JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 29, Number 0, 2022

Pp. 1–20
DOI: 10.1089/cmb.2021.0585

Waterman Special Issue

WITCH: Improved Multiple Sequence Alignment Through Weighted Consensus Hidden Markov Model Alignment

CHENGZE SHEN, MINHYUK PARK, and TANDY WARNOW

ABSTRACT

Accurate multiple sequence alignment is challenging on many data sets, including those that are large, evolve under high rates of evolution, or have sequence length heterogeneity. While substantial progress has been made over the last decade in addressing the first two challenges, sequence length heterogeneity remains a significant issue for many data sets. Sequence length heterogeneity occurs for biological and technological reasons, including large insertions or deletions (indels) that occurred in the evolutionary history relating the sequences, or the inclusion of sequences that are not fully assembled. Ultra-large alignments using Phylogeny-Aware Profiles (UPP) (Nguyen et al. 2015) is one of the most accurate approaches for aligning data sets that exhibit sequence length heterogeneity: it constructs an alignment on the subset of sequences it considers ''full-length,'' represents this ''backbone alignment'' using an ensemble of hidden Markov models (HMMs), and then adds each remaining sequence into the backbone alignment based on an HMM selected for that sequence from the ensemble. Our new method, WeIghTed Consensus Hmm alignment (WITCH), improves on UPP in three important ways: first, it uses a statistically principled technique to weight and rank the HMMs; second, it uses $k > 1$ HMMs from the ensemble rather than a single HMM; and third, it combines the alignments for each of the selected HMMs using a consensus algorithm that takes the weights into account. We show that this approach provides improved alignment accuracy compared with UPP and other leading alignment methods, as well as improved accuracy for maximum likelihood trees based on these alignments.

Keywords: divide and conquer, hidden Markov model, multiple sequence alignment.

Department of Computer Science, University of Illinois Urbana-Champaign, Urbana, Illinois, USA.

1. INTRODUCTION

Multiple sequence alignment is a necessary precursor for many problems in biology, including phylogeny estimation (Morrison and Ellis, 1997; Ogden and Rosenberg, 2006), protein family classification (Diplaris et al., 2005; Brown et al., 2007; Schwacke et al., 2019), and detection of selection (Fletcher and Yang, 2010; Jordan and Goldman, 2012). However, accurate estimation of multiple sequence alignments is challenging under many conditions, including large numbers of sequences and high rates of evolution.

There are now several methods that provide high accuracy even for very large data sets with 10,000 or more sequences (e.g., Katoh and Toh, 2008; Mirarab et al. 2015; Nguyen et al. 2015; Sievers and Higgins 2018; Smirnov and Warnow 2021a), but only a few have high accuracy under high rates of evolution. However, another issue that creates difficulties for alignment is sequence length heterogeneity, which can arise for many different reasons, such as the inclusion of reads or partially assembled sequences, or more simply through evolutionary processes that include large indels. As seen in Figure 1, sequence length heterogeneity is present in biological data sets (Cannone et al., 2002).

Studies evaluating alignment methods under conditions with many short sequences have shown that many otherwise excellent alignment methods can degrade in accuracy substantially under those conditions (Nguyen et al., 2015), and trees estimated on these alignments can also be poor (Smirnov and Warnow, 2021b). Thus, the alignment of large data sets with sequence length heterogeneity is an interesting and important bioinformatic challenge.

A natural approach to aligning data sets with sequence length heterogeneity uses two stages: the first stage selects and aligns sequences that are considered full-length, and the second stage adds the remaining sequences into the alignment computed in the first phase. This approach allows methods that have high accuracy on data sets that do not have substantial sequence length heterogeneity to be used in the first stage.

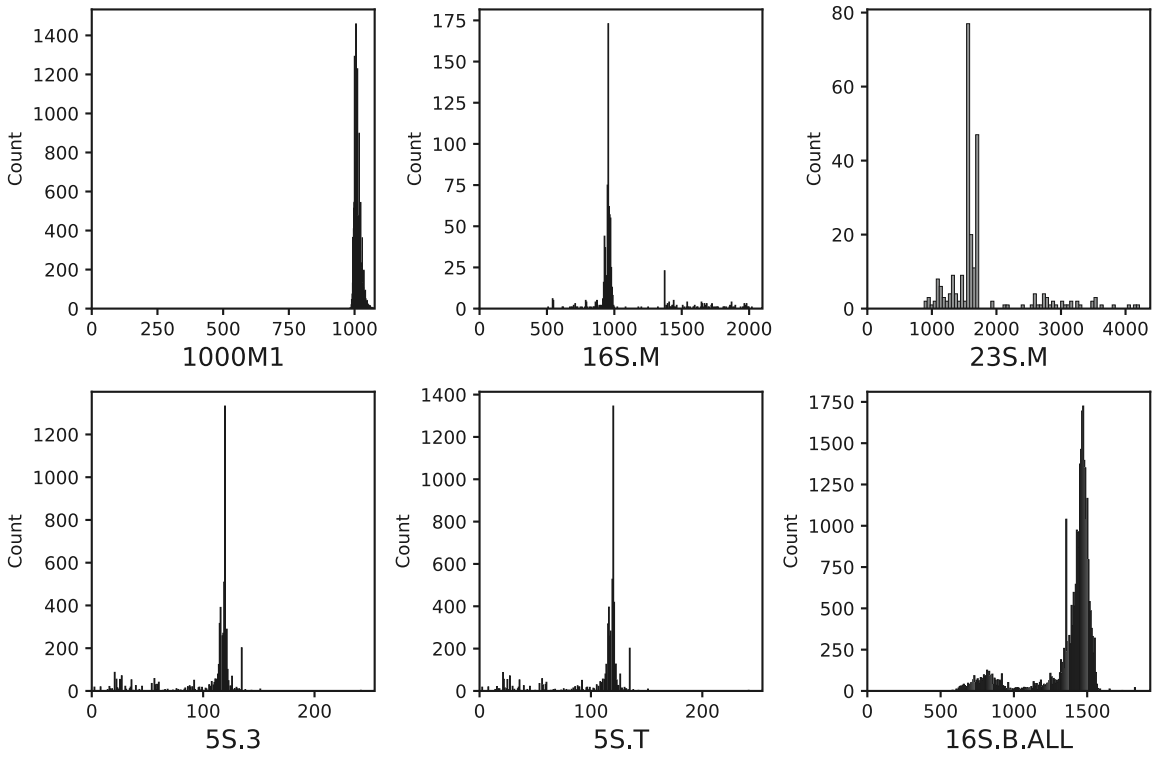

FIG. 1. Sequence length histograms of five biological data sets and one simulated data set (1000M1) from Liu et al. (2009). The biological data sets (16S.M, 23S.M, 5S.3, 5S.T, and 16S.B.ALL) are from the Comparative Ribosomal Website (Cannone et al., 2002). Note that the simulated data set shows essentially no sequence length heterogeneity, while each of the other data sets has many shorter sequences, and some have longer sequences as well.

The second stage is performed using methods designed for adding sequences into alignments, and so can be specifically designed to address sequence length heterogeneity.

Ultra-large alignments using Phylogeny-Aware Profiles (UPP) is a method that uses this two-stage approach (Nguyen et al., 2015). UPP performs the second stage by representing the backbone alignment with an ensemble of hidden Markov models (HMMs) [see Brown et al. (1993); Haussler et al. (1993); Krogh et al. (1994) for the earliest articles on HMMs in bioinformatics, and Durbin et al. (1998) for a standard textbook about HMMs in molecular sequence analysis]. Then, to add a given sequence into the backbone alignment, the HMM in the ensemble of HMMs (eHMM) with the best bitscore is selected and HMMAlign [one of the methods in HMMER (Finn et al., 2011)] is used to add the sequence into the backbone alignment (i.e., to compute an ''extended alignment'').

Finally, UPP uses transitivity to merge the extended alignments. UPP can run on very large data sets (even up to 1 million sequences), and was shown to have better accuracy than the alignment methods Muscle (Edgar, 2004), MAFFT (Katoh and Standley, 2013), Clustal-Omega (Sievers et al., 2011), and PASTA (Practical Alignments using SATé and TrAnsitivity) (Mirarab et al., 2015), when aligning data sets with large numbers of fragmentary sequences (Nguyen et al., 2015).

In this study, we present a novel method for aligning data sets with sequence length heterogeneity. We follow the same basic two-stage method that is used by UPP, but we change how sequences are added into the backbone alignment. An important component of this method is a statistically rigorous technique that we derive for computing the probability that a given HMM generates a given sequence, and we use this technique to weight each HMM-query pair. Then, to add a sequence x into the backbone alignment, we select the top k HMMs (ranked by their weights); each such HMM defines a single extended alignment that induces the backbone alignment and includes x, and each such extended alignment is weighted by the probability that the selected HMM generates x.

Finally, we compute the weighted consensus of these query alignments using a graph clustering method that is a modified version of Graph Clustering Merger (GCM), a method designed for use in MAGUS (Multiple sequence Alignment using Graph clUStering) (Smirnov and Warnow, 2021a).

We refer to the method as WeIghTed Consensus Hmm alignment (WITCH). We benchmark WITCH on a collection of simulated and biological data sets and compare it with PASTA (Mirarab et al., 2015), UPP (Nguyen et al., 2015), MAGUS (Smirnov and Warnow, 2021a), MAFFT (Katoh and Standley, 2013), and Clustal-Omega (Sievers and Higgins, 2018). Our study shows that WITCH produces the best alignment accuracy of these methods on data sets with high levels of fragmentation and matches or improves on the other methods under low fragmentation conditions. In addition, trees estimated on WITCH alignments match or improve accuracy compared with other methods under conditions with fragmentation.

2. THE WITCH ALGORITHM

2.1. Overview

The input to WITCH is a set S of unaligned sequences and the output is a multiple sequence alignment. WITCH follows the standard two-stage strategy as UPP (Nguyen et al., 2015), but differs in the details in ways that enable it to be more accurate, although at an increase in the running time (Supplementary Fig. S4). WITCH has two algorithmic parameters: a number z that impacts the construction of the ensemble of HMMs and a number k that impacts how each query sequence is added to the backbone alignment. In this study, we present the five phases of WITCH at a very high level, and provide details in the relevant subsections below.

- Phase 0: A subset S_0 of the input set S of unaligned sequences is selected as the ''backbone sequences,'' and an alignment and tree are computed on these backbone sequences [default: MAGUS (Smirnov and Warnow, 2021a)]. The remaining sequences are referred to as ''query sequences.''
- Phase 1: An ensemble of HMMs is computed on the backbone alignment.
- Phase 2: For every query sequence q, a ''weight'' is computed for every HMM in the ensemble, reflecting the fit between the HMM and q.
- Phase 3: For every query sequence q, the top k HMMs are selected and then used to create an ''extended alignment'' on $\{q\} \cup S_0$.
- Phase 4: The extended alignments are merged together into an alignment on the full data set using transitivity.

Phases 1–4 are illustrated in Figure 2, and Phase 3 is illustrated with additional detail in Figure 3.

Some comments about these phases should be helpful in explaining the techniques. First, Phase 0 is identical to how it is performed in UPP, with the use of MAGUS instead of PASTA for constructing the backbone alignment based on the recent study that showed this substitution improved accuracy (Shen et al., 2022b). Phase 1 is identical to how it is performed in UPP. Phase 2 is unique to WITCH and is based on a theoretical derivation provided in Section 2.4. Phase 3 is also unique to WITCH but builds on a technique used in MAGUS called the ''Graph Clustering Merger.'' Phase 4 is the same as used in UPP and is a simple use of transitivity of pairwise homologies.

Thus, Phases 2 and 3 are the most important innovations in WITCH compared with prior methods, but all five phases contribute to the final results. In the subsections that follow, we describe each phase in detail.

2.2. *Phase 0: backbone alignment and tree construction*

In this phase, a subset of the sequences is extracted to form the ''backbone sequences'' and ''backbone tree.'' We use the same technique as in UPP for this step, which we now describe. The set of backbone sequences can be indicated by the user, or else they can be selected based on their length: the median length of the unaligned sequences is computed, and those sequences that are within 25% of the median length of the sequences in the input are considered ''full-length.'' Then, 1000 sequences are randomly selected from the full-length sequences, which we denote as the backbone sequences (but if there are fewer than 1000 full-length sequences, all the sequences are selected). A MAGUS alignment is computed on the backbone sequences and a maximum likelihood tree is computed on the backbone alignment using FastTree2 (Price et al., 2010).

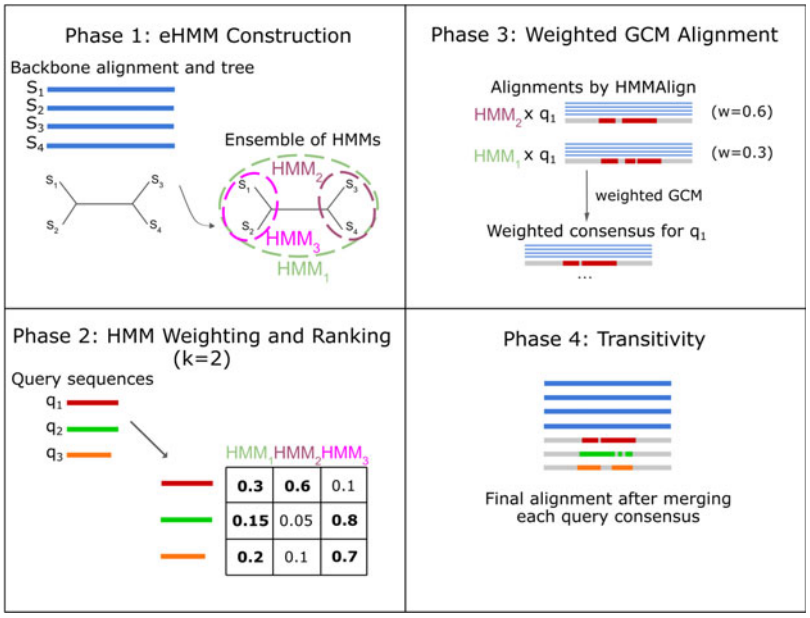

FIG. 2. Overview of the WITCH algorithm. The input is a set of unaligned sequences and values for k and z. Phase 0: The input sequences are split into two sets: the backbone sequences and the query sequences and an alignment and tree are built on the backbone sequences. Phase 1: An eHMM is built on the backbone alignment, stopping the decomposition when the subtrees have at most z leaves (default: $z=10$). Phase 2: For every query sequence, the HMMs in the eHMM are weighted and ranked, and the top k HMMs are selected (we illustrate this with $k=2$). Phase 3: For every query sequence and for each of its k selected HMM, an extended alignment (containing the backbone sequences and the query sequence) is computed using HMMAlign (Finn et al., 2011). A weighted consensus of these extended alignments is computed using a weighted version of the GCM (Smirnov and Warnow, 2021a). Phase 4: The final alignment is obtained by transitively merging all consensus alignments. Phases 1 and 4 are identical to the corresponding phases in UPP, and Phases 2 and 3 are extensions of techniques from UPP (Nguyen et al., 2015) and MAGUS (Smirnov and Warnow, 2021a). eHMM, ensemble of hidden Markov models; GCM, Graph Clustering Merger; MAGUS, Multiple sequence Alignment using Graph clUStering; WITCH, WeIghTed Consensus Hmm alignment.

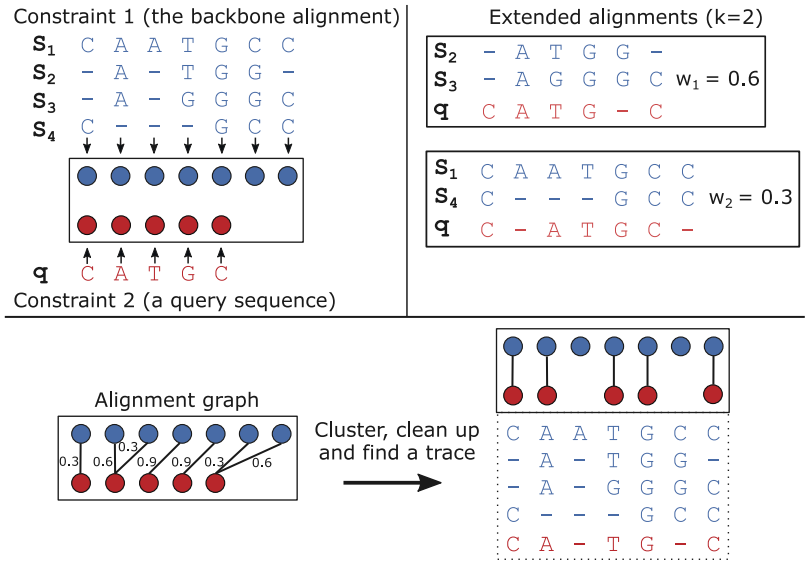

FIG. 3. An example of the weighted GCM alignment phase. We have two constraints, one the backbone alignment A and the other a query sequence q. Here we show the case of having two extended alignments (i.e., $k=2$) obtained by aligning q to the top two HMMs. An alignment graph is formed in the same way as GCM except that each extended alignment contributes to edge weights differently. The remaining steps are the same as the GCM algorithm (i.e., clustering, cleaning, and finding a trace).

2.3. Phase 1: eHMM construction

Given the backbone alignment A and its maximum likelihood tree T, an eHMM is constructed using the same procedure as in UPP. UPP decomposes the backbone alignment A into subsets, and takes as input a parameter z that specifies how small the subsets of sequences need to be before stopping the decomposition. UPP produces the decomposition using T: it repeatedly finds and removes a ''centroid edge'' (i.e., an edge that produces as balanced a split as possible) until all the parts have at most z leaves. This decomposition produces a set of disjoint subtrees, each with at most z leaves, and so also a set of disjoint subsets of sequences and the set of subset alignments induced by A.

The collection of subset alignments that UPP produces is hierarchical: it includes the alignment A on the full set S of sequences, as well as the alignment induced by A on every subset that is produced during the decomposition strategy. For example, if only one edge is deleted in constructing the ensemble, then UPP will produce three alignments: the alignment on the full set, and the alignment induced on each of the two subsets produced by deleting that edge. In particular, each edge deletion adds two alignments to the set.

For each alignment in the set, we build a profile HMM using HMMBuild from HMMER (Finn et al., 2011). The collection of HMMs is the eHMMs that represent the backbone alignment A.

2.4. Phase 2: HMM weighting and ranking

The input to Phase 2 is an ensemble of HMMs representing the backbone alignment and the set of query sequences (which are not part of the backbone alignment). The output of this phase is a selection of k HMMs for each query sequence, along with the weighting of each HMM-query pair.

The main innovation in Phase 2 is that the weighting we provide for a given HMM-query pair is designed to equal the probability that the given HMM generates the given query sequence. This enables us to rigorously select and then use $k \geq 1$ HMMs for each query sequence.

The weighting formula is an extension of the alignment support calculation provided in Nguyen et al. (2014). Given a query sequence q and HMM H_i, HMMSearch calculates the corresponding bitscore $BS(H_i, q)$ by

$$BS(H_i, q) = \log_2 \frac{P(q|H_i)}{P(q|H_{null})} \tag{1}$$

where $P(q|H_i)$ is the probability that H_i generates q, and $P(q|H_{null})$ is the probability that a random (null) HMM generates q. Assuming that q is generated by exactly one of the HMMs and that there are d HMMs, we can express the probability $P(H_i|q)$ that H_i generates q by

$$P(H_i|q) = \frac{P(q|H_i)P(H_i)}{\sum_{j=1}^{d} P(q|H_j)P(H_j)} \tag{2}$$

where we assume that the prior probability $P(H_i)$ is only affected by the number of sequences in the alignment used to construct HMM H_i. For the case where every two HMMs in the ensemble are constructed on disjoint sets of sequences, the prior probability $P(H_i)$ is

$$P(H_i) = \frac{s_i}{S} \tag{3}$$

where s_i is the number of sequences in the alignment used to build H_i and $S = \sum_{j=1}^{d} s_j$ is the total number of unique sequences across all these alignment subsets. For the case where the HMMs are based on overlapping sets of sequences (e.g., eHMMs in this study), the prior $P(H_i)$ is

$$P(H_i) = \frac{s_i}{S^*} \tag{4}$$

where $S^* = \sum_{j=1}^{d} s_j$ is the sum of total appearances of sequences in the HMMs.

Therefore, using Equation (4) we can reduce Equation (2) to

$$P(H_i|q) = \frac{1}{\sum_{j=1}^{d} 2^{\log_2 \frac{P(q|H_j)s_j}{P(q|H_i)s_i}}} \tag{5}$$

By Equation (1), we have

$$\begin{aligned} BS(H_j, q) - BS(H_i, q) &= \log_2 \frac{P(q|H_j)}{P(q|H_{null})} - \log_2 \frac{P(q|H_i)}{P(q|H_{null})} \\ &= \log_2 \frac{P(q|H_j)}{P(q|H_i)} \\ &= \log_2 \frac{P(q|H_j)}{P(q|H_i)} \end{aligned} \tag{7}$$

Substituting Equation (7) in (5) we have

$$w_{H_i, q} = P(H_i|q) = \frac{1}{\sum_{j=1}^{d} 2^{BS(H_j, q) - BS(H_i, q) + \log_2 \frac{s_j}{s_i}}} \tag{8}$$

where $w_{H_i, q}$ is the probability that q is generated by H_i and is denoted as the weight for the pair q, H_i.

It is trivial to show that for any given query sequence q, $\sum_{i=1}^{d} w_{H_i, q} = 1$. Thus, given a query sequence q, these weights define probabilities on the HMMs in the ensemble.

We refer to these weights as ''adjusted bitscores,'' and the original use within UPP (which uses the bitscores without any modification) as ''raw bitscores.''

2.5. Phase 3: weighted GCM alignment

For every query sequence q, we use HMMAlign to align q to its top k HMMs. This produces k alignments, each of which includes the sequences in S_0 and q. Note also that by construction, each induces the backbone alignment A. Hence, we refer to these as ''extended alignments.''

This step is based on the GCM from MAGUS (Smirnov and Warnow, 2021a), and so, a brief description of GCM is helpful. GCM is a technique for merging an input set of ''constraint alignments'' on disjoint sets of sequences; hence, the output merged alignment must induce the constraint alignments. GCM first computes a set of additional alignments (each on a subset of the input sequences sampled from different constraint alignments), and uses these additional alignments to define an ''alignment graph.'' This alignment graph has a node for each site in each constraint alignment, and the edges are obtained from the

additional alignments. The nodes in this graph are then clustered using a two-step process that begins with the Markov Clustering Algorithm (Van Dongen, 2000) and then modifies the clustering to enforce the requirement of being a valid alignment (also called a valid ''trace'').

As shown in Zaharias et al. (2021), this approach is a good heuristic for the Maximum Weight Trace problem posed in Kececioglu (1993), extended for use in merging disjoint alignments.

For WITCH, we slightly modify GCM so that it can be used as a method for combining extended alignments, each provided with its weight. We enforce the requirement that the backbone alignment (whether produced by MAGUS or by PASTA) be maintained so that the final set of extended alignments produced in Phase 3 (one for each query sequence) can be merged into an alignment of the entire data set using transitivity in Phase 4.

For each query sequence q, we create a ''weighted alignment graph'' based on the technique in GCM. Each column (site) in the backbone alignment A and each letter in q define a node in the alignment graph; thus, there are $L+L'$ nodes, where L is the length of the backbone alignment and L' is the length of the query sequence q. We denote the vertices derived from the backbone alignment by v_i (where i denotes the index of the site in the backbone alignment) and the vertices derived from the query sequence by q_j (where j denotes the index of the letter in the query sequence). The edges in the alignment graph are defined by the k extended alignments (each of which induces A and also contains q).

Specifically, given an extended alignment in which site i in the backbone alignment is aligned with the jth letter from q, we include an edge between v_i and q_j. The weight of the edge is the weight of the HMM (i.e., its adjusted bitscore) that was used to define this extended alignment; thus, every edge is given with a positive weight that is bounded by 1.

Given the weighted alignment graph, we then follow the steps used in GCM to produce a new extended alignment that combines the information from the k extended alignments. The remaining steps of GCM are unchanged (i.e., clustering, cleaning, and finding a trace), and the final merged alignment reported by GCM is the weighted consensus alignment on $S_0 \cup \{q\}$ (see Fig. 3 for an example).

2.6. Phase 4: transitivity

The last phase also follows the UPP procedure after we obtain consensus alignments for all query sequences. Since all consensus alignments induce the backbone alignment A, we transitively add each query sequence q into A using the weighted consensus alignment for q, introducing gaps when necessary.

3. EXPERIMENTAL DESIGN

3.1. Overview

We performed three experiments. In our first experiment, we use fragmentary versions of biological and simulated testing data sets to set defaults for the algorithmic parameters k and z and to determine whether using adjusted bitscores is beneficial. In the second experiment, we compare WITCH using these default parameters with leading multiple sequence alignment methods on additional simulated data sets with introduced fragmentation, while the third experiment performs this analysis on additional biological data sets. In all cases, we evaluate the methods for alignment and tree estimation error. We separate the training data sets (used in Experiment 1 for setting the algorithmic parameters) from the test data sets (used in Experiments 2 and 3 for comparing WITCH with other methods).

All analyses were run on the UIUC Campus Cluster, with 16 cores and 32 GB memory, and the runtime limit was set to 24 hours. For tree estimation, we ran RAxML-NG (Kozlov et al., 2019), a maximum likelihood-based tree estimation method, on 16 cores and 32 GB memory for up to 4 hours, and the last tree reported within the time limit was used for comparison. See Supplementary Materials section S1 for commands needed to reproduce the experiments.

3.2. Data sets

We evaluate WITCH on both simulated and biological nucleotide data sets, including versions with introduced fragmentation. The simulated data sets include nine model conditions generated using the ROSE (Stoye et al., 1998) software, each with 1000 sequences per replicate and 20 replicates per model condition, which have been used in prior studies to evaluate alignment methods (Liu et al., 2009, 2012; Mirarab et al.,

2015; Smirnov and Warnow, 2021a). The biological data sets include five ribosomal RNA data sets from the Comparative Ribosomal Website (CRW) (Cannone et al., 2002) (see Section 3.2.2.). Most data sets come from prior studies and are available in public databases, while the remaining data sets are available in the Illinois Data Bank (see the Data Availability section). The empirical statistics for all data sets can be found in Table 1, which includes numbers of sequences, average and maximum *p*-distances, percentages of gaps, average sequence lengths, and alignment lengths.

In addition, Figure 1 presents histograms about the sequence length distribution for the five biological data sets and a representative simulated data set.

The simulated data sets have known true alignments and trees, which allows us to evaluate alignment and tree accuracy. The biological data sets have reliable reference alignments based on the RNA structure (Cannone et al., 2002). However, the biological data sets do not have reliable reference trees, and so, we can only evaluate alignment accuracy on the biological data sets.

We made versions of these simulated data sets that had fragmentary sequences, with relative long fragments ($\sim$500 bp) and shorter fragments ($\sim$250 bp). All in all, we computed alignments on 18 model conditions (9 basic model conditions, each with low-fragmentary [LF] or high-fragmentary [HF] sequences) and each model condition had 19 or 20 replicates, making for 358 1000-sequence data sets (since we removed one replicate from one basic model condition).

3.2.1. Simulated data sets. We use nine 1000-sequence model conditions from prior studies to evaluate alignment methods SATé, PASTA, and MAGUS (Liu et al., 2009; Mirarab et al., 2015; Smirnov and Warnow, 2021a); these were simulated using the ROSE (Stoye et al., 1998) software, and so are named ''ROSE'' data sets. We used the 1000L4, 1000S4, 1000M3, 1000S2, 1000M2, 1000L1, 1000L3, 1000S1, and 1000M1 model conditions, which evolve with indels and also substitutions (under the GTRGAMMA model).

These models range in difficulty depending on the rate of evolution and probability of indels, with 1000S4 among the easiest models and 1000M1 among the most difficult. The letters ''L/M/S'' in the model name denote the indel length in the alignment, where ''L'' is for long indel, ''M'' for medium, and ''S'' for short. Each model condition has 20 replicates [although we removed one replicate from the 1000M1 condition as it was considered an outlier in a prior study (Smirnov and Warnow, 2021a)].

Table 1. Empirical Statistics for All Data Sets

		p-distance				
Data set	*No. of sequences*	*Average*	*Max*	*% gaps*	*Average sequence length*	*Alignment length*
Simulated						
1000M1(19)*	1000	0.694	0.781	74.3	1011	3960
1000S1(20)	1000	0.694	0.782	53.0	1002	2141
1000L3(20)	1000	0.687	0.770	85.2	1031	7042
1000L1(20)	1000	0.695	0.782	73.2	1015	3817
1000M2(20)*	1000	0.684	0.775	74.2	1014	3972
1000S2(20)	1000	0.693	0.776	35.0	1001	1546
1000M3(20)	1000	0.660	0.754	62.8	1007	2722
1000L4(20)	1000	0.500	0.627	58.6	1007	2446
1000S4(20)	1000	0.501	0.625	24.6	1000	1328
Biological						
16S.M(1)*	901	0.359	0.887	78.1	1035	4722
23S.M(1)*	278	0.377	0.703	83.7	1746	10,738
5S.3	5507	0.418	1.000	74.5	105	414
5S.T	5751	0.425	1.000	75.6	106	436
16S.B.ALL	27,643	0.210	0.769	80.0	1372	6857

Statistics are computed and averaged over all replicates (numbers of replicates are marked next to data set names) and before introducing any fragmentation. The *p*-distance denotes the fraction of sites between two aligned sequences that have different nucleotides and % gaps denote the percentage of the alignment that is occupied by dashes (gaps). All data sets are made high/low fragmentary (see Section 3.2 for definitions of high/low levels of fragmentation). Replicate 16 for 1000M1 is not included in this study because it is identified as a persistent outlier in Smirnov and Warnow (2021b). The training data sets are indicated by asterisks (*).

3.2.2. Biological data sets. We use five data sets from the CRW (Cannone et al., 2002), which are based on structural alignments. Two of these data sets (i.e., 16S.M and 23S.M) are used in Experiment 1 (training), and the remaining three are used in Experiment 3 (testing). Their sequence length histograms are shown in Figure 1.

3.2.3. Introduced fragmentation. The training data sets from Experiment 1 are from a prior study (and publicly available), and have fragmentation introduced using a protocol described in Nguyen et al. (2015). To create the testing data sets in Experiment 2, we used the same approach as was used to create the training data sets used in Experiment 1. Both Experiment 1 and Experiment 2 have two levels of fragmentation, referred to as HF and LF:

- *HF* means that 50% of the sequences are made into fragments. Fragment lengths are sampled from a normal distribution $\mathcal{N}(M, 60)$, where M corresponds to 25% of the original median sequence length.
- *LF* is similar to HF and uses $\mathcal{N}(M, 60)$ to sample fragment lengths, except that only 25% of the sequences are made into fragments and M corresponds to 50% of the original median sequence length.

On the simulated data sets we used, the HF versions produced fragments of mean between 250 and 260 bp in length, and the LF versions had fragments of mean between 490 and 510 in length. In Experiment 3 we also introduced fragmentation into the 16S.B.ALL data set, but with shorter fragments averaging 100 bp in length. The script to generate an alignment with fragments is available at https://git.io/JOGO1

3.2.4. Training versus testing data. We use HF/LF versions of 1000M1, 1000M2, 16S.M, and 23S.M as the training data for Experiment 1 (algorithmic parameter selection), and the remaining data sets for the experiments where we evaluate methods.

3.3. Other alignment methods

We compare WITCH with MAFFT (Katoh and Standley, 2013), Clustal-Omega (Sievers and Higgins, 2018), PASTA (Mirarab et al., 2015), MAGUS (Smirnov and Warnow, 2021a), and UPP (Nguyen et al., 2015). This collection includes methods that are closely related to WITCH (i.e., MAGUS and UPP) and other methods that have performed well in prior studies (i.e., Clustal-Omega, MAFFT, and PASTA).

The current default version of MAGUS uses a recursive approach on subset alignments if they contain >200 sequences; however, Smirnov (2021) noted that ''recursion does not improve accuracy.… recursion should be avoided if possible, and only engaged when the dataset becomes too large for the subsets to be reasonably aligned with the base method.'' Therefore, we use MAGUS without recursion in this study. Also, since UPP has better accuracy using the MAGUS backbone instead of PASTA (Shen et al., 2022b), we use UPP in this way, and denote this usage by MAGUS+UPP.

3.4. Criteria

Alignment error is calculated using sum-of-pairs false-negative (SPFN) and sum-of-pairs false-positive (SPFP) rates, where the SPFN rate is the fraction of pairs of homologies that are in the reference alignment but missing in the estimated alignment, and the SPFP rate is the fraction of pairs of homologies that are in the estimated alignment but missing in the reference alignment. We also report the average of these two values. These error rates are obtained using FastSP (Mirarab and Warnow, 2011).

We also compute trees on the estimated alignments using RAxML-NG (Kozlov et al., 2019), a popular maximum likelihood-based tree estimation method. We report tree error by computing the missing branch FN, or false negative error rate, which is the fraction of the branches in the reference tree (i.e., model tree for the simulated data sets) missing from the estimated tree. Since the true tree is not known in the biological data sets and the reference trees are unreliable, we do not report tree error for the biological data sets.

4. RESULTS AND DISCUSSION

4.1. Experiment 1: algorithmic parameter selection

We explore the following three algorithmic parameters for use within WITCH:

- z: the subtree size that determines the decomposition stopping condition. We vary z between 2, 5, 10, and 50. This impacts Phase 1.

- k: the number of HMMs selected to align a given query sequence. We vary k between 1, 2, 4, and 10. This impacts Phases 2 and 3.
- HMM ranking: unweighted or weighted. ''Unweighted'' means that the HMMs are ranked using raw bitscores, and then each created extended alignment has unit weight in the alignment graph. ''Weighted'' means that the HMMs are ranked using adjusted bitscores, and then, these weights are used when constructing the alignment graph. This impacts Phases 2 and 3.

These parameter settings thus impact Phases 1, 2, and 3. Phase 0, which selects the backbone sequences and then aligns them, is performed as in MAGUS+UPP, and Phase 4, which combines the extended alignments (one for each query sequence), is performed as in UPP.

We let z range from 2 to 50 and k range from 1 to 10. Note that setting $z=10$, $k=1$, and using unweighted HMMs is identical to MAGUS+UPP. We explore the impact of these choices on our training data sets: the HF and LF versions of 1000M1, 1000M2, 16S.M, and 23S.M. In Supplementary Materials, section S2 contains full results across these training data sets (Supplementary Tables S1–S5) as well as a detailed discussion. Here we summarize that discussion, and note how we used the results to set the algorithmic parameters for WITCH.

The impact of the parameters depends on the model condition, as we now discuss. The only noteworthy differences in alignment accuracy and tree accuracy mainly appear in the HF conditions (where the fragments are $\sim$250 bp in length) and even then the degree of difference depends on the average p-distance in the data set (where the p-distance between two sequences is the fraction of the sites where they have different nucleotides). Focusing therefore only on the HF conditions, we examine how the average p-distance in the data set affects algorithmic parameter selection. Of the four data sets, the highest p-distance is in the 1000M1 condition (0.694), followed by 1000M2 (0.683) and then by the 23S.M (0.377) and 16S.M (0.359).

We find essentially no impact from the choice of algorithmic parameters on the 16S.M and 23S.M data sets, even under HF conditions, indicating that data sets with low p-distances where the fragmentary sequences are $\sim$25% of full-length are still easy to align using MAGUS+UPP. We see that the algorithmic parameters have a larger impact on the 1000M1 condition than on the 1000M2 condition. Thus, results on these training data sets demonstrate that the impact is greatest when p-distances are large and there are a large number of short sequences (our HF condition).

Finally, we saw that the somewhat small differences in alignment error can nevertheless result in somewhat larger differences in tree error, indicating that algorithmic parameter selection may have a larger impact on tree estimation than on alignment estimation error (at least as measured using SPFN and SPFP).

However, we also noted that certain settings of the parameters provided consistent advantages over the other settings under challenging conditions and matched the other settings under easier conditions. Specifically, we found that setting $k=10$ and $z=2$, and using adjusted bitscores provided the best results. Some other settings came close to the same accuracy, with $z=5$ or even $z=10$ often close in accuracy as long as $k=10$ was used. Switching from adjusted bitscores to raw bitscores often had only a small impact on alignment accuracy, but reduced tree accuracy on the HF conditions.

We also saw that the improvement in alignment accuracy obtained by this setting was mainly through an improvement in recall (i.e., reduction in SPFN), indicating that the WITCH technique of using multiple HMMs and adjusted bitscores rather than raw bitscores is better at detecting true homologies than the technique in UPP. Based on these trends, we set default values for the WITCH parameters to be $z=2$, $k=10$, and using adjusted rather than raw bitscores, and used these settings in the subsequent experiments.

We provide a direct comparison of WITCH with the selected parameters to MAGUS+UPP (i.e., UPP with MAGUS backbone) on these training data sets (Fig. 4). Note that WITCH has improved alignment accuracy and tree accuracy on the HF versions of 1000M1 and 1000M2, while the impact on the LF condition is extremely small. We do not show results for 16S.M and 23S.M as there was no visible difference in alignment error between WITCH and MAGUS+UPP on these data sets, and it is not possible to evaluate tree error as the true evolutionary trees are not known.

4.2. Experiment 2: evaluation of WITCH on simulated data sets

4.2.1. Overview. We compare WITCH (using the default settings for the parameters from Experiment 1) with MAGUS+UPP (i.e., UPP with MAGUS backbone), MAGUS, two MAFFT variants (default and L-INS-i), and Clustal-Omega, evaluating all methods with respect to alignment error and tree estimation error.

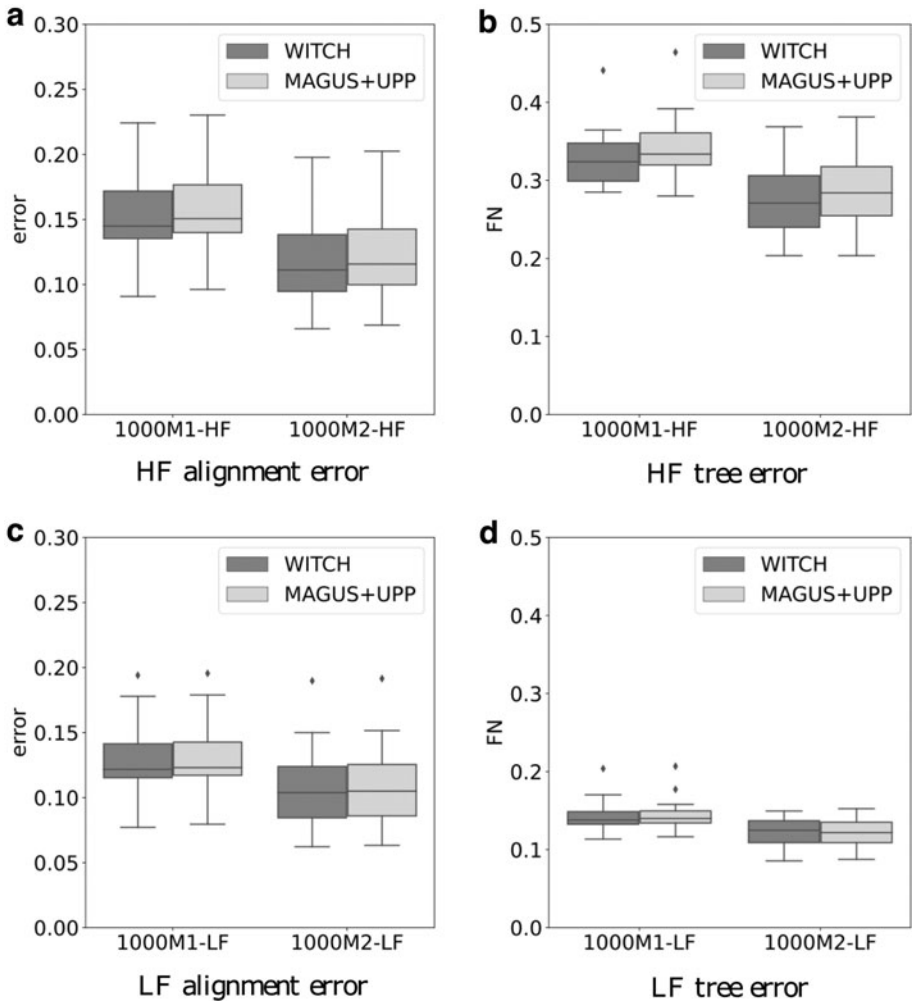

FIG. 4. Experiment 1: Alignment error (average of SPFN and SPFP) and tree error (FN, or false negative) of WITCH with selected parameters ($z=2$, $k=10$, and using adjusted bitscores) and MAGUS+UPP on ROSE 1000M1 and 1000M2 data sets in high and low fragmentation conditions. HF, high fragmentary; LF, low fragmentary; SPFN, sum-of-pairs false negative; SPFP, sum-of-pairs false positive. **(a)** Alignment error for HF condition. **(b)** Tree error for HF condition. **(c)** Alignment error for LF condition. **(d)** Tree error for LF condition.

An examination of the HF model conditions (Fig. 5 and Table 2) shows that the methods vary substantially with respect to both alignment error and tree error (see Supplementary Fig. S1 for SPFN and SPFP). The two-most accurate methods are WITCH and MAGUS+UPP, followed by MAGUS, then by PASTA and MAFFT L-INS-i, and then by Clustal-Omega and MAFFT in default mode (but with a small advantage to Clustal-Omega). There is a large gap between the two-most accurate methods (WITCH and MAGUS+UPP) and MAGUS, and an even larger gap between MAGUS and the remaining methods, especially for the more difficult model conditions (i.e., the ones with higher rates of evolution).

However, all methods except for default MAFFT and Clustal-Omega produce alignments and trees of comparable accuracy with the best methods for the two easiest model conditions (1000L4 and 1000S4). Although WITCH generally produces more accurate alignments and trees than MAGUS+UPP, the improvement is restricted to the harder model conditions; on the easiest model conditions (i.e., the ones with lower rates of evolution), the two methods are extremely close. Specifically, WITCH has lower alignment error and tree error than MAGUS+UPP for the five harder conditions (1000S1, 1000L3, 1000L1, 1000S2, and 1000M3) and then is slightly higher (by 0.1%) on the two easier model conditions (1000L4 and 1000S4).

An evaluation of the low-fragmentation conditions (Supplementary Fig. S2 and Supplementary Table S6) reveals the same relative performance between methods, although in some cases with smaller

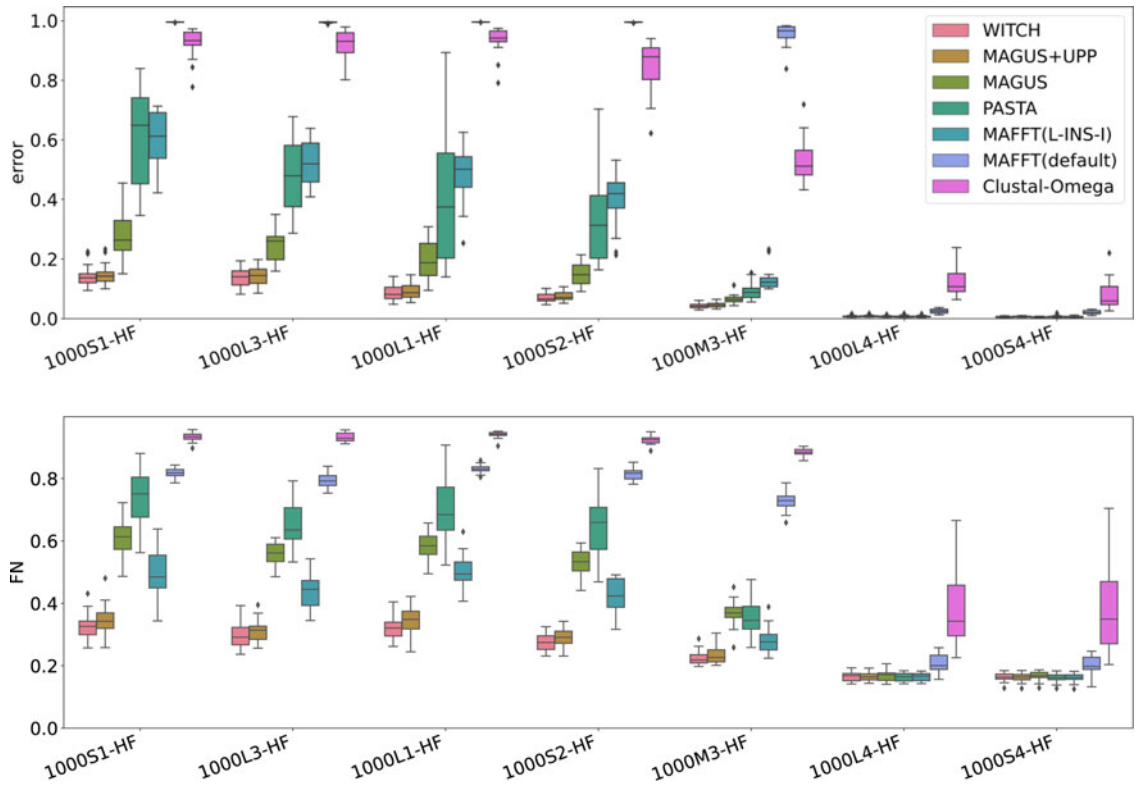

FIG. 5. Experiment 2: Comparison of WITCH with other methods on the ROSE 1000-taxon data sets with high fragmentation. Top: Alignment error. Bottom: Tree error. Results are shown across 20 replicates for each model condition (except for 1000M1, which is missing one replicate). Alignment error is the average of SPFN and SPFP, and tree error is the missing branch (FN) rate.

differences. Clustal-Omega and default MAFFT are still the least accurate, WITCH and MAGUS+UPP are still the most accurate, and all methods (other than Clustal-Omega and default MAFFT) have essentially the same accuracy for the two easiest model conditions. WITCH has an advantage over MAGUS+UPP on the model conditions with higher rates of evolution, and then ties (with no advantage to WITCH) under the two easiest model conditions.

Table 2. Experiment 2: Comparison of the Three-Most Accurate Methods (WITCH, MAGUS+UPP, and MAGUS) on the High-Fragmentation Model Conditions

	1000S1	*1000L3*	*1000L1*	*1000S2*	*1000M3*	*1000L4*	*1000S4*
Alignment error							
WITCH	**0.142**	**0.135**	**0.085**	**0.070**	**0.042**	**0.007**	**0.006**
MAGUS+UPP	0.147	0.140	0.091	0.075	0.045	**0.008**	**0.006**
MAGUS	0.281	0.246	0.198	0.150	0.066	**0.007**	**0.005**
Tree error							
WITCH	**0.327**	**0.299**	**0.324**	**0.276**	**0.226**	**0.165**	**0.163**
MAGUS+UPP	0.346	0.312	0.345	0.290	0.234	**0.164**	**0.162**
MAGUS	0.610	0.559	0.582	0.528	0.369	0.167	0.166

Top: average alignment error. Bottom: average tree error. Results shown are for the ROSE 1000-sequence data sets with high fragmentation. Alignment error is the average of SPFN and SPFP, and tree error is the missing branch FN, or false negative rate. The best result for each data set is boldfaced (methods are considered tied if the difference is at most 0.001).

MAGUS, Multiple sequence Alignment using Graph clUStering; SPFN, sum-of-pairs false negative; SPFP, sum-of-pairs false positive; UPP, ultra-large alignments using Phylogeny-Aware Profiles; WITCH, WeIghTed Consensus Hmm alignment.

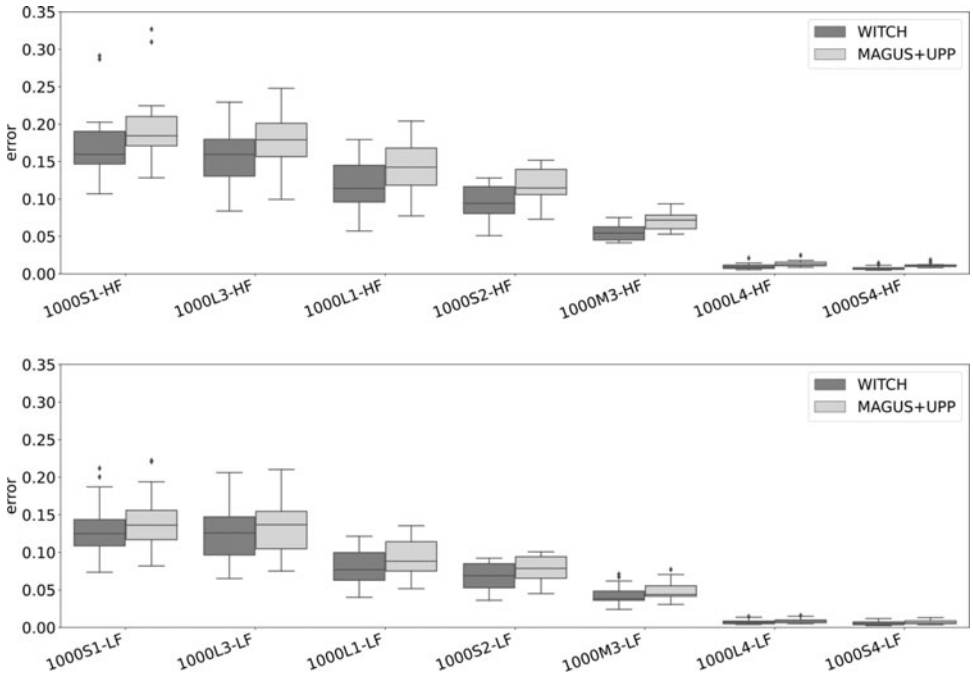

FIG. 6. Experiment 2: Average alignment error (average of SPFN and SPFP) of WITCH and MAGUS+UPP on alignments induced on the fragmentary sequences. Top: ROSE 1000-sequence HF conditions (i.e., 500 fragmentary sequences with average length ~250 bp); Bottom: ROSE 1000-sequence LF conditions (i.e., 250 fragmentary sequences with average length ~500 bp).

4.2.2. Comparison of WITCH and MAGUS+UPP. Since WITCH and MAGUS+UPP had better accuracy than the remaining methods, we directly compared them with respect to alignment and tree error, as well as with running time.

4.2.2.1. Alignment error

The difference in alignment error between WITCH and MAGUS+UPP is very small on the full data sets for both HF and LF conditions. The explanation is the obvious one: these two alignment methods always start with the same backbone alignments and so differ only in terms of how they add the remaining sequences to the backbone alignment. Therefore, we examine alignment error on the fragmentary sequences alone.

When restricted to alignments induced on the fragmentary sequences, WITCH has lower alignment error than MAGUS+UPP for both HF and LF conditions (Fig. 6). The improvement in alignment accuracy for WITCH under the individual HF model conditions is not statistically significant (Supplementary Table S10, p 0.055), but becomes statistically significant when pooling only the five hardest model conditions together (p 0.006). This suggests that the limited number of samples—at most 20 per individual model condition—is insufficient. In contrast, the improvement in alignment error on the LF conditions is not statistically significant for all seven conditions pooled, nor when pooling the five-most difficult conditions (p-values of 0.307 and 0.158, respectively, Supplementary Table S10).

We then decomposed alignment error on fragmentary sequences into SPFN and SPFP. WITCH is slightly higher than MAGUS+UPP for SPFP (by at most 0.7% per model condition), but distinctly lower with respect to SPFN, by up to 5.6% per model condition (Supplementary Table S7). Thus, the decrease in SPFN (increased recall) is larger than the increase in SPFP (decreased precision), which is why WITCH has an overall improvement in alignment accuracy over MAGUS+UPP. Furthermore, the increase in SPFP is not statistically significant (p-values of 0.29 and 0.37 for pooled HF and LF conditions, respectively, Supplementary Table S11), while the decrease in SPFN is statistically significant (p-values of 0.001 and 0.025 for pooled HF and LF conditions, respectively, Supplementary Table S12).

The trend that WITCH has a better overall alignment accuracy than MAGUS+UPP, and that this is due to improved SPFN but slightly worse SPFP, shows that the consensus alignment WITCH computes for each

query sequence detect additional homologies than the alignment MAGUS+UPP computes for each query sequence. Since these differ in terms of the number of HMMs used by each method and how they are weighted, this shows that these algorithmic changes implemented in WITCH improve recall, with a small decrease in precision, and lead to an overall improvement in accuracy.

4.2.2.2. Tree error

As seen in Figure 7, WITCH provides improved accuracy under the more difficult conditions (i.e., HF model conditions with high rates of evolution), but not under the easier model conditions (i.e., the HF conditions with very low rates of evolution and all the LF conditions).

We evaluate the statistical significance of the difference in tree error on the HF conditions. The differences on the pooled five hardest data sets on HF conditions are statistically significant (p 0.044), but otherwise not (Supplementary Table S9). Each of the five hardest data sets on HF conditions has a p-value <0.3, but in other individual cases, p-values are larger than 0.5. These trends suggest that the number of replicates per model (20) may not be enough to establish statistical significance for hard individual models on HF conditions, and that both methods may be just as good in terms of tree estimation error when data sets are easy.

4.2.2.3. Running time

For all the data sets in this experiment, MAGUS+UPP is strictly faster than WITCH (Supplementary Fig. S4). This is expected since WITCH does strictly more work than MAGUS+UPP given that $k = 10$. However, the differences in runtime are not very large: MAGUS+UPP ranges from about 6 minutes to about 15 minutes per data set, and WITCH ranges from about 11 minutes to about 20 minutes, each assuming 16 cores. Thus, both methods are reasonably fast on these 1000-sequence data sets.

4.3. Experiment 3: evaluation of WITCH on biological data sets

In this experiment, we compare WITCH with MAGUS+UPP on three biological data sets (5S.3, 5S.T, and 16S.B.ALL). These data sets range in size from 5507 sequences (5S.3) to 27,643 sequences (16S.B.ALL), and also in average sequence length (105 bp for 5S.3, 106 bp for 5S.T, and 1272 bp for

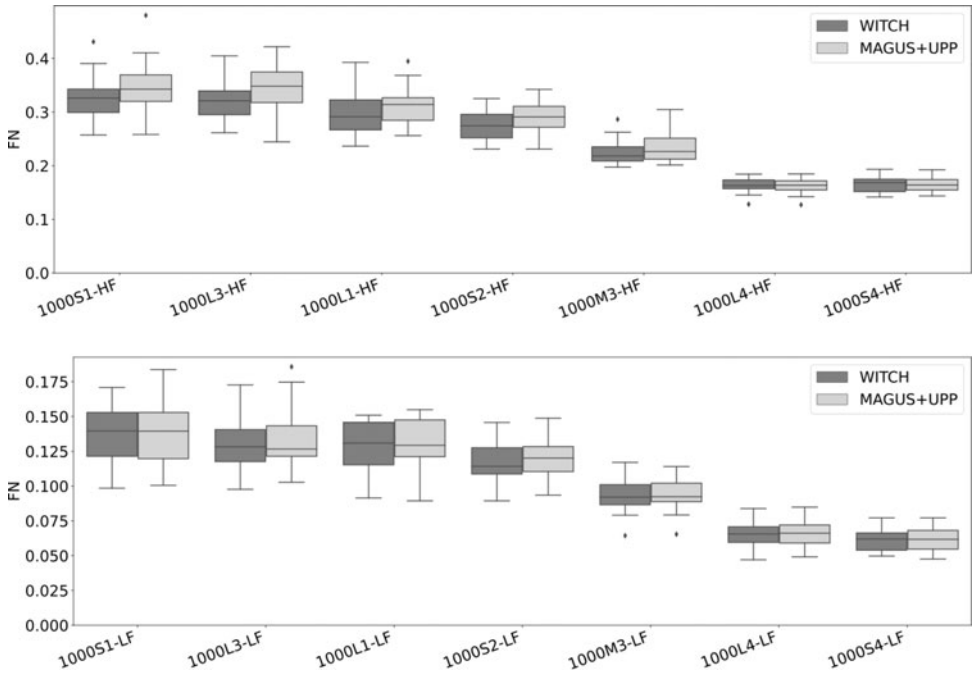

FIG. 7. Experiment 2: Tree error (FN rate) of WITCH and MAGUS+UPP. Top: ROSE 1000-sequence HF conditions (i.e., 500 fragmentary sequences with average length ~250 bp); Bottom: ROSE 1000-sequence LF conditions (i.e., 250 fragmentary sequences with average length ~500 bp).

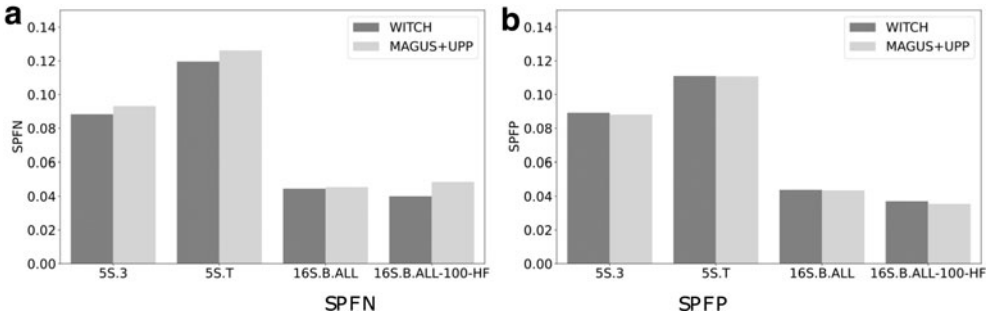

FIG. 8. Experiment 3: SPFN and SPFP of MAGUS+UPP and WITCH alignments on query sequences only for 5S.3, 5S.T, 16S.B.ALL, and 16S.B.ALL-100-HF data sets, both using the same backbone alignment and eHMM. **(a)** SPFN; **(b)** SPFP. Note that WITCH and MAGUS+UPP are nearly indistinguishable for SPFP (precision), but WITCH improves on MAGUS+UPP for SPFN (recall).

16S.B.ALL). Although these biological data sets have reference alignments, they do not have established phylogenies, and so, we can only evaluate alignment errors on these data sets. We also consider a version of 16S.B.ALL with introduced fragmentation where the fragmentary sequences averaged 100 bp in length and refer to it as 16S.B.ALL-100-HF. Because the sequences in the two 5S data sets are very short, we do not introduce fragmentation into them.

We report SPFN and SPFP for alignments induced on the query sequences for each data set (Fig. 8). WITCH provides an advantage over MAGUS+UPP for alignment error on the two 5S data sets and the 16S.B.ALL-100-HF data set, and it matches MAGUS+UPP on the 16S.B.ALL data set. The improvement in accuracy is mainly through improvement in SPFN (i.e., recall), indicating that WITCH is able to recover more true homologies than MAGUS+UPP.

Overall, this comparison shows an advantage for SPFN to WITCH over MAGUS+UPP on three of the four data sets (5S.3, 5S.T, and 16S.B.ALL-100-HF). SPFP is essentially unchanged on all four data sets, and analyses of 16S.B.ALL without introduced fragmentation also do not show any differences worth noting between WITCH and MAGUS+UPP.

To more fully explore differences between WITCH and MAGUS+UPP, we evaluated WITCH and MAGUS+UPP when using the reference alignments (instead of the MAGUS alignments) for the backbone alignments on these data sets (see Supplementary Materials section S4). Although the alignment error drops for both methods, WITCH continues to demonstrate improved overall alignment accuracy compared with MAGUS+UPP, largely through reducing SPFN (i.e., improving recall) more than SPFP is increased (i.e., reducing precision). We illustrate the difference between WITCH and MAGUS+UPP by showing how a single query sequence from 5S.T is aligned differently by the two methods when using the reference alignment as the backbone (Fig. 9). Note that both WITCH and MAGUS+UPP come very close to fully recovering the reference alignment for the query sequence, but that MAGUS+UPP misses more homologies than WITCH does.

4.4. Alignment method selection

A consistent finding across Experiments 1 and 2 is that WITCH always matches or improves on MAGUS+UPP for both alignment and tree accuracy, but that the degree of improvement depends on the data set properties. Specifically, under the LF condition on the simulated data sets, the data sets have only a few fragmentary sequences (and the fragments have average length $\sim$500 bp), WITCH alignments and MAGUS+UPP alignments do not differ substantially in terms of SPFN and SPFP error, and trees estimated on these alignments also do not differ substantially.

In contrast, under the HF condition, the fragments are shorter (about 250 bp for the simulated data sets) and there are more such fragments. Thus, the degree of fragmentation impacts whether WITCH improves on MAGUS+UPP. In addition, the degree of improvement also depends on the sequence similarity within the data set, so that data sets with higher average p-distances are more impacted by this choice, and WITCH provides a greater advantage over MAGUS+UPP.

Experiment 3 results are helpful in interpreting these trends. On the three data sets without any introduced fragmentation, WITCH provides an accuracy advantage over MAGUS+UPP on 5S.3 and 5S.T, but

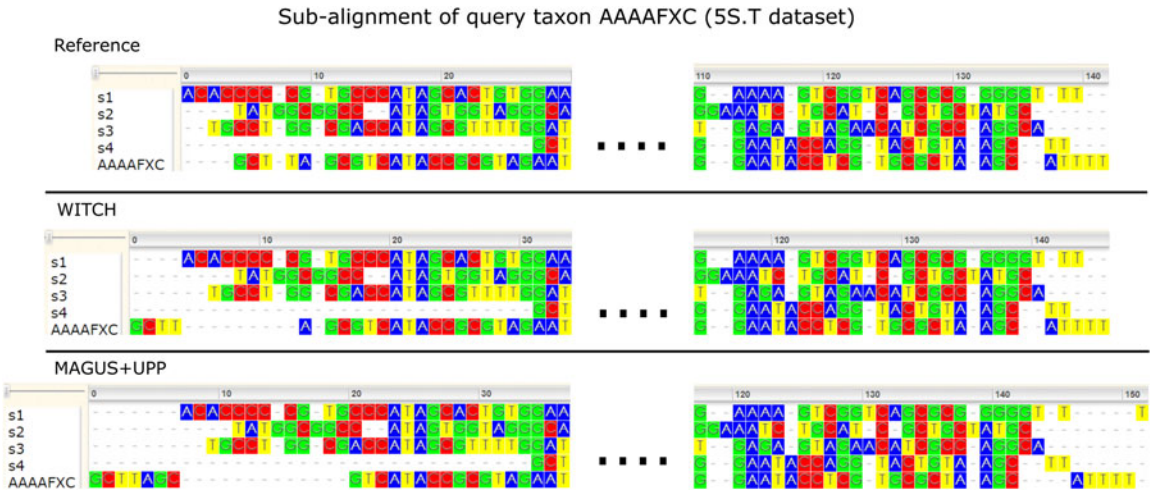

FIG. 9. Experiment 3: A visualization using Wasabi (Veidenberg et al., 2016) of three alignments, showing how a query sequence in 5S.T is aligned differently by WITCH and MAGUS+UPP. From top to bottom, we have the reference alignment, WITCH alignment, and MAGUS+UPP alignment, each induced on sequences $s1$, $s2$, $s3$, $s4$ and AAAAFXC. Sequences $s1$, $s2$, $s3$, $s4$ (namely AAAAAWP, AAAAAFV, AAAAEIS, and AAAAGKE) are from the reference backbone alignment, which is the same for both MAGUS+UPP and WITCH. Note that MAGUS+UPP underaligns, missing some homologies that WITCH is able to recover correctly.

not on 16S.B.ALL, where it matches MAGUS+UPP. Each of these data sets exhibits sequence length heterogeneity (Fig. 1). In examining their *p*-distances, we see that 5S.3 and 5S.T have higher average *p*-distances than 16S.B.ALL (i.e., mean *p*-distances for 5S.3 and 5S.E are both at least 0.418, whereas the mean *p*-distance for 16S.B.ALL is only 0.21). These trends together suggest that 16S.B.ALL (due to its low average *p*-distance and modest amount of sequence length heterogeneity) should be easily aligned by both WITCH and MAGUS+UPP, but that 5S.T and 5S.3 may benefit from using WITCH instead of MAGUS+UPP. This prediction matches what we see on these data sets.

An examination of results on 16S.B.ALL-100-HF (i.e., with short sequences of average length only 100 bp) shows that WITCH provides an accuracy advantage over MAGUS+UPP in this case. In contrast, recall that the HF conditions in Experiments 1 and 2 produced fragments that were $\sim$25% of the full-length sequences (e.g., about 250 bp for the simulated data sets). In other words, the HF conditions in Experiments 1 and 2 produced fragments that were substantially longer than the HF condition in Experiment 3. This may suggest that when the fragmentary sequences are sufficiently short, even data sets with low *p*-distances can become difficult to align, and WITCH may provide an advantage over MAGUS+UPP.

With this in mind, we consider the properties of biological data sets that might be important when choosing between methods. As Supplementary Materials section S3 shows, when there is no sequence length heterogeneity (e.g., on simulated data sets without introduced fragmentation), WITCH and MAGUS+UPP are both very accurate, but not the most accurate of the tested methods. Instead, MAGUS is the most accurate on these data and is slightly more accurate than WITCH, MAGUS+UPP, and PASTA (which are close in terms of alignment accuracy and better than MAFFT and Clustal-Omega).

However, as seen in Figure 1, biological data sets exhibit much higher levels of sequence length heterogeneity than these simulated data sets, so that these simulated data sets without introduced fragmentation are not typical of biological data. Therefore, the choice of alignment method clearly should be governed by a consideration of the sequence length heterogeneity and the average p-distance. However, additional research is needed to better understand the conditions under which each method provides the best accuracy.

5. CONCLUSIONS

We have presented WITCH, a method for multiple sequence alignment of data sets that contain fragmentary sequences. WITCH has the same two-stage structure as MAGUS+UPP, the previous

most accurate method for this problem: it builds an alignment and an eHMM on selected full-length sequences, and then adds the remaining ''query'' sequences into the alignment using the ensemble.

However, WITCH improves on MAGUS+UPP through two algorithmic innovations. First, WITCH computes an adjusted bitscore for each query-HMM pair, where the adjusted bitscore is an estimate of the probability that the HMM generates the query. Second, to add in a query sequence into the alignment on the full-length sequence, WITCH uses an ensemble approach that allows it to combine information from all of the HMMs in the ensemble. In contrast, MAGUS+UPP considers only one HMM for each query sequence and uses the raw bitscores. The ensemble approach itself is interesting: to add a single query sequence into the backbone alignment, WITCH uses the information from the HMMs to define a weighted graph (where weights are defined using the adjusted bitscores) and then clusters the graph.

Thus, the ability to use more than one HMM in the ensemble and to interpret the information provided by each HMM in a statistically rigorous manner (because of the use of the adjusted bitscore) are the reasons that WITCH provides improved accuracy over MAGUS+UPP.

Although WITCH provides an advance in alignment estimation given sequence length heterogeneity, fully addressing the challenge of improving alignment estimation for such data sets most likely will require additional algorithmic innovations. For example, this study did not modify how UPP selects the backbone sequences, which is likely to be an important aspect of this two-stage approach. For the simulated data sets explored in this study, all the sequence length heterogeneity is due to the inclusion of fragmentary sequences, which makes the selection of full-length sequences very easy. Future work will need to consider how to select the backbone sequence data sets where sequence length heterogeneity may be characterized by the inclusion of excessively long sequences as well as fragmentary sequences.

Another potential improvement is changing how sequences are added into backbone alignments. In WITCH, as in UPP, we used HMMER tools to add sequences into selected HMMs; this technique may work well under many conditions, but was only tested for adding sequences that are short. Since data sets with sequence length heterogeneity also contain very long sequences, it is possible that the HMM-based methods in HMMER may not provide sufficient accuracy, and new approaches may be needed. The success in using the weighted version of GCM to compute an extended alignment for each query sequence suggests that other ways of computing consensus alignments, potentially equipped with statistically defined weights, would yield improved alignment estimation beyond what is already achieved through WITCH.

In particular, other methods for computing consensus alignments have been developed (Prasad et al., 2003; Wallace et al., 2006; Collingridge and Kelly, 2012), and future work could explore the use of these methods instead of the GCM step in WITCH.

A basic problem that we have not touched on is how to estimate a tree when there is sequence length heterogeneity. While some studies have shown that FastTree 2 (a very fast heuristic for maximum likelihood) can be highly accurate for tree topology estimation, competitive with RAxML (a much slower heuristic) in many cases (e.g., Liu et al. 2011), other studies have shown that FastTree 2 can be much less accurate than RAxML given the alignments with substantial sequence length heterogeneity (Smirnov and Warnow, 2021b; Park et al., 2021). Moreover, the poor accuracy for FastTree 2 compared with RAxML holds even if the true alignment is provided (Sayyari et al., 2017). Thus, tree estimation itself is a problem of concern, when working with sequence length heterogeneity.

The approach presented in Ashkenazy et al. (2019) is relevant to this question, when alternative multiple sequence alignments are available. Specifically, given a collection of multiple different alignments for the same sequences, Ashkenazy et al. (2019) concatenated the alignments and then estimated the tree on the concatenated alignment, and found that this improved phylogenetic accuracy. Thus, the technique in Ashkenazy et al. (2019) could be used as a way to estimate a tree on a set of unaligned sequences with sequence length heterogeneity: first, estimate multiple sequence alignments using a variety of methods (e.g., WITCH or MAGUS+UPP run in different ways, MAGUS, MAFFT), then concatenate the alignments and estimate a tree on the concatenated alignment.

In closing, given how common sequence length heterogeneity is in biological data sets, multiple sequence alignment methods should be evaluated on data sets that exhibit this heterogeneity, and we predict that future studies will identify limitations in the existing alignment methods and suggest additional opportunities for method development. WITCH provides a few useful techniques for alignment of data sets with substantial sequence length heterogeneity, but additional techniques are needed to obtain highly accurate alignments and subsequent phylogeny estimation under these challenging circumstances.

AUTHORS' CONTRIBUTIONS

C.S.: methodology, software, formal analysis, investigation, data curation, and writing—original draft. M.P.: methodology and formal analysis. T.W.: conceptualization, methodology, formal analysis, resources, validation, writing—review and editing, supervision, project administration, and funding acquisition.

ACKNOWLEDGMENTS

The authors wish to thank the anonymous reviewers whose feedback led to improvements in the article.

DATA AVAILABILITY

All the data sets are available in public repositories. Experiment 1 data sets and their fragmentary versions are available at Smirnov and Warnow (2020). Data sets used in Experiment 2 can be accessed from Shen et al. (2021a). The 5S.3, 5S.T, and 16S.B.ALL data sets from Experiment 3 can be accessed from Shen et al. (2021b). The 16S.B.ALL-100-HF data set for Experiment 3 is available at Shen et al. (2022a).

AUTHOR DISCLOSURE STATEMENT

The authors declare they have no competing financial interests.

FUNDING INFORMATION

This work was supported, in part, by the U.S. National Science Foundation grant 2006069 (to T.W.).

SUPPLEMENTARY MATERIAL

Supplementary Material

REFERENCES

Ashkenazy, H., Sela, I., Levy Karin, E., et al. 2019. Multiple sequence alignment averaging improves phylogeny reconstruction. *Syst. Biol.* 68, 117–130.

Brown, D.P., Krishnamurthy, N., and Sjölander, K. 2007. Automated protein subfamily identification and classification. *PLoS Comp. Biol.* 3, e160.

Brown, M., Hughey, R., Krogh, A., et al. 1993. Using Dirichlet mixture priors to derive hidden Markov models for protein families, 47–55. *In Proceedings of the. Intelligent Systems for Molecular Biology (ISMB),* volume. 1. AAAI (Association for the Advancement of Artificial Intelligence), Palo Alto, CA, USA.

Cannone, J.J., Subramanian, S., Schnare, M.N., 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 Bioinformatics* 3, 2.

Collingridge, P.W., and Kelly, S. 2012. MergeAlign: Improving multiple sequence alignment performance by dynamic reconstruction of consensus multiple sequence alignments. *BMC Bioinformatics* 13, 1–10.

Diplaris, S., Tsoumakas, G., Mitkas, P.A., et al. 2005. Protein classification with multiple algorithms, 448–456. *In* Bozanis, P. and Houstis, E.N., eds, *Advances in Informatics*, *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, Heidelberg.

Durbin, R., Eddy, S.R., Krogh, A., et al. 1998. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, Cambridge, UK.

Edgar, R.C. 2004. MUSCLE: Multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res.* 32, 1792–1797.

Finn, R.D., Clements, J., and Eddy, S.R. 2011. HMMER web server: Interactive sequence similarity searching. *Nucleic Acids Res.* 390, (Web Server issue) W29–W37.

Fletcher, W., and 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.

Haussler, D., Krogh, A., Mian, I.S., et al. 1993. Protein modeling using hidden Markov models: Analysis of globins, 792–802. *In Proceedings of the Twenty-Sixth Hawaii International Conference on System Sciences*, volume 1. IEEE, New York, NY, USA.

Jordan, G., and Goldman, N. 2012. The effects of alignment error and alignment filtering on the sitewise detection of positive selection. *Mol. Biol. Evol.* 29, 1125–1139.

Katoh, K., and Standley, D.M. 2013. MAFFT Multiple Sequence Alignment Software Version 7: Improvements in performance and usability. *Mol. Biol. Evol.* 30, 772–780.

Katoh, K., and Toh, H. 2008. Recent developments in the MAFFT multiple sequence alignment program. *Brief. Bioinform.* 9, 286–298.

Kececioglu, J. 1993. The maximum weight trace problem in multiple sequence alignment, 106–119. *In Annual Symposium on Combinatorial Pattern Matching*, *Lecture Notes in Computer Science*. Springer.

Kozlov, A.M., Darriba, D., Flouri, T., et al. 2019. RAxML-NG: A fast, scalable and user-friendly tool for maximum likelihood phylogenetic inference. *Bioinformatics* 35, 4453–4455.

Krogh, A., Brown, M., Mian, I.S., et al. 1994. Hidden Markov models in computational biology: Applications to protein modeling. *J. Mol. Biol.* 235, 1501–1531.

Liu, K., Linder, C.R., and Warnow, T. 2011. RAxML and FastTree: Comparing two methods for large-scale maximum likelihood phylogeny estimation. *PLoS One* 6, e27731.

Liu, K., Raghavan, S., Nelesen, S., et al. 2009. Rapid and accurate large-scale coestimation of sequence alignments and phylogenetic trees. *Science* 324, 1561–1564.

Liu, K., Warnow, T.J., Holder, M.T., et al. 2012. SATé-II: Very fast and accurate simultaneous estimation of multiple sequence alignments and phylogenetic trees. *Syst. Biol.* 61, 90.

Mirarab, S., Nguyen, N., Guo, S., et al. 2015. PASTA: Ultra-large multiple sequence alignment for nucleotide and amino-acid sequences. *J. Comput. Biol.* 22, 377–386.

Mirarab, S., and Warnow, T. 2011. FASTSP: Linear time calculation of alignment accuracy. *Bioinformatics* 27, 3250–3258.

Morrison, D.A., and Ellis, J.T. 1997. Effects of nucleotide sequence alignment on phylogeny estimation: A case study of 18S rDNAs of apicomplexa. *Mol. Biol. Evol.* 14, 428–441.

Nguyen, N.-P., Mirarab, S., Liu, B., et al. 2014. TIPP: Taxonomic identification and phylogenetic profiling. *Bioinformatics* 30, 3548–3555.

Nguyen, N.-P.D., Mirarab, S., Kumar, K., et al. 2015. Ultra-large alignments using phylogeny-aware profiles. *Genome Biol.* 16, 124.

Ogden, T.H., and Rosenberg, M.S. 2006. Multiple sequence alignment accuracy and phylogenetic inference. *Syst. Biol.* 55, 314–328.

Park, M., Zaharias, P., and Warnow, T. 2021. Disjoint tree mergers for large-scale maximum likelihood tree estimation. *Algorithms* 14, 148.

Prasad, J.C., Comeau, S.R., Vajda, S., et al. 2003. Consensus alignment for reliable framework prediction in homology modeling. *Bioinformatics* 19, 1682–1691.

Price, M.N., Dehal, P.S., and Arkin, A.P. 2010. FastTree 2 approximately maximum-likelihood trees for large alignments. *PLoS One* 5, e9490.

Sayyari, E., Whitfield, J.B., and Mirarab, S. 2017. Fragmentary gene sequences negatively impact gene tree and species tree reconstruction. *Mol. Biol. Evol.* 34, 3279–3291.

Schwacke, R., Ponce-Soto, G.Y., Krause, K., et al. 2019. MapMan4: A refined protein classification and annotation framework applicable to multi-omics data analysis. *Mol. Plant.* 12, 879–892.

Shen, C., Park, M., and Warnow, T. 2021a. Seven simulated datasets in high and low fragmentation conditions. [Epub ahead of print]; DOI: 10.13012/B2IDB-6128941_V1. Published by the Illinois Data Bank.

Shen, C., Park, M., and Warnow, T. 2022a. The 16S.B.ALL dataset in 100-HF condition. [Epub ahead of print]; DOI: 10.13012/B2IDB-6604429_V1. Published by the Illinois Data Bank.

Shen, C., Zaharias, P., and Warnow, T. 2021b. Datasets for: MAGUS+eHMMs: Improved Multiple sequence alignment accuracy for fragmentary sequences. [Epub ahead of print]; DOI: 10.13012/B2IDB-2419626_V1. Published by the Illinois Data Bank.

Shen, C., Zaharias, P., and Warnow, T. 2022b. MAGUS+eHMMs: Improved multiple sequence alignment accuracy for fragmentary sequences. *Bioinformatics* 38, 918–924.

Sievers, F. and Higgins, D.G. 2018. Clustal Omega for making accurate alignments of many protein sequences. *Protein Sci.* 27, 135–145.

Sievers, F., Wilm, A., Dineen, D., et al. 2011. Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. *Mol. Syst. Biol.* 7, 539.

Smirnov, V. 2021. Recursive MAGUS: Scalable and accurate multiple sequence alignment. *PLoS Comput. Biol.* 17, e1008950.

Smirnov, V., and Warnow, T. 2020. Datasets for: Phylogeny estimation given sequence length heterogeneity. [Epub ahead of print]; DOI: 10.5061/dryad.95x69p8h8. Published by Dryad.

Smirnov, V., and Warnow, T. 2021a. MAGUS: Multiple sequence alignment using graph clUStering. *Bioinformatics* 37, 1666–1672.

Smirnov, V., and Warnow, T. 2021b. Phylogeny estimation given sequence length heterogeneity. *Syst. Biol.* 70, 268–282.

Stoye, J., Evers, D., and Meyer, F. 1998. Rose: Generating sequence families. *Bioinformatics.* 14, 157–163.

Van Dongen, S.M. 2000. A cluster algorithm for graphs. Technical report, National Research Institute for Mathematics and Computer Science in the Netherlands, Amsterdam.

Veidenberg, A., Medlar, A., and Löytynoja, A. 2016. Wasabi: An integrated platform for evolutionary sequence analysis and data visualization. *Mol. Biol. Evol.* 33, 1126–1130.

Wallace, I.M., O'Sullivan, O., Higgins, D.G., et al. 2006. M-Coffee: Combining multiple sequence alignment methods with T-Coffee. *Nucleic Acids Res.* 34, 1692–1699.

Zaharias, P., Smirnov, V., and Warnow, T. 2021. The maximum weight trace alignment merging problem, 159–171. *In Algorithms for Computational Biology*, *Lecture Notes in Computer Science*. Springer, Cham.

Address correspondence to:
Dr. Tandy Warnow
Department of Computer Science
University of Illinois Urbana-Champaign
Urbana, IL 61801
USA

E-mail: warnow@illinois.edu