Title	Making automated multiple alignments of very large numbers of protein sequences
Authors(s)	Sievers, Fabian; Dineen, David; Wilm, Andreas; Higgins, Desmond G
Publication date	2013-02-21
Publication information	Bioinformatics, 29 (8): 989-995
Publisher	Oxford University Press
Item record/more information	http://hdl.handle.net/10197/7308
Publisher's statement	This article has been accepted for publication in Bioinformatics ©: 2013 the Author. Published by Oxford University Press. All rights reserved.
Publisher's version (DOI)	10.1093/bioinformatics/btt093

BIOINFORMATICS ORIGINAL PAPER

Vol. 29 no. 8 2013, pages 989–995
doi:10.1093/bioinformatics/btt093

Sequence analysis

Advance Access publication February 21, 2013

Making automated multiple alignments of very large numbers of protein sequences

Fabian Sievers[1,*], David Dineen[2], Andreas Wilm[3] and Desmond G. Higgins[1]

[1]School of Medicine and Medical Science, Conway Institute, University College Dublin, Dublin 4, Ireland, [2]Department of Bioengineering, University of California, Berkeley, CA 94729-1762, USA and [3]Computational and Systems Biology, Genome Institute of Singapore, Singapore 138672

Associate Editor: Inanc Birol

ABSTRACT

Motivation: Recent developments in sequence alignment software have made possible multiple sequence alignments (MSAs) of >100 000 sequences in reasonable times. At present, there are no systematic analyses concerning the scalability of the alignment quality as the number of aligned sequences is increased.

Results: We benchmarked a wide range of widely used MSA packages using a selection of protein families with some known structures and **found** that the accuracy of such alignments decreases markedly as the number of sequences grows. This is more or less true of all packages and protein families. The phenomenon is mostly due to the accumulation of alignment errors, rather than problems in guide-tree construction. This is partly alleviated by using iterative refinement or selectively adding sequences. The average accuracy of progressive methods by comparison with structure-based benchmarks can be improved by incorporating information derived from high-quality structural alignments of sequences with solved structures. This suggests that the availability of high quality curated alignments will have to complement algorithmic and/or software developments in the long-term.

Availability and implementation: Benchmark data used in this study are available at http://www.clustal.org/omega/homfam-20110613-25.tar.gz and http://www.clustal.org/omega/bali3fam-26.tar.gz.

Contact: fabian.sievers@ucd.ie

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on November 16, 2012; revised on January 31, 2013; accepted on February 18, 2013

1 INTRODUCTION

Multiple sequence alignments (MSAs) of many thousands of protein sequences are becoming commonplace. The biggest families in Pfam (Punta *et al.*, 2012) have >100 000 sequences or domains and will expand greatly as new genome sequences are being sequenced. MSAs are an integral part of how Pfam and other domain databases are used and maintained. Metagenomics research routinely involves the use of alignments of tens of thousands of sequences, and almost all phylogenetic analysis involves generating an MSA as a starting point. However, the generation of the MSA can be computationally too intensive for very large scale phylogenetic analysis. Recent work on predicting protein structure from sequence alignments is based on having very high quality alignments of many thousands of sequences (e.g. Marks *et al.*, 2011). Alignments of thousands of sequences have also been used in the area of virus classification (Shi *et al.*, 2010) and epistasis (Breen *et al.*, 2012).

Only a few of the standard MSA packages are capable of aligning tens of thousands of sequences. In a recent article, we described a new package called Clustal Omega (Sievers *et al.*, 2011), which makes it practical to align >100 000 protein sequences on a desktop computer and is as accurate as some of the most computationally demanding methods that can only align a few hundred sequences. The PartTree program (Katoh *et al.*, 2007) of the MAFFT package (Katoh *et al.*, 2002) and Kalign (Lassmann and Sonnhammer, 2005) can also make alignments of this size, although with a lower accuracy.

Some studies (Katoh *et al.*, 2005; Pei and Grishin, 2007; Simossis and Heringa, 2005) suggest that the quality of an alignment may increase as more sequences are added. This is only true if the newly added sequences are few and carefully chosen. As of yet, no systematic analyses have been conducted for large numbers of homologous sequences.

In this article, we look at some of the issues that occur when making alignments of 100–50 000 sequences using standard automatic MSA packages. For small alignments, we confirm a limited increase in accuracy, however, only if the added sequences are carefully chosen. We find a universal trend towards marked decrease in alignment accuracy as large numbers of sequences are added indiscriminately. We explore strategies that attenuate this deterioration, but they are useful only in certain cases. The strategy that best preserves alignment accuracy with very large datasets is to use a very high quality alignment of a small subset of the sequences to help guide the alignment. This suggests that very large alignments of high quality may be possible, but only if very high quality alignments such as those from structure superpositions or expert curated alignments are available.

2 METHODS

We investigate the general effect on the alignment quality of an MSA when adding new sequences to an existing set of un-aligned sequences. For this, we require (i) a broad range of alignment programs and (ii) suitable benchmark data.

*To whom correspondence should be addressed.

2.1 Alignment programs

The alignment programs that are used in this study are as follows:

(1) Clustal Omega, v1.0.3 (Sievers *et al.*, 2011)
(2) ClustalW2, v2.1 (Larkin *et al.*, 2007)
(3) DIALIGN 2.2.1 (Morgenstern *et al.*, 1998)
(4) FSA 1.15.5 (Bradley *et al.*, 2009)
(5) Kalign 2.04 (Lassmann and Sonnhammer, 2005)
(6) MAFFT 6.857 (Katoh *et al.*, 2002)
(7) MSAProbs 0.9.4 (Liu *et al.*, 2010)
(8) MUMMALS 1.01 (Pei and Grishin, 2006)
(9) MUSCLE version 3.8.31 posted May 1, 2010 (Edgar, 2004)
(10) Opal v2.0.0 (Wheeler and Kececioglu, 2007)
(11) Pagan v.0.38 posted March 6, 2012 (Löytynoja *et al.*, 2012)
(12) POA V2 v1.0.0 (Lee *et al.*, 2002)
(13) PRANK v.100802, August 2, 2010 (Löytynoja and Goldman, 2008)
(14) Probalign v1.4 (Roshan and Livesay, 2006)
(15) PROBCONS version 1.12 (Do *et al.*, 2005)
(16) PSAlign (using TCoffee 1.37) (Sze *et al.*, 2006)
(17) SATé v1.4.0 (using MAFFT v6.717b) (Liu *et al.*, 2009)
(18) T-Coffee Version 8.99 (Notredame *et al.*, 2000)

All programs were run with default command-line settings, apart from SATé, where –iter-without-imp-limit = 1 was set, to speed up the alignment. We used four different flavours of the MAFFT program: (i) L-INS-i, (ii) PartTree (Katoh *et al.*, 2007), (iii) default FFT-NS-2 mode and (iv) DP-PartTree. DP-PartTree uses a reduced distance matrix like PartTree but calculates full dynamic programming distances. Pagan (Löytynoja *et al.*, 2012) does not construct its own guide-tree but requires an external one. We re-used the Clustal Omega guide-trees, as they produced higher scores with Pagan than (default) MAFFT guide-trees (results not shown).

Of these programs, we will particularly focus on: (i) Clustal Omega, (ii) Kalign, (iii) MAFFT PartTree, (iv) DP-PartTree and (v) MAFFT L-INS-i. In Sievers *et al.* (2011), it was shown that on the BAliBASE3 (Thompson *et al.*, 2005) benchmark, Clustal Omega was more accurate than all progressive aligners and faster than all consistency aligners. Kalign was the fastest progressive aligner, while still giving very good accuracy. MAFFT L-INS-i was the fastest consistency aligner, and MAFFT-PartTree was the only program, apart from Clustal Omega, that could align ~100 000 sequences. Of these, only Clustal Omega, Kalign and MAFFT-(DP-)PartTree could align 50 000 sequences in a reasonable time. The other programs were run for up to 500 sequences.

2.2 Benchmark datasets

To assess the quality of an automatically generated MSA, one can use benchmark reference alignments. These are carefully constructed alignments that are assumed to be correct. Established benchmarks are made up of families with relatively few sequences, for example, at most 50 for Prefab (Edgar, 2004), at most 142 for BAliBASE3 and at most 807 for BAliBASE10 (Thompson *et al.*, 2011). Neither of these benchmarks qualify as extremely large alignments w.r.t. the number of sequences. We therefore created our own benchmark (Sievers *et al.*, 2011), where we blended Homstrad (Mizuguchi *et al.*, 2008) (as of June 13, 2011) reference sequences with Pfam (version 25) non-reference sequences, whenever there was a one-to-one match between Homstrad and Pfam families and when the Homstrad reference alignment had five or more sequences. The Homstrad reference alignments are assumed to be known with perfect accuracy. The Pfam sequences are available in large numbers, some exceeding 100 000. We compiled 94 families, with between 5 and 41 reference sequences, reference alignments between 39 and 938 in length and between 88 and 93 675 non-reference sequences. Only three families have >50 000 sequences.

The HomFam dataset is composed of single-domain Homstrad reference sequences with an admixture of Pfam sequences from the same single domain. As a second dataset, we created BaliFam, where we blended reference sequences from BAliBASE3 with ≥1000 Pfam sequences. We augmented 100 (out of 218) BAliBASE3 families with Pfam sequences from just one family. The remaining 118 families were augmented with Pfam sequences from up to 16 families. This was either because the BAliBASE3 family was multi-domain and/or because the corresponding Pfam family/families did not contain the desired 1000 sequences. This study does not consider fragments or sequencing errors, which pose difficult problems. HomFam and BaliFam are much easier to resolve and are therefore the 'best case scenario'.

The quality of the automatically generated alignment is then usually expressed by the Sum of correctly aligned Pairs (SP score) or by the number of correctly aligned Total Columns, divided by the length of the alignment (TC score). It could be argued that the TC score is too strict if non-core regions are to be aligned and scored. In this case, the SP score is more forgiving. However, we will show that for the HomFam benchmark set SP score and TC score give similar results. We also show that considering only core regions produces equivalent results by using BAliBASE10. For the rest of this study, we will use TC score over the entire range of the alignments.

To compile the input sequences, we randomly re-shuffle the order of the non-reference sequences with random seed r and then add the first i non-reference sequences to the (unaligned) reference sequences of family F, where $i = 0, 1, 2, 5, 10, \ldots$ When $i = 0$ only reference sequences are aligned. This is the base alignment; its TC score $\mathrm{TC}_{\mathrm{def}}(i = 0, \forall r, F)$ is the base score. Sampling one or more non-reference sequences is random and is therefore repeated R times with different random number seed r. If non-reference sequences are aligned together with reference sequences, then only the alignment of the embedded reference sequences can be scored. For scoring HomFam, we use qscore (Edgar, 2004), and for BaliFam, bali_score (Thompson *et al.*, 2005). For the computationally most demanding programs, we re-sample as often as feasible; for Clustal Omega, Kalign, MAFFT L-INS-i and MAFFT (DP-)PartTree, we re-sample $R = 100$ times. Results for different r are averaged.

2.3 Change in alignment score

The score $\mathrm{TC}_{\mathrm{def}}(i, r, F)$ for the alignment of the Homstrad reference sequences (which can be scored) and i non-reference Pfam sequences (which are part of the alignment but cannot be scored) of family F during re-sampling round r is shifted by the score of the corresponding base alignment $\mathrm{TC}_{\mathrm{def}}(0, 0, F)$ (containing Homstrad sequences only). This gives $\delta_{\mathrm{def}}(i, r, F) = \mathrm{TC}_{\mathrm{def}}(i, r, F) - \mathrm{TC}_{\mathrm{def}}(0, 0, F)$, the change in TC score w.r.t. the base alignment. Here, 'def' stands for default, that is, for the alignment that is produced using the programs' default command-line arguments. For every i, the δ are averaged over r and F to give the average change in TC score for each alignment program as i non-reference sequences are added to the Homstrad references. These steps are illustrated in Supplementary Figures S1 and S2.

To improve on these alignment results, it is important to understand what mechanisms affect the score as non-reference sequences are added. Since there are two distinct stages to the MSA process—(i) profile alignment and (ii) guide-tree construction—we try to isolate these two mechanisms by (i) keeping the guide-tree constant and by (ii) analysing the default guide-trees, by removing the effect of non-reference residues during the profile alignment stage. In both cases, we use Clustal Omega as the alignment program of choice.

2.4 Fixed guide-tree

It was shown previously that guide-tree topology strongly affects alignment accuracy (Blackshields *et al.*, 2010). This effect of variations in the guide-tree can be eliminated by fixing the tree. We construct, for each family, the biggest possible guide-tree, that is, for all reference and all available non-reference sequences. This will be called the fixed master guide-tree. We then populate the fixed master guide-tree with reference sequences only. In general, the topology of this fixed base tree will be different from the default base tree. The alignment is the fixed base alignment with score $\mathrm{TC}_{\mathrm{fix}}(0,0,F)$. Successively more non-reference sequences are then added at the appropriate positions in the master guide-tree. As the master guide-tree is fixed, the relative order in which sequences are aligned is the same for every alignment. We record $\delta_{\mathrm{fix}}(i,r,F) = \mathrm{TC}_{\mathrm{fix}}(i,r,F) - \mathrm{TC}_{\mathrm{fix}}(0,0,F)$, the change in TC score w.r.t. the fixed base score. The sequences that are used to obtain $\delta_{\mathrm{def}}(i,r,F)$ and $\delta_{\mathrm{fix}}(i,r,F)$ are the same; however, the alignments are arrived at using possibly different guide-trees, giving different alignments with different TC scores. For $\delta_{\mathrm{def}}(i,r,F)$, the guide-tree organizes itself from scratch, while the guide-tree for $\delta_{\mathrm{fix}}(i,r,F)$ is based on the fixed master tree. This procedure is illustrated in Supplementary Figures S3 and S4.

2.5 Pruned guide-tree

Conversely, to focus on the effect of the guide-tree topology, we take the default guide-trees and prune away all non-reference sequences. Although containing only reference sequences, these pruned trees will in general be topologically different from the default base trees. The reason for this is explained in Supplementary Figure S5. However, the pruned base tree is always identical with the default base tree and hence $\mathrm{TC}_{\mathrm{prune}}(0,0,F) \equiv \mathrm{TC}_{\mathrm{def}}(0,0,F)$. Next, the reference sequences are aligned using the pruned guide-trees and $\delta_{\mathrm{prune}}(i,r,F) = \mathrm{TC}_{\mathrm{prune}}(i,r,F) - \mathrm{TC}_{\mathrm{prune}}(0,0,F)$, the change in TC score of the pruned-tree alignment w.r.t. default base alignment, is obtained.

2.6 Correlation of default / fixed / pruned guide-tree alignment scores

To quantify the effect of (i) profile alignment and (ii) guide-tree construction on the default alignment scores we ask whether $\delta_{\mathrm{def}}(i,r,F)$, $\delta_{\mathrm{fix}}(i,r,F)$ and $\delta_{\mathrm{prune}}(i,r,F)$ are correlated. We chose Spearman's rank correlation coefficient ρ over Pearson's correlation coefficient, as it only assumes a monotonic function describing the correlation, rather than a linear one. We calculate $\rho(i)$ as a function of i, the number of added sequences, to see how contributions of the two mechanisms vary as more sequences are added. We will present $\rho(i)$ for $1 \leq i \leq 5000$ because there are at most 20 families with >5000 sequences.

In this and the previous sections, we suggested to measure the alignment quality as a function of the number of (added) sequences. Alternatively, we also study alignment quality as a function of the tree topology as measured by its entropy and tree diameter.

2.7 Iteration

Several alignment programs, for example, Clustal Omega, MAFFT and Muscle, can refine an alignment in a subsequent stage, which we will call 'iteration'. Iteration attempts to improve the objective score by repeatedly adjusting an initial MSA that is typically constructed by a progressive algorithm.

Clustal Omega has, corresponding to the two stages of MSA, two iteration modes, (i) guide-tree iteration and (ii) Hidden Markov Model (HMM) iteration. The logic behind guide-tree iteration is that distance information from a full multiple alignment should be more meaningful than distances between pairs of sequences; consequently, a tree constructed with this information should yield a better alignment. HMM iteration tries to remedy the fact that a progressive alignment algorithm has no 'foresight'. An alignment of two residues may seem advantageous at an early stage of the MSA. However, as more residues get aligned to this particular position, the initial alignment may in fact turn out to be sub-optimal. 'Mistakes' made at an early stage cannot be undone later-on in a progressive alignment scheme. HMM iteration helps to 'anticipate' the final distribution of residues and gaps at a certain position. The initial alignment is turned into an HMM. During the progressive alignment stage of the iteration, individual sequences and small profiles are aligned with the HMM and pseudo-count information is transferred. Both iteration modes can be invoked independently from each other; they can be repeated and combined. As we disentangled the effects of the guide-tree construction and the profile alignment stage, we will also invoke both iteration modes separately, as well as combined.

MAFFT and Muscle can also perform guide-tree iteration, as outlined above. Additionally, they can perform a refinement where a (preliminary) alignment is broken up into two groups, and the groups are then re-aligned (Barton and Sternberg, 1987).

2.8 Homology extension

So far, when adding homologues to the reference sequences, we have randomly re-shuffled the non-reference sequences and taken sequences from the top of this list. We would now like to reproduce the effect that has been reported, for example in (Katoh *et al.*, 2005; Kemena and Notredame, 2009; Pei and Grishin, 2007; Simossis and Heringa, 2005), that carefully selected homologous sequences can boost the alignment quality. We group the added sequences into bands, based on the minimum distance they have from any of the reference sequences. For the distance measure, we use both, full alignment distances and pair-wise distances. As scale, we use the minimum distance m between the reference sequences themselves, the average distance a between the references and their geometric mean $g = \sqrt{m \times a}$. This gives rise to four bands, where the minimum distance between a test sequence and any reference sequence are $d(\text{very similar}) \in [0,m)$, $d(\text{similar}) \in [m,g)$, $d(\text{medium}) \in [g,a)$ and $d(\text{dissimilar}) \in [a,\infty)$. Alternatively, methods like Cd-hit (Li and Godzik, 2006) or UCLUST (Edgar, 2010) can be used to group the sequences.

2.9 External profile alignment

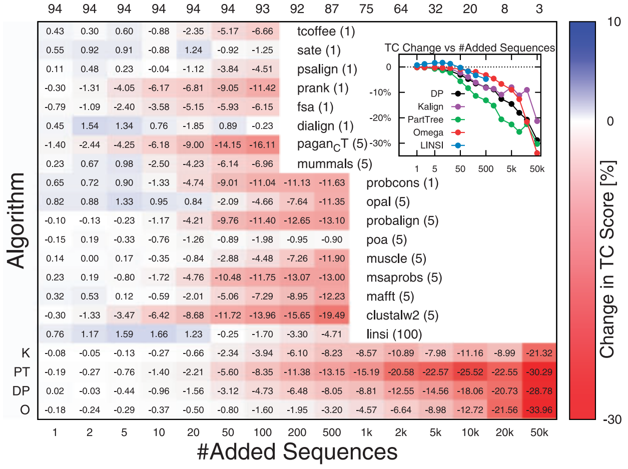
External Profile Alignment (EPA) is a combination of the iteration scheme from section 2.7 and the homologue scheme from section 2.8. During iteration, an HMM was produced from an *internally* created *alignment*; this HMM was used in a subsequent step to refine the alignment. The homologue scheme used *externally* stored *un-aligned* sequences to help with the alignment. EPA uses HMMs derived from *externally* produced *alignments*. Such alignments can be small and locally maintained ones or large generic alignments, as maintained, for example, by Pfam. Using HMMER (Finn *et al.*, 2011), we generated an HMM from the actual Homstrad reference alignment. This clearly is a blatant case of over-fitting, as it uses as input the alignment that is later used to score the alignment. These results therefore present an upper limit for the EPA scheme, using current aligners. However, it is a proof of principle that shows that EPA of carefully maintained alignments can significantly boost the quality of large alignments. The second method is more realistic, in that it uses HMMs that had been retrieved from Pfam. Pfam HMMs are produced from relatively small seed alignments, which in turn have been created using standard MSA programs, like MAFFT or MUSCLE. While the Pfam seed sequence selection may be representative, the actual seed alignment is presumably sub-optimal. The results for the Pfam–EPA scheme therefore present the lower limit for the potential of the EPA method.

3 RESULTS

3.1 Scalability of alignment quality

For a preliminary investigation, we took the three largest families in HomFam. HomFam families are composed of a small number of Homstrad sequences, for which the 'correct' structural alignment is known and can be scored, and a large number of Pfam sequences, for which no reliable alignment is known and therefore cannot be scored. The reference sequences were aligned together with non-reference sequences, using different aligners [Clustal Omega, MAFFT (DP-)PartTree and Kalign]. There is a clear trend: the alignment accuracy of the embedded Homstrad reference sequences falls as more non-reference Pfam sequences are aligned. This is shown in Supplementary Figure S6. To be sure that this phenomenon was not exclusive to just these test cases and these methods, we then tested all 94 HomFam test sets using 21 different alignment programs. In Figure 1, we show the change in alignment accuracy, as measured by the increase or decrease in TC score, as progressively more homologous Pfam non-reference sequences are added to Homstrad reference sequences.

The large scale tendency is the same for all alignment programs: the TC score goes down as large numbers of randomly selected homologous non-reference sequences are added. Most programs fall off in a more or less monotonic manner. A few programs enjoy a modest initial improvement. Most notably amongst these are MAFFT L-INS-i, Dialign, Opal, SATé and Probcons. However, even these programs inevitably end up below their respective base alignment TC score. Dialign and POA remain relatively constant over the sampled range. However, their absolute TC scores are comparatively low (Supplementary Figs S7–S9). The behaviour of the top five programs, for which the added sequences were re-sampled 100 times, is shown in the top right inset of Figure 1; the entire table is rendered in Supplementary Figure S10. Results for the multi-domain BaliFam test set are shown in Supplementary Figure S11. These results show the same tendency as for the single-domain HomFam. We also plotted the change in alignment scores against various tree measures like entropy and diameter. These correlated well for sequences of medium similarity (see section 2.8), but did not correlate well for sequences of high or low similarity (results not shown).



	1	2	5	10	20	50	100	200	500	1k	2k	5k	10k	20k	50k
	94	94	94	94	94	94	93	92	87	75	64	32	20	8	3
tcoffee (1)	0.43	0.30	0.60	-0.88	-2.35	-5.17	-6.66								
sate (1)	0.55	0.92	0.91	-0.88	1.24	-0.92	-1.25								
psalign (1)	0.11	0.48	0.23	-0.04	-1.12	-3.84	-4.51								
prank (1)	-0.30	-1.31	-4.05	-6.17	-6.81	-9.05	-11.42								
fsa (1)	-0.79	-1.09	-2.40	-3.58	-5.15	-5.93	-6.15								
dialign (1)	0.45	1.54	1.34	0.76	-1.85	0.89	-0.23								
pagan$_C$T (5)	-1.40	-2.44	-4.25	-6.18	-9.00	-14.15	-16.11								
mummals (5)	0.23	0.67	0.98	-2.50	-4.23	-6.14	-6.96								
probcons (1)	0.65	0.72	0.90	-1.33	-4.74	-9.01	-11.04	-11.13	-11.63						
opal (5)	0.82	0.88	1.33	0.95	0.84	-2.09	-4.66	-7.64	-11.35						
probalign (5)	-0.10	-0.13	-0.23	-1.17	-4.21	-9.76	-11.40	-12.65	-13.10						
poa (5)	-0.15	0.19	-0.33	-0.76	-1.26	-0.89	-1.98	-0.95	-0.90						
muscle (5)	0.14	0.00	0.17	-0.35	-0.84	-2.88	-4.48	-7.26	-11.90						
msaprobs (5)	0.23	0.19	-0.80	-1.72	-4.76	-10.48	-11.75	-13.07	-13.00						
mafft (5)	0.32	0.53	0.12	-0.59	-2.01	-5.06	-7.29	-8.95	-12.23						
clustalw2 (5)	-0.30	-1.33	-3.47	-6.42	-8.68	-11.72	-13.96	-15.65	-19.49						
linsi (100)	0.76	1.17	1.59	1.66	1.23	-0.25	-1.70	-3.30	-4.71						
K	-0.08	-0.05	-0.13	-0.27	-0.66	-2.34	-3.94	-6.10	-8.23	-8.57	-10.89	-7.98	-11.16	-8.99	-21.32
PT	-0.19	-0.27	-0.76	-1.40	-2.21	-5.60	-8.35	-11.38	-13.15	-15.19	-20.58	-22.57	-25.52	-22.55	-30.29
DP	0.02	-0.03	-0.44	-0.96	-1.56	-3.12	-4.73	-6.48	-8.05	-8.81	-12.55	-14.56	-18.06	-20.73	-28.78
O	-0.18	-0.24	-0.29	-0.37	-0.50	-0.80	-1.60	-1.95	-3.20	-4.57	-6.64	-8.98	-12.72	-21.56	-33.96

Fig. 1. Change in HomFam alignment score w.r.t. base alignment as non-reference sequences are added. Number of added sequences is along the bottom *x*-axis. Number of families that results are based on is along the top *x*-axis. Alignment algorithm along the *y*-axis—'K' = Kalign, 'PT' = MAFFT-PartTree, 'DP' = 'DP-PartTree' and 'O' = Clustal Omega. Number of re-samples *R* in parentheses, *R* = 100 for 'K', 'PT', 'DP' and 'O'. Improvement is highlighted blue, deterioration red. Top right hand inset shows graph of values for programs that were re-sampled 100 times: 'O' (red), 'K' (purple), 'PT' (green) 'DP' (black) and MAFFT L-INS-i (blue)

3.2 Contributions to the change in alignment quality

Next we wanted to establish if and how much different elements of the MSA scheme contribute to the change in alignment accuracy. In Figure 2a, we plot $\delta_{\text{fix}}(i, r, F)$, the change in the fixed tree score, against $\delta_{\text{def}}(i, r, F)$, the change in TC score if a default guide-tree is used and not the fixed master guide-tree, as *i* Pfam sequences of family *F* are added during re-sample *r* (Methods, section 2.4). The data points show two cases, that is, where only one sequence is added ($i = 1$, blue) and where five sequences are added ($i = 5$, red). There are 94 families with at least five non-reference sequences, which were re-sampled 50 times. So there are $(f = 94) \times (R = 50) = 4700$ blue and red dots. Visual inspection suggests that there is a positive correlation between the two changes in score, and that the correlation is stronger for one added sequence than for five. This is formalized in Figure 2d, where we show the Spearman coefficient $\rho(i)$, for the correlation of default and fixed tree scores within the range of 1–5000 added non-reference sequences (green). After a small drop (which attains a minimum at five) from a positive value, $\rho(i)$ is rising steadily. This means that errors outside the tree building phase,

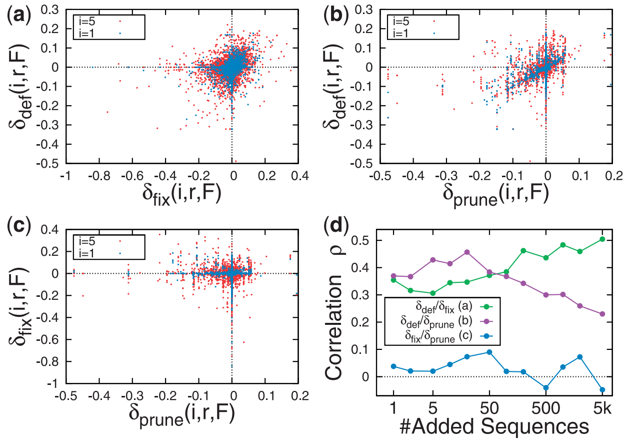


Fig. 2. Correlation of default score and contributions from tree building and profile alignment. (**a**) Correlation of change in default score (*y*-axis) and fixed tree score (*x*-axis), (**b**) default score (*y*-axis) and pruned tree score (*x*-axis), (**c**) fixed tree score (*y*-axis) and pruned tree score (*x*-axis); (a–c) as $i = 1$ (blue) and $i = 5$ (red) sequences are added. (**d**) Spearman's Rank Correlation coefficient for default/fixed (green), default/pruned (purple) and pruned/fixed (blue) changes in TC score as a function of the number of added sequences. x-Scale is logarithmic

that is, during the profile–profile alignment phase, contribute to the overall deterioration in TC score as more sequences are added; this effect is greater for larger numbers of sequences.

In Figure 2b, we plot the change in the pruned score $\delta_{\text{prune}}(i, r, F)$ (Methods, 2.5) against $\delta_{\text{def}}(i, r, F)$, for $i = 1, 5$. Again, visual inspection suggests that there is a positive correlation and that it is larger for five added sequence. This is quantified in Figure 2d, where the purple line shows the Spearman correlation coefficient of $\delta_{\text{prune}}(i, r, F)$ and $\delta_{\text{def}}(i, r, F)$. This curve starts at about the same value as the green curve but—after a brief rise, which attains its maximum at 20—falls off steadily. This means there is a contribution from errors during the tree building phase to the overall deterioration in TC score as non-reference sequences are added; this effect is greater for smaller numbers of sequences, and it is on average detrimental. Extensive reconstruction of the original guide-tree frequently leads to deterioration.

In Figure 2c, we plot $\delta_{\text{prune}}(i, r, F)$ against $\delta_{\text{fix}}(i, r, F)$ for $i = 1, 5$, and there is no apparent correlation. This is borne out in Figure 2d, where the blue line of Spearman's coefficient for $\delta_{\text{prune}}(i, r, F)$ and $\delta_{\text{fix}}(i, r, F)$ hovers around zero. This means that the contributions from the tree-building phase and the profile–profile alignments phase to the overall deterioration in TC score are decoupled.

3.3 Delay of alignment quality decay through iteration

Figure 3 shows the change in TC score for Clustal Omega, MAFFT PartTree and MUSCLE with increasing number of sequences and different iteration schemes.

The default result for Clustal Omega, with no iterations, is shown with bullets in Figure 3a. Results for various iteration schemes are overlaid. The main part of Figure 3a shows that iteration can indeed delay the onset of decay in alignment quality. Initially, single guide-tree iteration seems to be able to hold the TC score for up to 50 sequences. Double guide-tree iteration on its own appears to have no beneficial effect. After 50 added sequences, the guide-tree iteration results decline, and after 1000 sequences, they are worse than the default results. Guide-tree construction is based on distance matrix computation. Full distance matrices appear to give better results than mBed matrices (see Supplementary Fig. S12). Although for up to ten sequences HMM iteration has no appreciable effect, it then is able to stabilize the TC score until 200 sequences are added. After that it deteriorates, but it always remains above the default values. Initially, multiple HMM iteration has no advantage over single HMM iteration; however, as >1000 sequences are added, double and finally triple HMM iteration produce the best results. For very large numbers of added sequences, however, no iteration scheme can significantly reduce the decay in accuracy, as shown in the small inset. HMM and guide-tree iteration appear to be additive. When both HMM and guide-tree iteration outperform the default results, then combined iteration is better than either of the single schemes. When guide-tree iteration does worse than default then combined iteration fares worse than HMM iteration on its own. This is consistent with the correlation results from the last section 3.2.

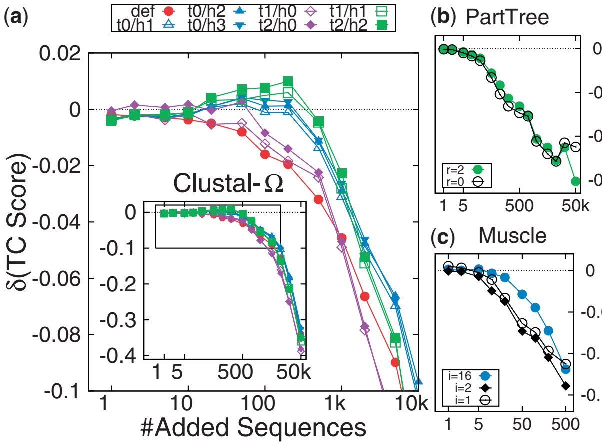


Fig. 3. Change in TC score for different aligners and iteration schemes. (**a**) Clustal Omega default result (not iterated) with bullets (same as red curve in Fig. 1). Guide-tree iteration with diamonds, HMM iteration with triangles, combined guide-tree/HMM iteration with squares. Single iteration with empty symbols, double iteration with filled-in symbols, triple iteration with upside–down symbols. Main part of Figure zooms in on results for small number of sequences; inset shows overview for large number of sequences. (**b**) MAFFT PartTree default results (retree = 2) with bullets, un-iterated results with circles. (**c**) MUSCLE default (maxiters = 16) results with bullets, lesser iterations with circles (1) and diamonds (2). Note the reduced x-range for MUSCLE

Figure 3b shows the change in the TC score for MAFFT PartTree. Here, the default setting is 'retree 2' and 'maxiterate 0' (bullets). It is not possible to increase the 'retree' value, only to reduce it. While there is a small absolute improvement for the higher retree value (1.5%, not shown), we also notice a small relative improvement for intermediate numbers of sequences. For very large numbers, the default values are worse than the un-iterated results; however, these results are based on very few families. The 'maxiterate' value cannot be changed in PartTree mode.

Figure 3c shows the change in TC score for MUSCLE. Here, the default setting is 'maxiters = 16' (bullets). MUSCLE uses just one flag to control both refinement modes. The default iterates the guide-tree twice and performs the alignment division 14 times. We increased 'maxiters' to 32 and 256, but there was no change in the absolute/relative scores. The results for four iterations (two guide-tree iterations, two alignment divisions, results not shown) were almost the same as for the default. The absolute accuracy drops by 1.2 and 3.4% if the number of iterations drops from 16 to 2 and 1, respectively (there are no alignment divisions and only one or two guide-tree iterations, respectively). The relative drop can be clearly seen in panel (c), as the default curve initially falls off less steeply. This is mainly due to the alignment divisions. The difference between the two black curves is due to the different number of guide-tree iterations only.

The overall result of Figure 3 is that iteration can, to a degree, delay the onset of decay in TC score for increased numbers of sequences but not indefinitely.

3.4 Effect of selectively sampling homologues

Figure 4 shows the TC scores if sequences are not added randomly, as in Figure 1, but selectively. Similarity was defined w.r.t. the minimum distance of an added sequence to any of

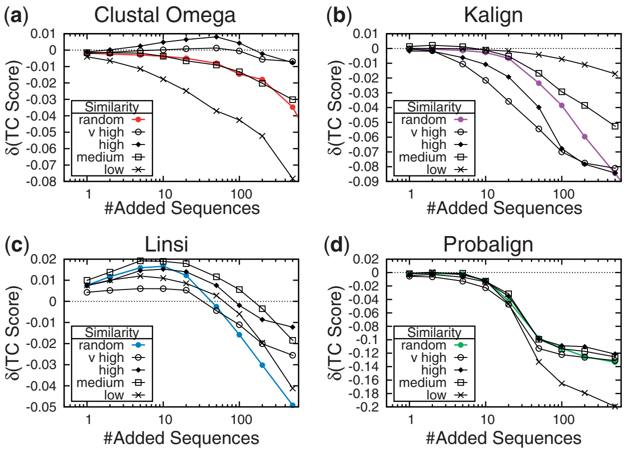


Fig. 4. Change in HomFam TC score for different algorithms and different sampling schemes. (**a**) Clustal Omega, (**b**) Kalign, (**c**) MAFFT L-INS-i, (**d**) Probalign. Random sampling with bullets and thicker lines, sampling of sequences of high similarity with circles, of low similarity with crosses and of in-between similarity with diamonds and boxes

the reference sequences. In Figure 4, we use full alignment distances; however, results for pair-wise distances are qualitatively similar (not shown). We present results for four example aligners (Clustal Omega, Kalign, MAFFT L-INS-i and Probalign). On average (not true for Kalign), adding sequences that are very similar (circles) or very dissimilar (crosses) does not improve on the alignment quality of the random scheme (bullets). Similar sequences simply 'cover' the reference sequences, not adding any new information, while very different sequences only get added once the reference sequences are already aligned and their alignment is fixed. These sequences will still affect the quality of the entire alignment. However, this quantity is unobservable in our scheme—as no reliable reference alignment exists for the Pfam sequences—but it is bounded from above by the observable TC score of the embedded Homstrad alignment. On the other hand, sequences of medium similarity (diamonds and boxes) appear to have a beneficial effect in small numbers. The minimum distance of these test sequences lies between the minimum distance of the reference sequences and the average distance of the reference sequences. This beneficial effect dissipates for >10–100 added sequences.

3.5 Effect of external profile alignment

Clustal Omega has a functionality called EPA where information from External Profiles can be added in the form of an HMM. Such HMMs are available from databases, such as Pfam, or can be built from locally maintained alignments using, for example, HMMER.

Figure 5 shows the effect of adding two different kinds of External Profiles: an HMM built from the actual reference alignment (stars) and an HMM retrieved from Pfam (circles). Panels (a–c) show the same three biggest HomFam test cases described in words in 3.1 and in Supplemental Figure S6. Panel (d) is the average of all 94 HomFam families. Both EPA-enhanced alignments show, for large numbers of added sequences, a significant improvement over the default results (bullets). Using the actual reference alignment particularly enhances the score for small numbers of sequences. Using Pfam, HMMs appears to be less beneficial for small numbers of sequences but seems to be more useful for larger numbers of sequences.

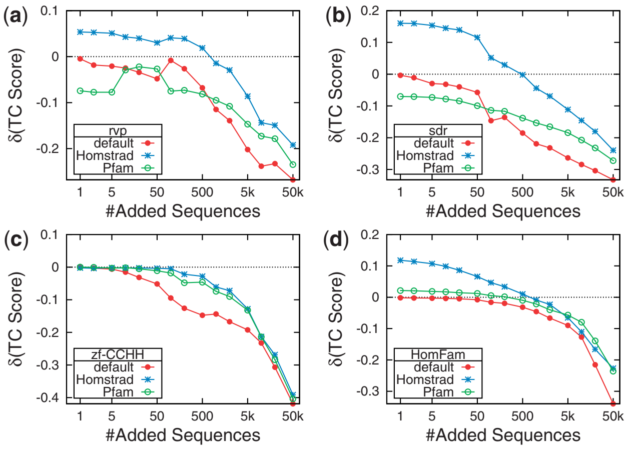


Fig. 5. Effect of EPA on alignment accuracies. (**a–c**) HomFam families with largest number of sequences; (**d**) average over all 94 HomFam families. Clustal Omega default results with bullets, using HMM built from reference alignment with stars, using off-the-shelf HMMs downloaded from Pfam with circles. TC scores measured against default Clustal Omega result (without EPA), when no extra sequences were added

Clearly, in terms of alignment benchmarking, this is circular. One cannot benchmark an alignment and alignment method, if one uses the benchmark itself. What it does show, however, is that if a user is faced with the problem of aligning large numbers of sequences, from one of these families, the use of a high-quality reference alignment helps enormously to maintain accuracy as one makes bigger alignments.

4 DISCUSSION

All of the standard automatic MSA packages behave very similarly, when the number of sequences to be aligned is increased into the thousands. Although few families exhibit a marked improvement in accuracy, the average accuracy—as measured on structure-based benchmarks—decreases steadily. This raises two obvious questions: what is the reason for the fall off and how can it be fixed?

The simplest explanation for the fall off in accuracy is attrition owing to the accumulation of noise and/or alignment errors as sequences are added. All of the widely used algorithms are based directly or indirectly on 'progressive alignment', which aligns the sequences according to the branching order in a 'guide-tree'. This requires a series of alignment steps, at any of which alignment errors can be made. These errors cannot be reversed, except by iteration of the alignment process. Such alignment errors occur less frequently with programs such as T-Coffee that use consistency (Notredame *et al.*, 2000), but such programs cannot easily cope with >1000 sequences. The presence of fragments, frameshifts, swapped domains and very large insertions or deletions will aggravate this situation.

With small numbers of sequences, the algorithms have proved to be very robust for general use. With very large numbers of sequences, however, the number of opportunities for irreversible alignment errors increases steadily. Even if during progressive alignment, only one sequence alignment in a thousand introduces a serious error, in a dataset of 100 000, this will occur 100 times. In large datasets, the scope for errors is simply very great. By fixing the guide-tree topology, we were able to separate out the effects of possible errors in guide-tree construction from alignment errors. Guide-tree construction certainly has an effect on alignment accuracy, but it is not the main source of error here. Iteration does help to delay the fall off in accuracy to an extent. We tested various combinations of iteration of guide-tree construction and alignment. For small-to-medium-sized datasets, the effects are noticeable, but the fall off in accuracy inevitably follows. Either the iteration strategy needs to be changed or it needs to be done more intensively. This would have the effect of greatly increasing alignment times. Carefully choosing the sequences to be aligned certainly has a beneficial effect, again, for modest increases in dataset size. We observe the best results when sequences of intermediate similarity are added. Figure 4 clearly demonstrates that sequences that are very similar did *not* improve accuracy. Perhaps, if huge alignments are desired, new sequences to be added to the dataset must be selected carefully.

Using progressive alignment packages is not the only way to make very large alignments, however. In the Pfam database, HMMER is used in a simple process, to add sequences one at a time to a smaller seed alignment. The accuracy of such alignment schemes has not been tested much, and it has probably been assumed that the accuracy is low. The full Pfam alignments are not intended as high-accuracy alignments. In Clustal Omega, there is a facility to use a pre-existing HMM to help the alignment of a new set of sequences, in a process called EPA. In the long-term, the most obvious solution to the issue of how to make very large alignments may be to use smaller high-quality alignments as seeds or ‘external profiles’ and algorithms for extending alignments such as Pagan (Löytynoja *et al.*, 2012), PaPaRa (Berger and Stamatakis, 2011) or as explained in (Katoh and Frith, 2012). Progressive alignment alone can make the alignments, but the accuracy will be a serious issue without new algorithms or strategies being developed.

Funding: Funding was provided by Science Foundation Ireland to D.H. through PI grants 11/PI/1034 and 07/IN.1/B1783.

Conflict of Interest: none declared.

REFERENCES

Barton,G.J. and Sternberg,M.J. (1987) Evaluation and improvements in the automatic alignment of protein sequences. *Protein Eng.*, **1**, 89–94.

Berger,S.A. and Stamatakis,A. (2011) Aligning short reads to reference alignments and trees. *Bioinformatics*, **27**, 2068–2075.

Blackshields,G. *et al.* (2010) Sequence emBedding for fast construction of guide trees for multiple sequence alignment. *Algorithms Mol. Biol.*, **5**, 21.

Bradley,R.K. *et al.* (2009) Fast statistical alignment. *PLoS Comput. Biol.*, **5**, e1000392.

Breen,M.S. *et al.* (2012) Epistasis as the primary factor in molecular evolution. *Nature*, **490**, 535–538.

Do,C.B. *et al.* (2005) ProbCons: probabilistic consistency-based multiple sequence alignment. *Genome Res.*, **15**, 330–340.

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

Edgar,R.C. (2010) Search and clustering orders of magnitude faster than BLAST. *Bioinformatics*, **26**, 2460–2461.

Finn,R.D. *et al.* (2011) HMMER web server: interactive sequence similarity searching. *Nucleic Acids Res.*, **39** (**Suppl. 2**), W29–W37.

Katoh,K. *et al.* (2002) MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic Acids Res.*, **30**, 3059–3066.

Katoh,K. *et al.* (2005) MAFFT version 5: improvement in accuracy of multiple sequence alignment. *Nucleic Acids Res.*, **33**, 511–518.

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

Katoh,K. and Frith,M.C. (2012) Adding unaligned sequences into an existing alignment using MAFFT and LAST. *Bioinformatics*, **28**, 3144–3146.

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

Larkin,M.A. *et al.* (2007) Clustal W and Clustal X version 2.0. *Bioinformatics*, **23**, 2947–2948.

Lassmann,T. and Sonnhammer,E.L (2005) Kalign–an accurate and fast multiple sequence alignment algorithm. *BMC Bioinformatics*, **6**, 298.

Lee,C. *et al.* (2002) Multiple sequence alignment using partial order graphs. *Bioinformatics*, **18**, 452–464.

Li,W. and Godzik,A. (2006) Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, **22**, 1658–1659.

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

Liu,Y. *et al.* (2010) MSAProbs: multiple sequence alignment based on pair hidden Markov models and partition function posterior probabilities. *Bioinformatics*, **26**, 1958–1964.

Löytynoja,A. and Goldman,N. (2008) Phylogeny-aware gap placement prevents errors in sequence alignment and evolutionary analysis. *Science*, **320**, 1632–1635.

Löytynoja,A. *et al.* (2012) Accurate extension of multiple sequence alignments using a phylogeny-aware graph algorithm. *Bioinformatics*, **28**, 1684–1691.

Marks,D.S. *et al.* (2011) Protein 3D structure computed from evolutionary sequence variation. *PLoS One*, **6**, e28766.

Mizuguchi,K. *et al.* (2008) HOMSTRAD: a database of protein structure alignments for homologous families. *Protein Sci.*, **7**, 2469–2471.

Morgenstern,B. *et al.* (1998) DIALIGN: finding local similarities by multiple sequence alignment. *Bioinformatics*, **14**, 290–294.

Notredame,C. *et al.* (2000) T-Coffee: a novel method for fast and accurate multiple sequence alignment. *J. Mol. Biol.*, **302**, 205–218.

Pei,J. and Grishin,N.V. (2006) MUMMALS: multiple sequence alignment improved by using hidden Markov models with local structural information. *Nucleic Acids Res.*, **34**, 4364–4374.

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

Punta,M. *et al.* (2012) The Pfam protein families database. *Nucleic Acids Res.*, **40**, D290–D301.

Roshan,U. and Livesay,D.R. (2006) Probalign: multiple sequence alignment using partition function posterior probabilities. *Bioinformatics*, **22**, 2715–2721.

Shi,W. *et al.* (2010) A complete analysis of HA and NA genes of influenza A viruses. *PLoS One*, **5**, e14454.

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

Sze,S.H. *et al.* (2006) A polynomial time solvable formulation of multiple sequence alignment. *J. Comput. Biol.*, **13**, 309–319.

Simossis,V.A. and Heringa,J. (2005) PRALINE: a multiple sequence alignment toolbox that integrates homology-extended and secondary structure information. *Nucleic Acids Res.*, **33** (**Suppl. 2**), W289–W294.

Thompson,J.D. *et al.* (2005) BAliBASE 3.0: latest developments of the multiple sequence alignment benchmark. *Proteins*, **61**, 127–136.

Thompson,J.D. *et al.* (2011) A comprehensive benchmark study of multiple sequence alignment methods: current challenges and future perspectives. *PloS One*, **6**, e18093.

Wheeler,T.J. and Kececioglu,J.D. (2007) Multiple alignment by aligning alignments. *Bioinformatics*, **23**, i559–i568.