Large-scale multiple sequence alignment and phylogeny estimation

Tandy Warnow[1,*]

1 Department of Computer Science, University of Texas at Austin, Austin, TX, USA

∗ E-mail: tandy@cs.utexas.edu

Abstract

With the advent of next generation sequencing technologies, alignment and phylogeny estimation of datasets with thousands of sequences is being attempted. To address these challenges, new algorithmic approaches have been developed that have been able to provide substantial improvements over standard methods. This paper focuses on new approaches for ultra-large tree estimation, including methods for co-estimation of alignments and trees, estimating trees without needing a full sequence alignment, and phylogenetic placement. While the main focus is on methods with empirical performance advantages, we also discuss the theoretical guarantees of methods under Markov models of evolution. Finally, we include a discussion of the future of large-scale phylogenetic analysis.

1 Introduction

Evolution is a unifying principle for biology, as has been noted by Dobzhansky, de Chardin, and others[1]. Phylogenies are mathematical models of evolution, and therefore enable insights into the evolutionary relationships between organisms, genes, and even networks. Indeed, phylogeny estimation is a major part of much biological research, including the inference of protein structure and function, of trait evolution, etc. [3].

Phylogeny estimation from molecular sequences generally operates as follows: first the sequences are aligned through the insertion of spaces between letters (nucleotides or amino-acids) in the sequences, and then a tree is estimated using the alignment. This "two-phase" approach to phylogeny estimation can produce highly accurate estimations

[1] The famous quote by Dobzhansky "Nothing in biology makes sense except in the light of evolution" [1] reflects the less known quote by the Jesuit priest Pierre Teilhard de Chardin [2], who wrote "Evolution is a light which illuminates all facts, a curve that all lines must follow."

of the tree for small to moderate-sized datasets that are fairly closely related. However, datasets that contain sequences that are quite different from each other (especially if the datasets are very large), can be very difficult to align - and even considered “un-alignable”, and trees based on poor alignments can have high error [4–11].

Although the estimation of both alignments and phylogenies is challenging for large, highly divergent datasets, there is substantial evidence that phylogenetic analyses of large datasets may result in more accurate estimations of evolutionary histories, due to improved taxonomic sampling [12–15]. Thus, although not all datasets will be improved through the addition of taxa, some phylogenetic questions - particularly the inference of deep evolutionary events - seem likely to require large datasets.

In this chapter, we discuss the challenges involved in estimating large alignments and phylogenies, and present some of the new approaches for large alignment and tree estimation. Thus, this chapter does not attempt to survey alignment estimation methods or tree estimation methods, each of which is an enormous task and discussed in depth elsewhere; see [16–20] for phylogeny estimation and [11, 21–27] for alignment estimation.

We begin with the basics of alignment and phylogeny estimation in Section 2, including Markov models of sequence evolution and statistical performance criteria; this section also discusses the class of “absolute fast converging methods” and presents one of these methods. Section 3 discusses some methods that co-estimate alignments and trees (rather than operating in two-phases). Section 4 presents methods that estimate trees without needing a full multiple sequence alignment. We close with a discussion about the future of large-scale alignment and phylogenetic tree estimation in Section 5.

2 Two-phase Alignment and Phylogeny Estimation

This section contains the basic material for this book chapter. We begin with a description of a phylogenomic pipeline (estimating the species history from a set of genes), and discuss the issues involved in resolving incongruence between different gene trees. We then discuss general issues for multiple sequence alignment estimation and evaluation, including how alignments can be used for different purposes, and hence evaluation metrics can differ. We describe certain algorithmic techniques for multiple sequence alignment that have been used to enable large-scale analyses, including template-based methods, divide-and-conquer, and progressive alignment, and we discuss some alignment methods that have

Alignment Method	Data	Largest dataset	Publications	Techniques
`MAFFT-PartTree`	all	93,681 [28]	[29]	progressive
`Clustal-Quicktree`	all	27,643 [30]	[31]	progressive
`Kalign-2`	all	50,175 [28]	[32]	progressive
`Clustal-Omega`	amino-acid	93,681 [28]	[28]	progressive HMMs
`Neuwald's method`	amino-acid	400,000 (approx.) [33]	[33]	template

Table 1. Table of methods for large-scale multiple sequence alignment estimation. We show methods that have published results on datasets with at least 25,000 sequences, showing the type of data (DNA, RNA, amino-acid, or all), the largest number of sequences in published dataset analyses, publications for the method, and techniques used. The methods listed in the table for co-estimation of alignments and trees can also be considered as alignment estimation methods, but are not listed here. Finally, the largest dataset size we note here for each method may not be the largest performed, but is the largest we were able to find and document. It is also possible that there are publications we are not aware of that present analyses of datasets of this size using methods not listed here.

been used on very large datasets.

We then turn to tree estimation, beginning with stochastic models of sequence evolution and statistical performance criteria for phylogeny estimation methods. We provide a brief background in the theoretical guarantees of different phylogeny estimation methods (e.g., which methods are statistically consistent under the basic sequence evolution models and the sequence length requirements of methods), as well as some discussion about their empirical performance on large datasets. We discuss gap treatment methods and their performance guarantees, and the empirical impact of these methods on phylogenetic analyses. We then discuss the analysis of datasets that contain short sequences (i.e., fragments of full-length sequences), including phylogenetic placement methods that insert short sequences into pre-computed trees, and how these methods can be used in metagenomic analysis.

Although this chapter provides some discussion about many methods - both for alignment estimation and phylogeny estimation - the focus is on those methods that can analyze large datasets. The main effort, therefore, is to describe just a few methods, and to try to identify the algorithmic techniques that make them able to analyze large datasets.

Phylogeny Method	Data	Largest dataset	Publications	Techniques
FastTree-2	all	1.06 million (approx.)	[34]	maximum likelihood
RAxML	all	55,473 [35]	[36]	maximum likelihood
TNT	all	73,060 [37]	[38]	maximum parsimony
DACTAL (almost alignment-free)	all	27,643 [30]	[30]	iteration divide-and-conquer supertree

Table 2. Table of methods for large-scale phylogeny estimation. We show methods that have published results on datasets with at least 25,000 sequences, showing the type of data (DNA, RNA, amino-acid, or all), the largest number of sequences in published dataset analyses, publications for the method, and techniques used. We do not show results for distance-based methods, although these tend to be able to run (efficiently) on very large datasets. With the exception of DACTAL, these methods require an input alignment. The methods listed in the table for co-estimation of alignments and trees can also be considered phylogeny estimation methods, but are not listed here. Finally, the largest dataset size we note here for each method may not be the largest performed, but is the largest we were able to find and document. It is also possible that there are publications we are not aware of that present analyses of datasets of this size using methods not listed here.

Phylogeny Method	Data	Largest dataset	Publications	Techniques
SATé co-estimates alignments and trees	all	27,643 [39]	[10, 39]	iteration, progressive divide-and-conquer maximum likelihood
Mega-phylogeny co-estimates alignments and trees	all	55,473 [35]	[35]	divide-and-conquer maximum likelihood

Table 3. Table of methods for large-scale co-estimation of phylogenies and alignments. We show methods that have published results on datasets with at least 25,000 sequences, showing the type of data (DNA, RNA, amino-acid, or all), the largest number of sequences in published dataset analyses, publications for the method, and techniques used. Finally, the largest dataset size we note here for each method may not be the largest performed, but is the largest we were able to find and document. It is also possible that there are publications we are not aware of that present analyses of datasets of this size using methods not listed here.

2.1 Standard phylogenomic analysis pipelines

The focus of this paper is on the estimation of alignments and trees for single genes, which normally follows a two-phase process: first the sequences are aligned, and then a tree is estimated on the alignment. However, a description of how a species tree is estimated can help put these methods into a larger context.

Because gene trees can differ from species trees due to biological causes (such as incomplete lineage sorting, gene duplication and loss, and horizontal gene transfer [40]), species tree estimations are based on multiple genes rather than any single gene. At the simplest level, this can involve just a handful of genes, but increasingly "phylogenomic" analyses (involving genes from throughout the genome) are being performed [41–44], followed by biological discoveries based on these phylogenomic analyses [45]. The following approaches are the dominant methods used to estimate species trees from multiple genes:

1. Markers are selected, and homologous regions within the genomes are identified across the species; these homologous regions are sometimes limited to the orthologous parts, so that gene duplication and loss does not need to be considered in estimating the species history.

2. Multiple sequence alignments are estimated on each marker.

3. At this point, the standard analysis pipeline continues in one of the following ways:

   (a) the gene sequence alignments can be concatenated, and a tree estimated on the "super-matrix",

   (b) gene trees can be estimated, and then combined together into a species tree using supertree methods [46–54] or methods that explicitly take biological causes (e.g., incomplete lineage sorting and gene duplication and loss) for gene tree incongruence into account [40, 42, 55–70], or

   (c) the species tree can be estimated directly from the set of sequence alignments, taking biological causes for gene tree incongruence into account (an example is *BEAST, which co-estimates the gene trees and species tree directly from the input set of alignments [71]).

The first approach is distinctly different in flavor from the last two approaches, and is called the "super-matrix" or "combined analysis" approach. The relative merits of these approaches with respect to accuracy are debated, but for statistical reasons, it makes sense to use methods that consider biological causes for incongruence when estimating species trees from multiple markers. The development and understanding of methods for estimating species trees given multiple genes, taking incomplete lineage sorting (ILS) and gene duplication and loss into account, is an active research area; see, for example, papers on this subject in the session on Phylogenomics and Population Genomics at the 2013 Pacific Symposium on Biocomputing [72–75].

2.2 Multiple Sequence Alignment

Introduction. Alignment methods vary in type of data (DNA, RNA, or amino-acid) they can handle, and also, to some extent, the objectives of the alignment method. Thus, some methods are designed exclusively for proteins [33, 76–81], some exclusively for RNAs [82–87], but many alignment methods can analyze both protein and nucleotide datasets. We refer to methods that can analyze all types of molecular sequences as “generic” methods.

Alignment methods are used to predict function and structure, to determine whether a sequence belongs to a particular gene family or superfamily, to recognize homology in the ‘twilight zone’ (where sequence similarity is so low that homology is difficult to detect), to infer selection, etc. [11]. On the other hand, alignment methods are also used in order to estimate a phylogeny. As we shall see, the design of alignment methods and how they are evaluated depend on the purpose they are being used for.

MSA Evaluation Criteria. The standard criteria used to evaluate alignments for accuracy are based on shared homologies between the true and the estimated alignment, with the SP-score [88, 89] (sum-of-pairs score) measuring the fraction of the true pairwise homologies correctly recovered, and the TC-score (“total column” score) measuring the number of identical columns. Variants on these criteria include the true metrics suggested by Blackburne *et al.* [90] and the consideration of different types of alignment error (i.e., both false positive and false negative) rather than one overall measure of “alignment accuracy” [89]. Other criteria, such as the identification and correct alignment of specific regions within a protein or rRNA, have also been used [33, 91]. Furthermore, because sequence alignment has the potential to impact phylogeny estimation, a third way of evaluating a multiple sequence alignment method is via its impact on phylogeny estimation

[11].

Thus, there are at least three different ways of assessing alignment accuracy: the first type uses standard criteria and their variants (e.g., SP, TC, Cline Shift scores, and the methods suggested in [89, 90]) that focus on shared homologies and treat them all identically; the second type reports accuracy with respect to only those sites with functional or structural significance; and the third type focuses on phylogenetic accuracy. These criteria are clearly related, but improved performance with respect to one criterion may not imply improved performance with respect to another! An example of this is given in [10], where some estimated alignments differed substantially in terms of their SP-scores, and yet maximum likelihood trees on these alignments had the same accuracy. Similarly, when the objective is protein structure and function prediction, mistakes in alignments that are not structurally or functionally important may not impact these predictions, and so two alignments could yield the same inferences for protein structure and function and yet have very different scores with respect to standard alignment evaluation criteria. Furthermore, the algorithmic techniques that lead to improved results for one purpose may not lead to improved performance for another, and benchmark datasets used to evaluate methods may also not be identical.

Benchmark datasets. Many studies (see [21, 24, 27, 87, 92, 93] for examples) have evaluated MSA methods using biological data for which structurally informed alignments are available. The best known of these benchmark datasets is probably BAliBASE [94], but others are also used [95–98]. The choice of benchmarks and how they are used has a large impact on the result of the evaluation, and so has been discussed in several papers [24, 78, 99–101].

However, the use of structurally-defined benchmarks has also been criticized [11, 100, 102] as being inappropriate for evaluating alignments whose purpose is phylogenetic estimation. The main criticism is the observation that structural or functional "homology" and "positional homology" (whereby two residues are positionally homologous if and only if they descend from a residue in their common ancestor by substitutions alone [103]) are different concepts, and that molecules with residues that are functionally or structurally homologous due to convergent evolution without being positionally homologous have been found [11, 102, 104]. Thus, alignments that are correct with respect to functional or structural homology may be incorrect with respect to positional homology, and therefore violate the assumptions used in phylogenetic estimation. However, even reputed benchmark alignments have errors, as discussed in [11, 100], making the use of these benchmarks even for detecting structural motifs questionable in some cases.

The use of benchmarks in general, and specifically structural benchmarks, is discussed at length by Iantoro *et al.* [100]. From the perspective of phylogeny estimation, one of the most important of their observations is that structurally-defined benchmarks often omit the highly variable parts of the molecule, including introns. Thus, an alignment can be considered completely correct as a structural alignment if it aligns the conserved regions, even if it fails to correctly align the variable regions. The problem with this criterion (as they point out) is that the highly variable portions are often the sites that are of most use for phylogeny estimation, whereas sites that change slowly have little phylogenetic signal.

An obvious response to the concerns about the potential disagreement between positional homology and structural homology is that while they two concepts are not identical, structural features tend to change slowly and so there is a close relationship between the two concepts [105]. Thus, for many datasets, and perhaps even most, these definitions may be identical, and so the use of structural benchmarks is acceptable (and advisable) for most cases. However, the criticism raise by Iantoro *et al.* regarding the elimination of the highly variable regions in the benchmark is more difficult to counter, except, perhaps, by saying that correct structural alignments of the variable regions are much more difficult to establish.

This is one of the reasons that simulated data are also used to evaluate MSA methods: the true alignment is known with certainty, including the alignment of the hyper-variable regions. Another advantage of simulated datasets is that they enable the exploration of a larger range of conditions, whereas only a few biological datasets are used as alignment benchmarks. Finally, simulated datasets, when simulated on model trees under an evolutionary process, also provide a true tree to which estimated trees can be compared. Thus, when the purpose of the alignment is to estimate the phylogeny, simulations of sequence evolution down model trees present definite advantages over structural benchmarks, and have become the standard technique for evaluating alignments.

Relative performance of MSA methods. While most studies have evaluated alignment methods in terms of standard criteria (notably, SP and TC scores) on biological benchmarks, some studies have explored alignment estimation for phylogeny estimation purposes. As commented on earlier, many studies have shown that alignment estimation impacts phylogenetic estimation, and that alignment and tree error increase with the rate of evolution. Also, on very large datasets, due to computational limitations, only a few alignment methods can even be run (and typically not the most accurate ones), which results in increased alignment error [6]. On the other hand, on large trees with rates of evolution that are sufficiently low, alignment estimation methods can differ substantially in terms of SP-score without impacting the accuracy of the phylogenetic tree estimated using the alignment [10]. More generally, standard alignment metrics may be only poorly correlated with tree accuracy in some conditions.

Not all alignment methods have been tested for their impact on phylogenetic accuracy; however, among those that have been tested, MAFFT [106] (when run in its most accurate settings) is among the best performing methods [4,6,10,39] on both proteins and nucleotides, especially on datasets with many sequences. For small nucleotide datasets, especially those with relatively low rates of evolution, other methods (e.g., Probcons [107] and Prank [108]) can give excellent results [109, 110].

Progressive aligners, and the impact of guide trees. Many alignment methods use progressive alignment on a guide tree to estimate the alignment; thus the choice of the guide tree and its impact on alignment and phylogeny estimation is also of interest [109, 111–113]. Nelesen *et al.* [109] studied the impact of the guide tree on alignment methods, and showed that improved phylogenetic accuracy can be obtained by first estimating a tree from the input using a good two-phase method (RAxML [36] on a MAFFT alignment). They noted particular benefits in using Probcons with this guide tree, and called the resultant method “Probtree”. Prank has also been observed to be very sensitive to guide trees [113], and to give improved results by the use of carefully computed guide trees (maximum likelihood on good alignments) [10, 112]. Another study showed that even when the alignment score doesn’t change, the alignment itself can change in important ways when guide trees are changed [111]. Finally, Capela-Gutierrez and Gabaldon [113] found that the placement of gaps in an alignment results from the choice of the guide tree, and hence the gaps are *not* phylogenetically informative. Based on these observations, Capela-Gutierrez and Gabaldon recommended that alignment estimation methods should use the true tree (if possible), or else use an iterative co-estimation method that infers both the tree and the alignment.

Template-based methods. Some alignment methods use a very different type of algorithmic structure, which is referred to as being “template-based” [24]. Instead of using progressive alignment on a guide tree, these methods use models (either profiles, templates, or Hidden Markov Models) for the gene of interest, and align each sequence to the model in order to produce the final multiple sequence alignment, as follows. First, a relatively small set of sequences from the family is assembled, and an alignment estimated for the set. Then, some kind of model (e.g., a template or a Hidden Markov Model) is constructed from this “seed” alignment. This model can be relatively simple or quite complex, typically depending on whether the model provides structural information. Once the model is estimated, the remaining sequences are added to the growing alignment. The model is used to align each sequence to the seed alignment (which does not change during the process), and then inserted into the growing alignment. Since the remaining sequences are only compared to the seed alignment, homologies between the remaining sequences can only be inferred through their homologies to the seed alignment. Thus, the choice of sequences in the seed alignment and how it is estimated can have a big impact on the resultant alignment accuracy. By design, once the seed alignment and the model are computed, the running time scales linearly with the number of sequences, and the algorithm is trivially parallelizable. Thus, these methods, which we will refer to jointly as “template-based methods”, can scale to very large datasets with hundreds of thousands of sequences.

There are several examples of methods that use this approach [33,85–87,114–116] (see pages 526-529 in [11]). Some of these methods use curated seed alignments based on structure and function of well-characterized proteins or rRNAs; for example, the protein alignment method by Neuwald [33] and the rRNA sequence alignment method by Gardner *et al.* [87] use curated alignments. Constraint-based methods, such as COBALT [117], 3DCoffee [79] and PROMALS [76], similarly use external information like structure and function, but then use progressive alignment techniques (or other such methods) to produce the final alignment. Clustal-Omega also has a version, called “External Profile Alignment”, that uses external information (in the form of alignments) to improve the alignment step.

Finally, PAGAN [116] is another member of this class of methods; however, it has some specific methodological differences to the others. First, unlike several of the others, it does not use external biological information (about structure, function, etc.) to define its seed alignment. Second, while the others tend to use either HMMs, profiles, or templates as a model to define the alignment of the remaining sequences, PAGAN estimates a tree on its seed alignment, and estimates sequences for the internal nodes. These sequences are then used to define the incorporation of the remaining sequences to the seed alignment. This technique is very similar to the technique used in PaPaRa [118], which was developed for the phylogenetic placement problem (see Section 2.4). Thus, PAGAN is one of the "phylogeny-aware" alignment methods, a technique that is atypical of these template-based methods, but shared by progressive aligners. PAGAN was compared to an HMM-based method (using HMMER on the reference alignment to build an HMM, and then using HMMALIGN to align the sequences to the HMM) on several datasets [116]. The comparison showed that PAGAN had very good accuracy, better than HMMALIGN, under low rates of evolution, and that both methods had reduced accuracy under high rates of evolution. They also noted that PAGAN failed to align some sequences under model conditions with high rates of evolution, while HMMER aligned all sequences; however, the sequences that both HMMER and PAGAN aligned were aligned more accurately using PAGAN.

Several studies [21, 33, 76, 80, 81, 87, 119] have shown that alignment methods that use high quality external knowledge can surpass the accuracy of some of the best purely sequence-based alignment methods. However, none of these template-based and constraint-based alignment estimation methods (whether or not based upon external biological knowledge) have been tested for their impact on phylogenetic estimation; instead, they have only been tested with respect to standard alignment criteria (e.g., SP-score), identification of functional or structural residues, or membership in a gene family. Thus, we do not know whether the improvements obtained with respect to traditional alignment accuracy metrics will translate to improvements in phylogeny estimation.

Methods that use divide-and-conquer on the taxon set. Some alignment methods use a divide-and-conquer strategy in which the taxon set is divided into subsets (rather than the sites) in order to estimate the alignment; these include the megaphylogeny method developed by Smith *et al.* [120], SATé [10, 39], SATCHMO-JS [78], PROMALS [76], and the method by Neuwald [33]. (The SATé and SATCHMO-JS methods co-estimate alignments and trees, and so are not strictly speaking just alignment methods.) Neuwald's method is a bit of an outlier in this set, because the user provides the dataset decomposition, but we include it here for comparative purposes.

While the methods differ in some details, they use similar strategies to estimate alignments. Most estimate an initial tree, and then use the tree to divide the dataset into subsets. The method to compute the initial trees differs, with SATCHMO-JS using a neighbor joining [121] (NJ) tree on a MAFFT alignment, SATé using a maximum likelihood tree on a MAFFT alignment, PROMALS using a UPGMA tree on k-mer distances, and mega-phylogeny using a reference tree and estimated alignment. (See the description of mega-phylogeny provided by Roquet *et al.* [122] for more details.)

The subsequent division into subsets is performed in two ways. In the case of mega-phylogeny, SATCHMO-JS, and PROMALS, the division into subsets is performed by breaking the starting tree into clades so as to limit the maximum dissimilarity between pairs of sequences in each set. In contrast, SATé-2 [39] removes centroid edges from the unrooted tree, recursively, until each subset is small enough (below 200 sequences). Thus, the sets produced by the SATé-2 decomposition do not form clades in the tree, unlike the other decompositions. Furthermore, the sets produced by the SATé-2 decomposition are guaranteed to be small (at most 200 taxa) but are not constrained to have low pairwise dissimilarities between sequences.

Alignments are then produced on each subset, with PROMALS, SATé, and mega-phylogeny estimating alignments on each subset, and SATCHMO-JS using the alignment induced on the subset by the initial MAFFT alignment.

These alignments are then merged together into an alignment on the full set, but the methods use different techniques. PROMALS and mega-phylogeny use template-based methods to merge the alignments together, while SATCHMO-JS and SATé use progressive alignment techniques. PROMALS also uses external knowledge about protein structure to guide the template-based merger of the alignments together. PROMALS, SATCHMO-JS, and mega-phylogeny use sophisticated methods to merge subset-alignments, but SATé uses a very simple method (Muscle) to merge subset-alignments.

Neuwald's method [33] shares many features with these four methods, but has some unique features that are worth pointing out. First, like SATCHMO-JS and PROMALS, Neuwald's method can only be used on proteins (mega-phylogeny and SATé can be used on both nucleotides and protein sequences). Neuwald's method requires the user to provide a dataset decomposition and also a manually curated seed alignment reflecting structural and functional features of the protein family. The algorithm operates by estimating alignments on the subsets using simple methods, and then uses the seed alignment to merge the subset-alignments together.

Note that Neuwald's method, PROMALS, and mega-phylogeny are essentially template-based methods, and as such are very scalable once their templates are computed (this first step, however, can be very labor-intensive, if it depends on expert curation). Mega-phylogeny has been used to analyze a dataset with more than 50,000 nucleotide sequences [35], and Neuwald's method has been used to analyze a dataset with more than 400,000 protein sequences. By contrast, because SATCHMO-JS and SATé both rely upon progressive alignment, their running times are longer. Furthermore, SATé uses iteration to obtain improved results (even though the first iteration gives the most improvement), and although most runs finish in just a few iterations, this also adds to the running time. SATé has been used to analyze a dataset with approximately 28,000 nucleotide sequences, but has not been tested on larger datasets.

In terms of performance evaluations, Neuwald's method, SATCHMO-JS, and PROMALS, have been assessed using protein alignment benchmarks, and shown to give excellent results over standard methods. Ortuno *et al.* [21] explored the conditions in which PROMALS gave improvements over the other methods, and showed that the conditions in which the improvements were substantial were when the sequences were close to the 'twilight zone' (i.e., almost random with respect to each other), which is where sequence homology is difficult to detect, and information about structure is the most helpful.

The accuracy of SATé has been assessed using nucleotide alignments, and shown to be very good, both for standard alignment criteria and for phylogenetic accuracy [39]. A recent study [119] evaluated protein alignment methods (including SATé) on large datasets with respect to the TC (total column) score. They found substantial differences in running time between the template-based methods (which had the best speed) and other methods, including SATé, and so only ran the fastest methods on the largest datasets, which had 50,000 protein sequences. To the best of our knowledge, the mega-phylogeny method has not been compared to other methods on benchmark datasets with curated alignments or trees.

Very large-scale alignment. When the datasets are very large, containing many thousands of sequences, only a few alignment estimation methods are able to run. As noted, the template-based methods (including PROMALS and mega-phylogeny) scale linearly with the number of taxa, and so can be used with very large datasets. SATé and SATCHMO-JS are not quite as scalable; however, SATé has been able to analyze nucleotide datasets with about 28,000 sequences. Other methods that have been shown to run on very large datasets include Clustal-Omega [28], MAFFT-PartTree [29], and Kalign-2 [32], but many methods fail to run on datasets with tens of thousands of sequences [6]. Of these, Clustal-Omega is only designed for protein sequences, but MAFFT-PartTree and Kalign-2 can analyze both nucleotide and amino-acid sequences.

SATé is computationally limited by its use of progressive alignment and maximum likelihood method (RAxML or FastTree-2 [34]) in each iteration; both impact the running time and - in the case of large numbers of long sequences - memory usage. However, although limited to datasets with perhaps only 30,000 sequences (or so), on fast-evolving datasets with 1000 or more sequences, SATé provides improvements in phylogenetic accuracy relative to competing methods [6, 39].

2.3 Tree estimation

Most phylogeny estimation methods are designed to be used with sequence alignments, and so presume that the alignment step is already completed. These methods are generally studied with respect to their performance under Markov models of evolution in which sequences evolve only with substitutions. Therefore, we begin with a discussion about site substitution models, and about statistical performance guarantees under these models.

2.3.1 Stochastic models of sequence evolution

We begin with a description of the simplest stochastic models of DNA sequence evolution, and then discuss amino-acid sequence evolution models and codon evolution models. The simplest models of DNA sequence evolution treat the sites within the sequences independently. Thus, a model of DNA sequence evolution must describe the probability distribution of the four states, A, C, T, G, at the root, the evolution of a random site (i.e., position within the DNA sequence) and how the evolution differs across the sites. Typically the probability distribution at the root is uniform (so that all sequences of a fixed length are equally likely). The evolution of a single site is modeled through the use of "stochastic substitution matrices," 4×4 matrices (one for each tree edge) in which every row sums to 1. A stochastic model of how a single site evolves can thus have up to 12 free parameters. The simplest such model is the Jukes-Cantor model, with one free parameter, and the most complex is the General Markov model, with all 12 parameters [123]:

Definition 1 *The General Markov (GM) model of single-site evolution is defined as follows.*

1. *The nucleotide in a random site at the root is drawn from a known distribution, in which each nucleotide has positive probability.*
2. *The probability of each site substitution on an edge e of the tree is given by a 4×4 stochastic substitution matrix $M(e)$ in which $det(M(e))$ is not 0, 1, or -1.*

This model is generally used in a context where all sites evolve identically and independently (the *i.i.d.* assumption), with rates of evolution drawn typically drawn from a gamma distribution. (Note that the distribution of the rates-across-sites has an impact on phylogeny estimation and dating, as discussed by Evans and Warnow [124].) In what follows, we will address the simplest version of the GM model so that all sites have the same rate of evolution.

We denote a model tree in the GM model as a pair, $(T, \{M(e) \colon e \in E(T)\})$, or more simply as (T, M). For each edge $e \in E(T)$, we define the length of the edge $\lambda(e)$ to be $-\log |det(M(e))|$. This allows us to define the matrix of leaf-to-leaf distances, $\{\lambda_{ij}\}$, where $\lambda_{ij} = \sum_{e \in P_{ij}} \lambda(e)$, and where P_{ij} is the path in T between leaves i and j. A matrix defined by path distances in a tree with edge weights is called "additive", and it is a well-known fact that given any additive matrix, it is easy to recover the underlying leaf-labelled tree T for that matrix in polynomial time.

This general model of site evolution subsumes the great majority of other models examined in the phylogenetic literature, including the popular General Time Reversible (GTR) model [125], which requires only that $M(e) = M(e')$ for all edges e and e'. Further constraints on the matrix $M(e)$ produce the Hasegawa-Kishino-Yang (HKY) model, the Kimura 2-parameter model (K2P), the Kimura 3-ST model (K3ST), the Jukes-Cantor model (JC), etc. These models are all special cases of the General Markov model, because they place restrictions on the form of the stochastic substitution matrices. The standard model used for nucleotide phylogeny estimation is GTR+gamma, i.e., the General Time Reversible (GTR) model of site substitution, equipped with a gamma distribution of rates across sites.

Protein models. Just as with DNA sequence evolution models, there are Markov models of evolution for amino-acid sequences, and also for coding DNA sequences. These models are described in the same way - a substitution matrix that governs the tree, and then branch lengths. While the GTR model can be extended to amino-acids (to produce a 20x20 matrix) or to codon-based models (to produce a 64x64 matrix), both of which must be estimated from the data, in practice these models use fixed matrices, each of which was estimated from external biological data. The most well known protein model is the Dayhoff model [126], but improved models have been developed in recent years [127–133]. Similarly, codon-based models have also been based on fixed 64x64 matrices (e.g., [134–136]). In practice, the selection of a protein model for a given dataset is often done using a statistical test, such as ProtTest [137], and then fixed. In the subsequent tree estimation performed under that model, only the tree and its branch lengths need to be estimated.

More general site evolution models. The models that are typically used in phylogenetic estimation tend to be fairly simple, and have come under serious criticism as a result, especially when used with proteins [138, 139]. For example, these models fail to account for GC content variation across the tree, rates of evolution that are not drawn from the gamma distribution (or similarly simple models), or heterotachy (where the substitution matrix depends on the edge and the site [140–143]). Studies of gene family evolution have also shown that the neutral model of evolution is unrealistic [144]. More general models of site evolution have been proposed, including the non-stationary, non-homogeneous model of Galtier and Guoy [145].

2.3.2 Phylogeny Estimation Methods

There are many different phylogeny estimation methods, too numerous to mention here. However, the major ones can be classified into the following types:

- distance-based methods, which first compute a pairwise distance matrix (usually based on a statistical model) and then compute the tree from the matrix [17],

- maximum parsimony and its variants [146], which seek a tree with a total minimum number of changes (as defined by edit distances between sequences at the endpoints on the edges of the tree),

- maximum likelihood [147], which seeks the model tree that optimizes likelihood under the given Markov model, and
- Bayesian MCMC methods, which return a distribution on trees rather than a single tree, and also use likelihood to evaluate a model tree.

2.3.3 Statistical Performance Criteria

We discuss three concepts here: *identifiability*, *statistical consistency*, and *sequence length requirements.*

Identifiability: A statistical model or one of its parameters is said to be "identifiable" if it is uniquely determined by the probability distribution defined by the model. Thus, in the context of phylogeny estimation, the unrooted model tree topology is identifiable if it is determined by the probability distribution (defined by the model tree, which includes the numeric parameters) on the patterns of nucleotides at the leaves of the tree. In the case of nucleotide models, the state at each leaf can be A, C, T, or G, and so there are 4^n possible patterns in a tree with n leaves (similarly, there are 20^n possible patterns for amino-acid models). It is well known that the unrooted tree topology is identifiable under the General Markov model [123], and recent work has extended this to other models [148–150].

Statistical Consistency: We say that a method Φ is "statistically consistent" for estimating the topology of the model tree (T, θ) if the trees estimated by Φ *converges* to the unrooted version of T (denoted by T^u) as the number of sites increases. (Note that under this definition, we are not concerned with estimating the numeric parameters.) Equivalently, for all $\varepsilon > 0$ there is a sequence length K so that if a set S of sequences of length $k \geq K$ are generated by (T, θ), then the probability that $\Phi(S) = T^u$ is at least $1-\varepsilon$. We say that a method is statistically consistent under the GM model if it is statistically consistent for all model trees in the GM model. Similarly, we say a method is statistically consistent under the GTR model if it is statistically consistent under all model trees in the GTR model.

Many phylogenetic methods are statistically consistent under the GM model, and hence also under its submodels (e.g., the GTR model). For example, maximum likelihood, neighbor joining (and other distance-based methods) for properly computed pairwise "distances", and Bayesian MCMC methods, are all statistically consistent [17, 151–153]. On the other hand, maximum parsimony and maximum compatibility are not statistically consistent under the GM model [154]. In addition, it is well known that maximum likelihood can be inconsistent if the generative model is different from the model assumed by maximum likelihood, but maximum likelihood can even be inconsistent when its assumptions match the generative model, if the generative model is too complex! For example, Tuffley and Steel showed that maximum likelihood is equivalent to maximum parsimony under a very general "no-common-mechanism" model [142], and so is inconsistent under this model. In this case, the model itself is not identifiable, and this is why maximum likelihood is not consistent [155–157]. However, there are identifiable models for which ML is not consistent, as observed by Steel [143].

Sequence length requirement: Clearly, statistical consistency under a model is a desirable property. However, statistical consistency does not address how well a method will work on finite data. Here, we address the "sequence length requirement" of a phylogeny estimation method Φ, which is the number of sites that Φ needs to return the (unrooted version of the) true tree with probability at least $1 - \epsilon$ given sequences that evolve down a given model tree (T, θ). Clearly, the number of sites that suffices for accuracy with probability at least $1 - \epsilon$ will depend on Φ and ϵ, but it also depends on both T and θ.

We describe this concept in terms of the Jukes-Cantor model, since this is the simplest of the DNA sequence evolution models, and the ideas are easiest to understand for this model. However, the same concepts can be applied to the more general models, and the theoretical results that have been established regarding sequence length requirements extend to the GM (General Markov) model, which contains the GTR model and all its submodels.

In the Jukes-Cantor (JC) model, all substitutions are equally likely, and all nucleotides have equal probability for the root state. Thus, a Jukes-Cantor model tree is completely defined by the rooted tree T and the branch lengths $\lambda(e)$, where $\lambda(e)$ is the expected number of changes for a random site on the edge e. It is intuitively obvious that as the minimum branch length shrinks, the number of sites that are needed to reconstruct the tree will grow, since a branch on which no changes occur cannot be recovered with high probability (the branch will appear in an estimated tree with probability at most one-third, since at best it can result from a random resolution of a node of degree at least 4).

It is also intuitively obvious that as the maximum branch length increases, the number of sites that are needed will increase, since the two sides of the long branch will seem random with respect to each other. Thus, the sequence length requirement for a given method to be accurate with probability at least $1 - \epsilon$ will be impacted by the shortest branch length f and the longest branch length g. It is also intuitively obvious that the sequence length requirement will depend on the number of taxa in the tree.

Expressing the sequence length requirement for the method Φ as a function of these parameters (f, g, n and ϵ) enables a different - and finer - evaluation of the method's performance guarantees under the statistical model. Hence, we consider f, g, and ϵ as fixed but arbitrary, and we let $JC_{f,g}$ denote all Jukes-Cantor model trees with $0 < f \leq \lambda(e) \leq g < \infty$ for all edges e. This lets us bound the sequence length requirement of a method as a function only of n, the number of leaves in the tree.

The definition of "absolute fast convergence" under the Jukes-Cantor model is formulated as an upper bound on the sequence length requirement, as follows:

Definition 2 *A phylogenetic reconstruction method* Φ *is* absolute fast-converging (afc) *for the Jukes-Cantor (JC) model if, for all positive* f, g, *and* ε, *there is a polynomial* $p(n)$ *such that, for all* (T, θ) *in* $JC_{f,g}$, *on set* S *of* n *sequences of length at least* $p(n)$ *generated on* T, *we have* $Pr[\Phi(S) = T^u] > 1 - \varepsilon$.

Note also that this statement only refers to the estimation of the unrooted tree topology T^u and not the numeric parameters θ. Also, note that the method Φ operates without any knowledge of parameters f or g—or indeed any function of f and g. Thus, although the polynomial p depends upon both f and g, the method itself will not. Finally, this is an upper bound on the sequence length requirement, and the actual sequence length requirement could be much lower.

The function $p(n)$ can be replaced by a function $f(n)$ that is not polynomial to provide an upper bound on the sequence length requirement for methods that are not proven to be absolute fast converging.

In a sequence of papers, Erdős *et al.* [158–160] presented the first absolute fast converging methods for the GM model, and presented techniques for establishing the sequence length requirements of distance-based methods. Following this, the sequence length requirement of neighbor joining (NJ) was studied, and lower bounds and upper bounds that are exponential in n were established [151, 161]. These papers were followed by a number of other studies presenting other afc methods (some with even better theoretical performance than the first afc methods) or evaluating the sequence length requirements of known methods [162–175].

2.3.4 Empirical Performance

So far, these discussions have focused on theoretical guarantees under a model, and have addressed whether a method will converge to the true tree given long enough sequences (i.e., statistical consistency), and if so, then how long the sequences need to be (sequence length requirements). However, these issues are purely theoretical, and do not address how accurate the trees estimated by methods are in practice (i.e., on data). In addition, the computational performance (time and memory usage) of phylogeny estimation methods is also important, since a method that is highly accurate but will use several years of compute time will not generally be useful in most analyses.

Phylogenetic tree accuracy can be computed in various ways, and there are substantive debates on the "right" way to calculate accuracy [176, 177]; however, although disputed, the Robinson-Foulds [178] (RF) distance, also called the "bipartition distance", is the most commonly used metric on phylogenetic trees. We describe this metric here.

Given a phylogenetic tree T on n taxa, each edge can be associated with the bipartition it induces on the leaf set; hence, the tree itself can be identified with the set of leaf-bipartitions defined by the edges in the tree. Therefore, two trees on the same set of taxa can be compared with respect to their bipartition sets. The RF distance between two trees is the size of the symmetric difference of these two sets, i.e., it is the number of bipartitions that are in one tree's dataset but not both. This number can be divided by $2(n - 3)$ (where n is the number of taxa) to obtain the "RF rate." In the context of evaluating phylogeny estimation methods, the RF distance is sometimes divided into false negatives and false positives, where the false negatives (also called "missing branches") are branches in the true tree that are not present in the estimated tree, and the false positives are the branches in the estimated tree that are not present in the true tree. This distinction between false positives and false negatives enables a more detailed comparison between trees that are not binary.

Many studies have evaluated phylogeny estimation methods on simulated data, varying the rate of evolution, the branch lengths, the number of sites, etc. These studies have been enormously informative about the differences between methods, and have helped biologists make informed decisions regarding methods for their phylogenetic analyses. Some of the early simulation studies explored performance on very small trees, including the fairly exhaustive study by Huelsenbeck and Hillis on 4-leaf trees [179], but studies since then have explored larger datasets [4, 10, 180, 181] and more complex questions. For example, studies have explored the impact of taxon sampling on phylogenetic inference [12, 180, 182, 183], the impact of missing data on phylogenetic inference [184–187], and the number of sites needed for accuracy with high probability [188]. In fact, simulation studies have become, perhaps, the main way to explore phylogenetic estimation.

Distance-based methods. Distance-based methods operate by first computing a matrix of distances (typically using a statistically defined technique, to correct for unseen changes) between every pair of sequences, and then construct the tree based on this matrix. Most, but not all, distance-based methods are statistically consistent, and so will be correct with high probability, given long enough sequences. In general, distance-based methods are polynomial time, and so have been popular for large-scale phylogeny estimation. While the best known distance-based method is probably neighbor joining [121], there are many others, and many are faster and/or more accurate [189–194].

One of the interesting properties about distance-based methods is that although they are typically guaranteed to be statistically consistent, not all distance-based methods have good empirical performance! A prime example of this lesson is the Naive Quartet Method, a method that estimates a tree for every set of four leaves using the Four-Point Method (a statistically-consistent distance method) and then returns the tree that is consistent with all the quartets *if it exists* [17]. It is easy to show that the Naive Quartet Method runs in polynomial time and is statistically consistent under the General Markov model; however, because it requires that every quartet be accurately estimated, it has terrible empirical performance! Thus, while statistical consistency is desirable, in many cases statistically inconsistent methods can outperform consistent ones [195, 196].

Maximum parsimony. Maximum parsimony (MP) is NP-hard [146], and so the methods for MP use heuristics (most without any performance guarantees). The most efficient and accurate maximum parsimony software for very large datasets is probably TNT [38], but PAUP* [197] is also popular and effective on datasets that are not extremely large. TNT is a particularly effective parsimony heuristic for large trees [54], and has been able to analyze a multi-marker sequence dataset with more than 73,000 sequences [37].

Maximum likelihood. Maximum likelihood (ML) is also NP-hard [198], and so attempts to solve ML are also made using heuristics. While the heuristics for MP used to be computationally more efficient than the heuristics for ML, the current set of methods for ML are quite effective at "solving" large datasets. (Here the quotes indicate that there is no guarantee, but reasonably good results do seem to be obtained using the current best software.)

The leading methods for large-scale ML estimation under the GTR+Gamma model include RAxML [36], FastTree-2 [34], PhyML [199], and GARLI [200]. Of these four methods, RAxML is clearly the most frequently used ML method, in part because of its excellent parallel implementations. However, a recent study [201] showed that trees estimated by FastTree-2 were almost as accurate as those estimated by RAxML, and that FastTree-2 finished in a fraction of the time; for example, FastTree-2 was able to analyze an alignment with almost 28,000 rRNA sequences in about 5 hours, but RAxML took much longer. Furthermore, FastTree-2 has been used to analyze larger datasets (ones with more sequences) than RAxML: the largest dataset published with a RAxML analysis had 55,000 nucleotide sequences [35], but FastTree has analyzed larger datasets. For example, FastTree-2 has analyzed a dataset with more than 1 million nucleotide sequences [202], and another with 330,556 sequences [203]. The reported running time for these analyses are 203 hours for the million-taxon dataset, and 13 hours (with 4 threads) for the 330K-taxon dataset[2]. By comparison, the RAxML analysis of 55,000 nucleotide sequences took between 100,000 and 300,000 CPU hours[3]. The difference in running time is substantial, but we should note two things: the RAxML analysis was a multi-marker analysis, and so the sequences were much longer (which impacts running time), and because RAxML is highly parallelized, the impact of the increased running time is not as significant (if one has enough processors). Nevertheless, for maximum likelihood analysis of alignments with large numbers of sequences, FastTree-2 provides distinct speed advantages over RAxML.

There are a few important limitations for FastTree-2, compared to RAxML. First, FastTree-2 obtains its speed by somewhat reducing the accuracy of the search; thus, the trees returned by FastTree-2 may not produce maximum likelihood scores that are

[2] Morgan Price, personal communication, May 1, 2013.

[3] Alexis Stamatakis, personal communication, May 1, 2013.

quite as good as those produced by RAxML. Second, FastTree-2 doesn't handle very long alignments with hundreds of thousands of sites very well, while RAxML has a new implementation that is designed specifically for long alignments. Third, FastTree-2 has a smaller set of models for amino-acid analyses than RAxML. Therefore, in some cases (e.g., for wide alignments, and perhaps for amino-acid alignments), RAxML may be the preferred method.

However, the ML methods discussed above estimate trees under the GTR+Gamma model, which has simplifying assumptions that are known to be violated in biological data. The nhPhyml [204] method is a maximum likelihood method for estimating trees under the non-stationary, non-homogeneous model of Galtier and Guoy [145], and hence provides an analytical advantage in that it can be robust to some violations of the GTR+Gamma model assumptions. However, nhPhyml seems to be able to give reliably good analyses only on relatively small datasets (i.e., with at most a few hundred sequences, or fewer sequences if they are very long). The explanation is computational - it uses NNI (nearest neighbor interchanges, see below) to search treespace, but NNI is relatively ineffective [205, 206], which means that it is likely to get stuck in local optima. This is unfortunate, since large datasets spanning substantial evolutionary distances are most likely to exhibit an increased incidence in model violations. Therefore, highly accurate phylogeny estimation of large datasets may require the use of new methods that are based upon more realistic, and more general, models of sequence evolution.

Bayesian MCMC methods. Bayesian methods are similar to maximum likelihood methods in that the likelihood of a model tree with respect to the input sequence alignment is computed during the analysis; the main difference is that maximum likelihood selects the model tree (both topology and numeric parameters) that optimizes the likelihood, while a Bayesian method outputs a distribution on trees. However, once the distribution is computed, it can be used to compute a single point estimate of the true tree using various techniques (e.g., a consensus tree can be computed, or the model tree topology with the maximum total probability can be returned).

The standard technique used to estimate this distribution is a random walk through the model tree space, and the distribution is produced after the walk has converged to the stationary distribution.

There are many different Bayesian methods (e.g., MrBayes [207], BEAST [208], PhyloBayes [209], p4 [210], and BayesPhylogenies [211]), differing in terms of the techniques used to perform the random walk, and the model under which the likelihood is computed; however, MrBayes [212] is the most popular of the methods. Bayesian methods provide theoretical advantages compared to maximum likelihood methods [213–216]. However, the proper use of a Bayesian MCMC method requires that it run to convergence, and this can take a very long time on large datasets [61]. Thus, from a purely empirical standpoint, Bayesian methods do not yet have the scalability of the best maximum likelihood methods, and they are generally not used on very large datasets.

Comparisons between methods. Simulation studies have shown some interesting differences between methods. For example, the comparison between neighbor joining and maximum parsimony reveals that the relative performance may depend on the number of taxa and the rate of evolution, with maximum parsimony sometimes performing better on large trees with high rates of evolution [195], even though the reverse generally holds for smaller trees [179].

More generally, most simulation studies have shown that maximum likelihood and Bayesian methods (when they can be run properly) outperform maximum parsimony and distance-based methods in many biologically realistic conditions (see Wang *et al.* [4] for one such study).

Heuristics for exploring treespace. Since both maximum likelihood and maximum parsimony are NP-hard, methods for "solving" these problems use heuristics to explore the space of different tree topologies. These heuristics differ by the techniques they use to score a candidate tree (with the best ones typically using information from previous trees that have already been scored), and how they move within treespace.

The standard techniques for exploring treespace (i.e., for changing the unrooted topology of the current tree) use either NNI (nearest-neighbor-interchanges), SPR (subtree prune-and-regraft) or TBR (tree-bisection-and-reconnection) moves. All these moves modify unrooted trees, as follows. In an NNI move, an edge in the tree is identified, and two subtrees (one on each side of the edge) are swapped. In an SPR move, a rooted subtree of the tree is deleted from the tree, and then reattached. In a TBR move, an edge in the tree is deleted, thus creating two separate (unrooted) trees, and then the two trees are attached through the addition of an edge. Another type of move, called p-ECR (p-edge-contract-and-refine) [217, 218], has also been suggested. In this move, p different edges are contracted, thus creating one or more high degree nodes; the resultant unresolved tree is then either randomly refined, or refined optimally with respect to the criterion of interest (see, for example, [218, 219] for results regarding maximum parsimony). By definition, an NNI move is an SPR move, and an SPR move is a TBR move; thus, the TBR move is the most general of these three moves. However, p-ECR moves generate different neighborhoods than these moves, although an NNI move is a 1-ECR move. Software that only use NNI moves (e.g., nhPhyml [204]) have the advantage of being faster, since they will reach local optima more quickly; however, they also have a tendency to get stuck in local optima more frequently. The TNT software for maximum parsimony uses more complicated techniques, including sectorial-search, for exploring treespace [38]. Theoretical evaluations of these techniques for exploring treespace have been made [205, 206, 217, 218] that help explain the trade-offs between search strategies.

Because local optima are a problem for heuristic searches for NP-hard problems, randomization is often used to move out of local optima. An example of a technique that uses randomness effectively is the parsimony ratchet [220], which was also implemented for maximum likelihood [221]. In the parsimony ratchet, the search alternates between heuristic searches based on the original alignment, and searches based on stochastically modified versions of the alignment; thus, the tree found during the search for the stochastically modified alignment is used to initiate a search based on the original alignment, etc. This technique thus uses randomness to modify the alignment, rather than to move to a random point in treespace; thus, randomness is a general technique that can be used to improve heuristic searches.

2.3.5 DCM_{NJ}: a fast converging method with good empirical performance

In Section 2.3.3, we discussed absolute fast converging methods, which are methods that provably reconstruct the true tree with high probability from sequences of lengths that grow only polynomially in the number of taxa. As stated, this is a mathematical property rather than an empirical property. Here we describe one of the early absolute fast converging methods, called DCM-neighbor joining (DCM_{NJ}) [222, 223].

The input to DCM_{NJ} is a distance matrix $[D_{ij}]$ for a set of n species, where the distance matrix is defined appropriately for the model (e.g., the use of the logdet [123] distances for the GTR model). DCM-neighbor joining has two phases. In the first phase, it computes a set X of $O(n^2)$ trees (one for each entry in the distance matrix), and in the second phase, it selects a tree from the set X based on a "true tree selection criterion". To obtain a theoretical guarantee of absolute fast convergence, both phases must have some statistical guarantees, which we now describe in the context of the Jukes-Cantor model.

Phase 1 Property: Given JC model tree (T, θ) with branch lengths satisfying $0 < f \leq \lambda(e) \leq g < \infty$ and given $\epsilon > 0$, there is a polynomial $p(n)$ (which can depend on f, g and ϵ) so that given sequences of length at most $p(n)$, then the set X of trees produced in Phase 1 will contains the unrooted true tree T^u with probability at least $1 - \epsilon$.

Phase 2 Property: Let C be the criterion used for Phase 2. Then the desired property for Phase 2 is defined as follows. Given JC model tree (T, θ) with branch lengths satisfying $0 < f \leq \lambda(e) \leq g < \infty$ and given $\epsilon > 0$, there is a polynomial $p(n)$ (which can depend on f, g, and ϵ) so that given sequences of length at most $p(n)$, and given a set X of trees on taxon set S that contains the T^u, then $T^{opt} = T^u$ with probability at least $1 - \epsilon$, where T^{opt} is the tree in X that optimizes criterion C.

We now describe these two phases.

Phase 1 of DCM_{NJ}: For each entry q in the distance matrix $[D_{ij}]$, DCM_{NJ} computes a tree T_q, as follows. First, a "threshold graph" is computed based on $[D_{ij}]$ and the threshold q, so that there is a vertex for every taxon, and an edge between two vertices v_i and v_j if and only if $D_{ij} \leq q$. If the distance matrix is additive (meaning that it equals the path distance in some edge-weighted tree [17]), the threshold graph will be triangulated (also called "chordal"), which means that either the graph is acyclic, or that every induced simple cycle in the graph is of size 3. Chordal graphs have special properties, including that the set of maximal cliques can be enumerated in polynomial time; thus, we can compute the set of (at most) n maximal cliques in the threshold graph [224]. Otherwise, we add edges to the threshold graph (minimizing the maximum "weight" of any added edge) to create a triangulated graph, and then continue. Each maximal clique thus defines a subset of the input sequence set, in the obvious way.

We use the "base method" (here, neighbor joining) to construct a tree on each of these subsets, and we combine these subset trees into a tree on the entire dataset using a particular supertree method called the Strict Consensus Merger [225] (the first phase of SuperFine [53]). For threshold values q that are very small, the set of neighbor joining trees

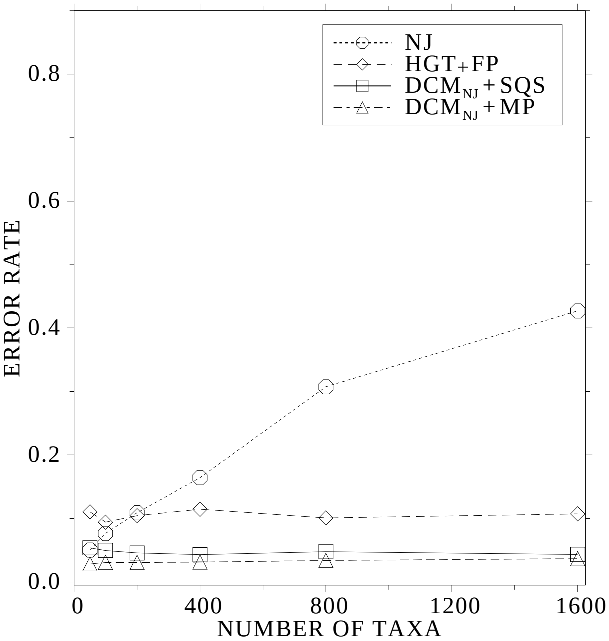

Figure 1. The performance of $DCM1_{NJ}$ with different techniques used in Phase II, compared to NJ and to another absolute fast converging method, HGT+FP [162], as a function of the number of taxa. In this experiment we simulated evolution of sequences with 1000 sites down K2P model trees with topologies drawn from the uniform distribution and with branch lengths drawn from a fixed range. K2P distances were used as inputs to each method. (This figure appeared in Nakhleh *et al.* [223].)

will be insufficient to define the full tree (because of failure to overlap sufficiently). For very large threshold values, there will be sufficient overlap, but the neighbor joining trees on the subsets may have errors. However, for intermediate threshold values, given polynomial length sequences, the neighbor joining subtrees will be correct with high probability and sufficient to define the full tree [222]. Under these conditions, the Strict Consensus Merger will produce the true tree (Lemma 6.2 in [222]). Hence, from polynomial length sequences, with high probability, the first phase will produce the true tree as *one of* the trees in the set of trees it produces (one for each threshold value).

Phase 2: Each tree in the set of trees produced in Phase I is scored using the desired "True Tree Selection" (TTS) criterion, and the tree with the best score is returned. Examples of criteria that have been considered are the maximum likelihood (ML) score, the maximum parsimony (MP) score, and the "short quartet support" (SQS) score. Of these, the MP and SQS scores can be computed exactly and in polynomial time, but the ML score can only be estimated heuristically [226]. Of these criteria, the SQS score is guaranteed to satisfy the required property (described above), but the use of the ML and MP scores gives somewhat more accurate trees.

To summarize, DCM-NJ uses a two-phase process, in which the first phase produces a set of trees, and the second phase selects a tree from the set. Furthermore, each tree computed in the first phase is obtained by using a graph-theoretic technique to decompose the dataset into small overlapping subsets, neighbor joining is used to construct trees on each subset, and then these subset trees are combined together using a supertree method. The result is a method that reconstructs the true tree with high probability from polynomial length sequences, even though the base method (NJ) has a sequence length requirement that is exponential [161]. Thus, DCM-NJ is a technique that estimates a tree on the full set of taxa and that "boosts" the performance of NJ. A similar but simpler method [164] was designed to boost another distance-based method called the "Buneman Tree" (named after Peter Buneman) to produce the DCM-Buneman method, also proven to be afc.

Figure 1 evaluates two variants of DCM-NJ, differing by the "true tree selection" criterion in the second phase, and compares them to neighbor joining (NJ) and to HGT+FP, another absolute fast converging method [162]. $DCM_{NJ}+SQS$ uses SQS for the true tree selection criterion, while $DCM_{NJ}+MP$ uses maximum parsimony. By design, SQS has the desired theoretical property, but MP does not. Instead, MP is used for its empirical performance, as the figure shows.

These methods are evaluated on simulated 1000-site datasets generated down K2P model trees, each with the same branch length distribution, but with varying numbers of taxa. Thus, as the number of taxa increases, the overall amount of evolution increases, and the dataset becomes more difficult to analyze (especially since the sequence length remains fixed at 1000 sites). As shown in Figure 1, the error rate of neighbor joining (using corrected distances to reflect the model of evolution) begins low but increases, so that at 1600 taxa, it is above 40%. By contrast, $DCM_{NJ} + SQS, DCM_{NJ} + MP$ and $HGT + FP$ all have fairly low error rates throughout the range of datasets we tested. Note also that $DCM_{NJ} + MP$ is slightly more accurate than $DCM_{NJ} + SQS$, even though it has no guarantees. Finally, $DCM_{NJ} + SQS, DCM_{NJ} + MP$ and $HGT + FP$ seem to have error rates that do not increase with the numbers of taxa; this is obviously impossible, and so for large enough numbers of taxa, the error rates will eventually increase, and this trend cannot continue indefinitely.

The development of methods with good sequence length requirements is an area of active research, and newer methods with even better theoretical performance have been developed. Because afc methods are designed to extract phylogenetic signal from small numbers of sites, this means that performance on very large datasets (with many taxa) might be improved using these methods, even without needing to use huge numbers of sites. These methods are not used in practice (and DCM_{NJ} is not publicly distributed), and so these methods are mostly of theoretical interest rather than practical.

Clearly there is the potential for these methods to give highly accurate trees for very large datasets; however, the methods and theorems described here assume that the sequences evolve without any indels, and under the General Markov model. While the results could be extended to other identifiable models (including ones with indels) for which statistically consistent distant estimation techniques are available, they would still require that the true alignment be known. Thus, none of this theory applies to more realistic conditions - sequences that evolve with indels, for which the true alignment is *not* known.

2.3.6 Gap treatment in phylogeny estimation methods.

Until now, the entire discussion about phylogeny estimation methods and their guarantees under Markov models of evolution has ignored the fact that sequence alignments often have gaps and that indels are part of sequence evolution. Instead, the Markov models we have discussed are entirely indel-free, and the methods were described as though the sequences were also indel-free. Obviously, since phylogeny estimation methods have been applied to real data, this means that modifications to the data or to the methods have been made to enable them to be used with sequence alignments that have gaps. The purpose of this section is to describe these modifications, and present some discussion about the pros and cons of each modification.

Given a sequence alignment containing gaps, the following approaches are the main ones used in practice for estimating phylogenies:

1. Remove all sites in which any indel appears;

2. Assign an additional state for each dash (thus, for nucleotides, this would result in a 5-state model);

3. Code all the gaps (contiguous segments of dashes) in the alignment, and treat the presence or absence of a gap as a binary character (complementing the original sequence alignment character data); and

4. Treat the gaps as missing data. In parsimony analyses, this is often treated by finding the best nucleotide to replace the gap, but in likelihood-based analyses, this is often treated by summing the likelihood over all possible nucleotides for each gap.

Note that the first three approaches specifically modify the data, and that with the exception of the first approach, all techniques change the input in such a way that the method used to estimate a tree on the alignment must also be changed. Thus, for approach #2, the method must be able to handle 5-state data (for DNA) or 21-state data (for proteins). For approach #3, the method has to be able to handle binary data. In the case of parsimony or likelihood, the challenge is whether changes from presence to absence are treated the same as from absence to presence, and also whether the Markov assumption still makes sense. There are arguments in favor and against each of these gap treatments, especially with respect to statistical consistency under a stochastic model that includes indels as well as substitutions [227].

The first approach of removing all sites with gaps has the advantage of being statistically consistent for stochastic models with indel events in which the substitution process and the mechanism producing insertions and deletions are independent. However, it removes data, and in practice, especially on datasets with many taxa, it could result in phylogenetically uninformative sequence alignments. (A less extreme version of removing all sites with gaps is called “masking”, whereby only some of the sites with gaps are removed. The benefits of using masking are debated, but some recent studies suggest that masking may not be desirable [228].)

The second and third approaches do not reduce the amount of data (which is good) and there are many different gap-coding techniques [229–231]. Simulation studies evaluating some of these methods have shown improvements in some cases for tree estimation obtained through gap-coding over treating gaps as missing data [232–234], but others have found differently [228, 235].

However, the use of gap-coding is controversial [232], in part because of the very substantive challenges in creating a statistically appropriate treatment (consider the meaning of positional homology [103]). Instead, the most frequently used option, and the default for most software, is to treat gaps as missing data. The simulation studies presented later in this paper are all based on analyses of data, treating gaps as missing data.

2.3.7 Theoretical guarantees for standard phylogeny estimation methods on alignments with gaps

Are any of the phylogeny estimation methods we have discussed guaranteed to be statistically consistent when treating gaps as missing data? This is one of the interesting open questions in phylogenetics, for which we give a partial answer.

To address this problem, we defined “monotypic” alignments to be ones in which each site has only one type of nucleotide (all As, all Cs, all Ts, or all Gs) and we proved the following [236]:

Theorem: When the true alignment is monotypic and gaps are treated as missing data, then all trees are optimal for the true alignment under Jukes-Cantor maximum likelihood. Therefore, if the model tree allows indels but not substitutions, then all trees are optimal for Jukes-Cantor maximum likelihood, when gaps are treated as missing data.

At first glance, this theorem might seem to be the result of monotypic alignments not having phylogenetic signal, but this is not the case! In fact, monotypic alignments have sufficient signal to enable accurate trees [237, 238]. Thus, there is phylogenetic signal in an alignment that contains gaps even for the case of monotypic alignments, and this signal can be used to estimate the true tree, provided that appropriate methods are used. In other words, the indels within an alignment can be phylogenetically informative, and indels can even be sufficient to define the tree topology. However, gap treatments can result in loss of information, or lead to erroneous trees (as in the case of ML, treating gaps as missing data, when handling monotypic alignments).

Note that this result does not imply that ML, treating gaps as missing data, is inconsistent under models with positive probabilities of substitutions, and it seems very likely that for most biologically realistic conditions, treating gaps as missing data will not lead to meaningless results. Furthermore, the simulations we and others have performed suggest that ML methods, treating gaps as missing data, do produce reasonably accurate trees. Even so, the potential for reductions in accuracy due to inappropriate handling of gaps is clearly present, and it raises the real possibility that the methods that are known to be statistically consistent under standard substitution-only models, such as GTR, may not be statistically consistent (even on the true alignment!) when sequences evolve with both substitutions and indels.

2.4 Handling Fragmentary Data: Phylogenetic Placement

Multiple sequence alignment methods are generally studied in the context of full-length sequences, and little is known about how well methods work when some of the sequences are very fragmentary. Furthermore, phylogenetic estimation in the context of fragmentary sequences is unreliable, even if the alignments of the fragmentary sequences are accurate [184, 239].

One approach to handling fragmentary sequences is phylogenetic placement: in the first step, an alignment and tree is estimated for the full length sequences for the same gene (these are called the “backbone alignment” and “backbone tree” [240]); in the second step, the fragmentary sequences are added into the backbone alignment to create an “extended alignment”; and finally in the third step, the fragments are then placed in the tree using the extended alignment. This is called the “phylogenetic placement problem”. The first methods for this problem were pplacer [241] and Evolutionary Placement Algorithm (EPA) [242]; both use HMMALIGN [243, 244] to insert the fragments into the alignment of the full-length sequences and then place the fragments into the tree using maximum likelihood for this extended alignment. The initial studies showed that EPA and pplacer exhibited little difference in accuracy or computational requirements [242]. An alternative method, PaPaRa [118], uses a very different technique to align the sequences to the backbone alignment: it infers ancestral state vectors in the phylogeny, and uses these ancestral state vectors to align the fragmentary sequences to the backbone alignment. PaPaRa can give improved accuracy over HMMER when the rate of evolution is slow enough, but otherwise HMMER gives more accurate results [240].

New methods showing improvements over EPA and pplacer have also been developed [240, 245]. Brown and Truskowski [245] use hashing to speed up the method, while SEPP [240] uses a divide-and-conquer technique to speed up the method and improve the accuracy.

Phylogenetic placement can be used in metagenomic analyses [246, 247], in order to estimate the taxonomic identity (what species it is, what genus, etc.) of the short reads produced in shotgun sequencing of a metagenomic sample. When all the reads are drawn from the same gene, then phylogenetic placement can be used to identify the species for the read as described above: first, full-length sequences for the gene are obtained, then an alignment is estimated for the full-length sequences, and finally the reads are inserted into a taxonomy for the species, using the estimated alignment and the phylogenetic placement method. However, since the reads are not drawn from the same gene, then the metagenomic sample must first be processed so that the reads are assigned to genes (or else left unassigned), and then each “bin” of reads for a given gene can be analyzed as described. Thus, phylogenetic placement can be used in a pipeline as a taxon identification method, but the process is substantially more complicated.

3 Co-estimation Methods

Co-estimation of trees and alignments is an obvious approach for several reasons. First, an alignment, like a tree, is a hypothesis about the evolutionary history of the given data. To separate the two hypotheses prevents them from being mutually informative. Thus, rather than first estimating the alignment, treating it as fixed, and then estimating the tree, a co-estimation procedure would try to find the tree and alignment at the same time. The challenge, of course, is how to do this.

In this section, we begin with a discussion of a co-estimation approach that seeks the tree with the minimum total edit distance, but where indels also count towards the total cost. We then continue with a discussion of co-estimation methods that use likelihood calculations on trees, under stochastic models that include indels as well as substitutions. Finally, we discuss SATCHMO-JS, SATé, and mega-phylogeny, methods that return an alignment and a tree from unaligned sequences; these methods were discussed earlier in the section on sequence alignment methods, and here we discuss them in the context of phylogeny estimation.

3.1 Treelength, or "Direct Optimization"

Probably the most commonly used approach to estimating the tree at the same time as estimating the alignment is the "Treelength" approach, also called "Direct Optimization" [248]. This is a natural extension of maximum parsimony to allow it to handle sequences that evolve with indels and so have different lengths.

In order to understand the approach, we begin by noting that a pairwise alignment between two sequences defines one or more edit transformations, each consisting of operations - some substitutions and some indels - that transform the first sequence into the second. Each indel event may be of a single letter (either a nucleotide or an amino-acid), or could be of a string of letters. The "cost" of the pairwise alignment is then the minimum cost of any transformation that is consistent with the alignment. Note that implicit in this definition is the limitation of the operations to just substitutions and indels; therefore, no more complicated operations (such as tandem repeats, inversions, etc.) are considered.

Similarly, each edit transformation that is based on indels and substitutions defines a pairwise alignment. In fact, pairwise alignments are typically computed using dynamic programming algorithms that explicitly compute the minimum cost edit transformation. Thus, there is a close relationship between edit transformations based on indels and substitutions and pairwise alignments.

We now define the treelength problem, where the tree is fixed, and each leaf is labelled by a sequence. The "length" of the tree would be computed by producing sequences at the internal nodes, and then calculating the edit distance between sequences on each edge, using the best possible labelling of the internal nodes (so as to minimize the output length). Since pairwise distances can be computed in these conditions, the length of a tree can be defined and computed, once the sequences at the internal nodes are provided. The treelength problem is then to find the best sequences for the internal nodes so that the total length is minimized.

Given this, we formalize the **Tree Alignment** problem as follows:

Definition 3 *Given a rooted tree T on n leaves which is leaf-labelled by a set $S = \{s_1, s_2, \ldots, s_n\}$ of sequences over Σ (for any fixed alphabet Σ) and an edit cost function $c(.,.)$ for comparing any two sequences over Σ, find sequences to label the internal nodes of T so as to minimize cost$(T) = \sum_{(v,w)\in E(T)} c(l_v, l_w)$, where l_x is the sequence labelling node x in T.*

Note that given sequences at the internal nodes, then for each edge e there is a pairwise alignment of the sequences l_v and l_w labelling the endpoints of e whose cost is identical to $c(l_v, l_w)$. By taking the *transitive closure* of these pairwise alignments we obtain a multiple sequence alignment of the entire dataset whose total cost is *cost*(T). Thus, the output of the Tree Alignment problem can be either considered to be the sequences at the internal nodes, or also the MSA that it defines on the sequences at the leaves of the tree.

The Tree Alignment problem has a rich literature, beginning with [249–251]. The Tree Alignment problem is NP-hard, even for simple gap penalty functions [252], but solutions with guaranteed approximation ratios can be obtained [253–255]. This contrasts with the maximum parsimony problem, which is polynomial time when the tree is fixed (i.e., the optimal sequences for the internal nodes of a given tree can be found in polynomial time using dynamic programming). Thus the treelength problem is *harder* than the maximum parsimony problem.

A generalization of this problem, named after David Sankoff due to his contributions [250, 251], is as follows:

Definition 1 The Generalized Sankoff Problem (GSP) *[From Liu and Warnow [256]]: The input is a set S of unaligned sequences and a function $c(x, y)$ for the edit cost between two sequences x and y. The output is a tree $T = (V, E)$ with leaves labelled by S and internal nodes labeled with additional sequences such that the treelength $\sum_{(v,w)\in E} c(l_v, l_w)$ is minimized, where l_x is the sequence labelling vertex x.*

Not surprisingly, GSP is NP-hard, since the case in which the edit distance function forbids gaps (by setting the cost for a gap to be infinite) is the NP-hard Maximum Parsimony (MP) problem [146].

3.1.1 POY

The standard method for "solving" the GSP problem is POY [248, 257]. Note that in the literature discussing POY, the treelength problem is called "Direct Optimization" (or "DO" for short). POY handles only certain types of edit distances, and in particular, it only enables affine gap penalties. Thus, the cost of a gap of length L is given by $cost(L) = c_0 + c_1 L$, where c_0 is the gap open cost and c_1 is the gap extend cost. When $c_0 = 0$ the gap cost is said to be "simple" (a special case of affine) and when $c_0 > 0$ the gap cost is said to be "affine". POY also enables different costs for transitions and transversions. Thus, the input to POY is a set of unaligned sequences, values for c_0 and c_1, and the cost of transitions and transversions.

The use of treelength optimization to find good trees (and/or alignments) is a matter of substantial controversy in phylogenetics [98,102,227,248,258–261]. Most of the studies that have examined the accuracy of POY trees and alignments explored performance under simple gap penalties, and found that POY did not produce trees and alignments of comparable accuracy to maximum parsimony on the ClustalW alignment, denoted MP(Clustal) [259,262]. A later study examined how the gap penalty affected the accuracy of trees and/or alignments computed by POY [263], and evaluated POY under affine gap penalties (where the gap open cost is non-zero). They found a particular affine gap penalty, which they called "Affine", for which POY produced very good results, and in fact was competitive with MP(Clustal). This study seemed to suggest that POY, and hence the treelength optimization approach to estimating trees and alignments, could be used to find highly accurate trees provided that the right edit distances were used. However, it was also observed that POY was not always effective at finding the best solutions to the treelength problem, and hence the accuracy of POY's trees might not indicate any value in optimizing treelength.

3.1.2 BeeTLe: Better TreeLength

A subsequent study [256] revisited this question, by focusing on whether finding good solutions to treelength criteria would yield improved alignments and trees. In order to understand the impact of the treelength criterion, they developed a new technique for treelength optimization, called "BeeTLe" (Better TreeLength), which is guaranteed to find solutions that are at least as good as those found by POY. BeeTLe runs a collection of methods, including POY, to produce a set of trees on a given input set of unaligned sequences, uses POY to compute the treelength of each tree, and then returns the tree that had the shortest treelength. Thus, BeeTLe is guaranteed to find trees at least as short as those found using POY, and thus enables us to evaluate the impact of using treelength to find trees.

Here we present some results from Liu and Warnow [256], in which BeeTLe was compared to various two-phase methods on simulated 100-taxon datasets, in which sequences evolve with substitutions and indels. We show results for alignments estimated using BeeTLe, MAFFT and ClustalW, and MP and ML trees on the MAFFT, ClustalW, and true alignments. We present alignment error rates (SP-FN, the fraction of true homologies missing from the estimated alignment) and tree topology error rates (the missing branch rate, which is the fraction of edges in the true tree that are not in the estimated tree). We study BeeTLe under three different treelength criteria, each of which has unit cost for substitutions: "Simple-1", which sets the cost of every indel to 1; "Simple-2" (the treelength criterion studied by Ogden and Rosenberg [259] that they found to produce more accurate trees than any other treelength criterion they considered), which assigns cost 2 to indels and transversions and cost 1 to transitions; and "Affine" (the treelength criterion studied in Liu *et al.* [263] that produced more accurate trees than Simple-1 or Simple-2), which sets the cost of a gap of length L to $4 + L$.

In Figure 2, we see that BeeTLe-Affine (BeeTLe using this affine gap penalty) produces the most accurate trees of all BeeTLe variants. We also see that BeeTLe-Affine improves on MP on ClustalW alignments, and matches MP on MAFFT alignments. It also is fairly close to MP on true alignments, except for the hardest 100-taxon model conditions. In Figure 3, we see a comparison of BeeTLe to ML trees computed on ClustalW, MAFFT, and the true alignment. Note how BeeTLe-Affine often produces more accurate trees than ML(ClustalW), but (with the exception of the very easiest model conditions, where there is very little difference between methods), also produces substantially less accurate trees than ML(MAFFT) and ML(TrueAln).

An evaluation of the alignment error on the same datasets (Figure 4) shows that BeeTLe alignments generally have very high alignment SP-FN error, and that BeeTLe-Affine

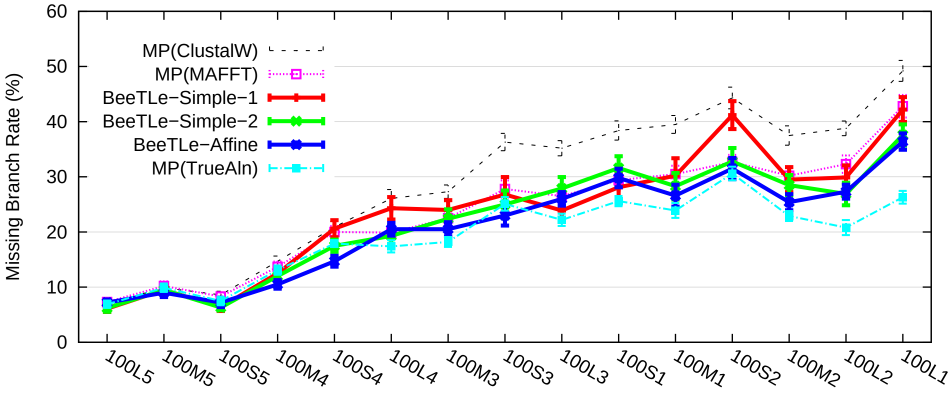

Figure 2. Comparing BeeTLe to MP-based analyses. We report missing branch rates on 100-taxon model conditions for BeeTLe (under three gap penalty treatments) in comparison to maximum parsimony on the ClustalW, MAFFT, and true alignment (TrueAln). Averages and standard error bars are shown; $n = 20$ for each reported value. (This figure appeared in Liu and Warnow [256].)

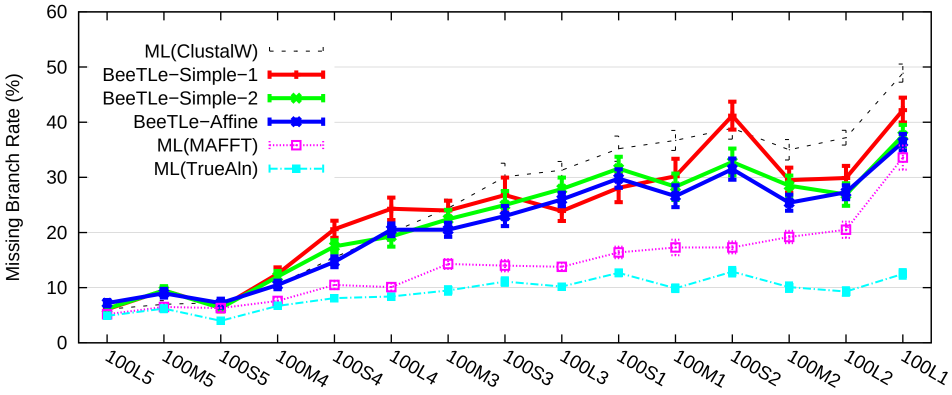

Figure 3. Comparing BeeTLe to ML-based analyses. We report missing branch rates on 100-taxon model conditions for BeeTLe (under three gap penalty treatments) in comparison to maximum likelihood on ClustalW, MAFFT, and the true alignment (TrueAln). Averages and standard error bars are shown; $n = 20$ for each reported value. (This figure appeared in Liu and Warnow [256].)

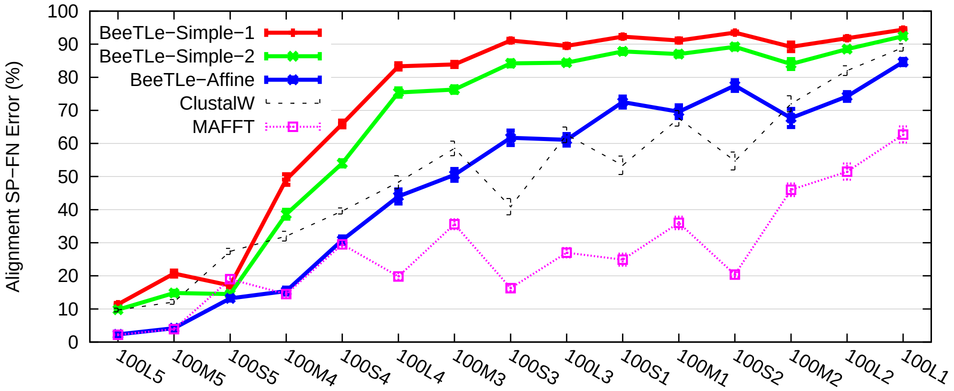

Figure 4. A comparison of alignment error for BeeTLe to other methods. We show alignment SP-FN error of BeeTLe in comparison to MAFFT and ClustalW on 100-taxon model conditions. Averages and standard error bars are shown; $n = 20$ for each reported value. (This figure appeared in Liu and Warnow [256].)

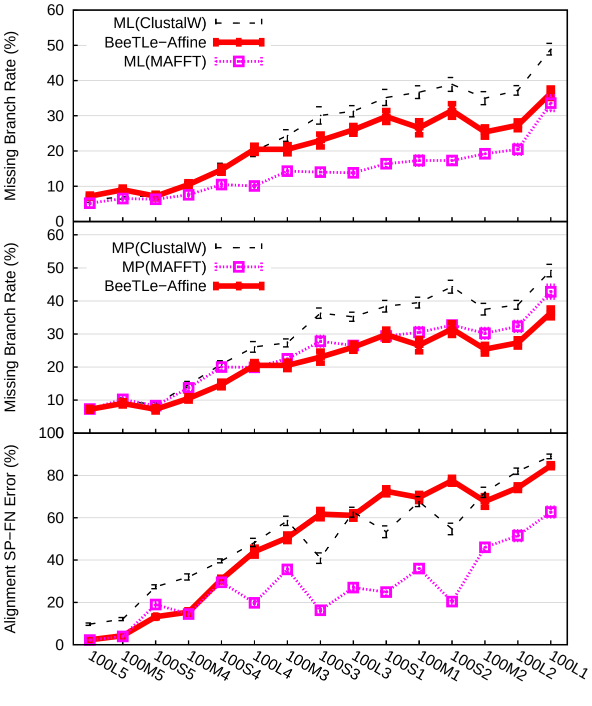

Figure 5. Performance of BeeTLe-Affine. We compare the BeeTLe-Affine trees to MP and ML trees on ClustalW and MAFFT alignments, and we also compare the alignments obtained by BeeTLe-Affine, ClustalW, and MAFFT, on 100-taxon model conditions. Averages and standard error bars are shown; n=20 for each reported value. (This figure appeared in Liu and Warnow [256].)

has lower SP-FN error than the other BeeTLe variants. The comparison to ClustalW shows that BeeTLe-Affine is less accurate on some models and more accurate on others; however, neither ClustalW nor BeeTLe-Affine comes close to the accuracy of MAFFT, except on the easiest 100-taxon models.

Figure 5 shows the direct comparison between BeeTLe-Affine and alignments and trees estimated using ClustalW and MAFFT. Since Figure 4 already gave the comparison with respect to alignment error, we focus only on tree estimation. Note that BeeTLe-Affine generally gives more accurate trees than MP(Clustal) and MP(MAFFT), except on the easiest models where they are all equally accurate. However, when compared to ML-based trees, the best results are clearly obtained using ML(MAFFT), with BeeTLe-Affine in second place and ML(Clustal) in last place.

3.1.3 Summary regarding the Treelength problem

Recall that BeeTLe is guaranteed to produce solutions to treelength optimization that are at least as good as POY, and in fact BeeTLe generally produces shorter trees than POY, as shown in Liu and Warnow [256]. Therefore, the performance of BeeTLe with respect to tree and alignment accuracy is a better indication of the consequences of using treelength for estimating alignments and trees than POY. As shown here, however, although improvements can be obtained by using this particular affine gap penalty (compared to the simple gap penalties that were examined), the alignments and trees are not as accurate as those produced using the better alignment methods (e.g., MAFFT) followed by maximum likelihood.

We note that the use of affine gap penalty treatments, although more general (and hence better) than simple gap penalties, are not necessarily sufficiently general for alignment estimation [264–268]; therefore, better trees might be obtained by optimizing treelength under other gap penalty treatments. In the meantime, the evidence suggests that optimizing treelength using the range of gap penalty treatments in common use (simple or affine penalties) is unlikely to yield the high quality alignments and trees that are needed for the best phylogenetic analyses.

3.2 Statistical co-estimation methods

Methods that co-estimate alignments and trees based upon statistical models of evolution that incorporate indels have also been developed. The simplest of these models are TKF1 [269] and TKF2 [270, 271], but more complex models have also been developed [272–278]. Many statistical methods (some which estimate trees from fixed alignments, and some which co-estimate alignments and trees) have been developed based on these models [273, 275, 276, 279–286], but only BAli-Phy [276] has been shown to be able to co-estimate alignments and trees on datasets with 100 sequences; the others are limited to much smaller datasets [287]. A recent technique [288] may be able to speed up calculations of likelihood under models that include indels and substitutions, but to date, none of the co-estimation methods has been able to run on datasets with more than about 200 sequences.

3.3 Other co-estimation methods

In addition to the statistical co-estimation methods described above, several methods are designed to return trees and alignments given unaligned sequences as input. Here we discuss three of these methods, SATé, SATCHMO-JS, and mega-phylogeny, each of which was discussed in the earlier section on alignment estimation. Of these three methods, SATé and mega-phylogeny are methods that compute an alignment and a maximum likelihood tree on the alignment, where mega-phylogeny uses RAxML and SATé uses either RAxML or FastTree-2 (depending on the user's preference). Although mega-phylogeny has been used on empirical datasets, to our knowledge it has not been tested on benchmark datasets, and so its performance (in terms of alignment and/or tree accuracy) is more difficult to assess. Therefore, we focus the rest of the discussion here on SATé and SATCHMO-JS.

3.3.1 SATé

SATé ("Simultaneous Alignment and Tree Estimation") [10, 39] is a method that was designed to estimate alignments and trees on very large datasets. SATé was discussed earlier in the section on multiple sequence alignment; here we focus on its performance as a method for estimating trees.

Unlike the statistical methods discussed earlier that explicitly consider Markov models of evolution that include indels and thus can have performance guarantees under such models, SATé has no such guarantees. Instead, the design of SATé is guided by the empirical objective of improving the accuracy of alignment and phylogeny estimations on very large datasets.

The observations that led to SATé come from studies that showed that existing nucleotide alignment methods had poor accuracy on large datasets that evolve down trees with high rates of substitutions and indels, and that some of the most accurate alignment estimation methods (e.g., MAFFT) have computational requirements (sometimes due to memory usage) that makes them unable to be run in their most accurate setting on datasets above a relatively small number of sequences. Therefore, while small datasets can be aligned with the best alignment methods, larger datasets must be aligned with less accurate methods. These observations together guided the design of SATé, which we now describe.

SATé uses an iterative process, in which each iteration begins with the tree from the previous iteration, and uses it (within a divide-and-conquer framework) to re-align the sequence dataset. Then a maximum likelihood tree is estimated on the new alignment, to produce a new tree. The first iteration begins with a fast two-phase method (for example, FastTree-2 on a MAFFT-PartTree alignment). The first few iterations provide the most improvement, and then the improvements level off. The user can provide a stopping rule based upon the number of iterations, the total amount of clock time, or stopping when the maximum likelihood score fails to improve.

Although iteration is an important aspect of SATé's algorithm design, the divide-and-conquer strategy used by SATé is equally important, and the strategy has changed since its initial version. In its first version [10], SATé divided the dataset into subsets by taking a centroid branch in the tree (which divides the dataset roughly into two equal parts) and branching out until the desired number of subsets is produced (32 by default, but this value could change, based on the dataset size). Each subset was then aligned using MAFFT in a highly accurate setting (-linsi), and the subset alignments were merged together using Muscle. Then, an ML tree was estimated on the resultant alignment using RAxML. SATé iterated until 24 hours had elapsed (finishing its final iteration if it began before the 24 hour deadline). SATé returns the tree/alignment pair with the best ML score.

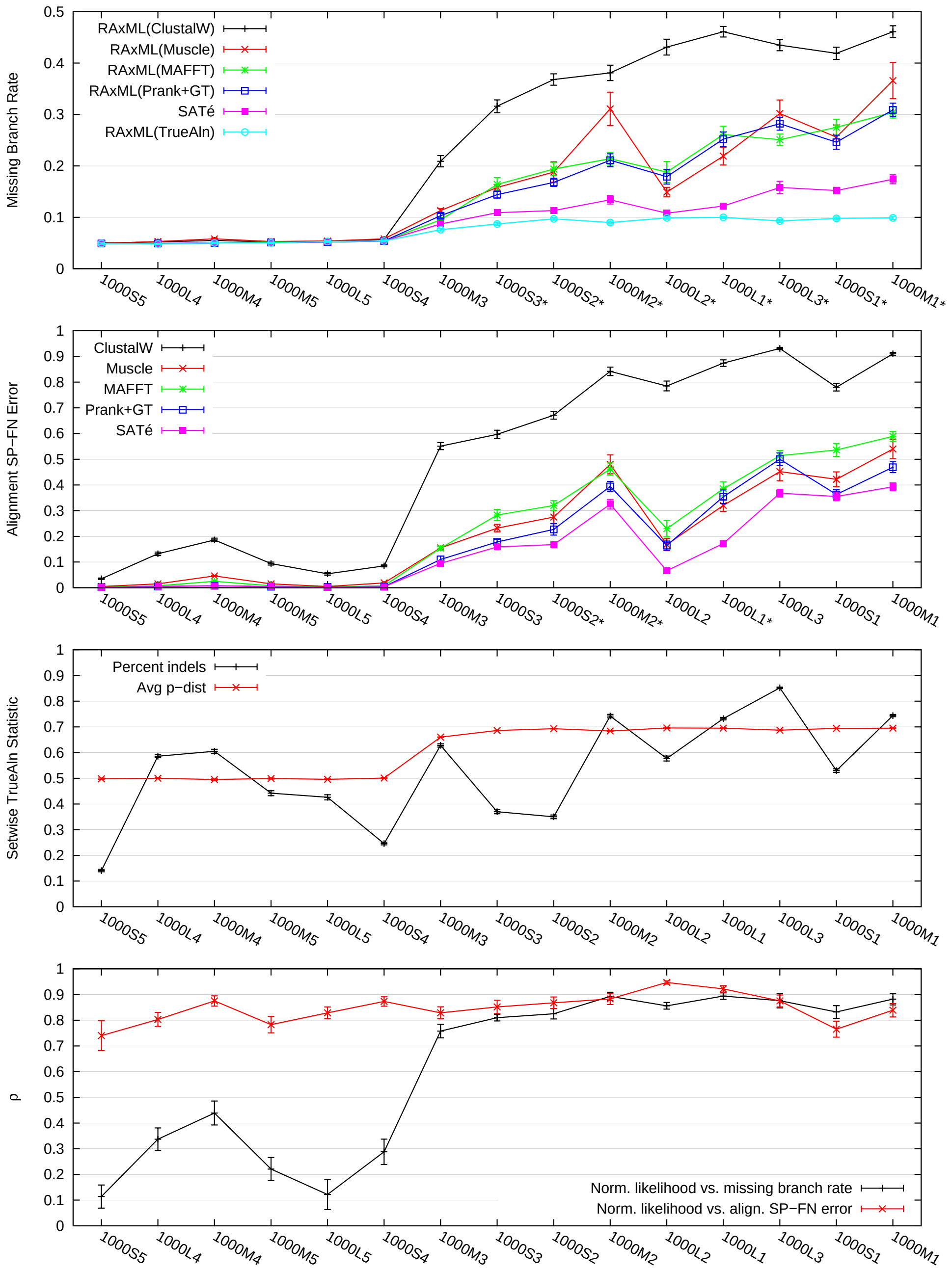

Figure 6. Performance of SATé on 1000-taxon model results. X-axes have the fifteen 1000-taxon models roughly sorted with respect to the phylogenetic estimation error, based on missing branch rates. The bottom two panels show true alignment (TrueAln) setwise statistics and Spearman rank correlation coefficients (ρ). All data points include standard error bars. For the top two panels, models on the x-axis followed by an asterisk indicate that SATé's performance was significantly better than the nearest two-phase method (paired t-tests, setwise $\alpha = 0.05$, $n = 40$ for each test). (This figure appeared in Liu *et al.* [10].)

This approach led to very good results, as shown in Figure 6, where we compare SATé to two-phase methods on 1000-taxon model trees. We include RAxML on ClustalW, Muscle, Prank+GT (Prank with a RAxML(MAFFT) guide tree), and the true alignment (TrueAln). The first two panels show the tree and alignment error for these methods. Note that on the easiest model conditions all methods produce the same level of accuracy as RAxML on the true alignment, although the estimated alignments have errors. However, on the harder models the phylogenetic estimations have different error rates, and SATé produces much more accurate trees and alignments than the other methods. Furthermore, SATé comes close to RAxML on the true alignment for all but the hardest models. The third panel gives the empirical statistics for the different models, and shows that factors that lead to datasets that are hard to align include the percent of the true alignment matrix that is gapped ("Percent indels") and the average p-distance[4] between pairs of sequences (again, based on the true alignment). Thus, alignments can be quite easy to estimate if the rate of substitutions is low enough (as reflected in the average p-distance), even if there are many indels, and it is only when the substitution rate is high enough and there are at least a moderate number of indels that alignments become difficult to estimate.

Recall that SATé generates a sequence of alignments and trees, and that the tree and alignment it outputs is the pair that has the best ML score. The fourth panel shows the results of a correlation analysis we performed to see if the ML score was correlated with either alignment accuracy or tree accuracy. What we observed is that tree error (measured using SP-FN) and ML score are very weakly correlated on the easier models, but then highly correlated on the harder models, while alignment error and ML score are highly correlated on all models. This correlation analysis suggests that using the ML score to select an alignment *might* be a good idea. It also suggests that when the conditions are such that improving the alignment would improve the tree (which seems to be the case for harder models rather than easier models), using the ML score to select the tree might also be a good idea. In other words, it suggested the following optimization problem:

Definition 4 *The ML_{GTR}* **Tree/Alignment Search Problem:** *Given a set S of unaligned sequences, find a tree T and an alignment A of S so as to maximize likelihood under GTR, treating gaps as missing data.*

[4] The p-distance between two aligned sequences is the number of positions in which the two sequences differ, and then normalized to give a number between 0 and 1.

However, the Jukes-Cantor version of this optimization problem is *not* a good way of trying to optimize alignments or trees, as we now show. Recalling the definition of "monotypic" alignments (see Section 2.3.6), we proved:

Theorem [From Liu *et al.* [39]]: If alignments are allowed to vary arbitrarily, then for all input sets of sequences, the alignment that gives the best Jukes-Cantor ML score (treating gaps as missing data) is monotypic, and every tree is an optimal solution for this alignment.

Thus, optimizing the ML score while allowing the alignments to change arbitrarily is not helpful. Given that we observed a correlation between the ML score and alignment and tree accuracy generated during a SATé analysis, how does this make sense? The explanation between these two seemingly contradictory statements is that the set of tree/alignment pairs in which we observed the correlation is not arbitrary. Instead, the set of alignments computed during a SATé run is not random at all, nor are the alignments and trees in that set explicitly modified to optimize ML. Instead, the alignments are computed using a divide-and-conquer strategy where MAFFT is used to align subsets and Muscle is used to merge these subset alignments. Thus, although allowing alignments to change arbitrarily is definitely not desirable (and will lead to an optimal ML score but poor trees), using ML to select among the alignments and trees produced during the SATé process *may* be beneficial.

SATé-II: Subsequent studies revealed that for some datasets, the SATé analysis was very slow, and this turned out to be due to some subset sizes being very large – too large, in fact, for MAFFT to comfortably analyze the dataset using its most accurate setting (-l -insi). A careful analysis revealed that these large subsets came about as a result of the specific decomposition we used (consider, for example, the result of using the SATé-decomposition on a caterpillar tree). We then changed the decomposition technique, as follows. The new decomposition keeps removing centroid edges in the subtrees that are created until every subtree has no more than a maximum number of leaves (200 by default). As a result, all the subsets are "small", and MAFFT -l -insi can comfortably analyze each subset. Note also that these subsets are not necessarily clades in the tree! The resultant version of SATé, called SATé-II, produces trees and alignments that are even more accurate than SATé, and is also faster. This is the version of SATé that is in the public distribution.

3.3.2 SATCHMO-JS

SATCHMO-JS [78] is another co-estimation method designed for empirical performance, and specifically for protein sequence alignment. SATCHMO stands for "Simultaneous Alignment and Tree Construction using Hidden Markov models", and SATCHMO-JS stands for "jump-start SATCHMO". The basic approach here has three steps. First, SATCHMO-JS computes a neighbor joining tree on a MAFFT alignment, and uses this tree to divide the dataset into clades, each of which has a maximum pairwise p-distance that is below some threshold. For each such subset, a tree is computed on the alignment using the SciPhy method [289]. These subtrees are then passed to the SATCHMO [77] method, which then completes the task of computing an overall alignment and tree. Finally, the branch lengths on the tree are optimized using maximum likelihood.

Thus, SATCHMO-JS is a hybrid between MAFFT and SATCHMO that combines the best of the two methods – MAFFT to align closely related sequences, and SATCHMO to align distantly related sequences. Since a tree is estimated at the same time as the alignment is estimated, the output is both a tree and an alignment - hence the name of the method.

As shown by Hagopian *et al.* [78], SATCHMO-JS produced more accurate alignments than MAFFT, SATCHMO, Clustalw and Muscle on several protein benchmarks. Furthermore, although by design SATCHMO-JS is slower than MAFFT (since it runs MAFFT to obtain its initial decomposition into subsets), it is much faster than SATCHMO, and was able to analyze a dataset with 500 protein sequences of moderate length (392 aa) in about 18 minutes.

It is worth noting that the way SATCHMO determines the tree cannot be described as finishing the alignment estimation and then computing a tree on that alignment; this distinguishes SATCHMO-JS from SATé and mega-phylogeny, which always return a maximum likelihood tree on the output alignment.

4 Tree estimation without full alignments

Because multiple sequence alignment estimation tends to be computationally intensive and inaccurate on large datasets that evolve with high rates of evolution [4, 6], estimating trees without any multiple sequence alignment step has obvious appeal. In this section, we first discuss alignment-free estimation, and then estimation methods that use multiple sequence alignment estimation on subsets but not on the full set of taxa.

4.1 Alignment-free Estimation

Potential Benefits of Alignment-free Estimation. There are several reasons that alignment-free estimation has been considered promising, which we briefly discuss here (see [290, 291] for longer and more detailed discussions). First, recall that standard multiple sequence alignments insert gaps between letters within sequences in order to "align" them, and hence only model homologies that result from substitutions and indels. However, genome-scale evolution is very complex, involving recombination, rearrangements, duplications, and horizontal gene transfers, none of which is easily handled in standard multiple sequence alignments. Thus, one of the points in favor of alignment-free estimation methods is that they may be more robust to these genome-scale events (rearrangements, recombination, duplications, etc.) than alignment-based methods. Another point in favor is that alignment-free methods are able to avoid the need to do each step of the standard analysis pipeline, and hence can be robust (possibly) to the difficulties in identifying orthology groups, aligning sequences, estimating gene trees, and combining gene trees and/or alignments. Thus, alignment-free estimation may be robust to some of the methodological challenges inherent in the standard phylogenetic pipeline.

Alignment-free estimation also have another distinct advantage, in that they enable the use of all the nucleotides in the genomes, not just the ones that fall into the regions identified for the phylogenomic analysis. Thus, alignment-free methods have the potential to utilize more of the genomic data, and this could enable more accurate trees.

Finally, as we shall see, alignment-free estimation is in general very fast, especially for datasets that involve many markers and/or many taxa. This is one of the big advantages over the standard analysis pipeline.

History of Alignment-free Estimation. The first method for alignment-free phylogeny estimation was developed in 1986 [292], and new methods continue to be developed [290, 291, 293–295].

Alignment-free methods are based upon computing distances (often, but not always, by computing k-mer distributions) between unaligned sequences. Once these distances are computed, trees can be computed on the resultant distance matrix, using distance-based methods (e.g., the well-known neighbor joining [121] method, but there are many others). Because both steps can be quite fast, these alignment-free methods can be applied to very large genomes.

While some of the alignment-free techniques for estimating pairwise distances are fairly heuristic, others use sophisticated statistical techniques, and some have provable performance under Markov models of evolution. A particularly exciting result is by Daskalakis and Roch [237], who gave a polynomial time distance-based alignment-free method and proved that it is statistically consistent under the TFK1 model [269].

These methods have shown some promise, as some simulation studies evaluating trees based on these distances have shown that these can be more accurate than trees based on distances calculated using estimated multiple sequence alignments [296]. Furthermore, plausible phylogenetic trees have been estimated on biological datasets using these methods [290, 291].

Comparison to two-phase methods Despite all the potential benefits of alignment-free estimation, there has not been any comparison of alignment-free methods to the most accurate ways of computing large trees, e.g., Bayesian or maximum likelihood analyses on good alignments. Instead, the only comparisons have been to distance-based phylogeny estimation, and in some cases only to distances computed using other alignment-free methods. Therefore, while there is distinct potential for alignment-free estimation to provide improved species tree estimations, this possibility has not been properly evaluated.

However, as noted before, alignment-free estimation does provide a distinct computational advantage over the standard pipelines, and it can enable the use of *all* of the genomic data. Thus, even if alignment-free estimation is not as accurate as maximum likelihood on alignments, the ability to use more data may offset the possible reduction in accuracy. The question might then become *whether more data analyzed using a less accurate method is as good as less data analyzed using a more accurate method*!

4.2 DACTAL

DACTAL [30], which stands for "Divide-And-Conquer Trees (almost) without ALignments", is a method that estimates trees without requiring an alignment on the full dataset. Unlike the methods discussed in the previous subsection, DACTAL is not truly

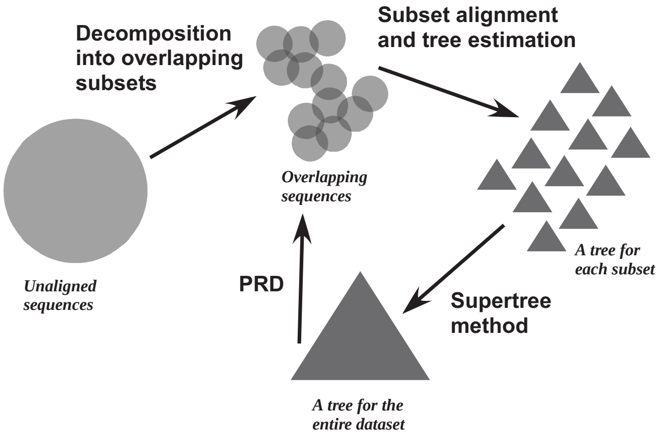
Decomposition into overlapping subsets
Subset alignment and tree estimation
Overlapping sequences
Unaligned sequences
PRD
A tree for each subset
Supertree method
A tree for the entire dataset

Figure 7. DACTAL algorithmic design. DACTAL can begin with an initial tree (bottom triangle), or through a technique that divides the unaligned sequence dataset into overlapping subsets. Each subsequent DACTAL iteration uses a decomposition strategy called "PRD" (padded recursive decomposition) to divide the dataset into small, overlapping subsets, estimates trees on each subset, and merges the small trees into a tree on the entire dataset. (This figure appeared in Nelesen *et al.* [30].)

alignment-free. Instead, DACTAL uses an iterative divide-and-conquer strategy, and estimates alignments and trees on small (and carefully selected) subsets of taxa. By ensuring that the subsets have sufficient overlap, this makes it possible to produce a tree on the full set of taxa. The key to making this work well is ensuring that the taxon sampling in each subset is favorable to tree and alignment estimation, the subsets are small enough that the best alignment and tree estimation methods can be used on them, and that the overlap patterns enable a highly accurate supertree to be estimated [297].

A DACTAL analysis can be initiated in one of several ways. The simplest way is to obtain a starting tree for the dataset (e.g., a taxonomy for the dataset, or an estimated tree obtained using a fast two-phase method). This starting tree is then used to decompose the dataset into small overlapping subsets of sequences that are close together in the starting tree, using the "PRD" technique [30, 298]. Alternatively, this decomposition into small overlapping subsets of similar sequences can be obtained using one of several BLAST-based decompositions [298].

Once this decomposition is computed, alignments and trees are computed on each subset, and the subtrees are merged into a tree on the full set of taxa using SuperFine [53, 299], a new supertree method that can produce more accurate trees on large datasets

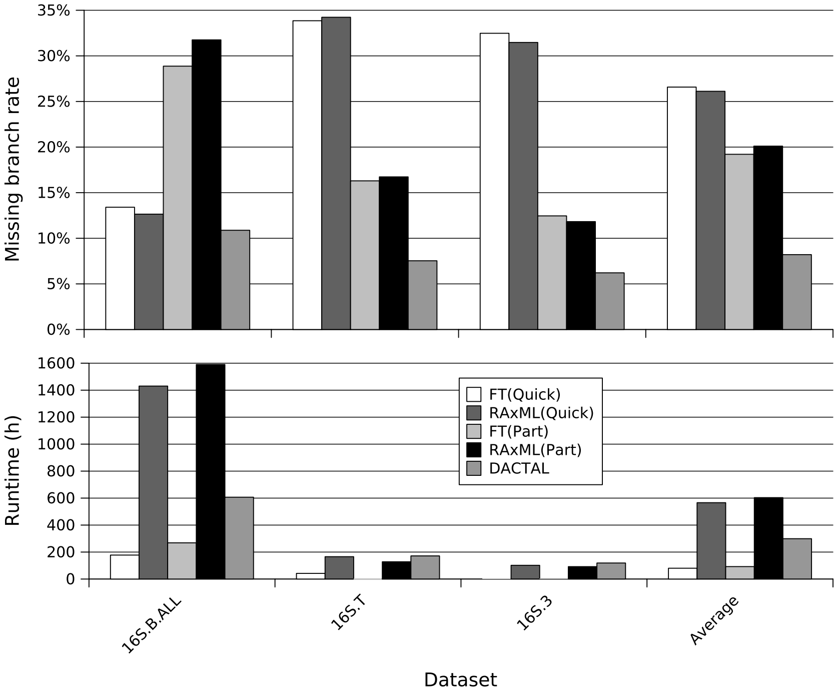

Figure 8. DACTAL (based upon five iterations) compared to ML trees computed on alignments of three large biological datasets with 6,323 to 27,643 sequences. We used FastTree-2 (FT) and RAxML to estimate ML trees on the MAFFT-PartTree (Part) and ClustalW-Quicktree (Quick) alignments. The starting tree for DACTAL on each dataset is FT(Part). (This figure appeared in Nelesen *et al.* [30].)

than other supertree methods [54, 300]. The resultant tree is then used to start the next iteration, which continues with a decomposition of the taxa into small, overlapping subsets, the estimation of alignments and trees on each subset, and the merger of the subset-trees into a tree on the full dataset. Finally, this iterative process (shown in Figure 7) continues for a user-defined maximum number of iterations.

DACTAL has comparable accuracy to SATé-I (the initial implementation of SATé as described by Liu *et al.* [10]), and substantially improved accuracy compared to the leading two-phase methods. on datasets with 1000 or more sequences. In addition, DACTAL is faster than SATé-I, and gives very good accuracy on large biological datasets.

Figure 8 [30] shows the results for five iterations of DACTAL on three large rRNA datasets with up to 27,643 sequences, in comparison to maximum likelihood trees (computed using FastTree-2 and RAxML) on two alignments (Clustal-Quicktree and MAFFT-Parttree). DACTAL was run using a subset decomposition size of 200 and RAxML(MAFFT) to estimate trees on the subsets. Each of these datasets has a reliable curated alignment based on rRNA structure [301]. The reference trees for this study were obtained by estimating maximum likelihood trees (using RAxML with bootstrapping) on the curated alignment, and then collapsing all branches with bootstrap support below 75%. Note the substantial reduction in tree error obtained using DACTAL in comparison to these two-phase methods. Furthermore, other analyses show that DACTAL is highly robust to its starting tree, and that even a single iteration produces a large improvement [30].

5 Lessons Learned and Future Directions

5.1 Basic Observations

This chapter has introduced several new methods and techniques for both alignment and phylogeny estimation, and shown that highly accurate large-scale estimation is beginning to be possible. However, there are several recurring themes in this chapter, which point to the limitations of the current research, and the need for future work. These are:

1. Only a few techniques have been developed that are capable of large-scale alignment or phylogeny estimation, and very few methods have been even tested on large datasets,

2. The standard criteria used to evaluate alignment estimation methods (e.g., the SP- and TC-scores) are not suited to predicting accuracy with respect to tree estimation, especially for large datasets,

3. Many promising alignment estimation methods have not been formally tested on simulated or biological benchmark datasets for their impact on phylogeny estimation,

4. The relative performance of phylogeny estimation methods (and perhaps of alignment estimation methods) can change with the number of taxa (e.g., compare HGT+FP and NJ in Figure 1), and so performance studies that only examine small datasets are not predictive of performance on larger datasets, and

5. In general, the simulation studies used to evaluate alignment or phylogeny estimation methods have been based on very simple models of evolution, and so it is not clear how well these methods will perform under more realistic conditions.

These limitations are due to a number of factors, one being that many (though not all) of the alignment estimation methods were developed for use by the structural biologists, and hence were tested with respect to the ability to identify structural and functional features. Since structural and functional homology may not be identical to positional homology, this contributes to the issues raised above. Furthermore, the two communities - phylogenetics and structural biology - are still fairly disconnected, and alignment methods continue to be tested almost exclusively on structural benchmarks, typically based on protein datasets. Bridging the gaps between the various disparate communities, including phylogenetics, structural biology, and functional genomics, may be necessary in order to change this practice.

Many of these issues also reflect the challenges in evaluating phylogeny estimation methods, especially on large datasets. For example, although we know a great deal about the performance of methods in terms of statistical consistency under the General Markov model and simpler models of sequence evolution, much less is known mathematically about performance on finite data, and even less about the performance under more complex models.

As a whole, these observations highlight the importance of benchmark datasets (whether biological or simulated), since mathematical results are typically limited to statistical consistency or model identifiability. However, biological datasets that are appropriate for evaluating phylogeny estimation methods are rare, since the true tree is rarely known. In our own studies, we have used biological datasets with curated alignments, and used maximum likelihood bootstrap trees estimated on these curated alignments as the benchmark trees. This approach has the advantage of being phylogenetically based, but has two disadvantages: the alignment itself (however well curated) may not be correct, and even if it is correct, the tree estimated on the alignment may also not be correct. Thus, more - and better - biological benchmarks are needed.

As noted, simulation studies are a standard technique used to evaluate phylogenetic estimation methods, but designing good simulation studies and extracting general principles from them is not easy. Among the many challenges, the first is that the models available in most (but not all) simulator software are too simple, generally no more complex than the models available in the phylogeny estimation software; thus, data generated by most simulation software do not exhibit the properties of biological data. Another challenge is that the relative performance of methods can change with the model condition, and exploring the parameter space is computationally infeasible. Perhaps because of this, many studies have focused on small trees, where more thorough exploration of the parameter space is feasible. However, computational challenges in analyzing large datasets also explains why few studies have examined performance on large datasets, and relatively little is known about performance on large datasets. All these issues add to the challenge in developing good benchmarks, and thus of understanding phylogeny estimation methods - especially on large datasets.

5.2 Future Research Directions

As noted, substantial progress towards developing methods that can estimate alignments and trees on very large datasets has been made. For example, we have presented new approaches for estimating alignments and trees, for co-estimating alignments and trees, and for phylogenetic placement, each of which has provided substantial improvements in accuracy (and in some cases scalability) over previous methods. Here we discuss additional research questions that remain, acknowledging that these are just a small sample of the open problems in this area.

5.2.1 Theoretical performance of phylogeny estimation methods under long indel models

While much is known about the theoretical performance of phylogeny estimation under indel-free models of evolution, much less is known about the statistical guarantees of methods when sequences evolve with indels; even the result by Daskalakis and Roch [237] establishing statistical consistency under a model with indels only allows indels of single nucleotides. Open questions here include the statistical guarantees (both statistical consistency and sequence length requirements) for methods under models with long indels given the true alignment, and when alignments must be estimated.

5.2.2 Sequence length requirements for phylogeny estimation

While much is known about the sequence length requirements under the General Markov model for many methods, several questions remain. For example, the best published upper bound on the sequence length requirement for maximum likelihood is no better than that of neighbor joining [166], but this bound is likely to be loose, as suggested by Roch [302]. Thus, a basic question is whether maximum likelihood is an absolute fast converging method? Other questions of this sort include the sequence length requirements of methods to recover some fixed percentage of the model tree bipartitions, or to estimate the model tree under more complex models than the General Markov model. Finally, from an empirical standpoint, it would be good to have implementations of absolute fast converging methods so that these methods can be compared to other methods, such as maximum likelihood.

5.2.3 Genome rearrangements and duplications

Genomes evolve with duplications, rearrangements (inversions, transpositions, transversions), fissions and fusions, and alignment and phylogeny estimation in the presence of these events is very complicated. While some work has been done on the problem of estimating whole genome alignments [303–311], much still needs to be done.

The problem of estimating trees in this case is similarly challenging, and is the subject of a chapter in this volume by Bernard Moret *et al.* A basic open problem here is whether genome-scale events and sequence evolution events can be analyzed together, since they are likely to be complementary.

5.2.4 Evolutionary Networks

The objective of finding a "Tree of Life" presents a very basic challenge, since not all evolution is tree-like (e.g., horizontal gene transfer [312–317] and hybridization [318–320]); thus, the phrase "Tree of Life" is in a sense a misnomer, as discussed by Mindell [321].

The failure of trees to completely represent the evolutionary history is most obvious in the case of hybridization, where two species hybridize to make a new species; clearly, this history cannot be represented by a tree, and instead requires a network (i.e., a graph that has cycles). However, in the case of horizontal gene transfer, the underlying species history may still be reasonably represented by a tree [45], and edges representing the HGT events can be added to the tree to create a network. (This is how the phylogenetic network for language evolution is obtained, where horizontal edges represent "borrowing" between languages [322, 323].) Thus, in the event of hybridization or horizontal gene transfer, the evolutionary history is best represented by a network, though the specific representation (and meaning of edges) can differ between these networks.

In the literature, the term "phylogenetic network" [324–326] has been used to describe these graphical models, but, as Morrison points out [325], this term is used for more than one purpose. That is, there are two types of networks that have been proposed: one type is suited for exploratory data analysis (EDA) and another that is suited for a hypotheses of evolutionary history. Morrison suggests that networks that are best suited for EDA of phylogenetic data should be referred to as "Data-Display Networks", and that networks that are graphical representations of a reticulate evolutionary history should be referred to as "Evolutionary Networks"; in accordance with his suggestions, we will use these terms here.

This distinction is important, since a data-display network does not provide any direct information about the evolutionary history. As Morrison says (page 47 [325]),

> The basic issue, of course, is the simple fact that data-display networks and evolutionary networks can *look* the same. That is, they both contain reticulations even if they represent different things... Many people seem to have confused the two types of network, usually by trying to interpret a data-display network as an evolutionary network... The distinction between the two types of network has frequently been noted in the literature, so it is hardly an original point for me to make here. Interestingly, a number of authors have explicitly

> noted the role of display networks in exploratory data analysis and then proceeded to treat them as genealogies anyway. It is perhaps not surprising, then, that non-experts repeatedly make the same mistake.

Both types of networks serve valuable purposes (as noted also by Morrison), but the purposes are different. However, there are many more methods that produce data-display networks than evolutionary networks, and the estimation of evolutionary networks is much more complicated and challenging. For examples of some of the few evolutionary network methods, see [327–333].

Note that while the species history is not tree-like, the evolution of individual genes may still be treelike. However, since reticulate events can result in genes with different trees, the detection of differences, sometimes strongly supported differences, between gene trees can indicate that some kind of reticulation may have occurred. However, gene trees can also be incongruent under other conditions, including cases where the species history is still treelike; a prime example of this is incomplete lineage sorting [40, 42, 57].

Many challenges remain for estimating evolutionary histories in the case of these events. One obvious challenge is the estimation of the underlying species tree from either gene sequence alignments or trees, for the case where genes evolve with horizontal gene transfer but without hybridization. Lapierre *et al.* [334] performed a simulation study to evaluate supertree and supermatrix methods for estimating the species tree, and found that supermatrix methods gave better results when there was only small amounts of horizontal gene transfer, and supertree methods were better when there were many horizontal gene transfers. A new method for estimating the underlying species tree, and a probabilistic analysis of this method, has also been developed by Roch and Snir [335], but has not been evaluated in simulation or on real data. New methods, and evaluations of these methods, are clearly needed.

In addition, since there are very few methods for estimating evolutionary networks, more work in this area needs to be done. However, all evolutionary networks - especially those that operate by combining estimated gene trees - need to be able to distinguish incongruence between estimated gene trees that is due to true reticulation (as in lateral gene transfer or hybridization), incongruence that is due to incomplete lineage sorting or gene duplication and loss (for which a species tree still makes sense), and incongruence due to estimation error. Recent progress has been made in developing statistical methods for distinguishing between different causes for incongruence, and for estimating evolutionary histories given a mixture of different events [333, 336, 337]. While these are still in their infancy, the potential for substantial advances in phylogenomic analysis could be very high.

5.2.5 Incorporating biological knowledge into alignment and phylogeny estimation

One of the most interesting developments in both alignment and phylogeny estimation is the attempt to utilize biological knowledge, especially about structure, into the process [102]. Although structurally-based alignments may not always reflect positional homology [100], the use of external knowledge has greatly impacted the alignment of sets of protein sequences that are close to the twilight zone, in which sequence similarity can be close to random while structural properties may still be conserved. A similar effort is occurring on the phylogeny estimation side, so that the Markov models used in phylogeny estimation (especially those involving protein analyses) are becoming more realistic [138, 139]. For example, Liberles *et al.* [139] say:

> At the interface of protein structure, protein biophysics, and molecular evolution there is a set of fundamental processes that generate protein sequences, structures and functions. A better understanding of these processes requires both biologically realistic models that bring structural and functional considerations into evolutionary analysis, and similarly incorporation of evolutionary and population genetic approaches into the analysis of protein structure and underlying protein biophysics... The potential benefits of the synergy between biophysical and evolutionary approaches can hardly be overestimated. Their integration allows us not only to incorporate structural constraints into improved evolutionary models, but also to investigate how natural selection interacts with biophysics and thus explain how both physical and evolutionary laws have shaped the properties of extant macromolecules.

But, as Claus Wilke said in “Bringing Molecules Back into Molecular Evolution” [138]:

> A side effect of the strong emphasis on developing sophisticated methods for sequence analysis has been that the underlying biophysical objects represented by the sequences, DNA molecules, RNA molecules, and proteins, have taken a

> back-seat in much computational molecular evolution work. The vast majority of algorithms for sequence analysis, for example, incorporate no knowledge of biology or biochemistry besides that DNA and RNA sequences use an alphabet of four letters, protein sequences use an alphabet of 20, and the genetic code converts one into the other. The choice to treat DNA, RNA, and proteins simply as strings of letters was certainly reasonable in the late 20th century. Computational power was limited and many basic aspects of sequence analysis were still relatively poorly understood. However, in 2012 we have extremely powerful computers and a large array of highly sophisticated algorithms that can analyze strings of letters. It is now time to bring the molecules back into molecular evolution.

Both Wilke and Liberles *et al.* point out excellent work that is being done to add more biological realism into existing models and methods, and are passionate about the potential benefits to science that could result. However, the challenges to using more realistic models in phylogenetic estimation are enormous. As Wilke's article suggests, the added complexity in these models will lead to increased computational challenges (e.g., more parameters to estimate for maximum likelihood, or longer MCMC runs to reach convergence for the Bayesian methods). However, there are other challenges as well: as discussed earlier, increased model complexity can lead to non-identifiable models, making inferences under the model less reliable and interpretable. Furthermore, it is not always the case that adding complexity (even if realistic) to a model will improve inferences under the model, and it may be that phylogeny estimation under simpler, not necessarily as realistic models, may give the most accurate results.

However, even this question depends on what one is trying to estimate. Is it just the gene tree, or also the numerical parameters on the tree? If more than just the topology, then which parameters? Or is the tree itself just a nuisance parameter, and the objective something else? Indeed, it is possible that more parameter rich and biologically realistic models may not have a substantial impact on phylogeny estimation, but they may improve the detection of selection, the estimation of dates at internal nodes in the tree, and the inference of protein function and structure. That said, the prospects for improved accuracy in biological understanding through these new approaches that integrate biological knowledge with statistical methods may be large, and are worth investigating.

A final point about biologically realistic models: even if they turn out not to be particularly useful for estimation, they are certainly useful for simulation! That is, simulating under more realistic models and then estimating under simpler models allows us to determine the robustness of the method to model violations, and to predict performance under more realistic conditions. Therefore, the development of improved statistical models of evolution may be important for testing methods, even if the use of these models for inference turns out to be impractical.

6 Conclusions

This chapter set out to survey large-scale phylogeny and alignment estimation. As we have seen, large-scale alignment and phylogeny estimation (whether of species or of individual genes) is a complicated problem. Despite the multitude of methods for each step, the estimation of very large sequence alignments or gene trees is still quite difficult, and the estimation of species phylogenies (whether trees or networks!), even more so. Furthermore, the challenges in large-scale estimation are not computational feasibility (running time and memory), as inferential methods can differ in their theoretical and empirical performance.

The estimation of very large alignments and trees, containing tens of thousands of sequences, is now feasible; see Tables 1 and 2 for a summary of the methods that have been demonstrated to be able to estimate alignments and trees, respectively, on at least 25,000 sequences. Even co-estimation of alignments and trees of this size is now feasible, as shown in Table 3. However, improvements in scalability of many alignment and tree estimation methods is likely, and some of the methods not listed in these tables may also be able to analyze datasets of this size.

Furthermore, very substantial progress has been made for large-scale estimation, and the field seems poised to move very dramatically forward. For example, there are new algorithmic approaches being used that have the ability to improve the performance of base methods for alignment and phylogeny estimation, including iteration, divide-and-conquer, probabilistic models, and the incorporation of external biological knowledge. At the same time, there is a definite move to incorporate more biological realism and knowledge into statistical estimation methods. The combination of these approaches is likely to be a very powerful tool towards substantial improvements in accuracy and, potentially, scalability.

Acknowledgments

During the time I wrote the paper, I was a Program Director at the National Science Foundation working on the BigData program; however, the research discussed in this paper took place over a span of many years. This research was therefore supported by U.S. National Science Foundation, Microsoft New England, the Guggenheim Foundation, the David and Lucile Packard Foundation, the Radcliffe Institute for Advanced Study, the Program for Evolutionary Dynamics at Harvard, the David Bruton Jr. Centennial Professorship in Computer Sciences at U.T. Austin, and two Faculty Research Assignments from the University of Texas at Austin.

It makes sense now to tell how some of the work in this paper came about. I was working with Randy Linder (UT-Austin Integrative Biology) on various problems, including large-scale alignment and phylogeny estimation. During our initial attempts to design a fast and accurate co-estimation method, we began by trying to come up with a better solution to the Treelength optimization problem. Our interest in treelength optimization convinced a colleague, Vijaya Ramachandran (UT-Austin Computer Science), to develop a fast exact median calculator [338], which led to an improved treelength estimator; however our subsequent studies [263] suggested that improving the treelength would not lead to improved alignments and trees. This led us to look for other approaches to obtain more accurate alignments and trees from large datasets. Our next attempts considered the impact of guide trees, which gave a small benefit [109], but even iterating in this manner also did not lead to substantial improvements. Finally, we developed SATé, the co-estimation method described earlier. In a very real sense, therefore, much of the work in this chapter was inspired by David Sankoff, since he introduced the treelength optimization problem. And so, I end by thanking David Sankoff for this, as well as many other things.

References

1. Dobzhansky T (1973) Nothing in biology makes sense except in the light of evolution. American Biology Teacher 35: 125–129.

2. de Chardin PT (1959) Le Phénomene Humain. Harper Perennial.

3. Eisen JA (1998) Phylogenomics: improving functional predictions for uncharacterized genes by evolutionary analysis. Genome Res 8: 163–167.

4. Wang L-S, Leebens-Mack J, Wall K, Beckmann K, de Pamphilis C, et al. (2011) The impact of protein multiple sequence alignment on phylogeny estimation. IEEE Trans Comp Biol Bioinf 8: 1108–1119.

5. Simmons M, Freudenstein J (2003) The effects of increasing genetic distance on alignment of, and tree construction from, rDNA internal transcribed spacer sequences. Mol Phylo Evol 26: 444–451.

6. Liu K, Linder CR, Warnow T (2010) Multiple sequence alignment: a major challenge to large-scale phylogenetics. PLoS Currents: Tree of Life.

7. Hall BG (2005) Comparison of the accuracies of several phylogenetic methods using protein and DNA sequences. Mol Evol Biol 22: 792–802.

8. Kumar S, Filipski A (2007) Multiple sequence alignment: In pursuit of homologous DNA positions. Genome Res 17: 127–135.

9. Ogden T, Rosenberg M (2006) Multiple sequence alignment accuracy and phylogenetic inference. Syst Biol 55: 314–328.

10. Liu K, Raghavan S, Nelesen S, Linder CR, Warnow T (2009) Rapid and accurate large-scale coestimation of sequence alignments and phylogenetic trees. Science 324: 1561–1564.

11. Morrison D (2006) Multiple sequence alignment for phylogenetic purposes. Australian Syst Bot 19: 479–539.

12. Graybeal A (1998) Is it better to add taxa or characters to a difficult phylogenetic problem? Syst Biol 47: 9–17.

13. Pollock D, Zwickl D, McGuire J, Hillis D (2002) Increased taxon sampling is advantageous for phylogenetic inference. Syst Biol 51: 664–71.

14. Zwickl D, Hillis D (2002) Increased taxon sampling greatly reduces phylogenetic error. Syst Biol 51: 588–98.

15. Hillis D (1996) Inferring complex phylogenies. Nature 383: 130–131.

16. Felsenstein J (2003) Inferring Phylogenies. Sinauer Associates, Sunderland, Massachusetts.

17. Kim J, Warnow T (1999). Tutorial on phylogenetic tree estimation. Presented at the ISMB 1999 conference, available on-line at http://www.cs.utexas.edu/users/tandy/tutorial.ps.

18. Linder CR, Warnow T (2005) An overview of phylogeny reconstruction. In: Aluru S, editor, Handbook of Computational Molecular Biology, CRC Press, volume 9 of *Chapman and Hall/CRC Computer and Information Science Series*.

19. Semple C, Steel M (2003) Phylogenetics. Oxford University Press.

20. Hillis D, Moritz C, Mable B, editors (1996) Molecular Systematics. Sinauer Associates, Inc.

21. Ortuno F, Valenzuela O, Pomares H, Rojas F, Florido J, et al. (2013) Predicting the accuracy of multiple sequence alignment algorithms by using computational intelligent techniques. Nucleic Acids Res 41.

22. Whelan S, Lin P, Goldman N (2001) Molecular phylogenetics: state-of-the-art methods for looking into the past. TRENDS in Genetics 17: 262–272.

23. Goldman N, Yang Z (2008) Introduction: Statistical and computational challenges in molecular phylogenetics and evolution. Philosophical Transactions of the Royal Society B: Biological Sciences 363: 3889-3892.

24. Kemena C, Notredame C (2009) Upcoming challenges for multiple sequence alignment methods in the high-throughput era. Bioinf 25: 2455–2465.

25. Do C, Katoh K (2008) Protein multiple sequence alignment. In: Methods in Molecular Biology: Functional Proteomics, Methods and Protocols, Humana Press, volume 484. pp. 379–413.

26. Mokaddem A, Elloumi M (2011) Algorithms for the alignment of biological sequences. In: Elloumi M, Zomaya A, editors, Algorithms in Computational Molecular Biology, John Wiley & Sons, Inc. doi:10.1002/9780470892107.ch12.

27. Pei J (2008) Multiple protein sequence alignment. Curr Opin Struct Biol 18: 382–386.

28. Sievers F, Wilm A, Dineen D, Gibson T, Karplus K, et al. (2011) Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. Mol Syst Biol 7.

29. Katoh K, Toh H (2007) PartTree: an algorithm to build an approximate tree from a large number of unaligned sequences. Bioinf 23(3): 372–374.

30. Nelesen S, Liu K, Wang LS, Linder CR, Warnow T (2012) DACTAL: divide-and-conquer trees (almost) without alignments. Bioinf 28: i274–i282.

31. Larkin MA, Blackshields G, Brown NP, Chenna R, Mcgettigan PA, et al. (2007) ClustalW and ClustalX version 2.0. Bioinf 23: 2947–2948.

32. Lassmann T, Frings O, Sonnhammer E (2009) Kalign2: high-performance multiple alignment of protein and nucleotide sequences allowing external features. Nucleic Acids Res 37: 858–865.

33. Neuwald A (2009) Rapid detection, classification, and accurate alignment of up to a million or more related protein sequences. Bioinf 25: 1869–1875.

34. Price MN, Dehal PS, Arkin AP (2010) FastTree-2 – approximately maximum-likelihood trees for large alignments. PLoS ONE 5: e9490. doi:10.1371/journal.pone.0009490.

35. Smith S, Beaulieu J, Stamatakis A, Donoghue M (2011) Understanding angiosperm diversification using small and large phylogenetic trees. American J Botany 98: 404–414.

36. Stamatakis A (2006) RAxML-VI-HPC: Maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. Bioinf 22: 2688-2690.

37. Goloboff PA, Catalano SA, Marcos Mirande J, Szumik CA, Salvador Arias J, et al. (2009) Phylogenetic analysis of 73,060 taxa corroborates major eukaryotic groups. Cladistics 25: 211–230.

38. Goloboff P, Farris J, Nixon K (2008) TNT, a free program for phylogenetic analysis. Cladistics 24: 774-786.

39. Liu K, Warnow T, Holder M, Nelesen S, Yu J, et al. (2011) SATé-II: Very fast and accurate simultaneous estimation of multiple sequence alignments and phylogenetic trees. Syst Biol 61: 90-106.

40. Maddison W (1997) Gene trees in species trees. Syst Bio 46: 523–536.

41. Delsuc F, Brinkmann H, Philippe H (2005) Phylogenomics and the reconstruction of the tree of life. Nature Reviews Genetics 6: 361–375.

42. Edwards SV (2009) Is a new and general theory of molecular systematics emerging? Evolution 63: 1–19.

43. Dunn CW, Hejnol A, Matus DQ, Pang K, Browne WE, et al. (2008) Broad phylogenomic sampling improves resolution of the animal tree of life. Nature 452: 745–749.

44. Wu D, Hugenholtz P, Mavromatis K, Pukall R, Dalin E, et al. (2009) A phylogeny-driven genomic encyclopaedia of bacteria and archaea. Nature 462: 1056–1060.

45. Eisen J, Fraser C (2003) Phylogenomics: intersection of evolution and genomics. Science 300: 1706–1707.

46. Bininda-Emonds O, editor (2004) Phylogenetic Supertrees: Combining information to reveal the Tree of Life. Kluwer Academic Publishers.

47. Baum B, Ragan MA (2004) The MRP method. In: Bininda-Emonds ORP, editor, Phylogenetic Supertrees: combining information to reveal The Tree Of Life. Kluwer Academic, Dordrecht, the Netherlands, pp. 17-34.

48. Chen D, Eulenstein O, Fernández-Baca D, Sanderson M (2006) Minimum-flip supertrees: Complexity and algorithms. IEEE/ACM Trans Comp Biol Bioinf 3: 165-173.

49. Bininda-Emonds ORP (2004) The evolution of supertrees. Trends in Ecology and Evolution 19: 315-322.

50. Snir S, Rao S (2010) Quartets MaxCut: a divide and conquer quartets algorithm. IEEE/ACM Trans Comput Biol Bioinf 7: 704-718.

51. Steel M, Rodrigo A (2008) Maximum likelihood supertrees. Syst Biol 57: 243–250.

52. Swenson M, Suri R, Linder C, Warnow T (2011) An experimental study of Quartets MaxCut and other supertree methods. Algs Mol Biol 6(1): 7.

53. Swenson M, Suri R, Linder C, Warnow T (2012) SuperFine: fast and accurate supertree estimation. Syst Biol 61: 214-227.

54. Nguyen N, Mirarab S, Warnow T (2012) MRL and SuperFine+MRL: new supertree methods. Algs Mol Biol 7:3.

55. Than CV, Nakhleh L (2009) Species tree inference by minimizing deep coalescences. PLoS Comp Biol 5.

56. Boussau B, Szollosi G, Duret L, Gouy M, Tannier E, et al. (2013) Genome-scale co-estimation of species and gene trees. Genome Res 23(2):323–330.

57. Degnan JH, Rosenberg NA (2009) Gene tree discordance, phylogenetic inference and the multispecies coalescent. Trends Ecol Evol 26: 332-340.

58. Chaudhary R, Bansal MS, Wehe A, Fernández-Baca D, Eulenstein O (2010) iGTP: a software package for large-scale gene tree parsimony analysis. BMC Bioinf 11: 574.

59. Larget B, Kotha SK, Dewey CN, Ané C (2010) BUCKy: Gene tree/species tree reconciliation with the Bayesian concordance analysis. Bioinf 26: 2910–2911.

60. Yu Y, Warnow T, Nakhleh L (2011) Algorithms for MDC-based multi-locus phylogeny inference: Beyond rooted binary gene trees on single alleles. J Comp Biol 18: 1543–1559.

61. Yang J, Warnow T (2011) Fast and accurate methods for phylogenomic analyses. BMC Bioinf 12 (Suppl 9): S4. doi:10.1186/1471-2105-12-S9-S4.

62. Liu L, Yu L, Edwards S (2010) A maximum pseudo-likelihood approach for estimating species trees under the coalescent model. BMC Evol Biol 10: 302.

63. Chauve C, Doyon JP, El-Mabrouk N (2008) Gene family evolution by duplication, speciation, and loss. J Comp Biol 15: 1043-1062.

64. Hallett MT, Lagergren J (2000) New algorithms for the duplication-loss model. In: Proceedings RECOMB 2000. New York: ACM Press, pp. 138–146.

65. Doyon JP, Chauve C (2011) Branch-and-bound approach for parsimonious inference of a species tree from a set of gene family trees. Adv Exp Med Biol 696: 287-295.

66. Ma B, Li M, Zhang L (2000) From gene trees to species trees. SIAM J Comput 30: 729–752.

67. Zhang L (2011) From gene trees to species trees II: Species tree inference by minimizing deep coalescence events. IEEE/ACM Trans Comp Biol Bioinf 8: 1685-1691.

68. Arvestad L, Berglung AC, Lagergren J, Sennblad B (2004) Gene tree reconstruction and orthology analysis based on an integrated model for duplications and sequence evolution. In: Bininda-Emonds O, editor, Proc RECOMB 2004, pp. 238-252.

69. Sennblad B, Lagergren J (2009) Probabilistic orthology analysis. Syst Biol 58: 411–424.

70. Edwards S, Liu L, Pearl D (2007) High-resolution species trees without concatenation. Proc Nat Acad Sciences (PNAS) 104: 5936-5941.

71. Heled J, Drummond AJ (2010) Bayesian inference of species trees from multilocus data. Mol Biol Evol 27: 570–580.

72. Roch S (2013) An analytical comparison of multilocus methods under the multispecies coalescent: The three-taxon case. In: Proc. Pacific Symposium on Biocomputing 18:297–306.

73. Kopelman NM, Stone L, Gascuel O, Rosenberg NA (2013) The behavior of admixed populations in neighbor-joining inference of population trees. In: Proc. Pacific Symposium on Biocomputing 18.

74. Degnan JH (2013) Evaluating variations on the STAR algorithm for relative efficiency and sample sizes needed to reconstruct species trees. In: Proc. Pacific Symposium on Biocomputing 18: 262–272.

75. Bayzid M, Mirarab S, Warnow T (2013) Inferring optimal species trees under gene duplication and loss. In: Proc. Pacific Symposium on Biocomputing 18:250–261.

76. Pei J, Grishin N (2007) PROMALS: towards accurate multiple sequence alignments of distantly related proteins. Bioinf 23: 802–808.

77. Edgar RC, Sjölander K (2003) SATCHMO: Sequence alignment and tree construction using hidden Markov models. Bioinf 19: 1404–1411.

78. Hagopian R, Davidson J, Datta R, Jarvis G, Sjölander K (2010) SATCHMO-JS: a webserver for simultaneous protein multiple sequence alignment and phylogenetic tree construction. Nucl Acids Res 38 (Web Server Issue): W29–W34.

79. O'Sullivan O, Suhre K, Abergel C, Higgins D, Notredame C (2004) 3DCoffee: combining protein sequences and structure within multiple sequence alignments. J Mol Biol 340: 385–395.

80. Zhou H, Zhou Y (2005) SPEM: improving multiple sequence alignment with sequence profiles and predicted secondary structures. Bioinf 21: 3615–3261.

81. Deng X, Cheng J (2011) MSACompro: protein multiple sequence alignment using predicted secondary structure, solvent accessibility, and residue-residue contacts. BMC Bioinf 12: 472.

82. Roshan U, Livesay DR (2006) Probalign: Multiple sequence alignment using partition function posterior probabilities. Bioinf 22: 2715-21.

83. Roshan U, Chikkagoudar S, Livesay DR (2008) Searching for RNA homologs within large genomic sequences using partition function posterior probabilities. BMC Bioinf 9: 61.

84. Do C, Mahabhashyam M, Brudno M, Batzoglou S (2006). PROBCONS: Probabilistic consistency-based multiple sequence alignment of amino acid sequences. Software available at `http://probcons.stanford.edu/download.html`.

85. Nawrocki EP, Kolbe DL, Eddy SR (2009) Infernal 1.0: Inference of RNA alignments. Bioinf 25: 1335-1337.

86. Nawrocki EP (2009) Structural RNA Homology Search and Alignment using Covariance Models. Ph.D. thesis, Washington University in Saint Louis, School of Medicine.

87. Gardner D, Xu W, Miranker D, Ozer S, Cannonne J, et al. (2012) An accurate scalable template-based alignment algorithm. In: Proc. International Conference on Bioinformatics and Biomedicine, 2012. pp. 237–243.

88. Edgar RC (2004) MUSCLE: a multiple sequence alignment method with reduced time and space complexity. BMC Bioinf 5: 113.

89. Mirarab S, Warnow T (2011) FastSP: Linear-time calculation of alignment accuracy. Bioinf 27: 3250–3258.

90. Blackburne B, Whelan S (2012) Measuring the distance between multiple sequence alignments. Bioinf 28: 495–502.

91. Stojanovic N, Florea L, Riemer C, Gumucio D, Slightom J, et al. (1999) Comparison of five methods for finding conserved sequences in multiple alignments of gene regulatory regions. Nucleic Acids Res 27: 3899–3910.

92. Edgar R (2010) Quality measures for protein alignment benchmarks. Nucleic Acids Research 7: 2415–2153.

93. Thompson JD, Plewniak F, Poch O (1999) A comprehensive comparison of multiple sequence alignment programs. Nucleic Acids Research 27: 2682–2690.

94. Thompson J, Plewniak F, Poch O (1999) BAliBASE: A benchmark alignments database for the evaluation of multiple sequence alignment programs. Bioinf 15: 87–88.

95. Raghava G, Searle SM, Audley PC, Barber JD, Barton GJ (2003) Oxbench: A benchmark for evaluation of protein multiple sequence alignment accuracy. BMC Bioinf 4: 47.

96. Gardner P, Wilm A, Washietl S (2005) A benchmark of multiple sequence alignment programs upon structural RNAs. Nucleic Acids Res 33: 2433–2439.

97. Walle ILV, Wyns L (2005) SABmark-a benchmark for sequence alignment that covers the entire known fold space. Bioinf 21: 1267–1268.

98. Carroll H, Beckstead W, O'Connor T, Ebbert M, Clement M, et al. (2007) DNA reference alignment benchmarks based on tertiary structure of encoded proteins. Bioinf 23: 2648–2649.

99. Blazewicz J, Formanowicz P, Wojciechowski P (2009) Some remarks on evaluating the quality of the multiple sequence alignment based on the BAliBASE benchmark. Int J Appl Math Comput Sci 19: 675–678.

100. Iantomo S, Gori K, Goldman N, Gil M, Dessimoz C (2012) Who watches the watchmen? An appraisal of benchmarks for multiple sequence alignment. arXiv:12112160 [q-bio.QM].

101. Aniba M, Poch O, Thompson JD (2010) Issues in bioinformatics benchmarking: the case study of multiple sequence alignment. Nucleic Acids Res 38: 7353–7363.

102. Morrison DA (2009) Why would phylogeneticists ignore computerized sequence alignment? Syst Biol 58: 150–158.

103. Reeck G, de Haen C, Teller D, Doolitte R, Fitch W, et al. (1987) "homology" in proteins and nucleic acids: a terminology muddle and a way out of it. Cell 50: 667.

104. Galperin M, Koonin E (2012) Divergence and convergence in enzyme evolution. J Biol Chem 287: 21–28.

105. Sjolander K (2010) Getting started in structural phylogenomics. PLoS Comput Biol 6: e1000621.

106. Katoh K, Kuma K, Miyata T, Toh H (2005) Improvement in the acccuracy of multiple sequence alignment MAFFT. Genome Inf 16: 22–33.

107. Do C, Mahabhashyam M, Brudno M, Batzoglou S (2005) PROBCONS: Probabilistic consistency-based multiple sequence alignment. Genome Res 15: 330–340.

108. Loytynoja A, Goldman N (2005) An algorithm for progressive multiple alignment of sequences with insertions. Proc Nat Acad Sci 102: 10557-10562.

109. Nelesen S, Liu K, Zhao D, Linder CR, Warnow T (2008) The effect of the guide tree on multiple sequence alignments and subsequent phylogenetic analyses. In: Proc Pacific Symposium on Biocomputing 13: 15–24.

110. Fletcher W, Yang Z (2010) The effect of insertions, deletions, and alignment errors on the branch-site test of positive selection. Mol Biol Evol 27: 2257–2267.

111. Penn O, Privman E, Landan G, Graur D, Pupko T (2010) An alignment confidence score capturing robustness to guide tree uncertainty. Mol Biol Evol 27: 1759–1767.

112. Toth A, Hausknecht A, Krisai-Greilhuber I, Papp T, Vagvolgyi C, et al. (2013) Iteratively refined guide trees help improving alignment and phylogenetic inference in the mushroom family *bolbitiaceae*. PLoS One 8: e56143.

113. Capella-Gutiérrez S, Gabaldón T (2013) Measuring guide-tree dependency of inferred gaps for progressive aligners. Bioinf 29(8):1011–7.

114. Preusse E, Quast C, Knittel K, Fuchs B, Ludwig W, et al. (2007) SILVA: a comprehensive online resource for quality checked and aligned ribosomal RNA sequence data compatible with ARB. Nucleic Acids Res 35: 718–96.

115. DeSantis T, Hugenholtz P, Keller K, Brodie E, Larsen N, et al. (2006) NAST: a multiple sequence alignment server for comparative analysis of 16S rRNA genes. Nucleic Acids Res 34: W394–9.

116. Lytynoja A, Vilella AJ, Goldman N (2012) Accurate extension of multiple sequence alignments using a phylogeny-aware graph algorithm. Bioinf 28: 1685–1691.

117. Papadopoulos JS, Agarwala R (2007) COBALT: constraint-based alignment tool for multiple protein sequences. Bioinf 23: 1073–1079.

118. Berger SA, Stamatakis A (2011) Aligning short reads to reference alignments and trees. Bioinf 27: 2068–2075.

119. Sievers F, Dineen D, Wilm A, Higgins DG (2013) Making automated multiple alignments of very large numbers of protein sequences. Bioinf 29(8): 989–995.

120. Smith S, Beaulieu J, Donoghue M (2009) Mega-phylogeny approach for comparative biology: an alternative to supertree and supermatrix approaches. BMC Evol Biol 9: 37.

121. Saitou N, Nei M (1987) The neighbor-joining method: A new method for reconstructing phylogenetic trees. Mol Biol Evol 4: 406–425.

122. Roquet C, Thuiller W, Lavergne S (2013) Building megaphylogenies for macroecology: taking up the challenge. Ecography 36: 013026.

123. Steel MA (1994) Recovering a tree from the leaf colourations it generates under a Markov model. Appl Math Lett 7: 19–24.

124. Evans S, Warnow T (2005) Unidentifiable divergence times in rates-across-sites models. IEEE/ACM Trans Comp Biol Bioinf 1: 130-134.

125. Tavaré S (1986) Some probabilistic and statistical problems in the analysis of DNA sequences. In: Lectures on Mathematics in the Life Sciences, volume 17. pp. 57–86.

126. Dayhoff M, Schwartz R, Orcutt B (1978) A model of evolutionary change in proteins. In: Dayhoff M, editor, Atlas of Protein Sequence and Structure, National Biomedical Research Foundation. p. 345352.

127. Lakner C, Holder M, Goldman N, Naylor G (2011) What's in a likelihood? Simple models of protein evolution and the contribution of structurally viable reconstructions to the likelihood. Syst Biol 60: 161–174.

128. Le S, Gascuel O (2008) An improved general amino acid replacement matrix. Mol Biol Evol 25: 1307–1320.

129. Whelan S, Goldman N (2001) A general empirical model of protein evolution derived from multiple protein families using a maximum-likelihood approach. Mol Biol Evol 18: 691-699.

130. Kosiol C, Goldman N (2005) Different versions of the Dayhoff rate matrix. Mol Biol Evol 22: 193–199.

131. Thorne J (2000) Models of protein sequence evolution and their applications. Curr Opin Genet Dev 10: 602–605.

132. Thorne J, Goldman N (2003) Probabilistic models for the study of protein evolution. In: Balding D, Bishop M, Cannings C, editors, Handbook of Statistical Genetics, John Wiley & Sons. pp. 209-226.

133. Adachi J, Hasegawa M (1996) Model of amino acid substitution in proteins encoded by mitochondrial DNA. J Mol Evol 42: 459-468.

134. Goldman N, Yang Z (1994) A codon-based model of nucleotide substitution for protein-coding DNA sequences. Mol Biol Evol 11: 725–736.

135. Scherrer M, Meyer A, Wilke C (2012) Modeling coding-sequence evolution within the context of residue solvent accessibility. BMC Evol Biol 12: 179.

136. Mayrose I, Doron-Faigenbom A, Bacharach E, Pupko T (2007) Towards realistic codon models: among site variability and dependency of synonymous and nonsynonymous rates. Bioinf 23: i319–i327.

137. Abascal F, Zardoya R, Posada D (2005) ProtTest: selection of best-fit models of protein evolution. Bioinf 21: 2104–2105.

138. Wilke C (2012) Bringing molecules back into molecular evolution. PLoS Comput Biol 8: e1002572.

139. Liberles D, Teichmann S, et al (2012) The inference of protein structure, protein biophysics, and molecular evolution. Protein Science 21: 769–785.

140. Lopez P, Casane D, Philippe H (2002) Heterotachy, an important process of protein evolution. Mol Biol Evol 19: 1–7.

141. Whelan S (2008) Spatial and temporal heterogeneity in nucleotide sequence evolution. Mol Biol Evol 25: 1683–1694.

142. Tuffley C, Steel M (1997) Links between maximum likelihood and maximum parsimony under a simple model of site substitution. Bull Math Bio 59: 581–607.

143. Steel MA (2011) Can we avoid 'SIN' in the House of 'No Common Mechanism'? Syst Biol 60: 96–109.

144. Lobkovsky A, Wolf Y, Koonin E (2013) Gene frequency distributions reject a neutral model of genome evolution. Genome Biol Evol 5: 233-242.

145. Galtier N, Gouy M (1998) Inferring pattern and process: maximum-likelihood implementation of a nonhomogeneous model of DNA sequence evolution for phylogenetic analysis. Mol Biol Evol 15: 871–9.

146. Foulds LR, Graham RL (1982) The Steiner problem in phylogeny is NP-complete. Adv Appl Math 3: 43–49.

147. Felsenstein J (1981) Evolutionary trees from DNA sequences: a maximum likelihood approach. J Mol Evol 17: 368-376.

148. Allman ES, Ané C, Rhodes J (2008) Identifiability of a Markovian model of molecular evolution with gamma-distributed rates. Advances Appled Probability 40: 229–249.

149. Allman ES, Rhodes J (2008) Identifying evolutionary trees and substitution parameters for the general Markov model with invariable sites. Mathematical Biosciences 211: 18–33.

150. Allman ES, Rhodes JA (2006) The identifiability of tree topology for phylogenetic models, including covarian and mixture models. J Comp Biol 13: 1101–1113.

151. Atteson K (1999) The performance of neighbor-joining methods of phylogenetic reconstruction. Algorithmica 25: 251278.

152. Chang J (1996) Full reconstruction of Markov models on evolutionary trees: identifiability and consistency. Mathematical Biosciences 137: 51-73.

153. Steel MA (2010) Consistency of Bayesian inference of resolved phylogenetic trees. ArXiv:10012684 [q-bioPE] .

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

155. Chang JT (1996) Inconsistency of evolutionary tree topology reconstruction methods when substitution rates vary across characters. Mathematical Biosciences 134: 189–215.

156. Matsen F, Steel M (2007) Phylogenetic mixtures on a single tree can mimic a tree of another topology. Syst Biol 56: 767–775.

157. Allman E, Rhodes J, Sullivant S (2012) When do phylogenetic mixture models mimic other phylogenetic models? Syst Biol 61: 1049-1059.

158. Erdos P, Steel M, Szekely L, Warnow T (1997) Local quartet splits of a binary tree infer all quartet splits via one dyadic inference rule. Computers and Artificial Intelligence 16: 217–227.

159. Erdos P, Steel M, Szekely L, Warnow T (1999) A few logs suffice to build (almost) all trees (i). Random Structures and Algorithms 14: 153–184.

160. Erdos P, Steel M, Szekely L, Warnow T (1999) A few logs suffice to build (almost) all trees (ii). Theoretical Computer Science 221: 77-118.

161. Lacey MR, Chang JT (2006) A signal-to-noise analysis of phylogeny estimation by neighbor-joining: insufficiency of polynomial length sequences. Math Biosci 199: 188215.

162. Csürős M, Kao MY (1999) Recovering evolutionary trees through harmonic greedy triplets. Proc SODA 99 : 261–270.

163. Csuros M (2002) Fast recovery of evolutionary trees with thousands of nodes. J Comp Biol 9: 277-297.

164. Huson D, Nettles S, Warnow T (1999) Disk-covering, a fast converging method for phylogenetic tree reconstruction. J Comp Biol 6: 369–386.

165. Steel MA, Szekely LA (1999) Inverting random functions. Ann Comb 3: 103–113.

166. Steel MA, Szekely LA (2002) Inverting random functions - II: Explicit bounds for discrete maximum likelihood estimation, with applications. SIAM J Discrete Math 15: 562575.

167. King V, Zhang L, Zhou Y (2003) On the complexity of distance-based evolutionary tree reconstruction. In: SODA: Proceedings of the ACM-SIAM Symposium on Discrete Algorithms. pp. 444-453.

168. Mossel E, Roch S (2005) Learning nonsingular phylogenies and hidden Markov models. In: Proc. 37th Symp. on the Theory of Computing (STOC'05). pp. 366–376.

169. Mossel E, Roch S (2006) Learning nonsingular phylogenies and hidden Markov models. Ann Appl Prob 16: 538-614.

170. Daskalakis C, Mossel E, Roch S (2006) Optimal phylogenetic reconstruction. In: STOC06: Proceedings of the 38th Annual ACM Symposium on Theory of Computing. pp. 159-168.

171. Daskalakis C, Hill C, Jaffe A, Mihaescu R, Mossel E, et al. (2006) Maximal accurate forests from distance matrices. In: RECOMB. pp. 281-295.

172. Mossel E (2007) Distorted metrics on trees and phylogenetic forests. IEEE/ACM Trans Comput Bio Bioinform 4: 108-116.

173. Gronau I, Moran S, Snir S (2008) Fast and reliable reconstruction of phylogenetic trees with very short edges. In: SODA (ACM/SIAM Symp. Disc. Alg). pp. 379-388.

174. Roch S (2008) Sequence-length requirement for distance-based phylogeny reconstruction: Breaking the polynomial barrier. In: FOCS (Foundations of Computer Science). pp. 729-738.

175. Daskalakis C, Mossel E, Roch S (2009) Phylogenies without branch bounds: Contracting the short, pruning the deep. In: RECOMB. pp. 451-465.

176. Lin Y, Rajan V, Moret B (2012) A metric for phylogenetic trees based on matching. IEEE/ACM Trans Comp Biol Bioinf 9: 1014–1022.

177. Rannala B, Huelsenbeck J, Yang Z, Nielsen R (1998) Taxon sampling and the accuracy of large phylogenies. Syst Biol 47: 702-710.

178. Robinson D, Foulds L (1981) Comparison of phylogenetic trees. Math Biosci 53: 131-147.

179. Huelsenbeck J, Hillis D (1993) Success of phylogenetic methods in the four-taxon case. Syst Biol 42: 247-265.

180. Hillis D (1998) Taxonomic sampling, phylogenetic accuracy, and investigator bias. Syst Biol 47: 3–8.

181. Nakhleh L, Moret B, Roshan U, St John K, Sun J, et al. (2002) The accuracy of fast phylogenetic methods for large datasets. In: Proc. 7th Pacific Symposium on BioComputing. World Scientific Pub, pp. 211-222.

182. Zwickl DJ, Hillis DM (2002) Increased taxon sampling greatly reduces phylogenetic error. Syst Biol 51: 588-598.

183. Pollock DD, Zwickl DJ, McGuire JA, Hillis DM (2002) Increased taxon sampling is advantageous for phylogenetic inference. Syst Biol 51: 664-671.

184. Wiens J (2006) Missing data and the design of phylogenetic analyses. J Biomed Inform 39: 36–42.

185. Lemmon A, Brown J, Stanger-Hall K, Lemmon E (2009) The effect of ambiguous data on phylogenetic estimates obtained by maximum-likelihood and Bayesian inference. Syst Biol 58: 130-145.

186. Wiens J, Morrill M (2011) Missing data in phylogenetic analysis: reconciling results from simulations and empirical data. Syst Biol 60: 719–731.

187. Simmons M (2012) Misleading results of likelihood-based phylogenetic analyses in the presence of missing data. Cladistics 28: 208–222.

188. Moret B, Roshan U, Warnow T (2002) Sequence-length requirements for phylogenetic methods. In: Guigo R, Gusfield D, editors, Proc. 2nd International Workshop on Algorithms in Bioinformatics, Lecture Notes in Computer Science (LNCS #2452). Springer Verlag, pp. 343–356.

189. Gascuel O (1997) BIONJ: an improved version of the NJ algorithm based on a simple model of sequence data. Mol Biol Evol 14: 685–695.

190. Bruno WJ, Socci ND, Halpern AL (2000) Weighted neighbor joining: A likelihood-based approach to distance-based phylogeny reconstruction. Mol Biol Evol 17: 189–197.

191. Wheeler T (2009) Large-scale neighbor-joining with NINJA. In: Proc. Workshop Algorithms in Bioinformatics (WABI). volume 5724, pp. 375–389.

192. Desper R, Gascuel O (2002) Fast and accurate phylogeny reconstruction algorithm based on the minimum-evolution principle. J Comput Biol 9: 687–705.

193. Price M, Dehal P, Arkin A (2009) FastTree: computing large minimum evolution trees with profiles instead of a distance matrix. Mol Biol Evol 7: 1641-50.

194. Brown D, Truszkowski J (2011) Towards a practical $O(n \log n)$ phylogeny algorithm. In: Proc. Workshop Algorithms in Bioinformatics (WABI). pp. 14–25.

195. Rice K, Warnow T (1997) Parsimony is hard to beat! In: Jiang T, Lee D, editors, Proceedings, Third Annual International Conference of Computing and Combinatorics (COCOON). pp. 124–133.

196. Hillis D, Huelsenbeck J, Swofford D (1994) Hobgoblin of phylogenetics. Nature 369: 363–364.

197. Swofford D (1996) PAUP*: Phylogenetic analysis using parsimony (and other methods), version 4.0. Sunderland, Mass.: Sinauer Assoc.

198. Roch S (2006) A short proof that phylogenetic tree reconstruction by maximum likelihood is hard. IEEE Trans Comput Biol Bioinf 3: 92–94.

199. Guindon S, Gascuel O (2003) A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood. Syst Biol 52: 696-704.

200. Zwickl D (2006) Genetic algorithm approaches for the phylogenetic analysis of large biological sequence datasets under the maximum likelihood criterion. Ph.D. thesis, The University of Texas at Austin.

201. Liu K, Linder C, Warnow T (2012) RAxML and FastTree: comparing two methods for large-scale maximum likelihood phylogeny estimation. PLoS-ONE 6: e27731.

202. Claesson MJ, Cusack S, O'Sullivan O, Greene-Diniz R, de Weerd H, et al. (2011) Composition, variability, and temporal stability of the intestinal microbiota of the elderly. Proceedings of the National Academy of Sciences 108: 4586-4591.

203. McDonald D, Price MN, Goodrich J, Nawrocki EP, DeSantis TZ, et al. (2012) An improved Greengenes taxonomy with explicit ranks for ecological and evolutionary analyses of bacteria and archaea. The ISME Journal 6: 610618.

204. Boussau B, Guoy M (2006) Efficient likelihood computations with non-reversible models of evolution. Syst Biol 55: 756–68.

205. Whelan S, Money D (2010) The prevalence of multifurcations in tree-space and their implications for tree-search. Mol Biol Evol 27: 2674-2677.

206. Whelan S, Money D (2012) Characterizing the phylogenetic tree-search problem. Systematic Biology 61: 228-239.

207. Ronquist F, Huelsenbeck J (2003) MrBayes 3: Bayesian phylogenetic inference under mixed models. Bioinf 19: 1572-1574.

208. Drummond A, Rambaut A (2007) BEAST: Bayesian evolutionary analysis by sampling trees. BMC Evol Biol 7: 214.

209. Lartillot N, Philippe H (2004) A Bayesian mixture model for across-site heterogeneities in the amino acid replacement process. Mol Biol Evol 21.

210. Foster P (2004) Modeling compositional heterogeneity. Syst Biol 53: 485–495.

211. Pagel M, Meade A (2004) A phylognetic mixture model for detecting pattern heterogeneity in gene sequence or character state data. Syst Biol 53: 571–581.

212. Huelsenbeck J, Ronquist R (2001) MrBayes: Bayesian inference of phylogeny. Bioinf 17: 754-755.

213. Ronquist F, Deans A (2010) Bayesian phylogenetics and its influence on insect systematics. Ann Rev Entomol 55: 189–206.

214. Huelsenbeck JP, Ronquist F, Nielsen R, Bollback JP (2001) Bayesian inference of phylogeny and its impact on evolutionary biology. Science 294: 2310–2314.

215. Holder M, Lewis P (2003) Phylogeny estimation: traditional and Bayesian approaches. Nature Reviews: Genetics 4: 275–284.

216. Lewis P, Holder M, Holsinger K (2005) Polytomies and Bayesian phylogenetic inference. Syst Biol 54: 241–253.

217. Ganapathy G, Ramachandran V, Warnow T (2004) On contract-and-refine-transformations between phylogenetic trees. In: ACM/SIAM Symposium on Discrete Algorithms (SODA'04). SIAM Press, pp. 893–902.

218. Ganapathy G, Ramachandran V, Warnow T (2003) Better hill-climbing seaches for parsimony. Proceedings of the Third International Workshop on Algorithms in Bioinformatics (WABI) : 245–258.

219. Bonet M, Steel M, Warnow T, Yooseph S (1999) Faster algorithms for solving parsimony and compatibility. J Comput Biol 5: 409–422.

220. Nixon KC (1999) The parsimony ratchet, a new method for rapid parsimony analysis. Cladistics 15: 407-414.

221. Vos R (2003) Accelerated likelihood surface exploration: the Likelihood Ratchet. Systematic Biology 52: 368–373.

222. Warnow T, Moret BME, St John K (2001) Absolute phylogeny: true trees from short sequences. In: Proc. 12th Ann. ACM/SIAM Symp. on Discr. Algs. SODA01. SIAM Press, pp. 186–195.

223. Nakhleh L, Roshan U, St John K, Sun J, Warnow T (2001) Designing fast converging phylogenetic methods. Bioinf 17: 190–198.

224. Warnow T (2005) Large-scale phylogenetic reconstruction. In: Aluru S, editor, Handbook of Computational Molecular Biology, CRC Press, volume 9 of *Chapman and Hall/CRC Computer and Information Science Series.*

225. Roshan U, Moret B, Williams T, Warnow T (2004) Rec-I-DCM3: A fast algorithmic technique for reconstructing large phylogenetic trees. In: Proc. 3rd Computational Systems Biology Conf. (CSB'05). Proceedings of the IEEE, pp. 98–109.

226. Steel M (1994) The maximum likelihood point for a phylogenetic tree is not unique. Syst Biol 43: 560–564.

227. Blair C, Murphy R (2011) Recent trends in molecular phylogenetic analysis: Where to next? J Heredity 102: 130–138.

228. Nagy L, Kocsube S, Csanadi Z, Kovacs G, Petkovits T, et al. (2012) Re-mind the gap! insertion and deletion data reveal neglected phylogenetic potential of the nuclear ribosomal internal transcribed spacer (its) of fungi. PLoS ONE 7: e49794.

229. Barriel V (1994) Molecular phylogenies and nucleotide insertion-deletions. CR Acad Sci III 7: 693–701.

230. Young N, Healy J (2003) GapCoder automates the use of indel characters in phylogenetic analysis. BMC Bioinf 4.

231. Muller K (2006) Incorporating information from length-mutational events into phylogenetic analysis. Mol Phylog Evol 38: 667-676.

232. Ogden T, Rosenberg M (2007) How should gaps be treated in parsimony? a comparison of approaches using simulation. Mol Phylog Evol 42: 817–26.

233. Dwivedi B, Gadagkar S (2009) Phylogenetic inference under varying proportions of indel-induced alignment gaps. BMC Evol Biol 9: 211.

234. Dessimoz C, Gil M (2010) Phylogenetic assessment of alignments reveals neglected tree signal in gaps. Genome Biol 11: R37.

235. Yuri T, Kimball RT, Harshman J, Bowie RCK, Braun MJ, et al. (2013) Biology 2: 419-444.

236. Warnow T (2012) Standard maximum likelihood analyses of alignments with gaps can be statistically inconsistent. PLoS Currents Tree of Life.

237. Daskalakis C, Roch S (2010) Alignment-free phylogenetic reconstruction. In: Berger B, editor, Proc. RECOMB 2010, Springer Berlin / Heidelberg, volume 6044 of *Lecture Notes in Computer Science*. pp. 123-137. URL http://dx.doi.org/10.1007/978-3-642-12683-3_9.

238. Thatte B (2006) Invertibility of the TKF model of sequence evolution. Math Biosci 200: 58-75.

239. Hartmann S, Vision T (2008) Using ESTs for phylogenomics: can one accurately infer a phylogenetic tree from a gappy alignment? BMC Evol Biol 8: 95.

240. Mirarab S, Nguyen N, Warnow T (2012) SEPP: SATé-enabled phylogenetic placement. In: Pacific Symposium on Biocomputing. pp. 247–58.

241. Matsen FA, Kodner RB, Armbrust EV (2010) pplacer: linear time maximum-likelihood and Bayesian phylogenetic placement of sequences onto a fixed reference tree. BMC Bioinf 11: 538.

242. Berger SA, Krompass D, Stamatakis A (2011) Performance, accuracy, and web server for evolutionary placement of short sequence reads under maximum likelihood. Syst Biol 60: 291-302.

243. Eddy S (2009) A new generation of homology search tools based on probabilistic inference. Genome Inform 23: 205211.

244. Finn R, Clements J, Eddy S (2011) HMMER web server: interactive sequence similarity searching. Nucleic Acids Research 39: W29-W37.

245. Brown DG, Truskowski J (2013) LSHPlace: fast phylogenetic placement using locality-sensitive hashing. In: Pacific Symposium on Biocomputing. volume 18, pp. 310–319.

246. Stark M, Berger S, Stamatakis A, von Mering C (2010) MLTreeMap - accurate Maximum Likelihood placement of environmental DNA sequences into taxonomic and functional reference phylogenies. BMC Genomics 11: 461.

247. Droge J, McHardy A (2012) Taxonomic binning of metagenome samples generated by next-generation sequencing technologies. Brief Bioinform .

248. Giribet G (2001) Exploring the behavior of POY, a program for direct optimization of molecular data. Cladistics 17: S60–S70.

249. Hartigan J (1973) Minimum mutation fits to a given tree. Biometrics 29: 53-65.

250. Sankoff D (1975) Minimal mutation trees of sequences. SIAM J Appl Math 28: 35 – 42.

251. Sankoff D, Cedergren RJ (1993) Simultaneous comparison of three or more sequences related by a tree. In: Sankoff D, Kruskall JB, editors, Time Warps, String Edits and Macromolecules: the Theory and Practice of Sequence Comparison, New York: Addison Wesley. pp. 253–263.

252. Wang L, Jiang T (1994) On the complexity of multiple sequence alignment. J Comp Biol 1: 337–348.

253. Wang L, Jiang T, Lawler E (1996) Approximation algorithms for tree alignment with a given phylogeny. Algorithmica 16: 302–315.

254. Wang L, Gusfield D (1997) Improved approximation algorithms for tree alignment. J Algorithms 25(2): 255–273.

255. Wang L, Jiang T, Gusfield D (2000) A more efficient approximation scheme for tree alignment. SIAM J Comput 30(1): 283–299.

256. Liu K, Warnow T (2012) Treelength optimization for phylogeny estimation. PLoS One 7: e33104.

257. Varón A, Vinh L, Bomash I, Wheeler W (2007). POY Software. Documentation by A. Varon, L.S. Vinh, I. Bomash, W. Wheeler, K. Pickett, I. Temkin, J. Faivovich, T. Grant, and W.L. Smith. Available for download at `http://research.amnh.org/scicomp/projects/poy.php`.

258. Kjer K, Gillespie J, Ober K (2007) Opinions on multiple sequence alignment, and an empirical comparison on repeatability and accuracy between POY and structural alignment. Syst Biol 56: 133-146.

259. Ogden TH, Rosenberg M (2007) Alignment and topological accuracy of the direct optimization approach via POY and traditional phylogenetics via ClustalW + PAUP*. Syst Biol 56: 182-193.

260. Yoshizawa K (2010) Direct optimization overly optimizes data. Syst Ent 35: 199–206.

261. Wheeler W, Giribet G (2009) Phylogenetic hypotheses and the utility of multiple sequence alignment. In: Rosenberg M, editor, Sequence alignment: methods, models, concepts and strategies, University of California Press. pp. 95-104.

262. Lehtonen S (2008) Phylogeny estimation and alignment via POY versus Clustal + PAUP*: A response to Ogden and Rosenberg. Syst Biol 57: 653-657.

263. Liu K, Nelesen S, Raghavan S, Linder C, Warnow T (2009) Barking up the wrong treelength: the impact of gap penalty on alignment and tree accuracy. IEEE Trans Comp Biol Bioinf 6: 7–21.

264. Gu X, Li WH (1995) The size distribution of insertions and deletions in human and rodent pseudogenes suggests the logarithmic gap penalty for sequence alignment. J Mol Evol 40: 464-473.

265. Altschul SF (1998) Generalized affine gap costs for protein sequence alignment. Proteins: Structure, Function and Genomics 32: 88–96.

266. Gill O, Zhou Y, Mishra B (2004) Aligning sequences with non-affine gap penalty: PLAINS algorithm, a practical implementation, and its biological applications in comparative genomics. Proc ICBA 2004.

267. Qian B, Goldstein R (2001) Distribution of indel lengths. Proteins 45: 102–104.

268. Chang M, Benner S (2004) Empirical analysis of protein insertions and deletions determining parameters for the correct placement of gaps in protein sequence alignments. J Mol Biol 341: 617–631.

269. Thorne JL, Kishino H, Felsenstein J (1991) An evolutionary model for maximum likeliihood alignment of DNA sequences. J Mol Evol 33: 114–124.

270. Thorne JL, Kishino H, Felsenstein J (1992) Inching toward reality: an improved likelihood model of sequence evolution. J Mol Evol 34: 3–16.

271. Thorne JL, Kishino H, Felsenstein J (1992) Erratum, an evolutionary model for maximum likeliihood alignment of DNA sequences. J Mol Evol 34: 91–92.

272. Rivas E (2005) Evolutionary models for insertions and deletions in a probabilistic modeling framework. BMC Bioinf 6: 30.

273. Rivas E, Eddy S (2008) Probabilistic phylogenetic inference with insertions and deletions. PLoS Comput Biol 4: e1000172.

274. Holmes I, Bruno WJ (2001) Evolutionary HMMs: a Bayesian approach to multiple alignment. Bioinf 17: 803–820.

275. Miklós I, Lunter GA, Holmes I (2004) A “long indel model” for evolutionary sequence alignment. Mol Biol Evol 21: 529-540.

276. Redelings B, Suchard M (2005) Joint Bayesian estimation of alignment and phylogeny. Syst Biol 54: 401-418.

277. Suchard MA, Redelings BD (2006) BAli-Phy: simultaneous Bayesian inference of alignment and phylogeny. Bioinf 22: 2047–2048.

278. Redelings B, Suchard M (2007) Incorporating indel information into phylogeny estimation for rapidly emerging pathogens. BMC Evol Biol 7: 40.

279. Fleissner R, Metzler D, von Haeseler A (2005) Simultaneous statistical multiple alignment and phylogeny reconstruction. Syst Biol 54: 548–561.

280. Novák A, Miklós I, Lyngso R, Hein J (2008) StatAlign: an extendable software package for joint Bayesian estimation of alignments and evolutionary trees. Bioinf 24: 2403-2404.

281. Lunter GA, Miklos I, Song YS, Hein J (2003) An efficient algorithm for statistical multiple alignment on arbitrary phylogenetic trees. J Comp Biol 10: 869-89.

282. Lunter G, Miklós I, Drummond A, Jensen JL, Hein J (2003) Bayesian phylogenetic inference under a statistical indel model. In: Benson G, Page R, editors, Lecture Notes in Bioinformatics: Third International Workshop, WABI 2003, LNBI. Berlin: Springer-Verlag, volume 2812, pp. 228–244.

283. Lunter G, Drummond A, Miklós I, Hein J (2005) Statistical alignment: Recent progress, new applications, and challenges. In: Nielsen R, editor, Statistical Methods in Molecular Evolution (Statistics for Biology and Health). Berlin: Springer, pp. 375-406.

284. Metzler D (2003) Statistical alignment based on fragment insertion and deletion models. Bioinf 19: 490-499.

285. Miklós I (2003) Algorithm for statistical alignment of sequences derived from a Poisson sequence length distribution. Disc Appl Math 127: 79-84.

286. Arunapuram P, Edvardsson I, Golden M, Anderson J, Novak A, et al. (2013) StatAlign 2.0: Combining statistical alignment with RNA secondary structure prediction. Bioinf 29(5): 654–655.

287. Lunter G, Miklós I, Drummond A, Jensen JL, Hein J (2005) Bayesian coestimation of phylogeny and sequence alignment. BMC Bioinf 6: 83.

288. Bouchard-Côté A, Jordan MI (2013) Evolutionary inference via the poisson indel process. Proceedings of the National Academy of Sciences 110: 1160-1166.

289. Brown D, Krishnamurthy N, Sjolander K (2007) Automated protein subfamily identification and classification. PLoS Comp Biol 3: e160.

290. Vinga S, Almeida J (2003) Alignment-free sequence comparison–a review. Bioinf 19: 513-523.

291. Chan C, Ragan M (2013) Next-generation phylogenomics. Biology Direct 8.

292. Blaisdell B (1986) A measure of the similarity of sets of sequences not requiring sequence alignment. Proc National Acad Sci (USA) 83: 5155–5159.

293. Sims G, Jun SR, Wu G, Kim SH (2009) Alignment-free genome comparison with feature frequency profiles (FFP) and optimal resolutions. Proc National Acad Sci (USA) 106: 2677-2682.

294. Jun SR, Sims G, Wu G, Kim SH (2010) Whole-proteome phylogeny of prokaryotes by feature frequency profiles: An alignment-free method with optimal feature resolution. Proc National Acad Sci (USA) 107: 133-138.

295. Liu X, Wan L, Li J, Reinert G, Waterman M, et al. (2011) New powerful statistics for alignment-free sequence comparison under a pattern transfer model. J Theoretical Biology 284: 106–116.

296. Yang K, Zhang L (2008) Performance comparison between k-tuple distance and four model-based distances in phylogenetic tree reconstruction. Nucleic Acids Res 36: e33.

297. Roshan U, Moret BME, Williams TL, Warnow T (2004) Performance of supertree methods on various dataset decompositions. In: Bininda-Emonds ORP, editor, Phylogenetic Supertrees: combining information to reveal The Tree Of Life. Kluwer Academic, Dordrecht, the Netherlands, pp. 301-328.

298. Nelesen S (2009) Improved Methods for Phylogenetics. Ph.D. thesis, The University of Texas at Austin.

299. Swenson M (2008) Phylogenetic Supertree Methods. Ph.D. thesis, The University of Texas at Austin.

300. Neves D, Warnow T, Sobral J, Pingali K (2012) Parallelizing SuperFine. In: 27th Symposium on Applied Computing (ACM-SAC).

301. Cannone J, Subramanian S, Schnare M, Collett J, D'Souza L, et al. (2002) The Comparative RNA Web (CRW) Site: An Online Database of Comparative Sequence and Structure Information for Ribosomal, Intron and Other RNAs. BMC Bioinf 3.

302. Roch S (2010) Towards extracting all phylogenetic information from matrices of evolutionary distances. Science 327: 1376–1379.

303. Darling A, Mau B, Blatter F, Perna N (2004) Mauve: multiple alignment of conserved genomic sequence with rearrangements. Genome Res 14: 1394–1403.

304. Darling A, Mau B, Perna N (2010) progressiveMauve: multiple genome alignment with gene gain, loss and rearrangement. PLoS One 5: e11147.

305. Raphael B, Zhi D, Tang H, Pevzner P (2004) A novel method for multiple alignment of sequences with repeated and shuffled elements. Genome Res 14: 2336-46.

306. Dubchak I, Poliakov A, Kislyuk A, Brudno M (2009) Multiple whole-genome alignments without a reference organism. Genome Res 19: 682-689.

307. Brudno M, Do C, Cooper G, Kim M, Davydov E, et al. (2003) LAGAN and Multi-LAGAN: efficient tools for large-scale multiple alignment of genomic DNA. Genome Res 13: 721–731.

308. Phuong T, Do C, Edgar R, Batzoglou S (2006) Multiple alignment of protein sequences with repeats and rearrangements. Nucleic Acids Res 34: 5932-42.

309. Paten B, Earl D, Nguyen N, Diekhans M, Zerbino D, et al. (2011) Cactus: Algorithms for genome multiple sequence alignment. Genome Res 21: 15121528.

310. Angiuoli S, Salzberg S (2011) Mugsy: fast multiple alignment of closely related whole genomes. Bioinf doi:10.1093/bioinformatics/btq665.

311. Agren J, Sundstrom A, Hafstrom T, Segerman B (2012) Gegenees: Fragmented alignment of multiple genomes for determining phylogenomic distances and genetic signatures unique for specified target groups. PLoS ONE 7: e39107.

312. Gogarten J, Doolittle W, Lawrence J (2002) Prokaryotic evolution in light of gene transfer. Mol Biol Evol 19: 2226-2238.

313. Gogarten J, Townsend J (2005) Horizontal gene transfer, genome innovation and evolution. Nature Reviews Microbiology 3: 679–687.

314. Bergthorsson U, Richardson A, Young G, Goertzen L, Palmer J (2004) Massive horizontal transfer of mitochondrial genes from diverse land plant donors to basal angiosperm Amborella. Proc National Acad Sci, USA 101: 17,747-17,752.

315. Bergthorsson U, Adams K, Thomason B, Palmer J (2003) Widespread horizontal transfer of mitochondrial genes in flowering plants. Nature 424: 197-201.

316. Wolf Y, Rogozin I, Grishin N, Koonin E (2002) Genome trees and the tree of life. TRENDS in Genetics 18: 472–478.

317. Koonin E, Makarova K, Aravind L (2001) Horizontal gene transfer in prokaryotes: quantification and classification. Annual Reviews in Microbiology 55: 709–742.

318. Linder C, Rieseberg L (2004) Reconstructing patterns of reticulate evolution in plants. American J Botany 91: 1700–1708.

319. Sessa E, Zimmer E, Givnish T (2012) Reticulate evolution on a global scale: A nuclear phylogeny for New World *Dryopteris* (Dryopteridaceae). Molecular Phylogenetics and Evolution 64: 563–581.

320. Moody M, Rieseberg L (2012) Sorting through the chaff, nDNA gene trees for phylogenetic inference and hybrid identification of annual sunflowers *Helianthus* (sect. *Helianthus*). Molecular Phylogenetics and Evolution 64: 145–155.

321. Mindell D (2013) The Tree of Life: metaphor, model, and heuristic device. Syst Biol 62(3): 479–489.

322. Warnow T, Evans S, Ringe D, Nakhleh L (2006) A stochastic model of language evolution that incorporates homoplasy and borrowing. In: Phylogenetic Methods and the Prehistory of Languages, Cambridge University Press. pp. 75–90.

323. Nakhleh L, Ringe DA, Warnow T (2005) Perfect phylogenetic networks: A new methodology for reconstructing the evolutionary history of natural languages. Language 81: 382-420.

324. Huson D, Rupp R, Scornovacca C (2010) Phylogenetic Networks: concepts, algorithms and applications. Cambridge University Press.

325. Morrison D (2011) Introduction to Phylogenetic Networks. RJR Productions, Uppsala, Sweden.

326. Nakhleh L (2011) Evolutionary phylogenetic networks: Models and issues. In: Problem Solving Handbook in Computational Biology and Bioinformatics, Springer. pp. 125–158.

327. van Iersel L, Kelk S, Rupp R, Huson D (2010) Phylogenetic networks do not need to be complex: Using fewer reticulations to represent conflicting clusters. Bioinf 26: i124–i131.

328. Wu Y (2013) An algorithm for constructing parsimonious hybridization networks with multiple phylogenetic trees. In: Proc. RECOMB.

329. Jin G, Nakhleh L, Snir S, Tuller T (2006) Maximum likelihood of phylogenetic networks. Bioinf 22: 2604–2611.

330. Jin G, Nakhleh L, Snir S, Tuller T (2007) Inferring phylogenetic networks by the maximum parsimony criterion: A case study. Mol Biol Evol 24: 324–337.

331. Nakhleh L, Warnow T, Linder C (2004) Reconstructing reticulate evolution in species—theory and practice. In: Proc. 8th Conf. Comput. Mol. Biol. (RECOMB'04). ACM Press, pp. 337–346.

332. Nakhleh L, Ruths D, Wang LS (2005) RIATA-HGT: A fast and accurate heuristic for reconstructing horizontal gene transfer. In: Proc. 11th Conf. Computing and Combinatorics (COCOON'05). Springer-Verlag, Lecture Notes in Computer Science.

333. Yu Y, Than C, Degnan J, Nakhleh L (2011) Coalescent histories on phylogenetic networks and detection of hybridization despite incomplete lineage sorting. Syst Biol 60: 138–149.

334. Lapierre P, Lasek-Nesselquist E, Gogarten J (2012) The impact of HGT on phylogenomic reconstruction methods. Brief Bioinform doi:10.1093/bib/bbs050.

335. Roch S, Snir S (2012) Recovering the tree-like trend of evolution despite extensive lateral genetic transfer: a probabilistic analysis. In: Proceedings RECOMB 2012.

336. Gerard D, Gibbs H, Kubatko L (2011) Estimating hybridization in the presence of coalescence using phylogenetic intraspecific sampling. BMC Evol Biol 11: 291.

337. Yu Y, Degnan J, Nakhleh L (2012) The probability of a gene tree topology within a phylogenetic network with applications to hybridization detection. PLoS Genetics 8: e1002660.

338. Chowdhury R, Ramachandran V (2006) Cache-oblivious dynamic programming. In: Proc. ACM-SIAM Symposium on Discrete Algorithms (SODA). pp. 591–600.