5.1 Introduction

Systematics is the production of cladograms that link taxa through their observed variation. These cladograms must optimize an objective function such that they can participate in hypothesis testing on the basis of this function. The core activity of systematics is to assay the relative merits of a pair of competing scenarios and judge one superior. The repeated and transitive application of this elemental comparison results in a globally optimal solution that is the ultimate goal of systematics.

This depiction of systematics raises three points: the nature of the objective optimality criterion; the manner of determination of this value; and the assessment of the relative merits of cladograms. This chapter is concerned with the second of these three, the realm of character analysis, homology and optimization. The arguments here will be based on the optimality criterion of parsimony or minimum cost. Likelihood or other criteria could well be used, however, and most of the character-optimization discussions would remain largely unchanged other than the specifics of their implementation and numerical values. The comparison of cladograms is the province of cladogram or tree searching and is not discussed in any depth here.

Homology is the relationship between features that is derived from their shared, unique origin. Given a single cladogram, two features are homologous if their origin can be traced back to a specific transformation on a branch of that cladogram, but the same pair of features may not be homologous on alternate cladograms. Homology is entirely cladogram-dependent and the relative optimality of alternate cladograms determines whether or not features exhibit this relationship.

The dynamic homology framework (Wheeler 2001a) is an analytical concept that extends through optimization of transformations to the correspondence among features (often referred to as putative or primary homologies) themselves. The joint scenario of correspondence and transformation is chosen such that the overall cladogram cost is minimal. The correspondences among features (nucleotides in this context) are not predetermined, but a result of the analysis. In this framework, there is no distinction between putative or primary and secondary homology (de Pinna 1991)—all variation is optimized de novo for each cladogram.

5.2 Sequence data

There are two properties that have been used to differentiate sequence data from other sorts of information: simplicity of states and length variation. Unlike complex anatomical features (e.g. limb or wing) that can express themselves in a myriad of forms, nucleotides exhibit only four conditions. Complexity and difference imply that states (e.g. presence/absence, or conditions) are not comparable across characters. Nucleotide states, on the other hand are identical no matter where they occur. Nucleotide sequences may also differ in length. These two aspects of molecular sequence data remove the complexity and positional information so often used in establishing primary homologies in anatomical systems.
5.3 Alignment and optimization

Two approaches have been developed to deal with the absence of preordained homologies and analyze sequence data. On one hand, methods have been devised to create the missing primary homology statements that are then analyzed by standard techniques—broadly referred to as multiple alignment. Traditionally, sequence data have undergone this pre-phylogenetic analysis step to permit familiar procedures akin to those used with anatomical characters. A second approach is to directly optimize sequence variation during cladogram searching. This methodology requires no notions of primary character homology or any global (i.e. topology-independent) homology statements whatsoever (other than that the compared sequences themselves be homologous). Direct optimization is also applicable to simple, serial morphologies, for which characters and their states are similarly constrained. For the sake of discussion here, the terms alignment and optimization will refer to these alternate approaches.

5.4 The problem

In computational terms, the problem of determining the cost of a given cladogram reduces to the determination of the set of internal vertex (hypothetical ancestral) sequences such that the overall cost is minimized. Whether expressed in terms of alignment or optimization, the problem (known as the tree-alignment problem) is NP-hard (see Wang and Jiang 1994); hence, we are very unlikely to achieve exact solutions. NP-hard problems are members of a class of computational problems for which there is no known polynomial time solution. These problems are often combinatorially ‘explosive’, with the size of the solution space expanding factorially. That is, when the sequences can vary in length, even the determination of the cost of a single cladogram will be heuristic (for 10 sequences of length 5—an unrealistically small and well-behaved case—there are \(1.35 \times 10^{38}\) homology schemes; Sadowinski 1998).

Given that determining cladogram cost is heuristic, the transformation and homology statements derived from the cladogram are heuristic as well. When coupled with cladogram search, we are faced with a compound NP-complete problem and all of our statements will be based on approximate solutions.

Both alignment and optimization may be viewed as heuristic approaches to solving this problem. Alignment accomplishes this based on static, global, primary homology statements, whereas optimization techniques propose cladogram-specific homology scenarios.

5.5 Alignment methods

As a heuristic solution, alignment decomposes the nested homology/search procedure into two sequential problems. Length-variable sequences are converted into a series of column vectors (primary homology statements) through the insertion of gap characters (−) as placeholders that denote the results of insertion/deletion events. Alignments minimize a cost function (in the case of two sequences the cost to ‘edit’ one sequence into the other) that is based on the relative costs of transformation events (especially insertion/deletion—‘indel’ costs), which may or may not be cladogram-based.

There are several components to the alignment process. These progress from pairwise alignment of two sequences to the exact solution for multiple sequences and then to the heuristic methods employed in real-world analyses.

5.5.1 Pairwise alignment

Alignment of sequence pairs is the foundation of all more elaborate procedures. The problem, simply stated, is to create the series of correspondences between the nucleotides in two sequences via the insertion of gaps, such that the edit cost (the weighted sum of all events—insertions, deletions, nucleotide substitutions—required to convert one sequence into another) between the sequences is minimized (or some other function optimized). Costs must be assigned to each type of event, or trivial, zero-cost alignments can result (e.g. indels costing zero and an alignment that places each nucleotide opposite a gap). The first algorithmic solution to this form of string-matching problem was proposed by Needleman
and Wunsch (1970) and is used throughout most alignment procedures (see Gusfield 1997 for more extensive discussion).

Consider two sequences ACCT and AGCT and alignment parameters of nucleotide-substitution cost equal to 1 (Cost subst) and indel cost equal to 10 (Cost indel). The algorithm follows a dynamic programming approach by solving a series of small, dependent sub-problems that implicitly examine all possible alignments. There are two components to the procedure. The first determines the cost of the best alignment (or alignments—there may be multiple solutions). This is often referred to as the wavefront update. The second component is the traceback, which yields the alignment itself (more complex examples can be found in Phillips et al. 2000). Needleman and Wunsch described a maximization-of-identity algorithm, where here a minimization of difference is presented. The underlying principles are unchanged.

\[
\text{Cost}_{i,j} = \min \{ \text{Cost}_{i-1,j} + \text{Cost}_{\text{InDel}}, \text{Cost}_{i,j-1} + \text{Cost}_{\text{InDel}}, \text{Cost}_{i-1,j-1} + \text{Cost}_{\text{Sub}}, \text{Cost}_{i,j} \}
\]

The first part of the algorithm fills a matrix \( M \) of size \((n+1) \times (m+1)\) to align a pair of sequences \(a\) and \(b\) of length \(n\) and \(m\) respectively. Each cell \((i,j)\) is the cost of aligning the first \(i\) characters of \(a\) with the first \(j\) characters of \(b\) (i.e. aligning \(a_1\ldots a_i\) and \(b_1\ldots b_j\)). Each value is calculated using the previously aligned subsequences: that is, the cost of cell \((i,j)\) will be

\[
\min\{ (i-1,j) + \text{indel}, (i,j-1) + \text{indel}, (i-1,j-1) + \text{align character } a_i \text{ and } b_j \}
\]

or less formally, the minimum among
- aligning \(a_1\ldots a_{i-1}\) and \(b_1\ldots b_j\) and
- aligning character \(a_i\) with a gap,
- aligning \(a_1\ldots a_i\) and \(b_1\ldots b_{j-1}\) and
- aligning character \(b_j\) with a gap,
- aligning \(a_1\ldots a_{i-1}\) and \(b_1\ldots b_{j-1}\)
- aligning character \(a_i\) and \(b_j\).

The additional first row and column (the reason for the +1 in the matrix dimensions) represents the alignment of a sequence with an empty string; that is, initial gaps. Each decision minimum is recorded, to follow the path that leads to the cost of aligning \(a\) and \(b\); that is, the cost in cell \((n,m)\) (Fig. 5.1).

In order to create the actual alignment between the sequences a traceback step is performed that proceeds back up and to the left of the matrix, keeping track of the optimal indels and substitutions performed in the matrix-update operations. The minimum cost path is followed back, where the best move is diagonal if the nucleotides of the sequences correspond, and the left and upwards moves signify indels (Fig. 5.1). The minimal cost alignment for these sequences (ACGT and AGCT) with the cost regime [indels = 10, substitutions = 1] is 2 with two base substitutions implied between the sequences (C → G, and G → C). If a complementary cost scenario is specified, e.g. indels = 1 and substitutions = 10, a different optimal solution is found (Fig. 5.1, right). In this case as well, the minimum cost is two, but no substitutions are implied—only indels (2).

![Figure 5.1 Needleman and Wunsch (1970) alignment matrix tables for two cost scenarios. On the left, Indel events cost 10 steps and nucleotide changes 1, while these are reversed on the right. Both cost scenarios yield minimum cost alignments of cost 2, although minimizing indels in the former (left) and nucleotide substitutions in the latter (right).](image-url)
Furthermore, there are two equally optimal solutions differing in the placement of the gaps. This ambiguity comes from the equally costly paths found at matrix element 3,3 (of 0,0 to 4,4). The non-unique nature of such solutions is a frequent property of alignments and can have dramatic effects on phylogenetic conclusions (Wheeler 1994).

5.5.2 Exact multiple alignment

The pairwise procedure can be generalized in a straightforward fashion to align more than two sequences. The matrix would have an axis for each sequence (l sequences would require l dimensions) and there would be $2^l - 1$ paths to each cell representing all the possible combinations of gaps and substitutions possible (seven in the case of three sequences). These two factors add enormously to the calculations, making true multidimensional alignments unattainable for real data sets.

An additional complexity arises in analyses of data sets with more than three sequences. The cost calculations at each cell may (as Sankoff and Cedergren 1983 suggested) be based on the cladogram of relationships of the sequences. If this is known, or at least specified a priori, the cell cost can be calculated directly. If, however, the cladogram is unspecified, a search would be performed for each cell, or the entire multidimensional alignment repeated for multiple (potentially all) cladograms.

The immense computational burden of exact multiple alignment ensures that heuristic solutions are used in nearly all real-world cases.

5.5.3 Heuristic multiple alignment

Current heuristic procedures are similar in that many attempt to render multiple alignment tractable by breaking down simultaneous n-dimensional alignments into a series of manageable pairwise alignments related by a “guide tree” (in the parlance of Feng and Doolittle 1987). These differ in the techniques used to generate the guide tree and conduct the pairwise alignments at the guide tree nodes. Furthermore, the procedures may or may not be explicitly linked to optimality criteria (Fig. 5.2).

By far the most commonly used heuristic multiple-alignment implementation is CLUSTAL, mainly because it is fast and relatively easy to use. Many others are freely available, however, and take different approaches to the problem. Several of these approaches are illustrated in this sample. More-complete lists can be found at http://pbil.univ-lyon1.fr/alignment.html and more comparisons in Phillips et al. (2000).

CLUSTAL (Higgins and Sharp 1988 et seq.) creates a single multiple alignment based on a single guide tree. A neighbor joining tree (Saitou and Nei 1987) is calculated from the pairwise alignments via a ‘corrected’ distance formula. This tree is used as a guide tree for progressive pairwise alignment of terminal sequences and internal consensus sequences (a down-pass). A second (up) pass resolves the placement of gaps in internal and ultimately observed sequences. There is no optimality value associated with a CLUSTAL alignment.

TREALIGN (Hein 1989a, b) also produces a single multiple alignment based on a single guide tree, but that guide tree is constructed (with some tree refinement) as the alignment is created. A parsimony step is included as part of the tree-reconstruction procedure. Although alignments are not searched as such, the generation of the guide tree examines multiple alternatives. A final, single multiple alignment is generated with an attached parsimony score, but no comparisons to other complete alignments are made.

DALIGN (Morgenstern et al. 1996) differs from other methods in looking for alignments of contiguous gap-free fragments of DNA that may have mismatches. This contrasts with the approach that attempts to align each position in a sequence. No gap penalty is employed. The idea behind this method is to create complete alignments by stitching together locally similar sequences that may be separated by highly divergent regions. An optimal alignment is one that maximizes the weighted sum of the matches in the smaller segments. Alignments can be compared on this basis. This method makes no reference to cladograms or trees whatsoever.
COFFEE (Notredame et al. 1998, 2000) behaves as a ‘wrapper,’ using a genetic algorithm to optimize multiple alignments based on consistency with the pairwise alignments of the same sequences. Any pairwise alignment procedure can be used under the COFFEE optimality function.

The following alignment methods involve ‘search’ procedures. In MALIGN and the method of Hein et al. (2003), tree searches are conducted to produce multiple alignments, whereas POY searches for optimal cladograms directly and can generate alignments *post facto* for the optimal cladogram.

MALIGN (Wheeler and Gladstein 1994) uses multiple guide trees to generate a diversity of multiple sequence alignments, choosing the best on the basis of the parsimony score (indels included) of the most parsimonious cladogram derived from that alignment. Guide trees are searched and multiple alignments created for each candidate guide tree. Each alignment is used as the basis for a heuristic cladogram search (indels weighted and included). The cost of the most parsimonious cladogram is attached to the alignment as its optimality score. MALIGN will output multiple multiple-alignments if they are equally optimal.
Hein et al. (2003) employ the Thorne–Kishino–Felsenstein (TKF) model (Thorne et al. 1991) for likelihood-based multiple alignments related by a tree. The algorithm employed is based on Sankoff (1975) for likelihood. Currently, the implementation (designed for demonstration purposes) can manage a few sequences (ca. 7) but could well be extended to larger data sets.

POY Implied Alignment (Wheeler 2003a; Wheeler et al. 2003) is not an alignment program, but searches for parsimonious cladograms directly (see the next section). A multiple alignment can be generated, however, from the transformation series implied by the optimal cladogram. This is not a multiple alignment in the sense of other methods, but rather is inextricably linked to the cladogram from which it was derived (Wheeler 2003a).

5.6 Optimization methods

In contrast to alignment procedures, optimization methods skip the alignment step and proceed directly to the determination of cladogram cost. This is achieved by focusing on determining optimal hypothetical taxonomic unit (HTU) sequences at internal tree nodes. In doing so, homology schemes are created for each cladogram uniquely, and for cladogram costs based on them. Multiple-alignment methods create a single alignment upon which all cladograms are diagnosed. Optimization methods create individualized homology schemes for each cladogram.

5.6.1 Exact solutions

As mentioned earlier, the determination of the lowest cost for a single cladogram depends on the lowest cost assignment of HTU sequences, and this is an NP-hard problem (Wang and Jiang 1994). Exact solutions, therefore, will not be available generally.

Sankoff (1975) proposed a recursive procedure that would calculate the minimum-cost cladogram exactly. This method requires a number of steps proportional to $(2n)^m$ where $n$ is the average length of the sequences and $m$ the number of sequences for a given cladogram. An alternate, simple-minded exhaustive approach would be to simply generate a list of all possible sequences, determine the edit cost between each pair (via some procedure akin to that of Needleman and Wunsch 1970), and try each possible sequence at each internal cladogram node by dynamic programming (Sankoff and Rousseau 1975). This type of explicit enumeration could be accomplished by extending the candidate set of sequences employed by search-based optimization (Wheeler 2003b) to include all possible sequences. Since this would entail an explosively increasing number of sequences this technique would become untenable rapidly. Some sort of branch-and-bound technique could be applied to this search given an initial upper-bound estimate, but it is unclear whether much additional headway can be made towards exact solutions.

5.6.2 Heuristic solutions

The operational goal of heuristic optimization procedures is to determine a set of HTU sequences that minimizes the overall cladogram length (= edge weight). Two general sorts of approach have been proposed based on attempts to estimate these internal vertex sequences using known sequences or on a search for them within the world of possible sequences.

The first-estimation heuristic was proposed by Sankoff et al. (1973) and Sankoff (1975). Given the high dimensionality of the exact recursive solution proposed by Sankoff (1975), a three-dimensional local-optimum heuristic was proposed. This would break the problem down into a series of single-point estimations surrounded by three known or previously estimated sequences (Fig. 5.3; as opposed to the two-point problem reduction in many heuristic alignment approaches). At the time, the method was too time-consuming for real data sets.

Wheeler (1996, 2002) proposed a two-dimensional heuristic (optimization alignment, later called direct optimization), which though more approximate that the three-dimensional approach, was more rapid (Fig. 5.4a). Later, Wheeler combined the Sankoff method with direct optimization and incremental character optimization (Gladstein 1997) in iterative-pass optimization,
**Figure 5.3** Median-state heuristic for n-dimensional optimization proposed by Sankoff (1975). The state of X (which could be an entire sequence) is that which minimizes the summed distances to the nodes which connect to it.

**Figure 5.4** Estimation methods. (a) Direct optimization (Wheeler 1996) results in a cladogram of cost 8 for the input sequences AA, ATTA, AAAA, and AAA when all events (indels and nucleotide substitutions) are equally costly. (b) Iterative-pass optimization (Wheeler 2003c) improves on this by 1 step. The horizontal bars signify indels and Δ represent nucleotide substitutions whose location may be ambiguous.

which improved cladogram-length calculations and can be used for larger numbers of sequences (Fig. 5.4b; Wheeler 2003c).

Sequence-search heuristics first appeared with fixed-state optimization (Wheeler 1999). The Fixed-state method limited the possible set of HTU sequences to those observed in terminal taxa, which are then diagnosed via dynamic programming based on a matrix of edit costs between the sequences. Given this constraint, less-satisfactory lower bounds on cladogram length are usually found (Fig. 5.5a; when sequences differ greatly in length this may not be true). Since the method is not calculating ancestral sequence states but simply optimizing states, cladogram optimization time, after initialization, is independent of sequence length. As the number of sequences increases, the number of potential sequence states rises as well, both improving the cladogram cost estimation and increasing the cost of computation of a given cladogram (roughly $m^3$ for $m$ sequences) (Fig. 5.5b).

Search-based optimization (Wheeler 2003b) relaxes the strict limit on sequence states by the addition of heuristically chosen sequences (Fig. 5.5c). Through the increase of the state set at...
the cost of execution time, progressively lower bounds can be found until further enlargement of the set is unproductive. The set could be made all-inclusive with an exact solution the result (but at great time cost).

5.7 Comparison of alignment and optimization

Although the goals of alignment (at least in phylogenetics) and optimization are the same—to find minimum-cost cladograms—the approaches are quite different. Alignment methods seek to find a single putative homology scheme upon which all cladograms are evaluated. Optimization methods perform this operation for each evaluated cladogram. As such, cladogram searches based on alignment methods are likely to be consistently faster than optimization approaches since the steps involved in determining the cost of a cladogram from a fixed alignment are much less burdensome. Optimization methods, however, are likely to find lower-cost cladograms (Wheeler 1996; Gribet et al. 2002; T. Grant. pers. comm.) and execution time comparisons should include the time consumed by alignment.

This can be illustrated by examining a simple set of three sequences (Fig. 5.6). There is not necessarily one globally optimal alignment. An alignment may be optimal for a particular cladogram (a la Sankoff and Cedergren) but any cladogram search based on such an alignment may well overlook other equal or lower-cost solutions. Optimization procedures, by examining the cladograms themselves, do not suffer this shortcoming (direct optimization as implemented in POY (Wheeler et al. 2003) finds both cladograms).

5.7.1 Evaluation

Given the identical goals of alignment and optimization, how can these somewhat competing methods be evaluated? Speed and effectiveness are two obvious criteria. Speed would be measured straight-forwardly as the time required to complete the combined alignment/cladogram-search operation versus that for the optimization-based cladogram search. The determination of cladogram cost for fixed alignments can be accomplished extremely efficiently (Goloboff 1994; 1998b) even for fairly general dynamic-programming characters (Goloboff 1995, 1996a). Implementations such as TNT (Goloboff et al. 2002) are able to evaluate many tens of thousands of cladograms (containing hundreds of taxa) per second. Multiple alignment implementations that generate a single multiple alignment (such as CLUSTAL) can create an alignment of a thousand nucleotides for a hundred taxa in a few minutes. Multiple-alignment procedures that evaluate many candidate multiple alignments (such as MALIGN) will absorb much more time. Such a search using dynamic homology optimization (at least under present implementations) could take yet longer.

An example (for illustrative purposes, not exhaustive by any means) is provided by the analysis of 100 mollusk 18S rRNA sequences (G. Giribet, personal communication). The alignment programs CLUSTAL and MALIGN were used and compared to optimization-based POY. CLUSTAL produced alignments most quickly and MALIGN most slowly. When comparing the approach of CLUSTAL and POY, CLUSTAL was faster by 20% (without cladogram search and when minimal POY options were specified), but the multiple alignment (really an implied alignment) produced by POY was 30% less costly in terms of parsimony (see Table 5.1). Cladogram searching would add time to the total solution of the alignment methods, but this would be a small premium.

![Figure 5.6 Simple alignments of four sequences GGGG, GGG, GAAG, and GAA. The alignments in (a) and (b) result in minimal cost, but different cladograms. The alignment in (c) yields both.](image-url)
Table 5.1 Performance of CLUSTAL, MALIGN, and POY on mollusc test set. The analyses were performed with all transformations (indels included) costing 1. The data set consisted of 100 mollusk 18S DNA sequences of approximately 1000 bp (G. Giribet, personal communication). All runs were on a Pentium M computer at 1.7 Ghz under LINUX. Runs for MALIGN and POY specified indel cost as 1. CLUSTAL was run twice; once under the default values (Default) and a second time specifying all gaps and transformations as 1 (1:1:1:1). The POY run with TBR branch swapping yields two equally costly cladograms. The arrows denotes the cost of the implied alignments when analyzed using NONA (Goloboff 1993b). NONA diagnosed the POY cladograms as the cost found by POY, but was able to find more-parsimonious solutions using the implied alignments.

<table>
<thead>
<tr>
<th>Method</th>
<th>Options</th>
<th>Execution time(s)</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLUSTALW</td>
<td>1:1:1:1</td>
<td>688</td>
<td>11999</td>
</tr>
<tr>
<td>CLUSTALW</td>
<td>Default</td>
<td>722</td>
<td>10642</td>
</tr>
<tr>
<td>MALIGN</td>
<td>'Build' only</td>
<td>26270</td>
<td>11790</td>
</tr>
<tr>
<td>POY + Implicated Alignment</td>
<td>'Build' only</td>
<td>920</td>
<td>7989 → 7970</td>
</tr>
<tr>
<td>POY + Implicated Alignment</td>
<td>TBR</td>
<td>134470</td>
<td>2 @ 7690 → 7684</td>
</tr>
</tbody>
</table>

in comparison with the alignment time. This could, however, narrow the CLUSTAL vs. POY execution-time difference.

The second criterion, effectiveness, favors optimization methods. Given that optimization methods are creating homology schemes specifically tailored to be optimal for each cladogram examined, it is only logical that this approach should result in less-costly (or higher-likelihood, for that matter) cladograms. This has been shown several times (Wheeler and Hiyashi 1998; Giribet et al. 2002; T. Grant, pers. comm.) and in the example above. Decreases in cladogram cost of 10% over alignment methods are not unexpected. The more length variation present in the sequences, the more opportunity there is for dynamic homology to find more-effective topology-specific solutions.

5.7.2 Interrelationship

There is a connection between multiple alignment and cladogram optimization. Transformation series inherent in a cladogram can be extracted and represented as an implied alignment (Wheeler 2003a). Such an implied alignment contains all the synapomorphy statements and transformation events required by the topology under an optimization approach. As such, they resemble standard multiple alignments, but are actually derived from the analysis of a specific cladograms as opposed to the basis for a search. Given the dependence of this sort of alignment on a specific cladogram, the object created is not necessarily fair to topologies other than its basis cladogram. Each of those topologies would be tested best by their own implied alignments. Such a unique-alignment procedure is the approach optimization methods bring to phylogenetic analysis.

An effect of this is seen in the calculations of Bremer (1994) support. Support values calculated on the basis of a global alignment can overestimate support compared to those based on dynamic homology. Given the specific homology schemes created by optimization methods, alternate cladogram lengths should be lower (or at worst equal) to those based on an alignment that is optimal for some other cladogram. This will tend to inflate the differences in cladogram lengths, hence Bremer values.

Alignment and optimization can be used in tandem to reduce execution time in optimization-based searches. In essence, an implied alignment (or any alignment for that matter), represents a static approximation of dynamic homology. Given that an implied alignment is generated for a specific cladogram, it can be used as the basis for rapid cladogram cost evaluations among similar topologies. The implied alignment is used to identify candidate cladograms quickly for further, more time consuming, analysis. If a cladogram is found to be superior to previously identified solutions, a new implied alignment is created based on the new topology and the process continued. This approach can accelerate searches
by a factor of four or more depending on the problem at hand (Wheeler 2003a).

5.8 Conclusion

Traditionally, alignment has been used to convert data without inherent putative homology statements into those that do. This step is operationally logical, but, given the ultimate goal of optimal cladograms, unnecessary. The criticism of optimization-based methods as lacking primary homology is largely based on this historical exercise. Clearly, *a priori* notions of homology (at least at the nucleotide level) are not logical or computational requisites of phylogenetic analysis. Criticisms of the optimization approach need to be based in effectiveness and logic—not on appeals to tradition.

As such, multiple alignment does not have separate standing in phylogenetic analysis. It is one approach to solving a complex, NP-complete problem. In comparison to optimization-based procedures, it may be fast, but it is approximate. In essence, alignment is a heuristic—and not a very effective one.

5.9 Acknowledgments

I thank Vic Albert for initiating and managing this effort, Gonzalo Gribet for use of unpublished sequences, Vic Albert, Lorenzo Prendini, Andres Varon, and an anonymous reviewer for helpful critique of the manuscript, NSF and NASA for support, and Steve Farris for decades of guidance.