Check for updates

Molecular Systems Biology 7; Article number 539; doi:10.1038/msb.2011.75
Citation: ***Molecular Systems Biology* 7: 539**
© 2011 EMBO and Macmillan Publishers Limited All rights reserved 1744-4292/11
www.molecularsystemsbiology.com

molecular systems biology

REPORT

Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega

Fabian Sievers[1,8], Andreas Wilm[2,8], David Dineen[1], Toby J Gibson[3], Kevin Karplus[4], Weizhong Li[5], Rodrigo Lopez[5], Hamish McWilliam[5], Michael Remmert[6], Johannes Söding[6], Julie D Thompson[7] and Desmond G Higgins[1,*]

[1] School of Medicine and Medical Science, UCD Conway Institute of Biomolecular and Biomedical Research, University College Dublin, Dublin, Ireland, [2] Computational and Systems Biology, Genome Institute of Singapore, Singapore, [3] Structural and Computational Biology Unit, European Molecular Biology Laboratory, Heidelberg, Germany, [4] Department of Biomolecular Engineering, University of California, Santa Cruz, CA, USA, [5] EMBL Outstation—European Bioinformatics Institute, Wellcome Trust Genome Campus, Hinxton, Cambridge, UK, [6] Gene Center Munich, University of Munich (LMU), Muenchen, Germany and [7] Département de Biologie Structurale et Génomique, IGBMC (Institut de Génétique et de Biologie Moléculaire et Cellulaire), CNRS/INSERM/Université de Strasbourg, Illkirch, France
[8] These authors contributed equally to this work
* Corresponding author. UCD Conway Institute of Biomolecular and Biomedical Research, University College Dublin, Belfield, Dublin 4, Ireland. Tel.: +353 1 716 6833; Fax: +353 1 716 6713; E-mail: des.higgins@ucd.ie

Received 23.7.11; accepted 6.9.11

Multiple sequence alignments are fundamental to many sequence analysis methods. Most alignments are computed using the progressive alignment heuristic. These methods are starting to become a bottleneck in some analysis pipelines when faced with data sets of the size of many thousands of sequences. Some methods allow computation of larger data sets while sacrificing quality, and others produce high-quality alignments, but scale badly with the number of sequences. In this paper, we describe a new program called Clustal Omega, which can align virtually any number of protein sequences quickly and that delivers accurate alignments. The accuracy of the package on smaller test cases is similar to that of the high-quality aligners. On larger data sets, Clustal Omega outperforms other packages in terms of execution time and quality. Clustal Omega also has powerful features for adding sequences to and exploiting information in existing alignments, making use of the vast amount of precomputed information in public databases like Pfam.

Molecular Systems Biology **7**: 539; published online 11 October 2011; doi:10.1038/msb.2011.75
Subject Categories: bioinformatics
Keywords: bioinformatics; hidden Markov models; multiple sequence alignment

Introduction

Multiple sequence alignments (MSAs) are essential in most bioinformatics analyses that involve comparing homologous sequences. The exact way of computing an optimal alignment between N sequences has a computational complexity of $O(L^N)$ for N sequences of length L making it prohibitive for even small numbers of sequences. Most automatic methods are based on the 'progressive alignment' heuristic (Hogeweg and Hesper, 1984), which aligns sequences in larger and larger subalignments, following the branching order in a 'guide tree.' With a complexity of roughly $O(N^2)$, this approach can routinely make alignments of a few thousand sequences of moderate length, but it is tough to make alignments much bigger than this. The progressive approach is a 'greedy algorithm' where mistakes made at the initial alignment stages cannot be corrected later. To counteract this effect, the consistency principle was developed (Notredame *et al*, 2000). This has allowed the production of a new generation of more accurate aligners (e.g. T-Coffee (Notredame *et al*, 2000)) but at the expense of ease of computation. These methods give 5–10% more accurate alignments, as measured on benchmarks, but are confined to a few hundred sequences.

In this report, we introduce a new program called Clustal Omega, which is accurate but also allows alignments of almost any size to be produced. We have used it to generate alignments of over 190 000 sequences on a single processor in a few hours. In benchmark tests, it is distinctly more accurate than most widely used, fast methods and comparable in accuracy to some of the intensive slow methods. It also has powerful features for allowing users to reuse their alignments so as to avoid recomputing an entire alignment, every time new sequences become available.

The key to making the progressive alignment approach scale is the method used to make the guide tree. Normally, this involves aligning all N sequences to each other giving time and memory requirements of $O(N^2)$. Protein families with $>50\,000$ sequences are appearing and will become common from various wide scale genome sequencing projects. Currently, the only method that can routinely make alignments of more than

© 2011 EMBO and Macmillan Publishers Limited

Downloaded from https://www.embopress.org on November 3, 2024 from IP 208.127.64.87.

Petitioner Merck, Ex. 1043, p. 1

about 10 000 sequences is MAFFT/PartTree (Katoh and Toh, 2007). It is very fast but leads to a loss in accuracy, which has to be compensated for by iteration and other heuristics. With Clustal Omega, we use a modified version of mBed (Blackshields *et al*, 2010), which has complexity of O(N log N), and which produces guide trees that are just as accurate as those from conventional methods. mBed works by 'emBedding' each sequence in a space of n dimensions where n is proportional to log N. Each sequence is then replaced by an n element vector, where each element is simply the distance to one of n 'reference sequences.' These vectors can then be clustered extremely quickly by standard methods such as K-means or UPGMA. In Clustal Omega, the alignments are then computed using the very accurate HHalign package (Söding, 2005), which aligns two profile hidden Markov models (Eddy, 1998).

Clustal Omega has a number of features for adding sequences to existing alignments or for using existing alignments to help align new sequences. One innovation is to allow users to specify a profile HMM that is derived from an alignment of sequences that are homologous to the input set. The sequences are then aligned to these 'external profiles' to help align them to the rest of the input set. There are already widely available collections of HMMs from many sources such as Pfam (Finn *et al*, 2009) and these can now be used to help users to align their sequences.

Results

Alignment accuracy

The standard method for measuring the accuracy of multiple alignment algorithms is to use benchmark test sets of reference alignments, generated with reference to three-dimensional structures. Here, we present results from a range of packages tested on three benchmarks: BAliBASE (Thompson *et al*, 2005), Prefab (Edgar, 2004) and an extended version of HomFam (Blackshields *et al*, 2010). For these tests, we just report results using the default settings for all programs but with two exceptions, which were needed to allow MUSCLE (Edgar, 2004) and MAFFT to align the biggest test cases in HomFam. For test cases with >3000 sequences, we run MUSCLE with the –maxiter parameter set to 2, in order to finish the alignments in reasonable times. Second, we have run several different programs from the MAFFT package. MAFFT (Katoh *et al*, 2002) consists of a series of programs that can be run separately or called automatically from a script with the *–auto* flag set. This flag chooses to run a slow, consistency-based program (L-INS-i) when the number and lengths of sequences is small. When the numbers exceed inbuilt thresholds, a conventional progressive aligner is used (FFT-NS-2). The latter is also the program that is run by default if MAFFT is called with no flags set. For very large data sets, the *–parttree* flag must be set on the command line and a very fast guide tree calculation is then used.

The results for the BAliBASE benchmark tests are shown in Table I. BAliBASE is divided into six 'references.' Average scores are given for each reference, along with total run times and average total column (TC) scores, which give the proportion of the total alignment columns that is recovered. A score of 1.0 indicates perfect agreement with the benchmark. There are two rows for the MAFFT package: MAFFT (auto) and MAFFT default. In most (203 out of 218) BAliBASE test cases, the number of sequences is small and the script runs L-INS-i, which is the slow accurate program that uses the consistency heuristic (Notredame *et al*, 2000) that is also used by MSAprobs (Liu *et al*, 2010), Probalign, Probcons (Do *et al*, 2005) and T-Coffee. These programs are all restricted to small numbers of sequences but tend to give accurate alignments. This is clearly reflected in the times and average scores in Table I. The times range from 25 min up to 22 h for these packages and the accuracies range from 55 to 61% of columns correct. Clustal Omega only takes 9 min for the same runs but has an accuracy level that is similar to that of Probcons and T-Coffee.

The rest of the table is mainly taken by the programs that use progressive alignment. Some of these are very fast but this speed is matched by a considerable drop in accuracy compared with the consistency-based programs and Clustal Omega. The weakest program here, is Clustal W (Larkin *et al*, 2007) followed by PRANK (Löytynoja and Goldman, 2008). PRANK is not designed for aligning distantly related sequences but at giving good alignments for phylogenetic work with special

Table I BAliBASE results

Aligner	Av score (218 families)	BB11 (38 families)	BB12 (44 families)	BB2 (41 families)	BB3 (30 families)	BB4 (49 families)	BB5 (16 families)	Tot time (s)	Consistency
MSAprobs	0.607	0.441	0.865	0.464	0.607	0.622	0.608	12 382.00	Yes
Probalign	0.589	0.453	0.862	0.439	0.566	0.603	0.549	10 095.20	Yes
MAFFT (auto)	0.588	0.439	0.831	0.450	0.581	0.605	0.591	1475.40	Mostly (203/218)
Probcons	0.558	0.417	0.855	0.406	0.544	0.532	0.573	13 086.30	Yes
Clustal Ω	0.554	0.358	0.789	0.450	0.575	0.579	0.533	539.91	No
T-Coffee	0.551	0.410	0.848	0.402	0.491	0.545	0.587	81 041.50	Yes
Kalign	0.501	0.365	0.790	0.360	0.476	0.504	0.435	21.88	No
MUSCLE	0.475	0.318	0.804	0.350	0.409	0.450	0.460	789.57	No
MAFFT (default)	0.458	0.258	0.749	0.316	0.425	0.480	0.496	68.24	No
FSA	0.419	0.270	0.818	0.187	0.259	0.474	0.398	53 648.10	No
Dialign	0.415	0.265	0.696	0.292	0.312	0.441	0.425	3977.44	No
PRANK	0.376	0.223	0.680	0.257	0.321	0.360	0.356	128 355.00	No
ClustalW	0.374	0.227	0.712	0.220	0.272	0.396	0.308	766.47	No

The figures are total column scores produced using bali score on core columns only. The average score over all families is given in the second column. The results for BAliBASE subgroupings are in columns 3–8. The total run time for all 218 families is given in the second last column. The last column indicates whether the method is consistency based.

© 2011 EMBO and Macmillan Publishers Limited

Downloaded from https://www.embopress.org on November 3, 2024 from IP 208.127.64.87.

Petitioner Merck, Ex. 1043, p. 2

attention to gaps. These gap positions are not included in these tests as they tend not to be structurally conserved. Dialign (Morgenstern *et al*, 1998) does not use consistency or progressive alignment but is based on finding best local multiple alignments. FSA (Bradley *et al*, 2009) uses sampling of pairwise alignments and 'sequence annealing' and has been shown to deliver good nucleotide sequence alignments in the past.

The Prefab benchmark test results are shown in Table II. Here, the results are divided into five groups according to the percent identity of the sequences. The overall scores range from 53 to 73% of columns correct. The consistency-based programs MSAprobs, MAFFT L-INS-i, Probalign, Probcons and T-Coffee, are again the most accurate but with long run times. Clustal Omega is close to the consistency programs in accuracy but is much faster. There is then a gap to the faster progressive based programs of MUSCLE, MAFFT, Kalign (Lassmann and Sonnhammer, 2005) and Clustal W.

Results from testing large alignments with up to 50 000 sequences are given in Table III using HomFam. Here, each alignment is made up of a core of a Homstrad (Mizuguchi *et al*, 1998) structure-based alignment of at least five sequences. These sequences are then inserted into a test set of sequences from the corresponding, homologous, Pfam domain. This gives very large sets of sequences to be aligned but the testing is only carried out on the sequences with known structures. Only some programs are able to deliver alignments at all, with data sets of this size. We restricted the comparisons to Clustal Omega, MAFFT, MUSCLE and Kalign. MAFFT with default settings, has a limit of 20 000 sequences and we only use MAFFT with –parttree for the last section of Table III. MUSCLE becomes increasingly slow when you get over 3000 sequences. Therefore, for >3000 sequences we used MUSCLE with the faster but less accurate setting of –maxiters 2, which restricts the number of iterations to two.

Overall, Clustal Omega is easily the most accurate program in Table III. The run times show MAFFT default and Kalign to be exceptionally fast on the smaller test cases and MAFFT –parttree to be very fast on the biggest families. Clustal Omega does scale well, however, with increasing numbers of sequences. This scaling is described in more detail in the Supplementary Information. We do have two further test cases with >50 000 sequences, but it was not possible to get results for these from MUSCLE or Kalign. These are described in the Supplementary Information as well.

Table III gives overall run times for the four programs evaluated with HomFam. Figure 1 resolves these run times case by case. Kalign is very fast for small families but does not scale as well. Overall, MAFFT is faster than the other programs over all test case sizes but Clustal Omega scales similarly. Points in Figure 1 represent different families with different average sequence lengths and pairwise identities. Therefore, the scalability trend is fuzzy, with larger dots occurring generally above smaller dots. Supplementary Figure S3 shows scalability data, where subsets of increasing size are sampled from one large family only. This reduces variability in pairwise identity and sequence length.

External profile alignment

Clustal Omega can read extra information from a profile HMM derived from preexisting alignments. For example, if a user

Table II Prefab results

Aligner	0 < %ID⩽100 (1682 families)	0⩽%ID⩽20 (912 families)	20⩽%ID⩽40 (563 families)	40⩽%ID⩽70 (117 families)	70⩽%ID⩽100 (90 families)	Total time (s) (1682 families)	Consistency
MSAprobs	0.737	0.591	0.889	0.965	0.971	51 286.00	Yes
MAFFT (auto)	0.721	0.569	0.876	0.961	0.979	4544.45	Yes
Probalign	0.719	0.563	0.881	0.961	0.977	35117.30	Yes
Probcons	0.717	0.562	0.876	0.955	0.972	46 908.30	Yes
T-Coffee	0.710	0.558	0.865	0.950	0.972	175 789.00	Yes
Clustal Ω	0.700	0.535	0.866	0.967	0.980	1698.06	No
MUSCLE	0.677	0.507	0.850	0.946	0.976	2068.56	No
MAFFT	0.677	0.513	0.836	0.961	0.979	225.56	No
Kalign	0.649	0.474	0.817	0.957	0.979	80.81	No
ClustalW2	0.617	0.430	0.797	0.933	0.975	3433.53	No
Dialign	0.595	0.398	0.783	0.940	0.974	18 909.70	No
PRANK	0.586	0.390	0.767	0.951	0.978	351 498.00	No
FSA	0.534	0.277	0.791	0.965	0.976	229 391.00	No

Total column scores (TC) are shown for different percent identity ranges; the second column is the average score over all test cases. The total run time in seconds is shown in the second last column. The last column indicates if the method is consistency based.

Downloaded from https://www.embopress.org on November 3, 2024 from IP 208.127.64.87.

© 2011 EMBO and Macmillan Publishers Limited

Petitioner Merck, Ex. 1043, p. 3

wishes to align a set of globin sequences and has an existing globin alignment, this alignment can be converted to a profile HMM and used as well as the sequence input file. This HMM is here referred to as an 'external profile' and its use in this way as 'external profile alignment' (EPA). During EPA, each sequence in the input set is aligned to the external profile. Pseudocount information from the external profile is then transferred, position by position, to the input sequence. Ideally, this would be used with large curated alignments of particular proteins or domains of interest such as are used in metagenomics projects. Rather than taking the input sequences and aligning them from scratch, every time new sequences are found, the alignment should be carefully maintained and used as an external profile for EPA. Clustal Omega also can align sequences to existing alignments using conventional alignment methods. Users can add sequences to an alignment, one by one or align a set of aligned sequences to the alignment.

In this paper, we demonstrate the EPA approach with two examples. First, we take the 94 HomFam test cases from the previous section and use the corresponding Pfam HMM for EPA. Before EPA, the average accuracy for the test cases was 0.627 of correctly aligned Homstrad positions but after EPA it rises to 0.653. This is plotted, test case for test case in Figure 2A. Each dot is one test case with the TC score for Clustal Omega plotted against the score using EPA. The second example is illustrated in Figure 2B. Here, we take all the BAliBASE reference sets and align them as normal using Clustal Omega and obtain the benchmark result of 0.554 of columns correctly aligned, as already reported in Table I. For EPA, we use the benchmark reference alignments themselves as external profiles. The results now jump to 0.857 of columns correct. This is a jump of over 30% and while it is not a valid measure of Clustal Omega accuracy for comparison with other programs, it does illustrate the potential power of EPA to use information in external alignments.

Iteration

EPA can also be used in a simple iteration scheme. Once a MSA has been made from a set of input sequences, it can be converted into a HMM and used for EPA to help realign the input sequences. This can also be combined with a full recalculation of the guide tree. In Figure 3, we show the results of one and two iterations on every test case from HomFam. The graph is plotted as a running average TC score for all test cases with N or fewer test cases where N is plotted on the horizontal axis using a log scale. With some smaller test cases, iteration actually has a detrimental effect. Once you get near 1000 or more sequences, however, a clear trend emerges. The more sequences you have, the more beneficial the effect of iteration is. With bigger test cases, it becomes more and more beneficial to apply two iterations. This result confirms the usefulness of EPA as a general strategy. It also confirms the difficulty in aligning extremely large numbers of sequences but gives one partial solution. It also gives a very simple but effective iteration scheme, not just for guide tree iteration, as used in many packages, but for iteration of the alignment itself.

Discussion

The main breakthroughs since the mid 1980s in MSA methods have been progressive alignment and the use of consistency. Otherwise, most recent work has concerned refinements for speed or accuracy on benchmark test sets. The speed increases have been dramatic but, with just two major exceptions, the methods are still basically $O(N^2)$ and incapable of being extended to data sets of $>10\,000$ sequences. The two exceptions are mBed, used here, and MAFFT PartTree. PartTree is faster but at the expense of accuracy, at least as judged by the benchmarking here. The second group of recent developments

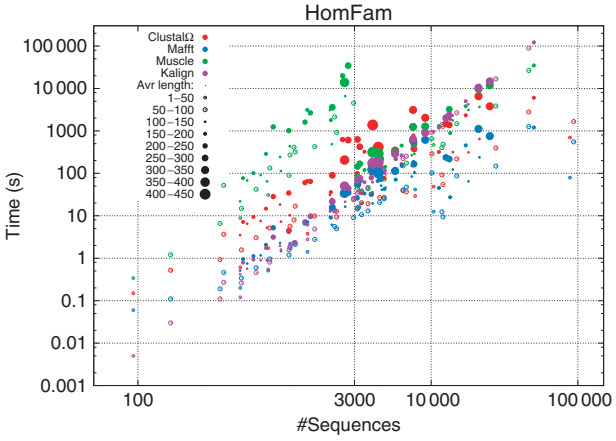


Figure 1 Alignment time for Clustal Omega (red), MAFFT (blue), MUSCLE (green) and Kalign (purple) against the number of sequences of HomFam test sets. Average sequence length is rendered by point size. Both axes have logarithmic scales. Clustal Omega and Kalign were run with default flags over the entire range. MUSCLE was run with –maxiters 2 for $N>3000$ sequences. MAFFT was run with –parttree for $N>10\,000$ sequences.

Table III HomFam benchmarking results

	$93 \leqslant N \leqslant 2957$ (41 families)	$3127 \leqslant N \leqslant 9105$ (33 families)	$10\,099 \leqslant N \leqslant 50\,157$ (18 families)
Aligner	TC/t (s)	TC/t (s)	TC/t (s)
Clustal Ω	0.708/2114.0	0.639/11 719.5	0.464/27 328.9
Kalign	0.569/324.9	0.563/6752.0	0.420/286 711.0
MAFFT default	0.550/238.9	0.462/3115.4	–/–
MAFFT –parttree	–/–	–/–	0.253/6119.4
MUSCLE default	0.533/104 587.0	–/–	–/–
MUSCLE –maxiters 2	–/–	0.416/8239.2	0.216/110 292.0

The columns show total column score (TC) and total run time in seconds for groupings of small (<3000 sequences), medium (3000–10 000 sequences) and large ($>10\,000$ sequences) HomFam test cases.

© 2011 EMBO and Macmillan Publishers Limited

Downloaded from https://www.embopress.org on November 3, 2024 from IP 208.127.64.87.

Petitioner Merck, Ex. 1043, p. 4

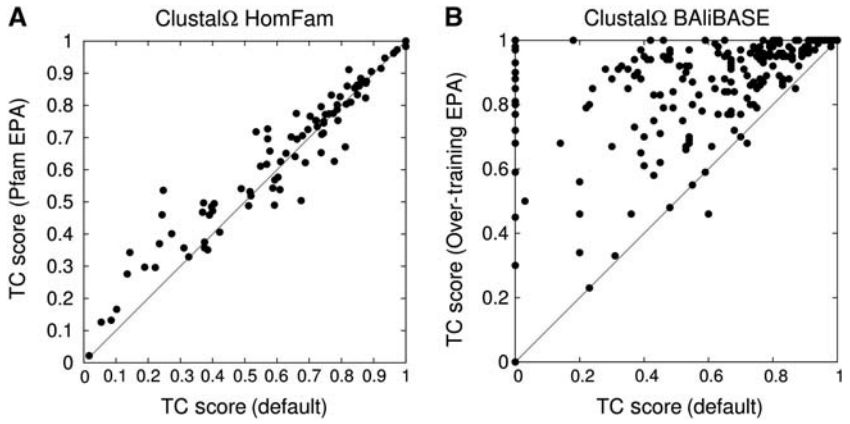


Figure 2 EPA for HomFam and BAliBASE. Points represent TC scores of Clustal Omega alignment with EPA versus TC scores of default Clustal Omega alignment (without EPA). Points above bisectrix represent beneficial effect of EPA, points below deleterious effect. Average improvement in (**A**) 2.5%. HMMs taken from Pfam, benchmarking carried out using corresponding structure-based alignment in Homstrad. Average improvement in (**B**) over 30%. Here, test sets and EPA-HMMs were both derived from BAliBASE reference alignments.

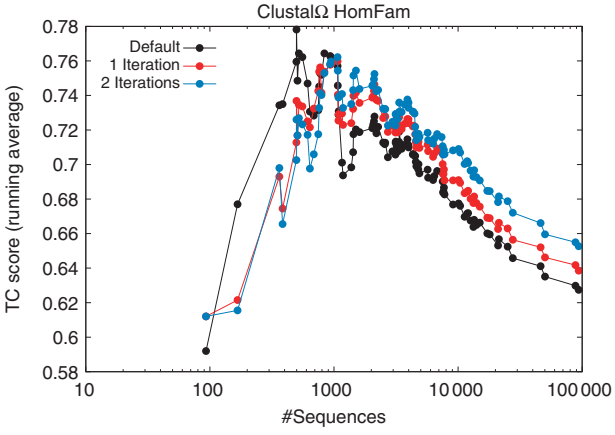


Figure 3 Iteration of HomFam alignments. Points represent cumulative running averages of TC scores. Clustal Omega default results in black, results after 1 iteration in red, after 2 iterations in blue. Iterations are combined HMM/guide tree iterations; *x* axis, logarithmic and *y* axis, linear scale.

have concerned accuracy. This has tended to focus on results from benchmarking, a potentially contentious issue (Aniba *et al*, 2010; Edgar, 2010). The benchmark test sets that we have are limited in scope and heavily biased toward single domain globular proteins. This has the potential to lead to methods that behave well on benchmarks but which are not so flexible or useful in real-world situations. One development to improve accuracy has been the recruitment of extra homologs to bulk up input data sets. This seems to work well with the consistency-based methods and for small data sets. It appears, however, that there is a limit to the extra accuracy that can be obtained this way, without further development. The extra sequences may also bring in noise and dramatically increase the complexity of the computational problem. This can be partly fixed by iteration but, EPA to a high-quality reference alignment might be a better solution. This also raises the need for methods to visualize such large alignments, in order to detect problems. A second major focus for development has been the use of external information such as RNA structure (Wilm *et al*, 2008) or protein structure predictions (Pirovano *et al*, 2008).

EPA is a new approach that allows users to exploit information in their own or in publicly available alignments. It does not force new sequences to follow the older alignment exactly. The new sequences get aligned to each other using progressive alignment but the information in the external profile can help provide information as to which amino acids are most likely to occur at each position in a sequence. Most methods attempt to predict this from general models of protein evolution with secondary structure prediction as a refinement. In this paper, we have shown that even using the mass produced alignments from Pfam as external profiles provides a small increase in accuracy for a large general set of test cases. This opens up a new set of possibilities for users to make use of the information contained in large, publicly available alignments and creates an incentive for database providers to make very high-quality alignments available.

One of the reasons for the great success of Clustal X was the very user-friendly graphical user interface (GUI). This, however, is not as critical as in the past due to the widespread availability of web-based services where the GUI is provided by the web-based front-end server. Further, there are several very high-quality alignment viewers and editors such as Jalview (Clamp *et al*, 2004) and Seaview (Gouy *et al*, 2010) that read Clustal Omega output or which can call Clustal Omega directly.

Materials and methods

Clustal Omega is licensed under the GNU Lesser General Public License. Source code as well as precompiled binaries for Linux, FreeBSD, Windows and Mac (Intel and PowerPC) are available at http://www.clustal.org. Clustal Omega is available as a command line program only, which uses GNU-style command line options, and also accepts ClustalW-style command options for backwards compatibility and easy integration into existing pipelines.

Clustal Omega is written in C and C++ and makes use of a number of excellent free software packages. We used a modified version of

Downloaded from https://www.embopress.org on November 3, 2024 from IP 208.127.64.87.

© 2011 EMBO and Macmillan Publishers Limited

Petitioner Merck, Ex. 1043, p. 5

Sean Eddy's Squid library (http://selab.janelia.org/software.html) for sequence I/O, allowing the use of a wide variety of file formats. We use David Arthur's k-means + + code (Arthur and Vassilvitskii, 2007) for fast clustering of sequence vectors. Code for fast UPGMA and guide tree handling routines was adopted from MUSCLE (Edgar, 2004). We use the OpenMP library to enable multithreaded computation of pairwise distances and alignment match states. The documentation for Clustal Omega's API is part of the source code, and in addition is available from http://www.clustal.org/omega/clustalo-api/. Full details of all algorithms are given in the accompanying Supplementary Information.

The benchmarks that were used were BAliBASE 3 (Thompson *et al*, 2005), PREFAB 4.0 (posted March 2005) (Edgar, 2010) and a newly constructed data set (HomFam) using sequences from Pfam (version 25) and Homstrad (as of 2011-06-13) (Mizuguchi *et al*, 1998). The programs that were compared can be obtained from:

ClustalW2, v2.1 (http://www.clustal.org)
DIALIGN 2.2.1 (http://dialign.gobics.de/)
FSA 1.15.5 (http://sourceforge.net/projects/fsa/)
Kalign 2.04 (http://msa.sbc.su.se/cgi-bin/msa.cgi)
MAFFT 6.857 (http://mafft.cbrc.jp/alignment/software/source.html)
MSAProbs 0.9.4 (http://sourceforge.net/projects/msaprobs/files/)
MUSCLE version 3.8.31 posted 1 May 2010 (http://www.drive5.com/muscle/downloads.htm)
PRANK v.100802, 2 August 2010 (http://www.ebi.ac.uk/goldman-srv/prank/src/prank/)
Probalign v1.4 (http://cs.njit.edu/usman/probalign/)
PROBCONS version 1.12 (http://probcons.stanford.edu/download.html)
T-Coffee Version 8.99 (http://www.tcoffee.org/Projects_home_page/t_coffee_home_page.html#DOWNLOAD).

Supplementary information

Supplementary information is available at the *Molecular Systems Biology* website (www.nature.com/msb).

Acknowledgements

This work was supported by Science Foundation Ireland (Grant number: 07/IN.1/B1783).

Author contributions: DGH initiated the project; the original ClustalW software was produced by JDT, DGH and TJG. The HHalign code was written by JS and is maintained by MR. The idea of EPA was suggested by KK. FS adapted HHalign to Clustal Omega; AW devised and adapted guide tree construction routines. DD parallelised the code. AW, DD and FS carried out all software development. The EBI server was set up by RL, HMcW and WL. Benchmarking was carried out by JDT, AW, DD and FS. All authors contributed to the manuscript.

Conflict of interest

The authors declare that they have no conflict of interest.

References

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

Arthur D, Vassilvitskii S (2007) *k*-means + + : the advantages of careful seeding. Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms. pp 1027–1035

Blackshields G, Sievers F, Shi W, Wilm A, Higgins DG (2010) Sequence emBedding for fast construction of guide trees for multiple sequence alignment. *Algorithms Mol Biol* **5:** 21

Bradley RK, Roberts A, Smoot M, Juvekar S, Do J, Dewey C, Holmes I, Pachter L (2009) Fast statistical alignment. *PLoS Comput Biol* **5:** e1000392

Clamp M, Cuff J, Searle SM, Barton GJ (2004) The Jalview Java alignment editor. *Bioinformatics* **20:** 426–427

Do CB, Mahabhashyam MSP, Brudno M, Batzoglou S (2005) ProbCons: probabilistic consistency-based multiple sequence alignment. *Genome Res* **15:** 330–340

Eddy SR (1998) Profile hidden Markov models. *Bioinformatics* **14:** 755–763

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

Edgar RC (2010) Quality measures for protein alignment benchmarks. *Nucleic Acids Res* **38:** 2145–2153

Finn RD, Mistry J, Tate J, Coggill P, Heger A, Pollington JE, Gavin OL, Gunasekaran P, Ceric G, Forslund K, Holm L, Sonnhammer EL, Eddy SR, Bateman A (2009) The Pfam protein families database. *Nucleic Acids Res* **38:** D211–D222

Gouy M, Guindon S, Gascuel O (2010) SeaView version 4: a multiplatform graphical user interface for sequence alignment and phylogenetic tree building. *Mol Biol Evol* **27:** 221–224

Hogeweg P, Hesper B (1984) The alignment of sets of sequences and the construction of phyletic trees: an integrated method. *J Mol Evol* **20:** 175–186

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

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

Larkin MA, Blackshields G, Brown NP, Chenna R, McGettigan PA, McWilliam H, Valentin F, Wallace IM, Wilm A, Lopez R, Thompson JD, Gibson TJ, Higgins DG (2007) Clustal W and Clustal X version 2.0. *Bioinformatics* **23:** 2947–2948

Lassmann T, Sonnhammer ELL (2005) Kalign—an accurate and fast multiple sequence alignment algorithm. *BMC Bioinformatics* **6:** 298

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

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

Mizuguchi K, Deane CM, Blundell TL, Overington JP (1998) HOMSTRAD: a database of protein structure alignments for homologous families. *Protein Sci* **7:** 2469–2471

Morgenstern B, Frech K, Dress A, Werner T (1998) DIALIGN: finding local similarities by multiple sequence alignment. *Bioinformatics* **14:** 290–294

Notredame C, Higgins DG, Heringa J (2000) T-coffee: a novel method for fast and accurate multiple sequence alignment. *J Mol Biol* **302:** 205–217

Pirovano W, Feenstra KA, Heringa J (2008) PRALINETM: a strategy for improved multiple alignment of transmembrane proteins. *Bioinformatics* **24:** 492–497

Söding J (2005) Protein homology detection by HMM–HMM comparison. *Bioinformatics* **21:** 951–960

Thompson JD, Koehl P, Ripp R, Poch O (2005) BAliBASE 3.0: latest developments of the multiple sequence alignment benchmark. *Proteins* **61:** 127–136

Wilm A, Higgins DG, Notredame C (2008) R-Coffee: a method for multiple alignment of non-coding RNA. *Nucleic Acids Res* **36:** e52

SOME RIGHTS RESERVED *Molecular Systems Biology* is an open-access journal published by *European Molecular Biology Organization* and *Nature Publishing Group*. This work is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 Unported License.

Downloaded from https://www.embopress.org on November 3, 2024 from IP 208.127.64.87.

© 2011 EMBO and Macmillan Publishers Limited

Petitioner Merck, Ex. 1043, p. 6

Supplementary Information

Guide Tree Construction

Standard progressive alignment schemes build up the final alignment by aligning individual sequences and intermediate profiles in a particular order. This order is decided by the guide-tree which is based on pairwise dis/similarities of the individual sequences. Calculation time and memory requirements to construct a full pairwise distance matrix grow quadratically with the number of sequences which is prohibitive for large numbers of sequences.

It was shown in Blackshields et al. (2010) that this quadratic scalability can be improved on, using a scheme called mBed. In the mBed scheme a small number of sequences are selected as seeds. Following Linial et al. (1995) this number is taken to be $\log_2(N)$. The seed selection can be random or follow a systematic strategy. Distances of all sequences are then evaluated with respect to the seed sequences only. Using this emBedding, sequences can be either clustered directly or a full distance matrix can be approximated by computing vector distances.

The original mBed algorithm used those vectors to approximate a full pairwise distance matrix by computing vector distances, which are fast to compute. Here, we instead apply bisecting K-Means to the vectors as to create subclusters of a certain size. Within each cluster 'real' distances can be computed and clusters are later joined. Thereby we avoid the typical memory contraints encountered when computing a pairwise distance matrix needed to construct a guide-tree.

In Clustal Omega we use the k-tuple distance measure of Wilbur & Lipman (1983) as employed in ClustalW 1.83 and ClustalW2 (Larkin et al., 2007), to construct the distance matrix. If the sequences are already aligned (and are to be re-aligned during iteration, for example) then Kimura-corrected pairwise aligned identities (Kimura, 1983) are used. The vector distances in the mBed matrix are then used to cluster the sequences into subclusters using a bisecting k-means approach, which allows full control over the maximum cluster size and thereby avoids the usual memory limitations. For each subcluster a full distance matrix is constructed. As the size of the clusters affects the execution time, we have limited the maximum size of clusters. Currently, this limit is set to 100. All pairwise distances and a UPGMA sub-tree are calculated for each cluster. At this stage we use the fast UPGMA code as implemented in MUSCLE (Edgar, 2004). The sub-trees for the individual clusters are then linked up by an overarching tree built from the distance matrix of the clusters' barycentres using vector distances.

Clustal Omega can output the guide-tree used for the alignment, as well as input pre-computed guide-trees. Depending on the size of the alignment, re-cycling guide-trees presents a considerable saving in computation time. It is also possible to turn off the mBed mode and switch to full distance matrix calculation. In this particular mode it is possible to output the distance matrix. This is not possible in default (mBed) mode, as a full distance matrix is never calculated.

Profile HMM Alignment

For the actual sequence/profile alignment we use an adapted version of the HHalign package of Söding (2005). Here sequences/profiles are converted into profile HMMs which describe transition, insertion, and deletion probabilities, as well as emission probabilities for residues. Both kinds of probabilities are then augmented with pseudo-counts. HMMs for the two sequences/profiles are then aligned. If the sequences/profiles are short enough, such that the dynamic programming matrices fit into the computer's memory, then a Maximum Accuracy (MAC) algorithm, as proposed by Holmes & Durbin (1998) and extended to local pairwise alignment of profile HMMs in Biegert & Soeding (2007), is used. A computer with 2GB RAM can accommodate, for example, 2 sequences/profiles of over 6,500 residues in length, each.

For problems that exceed the available memory, the Viterbi algorithm is used. HHalign only aligns the overlapping core region of both HMMs and truncates the un-alignable end sections corresponding to end-gaps in the corresponding HMM. For the final alignment the aligned HMMs are reconverted into profiles and the truncated end sections are re-attached.

External Profile Alignment

External Profile Alignment or EPA involves using a profile HMM which describes an alignment of a set of sequences that are homologous to the input set. This is read from a file in HMMER2 or HMMER3 format. Clustal Omega takes this external HMM and aligns to it in turn the two sequences/profiles to be aligned to each other at each stage in progressive alignment. After alignment to the HMM, pseudo-count information for the residue frequencies is transferred from the external HMM to the HMMs describing the sequences/profiles, position by position. We do not transfer pseudo-count information for the transition probabilities. Finally, these 'softened-up' HMMs, representing the 2 sequences/profiles, are aligned using the MAC or Viterbi algorithm. EPA information is particularly useful during the early stages of progressive alignment. During the later alignment stages, the intermediate profiles have already accumulated enough information, so that pseudo-count transfer becomes less necessary. Clustal Omega therefore scales the contribution of the external HMM during progressive alignment. Pseudo-count transfer to profiles comprising of more than ten sequences is negligible. Each sequence/profile has to be pre-aligned to the external HMM. This represents a computational overhead of at most twice the original time required for aligning two sequences/profiles, so the final alignment time (not counting guide-tree construction) may triple.

Iteration

Iteration is done by iterating the alignment and, optionally, also the guide tree. The alignment itself is iterated using EPA while the guide tree is simply re-calculated from the full alignment produced in an earlier iteration. Initially, Clustal Omega aligns the input sequences. This alignment is then converted into a HMM, which can then be used for EPA as explained above. At the same time, the initial alignment can be used to construct a new guide-tree. At this stage Kimura-corrected (Kimura, 1983) pairwise aligned identities are used, rather than k-tuple distances. The guide-tree is, by default, still constructed using the mBed scheme. This can be switched to a full distance matrix. Using aligned sequences as input is equivalent to performing one iteration as the initial alignment is converted into a HMM and Kimura distances are used to construct the guide-tree. Iteration can be performed more than once, and HMM iteration can be decoupled from guide-tree iteration, that is, the background HMM can be frozen while the guide-tree is still being iterated, or vice versa. All of these options are set by users on the command-line of the program.

The effects of HMM iteration are illustrated in Supplementary Figure 1. Here the lower part of the figure displays the guide tree used for aligning the 6 sequences of BB12021, which is part of BAliBASE 3 (see below). The upper part of the figure shows an alignment generated with Clustal Omega, using default settings. This alignment achieves a Total Column score of 0.820 with respect to the BAliBASE 3 reference alignment. The middle part of the figure shows an alignment of the same sequences using one HMM iteration and no guide-tree iteration; this is achieved by setting --iter=1 --max-guidetree-iterations=0 on the command-line. This alignment has a TC score of 0.890. The alignments differ subtly in two places. According to the guide-tree, 1hpi_ and 2hip_A are to be aligned first. A locally desirable alignment is to match up the 6 residues at alignment positions 5-10. This achieves 3 perfect matches (E,E,D) while opening up a gap of length 3 in 2hip_A. The aligner does not have any foresight as to what other residues will align at these positions later during the alignment process. However, after the initial alignment an overall (background) residue distribution is available as a HMM. This information can be used to guide a re-alignment, by aligning the original sequences (1hpi_ and 2hip_A in this case) to the background HMM, transferring the HMMs pseudocounts, position for position to the original sequences and then aligning the ‘softened-up’ sequences. In this example the background ‘convinces’ 2hip_A to close the internal gap and move the 6-residue block by 3 positions to the right. A similar move of residues can be seen around positions 31-36. Both moves give rise to an improvement in the alignment’s TC score of 7%.

Benchmarks

In order to assess the performance and quality of Clustal Omega alignments, we compared it to various other alignment engines using three different benchmarks. The set of tested alignment programs is:

* ClustalW2, v2.1 [http://www.clustal.org]

* Clustal Omega v1.0.2 [http://www.clustal.org]

* DIALIGN 2.2.1 [http://dialign.gobics.de/]

* FSA 1.15.5 [http://sourceforge.net/projects/fsa/]

* Kalign 2.04 [http://msa.sbc.su.se/cgi-bin/msa.cgi]

* MAFFT 6.857 [http://mafft.cbrc.jp/alignment/software/source.html]

* MSAProbs 0.9.4 [http://sourceforge.net/projects/msaprobs/files/]

* MUSCLE version 3.8.31 posted 1st May 2010 [http://www.drive5.com/muscle/downloads.htm]

* PRANK v.100802, 02-Aug-2010 [http://www.ebi.ac.uk/goldman-srv/prank/src/prank/]

* Probalign v1.4 [http://cs.njit.edu/usman/probalign/]

* PROBCONS version 1.12 [http://probcons.stanford.edu/download.html]

* T-Coffee Version 8.99
[http://www.tcoffee.org/Projects_home_page/t_coffee_home_page.html#DOWNLOAD]

As benchmarks we use BAliBASE 3 (Thompson et al., 2005), PREFAB 4.0 (as posted, March 2005) (Edgar, 2004) and a newly constructed data set (HomFam) using sequences from Pfam (version 25) and Homstrad (as of 2011-06-13) (Mizuguchi et al., 1998).

The BAliBASE benchmark is comprised of 218 families, grouped into 38+44 families exhibiting variability and length (BB11+BB12), 41 families containing orphan sequences (BB2), 30 families with sub-families (BB3), 49 families with extensions (BB4) and 16 families with insertions (BB5). The smallest BAliBASE alignments are comprised of 4 sequences; the largest of 142 sequences; the average is 28.5. The longest sequence in BAliBASE is 7923 residues long. Reference alignments vary between 52 and 8481 positions. The alignments are scored using the bali_score program [http://bips.u-strasbg.fr/fr/Products/Databases/BAliBASE/bali_score.c], which considers only core regions of the alignments. We also used Qscore (see below) to score the core regions of the alignments. The numerical values of scores were slightly different but not different enough to change the relative ranking of the different alignment algorithms' accuracy. We only quote the Total Column Score, giving the number of columns exactly shared by the test and reference alignments.

The PREFAB benchmark is comprised of 1682 families. All families consist of 50 test sequences. The alignments are scored by comparing to reference alignments of two sequences. The entire benchmark set can be grouped according to the pairwise identities of the reference sequences. There are 912 families with 0%-20% pairwise identity of reference sequences, 563 families in the range 20%-40%, 117 families in the range 40%-70% and 90 families in the range 70%-100%. Reference alignments vary in length from 54 to 1047 positions. The longest sequence in PREFAB is 1132 residues long (which is not a reference sequence). Alignments are scored using the Qscore program [http://www.drive5.com/qscore/], and we also only quote the Total Column Score.

For the results in Table 1 and Table 2 we evaluated the above mentioned 12 alignment programs using BAliBASE and PREFAB. For these two benchmarks, all programs are run with default command-line arguments, with the exception of MAFFT, where we used the default and the auto mode. All programs are run single threaded, total times are therefore single core times.

Both benchmark sets, BAliBASE and PREFAB, are comprised of families consisting of at most 142 sequences or 50 sequences, respectively. Neither of which is sufficient to explore scalability to large numbers of sequences. We therefore created a third benchmark set of testcases called HomFam, with very large numbers of sequences. For this we blended all Homstrad (Miziguchi, et al., 1998) families with more than 5 reference sequences with their corresponding Pfam families, if there was a one-to-one link, similar to the previously used approach in (Blackshields et al., 2010). Then we generate a test alignment using the full set of sequences, including the small number from Homstrad. The alignment of the Homstrad sequences in the test alignment is compared to the alignment in the Homstrad structure based alignment using Qscore. The largest Homstrad (reference) family in this third benchmark set is comprised of 41 reference sequences. Alignments vary in length from 39 to 938 positions. The HomFam dataset can be downloaded from http://www.clustal.org/omega/homfam.tgz

There are 94 HomFam families, which we grouped into 41 families with less than 3,000 (93-2,957) sequences, 20 families with more than 10,000 (10,099-93,681) sequences and 33 families in the intermediate sequence number range (3,127-9,105). We settled on a lower cut-off of 3,000 because default MUSCLE performed unacceptably slowly in our test setting for most families with more than 3,000 sequences. Default MAFFT cannot deal with more than 20,000 sequences. However, there are only 8 families with more than 20,000 sequences in the HomFam benchmark. This would have been too small a grouping, so we decided on a cut-off of 10,000, which then split the benchmark more evenly. For scoring the alignments we used the Qscore program, scoring the entire alignment. Only the embedded Homstrad reference sequences were used for scoring the entire alignment.

Given the computational demands of the HomFam benchmark, we could only evaluate four programs. These were: Clustal Omega, Kalign, MAFFT and MUSCLE. For the sub-group of small families (< 3,000 sequences) all programs were run using default settings. For the sub-group of intermediate families (3,000-10,000 sequences) all programs were run using default settings, except for MUSCLE, where -maxiters was set to 2. For the sub-group of large families (> 10,000 sequences) Clustal Omega and Kalign were run with default settings, MUSCLE was run with -maxiters set to 2 and MAFFT was run using the --parttree flag. Three families contained 'U' residues; where this occurred the --anysymbol flag was used in MAFFT. On our machine (Intel(R) Xeon(R) X5470@3.33GHz, 8 core, 46.8GB RAM) MUSCLE and Kalign did terminate the largest two runs with a signal 11 (segmentation violation).
These are the CCHH zinc-finger proteins (Homstrad ID: zf-cchh and Pfam accession: PF00096) with 88,345 sequences and 16 known structures and the retroviral proteases (Homstrad ID: rvp and Pfam accession: PF00077) with 93,681 sequences and 6 known structures. MAFFT --parttree did better on the zinc finger proteins (column score 0.457 vs. 0.143) but Clustal Omega did better on the proteases (column score 0.405 vs. 0.324). Comparison of overall execution time and average accuracy for the large families in Table 3 is therefore given for 18 families (instead of 20) only. The results for the 2 largest families for MAFFT/PartTree and Clustal Omega are included in Figure 1.

Results for the BAliBASE benchmark can be found in Table 1. Supplementary Figure 2 is a graphical representation of the summary results of Table 1. There we plot the average total column score against the total run time (to align all 218 families). Programs using the consistency principle are depicted as blue dots, progressive aligners as red dots, and programs that fit neither category as purple dots.

We have added two programs to Supplementary Figure 2 which are not listed in Table 1. These are:

* Opal v2.0.0 http://opal.cs.arizona.edu (Wheeler et al., 2007)

* SATe v1.4.0 http://phylo.bio.ku.edu/software/sate/sate.html (Liu et al., 2009)

The reason for not including SATe in the main result section was that it was difficult to decide on a convergence criterion that fitted into the current benchmarking scheme of assessing quality as well as performance. SATe's default convergence criterion is to take the best solution arrived at after 24 hours. Not only would this take much longer to evaluate than all the other aligners, it also does not scale with the size of the problem. Therefore we tried to find a criterion that would make SATe finish within a similar time frame as the other aligners. We used the --iter-without-imp-limit flag which specifies the maximum number of iterations without an improvement in score that the SATe algorithm will run. If the number is less than 1, then no iteration limit will be used. Setting --iter-without-imp-limit=1 already took approximately 350,000 seconds. While selecting a higher value for --iter-without-imp-limit might improve on the accuracy of the results (0.539) it would also increase the run time beyond the limits of this study. In default mode SATe uses MAFFT (v6.717) for aligning and merging sub-problems and RAxML (v7.2.6) to infer trees for sub-problems.

Unlike the programs in the main result section, which are compiled C/C++ codes, Opal is written in Java. Nevertheless its performance and accuracy are respectable (15,250 seconds to achieve a TC score of 0.534).

Scalability

The scaling of MAFFT and Clustal Omega with large numbers of sequences is explored in more detail in Supplementary Figure 3 using the biggest family in Pfam as an example. These are the bacterial ABC transporters which have over 160,000 sequences in Pfam and almost 200,000 sequences in the NCBI sequence databases. Subsets of different numbers of sequences are aligned and the times and memory usage and benchmark accuracy are plotted. In Figure 3a the run times of the methods can be seen to scale linearly on log-log plots over the full range of test case sizes with MAFFT up to 10 times faster than Clustal. In Figure 3b, the peak memory usage is given. These are similar over most of the range with MAFFT tending to use more peak memory at high N. In Figure 3c, the average memory usage is given. Here Clustal uses, on average, about a tenth of the memory of MAFFT. Finally, the accuracy of the alignment of the six included Homstrad sequences is plotted against N. Here both methods tend to give lower and lower quality alignments as more sequences are added but with Clustal Omega almost 20% more accurate over the entire range. Overall, this figure shows that MAFFT, with the PartTree option is indeed very fast and scales extremely well with large N but at the expense of memory usage and accuracy. Both methods suffer at high N. This runs counter to the expectation in the field that increasing the number of sequences will increase alignment accuracy. Accuracy will probably increase if you add in extra sequences with very small N say in the range of 10 to 100 sequences and when combined with a consistency based method such as T-Coffee. At very large N however, it looks like, adding in more and more sequences simply makes the problem

more demanding. There is more and more noise added due to sequence errors and outliers and the computational landscape becomes enormously more complicated. This can be compensated for in two ways as described in the main section: 1) by alignment to an external profile and 2) by iteration.

Software

Clustal Omega is licensed under the GNU Lesser General Public License. Source code as well as precompiled binaries for Linux, FreeBSD, Windows, and Mac (Intel and PowerPC) are available at http://www.clustal.org. Clustal Omega is available as a command line program only, which uses GNU-style command line options, and also accepts ClustalW-style command options for backwards compatibility and easy integration into existing pipelines.

Clustal Omega is written in C and C++ and makes use of a number of excellent free software packages, which are listed in the following: A modified version of Sean Eddy's Squid library (http://selab.janelia.org/software.html) is used for sequence I/O, allowing the use of a wide variety of file formats. We use David Arthur's k-means++ code (Arthur & Vassilvitskii, 2007) for fast clustering of sequence vectors. Code for fast UPGMA and guide tree handling routines was adopted from MUSCLE (Edgar, 2006). We use the OpenMP library to enable multithreaded computation of distance matrices and alignment match states. The documentation for Clustal Omega's API is part of the source code, and in addition is available from http://www.clustal.org/omega/clustalo-api/.

Figures

Supplementary Figure 1: Effect of HMM iteration. (Top) alignment of BB12021 using default Clustal Omega. (Middle) alignment of BB12021 using Clustal Omega and one HMM iteration. (Bottom) guide tree for aligning BB12021.

Supplementary Figure 2: Average Total Column scores plotted against total time for BAliBASE 3. X-axis logarithmic, y-axis linear. Consistency based methods depicted as blue dots, progressive methods as red dots, other methods in purple.

Supplementary Figure 3: Computational demands and accuracy results for the largest HomFam family: the bacterial ABC transporters. These include 6 Homstrad sequences with known structures and 194,587 sequences from the NCBI protein sequence databases. Clustal Omega results are plotted in red and MAFFT --PartTree results in blue. All axes use logarithmic scales except the vertical axis in panel (d) which uses a linear scale. (a) Single-threaded run time plotted against number of sequences. (b) Peak memory usage plotted against number of sequences. (c) Average memory usage plotted against number of sequences. (d) Total Column score of the 6 embedded Homstrad reference sequences, plotted against total number of test sequences.

References

Arthur, D., Vassilvitskii, S. (2007). *k*-means++: the advantages of careful seeding. *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. pp. 1027–1035.

Biegert, A., Söding, J., (2008) De novo identification of highly diverged protein repeats by probabilistic consistency, Bioinformatics 24 (6), 807-814

Blackshields, G., Sievers, F., Shi, W., Wilm, A., and Higgins, D. G. (2010). Sequence emBedding for fast construction of guide trees for multiple sequence alignment. Algorithms Mol Biol 5, 21.

Holmes, I., and Durbin, R. (1998). Dynamic programming alignment accuracy. J. Comput. Biol 5, 493-504.

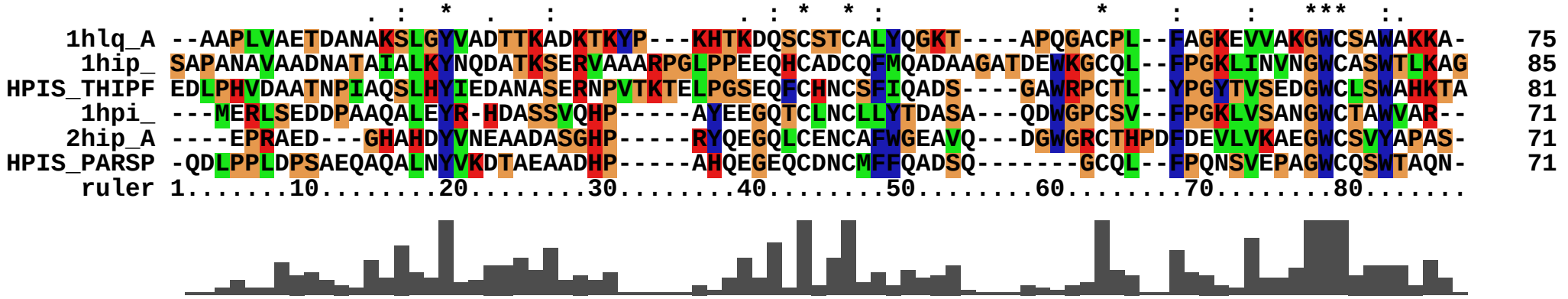
Kimura, M. (1983). The Neutral Theory of Molecular Evolution, page 75. Cambridge University Press, Cambridge, UK.

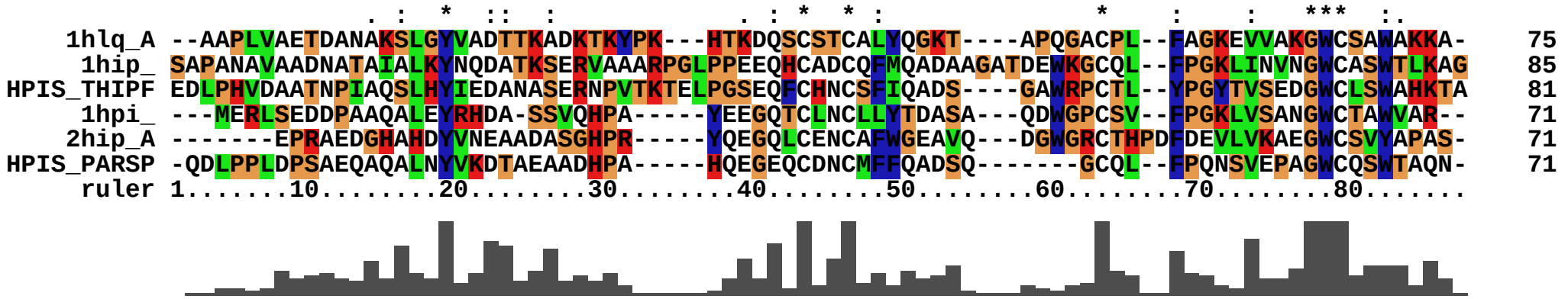
Linial, N., London, E., and Rabinovich, Y. (1995). The geometry of graphs and some of its algorithmic applications. Combinatorica 15, 215-245.

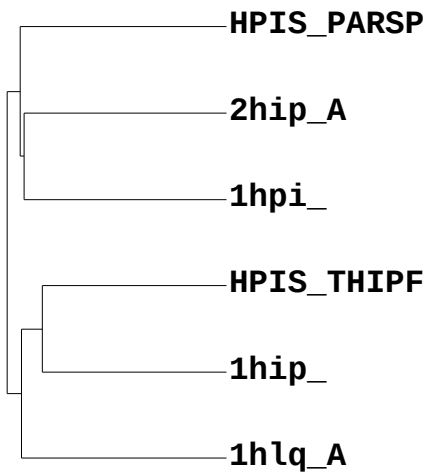
Liu, K., S. Raghavan, S. Nelesen, C. R. Linder, T. Warnow (2009). Rapid and accurate large scale coestimation of sequence alignments and phylogenetic trees. Science, vol. 324, no. 5934, pp. 1561-1564.

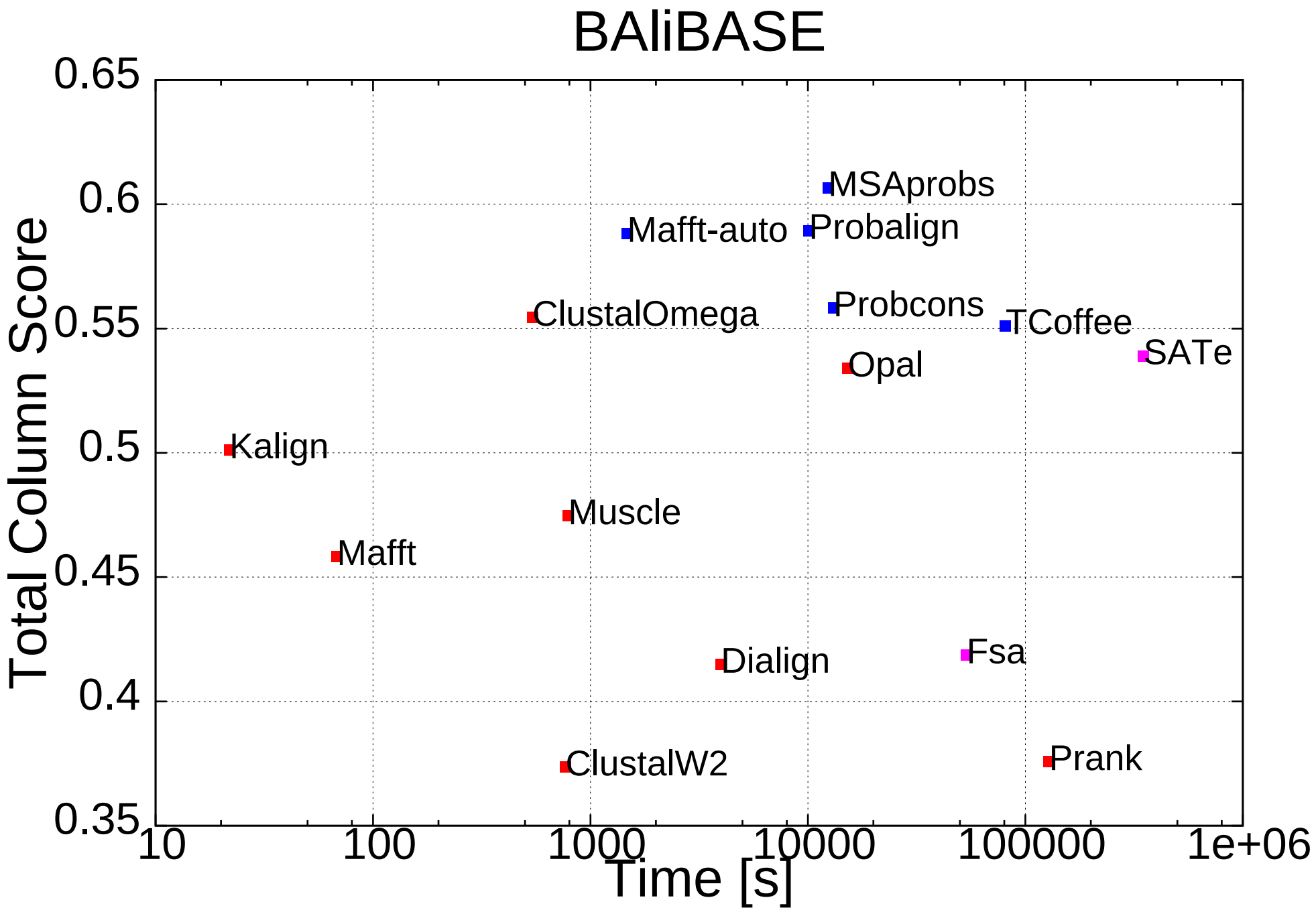
Wheeler, T.J. and Kececioglu, J.D. (2007). Multiple alignment by aligning alignments, Proceedings of the 15th ISCB Conference on Intelligent Systems for Molecular Biology, Bioinformatics 23, i559-i568.

Wilbur, W. J., and Lipman, D. J. (1983). Rapid similarity searches of nucleic acid and protein data banks. Proceedings of the National Academy of Sciences 80, 726 -730.

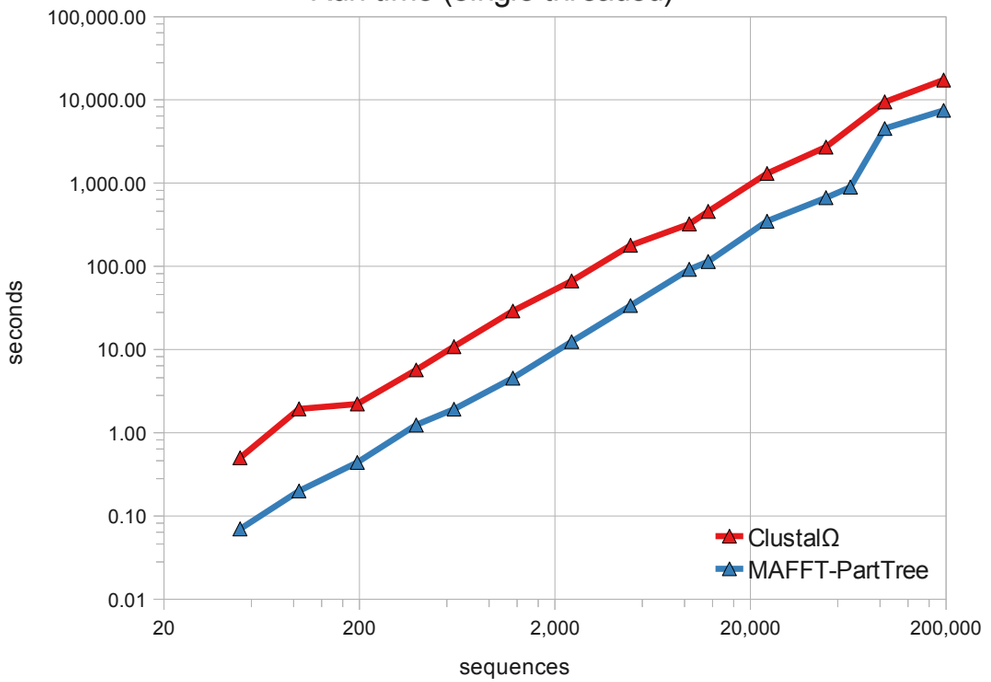

1hlq_A --AAPLVAETDANAKSLGYVADTTKADKTKYP---KHTKDQSCSTCALYQGKT----APQGACPL--FAGKEVVAKGWCSAWAKKA- 75
1hip_ SAPANAVAADNATAIALKYNQDATKSERVAAARPGLPPEEQHCADCQFMQADAAGATDEWKGCQL--FPGKLINVNGWCASWTLKAG 85
HPIS_THIPF EDLPHVDAATNPIAQSLHYIEDANASERNPVTKTELPGSEQFCHNCSFIQADS----GAWRPCTL--YPGYTVSEDGWCLSWAHKTA 81
1hpi_ ---MERLSEDDPAAQALEYR-HDASSVQHP-----AYEEGQTCLNCLLYTDASA---QDWGPCSV--FPGKLVSANGWCTAWVAR-- 71
2hip_A ----EPRAED---GHAHDYVNEAADASGHP-----RYQEGQLCENCAFWGEAVQ---DGWGRCTHPDFDEVLVKAEGWCSVYAPAS- 71
HPIS_PARSP -QDLPPLDPSAEQAQALNYVKDTAEAADHP-----AHQEGEQCDNCMFFQADSQ-------GCQL--FPQNSVEPAGWCQSWTAQN- 71
ruler 1.....10.....20.....30.....40.....50.....60.....70.....80.....


1hlq_A --AAPLVAETDANAKSLGYVADTTKADKTKYPK---HTKDQSCSTCALYQGKT----APQGACPL--FAGKEVVAKGWCSAWAKKA- 75
1hip_ SAPANAVAADNATAIALKYNQDATKSERVAAARPGLPPEEQHCADCQFMQADAAGATDEWKGCQL--FPGKLINVNGWCASWTLKAG 85
HPIS_THIPF EDLPHVDAATNPIAQSLHYIEDANASERNPVTKTELPGSEQFCHNCSFIQADS----GAWRPCTL--YPGYTVSEDGWCLSWAHKTA 81
1hpi_ ---MERLSEDDPAAQALEYRHDA-SSVQHPA-----YEEGQTCLNCLLYTDASA---QDWGPCSV--FPGKLVSANGWCTAWVAR-- 71
2hip_A -------EPRAEDGHAHDYVNEAADASGHPR-----YQEGQLCENCAFWGEAVQ---DGWGRCTHPDFDEVLVKAEGWCSVYAPAS- 71
HPIS_PARSP -QDLPPLDPSAEQAQALNYVKDTAEAADHPA-----HQEGEQCDNCMFFQADSQ-------GCQL--FPQNSVEPAGWCQSWTAQN- 71
ruler 1.....10.....20.....30.....40.....50.....60.....70.....80.....

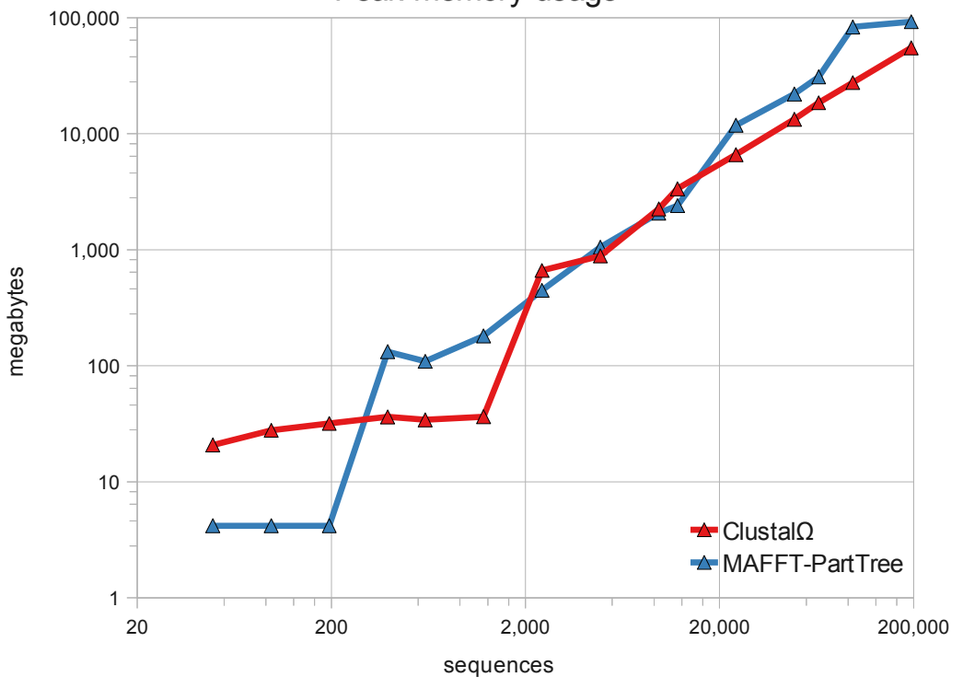

HPIS_PARSP
2hip_A
1hpi_
HPIS_THIPF
1hip_
1hlq_A


BAliBASE
Total Column Score
Time [s]
0.65
0.6
0.55
0.5
0.45
0.4
0.35
10
100
1000
10000
100000
1e+06
MSAprobs
Mafft-auto
Probalign
ClustalOmega
Probcons
TCoffee
SATe
Opal
Kalign
Muscle
Mafft
Dialign
Fsa
ClustalW2
Prank

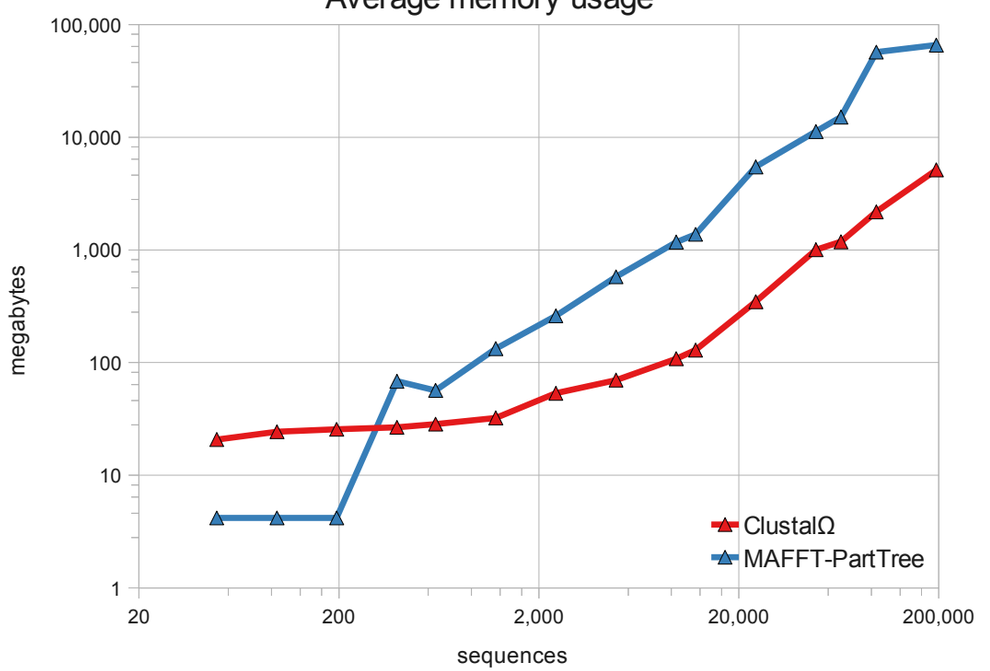
(a)


Run time (single threaded)
100,000.00
10,000.00
1,000.00
100.00
10.00
1.00
0.10
0.01
seconds
20
200
2,000
20,000
200,000
sequences
ClustalΩ
MAFFT-PartTree

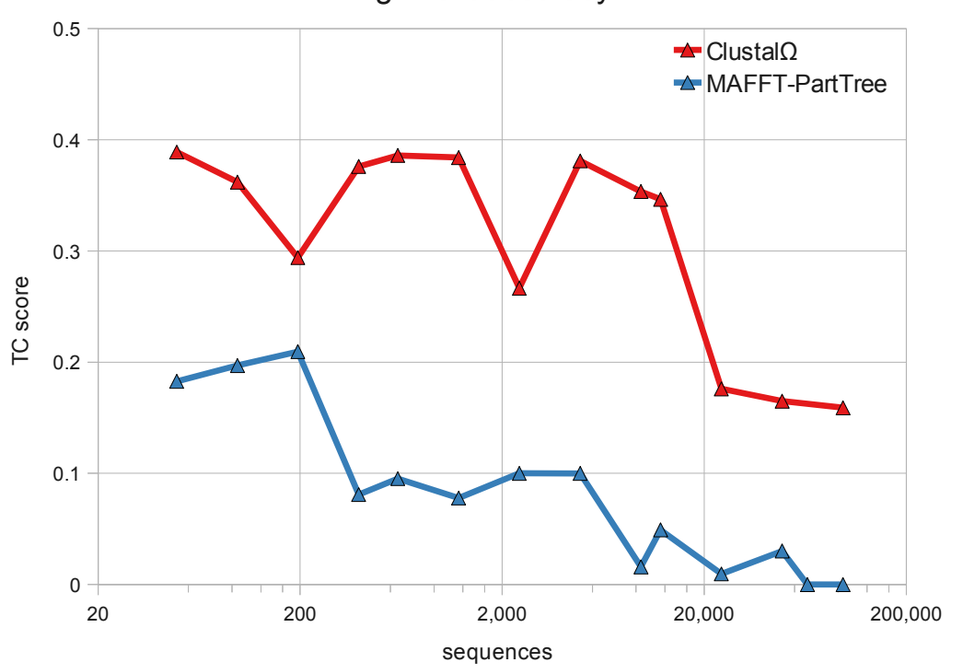
(b)


Peak memory usage
100,000
10,000
1,000
100
10
1
megabytes
20
200
2,000
20,000
200,000
sequences
ClustalΩ
MAFFT-PartTree

(c)


Average memory usage
100,000
10,000
1,000
100
10
1
megabytes
20
200
2,000
20,000
200,000
sequences
ClustalΩ
MAFFT-PartTree

(d)


Alignment accuracy
0.5
0.4
0.3
0.2
0.1
0
TC score
20
200
2,000
20,000
200,000
sequences
ClustalΩ
MAFFT-PartTree