Algorithms for
Molecular Biology

RESEARCH Open Access

EMMA: a new method for computing multiple sequence alignments given a constraint subset alignment

Chengze Shen[1], Baqiao Liu[1], Kelly P. Williams[2] and Tandy Warnow[1*]

Abstract

Background 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.

Results 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 constraint 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.

Conclusions EMMA is a new tool that provides high accuracy and scalability for adding sequences into an existing alignment.

Keywords Multiple sequence alignment, Constraint alignment, MAFFT

Background

Adding sequences to a multiple sequence alignment

Multiple sequence alignment (MSA) is a crucial precursor to many downstream biological analyses, such as phylogeny estimation [1], RNA structure prediction [2], protein structure prediction [3], 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 a given alignment without allowing the given alignment to change; for this reason, the given alignment is referred to as a "constraint 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, adding sequences into the existing alignment avoids the need to recompute the

*Correspondence:
Tandy Warnow
warnow@illinois.edu
[1] Computer Science, University of Illinois, Urbana-Champaign, 201 N. Goodwin Ave, Urbana 61801, IL, USA
[2] Sandia National Laboratories, 7011 East Ave., Livermore 94550, CA, USA

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 [4], UPP2 [5], WITCH [6], WITCH-ng [7], HMMerge [8], and MAFFT-sparsecore [9]. 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 [4]. 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 [10] perhaps the most well-known. However, multiple sequence alignment methods that operate in two steps—i.e., they first extract and align the backbone sequences and then add the remaining sequences into this backbone alignment—can be modified to enable sequences to be added into a user-provided alignment.

HMM-based methods

Many of the methods that have been developed for adding sequences into a given 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, which we call "query sequences," 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 [4], UPP2 [5], WITCH [6], WITCH-ng [7], and HMMerge [8]. 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 the performance of WITCH-ng-add as it has been shown to be at least as accurate and generally faster than WITCH-add, and both are, in turn, at least as accurate as UPP-add and UPP2-add. Finally, HMMerge-add is slower than WITCH-ng-add, and so is omitted from this study. See Additional file 1: 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.

MAFFT--add and MAFFT-linsi--add

MAFFT--add [10] 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 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 alignment 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.

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 [12].

Limitations of HMM-based methods

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.

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.

Figure 1 gives an example of a backbone alignment and query sequences, comparing MAFFT-linsi--add and UPP-add. Note that MAFFT-linsi--add finds homologous nucleotides in the query sequences that do not correspond to homologs in the backbone alignment, while UPP-add fails to recover these. This is an inherent limitation of HMM-based methods and motivates the development of EMMA.

New method: EMMA

Comparing MAFFT--add and MAFFT-linsi--add

In a preliminary study (Experiment 0), we compared MAFFT--add and MAFFT-linsi--add to determine their relative accuracy and computational performance, especially as the number of sequences increased. We used 5000M2, a simulated dataset developed for this study (see "Datasets" section), for this comparison. For this experiment, the backbone alignment has 1000 sequences and we varied the total number of added sequences from 100 to 2000.

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.

EMMA's algorithmic design

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 a backbone alignment or as a constraint alignment.

EMMA was designed to enable MAFFT-linsi--add to scale to datasets where the total number of sequences in S (backbone plus query sequences) is very large. To achieve this, we use a divide-and-conquer strategy where we create a number of problems with smaller numbers of backbone sequences and query sequences on which MAFFT-linsi--add can efficiently and accurately run. We then show how we can combine the results into a solution to the overall problem of adding sequences into the backbone alignment. Some of the techniques in EMMA are adapted from PASTA [14] and UPP [4], as described below and in the supplementary materials.

EMMA has two algorithmic parameters, l and u, that govern the decomposition strategy; our study set default values for these parameters, but they can be supplied by the user. Given the input backbone alignment C, unaligned sequences, and (optional) values for parameters l and u, EMMA operates as follows:

1. Step 1: Construct set of constraint subalignments: Compute a tree on C (default: using the maximum likelihood heuristic FastTree2 [15]), 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 l and at most u sequences. Each subset of the tree defines a suba-

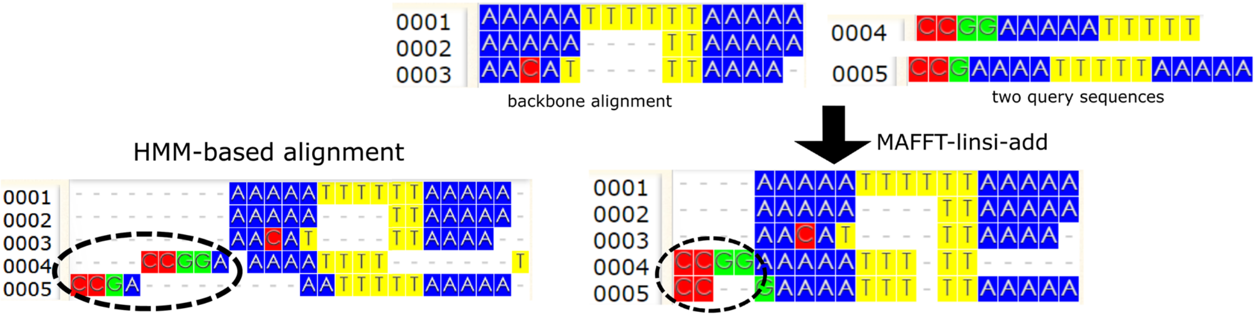

Fig. 1 Comparing MAFFT-linsi--add to UPP-add We ran MAFFT-linsi--add and UPP-add on a backbone alignment and two query sequences. The initial two sites in the MAFFT-linsi--add alignment each contains two letters from the query sequences, thus indicating it detects them as homologous to each other, although there are no letters from the backbone alignment in these two sites. This is impossible for any method that represents the backbone alignment by one or more HMMs and then adds sequences to the backbone alignment using the HMMs. Thus, UPP-add places these initial letters of the query sequences into separate sites. Visualization is done with WASABI [11]

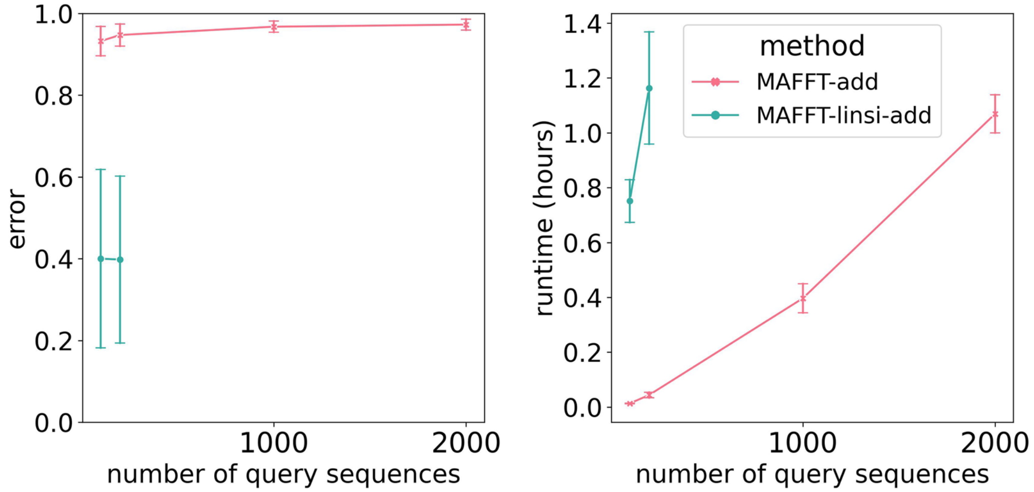

Fig. 2 Experiment 0: MAFFT-linsi--add scalability issues. Alignment error (left) and runtime in hours (right) of MAFFT--add and MAFFT-linsi--add for adding 100, 200, 1000, or 2000 sequences to a 1000-taxon backbone alignment computed using MAGUS [13] on the INDELible 5000M2-het dataset with 5000 sequences. 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 h. 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 show the same trends and can be found in Additional file 1: Fig. S1

lignment C_i of C (defined by the set S_i of sequences in the subset). For each subalignment $C_i, i = 1 \ldots k$, construct a profile HMM.

2. Step 2: Define the set of subproblems: For each query sequence q, assign q to the constraint subalignment whose HMM has the best fit, as determined by the adjusted bitscore (introduced in [6]), which provides an estimate of the probability that the given HMM generates the given query sequence. This defines a set of subproblems: the constraint subalignment C_i and the set of query sequences Q_i assigned to C_i. For each i, if the total number of sequences in $S_i \cup Q_i$ exceeds 500, then partition the query sequences in Q_i into the smallest number of subsets needed in order to have at most 500 sequences in each subset, and ensure that have the subsets are as balanced in size as possible.
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 subalignment. The output of this step is a collection of extended subalignments (i.e., an alignment that agrees with the constraint alignment C_i but has some query sequences added).
4. Step 4: Merge the extended subalignments using transitivity: Each subalignment 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-linsi--add). The sites from the backbone alignment allow us to merge these subalignments in an obvious way: if subalignments 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. Furthermore, the left-to-right ordering of columns in the subalignments can be used to define the left-to-right ordering in the output alignment. Transitivity was first used in PASTA and has been used in many subsequent methods, such as UPP, WITCH, and WITCH-ng. See Additional file 1: Fig. S2 for an example of transitivity merging.

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 subalignment C' and a set of query sequences Q'. MAFFT-linsi--add applied to subproblem (C', Q') is guaranteed to return an extended subalignment that induces C' when restricted to the sequences for that set. Since every query sequence is in exactly one subalignment, all the extended subalignments will be consistent with C, the constraint alignment given to EMMA. By definition, the transitivity merge cannot undo homologies in any extended subalignment nor in the i. Furthermore, because each query sequence and each backbone sequence are in exactly one subalignment, 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 a large number of sequences: (1) all runs of MAFFT-linsi--add have at most 500 sequences in total and (2) the division into subalignments is based on an estimated phylogeny so that the sequences in each subalignment are likely to be closely related, and (3) each query sequence is assigned to a subalignment 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 subalignment). These properties together make for subproblems 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 but does not use the HMMs to perform the alignment. Instead, the alignment of the query sequence to the subalignment is performed using MAFFT-linsi--add in batch mode (i.e., all the query sequences assigned to the same subalignment 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.

Experimental design

Overview

Experiment 1 sets the parameters l and u in EMMA; these experiments are performed on the training data. 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 running time as well as alignment error (see below). Experiment 1 analyses were limited to 64 GB and 12 h. Experiment 2 was run with 16 cores and 64 GB memory, with 24 h of runtime. For MAFFT-linsi--add and MAFFT--add, we allowed 128 GB memory. All experiments were run on the UIUC Campus Cluster. See Additional file 1: Section S3 for exact commands of all methods, and Additional file 1: Section S4 for additional details on dataset generation.

Alignment error

For alignment error, we compare the estimated to the reference/true alignment, restricted to the (added) query sequences from the reference alignment. We report SPFN, SPFP, and expansion scores, which are 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 [16].

Datasets

We use both simulated and biological datasets of nucleotide and protein to evaluate EMMA. In addition to datasets from prior studies [14, 17, 18], we also generated one new simulated dataset using INDELible [19] 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 Additional file 1: Table S1. All datasets are described below, with the new datasets freely available online through the Illinois Data Bank (see Data Availability statement).

Simulated datasets

All simulated datasets are based on evolving sequences down model trees with indels so that all sequence alignments involve gaps. As described below, these simulated datasets vary in terms of indel length distribution, rate of evolution, and whether the sites evolve identically and independently or under selection. All simulated datasets have at least 1000 sequences.

ROSE: We included datasets from [17], which were generated using the ROSE [20] software. We used four model conditions (1000M1, 1000M2, 1000M3, and 1000M4) from [17]. 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 evaluate the impact of rates of evolution on alignment difficulty. All site evolution is non-ultrametric, and the sites evolve identically and independently.

The INDELible simulated datasets: These datasets were generated for this study. We used INDELible [19] 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, and all sites evolved identically and independently. See Additional file 1: Section S4 for details of the data generation and Additional file 1: Tables S2 and S3 for the parameter values used in the simulations.

RNASim: The RNASim million-sequence dataset is from [14] and has been used in prior studies to evaluate alignment methods [4, 21]. In the RNASim simulation, RNA sequences evolve under a biophysical model and under selection in order to conserve the rRNA structure. Thus, the sites do not evolve independently. In this study, we subsampled 10 replicates, each with 10,000 sequences.

Biological datasets

CRW: The Comparative Ribosomal Website [18] (CRW) is a collection of nucleotide datasets with curated alignments based on secondary structure. We include 16 S.3, 16 S.T, and 16 S.B.ALL, three large datasets from the CRW, with 5323, 6350, and 27,643 sequences, respectively. These datasets have been used in previous studies [4, 21, 22] and exhibit sequence length heterogeneity. We use the cleaned versions from [21], 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 [14], but also used to evaluate multiple sequence alignment methods [4, 22]. These curated alignments are based on protein structure and range from 320 to 807 sequences and include the eight largest BAliBASE datasets [23] and two datasets from [24].

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 (Additional file 1: Section S4) using Prodigal [25]. Serine recombinases were identified using the Pfam HMMs [26] 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 having 112 seed sequences. Hence, the reference set for each dataset is much smaller than the entire set of sequences. Alignment error is evaluated only on these specific reference sequences.

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. Alignment error is evaluated only on sequences for which we have reference alignments, which means that for the Rec and Res datasets we can only evaluate alignment error on a subset of the sequences.

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 scenarios for the selection of the backbone sequences, for which a constraint alignment would be provided by the user. The first two scenarios select sequences randomly from across the assume that the backbone sequences are selected randomly from the input. 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). The third scenario is designed for the case where a curated alignment is available for a small and likely closely related set of sequences found within a clade.

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), selecting a clade that has at most 1000 sequences, and that comes as close as possible to 25% of the reference sequences. In the case of ties, we pick the clade randomly,

Results

Experiment 1: Designing EMMA

Experiment 1 is the experiment where we set the algorithmic parameters, l and u, for EMMA. We vary l between 10 and 50 and u between l and 100, using the INDELiBLE 5000M2-het model condition, which has a high rate of evolution and so makes for difficult alignments. We use the *large random subset* scenario so that the backbone alignment contains 1000 randomly selected sequences, and the remaining 4000 sequences are query sequences.

Results for EMMA with different settings of (l, u) on 5000M2-het (10 replicates) are presented in Additional file 1: Fig. S3. The lowest error is found when $l = 10$, and for this setting of l, u then has little impact. However, the setting for u does impact runtime, with the fastest runtime found (across all settings) with $l = 10$ and $u = 25$. Based on this experiment, we set these as the default settings for l and u.

Experiment 2: Evaluating EMMA to other sequence-adding methods

Here, we show comparisons of EMMA using the default settings of $l = 10$ and $u = 25$ to WITCH-ng-add and MAFFT-linsi--add. We show a comparison to MAFFT--add in default mode only for a limited set of analyses since its error rates were much higher than the other methods; see Additional file 1: Figs. S4–S14 for full results, including MAFFT--add.

Expansion scores and SPFP

The expansion score is the estimated alignment length normalized by the reference or true alignment length. As seen in the Additional file 1: Table S4, MAFFT-linsi--add had the best expansion scores, coming 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. For the 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. Results for large random backbones are provided in Fig. 3, and results for the two other conditions are shown in Additional file 1: Figs. S4 and S5. 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 Additional file 1: Figs. S3–S8 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).

SPFN alignment error on large random backbones

Figure 4 shows the SPFN of the three sequence-adding methods (i.e., EMMA, WITCH-ng-add, and MAFFT-linsi--add) 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 h). For all datasets, EMMA had the best SPFN score, better than both MAFFT-linsi--add and WITCH-ng. We also note that its advantage in SPFN was more noticeable on datasets with high rates of evolution (1000M1, 1000M2).

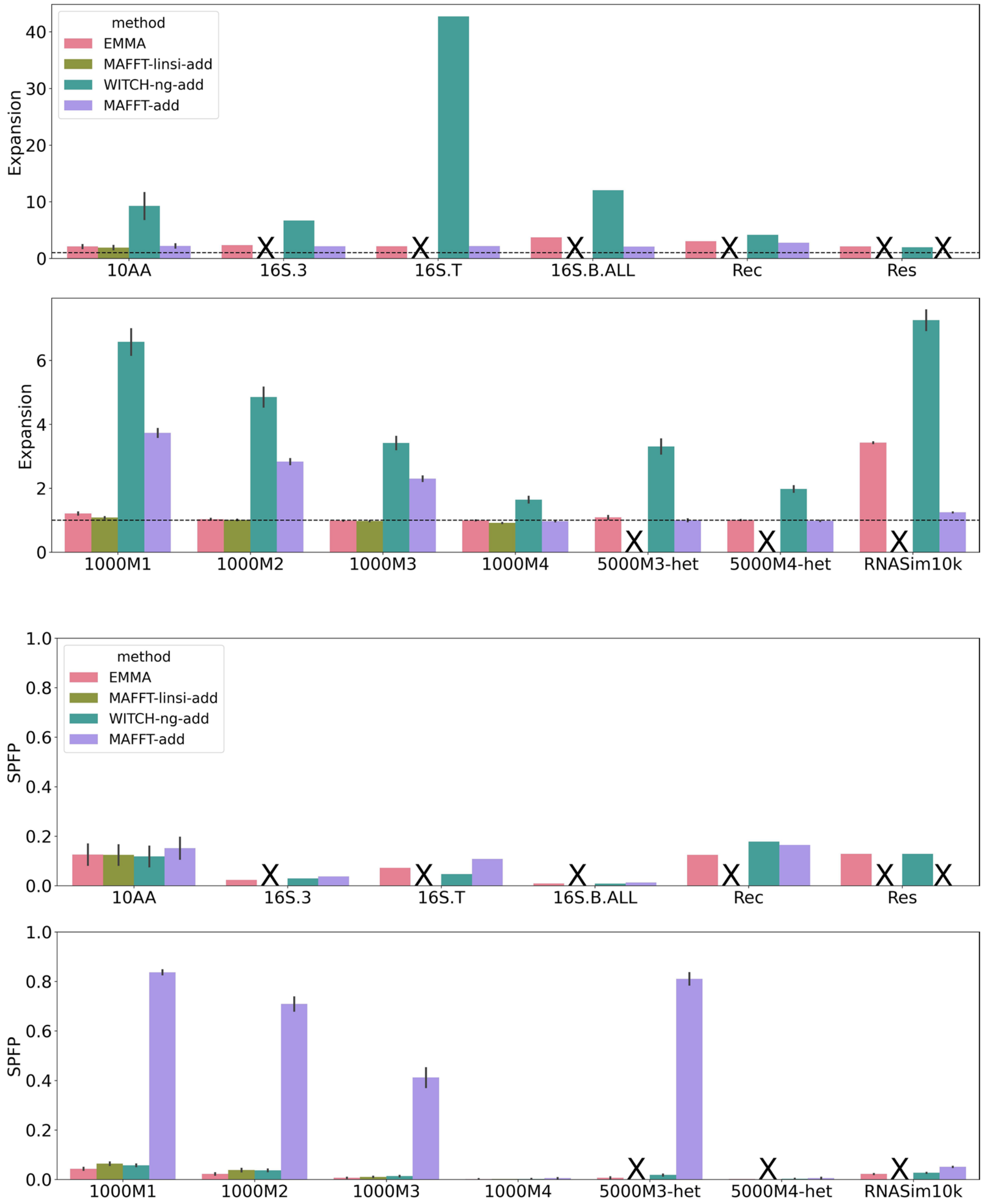

Fig. 3 Expansion score (top) and SPFP (bottom) on large random backbones. In each subfigure, the top panel denotes biological datasets, and the bottom panel denotes simulated datasets (note the change in the y-axis range for expansion scores between the top and bottom panels). The horizontal dashed line indicates a perfect expansion score of 1. MAFFT-linsi--add failed to finish within 24 h 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)

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 (Fig. 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: EMMA was still the most accurate, followed by MAFFT-linsi--add when it can run, then by WITCH-ng. Finally, the gaps in accuracy between methods were smaller than when placing into

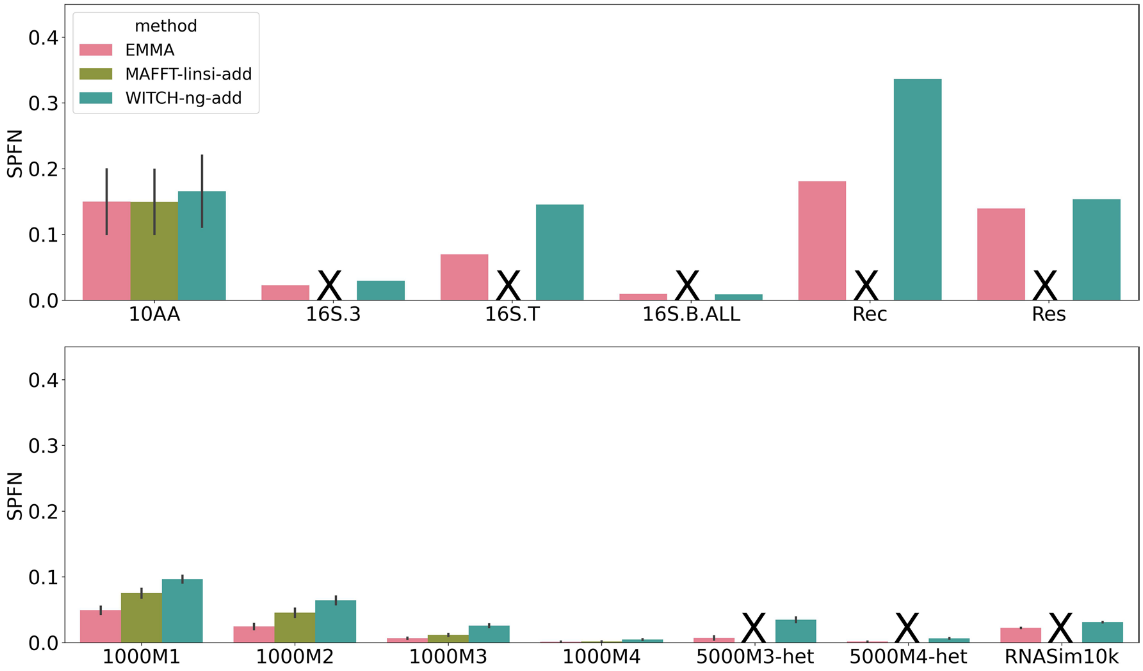

Fig. 4 SPFN of EMMA, WITCH-ng-add, and MAFFT-linsi--add when adding sequences into large random backbone alignments. The top panel denotes biological datasets, and the bottom panel denotes simulated datasets. Error bars indicate standard errors. MAFFT-linsi--add only completed within the allowed 24 h 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"). Comparisons with MAFFT--add can be found in Additional file 1: Fig. S7. SPFN results for each of the 10AA datasets are in Additional file 1: Fig. S12

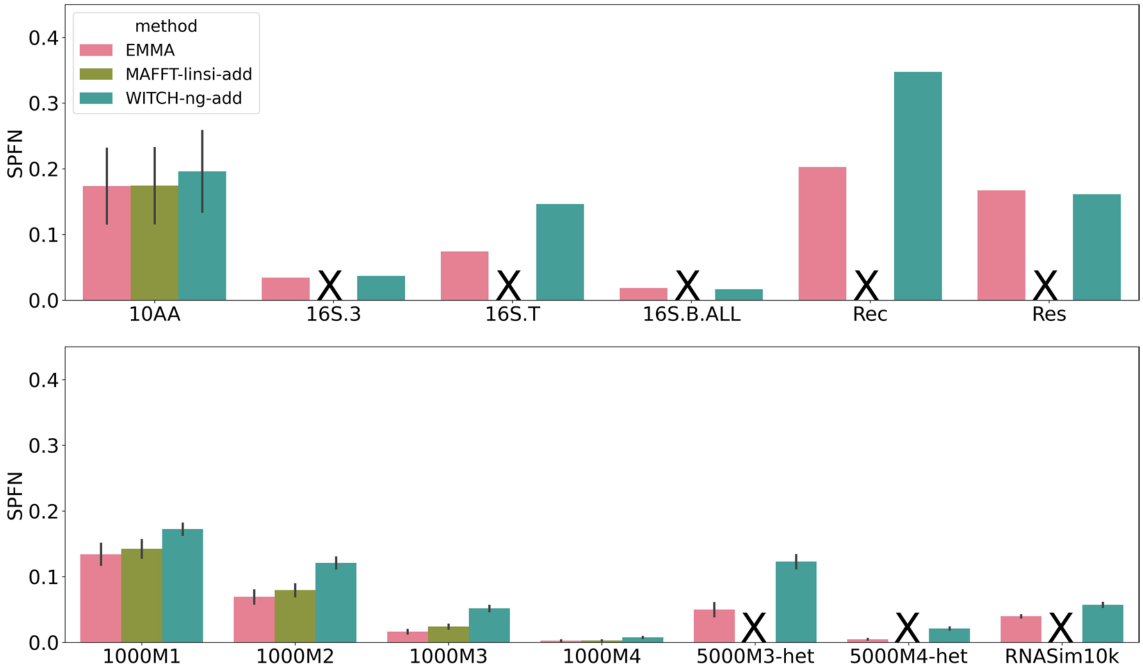

Fig. 5 SPFN of EMMA, WITCH-ng-add, and MAFFT-linsi--add when adding sequences into small random backbone alignments. The top panel denotes biological datasets, and the bottom panel denotes simulated datasets. Error bars indicate standard errors. MAFFT-linsi--add failed to finish within 24 h for datasets 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 Additional file 1: Fig. S8. SPFN results for each of the 10AA datasets are in Additional file 1: Fig. S13

large random backbones, while the absolute error rates were higher.

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 (Fig. 6). The most noteworthy difference is that error rates increased for all methods when given clade-based backbones instead of random backbones, but the increase was largest for WITCH-ng-add. MAFFT-linsi--add is clearly the most accurate of all the methods on the 1000M1, 1000M2, and 1000M3 datasets and then ties for best on 1000M4. EMMA is strictly more accurate than WITCH-ng-add, usually by a large margin, on all datasets. Nevertheless, both EMMA and WITCH-ng-add have high errors on datasets with high rates of evolution (1000M1, 5000M3-het).

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 h), using 16 cores and 128 GB memory. Additional file 1: Section S5 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.

Figure 7 compares the methods given a large random backbone. MAFFT-linsi--add was the slowest method on all the datasets when it could run (as noted above, it failed to complete on any dataset with more than 1000 sequences). WITCH-ng-add was overall the next slowest method, but it was faster than EMMA on the 10AA datasets (which are the smallest datasets we examined, all below 1000 sequences). MAFFT--add was overall the fastest method, but WITCH-ng-add was faster on the Rec dataset, and EMMA was faster on the 5000M3-het dataset. Overall, EMMA was in between WITCH-ng-add and MAFFT--add in terms of running time on these datasets.

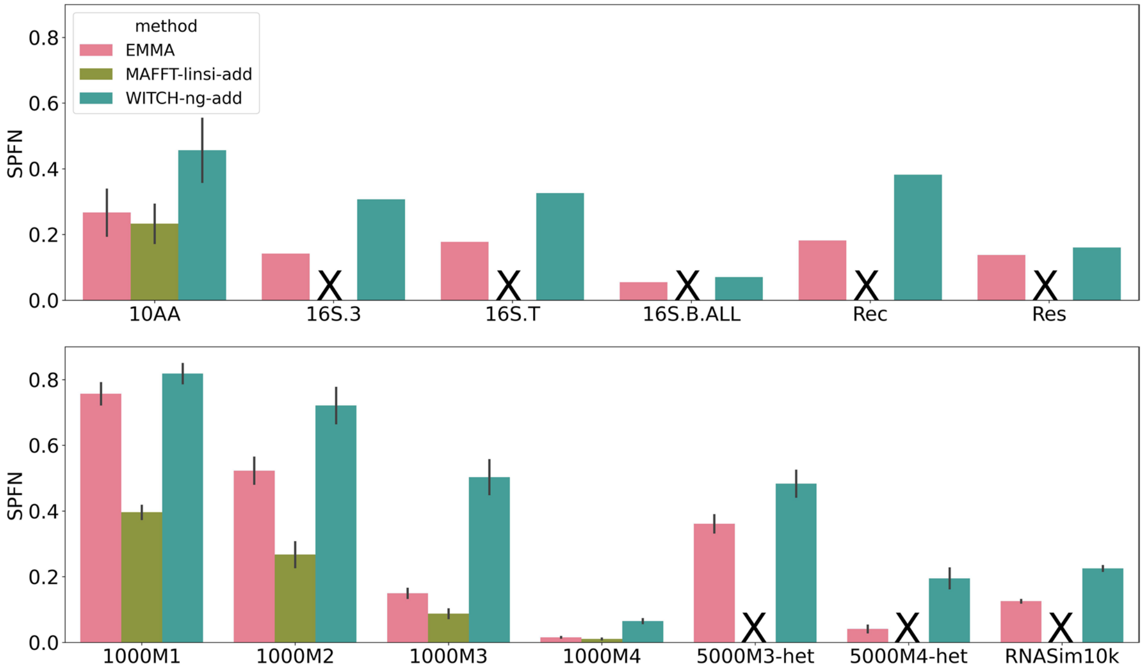

Fig. 6 SPFN of EMMA, WITCH-ng-add, and MAFFT-linsi--add when adding sequences into clade-based backbone alignments. The top panel denotes biological datasets, and the bottom panel denotes simulated datasets. Error bars indicate standard errors. MAFFT-linsi--add failed to finish within 24 h for datasets 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 Additional file 1: Fig. S9. SPFN results for each of the 10AA datasets are in Additional file 1: Fig. S14

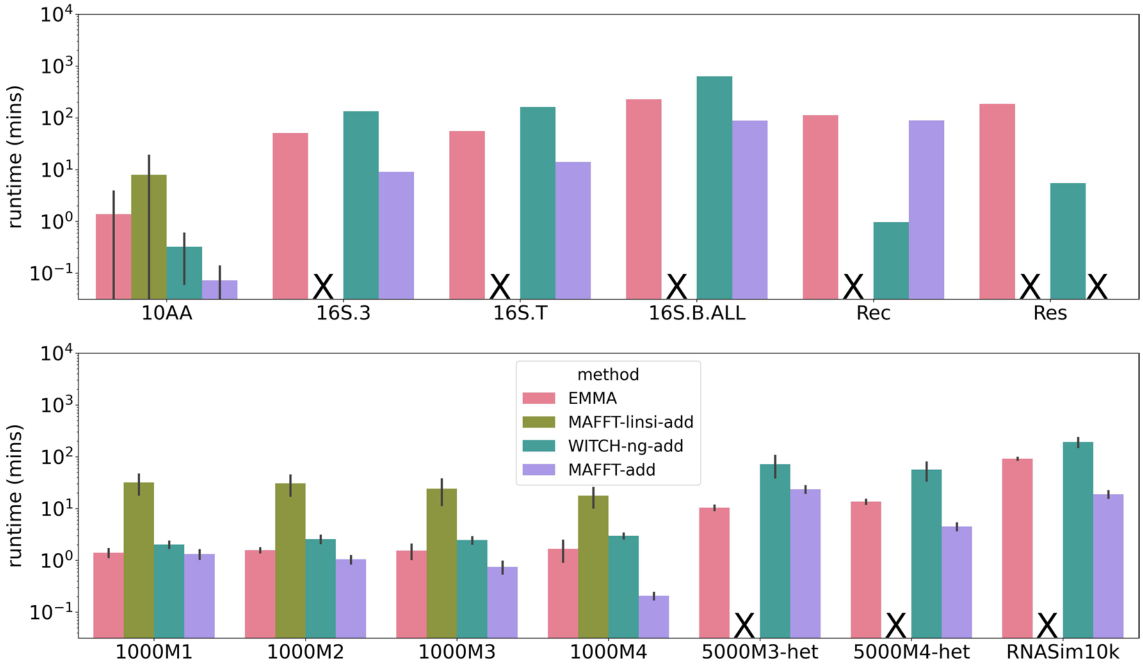

Fig. 7 Experiment 2: Runtime (log-scale) in minutes when adding to large random backbone alignments. The top panel denotes biological datasets, and the bottom panel denotes simulated datasets. Except for the datasets with at most 1000 sequences, MAFFT-linsi--add failed to finish within 24 h and was not attempted on Rec and Res due to their large number of sequences. MAFFT--add encountered out-of-memory issues on the Res dataset. Failed runs, or runs not attempted due to dataset size, are marked with "X". Error bars indicate standard deviation

The runtimes of all methods on the other conditions (small random backbones and clade-based backbones) are shown in Additional file 1: Figs. S10 and S11. On these datasets, we see somewhat similar trends, with EMMA faster than MAFFT-linsi--add on the datasets where MAFFT-linsi--add can run, but now EMMA is slower than both WITCH-ng-add and MAFFT--add.

The only methods that succeeded in completing all datasets were EMMA and WITCH-ng-add, and the maximum runtimes used by EMMA and WITCH-ng-add were 12.2 and 10.5 h, 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.

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

Discussion

The four methods we have evaluated—EMMA, MAFFT--add, MAFFT-linsi--add, and WITCH-ng-add, vary in their ability to run on large datasets (i.e., scalability), runtime, and accuracy (here focused on SPFN). The relative performance is summarized in Table 1, and discussed further below.

Our study showed clear differences in scalability, with EMMA and WITCH-ng-add the most scalable (completing on all datasets within the time and memory limits), MAFFT--add the next most scalable (failing to complete on one dataset), and MAFFT-linsi--add the least scalable (failing on all but the datasets with at most 1000 sequences). It is easy to see why EMMA can scale to the

Table 1 Overall performance of the tested methods

Criterion	EMMA	WITCH-ng-add	MAFFT--add	MAFFT-linsi--add
Scalability	All	All	All but largest	≤ 1000 sequences
SPFN-random	1	2	3	1
SPFN-clade	2	3	4	1
Speed	2	2	1	3

For scalability, we indicate which study datasets the method completed on

For SPFN and speed, we indicate the relative position (1 is best, 2 is second best, etc.)

largest datasets, as the decomposition strategy and transitivity merge are both very fast, and the only potentially computationally intensive part is when it runs MAFFT-linsi--add on subsets of query sequences to add them to subsets of the backbone alignment. However, by design, the subproblems are all small (at most $u = 25$ backbone sequences and at most 500 query sequences). This design strategy allows EMMA to complete on all the datasets in our study. It is also easy to see why MAFFT-linsi--add is limited in scalability since its algorithmic design involves a quadratic runtime.

Accuracy differences were also apparent but were generally very small when the rates of evolution were low enough. This also is as expected since prior studies have shown differences in alignment error often are very small or negligible when sequence identity is high, which occurs under low rates of evolution (e.g., [17, 27]).

When comparing accuracy under higher rates of evolution, we see that the selection strategy for the backbone sequences—i.e., whether they were selected randomly or from within a clade and the number of backbone sequences—impacts both absolute and relative alignment error. In particular, all methods have better accuracy with random sampling than with clade-based sampling and also have better accuracy when there are more sequences in the backbone alignment. These trends are expected since the constraint alignment is used as a model of the family, and dense sampling throughout the family (i.e., the large random backbone) ensures a better model than sparse random sampling (i.e., the small random backbone) or sampling from just within a clade.

The comparison between EMMA and WITCH-ng-add with respect to accuracy favors EMMA across all sampling strategies, although the two methods have nearly identical accuracy for some conditions. The accuracy advantage of EMMA is more noticeable on datasets with high rates of evolution, such as ROSE 1000M1, 1000M2, and INDELible 5000M3-het. Since WITCH-ng-add relies on using HMMs created from the backbone to align the queries, whereas EMMA uses MAFFT-linsi--add to align the queries, the improvement in accuracy for EMMA is likely to be due to the greater sensitivity of MAFFT-linsi--add for detecting homologies than the HMM-based approach within WITCH-ng-add.

The comparison between MAFFT-linsi--add and EMMA, although restricted to just those datasets on which MAFFT-linsi--add could run (i.e., the datasets with at most 1000 sequences), reveals the following trends. When given a large random backbone, EMMA matches or improves on MAFFT-linsi--add under all conditions, but has an advantage given on the ROSE 1000M1 and 1000M2 conditions, and then matches MAFFT-linsi--add for the 1000M3 and 1000M4 conditions. The same relative performance is seen for the small random backbone, but with a smaller difference between the methods. Thus, the rate of evolution impacts the relative accuracy of the two methods, favoring EMMA when the rate of evolution is high, suggesting that MAFFT-linsi--add accuracy is impacted more substantially by the rate of evolution than EMMA. This trend is consistent with prior studies showing that MAFFT-linsi is sensitive to rate of evolution, an observation that led to the design of SATé [17] and its descendent methods [13, 14, 28].

Interestingly, MAFFT-linsi--add improves on EMMA when the sequences are drawn from a clade, with a substantial improvement under the higher rate of evolution. In other words, when the backbone is drawn from a clade, the algorithmic strategy in EMMA—which applies MAFFT-linsi--add to subsets that contain at most 25 backbone sequences from the clade – is inferior to using the entire backbone. This is clearly a condition where the EMMA divide-and-conquer strategy is not well-suited.

In understanding the conditions where MAFFT-linsi--add is more accurate than EMMA, it is helpful to understand that the homologies between query sequences found by EMMA are either found directly by MAFFT-linsi-add on a small subset (with at most 500 sequences) or inferred when combining these extended alignments using transitivity. Thus, the relative accuracy of EMMA and MAFFT-linsi--add is impacted by the divide-and-conquer strategy and, in particular, the size of the subsets in the decomposition of the backbone tree. Given that our default setting currently sets $u = 25$, this choice creates subsets that have very few backbone sequences (at most 25) and then adds query sequences, but not so many as to exceed 500 total sequences. This is potentially why MAFFT-linsi-add can be more accurate than EMMA since MAFFT-linsi-add can find homologies between all pairs of query sequences, whereas EMMA (which applies MAFFT-linsi-add to small subsets) can only achieve this within the subsets it produces. Our Experiment 1, where we set the parameters $l = 10$ and $u = 25$, was based on the INDELible 5000M2 dataset, which has a high rate of evolution. It seems likely that other settings for l and u might lead to improved accuracy when the model condition has a lower rate of evolution since MAFFT-linsi--add's accuracy improves as the rate of evolution decreases. In other words, EMMA's algorithmic design aimed to reduce runtime and improve scalability, but for some challenging inputs (e.g., clade-based backbones) where MAFFT-linsi--add can run, the algorithmic design of EMMA may reduce accuracy relative to MAFFT-linsi--add. Hence, there is room for improvement in designing EMMA.

However, MAFFT-linsi--add could not complete the study datasets with more than 1000 sequences given the limitations in computational resources we imposed. In contrast, EMMA was able to complete all study datasets. Thus, while MAFFT-linsi--add had a clear accuracy advantage over EMMA for the clade-based backbones (though not on the random backbones), EMMA has a computational advantage over MAFFT-linsi--add in being able to scale to larger datasets.

Given the impact of clade-based sampling for the backbone sequences, it is important to remember that EMMA is never identical to MAFFT-linsi--add, even when analyzing small subsets. When we run MAFFT-linsi--add on the datasets with at most 1000 sequences, all the query sequences are considered together when added to the backbone alignment. In contrast, by construction, EMMA only adds query sequences to subsets of the backbone alignment with at most u sequences, and in our study, we set $u = 25$ by default. From a computational standpoint, this strategy allows EMMA to scale to very large datasets. From an accuracy perspective, however, the impact clearly depends on how the backbone sequences were sampled. EMMA has an accuracy advantage over MAFFT-linsi--add when the backbone sequences are randomly sampled and there is a high rate of evolution and matches MAFFT-linsi--add for accuracy when the backbone sequences are randomly sampled with a lower rate of evolution. However, as we have seen, this strategy is poor when the backbone sequences are sampled from a clade, and the rate of evolution is high.

This observation also indicates that two-phase multiple sequence alignment methods that operate by selecting and aligning a set of sequences for the backbone and then adding in the remaining sequences should—if possible—select the backbone sequences from across the evolutionary tree for the dataset to obtain the best representative model of the gene family.

Conclusions

This study presented EMMA, a new method for adding sequences into existing constraint alignments (also called backbone alignments). EMMA uses a divide-and-conquer strategy to enable MAFFT-linsi--add, a highly accurate version of MAFFT--add, to scale to large datasets. By itself, MAFFT-linsi--add was unable to complete using the computational resources on the datasets we studied with more than 1000 sequences, but EMMA succeeded in completing with even fewer resources on the largest dataset we studied, with more than 180,000 sequences.

Our study shows that EMMA has comparable or better alignment accuracy than MAFFT--add and WITCH-ng-add under all the conditions tested, and also comparable or better accuracy than MAFFT-linsi--add when the backbone sequences are selected randomly. However, our study also showed that when the backbone sequences are selected from a clade and the dataset is sufficiently small, then MAFFT-linsi--add is more accurate than EMMA. This finding indicates clearly that the design of EMMA needs to be reconsidered for the case where the user provides a curated alignment for a closely related group of sequences contained within a clade and wishes to add additional sequences that are distantly related to the backbone sequences.

Thus, future work should investigate appropriate modifications to the divide-and-conquer strategy in EMMA to address the case where the backbone sequences are distantly related to the query sequences. EMMA alignment accuracy could potentially be improved with a re-alignment stage to see if sets of sites could be merged together through the detection of additional homologies between query sequences. 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/UPP2, 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.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s13015-023-00247-x.

Additional file 1. Supplementary text, tables, and figures.

Author contributions

CS developed the EMMA software, performed the analyses, interpreted the data, and drafted the manuscript. BL provided simulated datasets. KPW provided biological datasets and interpreted the data.. TW conceived of the approach, designed the study, interpreted the data, wrote the final draft, and supervised the research. All authors read and approved the final manuscript.

Funding

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.

Availability of data and materials

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. The subsampled RNASim datasets (10,000 sequences for each replicate) are available at https://doi.org/10.13012/B2IDB-4194451_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. The 10AA datasets are available at https://sites.

google.com/eng.ucsd.edu/datasets/alignment/pastaupp. The three CRW datasets are available at https://databank.illinois.edu/datasets/IDB-2419626. The ROSE datasets are available at https://sites.google.com/eng.ucsd.edu/datasets/alignment/sate-i.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Received: 14 September 2023 Accepted: 19 November 2023
Published online: 07 December 2023

References

1. Morrison DA. Multiple sequence alignment for phylogenetic purposes. Aust Syst Bot. 2006;19(6):479–539.
2. Shapiro BA, Yingling YG, Kasprzak W, Bindewald E. Bridging the gap in RNA structure prediction. Curr Opin Struct Biol. 2007;17(2):157–65.
3. Jumper J, Evans R, Pritzel A, Green T, Figurnov M, Ronneberger O, Tunyasuvunakool K, Bates R, Žídek A, Potapenko A, Bridgland A, Meyer C, Kohl SAA, Ballard AJ, Cowie A, Romera-Paredes B, Nikolov S, Jain R, Adler J, Back T, Petersen S, Reiman D, Clancy E, Zielinski M, Steinegger M, Pacholska M, Berghammer T, Bodenstein S, Silver D, Vinyals O, Senior AW, Kavukcuoglu K, Kohli P, Hassabis D. Highly accurate protein structure prediction with AlphaFold. Nature. 2021;596(7873):583–9. https://doi.org/10.1038/s41586-021-03819-2.
4. Nguyen NpD, Mirarab S, Kumar K, Warnow T. Ultra-large alignments using phylogeny-aware profiles. Genome Biol. 2015;16(1):124. https://doi.org/10.1186/s13059-015-0688-z.
5. Park M, Ivanovic S, Chu G, Shen C, Warnow T. UPP2: fast and accurate alignment of datasets with fragmentary sequences. Bioinform. 2023;39(1):007. https://doi.org/10.1093/bioinformatics/btad007.
6. Shen C, Park M, Warnow T. WITCH: improved multiple sequence alignment through weighted consensus hidden Markov model alignment. J Comput Biol. 2022. https://doi.org/10.1089/cmb.2021.0585.
7. Liu B, Warnow T. WITCH-NG: efficient and accurate alignment of datasets with sequence length heterogeneity. Bioinform Adv. 2023;3(1):024. https://doi.org/10.1093/bioadv/vbad024.
8. Park M, Warnow T. HMMerge: an ensemble method for multiple sequence alignment. Bioinform Adv. 2023;3:vbad052.
9. Yamada KD, Tomii K, Katoh K. Application of the MAFFT sequence alignment program to large data-reexamination of the usefulness of chained guide trees. Bioinformatics. 2016;32(21):3246–51. https://doi.org/10.1093/bioinformatics/btw412.
10. Katoh K, Frith MC. Adding unaligned sequences into an existing alignment using MAFFT and LAST. Bioinformatics. 2012;28(23):3144–6. https://doi.org/10.1093/bioinformatics/bts578.
11. Veidenberg A, Medlar A, Löytynoja A. Wasabi: an integrated platform for evolutionary sequence analysis and data visualization. Mol Biol Evol. 2016;33(4):1126–30. https://doi.org/10.1093/molbev/msv333.
12. Katoh K, Frith MC. MAFFT – a multiple alignment program for amino acid or nucleotide sequences. https://mafft.cbrc.jp/alignment/software/addsequences.html. Accessed 20 May 2022.
13. Smirnov V, Warnow T. MAGUS: Multiple sequence Alignment using Graph clUStering. Bioinformatics. 2021;37(12):1666–72. https://doi.org/10.1093/bioinformatics/btaa992.
14. Mirarab S, Nguyen N, Guo S, Wang L-S, Kim J, Warnow T. PASTA: ultra-large multiple sequence alignment for nucleotide and amino-acid sequences. J Comput Biol. 2015;22(5):377–86. https://doi.org/10.1089/cmb.2014.0156.
15. Price MN, Dehal PS, Arkin AP. FastTree 2—approximately maximum-likelihood trees for large alignments. PLoS ONE. 2010;5(3):9490. https://doi.org/10.1371/journal.pone.0009490.
16. Mirarab S, Warnow T. FASTSP: linear time calculation of alignment accuracy. Bioinformatics. 2011;27(23):3250–8. https://doi.org/10.1093/bioinformatics/btr553.
17. Liu K, Raghavan S, Nelesen S, Linder CR, Warnow T. Rapid and accurate large-scale coestimation of sequence alignments and phylogenetic trees. Science. 2009;324(5934):1561–4. https://doi.org/10.1126/science.1171243.
18. Cannone JJ, Subramanian S, Schnare MN, Collett JR, D'Souza LM, Du Y, Feng B, Lin N, Madabusi LV, Müller KM, Pande N, Shang Z, Yu N, Gutell RR. The Comparative RNA Web (CRW) Site: an online database of comparative sequence and structure information for ribosomal, intron, and other RNAs. BMC Bioinform. 2002;3(1):2. https://doi.org/10.1186/1471-2105-3-2.
19. Fletcher W, Yang Z. INDELible: a flexible simulator of biological sequence evolution. Mol Biol Evol. 2009;26(8):1879–88. https://doi.org/10.1093/molbev/msp098.
20. Stoye J, Evers D, Meyer F. Rose: generating sequence families. Bioinformatics (Oxford, England). 1998;14(2):157–63. https://doi.org/10.1093/bioinformatics/14.2.157.
21. Shen C, Zaharias P, Warnow T. MAGUS+eHMMs: improved multiple sequence alignment accuracy for fragmentary sequences. Bioinformatics. 2022;38(4):918–24. https://doi.org/10.1093/bioinformatics/btab788.
22. Collins K, Warnow T. PASTA for proteins. Bioinformatics. 2018;34(22):3939–41. https://doi.org/10.1093/bioinformatics/bty495.
23. Thompson JD, Linard B, Lecompte O, Poch O. A comprehensive benchmark study of multiple sequence alignment methods: current challenges and future perspectives. PLoS ONE. 2011;6(3):18093. https://doi.org/10.1371/journal.pone.0018093.
24. Gloor GB, Martin LC, Wahl LM, Dunn SD. Mutual information in protein multiple sequence alignments reveals two classes of coevolving positions. Biochemistry. 2005;44(19):7156–65. https://doi.org/10.1021/bi050293e.
25. Hyatt D, Chen G-L, LoCascio PF, Land ML, Larimer FW, Hauser LJ. Prodigal: prokaryotic gene recognition and translation initiation site identification. BMC Bioinform. 2010;11(1):119. https://doi.org/10.1186/1471-2105-11-119.
26. Mistry J, Chuguransky S, Williams L, Qureshi M, Salazar G, Sonnhammer ELL, Tosatto SCE, Paladin L, Raj S, Richardson LJ, Finn RD, Bateman A. Pfam: The protein families database in 2021. Nucleic Acids Res. 2021;49(D1):412–9. https://doi.org/10.1093/nar/gkaa913.
27. Sievers F, Wilm A, Dineen D, Gibson TJ, Karplus K, Li W, Lopez R, McWilliam H, Remmert M, Söding J, et al. Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. Mol Syst Biol. 2011;7(1):539.
28. Liu K, Warnow TJ, Holder MT, Nelesen SM, Yu J, Stamatakis AP, Linder CR. SATé-II: very fast and accurate simultaneous estimation of multiple sequence alignments and phylogenetic trees. Syst Biol. 2011;61(1):90–90. https://doi.org/10.1093/sysbio/syr095.

Publisher's Note