Computing Multiple Sequence Alignments given a Constraint Subset Alignment using EMMA

Chengze Shen[1], Baqiao Liu[1], Kelly P. Williams[2], and Tandy Warnow[1*]
1 - University of Illinois, Urbana-Champaign
2 - Sandia National Laboratories

Abstract

Adding sequences into an existing (possibly user-provided) alignment has multiple applications, including updating a large alignment with new data, adding sequences into a constraint alignment constructed using biological knowledge, or computing alignments in the presence of sequence length heterogeneity. Although this is a natural problem, only a few tools have been developed to use this information with high fidelity. We present EMMA (Extending Multiple alignments using MAFFT--add) for the problem of adding a set of unaligned sequences into a multiple sequence alignment (i.e., a constraint alignment). EMMA builds on MAFFT--add, which is also designed to add sequences into a given constraint alignment. EMMA improves on MAFFT--add methods by using a divide-and-conquer framework to scale its most accurate version, MAFFT-linsi--add, to backbone alignments with many sequences. We show that EMMA has an accuracy advantage over other techniques for adding sequences into alignments under many realistic conditions and can scale to large datasets with high accuracy (hundreds of thousands of sequences). EMMA is available at `https://github.com/c5shen/EMMA`.

*corresponding author - warnow@illinois.edu

1 Introduction

Multiple sequence alignment (MSA) is a crucial precursor to many downstream biological analyses, such as phylogeny estimation [15], RNA structure prediction [20], protein structure prediction [7], etc. Obtaining an accurate MSA can be challenging, especially when the dataset is large (i.e., more than 1000 sequences). In some cases, the problem of estimating an alignment on a large dataset can be addressed through approaches that seek to add sequences into an existing alignment. For example, biological knowledge can be used to form a reference alignment on a subset of the sequences, and then the remaining sequences can be added to the reference alignment; this has the potential to improve accuracy, compared to methods that do not include biological knowledge to define a constraint alignment. Another case where adding sequences into an existing alignment occurs is when new sequences or genomes are added to databases, leading to the opportunity to add the new sequences for each gene in the genome into a growing alignment. In this second case, being able to add sequences into the existing alignment avoids the need to recompute the alignment from scratch, and could lead to substantial running time benefits. A third case is for *de novo* multiple sequence alignment, where a subset of the sequences is selected and aligned, and then the remaining sequences are added into this "backbone alignment"; examples of such methods include UPP [16], UPP2 [17], WITCH [21], WITCH-ng [10], HMMerge [18], and MAFFT-sparsecore [27]. One of the motivations for this type of alignment method (which we refer to as "two-stage" methods) is when the input dataset has substantial sequence length heterogeneity, which can result in poor alignment accuracy using standard methods [16]. Thus, adding sequences into existing alignments is a natural problem with multiple applications to biological sequence analysis.

A few methods have been developed to add sequences into an existing alignment, with MAFFT--add [9] perhaps the most well-known. However, the previously mentioned multiple sequence alignment methods that operate in two steps (first extract and align the backbone sequences and then add the remaining sequences into this backbone alignment) also provide utilities for adding sequences into a user-provided alignment. We present EMMA, a new approach for adding sequences into an existing alignment. The input to EMMA is a multiple sequence alignment (which we refer to as a "backbone alignment") and an additional set of unaligned "query" sequences; EMMA adds the unaligned sequences into the backbone alignment, thus producing an alignment on the full set of sequences (backbone sequences and additional sequences). By construction, EMMA never changes the constraint alignment, except through the introduction of additional sites to represent homologies between the query sequences. EMMA uses a divide-and-conquer technique combined with MAFFT--add (using the most accurate setting, MAFFT-linsi--add) to add sequences into a user-provided alignment. We evaluate EMMA by comparing it to MAFFT-linsi--add, MAFFT--add (the default setting), and WITCH-ng-add. We include a range of biological and simulated datasets (nucleotides and proteins) ranging in size from 1000 to almost 200,000 sequences and evaluate alignment accuracy and scalability. MAFFT-linsi--add was the slowest and least scalable method, only able to run on datasets with at most 1000 sequences in this study, but had excellent accuracy (often the best) on those datasets. We also see that EMMA has better recall than WITCH-ng-add and MAFFT--add on large datasets, es-

pecially when the backbone alignment is small or clade-based.

The rest of the paper is structured as follows. In Section 2 we present the background material, including a preview of our experiments comparing MAFFT-linsi--add and MAFFT--add, which motivates the design of EMMA. In Section 3 we present the high-level design of EMMA, including the experimental study where we set the algorithmic parameters. We present the design of the experimental study comparing EMMA to other methods in Section 4, and the results of the study are provided in Section 5. We discuss trends in Section 6 and conclude with directions for future work in Section 7.

2 Background

2.1 HMM-based methods

Many of the methods that have been developed for adding sequences into a given (often user-provided) multiple sequence alignment operate by representing the existing alignment by either a single HMM or by an ensemble of HMMs. Then, for every additional sequence (called a query sequence), the HMM or ensemble of HMMs is used to find an alignment of the query sequence to the given alignment. Examples of such approaches include functionality provided in UPP [16], UPP2 [17], WITCH [21], WITCH-ng [10], and HMMerge [18]. We refer to these functions by appending "-add" to the MSA method name (e.g., this functionality in WITCH is referred to as WITCH-add). In this study, we examine performance of WITCH-ng-add as it has been shown to improve in accuracy and speed on WITCH-add (and WITCH-add is more accurate than UPP-add and UPP2-add). HMMerge-add is slower than WITCH-ng-add and so is omitted from this study. See Supplementary Materials Section S1 for additional details about these HMM-based methods.

Because HMM-based methods operate by aligning the query sequences to one or more HMMs, homologies between query sequences can only be discovered if these homologies align through match states in the HMMs. This approach has the potential to miss valid homologies between query sequences, for example when the given alignment is insufficiently representative of the entire family. Thus, only query sequence characters (i.e., nucleotides or residues) that are aligned through match states in the HMMs can be placed in columns that have other nucleotides or residues; if the character is added to the alignment through an insertion state, it will never be detected as homologous to any other character in any other sequence. This aspect of using HMMs for alignment has a potentially significant impact on the accuracy of the alignments of query sequences to the backbone alignment.

2.2 MAFFT--add and MAFFT-linsi--add

MAFFT--add [9] in its default setting uses a standard progressive alignment procedure with two iterations to add query sequences. In each iteration, it computes the pairwise distance matrix between the complete set of sequences (both in the backbone and in the query sequence set) using shared 6-mers. Then, it computes a guide tree using the distance matrix and builds an alignment. More specifically, for each node in the guide tree, MAFFT--add does an align-

ment computation only if a query sequence is involved at the node (i.e., at least one child has some query sequences). Otherwise, it simply uses the alignment from the backbone, and so is guaranteed to preserve the input backbone alignment. This property of guaranteeing that the input backbone alignment is not changed is true of the more accurate variants of MAFFT--add, including the variant that uses MAFFT-linsi to add query sequences (which we refer to as MAFFT-linsi--add), which we briefly describe below.

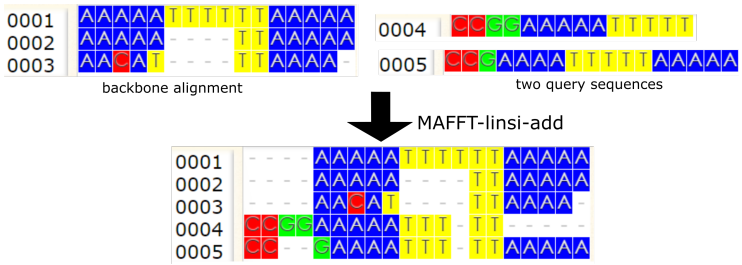


Figure 1: A simple demonstration of MAFFT-linsi--add recovering homologies between query sequences that do not have homologs in the backbone alignment, and also preserving the backbone alignment. Visualization is done with WASABI [26].

MAFFT-linsi--add has two differences to the default version of MAFFT--add. First, MAFFT-linsi--add uses *localpair* (local pairwise alignment scores) for the distance matrix calculation, which is more accurate than shared 6-mers. Second, it only runs for one iteration of progressive alignment and uses at most 1000 iterations of iterative refinement after the progressive alignment finishes. MAFFT--add and MAFFT-linsi--add have runtimes that are at least quadratic in the input size due to the $O((m+q)^2)$ distance matrix calculation, where there are q query sequences and m sequences in the provided backbone alignment. MAFFT-linsi--add is even less scalable since its distance calculation is more costly. In addition, MAFFT-linsi--add does many steps of refinement that further impact the runtime. Hence, the developers of MAFFT recommend that MAFFT-linsi--add be limited to a few hundred sequences [8].

Both MAFFT-linsi--add and MAFFT--add can recover homologies between query sequences that do not have homologous characters in the backbone alignment, and – as mentioned above – they are guaranteed to leave the backbone alignment unchanged as they add query sequences. Figure 1 gives an example of MAFFT-linsi--add adding two query sequences into a backbone alignment, and demonstrates this ability.

In contrast, methods such as WITCH-add, WITCH-ng-add, etc., that use HMMs or ensembles of HMMs to represent the backbone alignment cannot find homologies between letters in the query sequences that do not also have homologs in the backbone alignment. This is an inherent limitation of HMM-based methods for adding sequences into backbone alignment, and implies that MAFFT--add and MAFFT-linsi--add may be more robust to the selection of the backbone sequences than the HMM-based methods. Given the potential for MAFFT--add and MAFFT-linsi--add to provide improved accuracy in adding sequences into existing alignments, we explored these two methods to see how well they scaled as the size of the dataset increased, in terms of running time

and accuracy. We present these results here (see Figure 2), as they inform the design of EMMA.

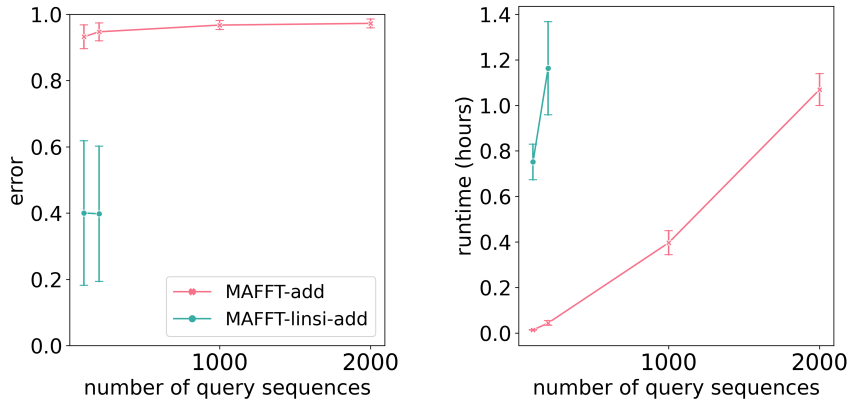


Figure 2: Experiment 0: Alignment error (left) and runtime in hours (right) of MAFFT--add and MAFFT-linsi--add for adding 100, 200, 1000, or 2000 sequences to a 1000-taxon backbone alignment computed using MAGUS [23] on the INDELible 5000M2-het dataset with 5000 sequences. Averages over ten replicates are shown. Error bars shown for alignment errors are standard errors and standard deviations for runtime. We exclude replicate 4 because MAFFT-linsi--add encountered out-of-memory issues (64 GB) when adding 100 or 200 query sequences. Additionally, MAFFT-linsi--add is not shown for 1000 or 2000 query sequences because it either encountered out-of-memory issues or failed to complete within 12 hours. Alignment error is the average of SPFN (fraction of true pairwise homologies missing in the estimated alignment) and SPFP (fraction of pairwise homologies in the estimated alignment not found in the true alignment). Results for SPFN and SPFP separately can be found in Supplementary Figure S1.

A preview of our results, shown in Figure 2, demonstrates that MAFFT-linsi--add (i.e., MAFFT--add run using -linsi) has a substantial accuracy improvement over MAFFT--add when run in default mode, but is limited to relatively small datasets. Hence, we are motivated to see if we can improve the scalability of MAFFT-linsi--add to large datasets.

3 New Method: EMMA

EMMA takes as input a multiple sequence alignment C on subset S_0 and a set S_1 of additional "query" sequences, and returns an alignment on $S = S_0 \cup S_1$ that is required to induce alignment C (thus C is treated as a constraint). We refer to C as both a backbone alignment or as a constraint alignment.

In designing EMMA, we wished to use MAFFT-linsi--add to add the sequences from S_1 into C, and our goal was to enable MAFFT-linsi--add to scale to datasets where the total number of sequences in S (backbone plus query sequences) is very large. The key innovation of EMMA is that we break the

sequence-adding problem into multiple smaller sub-problems on which MAFFT-linsi--add can efficiently and accurately run (i.e., divide-and-conquer). We describe this at high level here, providing details in the supplementary materials (but we note that some of the techniques here are adapted from [12] and [16]).

Given the input constraint alignment C, unaligned sequences, and (optional) value for parameter u, EMMA operates as follows:

1. **Step 1: Construct set of constraint sub-alignments.** Compute a tree on C (default: using the maximum likelihood heuristic FastTree2 [19]), and break it into smaller subsets by repeatedly deleting centroid edges (i.e., edges whose removal splits the leaf set into two sets of roughly equal size), and retaining only those subsets that contain at least 50 and no more than q sequences. For each sub-alignment C_i, construct a profile HMM. This produces a set $C_1, C_2, \ldots, C_k$ of sub-alignments, with S_i the sequences in alignment C_i.

2. **Step 2: Define the set of subproblems.** For each query sequence q, assign q to the constraint sub-alignment whose HMM has the best fit, as determined by the bitscore computed by HMMER [3] for that specific query and sub-alignment pair.

   This defines a set of subproblems: the constraint sub-alignment C_i and the set of query sequences Q_i assigned to C_i. For each i, if the total number of sequences $S_i \cup Q_i$ exceeds 500, then minimally divide the query sequences in Q_i into smaller subsets so that each final subproblem has at most 500 sequences.

3. **Step 3: Apply MAFFT-linsi--add to add the query sequences.** For each resultant subproblem, use MAFFT-linsi--add to add all assigned query sequences to the selected constraint sub-alignment. The output of this step is a collection of extended sub-alignments.

4. **Step 4: Merge the extended sub-alignments using transitivity.** Each sub-alignment contains sites that come from the backbone alignment as well as newly introduced sites (representing homologies inferred between query sequence letters through the use of MAFFT-lini--add). The sites from the backbone alignment allow us to merge these subalignments, in the obvious way: if sub-alignments A_1 and A_2 both have sites drawn from site j from the backbone alignment, then these two sites are merged in the output alignment, and the left-to-right ordering of columns in the sub-alignments can be used to define the left-to-right ordering in the output alignment. See Supplementary Materials for additional details and discussion.

Theorem 1. *Given backbone alignment C on set S_0 and query sequences S_1, EMMA outputs an alignment on $S_0 \cup S_1$ that induces C when restricted to S_0.*

Proof. Every subproblem consists of a constraint sub-alignment C' and set of query sequences Q'. MAFFT-linsi--add applied to subproblem (C', Q') is guaranteed to return an extended sub-alignment that induces C' when restricted to the sequences for that set. Since every query sequence is in exactly one sub-alignment, all the extended sub-alignments will be consistent with C, the

constraint alignment given to EMMA. By definition, the transitivity merge cannot undo homologies in any extended sub-alignment nor in the i. Furthermore, because each query sequence and each backbone sequence is in exactly one sub-alignment, the transitivity merge never merges columns in the constraint alignment; hence, it is guaranteed to return an alignment that induces the backbone alignment when restricted to the backbone sequences. □

In addition to guaranteeing that the output of EMMA is consistent with the constraint alignment, the four-step design achieves several properties that are beneficial for alignment accuracy and scalability to large backbone alignments: (1) all runs of MAFFT-linsi--add have at most 500 sequences in total and (2) the division into sub-alignments is based on an estimated phylogeny, so that the sequences in each sub-alignment are likely to be closely related, and (3) each query sequence is assigned to a sub-alignment of sequences for which they are likely to be closely related (based on the bit-score calculation of the fit between the query sequence and the sub-alignment). These properties together make for sub-problems that are small enough for MAFFT-linsi--add to run well on (i.e., we reduce the computational effort) and closely related enough for MAFFT-linsi--add to be highly accurate.

Note that EMMA uses HMMs to determine *which* subset alignment to assign a given query sequence to, but does not use the HMMs to perform the alignment. Instead, the alignment of the query sequence to the sub-alignment is performed using MAFFT-linsi--add in batch mode (i.e., all the query sequences assigned to the same sub-alignment are aligned together), which allows MAFFT-linsi--add to find homologies between letters in query sequences even if the backbone alignment does not have homologies as well.

4 Experimental design

4.1 Overview

Experiment 0 compares MAFFT-linsi--add and MAFFT--add for scalability and accuracy in one model condition, and Experiment 1 is used to set the parameter u in EMMA; these experiments are performed on the training data only. In Experiment 2, we compare EMMA to MAFFT--add (v7.490, run in default mode), MAFFT-linsi--add (v7.490), and WITCH-ng-add (v0.0.2), using the testing datasets (disjoint from the training data). All methods are evaluated for alignment accuracy (described in more detail in Section 4.3) on the added query sequences and running time. Experiments 0 and 1 analyses were limited to 64 GB and 12 hours. Experiment 2 was run with 16 cores and 64 GB memory, with 24 hours of runtime. For MAFFT-linsi--add and MAFFT--add, we allowed 128 GB memory. All experiments were run on the UIUC Campus Cluster. See Supplementary Materials Section S2 for exact commands of all methods, and Sections S3 and S4 for additional details on dataset generation.

4.2 Datasets

We use both simulated and biological datasets of nucleotide and protein to evaluate EMMA. In addition to datasets from prior studies [11, 12, 1], we also generated one new simulated dataset using INDELible [4] and we use two new

biological datasets (Rec and Res). The training datasets were used in Experiments 0 and 1, and all subsequent experiments used the testing data (a separate collection of datasets). Empirical statistics for these datasets can be found in Supplementary Materials Table S1. All datasets are described below, with the new datasets freely available online through the Illinois Data Bank (see Data Availability statement in Section 8).

4.2.1 Simulated datasets

All simulated datasets are based on evolving sequences down model trees with indels, and so all present alignment challenges. As described below, these simulated datasets vary in terms of indel length distribution, rate of evolution, and whether the sequences evolve under selection, as described below. All datasets have at least 1000 sequences.

The INDELible simulated datasets These datasets were generated for this study. We used INDELible [4] to evolve sequence datasets down a tree with a heterogeneous indel (insertion and deletion) model. Under this model, with a small probability, an indel event can be promoted to long indel events, modeling infrequent large gain or loss during the evolutionary process (e.g., domain-level indels). Hence, we name these new model conditions "het" to reflect their heterogeneous indels. Each replicate has 5000 sequences, and the model conditions range in evolutionary rates (and hence alignment difficulty), with model condition 5000M2-het having the highest rate of evolution, 5000M3-het somewhat slower, and 5000M4-het the slowest. We used non-ultrametric model trees. See Supplementary Section S3 for details of the data generation.

RNASim The RNASim million-sequence dataset is from [12] and has been used in prior studies to evaluate alignment methods [16, 22]. In the RNASim simulation, RNA sequences evolve under a biophysical model and under selection, in order to conserve the rRNA structure. In this study, we subsampled 10 replicates, each with 10,000 sequences.

ROSE We included datasets from [11], which were generated using the ROSE [24] software. We used four model conditions (1000M1, 1000M2, 1000M3, and 1000M4) from [11]. Each model condition has 10 replicates and each replicate has 1000 sequences, with an average sequence length of ∼1000 bp. The 1000M1–1000M4 models vary in rate of evolution (with 1000M1 the highest, and reducing rates as the index increases), thus enabling us to evaluat the impact of rate of evolutino on alignment difficulty.

4.2.2 Biological datasets

CRW The Comparative Ribosomal Website [1] (CRW) is a collection of nucleotide datasets with curated alignments based on secondary structure. We include 16S.3, 16S.T, and 16S.B.ALL, three large datasets from the CRW, with 5323, 6350, and 27643 sequences, respectively. These datasets have been used in previous studies [2, 16, 22] and exhibit sequence length heterogeneity. We use the cleaned versions from [22], for which any ambiguity codes or entirely gapped columns are removed.

10AA The "10 AA" dataset is a publicly available collection of large curated protein alignments, originally assembled for the study evaluating PASTA [12], but also used to evaluate multiple sequence alignment methods [16, 2]. These

curated alignments are based on protein structure and range from 320 to 807 sequences, and includes the eight largest BAliBASE datasets [25] and two datasets from [5].

Serine Recombinases (Rec and Res) These datasets were assembled for this study, and are for two domains from serine recombinase. Protein sequences were taken from 350,378 GenBank bacterial and archaeal genome assemblies (supplementary materials) using Prodigal [6]. Serine recombinases were identified using the Pfam HMMs [14] Resolvase (Res) for the catalytic domain, and Recombinase (Rec) for the integrase-specific domain using *hmmsearch* from the HMMER package with the "trusted cutoffs" supplied by Pfam. Standards phiC31 and Bxb1 were added. From the 199,090 unique protein sequences, the Rec and Res domains were separately extracted using the boundaries determined by the HMM hits.

The Rec and Res datasets have reference alignments on a subset of the sequences (i.e., the seed alignment from Pfam), with Rec having 66 and Res 112 seed sequences. Hence, the reference set is much smaller than the entire set of sequences. Alignment accuracy is evaluated only on these specific reference sequences. See Supplementary Section S3 for details of data generation.

4.2.3 Constraint alignment selection

Recall that we have true alignments for the simulated datasets and reference alignments (based on structure) for the biological datasets. However, for the Rec and Res datasets, we have reference alignments only on a subset of the sequences.

The constraint alignment is constructed to be a subset of the sequences on which we have a reference alignment so that we can test the accuracy of the estimated alignment using the reference alignment on the remaining reference sequences. Thus, in Experiment 2, we designed different ways of selecting a subset of the reference sequences to form the constraint alignment. The reference alignment, restricted to the selected sequences, is treated as the backbone alignment (i.e., the constraint alignment), and the remaining sequences are query sequences.

We considered three different techniques for the selection of the backbone sequences, for which a constraint alignment would be provided by the user. The first two scenarios assume that the selection is random across the gene family, which would likely be the approach used when designing a constraint alignment that was most representative of the family. For these two scenarios, we vary the number of sampled sequences from "large" (at 25% of the full set of sequences) to "small" (at 10% of the sequences). On the other hand, in practice it is likely to be the case that biological knowledge that is most usable in obtaining a curated alignment would be for very closely related sequences, such as those that are found in a relatively small clade. This motivates the third scenario.

1. *Large random subset:* We begin by determining the set of reference sequences that are "full-length", which means their length is within 25% of the median length. From the set of F full-length sequences, we randomly select $\min(1000, 0.25F)$ sequences.

2. *Small random subset:* The protocol is identical to the large random subset protocol, except that we randomly select $\min(100, 0.1F)$ full-length

sequences. However, if this number is less than 10, we just pick 10 sequences at random.

3. *Large clade-based subset:* Here we restrict to sequences from a clade (as defined on either the true tree for simulated data or a maximum likelihood tree for the biological data). We pick the clade that contains as close as possible to 25% of the reference sequences; if there are ties, we pick one randomly as the backbone. If the number of sequences in the dataset is more than 4000, then instead, we pick a clade of size 1000 (or as close to 1000 as possible).

4.3 Evaluation criteria

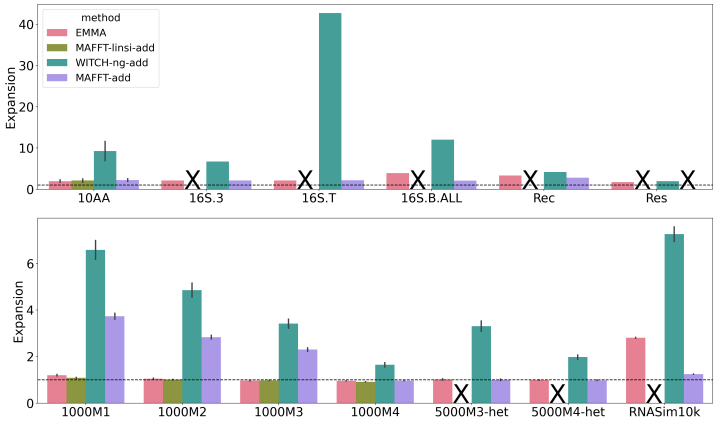
We evaluate alignment methods with respect to wall-clock running time and alignment error. For alignment error, we compare estimated and reference/true alignments, restricted to the query sequences from the reference alignment. We report SPFN, SPFP, and expansion score, defined as follows. The expansion score is the length of the estimated alignment divided by the length of the true or reference alignment; thus, the best result is 1.0, alignments that have expansion scores less than 1.0 are said to be "over-aligned", while alignments that have expansion scores greater than 1.0 are "under-aligned". The SPFN and SPFP error metrics are based on homologies, i.e., pairs of letters (nucleotides or amino acids) found in the same column in the true or estimated alignment. The sum-of-pairs false-negative (SPFN) rate is the fraction of homologies in the reference alignment that are missing in the estimated alignment. The sum-of-pairs false-positive (SPFP) rate is the fraction of pairs of homologies in the estimated alignment that are missing in the reference alignment. These metrics are calculated using FastSP [13].

5 Results

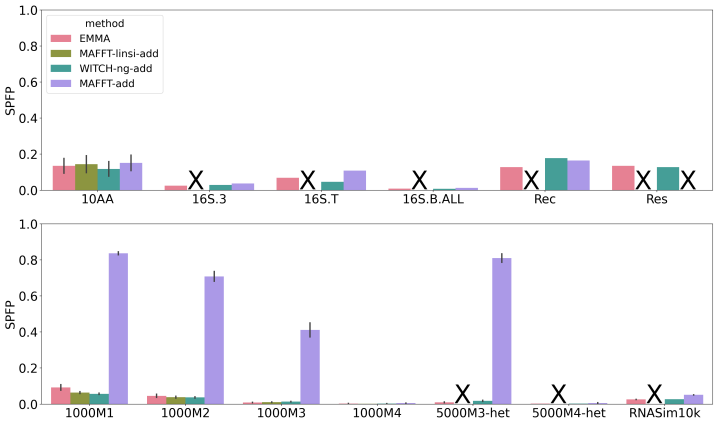
5.1 Experiment 1: Designing EMMA

Experiment 1 explores the impact of the parameter u in the divide-and-conquer strategy in EMMA. The parameter u specifies the upper bound on the size of sub-alignments we use in the decomposition strategy in EMMA, which become the backbone alignments when we run MAFFT-linsi--add. We explore the impact of this parameter by varying u between 100, 200, and 400, using the training dataset 5000M2-het.

Results for EMMA with different settings of u on all sequences of 5000M2-het (10 replicates) are presented in the supplementary materials (Supplementary Figure S2). For a fixed backbone size, the differences between EMMA variants are negligible for both SPFN and SPFP. Based on these experiments, we choose $u = 100$ as our default for EMMA because this variant gives comparable alignment accuracy as other settings while having a relatively lower running time.



(a) Expansion score



(b) SPFP

Figure 3: **Expansion score (top) and SPFP (bottom) on large random backbones.** In each sub-figure, the top panel denotes biological datasets, and the bottom panel denotes simulated datasets (note the change in the y-axis range for expansion scores between the top and bottom panels). The horizontal dashed line indicates a perfect expansion score of 1. MAFFT-linsi--add failed to finish within 24 hours for datasets except for 10AA and ROSE 1000M1-4 and was not run on Rec and Res (due to their very large size, we knew it would not complete within the allowed time), and MAFFT--add encountered out-of-memory issues on the Res dataset (failed runs, or runs that were not attempted, are marked with "X"). The expansion score is the length of the estimated alignment normalized by the length of the reference or true alignment; optimal is 1.0 and values above 1.0 indicate alignments that are too long (and so are under-aligned). Note the extremely high expansion scores for WITCH-ng-add on nearly all datasets, indicating *extreme under-alignment*. Note also the very high SPFP score for MAFFT--add on the simulated datasets with high rates of evolution, indicating *over-alignment* (inferring many false homologies).

5.2 Experiment 2: Evaluating EMMA to other alignment methods

5.2.1 Expansion scores and SPFP

The expansion score is the length of the estimated alignment normalized by the length of the reference or true alignment. As seen in the Supplementary Materials Table S4, MAFFT-linsi--add had the best expansion scores, coming very close to 1.0 in most cases. All the other methods, however, produced alignments that were longer than the true alignment (and hence were under-aligning), as indicated by expansion scores being greater than 1. MAFFT--add under-aligned the least of these methods, followed closely by EMMA, and WITCH-ng-add has by far the largest expansion scores. With respect to expansion score, WITCH-ng is the outlier, as its expansion scores were excessively high. This level of under-alignment also means that interpreting its SPFP rates is difficult, since under-alignment also reduces false positives.

With these observations, we now consider the SPFP scores. Full results are shown in Supplementary Materials Figures S3–S5, and the results for large random backbones are provided in Figure 3. WITCH-ng-add typically had among the best SPFP scores, but MAFFT-linsi-add and EMMA sometimes had better SPFP scores. Thus, the improved SPFP scores achieved by WITCH-ng-add are partially the result of its very high degree of under-alignment.

EMMA and MAFFT-linsi--add both had SPFP scores that are relatively close. There is an advantage to MAFFT-linsi--add over EMMA on the simulated ROSE datasets (1000M1–1000M4) but not on the 10AA datasets (the only biological datasets on which MAFFT-linsi--add completed).

We also see that MAFFT--add had higher SPFP scores than MAFFT-linsi--add on the datasets on which MAFFT-linsi--add could run. Given that MAFFT--add also had worse expansion scores than MAFFT-linsi--add, this indicates that MAFFT--add was generally inferior to MAFFT-linsi--add (consistent with Experiment 0, and also with other prior studies). Interestingly, although MAFFT--add had lower expansion scores than EMMA, it had higher SPFP scores. Given these trends, we omit MAFFT--add from the rest of the study (though see Supplementary Materials for comparisons involving MAFFT--add).

We note that the SPFP scores, other than for WITCH-ng-add, were all relatively close (and generally low), making SPFP scores not very informative. Therefore, for the remainder of this study, we focus on SPFN, noting that 1-SPFN is the same as *recall* (or SP-Score).

5.2.2 SPFN alignment error on large random backbones

Figure 4 shows the SPFN of the three sequence-adding methods when adding to large random backbone alignments. The most striking observation is that MAFFT-linsi--add only succeeded in completing the datasets with at most 1000 sequences (i.e., the 10AA biological datasets and the ROSE simulated datasets with 1000 sequences); on all the other datasets, it failed to complete within the allowed time (24 hours). However, for those datasets that it completed, it had very good SPFN scores: better than the other methods for high rates of evolution (1000M1, 1000M2), in third place but still close to the best on the 10AA datasets, and tied for best on the ROSE datasets with low rates of evolution (1000M3 and 1000M4). We also see EMMA matching or better than

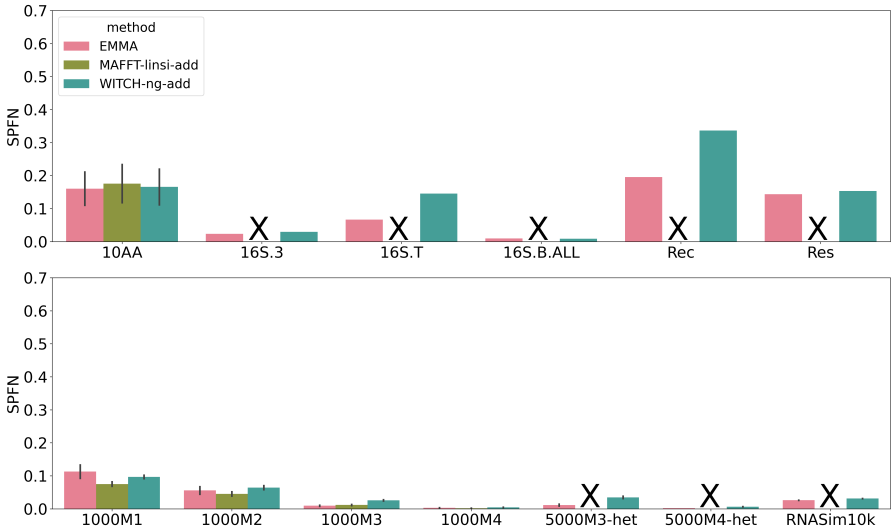


Figure 4: **SPFN of EMMA, WITCH-ng-add, and MAFFT-linsi--add when adding sequences into large random backbone alignments.** The top panel denotes biological datasets, and the bottom panel denotes simulated datasets. Error bars indicate standard errors. MAFFT-linsi--add only completed within the allowed 24 hours on datasets with at most 1000 sequences, and we did not run it on Rec or Res due to their large numbers of sequences (failed runs, or runs that were not attempted, are marked with "X"). Results on other model conditions, including MAFFT--add, can be found in Supplementary Figures S6–S8.

WITCH-ng-add on nearly all datasets, except that WITCH-ng-add is better than EMMA on 1000M1.

5.2.3 SPFN alignment error on small random backbones

As with large random backbones, MAFFT-linsi--add only succeeded in finishing on the datasets with at most 1000 sequences (Figure 5), but on those datasets, it was among the most accurate methods. The relative accuracy between the three methods is very similar to that displayed on the large backbones, so that EMMA was generally at least as accurate as WITCH-ng-add and in some cases was much more accurate, while MAFFT-linsi--add had the best accuracy of all methods when it can run). Finally, all methods had somewhat larger error rates in this condition than when placing into a large random backbone.

5.2.4 SPFN alignment error on clade-based backbones

As with the other conditions, when adding sequences into a clade-based backbone alignment, MAFFT-linsi--add failed to complete on any dataset with more than 1000 sequences. However, in many respects, the results for the clade-based backbone are different than for the random backbones (Figure 6). The most noteworthy difference is that error rates increased for all methods when

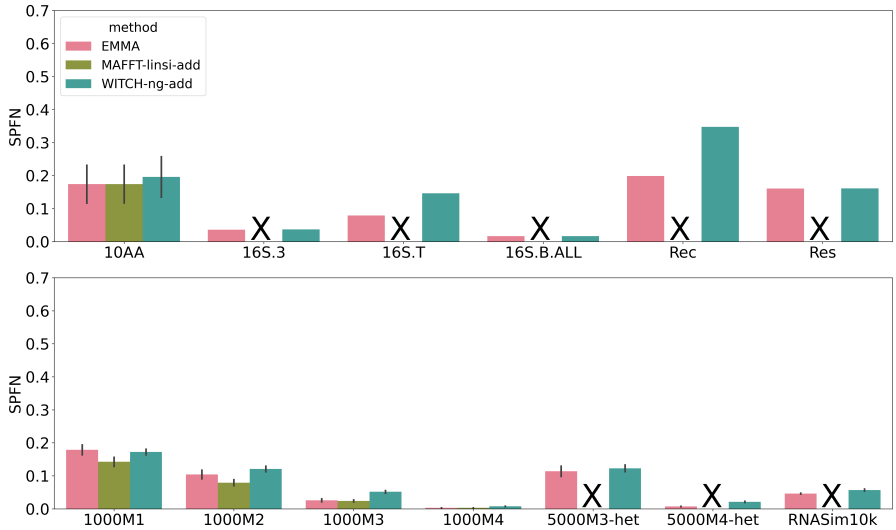


Figure 5: **SPFN of EMMA, WITCH-ng-add, and MAFFT-linsi--add when adding sequences into small random backbone alignments.** The top panel denotes biological datasets, and the bottom panel denotes simulated datasets. Error bars indicate standard errors. MAFFT-linsi--add failed to finish within 24 hours for datasets except for 10AA and ROSE 1000M1-4, and we did not run it on Rec or Res due to their large numbers of sequences (failed runs, or runs that were not attempted, are marked with "X"). Comparisons with MAFFT--add can be found in Supplementary Figure S7.

given clade-based backbones instead of random backbones, but the increase was largest for WITCH-ng-add. The gaps between methods increased, with WITCH-ng-add no longer tying for best in any condition. EMMA strictly improved on WITCH-ng-add by a large margin on all datasets other than the 16S.B.ALL and Res datasets, where nevertheless EMMA had a lower SPFN than WITCH-ng-add.

5.2.5 Computational performance

MAFFT-linsi--add and MAFFT--add are the only methods that failed to complete on at least one of the datasets within the allowed runtime (24 hours), using 16 cores and 128 GB memory. Supplementary Materials Section S4 shows that both methods reported out-of-memory issues or crashes. Specifically, MAFFT-linsi--add had an out-of-memory issue when attempting to analyze the INDELible 5000M2-het training dataset and allowed 64 GB. We also noted that MAFFT--add had an out-of-memory issue on the Res dataset with ∼186K sequences, even when allowed 128 GB of memory. This is perhaps not surprising, since MAFFT--add also computes the $n \times n$ pairwise distance matrix, where n is the number of sequences in total (backbone sequences and query sequences together), and shows that MAFFT--add as well as MAFFT-linsi--add both have large memory requirements.

The runtimes of all methods on all datasets are shown in Supplementary

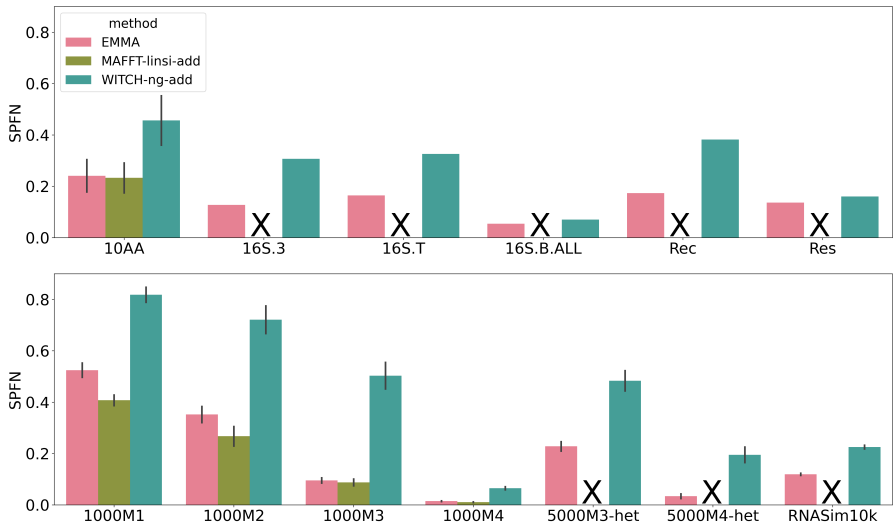


Figure 6: **SPFN of EMMA, WITCH-ng-add, and MAFFT-linsi--add when adding sequences into clade-based backbone alignments.** The top panel denotes biological datasets, and the bottom panel denotes simulated datasets. Error bars indicate standard errors. MAFFT-linsi--add failed to finish within 24 hours for datasets except for 10AA and ROSE 1000M1-4, and we did not run it on Rec or Res due to their large numbers of sequences (failed runs, or runs that were not attempted, are marked with "X"). Comparisons with MAFFT--add can be found in Supplementary Figure S8.

Materials, Figures S9–S11, and allow a more detailed evaluation. MAFFT-linsi--add was the slowest method on all the datasets on which it could run (but as noted above, it failed to complete on any dataset with more than 1000 sequences). MAFFT--add was overall the fastest method, but WITCH-ng-add was faster in some cases (e.g., on the Rec dataset, under all three backbone selection strategies). The comparison between WITCH-ng-add and EMMA generally favors WITCH-ng-add, but there are cases where EMMA was faster than WITCH-ng-add (e.g., 16S.B.ALL given a large random backbone).

Across all the datasets in this study, the maximum runtime used by EMMA and WITCH-ng-add are 13.5 and 10.5 hours, respectively. Given that WITCH-ng-add and EMMA completed so quickly using under 64 GB memory on these datasets and that the datasets had up to ∼186,000 sequences, it is reasonable to say that WITCH-ng-add and EMMA are acceptable with respect to speed and scalability.

In contrast, MAFFT-linsi--add was the slowest and least scalable. We also tried to give MAFFT-linsi--add three days (72 hours instead of 24 hours) to run on 16S.3 and 16S.T for the small random backbone and large clade-based backbone, but it still was not able to complete. Finally, we consider MAFFT--add to be reasonably scalable, since it only had computational problems with the very largest dataset (Res), and it is by far the fastest method we tested.

6 Discussion

Alignment error rates for all methods were lowest for the large random backbone, somewhat higher for the small random backbone, and then substantially higher for the clade-based backbone model conditions. The increase in error rates were most noticeable for WITCH-ng-add, indicating the vulnerability of WITCH-ng-add to the sampling strategy used for the backbone alignment. This vulnerability is also most noteworthy for the datasets with high rates of evolution, and almost not detectable for the model condition with the lowest rate of evolution (i.e., 1000M4). These trends are essentially as expected, since the constraint alignment is used as a model of the entire gene family, and dense sampling throughout the family (i.e., the large random backbone) ensures a better model in contrast to sparse random sampling (the small random backbone) or sampling within a clade (clade-based backbones).

Our study showed that EMMA and MAFFT-linsi--add both provide very good accuracy (using SPFN, i.e., recall) for adding sequences into a user-provided constraint alignment (i.e., backbone alignment), but MAFFT-linsi--add fails to run on datasets with more than 1000 sequences in this experiment. While there are conditions where EMMA, WITCH-ng-add, and MAFFT-linsi--add all had good accuracy, EMMA had better accuracy than WITCH-ng-add when the constraint alignment was either small or is from a clade in the reference tree.

It is easy to see why EMMA can scale to large datasets, as its design ensures that no problem is ever too big for MAFFT-linsi--add to complete. The differences in accuracy between the methods are very interesting, and somewhat more difficult to explain. However, for the three datasets (10AA, 1000M1–1000M4) on which MAFFT-linsi--add can run, it is generally at least as accurate as EMMA, but only improves in accuracy reliably on datasets with high rates of evolution (i.e., 1000M1 and sometimes 1000M2) but not on 10AA, which had a lower evolutionary diameter. This suggests that the divide-and-conquer strategy used in EMMA is not optimized for datasets with high evolutionary rates.

We also see that WITCH-ng-add and EMMA vary in terms of where each is more accurate, with WITCH-ng-add clearly less accurate for many biological datasets but more accurate in some conditions. However, WITCH-ng-add is always less accurate than EMMA on the clade-based backbone conditions, whereas when using random backbones it is only more accurate than EMMA on the 1000M1 model condition (which has the highest rate of evolution). Furthermore, the improvement in WITCH-ng-add over EMMA for that 1000M1 condition is small (and perhaps not noteworthy, except possibly for the large random backbone). Thus, EMMA has an advantage over WITCH-ng-add in general, except when the backbone alignment is a random backbone. This suggests that WITCH-ng-add is not as robust to the backbone selection as EMMA, and may require dense sampling.

The explanation for WITCH-ng-add's lack of robustness to backbone sampling strategy is that WITCH-ng-add uses the HMM-based approach, and so can only detect homologies that use match states in these HMMs (which correspond to sites in the backbone alignment), while EMMA uses MAFFT-linsi--add and can detect homologies between query sequences that do not use sites in the backbone alignment; this advantage to EMMA (and also to MAFFT-linsi--add) seems to be lost when the rate of evolution is sufficiently high.

7 Conclusions

This study presented EMMA, a new method for adding sequences into existing constraint alignments (also called backbone alignments). Prior to EMMA, several methods had been developed that could enable this functionality, including MAFFT--add (and its accurate setting, MAFFT-linsi--add) and WITCH-ng-add. However, of these, MAFFT--add run in default mode does not have adequate accuracy on large datasets, and MAFFT-linsi--add is limited (due to runtime requirements) to only moderate-sized datasets (i.e., in our experiments, it failed on datasets with more than 1000 sequences). In contrast, EMMA and WITCH-ng-add can scale to very large datasets, including the condition where the backbone alignment has 180,000 sequences and there are 20,000 sequences to be added. EMMA and WITCH-ng-add have similar accuracy under some conditions. EMMA is often more accurate when the backbone is small or drawn from a clade, and WITCH-ng-add was only more accurate when the backbone alignment is for a large random subset, and even then only for high rates of evolution. Thus, EMMA shows greater robustness to the backbone selection strategy than WITCH-ng-add. Interestingly we noted that WITCH-ng-add has lower SPFP (indicating better precision) than EMMA, but achieves this largely at the cost of significantly under-aligning (i.e., producing much longer sequence alignments than the true or reference alignment).

This study suggests several directions for future work. Perhaps most importantly, EMMA should be designed in such a way that its accuracy improves for the highest rates of evolution, which is the only condition where it was less accurate than WITCH-ng-add. One possibility is to consider changes to the sub-alignment construction to ensure larger subsets, as our study shows that MAFFT-linsi--add does not degrade in accuracy with large datasets, even though it becomes more computationally expensive. Hence, there may be opportunities for improved accuracy, even if runtime and memory usage increase.

We should also explore other biological datasets to document the accuracy of these methods under a wider range of real-world conditions. EMMA is not yet optimized for speed, and a more careful parallel implementation may provide a substantial speed-up. Finally, EMMA could be studied as the second stage of the standard two-stage pipeline protocol used by UPP, WITCH, and WITCH-ng, for *de novo* multiple sequence alignment. Given the high accuracy of these two-stage methods for aligning datasets with high sequence length heterogeneity, this is likely to provide improved accuracy.

8 Data Availability

The INDELible, Rec, and Res datasets were created for this study. The Rec and Res datasets are available in the Illinois Data Bank at `https://doi.org/10.13012/B2IDB-2567453_V1`. Pfam seed sequences are included for both datasets (66 and 112 sequences for Rec and Res, respectively). The INDELible datasets (5000M2-het, 5000M3-het, and 5000M4-het) are available at `https://databank.illinois.edu/datasets/IDB-3974819`. All other datasets are from prior studies, and are available in public repositories.

9 Acknowledgements

CS and TW were supported by the US National Science Foundation grant 2006069 (to TW). K.P.W., C.S., and B.L. were supported by the Laboratory Directed Research and Development program at Sandia National Laboratories, which is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

References

[1] Jamie J. Cannone, Sankar Subramanian, Murray N. Schnare, James R. Collett, Lisa M. D'Souza, Yushi Du, Brian Feng, Nan Lin, Lakshmi V. Madabusi, Kirsten M. Müller, Nupur Pande, Zhidi Shang, Nan Yu, and Robin R. Gutell. The Comparative RNA Web (CRW) Site: an online database of comparative sequence and structure information for ribosomal, intron, and other RNAs. *BMC Bioinformatics*, 3(1):2, January 2002. doi:10.1186/1471-2105-3-2.

[2] Kodi Collins and Tandy Warnow. PASTA for proteins. *Bioinformatics*, 34(22):3939–3941, November 2018. doi:10.1093/bioinformatics/bty495.

[3] Robert D. Finn, Jody Clements, and Sean R. Eddy. HMMER web server: interactive sequence similarity searching. *Nucleic Acids Research*, 39(Web Server issue):W29–W37, July 2011. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3125773/, doi:10.1093/nar/gkr367.

[4] William Fletcher and Ziheng Yang. INDELible: A Flexible Simulator of Biological Sequence Evolution. *Molecular Biology and Evolution*, 26(8):1879–1888, August 2009. doi:10.1093/molbev/msp098.

[5] Gregory B. Gloor, Louise C. Martin, Lindi M. Wahl, and Stanley D. Dunn. Mutual Information in Protein Multiple Sequence Alignments Reveals Two Classes of Coevolving Positions. *Biochemistry*, 44(19):7156–7165, May 2005. Publisher: American Chemical Society. doi:10.1021/bi050293e.

[6] Doug Hyatt, Gwo-Liang Chen, Philip F. LoCascio, Miriam L. Land, Frank W. Larimer, and Loren J. Hauser. Prodigal: prokaryotic gene recognition and translation initiation site identification. *BMC Bioinformatics*, 11(1):119, March 2010. doi:10.1186/1471-2105-11-119.

[7] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, August 2021. Number: 7873 Publisher: Nature Publishing Group. doi:10.1038/s41586-021-03819-2.

[8] Kazutaka Katoh and Martin C. Frith. MAFFT – a multiple alignment program for amino acid or nucleotide sequences. https://mafft.cbrc.jp/alignment/software/addsequences.html. Accessed: 2022-05-20.

[9] Kazutaka Katoh and Martin C. Frith. Adding unaligned sequences into an existing alignment using MAFFT and LAST. *Bioinformatics*, 28(23):3144–3146, December 2012. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3516148/, doi:10.1093/bioinformatics/bts578.

[10] Baqiao Liu and Tandy Warnow. WITCH-NG: efficient and accurate alignment of datasets with sequence length heterogeneity. *Bioinformatics Advances*, 3(1):vbad024, January 2023. doi:10.1093/bioadv/vbad024.

[11] Kevin Liu, Sindhu Raghavan, Serita Nelesen, C. Randal Linder, and Tandy Warnow. Rapid and Accurate Large-Scale Coestimation of Sequence Alignments and Phylogenetic Trees. *Science*, 324(5934):1561–1564, June 2009. Publisher: American Association for the Advancement of Science. URL: http://www.science.org/doi/10.1126/science.1171243, doi:10.1126/science.1171243.

[12] Siavash Mirarab, Nam Nguyen, Sheng Guo, Li-San Wang, Junhyong Kim, and Tandy Warnow. PASTA: Ultra-Large Multiple Sequence Alignment for Nucleotide and Amino-Acid Sequences. *J Comput Biol*, 22(5):377–386, May 2015. doi:10.1089/cmb.2014.0156.

[13] Siavash Mirarab and Tandy Warnow. FASTSP: linear time calculation of alignment accuracy. *Bioinformatics*, 27(23):3250–3258, December 2011. doi:10.1093/bioinformatics/btr553.

[14] Jaina Mistry, Sara Chuguransky, Lowri Williams, Matloob Qureshi, Gustavo A Salazar, Erik L L Sonnhammer, Silvio C E Tosatto, Lisanna Paladin, Shriya Raj, Lorna J Richardson, Robert D Finn, and Alex Bateman. Pfam: The protein families database in 2021. *Nucleic Acids Research*, 49(D1):D412–D419, January 2021. doi:10.1093/nar/gkaa913.

[15] David A Morrison. Multiple sequence alignment for phylogenetic purposes. *Australian Systematic Botany*, 19(6):479–539, 2006.

[16] Nam-phuong D. Nguyen, Siavash Mirarab, Keerthana Kumar, and Tandy Warnow. Ultra-large alignments using phylogeny-aware profiles. *Genome Biology*, 16(1):124, June 2015. doi:10.1186/s13059-015-0688-z.

[17] Minhyuk Park, Stefan Ivanovic, Gillian Chu, Chengze Shen, and Tandy Warnow. UPP2: fast and accurate alignment of datasets with fragmentary sequences. *Bioinformatics*, 39(1):btad007, January 2023. doi:10.1093/bioinformatics/btad007.

[18] Minhyuk Park and Tandy Warnow. HMMerge: an ensemble method for multiple sequence alignment. *Bioinformatics Advances*, page vbad052, 2023.

[19] Morgan N. Price, Paramvir S. Dehal, and Adam P. Arkin. FastTree 2 – Approximately Maximum-Likelihood Trees for Large Alignments. *PLOS ONE*, 5(3):e9490, March 2010. URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0009490, doi:10.1371/journal.pone.0009490.

[20] Bruce A Shapiro, Yaroslava G Yingling, Wojciech Kasprzak, and Eckart Bindewald. Bridging the gap in RNA structure prediction. *Current Opinion in Structural Biology*, 17(2):157–165, 2007.

[21] Chengze Shen, Minhyuk Park, and Tandy Warnow. WITCH: Improved Multiple Sequence Alignment Through Weighted Consensus Hidden Markov Model Alignment. *Journal of Computational Biology*, May 2022. Publisher: Mary Ann Liebert, Inc., publishers. URL: https://www.liebertpub.com/doi/abs/10.1089/cmb.2021.0585, doi:10.1089/cmb.2021.0585.

[22] Chengze Shen, Paul Zaharias, and Tandy Warnow. MAGUS+eHMMs: improved multiple sequence alignment accuracy for fragmentary sequences. *Bioinformatics*, 38(4):918–924, February 2022. doi:10.1093/bioinformatics/btab788.

[23] Vladimir Smirnov and Tandy Warnow. MAGUS: Multiple sequence Alignment using Graph clUStering. *Bioinformatics*, 37(12):1666–1672, June 2021. doi:10.1093/bioinformatics/btaa992.

[24] J. Stoye, D. Evers, and F. Meyer. Rose: generating sequence families. *Bioinformatics (Oxford, England)*, 14(2):157–163, 1998. doi:10.1093/bioinformatics/14.2.157.

[25] Julie D. Thompson, Benjamin Linard, Odile Lecompte, and Olivier Poch. A Comprehensive Benchmark Study of Multiple Sequence Alignment Methods: Current Challenges and Future Perspectives. *PLOS ONE*, 6(3):e18093, March 2011. Publisher: Public Library of Science. URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0018093, doi:10.1371/journal.pone.0018093.

[26] Andres Veidenberg, Alan Medlar, and Ari Löytynoja. Wasabi: An Integrated Platform for Evolutionary Sequence Analysis and Data Visualization. *Molecular Biology and Evolution*, 33(4):1126–1130, April 2016. doi:10.1093/molbev/msv333.

[27] Kazunori D. Yamada, Kentaro Tomii, and Kazutaka Katoh. Application of the MAFFT sequence alignment program to large data—reexamination of the usefulness of chained guide trees. *Bioinformatics*, 32(21):3246–3251, November 2016. doi:10.1093/bioinformatics/btw412.