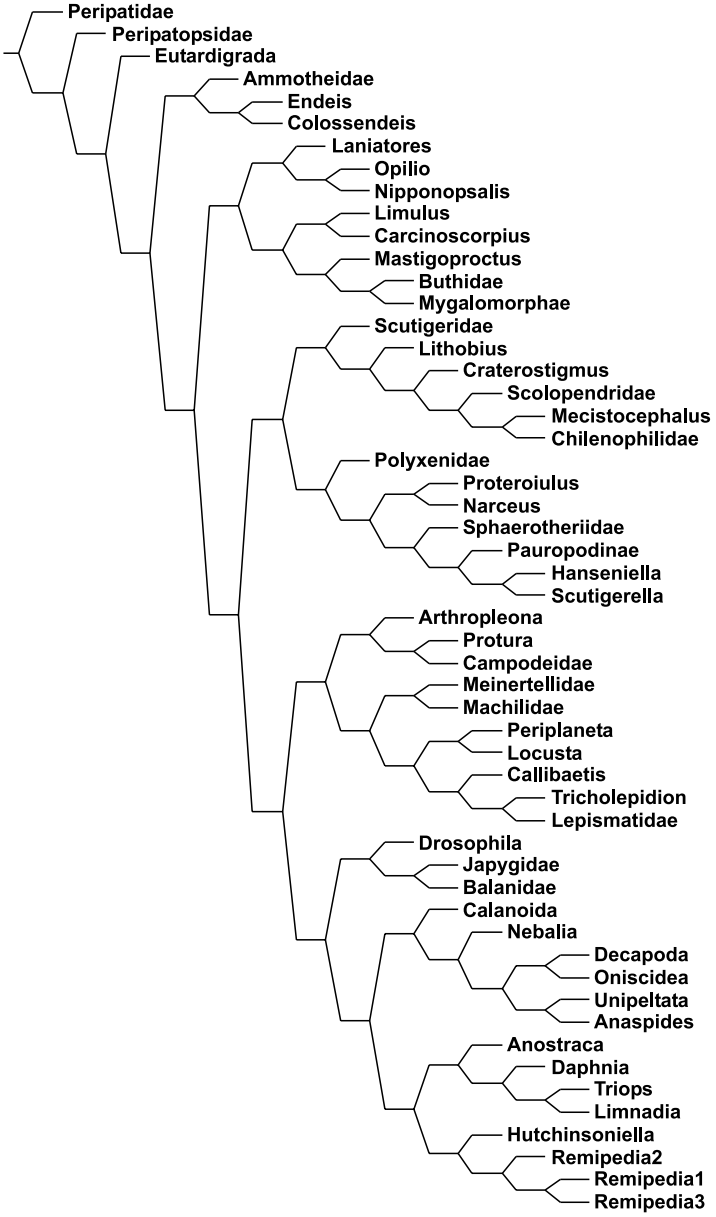

Fig. 5. Cladogram of arthropod relationships of Giribet et al. (2001). Optimization was used to search for this cladogram at cost 27,375 steps. All changes were given equal weight (indel = transversion = transition).

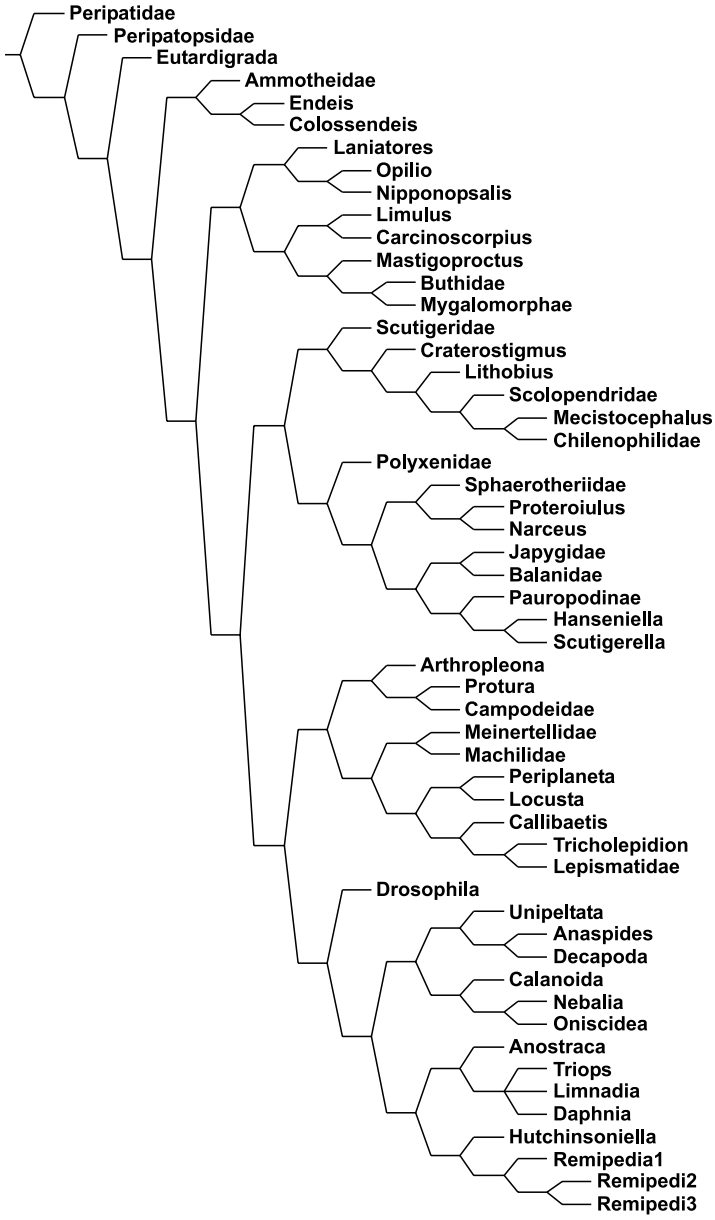

Fig. 6. Search-based optimization of the cladogram of Giribet et al. (2001) data. Cladogram cost was 23,408 steps. All changes were given equal weight (indel = transversion = transition).

example, there were multiple molecular loci (5) and a morphological data set. Analysis proceeded as in the previous example with two exceptions. First, only 10 random replicates were used to generate candidate ancestral sequences, and second, all resultant states were held and used for diagnosis and search. For homogeneous weighting (all transformations = 1), using DO, the published lowest cost cladogram is at 4376, this was based on 1000 replicate runs with elaborate search options. A single addition sequence run with DO yielded a cladogram cost of 4394. FSO yielded 4810, and search-based optimization yielded a cladogram at cost 4307 (1.98% less costly) for this simple search.

Shortcuts and speed-ups

As mentioned above, the execution time of Search-based optimization is almost entirely dependent on the number of potential states (Table 3). This suggests two possible avenues for reduction in execution time. The first is to employ approximate solutions based on the reconstructed final states of the internal nodes, and the second to dynamically alter the number of states during analyses.

A full down-pass dynamic optimization of a cladogram of "n" taxa and "m" states would have a complexity on the order of nm^2 for each character. This dependence on n taxa can be reduced almost to a constant factor of 2 via the shortcuts described by Goloboff

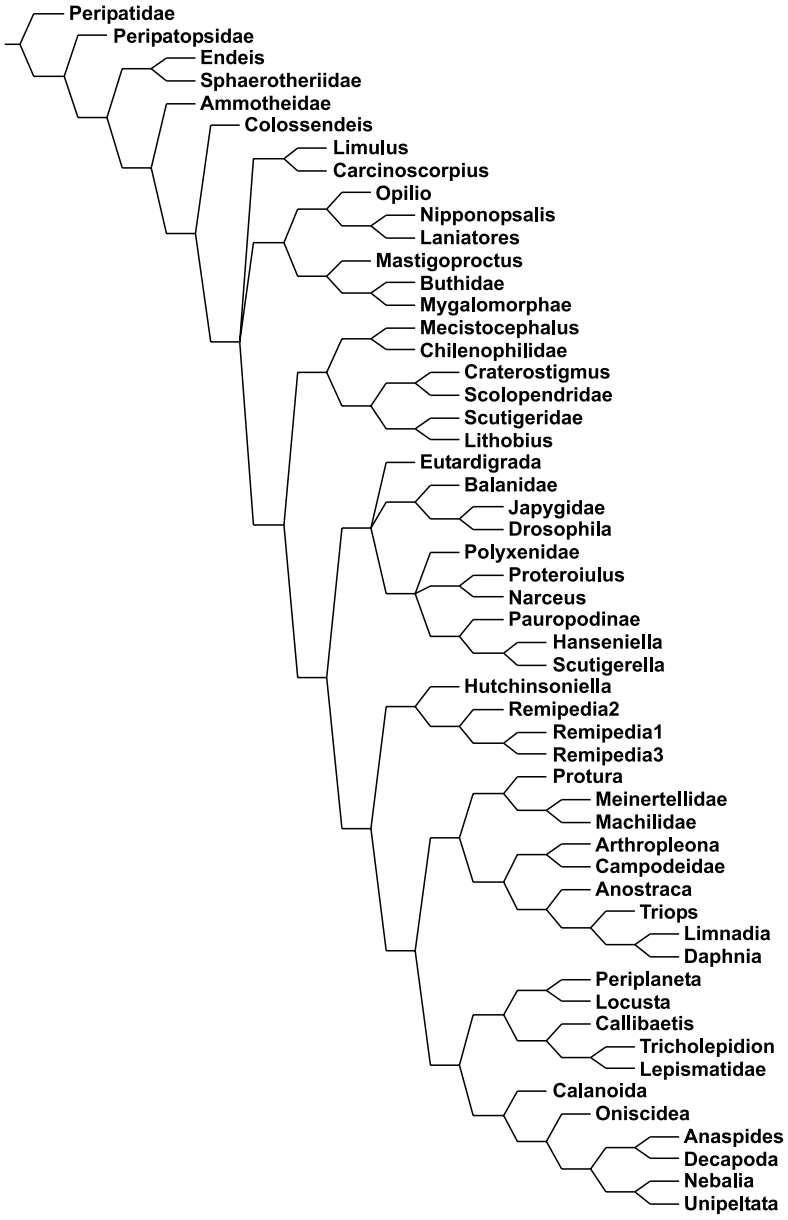

Fig. 7. Fixed-states optimization of the cladogram of Giribet et al. (2001) data. Cladogram cost was 26,670 steps. All changes were given equal weight (indel = transversion = transition).

Table 3
Search-based optimization execution times

No. sequences	Cladogram cost	No. cladograms	Time	Cladograms/s
17	476	5208	0	>5208
25	467	8580	1	8580
50	456	14,304	2	7152
100	435	10,030	4	2508
200	416	12,412	32	387.9
400	392	9551	114	83.78
1000	388	10,100	1262	8.003

(1993) and would also greatly reduce m^2 to as low as 1 or 2. This is because the Goloboff shortcuts are dependent on the final ancestral states reconstructions. Although there are m possible states, the number of states reconstructed at the internal nodes is more usually one or two. These cost calculations are approximate however, and require checking with a down-pass cost calculation that is dependent on the full state set. Incremental optimization can reduce the complete down-pass costs as well from n to approximately log n (Gladstein, 1997).

A second possibility is to reduce the number of states examined during the search. It is possible to note the states used in intermediate solutions, perhaps during the initial cladogram building steps, and remove unused states for an initial refinement step. After this refinement went to completion, the state set could be returned to its

full complement and refinement repeated. Such an operation could be performed iteratively until stability occurred. This compression of states should accelerate cladogram cost calculation considerably. This might also have the additional benefit of decreasing the memory consumption of the procedure (currently also nm^2).

Discussion

There are only two required elements for Search-based optimization. The first is a set of possible states, and the second a cost function to determine the transformation costs between each pair of states. In the case of DNA sequences, the state set consists of the possible sequences (or a heuristic subset) and the cost matrix of the pair-wise minimum edit costs between the sequences. If the characters were binary, there would be only two states and the single edit cost would be one step. If the characters were prealigned nucleotide positions, there would be 5 states and a traditional Sankoff and Rousseau (1975) matrix would specify all the possible transformations. Hence, this type of optimization is a generalization of other character types and even can be applied to the character data generated by gene order studies.

Genomic breakpoint or rearrangement data are a source of information that has been available from complete mtDNA sequences for some time (Boore et al., 1995) but only recently have the tools become available for its analysis (Blanchette et al., 1997; Sankoff and Blanchette, 2000). As with sequence data, the world of possible sequence rearrangements is huge and the transformation paths among these correspondingly diverse. Algorithms to optimize these variations parsimoniously have been developed by Blanchette et al. (1997) and Sankoff and Blanchette (2000) but they can be very time consuming. A simple application of Search-based optimization can also deal with such problems through a priori definition of possible genomes and their edit costs. Using the data of Blanchette et al. (1999) on arthropod gene order, search based optimization yields the same solution as that of Blanchette et al. (201 steps) when using only the observed genomes as a state set.

Conclusions

Search-based optimization promises to be an efficient and effective means of optimizing sequence and many other types of complex phylogenetic data. At present this effectiveness is limited by the computational problems inherent in dynamic programming of such large state sets. These are likely to be ameliorated by algorithmic improvements for a given states set, and by methods to define the states set in a more efficient manner.

Acknowledgments

I would like to acknowledge the help of Julian Favovich, Cyrille D'Haese, Gonzalo Giribet, Taran Grant, Daniel Janies, Jan De Laet, Leo Smith, and Susanne Schulmeister in critiquing this manuscript, and Steven Thurston for the good artwork. NSF Systematic Biology and NASA Fundamental Space Biology Program have generously provided Grant support.

References

Blanchette, M., Bourque, G., Sankoff, D., 1997. Breakpoint phylogenies. In: Miyano, S., Takagi, T. (Eds.), Genome Informatics. University of Academy Press, Tokyo, pp. 25–34.

Blanchette, M., Kunisawa, T., Sankoff, D., 1999. Gene order breakpoint evidence in animal mitochondrial phylogeny. J. Mol. Evol. 49, 193–203.

Boore, J.L., Collins, T.M., Stanton, D., Daehler, L.L., Brown, W.M., 1995. Deducing the pattern of arthropod phylogeny from mitochondrial rearrangements. Nature 376, 163–165.

Edgecombe, G.D., Giribet, G., Wheeler, W.C., 2002. Phylogeny of Henicopidae (Chilopoda: Lithobiomorpha): a combined analysis of morphology and five molecular loci. Syst. Ent. 27, 31–64.

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

Gladstein, D.G., 1997. Incremental evaluation and the diagnosis of cladograms. Cladistics 13, 21–26.

Goloboff, P.A., 1993. Character optimization and calculation of tree lengths. Cladistics 9, 433–436.

Goloboff, P., 1996. PHAST. Program and Documentation, ver. 1.5.

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

Sankoff, D., Blanchette, M., 2000. Multiple genome rearrangement and breakpoint. Journal of Computational Biology 5, 555–570.

Thompson, J.D., Higgins, D.G., Gibson, T.J., 1994. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, positions-specific gap penalties and weight matrix choice. Nucleic Acids Res. 22, 4673–4680.

Thompson, J.D., Gibson, T.J., Plewniak, F., Jeanmougin, F., Higgins, D.G., 1997. The CLUSTAL_X windows interface: flexible strategies for multiple sequence alignment aided by quality analysis tools. Nucleic Acids Res. 25, 4876–48821.

Wang, L., Jiang, T., 1994. On the complexity of multiple sequence alignment. Journal of Computational Biology 1, 337–348.

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

Wheeler, W.C., 1998. Alignment characters, dynamic programming, and heuristic solutions. In: DeSalle, R., Schierwater, B. (Eds.), Molecular Approaches to Ecology and Evolution, second ed. Birkhäuser Verlag, Basel, Switzerland, pp. 243–251.

Wheeler, W.C., 1999. Fixed character states and the optimization of molecular sequence data. Cladistics 15, 379–385.

Wheeler, W.C., 2001. Homology and the optimization of DNA sequence data. Cladistics 17, S3–S11.

Wheeler, W.C., Gladstein, D.G., 1994. MALIGN: a multiple sequence alignment program. J. Hered. 85, 417–418.

Wheeler, W.C., Gladstein, D.G., 1991–1998. Malign. Program and Documentation. New York, NY. Documentation by Daniel Janies and Ward Wheeler.

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