SANDIA REPORT

SAND97-0400 • UC-605
Unlimited Release
Printed March 1997

Final Report for LDRD Project "A New Approach to Protein Function and Structure Prediction"

Cynthia A. Phillips

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

Approved for public release; distribution is unlimited.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.
NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Prices available from (615) 576-8401, FTS 626-8401

Available to the public from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Rd
Springfield, VA 22161

NTIS price codes
Printed copy: A03
Microfiche copy: A01

SAND97–0400
Unlimited Release
Printed April 1997

Distribution
Category UC–605

Final Report for LDRD project "A New Approach to Protein Function and Structure Prediction"

Cynthia A. Phillips
Algorithms and Discrete Mathematics Department
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-1110
caphill@cs.sandia.gov

Abstract

This report describes the research performed under the Laboratory-Directed Research and Development (LDRD) grant "A new approach to protein function and structure prediction", funded FY94-6. We describe the goals of the research, motivate and list our improvements to the state of the art in multiple sequence alignment and phylogeny (evolutionary tree) construction, but leave technical details to the six publications resulting from this work. At least three algorithms for phylogeny construction or tree consensus have been implemented and used by researchers outside of Sandia.

1 Introduction and Project Goals

The *primary structure* of a protein is its *amino acid sequence*, simply a list in order of the amino acids which make up the one-dimensional chain structure. In vivo, these proteins fold locally into small regular elements such as beta strands and alpha helices, giving the 2D or secondary structure. This in turn folds in three dimensions to form the 3D or tertiary structure. This tertiary structure largely determines the protein's function. The ability to accurately predict protein structure and/or function from amino acid sequences has application to disease understand, gene recognition, and drug design.

Ultimately it is the sequence that determines this 3D structure, but it is not yet known how to predict 3D structure from 1D structure (which can be determined more easily in the laboratory). Many methods are being investigated for this problem including first-principle physics simulations. In this work, we used the concept of mathematical similarity to locate conserved (highly similar) regions among sequences viewed as strings. One can then use databases of proteins with known structure, determined for example via crystalization and/or NMR. One can compare proteins with unknown structure to proteins with known structure, using the generally-believed assumption that proteins with highly-similar sequences will have highly-similar tertiary structure. This general idea is widely accepted in the biology and computer-science communities, but our emphasis is different, as detailed below.

The primary mathematical tools used to determine similarities are *multiple sequence alignment* and *phylogenetic (evolutionary) trees*. This research has concentrated on improving algorithmic technology for these tools.

A multiple alignment for a set of proteins matches each element in the sequence of a protein to another element (or a gap) in each other protein. Except for the possible addition of gaps, all sequences remain in order. See Figure 1 for an example. Considering for the moment just two sequences, the *edit distance* is what biologists consider a basic measure of the cost of changing one protein into the other by evolutionary methods. A match (two identical letters) costs nothing. A mismatch (matching two different amino acids) costs a constant amount dependent on the pair, essentially a measure of the likelihood of the substitution of one for the other in an evolutionary process. Inserting a gap in a sequence costs an amount determined by the gap function, a measure of the likelihood of insertion or deletion of amino acids in evolution.

The cost of a multiple sequence alignment builds upon these pairwise costs. In general, the cost of a multiple sequence alignment is the sum of the induced pairwise edit distances (how the pair is aligned within the multiple ensemble), with the sum taken over some subset of the pairs. Although the concept of cost for a pairwise alignment is relatively well-accepted [37], other than specification of parameters, there is no general agreement on a best multiple-alignment metric. The two most studied metrics are *sum of pairs* and *tree alignment*. In the sum-of-pairs metric, the pairwise edit distances are summed over all pairs. In the tree-alignment metric, on the assumption that the set of proteins evolved from a common ancestor, one constructs a tree with these sequences labeling the leaves (bottom nodes), fill internal nodes with hypothesized ancestors or original sequences, and sum only induced edit distances between each ancestor/descendent pair. Thus the alignment of most pairs is ignored.

Finding an optimal sum-of-pairs alignment is NP-complete, and finding an optimal tree alignment, even when the topology is given, is MAX SNP-complete [32]. Thus it is computationally difficult to find optimal solutions in general, particularly as the number of sequencess grows.

Dynamic programming can be used to find optimal multiple alignments for the metrics proposed to date. However, the running time becomes prohibitively large even for a small number of sequences. Pairwise optimal alignments can be computed efficiently however. A natural strategy is then to build a multiple alignment *progressively* using repeated efficient pairwise alignments, adding gaps as necessary. Thus a tree alignment suggests a progressive multiple alignment, aligning siblings and moving up the tree. This strategy is a valid heuristic for any metric, not just the tree-alignment score. The order in which pairs are aligned has a large effect on the quality of the final alignment [30]. One should align the sequencess in order of their biological similarity.

Because a good input topology is critical to the performance of progressive alignment heuristics, we also devoted considerable effort to the development of new phylogeny-producing technology. The topology for progressive alignment should be produced from data that is independent of the biosequences. In particular, we assume *character-based* data. Characters are functions from species (e.g. proteins) to states. For example, a character may map each protein to its species of origin (coded as an integer). Other examples of characters on proteins include numerical values of tests, length, fraction of locations with a certain amino acid (perhaps in ranges), or biological function. These phylogenies are scored with character-based metrics which are described in Section 2. Such phylogenies can give independent clues as to the closeness of pairs of proteins and used to order the progressive alignment.

Our approach to multiple sequence alignment and phylogenies had a unique emphasis. We sought improved algorithmic strategies for these two fundamental problems which had robust mathematically-provable performance in both speed and accuracy. For example, we sought provably good approximation algorithms whenever possible, not just heuristics. We sought to bring mathematically-good solutions closer to biologically-correct solutions by proposing new cost functions and new strategies for parameter selection. In general there are many (frequently exponentially many) optimal solutions based on currently-used cost functions and the "biologically correct" ones may not be in this set. Yet for a good scoring metric the biologically correct solution should be "near optimal". Structure common to many near-optimal (mathematically) solutions is likely to be biologically correct. One way to infer this common structure is to randomly sample the huge space of near-optimal solutions and apply machine learning techniques. Thus whenever possible we developed methods for randomly sampling these objects.

The remainder of this report is organized as follows. Section 2 describes new metrics for phylogeny, motivates them, and states our results. It also describes a new metric for combining data from many conflicting phylogenies. Section 3, describes and motivates a new metric for multiple sequence alignment and states our results. It also contains a summary of our improvements in pairwise sequence alignment technology: an improved parallel algorithm and a method for infering alignment parameters. Section 4 gives conclusions and future directions.

2 Phylogeny Results

The evolutionary biologist collects information on extant species (and fossil evidence) and attempts to infer the evolutionary history of a set of species. Most mathematical models of this process assume divergent evolution, meaning that once two species diverge, they never share genetic material again. Therefore, evolution is modeled as a tree (*phylogeny*), with extant species as leaves and (extant, extinct, or hypothesized) ancestors as internal nodes. Most of our work has followed this model, though we consider non-tree models suitable for computer viruses and bacterial evolution in Section 2.3. Species have been modeled in several ways, depending upon the nature of available information and the mechanism for gathering that information. Based upon these representations, differing measures of evolutionary distance and objective function are used to evaluate the goodness of a proposed evolutionary tree.

In our work we assumed that input data is character-based. Let S be an input set of n species. A *character* c is a function from the species set S to a set R_c of **states**. If we are given a set of characters $c_1, \ldots, c_k$ for S, each species is a vector from $R_{c_1} \times \ldots \times R_{c_k}$, and any such vector can represent a hypothesized ancestor. Characters can be used to model biomolecular data, such as a column in a multiple sequence alignment, but, as described above, we think of characters as morphological properties such as coloration or the ability to fly.

Character-based phylogenies are typically evaluated by some *parsimony-like* measure, meaning that the total evolutionary change is somehow minimized. Competing methods include maximum likelihood (e.g. [14]), when the data are believed to be generated under a stochastic model, and distance-based methods (e.g [1, 15]). Given a phylogenetic tree, a character c_i and a state $j \in R_{c_i}$, let ℓ_{ij} be the number of connected components in ${c_i}^{-1}(j)$ (the subtree induced by the species with state j in character i). The classic *parsimony problem* is to find a tree that minimizes the total number of changes over all (parent,child) pairs, namely to minimize $\sum_{c_i, j \in R_{c_i}} \ell_{ij}$. The *compatibility problem* is to maximize $|\{c_i : \ell_{ij} = 1 \text{ for all } j \in R_{c_i}\}|$. A character with this property (one component for each state) is called *compatible*. A *perfect phylogeny* is one where all the species containing a particular state for a particular character are connected (all are compatible). A perfect phylogeny is optimal under both parsimony and compatibility. All three problems (finding a perfect phylogeny, parsimony, compatibility) are $\mathcal{NP}$-complete [7, 10, 12, 29].

Phylogeny problems have *fixed-topology* variants, where in addition to the species set and characters, we are also given a tree T in which internal nodes are unlabeled, each leaf is labeled with a species $s \in S$ and each species $s \in S$ is the label of exactly one leaf of T. The fixed-topology variant of a metric is determining labels for the internal nodes (from the set of all possible choices of states for each character) so that the resulting phylogeny is optimal for the given metric, or, for some metrics, determining that no suitable labeling exists. If internal labels are chosen that do not correspond to a input species, they correspond to hypothesized ancestors. In the fixed-topology setting, optimal trees for the parsimony and compatibility metrics can be found in polynomial time using dynamic programming [16].

Fixed-topology algorithms can be used as *filters*. Current phylogeny-producing software can generate thousands of trees which are (approximately) equally good under some metric such as maximum likelihood or parsimony. We can think of these outputs as proposed

topologies. One way to differentiate these hypotheses is to see which topologies are also good under some secondary measure. For example, the original trees can be generated by biomolecular sequence data, and they can then be filtered using morphological data with slowly-evolving traits.

In the next three sections, we describe, motivate, and summarize results for new models of phylogeny. In Section 2.1 we describe a new parsimony-like metric *ℓ-phylogeny*, which is superior to parsimony and compatibility in settings where characters evolve slowly, but more than once (and thus there is no perfect phylogeny). In Section 2.2 we modify the definition of a character to include multiple states per character per species (*polymorphism*). This has application in biology and historical linguistics. In fact our methods have been used to help construct the evolutionary tree proposed by Warnow, Ringe, and Taylor for the Indo-European family of languages, which was presented by invitation at the National Academy of Sciences in November 1995. For both ℓ-phylogeny and polymorphic characters, we consider both fixed-topology problems, and problems where a topology must be computed. In Section 2.3, we consider evolution with multiple ancestors, that is, no longer a tree. This has application to bacterial and computer virus evolution.

Even with fixed-topology screening with secondary metrics, a practicing biologist may have many competing trees of approximately equal value under one optimization metric or may have many trees approximately optimal under differing optimization criteria. In situations where one must form a single hypothesis, one must compute a *consensus tree*. We examine methods for building consensus trees based upon the commonly-used *character-encoding* of trees. We show that these methods are only suitable when the input trees are all highly-resolved evolutionary trees. When the input trees are small (i.e. have few edges), the popular methods tend to perform badly in that the output trees also tend to be small, and hence there is a loss of evolutionary information. We developed a new consensus tree, the Asymmetric Median Tree (AMT). This consensus tree is appropriate for a larger range of data types, and will provably give at least as much information (in a well-defined sense) as the current popular methods on any data set. In Section 2.4, we define and motivate the AMT and summarize our results.

2.1 ℓ-Phylogeny

In this section we define, motivate, and summarize results for the ℓ-phylogeny metric. Further motivation and all technical details can be found in our publications [9, 18].

Parsimony and compatibility, defined above, each target a different type of character data and handle deviations from the assumptions differently. Parsimony targets the case where characters evolve slowly but not necessarily so as to produce compatible characters, and penalizes for each extra character state change without regard to how the extra changes are distributed. Compatibility targets the case where characters are presumed to evolve in such a way as to produce compatible characters, and penalizes for each *character* that is not compatible on the tree. Both criteria are used in practice for different types of datasets.

We proposed a new metric the *ℓ-phylogeny* metric, which combines the good aspects of both parsimony and compatibility. Given a phylogenetic tree, a character c_i and a state

$j \in R_{c_i}$, let ℓ_{ij} be the number of connected components in ${c_i}^{-1}(j)$ (the subtree induced by the species with state j in character i). A phylogeny is an ℓ-phylogeny if $\max_{c_i, j \in R_{c_i}} \ell_{ij} \leq \ell$. The *$\ell$-phylogeny problem* is to determine if an input consisting of a species set S and a set of characters $c_1, \ldots, c_k$ has an ℓ-phylogeny. The *phylogenetic number* problem is to determine the minimum ℓ such that it has an ℓ-phylogeny. The 1-phylogeny problem is the same as the perfect phylogeny problem.

The *generalized ℓ-phylogeny problem* is a variant of ℓ-phylogeny in which ℓ_i is specified for each character c_i. This allows the characters to be of varying types; thus, some can evolve quickly, and can potentially have many extra character state changes, while others may be compatible on the evolutionary tree, and others can fall between the two extremes. This model allows individual characters to follow the compatibility criteria or parsimony critera.

Parsimony, ℓ-phylogeny, and compatibility all allow states of a character to evolve multiple times. However, both parsimony and compatibility allow some characters to evolve many times. The ℓ-phylogeny metric requires *balanced* evolution, in that no one character can pay for most of the evolutionary changes. Thus, ℓ-phylogeny is a better measure than parsimony or compatibility in biological situations in which all characters are believed to evolve slowly.

Results: The hardness of 1-phylogeny problem [7, 29] implies that the phylogenetic number problem is NP-hard. We proved in [18] that for any fixed $\ell > 1$ the ℓ-phylogeny problem is also NP-hard. It is known that the fixed-topology 1-phylogeny problem can be solved in polynomial time[16]. In [18] we give a polynomial-time algorithm to solve the fixed-topology 2-phylogeny problem, but we proved that the fixed-topology ℓ-phylogeny problem is NP-hard for fixed $\ell > 2$. In fact, we show that the fixed-topology ℓ-phylogeny problem is NP-hard for fixed $\ell > 2$ even when the input is guaranteed to have an $\ell + 1$-phylogeny and the degree of the topology is restricted to be at most 3. We show that the fixed-topology phylogenetic number problem can be solved in polynomial time for fixed r, where r is the maximum number of states for any character.

The ℓ-phylogeny metric adheres to a common theme in combinatorial optimization:the more global nature of minimax makes it harder to compute than summation objectives, but also more useful. Although dynamic programming yields efficient algorithms for many fixed-topology phylogeny problems, it cannot be applied here because one cannot tell *a priori* which state will be limiting.

In [9], we give a simple 2-approximation for the fixed-topology ℓ-phylogeny problem that works for arbitrary input topologies. It is based on rounding the linear-programming relaxation of an integer programming formulation for fixed-topology ℓ-phylogeny. To our knowledge, this is the first application of linear-programming technology to phylogeny problems.

As we described earlier, ℓ-phylogeny is most appropriate for slowly-evolving characters. It is most restrictive (and hence most different from parsimony) when ℓ is small. Therefore, we look more closely at the first NP-hard case: $\ell = 3$. For this case, we give an optimal approximation algorithm based upon the structure of a 3-phylogeny that will construct a 4-phylogeny if the input instance has a 3-phylogeny.

In [18], we also considered the restricted ℓ-phylogeny problem, where one must contruct

a phylogeny on the input species without adding any hypothesized ancestors. We showed that there is a polynomial-time algorithm for the restricted 1-phylogeny problem, but the restricted ℓ-phylogeny problem is NP-hard for fixed $\ell \geq 2$.

Although the 1-phylogeny problem is NP-hard, it can be solved in polynomial time if the number, n, of species is fixed, or the number, k, of characters is fixed[3, 24], or the quantity $r = \max_j r_{c_j}$ is fixed[2, 23]. A full analysis of fixed parameter ℓ-phylogeny problems is outside the scope of our research. However, we observe that all of the phylogeny problems can be solved in polynomial time (by brute force) if n is fixed. We use interesting combinatorial techniques to show that for $k = 2$ the phylogenetic number problem can be solved in $O(n^2)$ time. The complexity of the ℓ-phylogeny problem remains open for fixed $\ell > 1$ and fixed $k > 2$. The difficulty of fixed-topology phylogeny problems does not change if k is fixed. On a related note, we showed that if r is fixed, there is a polynomial-delay algorithm for listing fixed-topology ℓ-phylogenies. We also show that for fixed $r \geq 2$ and fixed $\ell \geq 3$ the restricted ℓ-phylogeny problem is NP-hard. (This result follows from a more general result. Namely, we show that the restricted (ℓ_1, ℓ_2)-phylogeny problem is NP-hard for fixed $\ell_1 \geq 2$ and $\ell_2 \geq 2$ as long as one of ℓ_1, ℓ_2 is greater than 2.)

2.2 Phylogenies for Polymorphic Characters

In this section we define, motivate, and summarize results for the polymorphic-character model. Further motivation and all technical details can be found in our publication [8]. A character which is permitted to have more than one state on a given object will be called a *polymorphic character*, and (the traditional) one which can have only one state for every object is referred to as a *monomorphic* character.

A polymorphic character has a set of states for each character for each species. If there are r states, a polymorphic character is a function $c : S \rightarrow (2^{\{1,\ldots,r\}} - \emptyset)$. For a given set of species, the *load* is the maximum number of states for any character for any species. A perfect phylogeny for polymorphic characters, has all characters convex, meaning that for each state of each character, the set of species that contain that state are connected in the tree.

Polymorphism is well-documented in both the molecular genetics and comparative linguistics domains. For example, the population geneticist Masatoshi Nei writes: *The study of protein polymorphism has indicated that the extent of genetic variation in natural populations is enormous. However, the total amount of genetic variation cannot be known unless it is studied at the DNA level. The study of DNA polymorphism is still in its infancy, but the results so far obtained indicate that the extent of DNA polymorphism is far greater than that of protein polymorphism.** Polymorphism also arises in the comparison of different languages. The Indo-Europeanist Donald Ringe writes: *In choosing lexical characters we try to work with basic meanings (semantic slots), choosing from each language the word that most usually expresses each basic meaning. Languages typically have one word for each basic semantic slot, but instances of two (or even more) words apparently filling the same basic slot are not rare.[28]*

Thus, polymorphic data is a reality when working with evolutionary tree construction

*From [25], page 254.

for both linguistic analysis and biological taxa, and methods appropriate for such construction must be devised. In the phylogenetics literature and programs (such as Phylip, PAUP, and MacClade), algorithms and software to *evaluate* fixed leaf-labeled tree topologies for polymorphic data have explicitly required that the number of states be kept quite small because the evaluation requires time exponential in the number of states. This is the first algorithmic study of this problem to go beyond fixed topology problems for bounded number of states.

Results: Recent work in Historical Linguistics [36] has shown that perfect phylogenies should be obtainable from properly selected and encoded linguistic characters. In [8], we argue that a perfect phylogeny is also an appropriate objective when working with some biological data.

We began analyzing the complexity of the general problem of finding a polymorphic perfect phylogeny with minimum load. For an input instance with n species, k polymorphic characters with at most r states per character, we have shown that finding a minimum-load perfect phylogeny can be solved efficiently provided n is fixed or $r = 2$. However, it is a $\mathcal{NP}$-complete even if k is fixed at any value or even if r is a fixed value at least 3. For the special case of constructing a 2-load phylogeny, or determining that one does not exist, we have shown this problem can be solved efficiently for fixed n, if $r = 2$, or if k is fixed. However is it $\mathcal{NP}$-complete for all fixed $r \geq 3$.

We present two algorithms, one graph theoretic and one combinatorial, for the problem of inferring perfect phylogenies from polymorphic data. Our previous complexity analysis shows that *any* exact algorithm *must* be exponential in *both* k (number of characters) and l (load), under the well-believed $P \neq NP$ assumption[†]. The combinatorial algorithm runs in time $O(r^{lk+1}lkn)$. The graph-theoretic algorithm runs in time $O(nk^2l^2 + (rk^3l^2)^{kl+1})$. Both algorithms can be extended to allow character-specific load bounds rather than a single bound l applied to all characters. We can enumerate or randomly sample solutions.

We present a methodology for inferring perfect phylogenies from data which combine monomorphic and polymorphic data. The methodology we propose here significantly extends the range of the data that can be analyzed in Historical Linguistics. We have applied this methodology to the data set studied by Warnow, Ringe, and Taylor [36] to analyze the Indo-European family of languages, whose first-order subgrouping had been argued for decades without resolution. Detection and resolution of polymorphism led to a modification of their initially proposed phylogeny, which was based only on monomorphic characters. Our methodology and its results were presented at the Symposium on the Frontiers of Science at the National Academy of Sciences in November 1995.

We consider the problem of inferring evolutionary trees from polymorphic data when a perfect phylogeny is an unlikely outcome. We extend the standard parsimony costs used for monomorphic characters to polymorphic characters, and investigate the complexity of the fixed-topology problem with these new cost functions. We show that when the load is part of the input, the problem is NP-complete. When the load l is fixed we give an $O(nk(2r)^l)$-time algorithm to find labels for the internal nodes to maximize global convexity. Here n is the number of species, k is the number of characters, and r is the maximum number of states in any character. For more general cost functions including loss and duplication

[†]Actually an algorithm can be exponential in both k and r, but since $l \leq r$, l is the better bound.

costs, we can find a most parsimonious solution in time $O(nkl(2r)^l)$. Finally, for the most general form of cost, where mutations, losses, and duplications are state-dependent, we give an $O(nkr^{2l})$-time algorithm. We can randomly sample optimal solutions.

An (α, β)-approximation algorithm for the load problem computes a phylogeny with load at most $\alpha\ell$ and cost at most βc provided there is a load-ℓ cost-c phylogeny. Note that this is a pseudoapproximation algorithm, since the cost of the best $\alpha\ell$-load phylogeny may be significantly lower than the cost of the best ℓ-load phylogeny. In [9], we give a polynomial-time LP-based $(\alpha, \frac{\alpha}{\alpha-1})$-approximation algorithm for any $\alpha > 1$ when the cost of losing a state from parent to child is zero and the input topology is arbitrary. Note that taking $\alpha = 2$ gives a $(2, 2)$-approximation algorithm.

2.3 Viral phylogenies

In this section we define, motivate, and summarize results for new phylogenetic structures motivated by computer viruses and bacteria. Further motivation and all technical details can be found in our publication [17]. One of the coauthors, Greg Sorkin, is a member of the computer anti-virus group at IBM research. This work is implemented and has been used.

There are now several thousand different computer viruses in existence, with new ones being written at a rate of 3 to 4 per day. Most of these are based upon previous ones: someone copies and modifies a virus, or creates a new virus with subroutines borrowed from one or more *ancestors*. We model a computer virus as a set of species on *binary* characters (i.e. states 0 and 1) only. Each binary character represents the presence or absence of a byte string (piece of code).

We assume that each code fragment is invented only once. For sufficiently long fragments this is justified by differences in programming style, the many possible orderings of unconstrained events, etc. We model the evolution of a set of viral species with a directed graph in which an edge $s_i \rightarrow s_j$ indicates that species s_i is an ancestor of species s_j (i.e. s_j inherited some character(s) from s_i). A *phyloDAG* for input species $\mathcal{S}$ and characters $\mathcal{C}$ is a directed acyclic graph (DAG) with node set $\mathcal{S}$. For each character $c \in \mathcal{C}$, the subgraph induced by species with a 1 is character c is connected, in the sense that from a single *archetype* $a_c \in S_c$ there is a directed path, within S_c, to every other $s \in S_c$.

The phyloDAG model allows the possibility that a species may be derived from several ancestors rather than from a single ancestor, which is motivated in detail in [17]. A phyloDAG exists for any inputs $(\mathcal{S}, \mathcal{C})$: for any chronology ascribed to the species (i.e. any total ordering of the species set), the directed graph with edges from each species to all later species is a phyloDAG. However, every pair of species is related by an edge in this graph. Since most virus species presumably have few ancestors, we seek a *Minimum PhyloDAG*, one with a minimum number of directed edges.

Our approach to the evolution problem corresponds to a *restricted* model of evolution: one in which we are not allowed to introduce hypothetical species outside of the input set. This model is well-suited to computer viruses, where because of good world-wide communications, sharing of data between anti-virus organizations, and the brief history involved, there are likely to be very few gaps in our viral database. This is less reasonable for the

biological setting. However, to prevent trivial solutions, we would have to define additional biologically-reasonable restrictions on the hypothetical species that can be added.

Results: We showed that the Minimum PhyloDAG problem is "hard": in polynomial time, it cannot be solved exactly unless $\mathcal{P} = \mathcal{NP}$, nor can it approximated to within better than a logarithmic factor unless $\mathcal{NP} \subseteq \mathrm{DTIME}(n^{O(\log\log n)})$. In fact, we know of no way to approximate Minimum PhyloDAG to within a logarithmic factor: we show that various natural greedy strategies (including randomized ones) do not even approximate within a factor of cn.

Because of the difficulty of the phyloDAG problem, we considered two variants. In the first variant, we require that each species have just one ancestor, so that the phyloDAG is an arborescence (a tree with edges directed away from a root). The undirected version is called a *0–1–0 phylogeny*. We give two polynomial-time algorithms to randomly sample 0–1–0 phylogenies if any exist.

The first algorithm computes a concise data structure that represents all 0–1–0 phylogenies for the input data and can be used to select a phylogeny uniformly at random in time $O(n\ell)$, where ℓ is the total number of 1's in the representation of the input species. When no solution exists the algorithm returns a *witness set*: a concise indication of *why* there can be no phylogenetic tree.

The second algorithm characterizes a 0–1–0 phylogeny of the input species set as a minimum spanning tree (MST) of a particular undirected edge-weighted graph. With it, 0–1–0 phylogenies can be constructed in deterministic time $O(\ell n + n^2 \log n)$ or (with high probability) in randomized time $O(\ell n)$, and sampled uniformly at random in time $O(\ell n + M(n))$, where $M(n)$ is the time needed to multiply two $n \times n$ matrices. It does not produce a concise witness when there is no 0–1–0 phylogeny, but it uses the MST subroutine which is simple and widely available.

The second variant of phyloDAG is simply its undirected analogue. A *phylograph* for species $\mathcal{S}$ and characters $\mathcal{C}$ is an undirected graph with vertex set $\mathcal{S}$, with the property that the subgraph induced by the species containing a 1 for character $c \in \mathcal{C}$ is connected. The *Minimum Phylograph* problem is to find a phylograph with the minimum number of edges. We show that it is hard to approximate Minimum Phylograph within a factor less than $\frac{1}{4}\ln \ell$, but give a simple greedy algorithm that approximates the minimum number of edges in a phylograph within a factor of $\ln \ell$.

2.4 Consensus Trees

In this section we define, motivate, and summarize results for a new metric for finding a consensus tree. Further motivation and all technical details can be found in our publication [27]. This new metric may be incorporated into current software packages used by practicing biologists.

The problem of finding a consensus tree arises when a practicing biologist is confronted with a set of different possible phylogenies, each of which seems equally plausible. One of the reasons this arises is that different optimization criteria may be used to construct the tree, and each criterion may produce several optimal trees. In practice, finding optimal or even near-optimal solutions can be difficult, and therefore approximation algorithms

or heuristics are used, which can produce many equally-plausible solutions. Additionally, sometimes different data sets are used to represent the same organisms. Thus handling multiple hypotheses of evolution is a necessity in evolutionary tree construction methodology. In the end the objective is, if possible, a single *consensus* evolutionary tree.

In keeping with our other research, we considered character-based methods where all species must appear in the consensus tree. Given a tree on a species set, an edge encodes a binary character on the species: all species on one side are given a 1 for that character, and all species on the other side are given a 0. The *character encoding* of a tree T, denoted $C(T)$ is therefore, the set of characters representing exactly the edges of the tree. Two characters are *compatible* if they can be in the same tree (essentially if the species given 1 by the two characters do not properly intersect).

We briefly define the five main character-based consensus methods used previously. Let $\mathcal{T} = \{T_1, T_2, \ldots, T_k\}$ be a profile of trees, each leaf-labeled by S, with $|S| = n$.

The Compatibility Tree [13, 19, 35]: In the classical *Tree Compatibility Problem*, we wish to find T such that $C(T) = \cup_i C(T_i)$. When such a tree exists, the profile $\mathcal{T}$ is said to be "compatible," and T is called the *compatibility tree*. Determining whether the compatibility tree exists and constructing it when it does can be done in $O(nk)$ time.

The Strict Consensus[5, 11]: The *strict consensus tree* contains only the common information; that is, the strict consensus tree T satisfies $C(T) = \cap_i C(T_i)$. The strict consensus tree always exists, and can be constructed in $O(nk)$ time.

The Majority Tree[34]: The *majority tree* contains exactly characters that appear in more than half the input trees. That is, $C(T) = \{\alpha : |\{i : \alpha \in C(T_i)\}| > k/2\}$. The majority tree is unique and always exists, and can be constructed in $O(nk)$ time.

The Median Tree[6]: Letting $\triangle$ denote the symmetric difference, a median tree minimizes the function $f_{med}(T, \mathcal{T}) = \sum_i |C(T) \triangle C(T_i)|$. The majority tree is a median tree, so that there is always at least one median tree and it can be constructed in $O(nk)$ time. When k is even, the median tree can also contain characters appearing in *exactly* half the input trees.

The Nelson Consensus[26]: Given a profile of trees and character $c : S \rightarrow \{0, 1\}$, we weight c by the number of trees in the profile whose encodings include c. Thus $w(c) = |\{i : c \in C(T_i)|\}$. Characters not contained in any profile's encoding have weight 0. Given a profile $P = T_1, T_2, \ldots, T_k$ of trees, we define the *Nelson basis* of P to be the set $NB(P)$ of trees T such that $Nelson(T) \equiv w(T) - |C(T)|$ is maximum over all trees on S. Note that characters c which are not in $\cup C(T_i)$ will not be in any tree in the Nelson Basis, because the removal of such characters (by contracting the edge defining that character) results in a tree of higher value. Thus, all trees in the Nelson Basis satisfy $C(T) \subset \cup C(T_i)$. The Nelson tree is then defined to be the strict consensus of the trees in the Nelson Basis; i.e. the unique tree T_N such that $C(T_N) = \cap_{T \in NB(P)} C(T)$. The complexity of computing the Nelson tree is unknown, but probably difficult.

Of these models the median tree is perhaps the most used in the biological community since it provides more evolutionary information than the strict consensus tree, is efficiently computable, and always exists; but the Nelson Tree is also popular. The compatibility tree seems desirable, but is rarely found since if any two characters are incompatible, the compatibility tree will not exist. On many data sets both the strict consensus and the

majority tree can be rather unresolved, and hence may not indicate many decisions about the evolutionary history of the taxa. We assume that all input trees are equally reliable when they do make an evolutionary decision, or at least that they meet some user-specified "reliability threshold".

Motivated by a desire to produce resolved consensus trees, we defined a new consensus tree, which we call an *Asymmetric Median Tree (AMT)*. If we expand f_{med}, the median-tree objective function, we find that $f_{med}(T, \mathcal{T}) = \sum_i [|C(T) - C(T_i)| + |C(T_i) - C(T)|]$. This explicitly penalizes for characters in T which are not in at least half of the trees in $\mathcal{T}$. This seemed an unreasonable requirement for the consensus tree. In the AMT, essentially, characters are included weighted by the number of input trees that contain them. The exact, somewhat techical, definition is given in [27].

Results: We showed that the computation of the optimal AMT for k trees is equivalent to the maximum independent set problem on k-colored graphs. We used this characterization to obtain the following algorithmic results. We found an $O(n^{2.5})$-time algorithm to find an asymmetric median tree for a profile of two trees on n species. We found a polynomial-delay algorithm to enumerate all asymmetric median trees for a profile of two trees, which outputs a new AMT in time at most $O(n^{3.5})$. We present an $O(ndk^d)$-time algorithm to determine if a degree-d asymmetric median tree exists for an arbitrary profile of k trees on n species. In time $O(nD^2k^D)$ we can determine the minimum D such that an AMT of degree D exists and construct such an AMT. We present a polynomial-time algorithm to *approximate* the asymmetric median tree for an arbitrary profile of trees of k trees, with an approximation guarantee of $2/k$. We also give a second algorithm that is better when the value of the AMT is large.

We obtained the following hardness results. For three or more arbitrary trees, and for an unbounded number of binary trees, finding an asymmetric median tree is NP-hard. Approximating the AMT of an unbounded number of arbitrary trees to within a polynomial factor is hard.

We propose a measure of how informative an evolutionary tree is, which we call the *degree of resolution*, and we show that for any profile of trees, both the exact and approximate solutions for the asymmetric median tree are at least as informative (according to the definition we will propose) as the Nelson Consensus, the majority tree, the strict consensus tree, and any median tree. We show that when the compatibility tree exists, our methods (both exact and approximate) will return it as the asymmetric median tree. Our results therefore provide methods for efficiently inferring from an arbitrary profile of evolutionary trees a consensus tree that contains at least as much information, and potentially significantly more information, than the most popular methods used today for consensus tree construction.

3 Sequence-Alignment Results

Multiple sequence alignment is a classic tool for comparing biosequences which is currently used to predict protein structure, hypothesize evolution, and correct errors in shotgun sequencing. In this section we motivate and summarize results for research in multiple

sequence alignment. Further motivation and all technical details can be found in our publication [4].

The two measures of multiple-alignment cost currently used, tree alignment and sum-of-pairs, do not allow the fairness of interaction we feel is critical when aligning a set of proteins hypothesized to be highly similar. The cost of a multiple alignment in the sum-of-pairs metric is the sum over all pairs of induced edit distances. This seems fair, in that all pairs contribute to the cost. However, pairs of sequences which are less closely related will get higher priority, which seems counter to our goal. Furthermore, since it is NP-complete (formally intractable) to compute the optimal alignment, one must approximate the optimal, which exacerbates this effect. The worst-case pair of sequences (most disparate) allows "breathing room" in an approximation, since it provides a lower bound. Doubling this cost, for instance could allow closely-related sequences to be quite poorly aligned. Furthermore, current approximation algorithms for sum-of-pairs rely upon computing optimal alignments for certain pairs, which leads to "dictator" sequences. That is, alignments tend to fix certain sequences and modify others with gaps to agree with them, instead of allowing all sequences to affect others to find global similarities. More specifically, the first approximation algorithm for sum-of-pairs, aligned all sequences against the single most "central" sequence [20]. This yields an alignment with cost no more than a factor of $2 - 2/k$ greater than optimal for k sequences. This alignment, though theoretically and even practically good in terms of its closeness to the optimal sum-of-pairs cost, ignores all other pairwise interactions. Therefore, we feel that sum-of-pairs is not the "fairest" way to measure all pairwise interactions.

Tree alignments place the sequences in a phylogeny and the cost is the sum of the edit distance in the induced alignment (how the pair is aligned within the multiple ensemble) only between ancestor/descendent pairs. Thus the alignment of most pairs is ignored. Note that we agree that a character-based phylogeny can be a guide to sequence alignment, but disagree with this as a scoring metric for multiple alignments. Jiang, Lawler, and Wang [22] give a 2-approximation for the fixed-topology tree alignment problem with bounded-degree input topologies. They prove that the best lifted tree (in which the label of each internal node is equal to the label of one of its children) is within a factor of 2 of the best tree with arbitrary labels. Gusfield and Wang [31] take the approach of [22] a step further by proving that the best uniform lifted tree (ULT) is within a factor of 2 of the best arbitrarily-labeled tree. In a uniform lifted tree on each level, all internal nodes are labeled by the same child (e.g. all nodes at level one take the label of their leftmost child). There are only a small number of choices for ULT's and they are almost certainly not the biologically-correct model of evolution. Yet, they provide a good approximation for tree alignment[‡]. In [20], Gusfield proves that a minimum spanning tree built on the species set provides a tree alignment less than a factor of 2 of optimal.

We propose a new metric for scoring multiple sequence alignments: *minimum dilation*. Like the popular tree-alignment metric, highly-disparate sequences do not have a large impact on the structure of a min-dilation alignment. However, sequences which are closely related have full influence on the alignment of that set, much as with the sum-of-pairs

[‡]There are dynamic-programming-based polynomial-time approximation schemes (PTAS) for this problem which are more computationally expensive and should give more interesting alignments. The first PTAS was given in [22] and the currently most efficient one is in [33].

metric but in a fairer way. See [21] for a subsequent discussion proposing other metrics for multiple sequence alignment.

3.1 Minimum-Dilation Alignment

Suppose we are given k sequences $s_1, \ldots, s_k$, each of length at most n. Let $d(i,j)$ be the cost of an optimal pairwise alignment between sequences s_i and s_j. Consider a multiple alignment and let $D(i,j)$ be the induced distance between sequences s_i and s_j in this multiple alignment. The *dilation* between this pair $\alpha(i,j) = D(i,j)/d(i,j)$. The cost of an alignment under the dilation metric is $max_{ij}\alpha(i,j)$.

Figure 1 illustrates two different multiple alignments of the set of cyclic shifts of the sequence 123456. A gap is represented by a dash. The top "staircase" alignment is optimal for the both sum-of-pairs and tree alignment. The tree for the alignment is a chain connecting 123456 to 234561, connecting 234561 with 345612, and so on through each cyclic shift in order ending with 612345. The choice of which sequence to put at the top of the staircase is arbitrary, so there are five other equally good alignments. Sequences 123456 and 612345, though closely related (edit distance 2 for the case where matching with a gap costs one for all characters), but they are very poorly aligned (induced edit distance of 10 for a dilation of 5). Each of the other five alignments does an equally poor job with some other pair. The lower alignment has a dilation of 3. Although in general one will will be working with a bounded alphabet size (possible sequence elements), this example can be generalized to the cyclic shifts on $1, \ldots, n$. The staircase alignment has a dilation of about n and the generalized double-staircase has a dilation of 3.

The primary advantage of tree alignments is that clusters of related sequences are aligned (albeit sparsely) with each other, and one can ignore interactions with sequences that are evolutionarily disjoint. Because the min-dilation metric keeps more closely-related sequences closer together, it gives this clustering automatically. However, among closely-related sequences, the min-dilation metric allows full interaction in a much fairer way than sum-of-pairs.

Results: We began to investigate the structure of optimal min-dilation alignments and the complexity of computing them. We give examples of sequence sets which counterintuitively have low dilation alignments. We show there are set of k sequences of maximum length n which have minimum dilation $\Omega(k) = \Omega(\sqrt{n})$. Progressive alignment, namely optimally aligning $k-1$ sequences, is a common heuristic that leads to 2-approximations for both tree alignment and sum of pairs. This cannot be the case for minimum dilation: we give an example where all progressive alignments have dilation $\Omega(k)$ times the optimal dilation. A progressive alignment aligns $k-1$ pairs optimally, but sometimes it is better to have fewer optimal pairs to reduce global error. The progessive alignment that yields a 2-approximation for sum of pairs also yields an existentially-optimal $(k-1)$-approximation for dilation. We give dynamic programming algorithms which find optimal minimum-dilation alignments and run in polynomial time for fixed k. Finally we give a valid integer linear programming formulation for the minimum-dilation problem and suggest possible heuristic solution methods.

Although we feel optimal or near-optimal minimum-dilation alignments could provide biological insight, we feel that more positive results, in the form of efficient approximation

1	2	3	4	5	6	–	–	–	–	–
–	2	3	4	5	6	1	–	–	–	–
–	–	3	4	5	6	1	2	–	–	–
–	–	–	4	5	6	1	2	3	–	–
–	–	–	–	5	6	1	2	3	4	–
–	–	–	–	–	6	1	2	3	4	5

(a)

–	–	–	–	–	1	2	3	4	5	6	–	–	–
2	3	–	–	–	–	–	–	4	5	6	1	–	–
–	3	4	5	–	–	–	–	–	–	6	1	2	–
–	–	4	5	6	1	–	–	–	–	–	–	2	3
–	–	–	5	6	1	2	3	4	–	–	–	–	–
–	–	–	–	6	1	2	3	4	5	–	–	–	–

(b)

Figure 1: Two different multiple alignments of the cyclic shifts of the sequence 123456. Dashes indicate gaps. (a) The "staircase" alignment is optimal for the both sum-of-pairs and tree alignment (using a tree that is a chain linking the sequences top to bottom). (b) This alignment has better dilation.

algorithms or heuristics or more efficient exact algorithms for small k, are necessary before this metric can be used in practice.

3.2 Parallel Pairwise Sequence Alignment

Evaluating the dilation of an alignment requires an algorithm for computing an optimal alignment between a pair of sequences, and such a procedure is a bottleneck in virtually all current heuristics for multiple sequence alignment. We present a new parallel algorithm to align two length-n sequences which is the most work-efficient (product of time and number of processors) to date at the cost of a $O(1 + 1/\lg n)$-factor loss in accuracy. More precisely, we dropped the work bound from $O(n^2 \lg^3 n)$ for the best previous deterministic algorithm (which exploited the grid structure of the computation) to $O(n^2(\lg\lg n)^3)$, significantly closer to the best possible bound of $O(n^2)$.

3.3 Learning Edit-Distance Parameters

All multiple-alignment metrics are based upon the edit distance between two sequences. Intuitively, this measures the cost of changing one into the other by a series of insertions, deletions, and substitutions. The fully general model has a cost defined for every possible transition over the entire alphabet of the sequences. Commonly, however, one defines a single mutation and gap (insertion/deletion) cost. The optimal alignment is sensitive to the choice of parameters, and there is no agreement upon reasonable parameter values. We sketch an algorithm that can learn values of these parameters so that a set of known proteins are properly classified. This means that proteins that are more closely related have lower-cost optimal alignments than proteins that are not as closely related.

4 Conclusions

This research has resulted in 2 journal publications[18, 27] (each appearing previously in refereed conferences), 2 other refereed conference papers[8, 17] (submitted for journal publication), 1 paper submitted to a refereed conference[9] (soon to be submitted to a journal), and one SAND technical report[4]. At least 3 different types of algorithms have been implemented and applied to real data (AMT, viral phyogenies, polymorphic phylogenies). The polymorphism work was invited for presentation at Symposium on the Frontiers of Science at the National Academy of Sciences in November 1995, the Second Sandia National Laboratories Workshop on Computational Molecular Biology in March 1996, and the DIMACS workshop on Mathmatics of Hierarchies and Biology, November 1996.

Although we have made significant algorithmic progress on the tools of phylogeny construction and multiple sequence alignment, additional work remains for other researchers. Minimum-dilation alignment cannot be practical until there exist better (approximation) algorithms. Proteins can be coded as binary strings, representing the hydrophobic-hydrophilic nature of the amino acids. Thus improved algorithms for binary input sequences would be of value. One important step to practical application of these algorithms would be the application of rigorous experimental analysis techniques to the competing methods for

constructing multiple sequence alignments (both directly, and using progressive alignments keyed off phylogenies constructed in various ways), using real data. Another important step is the implementation of learning-theoretic algorithms to infer structure from (near) optimal solutions using the random-sampling techniques we have developed.

Acknowledgments Colleagues participating in the research include: Jonathan Atkins (Infinity Financial Technology), Maria Bonet (University of Pennsylvania), Leslie Goldberg (University of Warwick), Paul Goldberg (Aston University), Phil MacKenzie (Boise State University), R. Ravi (Carnegie Melon University), Greg Sorkin (IBM T.J. Watson Research Center), Tandy Warnow (University of Pennsylvania), and Shibu Yooseph (University of Pennsylvania). We thank Sorin Istrail and Bill Hart of Sandia National Laboratories for useful discussions during this research.

References

[1] R. Agarwala, V. Bafna, M. Farach, B. Narayanan, M. Paterson, and M. Thorup, "On the approximability of numerical taxonomy: fitting distances by tree metrics", *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, 1996, pp. 365-372.

[2] R. Agarwala, D. Fernández-Baca, "A Polynomial-Time Algorithm for the Perfect Phylogeny Problem when the Number of Character States is Fixed", *Proceedings of the 34th annual Symposium on Foundations of Computer Science*, 1993, pp. 140–147.

[3] R. Agarwala and D. Fernandez-Baca, "Fast and Simple Algorithms for Perfect Phylogeny and Triangulating Colored Graphs", *DIMACS Tech. Rept.* #94-51, 1994.

[4] J.E. Atkins, P.W. Goldberg, P. Mackenzie, C.A. Phillips, and R. Ravi, "Minimum-dilation multiple sequence alignment", Sandia National Laboratories Technical Report number SAND97-0273, 1997, to appear.

[5] J. Barthélemy, and F. Janowitz, "A formal theory of consensus", *SIAM Journal on Discrete Mathematics*, Vol. 3, 1991, pp. 305-322.

[6] J. Barthélemy, and F. McMorris, "The median procedure for n-Trees". *Journal of Classification*, Vol. 3, 1986, pp. 329-334.

[7] H. Bodlaender, M. Fellows, T. Warnow, "Two Strikes Against Perfect Phylogeny", *Proceedings of the 19th International Congress on Automata, Languages and Programming (ICALP)*, pp. 273-287, Springer-Verlag Lecture Notes in Computer Science, 1992.

[8] M. Bonet, C. Phillips, T.J. Warnow and S. Yooseph, "Constructing Evolutionary Trees in the Presence of Polymorphic Characters", *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing* (1996).

[9] M. Cryan, L.A. Goldberg, and C.A. Phillips, "Approximation algorithms for the fixed-topology phylogenetic number problem", submitted.

[10] W.H.E. Day, "Computationally difficult parsimony problems in phylogenetic systematics", *Journal of Theoretical Biology*, Vol 103 (1983).

[11] W.H.E. Day, "Optimal algorithms for comparing trees with labelled leaves", *Journal of Classification*, Vol. 2, 1985, pp. 7-28.

[12] W.H.E. Day and D. Sankoff, "Computational complexity of inferring phylogenies by compatibility", *Systematic Zoology*, Vol. 35, No. 2, 1986, pp. 224-229.

[13] G.F. Estabrook, and F.R. McMorris, "When is one estimate of evolutionary relationships a refinement of another?", *J. Math. Biosci.*, Vol. 10, 1980, pp. 327-373.

[14] M. Farach and S. Kannan, "Efficient algorithms for inverting evolution", *Proceedings of the 28th Annual Symposium on the Theory of Computing*, May 1996.

[15] M. Farach, S. Kannan, and T. Warnow, "A robust model for finding optimal evolutionary trees", *Algorithmica*, Vol. 13, No. 1, 1995, pp. 155-179.

[16] W. Fitch, "Towards defining the course of evolution: minimum change for a specified tree topology", *Systematic Zoology*, Vol 20, 1971.

[17] L.A. Goldberg, P.W. Goldberg, C.A. Phillips, and G. B. Sorkin, "Constructing Computer Virus Phylogenies", *Proceedings of the 7th Symposium on Combinatorial Pattern Matching*, Lecture Notes in Computer Science 1075, Springer, June 1996, pp. 253-270.

[18] L.A. Goldberg, P.W. Goldberg, C.A. Phillips, E. Sweedyk, and T. Warnow, "Minimizing phylogenetic number to find good evolutionary trees", *Discrete Applied Mathematics*, Volume 71, Numbers 1-3, 5 December 1996, pages 111-136.

[19] D. Gusfield, "Efficient algorithms for inferring evolutionary trees", *Networks*, Vol. 21, 1991, pp. 19-28.

[20] D. Gusfield, "Efficient methods for multiple sequence alignment with guaranteed error bounds", *Bulletin of Mathematical Biology*, Vol. 55, 1993, pp. 141-154.

[21] S. Istrail and R. Ravi, "What is a good objective function for multiple sequence alignment?", unpublished manuscript, August 1996.

[22] T. Jiang, E.L. Lawler and L. Wang, "Aligning Sequences via an Evolutionary Tree: Complexity and Approximation", *Proceedings of the 26th Annual ACM Symposium on the Theory of Computing*, 1994, pp. 760-769.

[23] S. Kannan and T. Warnow, "A Fast Algorithm for the Computation and Enumeration of Perfect Phylogenies when the Number of Character States is Fixed", *Proceedings of the 6th ACM/SIAM Symposium on Discrete Algorithms*, pp. 595-603, 1995.

[24] F.R. McMorris, T. Warnow, T. Wimer, "Triangulating Colored Graphs", *SIAM J. on Discrete Mathematics*, Vol. 7, No. 2, pp. 296-306, 1994.

[25] M. Nei, *Molecular Evolutionary Genetics*, Columbia University Press, New York. 1987.

[26] G. Nelson, "Cladistic analysis and synthesis: Principles and definitions, with a historical note on Adanson's *Familles des Plantes* (1763-1764)", *Syst. Zool.*, Vol. 28, 1979, pp. 1-21.

[27] C.A. Phillips and T.J. Warnow, "The asymmetric median tree - a new model for building consensus trees", *Discrete Applied Mathjematics*, Volume 71, Numbers 1-3, 5 December 1996, pages 311-336.

[28] D. Ringe, personal communication, 1995.

[29] M.A. Steel, "The complexity of reconstructing trees from qualitative characters and subtrees", *Journal of Classification*, Vol. 9, pp. 91-116, 1992.

[30] M. Vingron, "Multiple sequence alignment and applications in molecular biology", Preprint 91-12, Universität Hiedelberg, 1991.

[31] L. Wang and D. Gusfield, "Improved Approximation Algorithms for Tree Alignment", *Proceedings of the 7th Symposium on Combinatorial Pattern Matching*, Lecture Notes in Computer Science 1075, Springer, June 1996, pp. 220–233.

[32] L. Wang and T. Jiang, "On the complexity of multiple sequence alignment", *Journal of Computational Biology*, Vol. 1, No. 4, 1994, pp. 337-48.

[33] L. Wang, T. Jiang, and D. Gusfield, "A more efficient approximation scheme for tree alignment", *Proceedings of the of First Annual International Conference on Computational Molecular Biology*, Jan. 1997, pp. 310–319.

[34] H.T. Wareham, "An Efficient Algorithm for Computing Ml Consensus Trees", Honors Dissertation, Department of Computer Science, Memorial University of Newfoundland, St. John's, Newfoundland, 1985.

[35] T.J. Warnow, "Tree compatibility and inferring evolutionary history", *Journal of Algorithms*, Vol. 16, 1994, pp. 388-407.

[36] T. Warnow, D. Ringe and A. Taylor, "A character based method for reconstructing evolutionary history for natural languages", Tech Report, Institute for Research in Cognitive Science, 1995, and *Proceedings 1996 ACM/SIAM Symposium on Discrete Algorithms*, pp. 314–322.

[37] M. Waterman, *Introduction to Computational Biology: Maps, Sequences, and Genomes*, Chapman & Hall, London, 1995.