Fast and accurate large multiple sequence alignments using root-to-leave regressive computation

Authors: Edgar Garriga[1,2], Paolo Di Tommaso[1], Cedrik Magis[1,2], Ionas Erb[1], Hafid Laayouni[3,4], Fyodor Kondrashov[5], Evan Floden[1] *, Cedric Notredame[1,2] *

Affiliations:

[1]Centre for Genomic Regulation (CRG), The Barcelona Institute of Science and Technology, Dr. Aiguader 88, Barcelona 08003, Spain.

[2]Universitat Pompeu Fabra (UPF), Dr. Aiguader 88, Barcelona 08003, Spain.

[3]Institut de Biologia Evolutiva (UPF-CSIC), Universitat Pompeu Fabra, Barcelona, Catalonia, Spain.

[4]Bioinformatics Studies, ESCI-UPF, Barcelona, Spain.

[5]Institute of Science and Technology, Klosterneuburg, Austria.

*Correspondence to: evan.floden@crg.eu (E.F); cedric.notredame@crg.eu (C.N)

Abstract:

Inferences derived from large multiple alignments of biological sequences are critical to many areas of biology, including evolution, genomics, biochemistry, and structural biology. However, the complexity of the alignment problem imposes the use of approximate solutions. The most common is the progressive algorithm, which starts by aligning the most similar sequences, incorporating the remaining ones following the order imposed by a guide-tree. We developed and validated on protein sequences a regressive algorithm that works the other way around, aligning first the most dissimilar sequences. Our algorithm produces more accurate alignments than non-regressive methods, especially on datasets larger than 10,000 sequences. By design, it can run any existing alignment method in linear time thus allowing the scale-up required for extremely large genomic analyses.

One Sentence Summary:

Initiating alignments with the most dissimilar sequences allows slow and accurate methods to be used on large datasets

Main Text:

Structural and evolutionary predictions are improved when using accurate large Multiple Sequence Alignments (MSAs) featuring thousands of sequences (*1*, *2*). Until the first benchmarking of large-scale MSAs, analyses made on smaller datasets had suggested that scale-up would automatically lead to increased accuracy (*3*). However, all things being equal, alignments with over a thousand sequences are less accurate than their smaller counterparts (*4*). Under the most common multiple alignment procedure (*5*), small MSAs are built in parallel and progressively merged into increasingly large intermediate MSAs, following the order of a guide-tree. This sometimes results in extremely gapped MSAs (*6*) that have been speculated to account for the accuracy drop on large datasets (*7*). Recent attempts to address this problem have included SATé (*8*) and its follow-up PASTA (*9*), a progressive algorithm in which the guide-tree is split into subsets that are independently aligned and later merged. This divide and conquer strategy allows computationally intensive methods to be deployed on large datasets but does not alleviate the challenge of merging very large intermediate MSAs. More recent alternatives include the MSA algorithms UPP (*10*) and MAFFT-Sparsecore (Sparsecore) (*11*). Both rely on selecting a subset of sequences - the seeds - and turning them into a Hidden Markov Model (HMM) using either PASTA or the slow/accurate version of MAFFT. The HMM is then used to incorporate all the remaining sequences one by one. The downside of this approach is that the seed sequences can be insufficiently diverse and therefore preclude the accurate alignment of distantly related homologous to the seed HMM. We propose a regressive algorithm that addresses this problem by combining the benefits of a progressive approach when incorporating distant homologues with the improved accuracy of seeded methods. This synthesis was achieved by fulfilling two simple constraints: the splitting of the sequences across sub-MSAs each containing a limited number of sequences and their combination into a final MSA without the requirement of an alignment procedure. Thus, the main difference between our approach and existing ones lies in the order in which sequences are aligned, starting with the most diverse.

Given M sequences the sub-MSAs are collected as follows. A clustering algorithm is used to define N non-overlapping sequence groups - the children. The first sub-MSA - the parent - is computed by selecting a representative sequence from each child group. The same method is then applied onto every child group in which N new representatives are collected and multiply aligned to yield one child sub-MSA for each sequence in the parent. The procedure runs recursively by treating each child as a parent for the next generation until every sequence has been incorporated. The final MSA is produced by merging all the sub-MSAs. The merging of a child with its parent is done without additional alignment thanks to the representative sequence. This sequence, present in both the child and its parent, enables the stacking of the corresponding positions (Fig. 1A). When doing so, insertions occurring within the representative, either in the child or in its parent, are projected as deletions (i.e. gaps) in the other.

A key step of this recursion is the clustering method and the subsequent selection of the N representative sequences. N was set to 1,000, a figure reported to be the largest number of sequences that can be directly aligned without accuracy loss (*4*). The clusters were estimated from binary guide-trees produced by existing large-scale MSA algorithms such as Clustal Omega (ClustalO) and MAFFT. The use of a binary tree to extract the most diverse sequences was inspired by an existing taxa sampling procedure (*12*). In our implementation, every node gets labeled with the longest sequence among its descendants. Given a fully labeled tree, the sequences of the first parent sub-MSA are collected by the breadth-first traversal of the tree, starting from the root through as many generations as required to collect N sequences (Fig. 1B). Because of the way they are collected along the tree these N first sequences are as diverse as possible. Within the resulting sub-MSA every sequence is either a leaf or the representative of an internal node ready to processed (Fig. 1C).

Our algorithm does not depend on specific alignment or guide-tree methods and therefore lends itself to be combined with any third-party software. This property enabled us to run various alignment software both directly and in combination with the regressive algorithm. A combination involves estimating a guide-tree with an existing method, collecting sequences with the regressive algorithm and then computing the sub-MSAs with an existing MSA algorithm. By doing so we were able to precisely quantify the impact of our algorithm on both accuracy and computational requirements. We used as a benchmark the HomFam protein datasets (*4*) in which sequences with known structures - the references - are embedded among large numbers of homologues. Accuracy is estimated by aligning the large dataset and then comparing the induced alignment of the references with a structure-based alignment of these same references (*13*). We started by benchmarking the ClustalO and MAFFT-FFT-NS-1 (Fftns1) MSA algorithms using two guide-tree methods: ClustalO embedded k-means trees (mBed) (*14*) and MAFFT-PartTree (PartTree) (*15*). These widely adopted software were selected because they support large-scale datasets, are strictly progressive and allow the input and output of binary guide-trees.

In three out of four combinations of guide-tree and MSA algorithms, the regressive combination outperformed the progressive one. When considering the most discriminative measure (total column score, TC, Table 1) on the datasets with over 10,000 sequences, the regressive combination delivered MSAs that were on average 5.13 percentage points more accurate than when computed progressively (39.31 and 34.24 respectively). These differences remained comparable, albeit reduced, when considering the contribution of smaller datasets (Table S1, S2). Within this first set of analysis, the regressive combination of ClustalO with PartTree was the most accurate and on the large datasets it outperformed its progressive counterpart by 15.27 percentage points (42.21 and 26.94 respectively, Wilcoxon p-value <0.001).

We also tested the seed-based non-progressive MSA algorithms Sparsecore (*10*) and UPP (*11*). Both were improved when combined with the regressive algorithm. For instance, the regressive combination of Sparsecore with mBed guide-trees yielded the best readouts of this study on the very large alignments, and a clear improvement over the default Sparsecore (TC score 51.07 vs. 44.98, Wilcoxon p-value<0.1). Comparable results were observed when extending this analysis to the Sum-of-Pair (SoP) metrics or to smaller datasets (Table S1, S2). The regressive algorithm is especially suitable for the scale-up of computationally expensive methods. For instance, the consistency-based variant of MAFFT named MAFFT-G-INS-1 (Gins1) (*16*), was among the most accurate small-scale MSA algorithms on the reference sequences. Gins1 cannot, however, be deployed on the HomFam datasets because its computational requirements are cubic with the number of sequences thus restricting it to a few hundred sequences. By combining Gins1 with the regressive algorithm we overcame this limitation and produced the most accurate readouts on datasets larger than 1,000 sequences (Table S1, S2).

We complemented these measures of absolute accuracy with an estimate of accuracy degradation when scaling up. The effect of extra homologous sequences degrading the alignment accuracy of an MSA can be quantified by comparing the small MSAs of the reference sequences alone with their corresponding large-scale datasets. With the default progressive MSA algorithms ClustalO and Fftns1, the large datasets were on average 16.79 percentage points less accurate than when aligning the reference sequences (Table 1, 34.24 and 51.03 respectively) with the trend being amplified on the larger alignments (Fig. 2A). Yet, on this same comparison the regressive combinations were only affected by 11.72 points (Fig. S1). The improved stability of the regressive combination was especially clear when considering Gins1 (Fig. 2A, Fig. S1A) that was merely degraded by 2.87 percentage points thus achieving on the large datasets a level of accuracy close to the one measured on the reference sequences alone (Table 1, 50.20 and 53.07 respectively).

Identifying the factors driving accuracy improvement can be challenging considering that each alignment procedure relies on different combinations of algorithmic components (i.e. regressive/non-regressive, tree method, MSA algorithm). For this purpose, we used Constrained Correspondence Analysis (CCA) (*17*), a dimensionality reduction method adapted for categorical variables. When applied to Table 1 data, CCA allowed us to estimate the relative impact of each method's algorithmic component with respect to a constrained variable - accuracy in this case. As one would expect, the MSA algorithm is the most influential variable with respect to accuracy but CCA confirmed the general benefits of switching from a non-regressive to a regressive combination (Fig. 2B).

When using the same guide-tree for the regressive and non-regressive alignment combinations, improved accuracy comes along with substantially improved computational performance. On average the regressive combinations require about 4-fold less CPU time that their non-regressive equivalent on datasets larger than 10,000 sequences (Table 1). Seeded methods like UPP or Sparsecore appear to benefit less from the regressive deployment with marginal differences in CPU requirements (Fig. 3A). When considering MSA algorithms like ClustalO or Fftns1 that scale linearly with the number of sequences, the improvement yielded by the regressive combination was roughly proportional to the original non-regressive CPU requirements. For instance, in the case of ClustalO using mBed trees, the regressive combination was about twice as fast as the progressive alignment and appeared to have a linear complexity (Fig. 3B). The situation was even more favorable when considering CPU intensive MSA algorithms like Gins1 for which the non-regressive computation had been impossible.

The ability to use slow and accurate MSA algorithms in linear time regardless of their original computational complexity is the most important feature of the regressive algorithm. It allows the application of any of these methods - natively - onto extremely large sequence datasets. This linearization is an inherent property of the regressive procedure in which all the sequences are split across sub-MSAs of a bounded size (1,000 sequences). This bounding in size results in a bounded computational cost. Since the total number of sub-MSAs is proportional to the initial number of sequences, the resulting complexity for the full MSA computation is linear. Furthermore, owing to the computational independence of the sub-MSAs, the regressive algorithm turns MSA computation into an embarrassingly parallel problem (*18*).

Our regressive algorithm provides a practical and generic solution to the critical problem of MSA scalability. It is a versatile algorithm that lends itself to further improvements, for instance by exploring the impact of more sophisticated clustering structures, such as k-guide-trees and b-guide-trees or by testing different ways of selecting the representative sequences. The regressive algorithm is nonetheless a mature development framework that will enable a clean break between the improvement of highly accurate small-scale MSA algorithms - like Gins1 - and the design of more efficient large-scale clustering algorithms, like PartTree and mBed. This divide will help potentiate the large body of work carried out in the clustering and alignment communities over the last decades and hopefully speed up the development of new improved methods. Achieving this goal is not optional. There is a Red Queen's race going on in genomics. It started the day omics' data growth overtook computing power and it shows no signs of slowing down (*19*).

References and Notes:

1. G. Uguzzoni et al., Large-scale identification of coevolution signals across homo-oligomeric protein interfaces by direct coupling analysis. Proc. Natl. Acad. Sci. U. S. A. 114, E2662–E2671 (2017).

2. S. Mirarab, M. S. Bayzid, B. Boussau, T. Warnow, Statistical binning enables an accurate coalescent-based estimation of the avian tree. Science. 346, 1250463 (2014).

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

4. F. Sievers, D. G. Higgins, in Methods in Molecular Biology (2013), pp. 105–116.

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

6. M. S. Breen, C. Kemena, P. K. Vlasov, C. Notredame, F. A. Kondrashov, Epistasis as the primary factor in molecular evolution. Nature. 490, 535–538 (2012).

7. M. Chatzou et al., Multiple sequence alignment modeling: methods and applications. Brief. Bioinform. 17, 1009–1023 (2016).

8. K. Liu, S. Raghavan, S. Nelesen, C. R. Linder, T. Warnow, Rapid and accurate large-scale coestimation of sequence alignments and phylogenetic trees. Science. 324, 1561–1564 (2009).

9. S. Mirarab et al., PASTA: Ultra-Large Multiple Sequence Alignment for Nucleotide and Amino-Acid Sequences. J. Comput. Biol. 22, 377–386 (2015).

10. N.-P. D. Nguyen, S. Mirarab, K. Kumar, T. Warnow, Ultra-large alignments using phylogeny-aware profiles. Genome Biol. 16, 124 (2015).

11. K. D. Yamada, K. Tomii, K. Katoh, Application of the MAFFT sequence alignment program to large data-reexamination of the usefulness of chained guide trees. Bioinformatics. 32, 3246–3251 (2016).

12. B. Q. Minh, S. Klaere, A. von Haeseler, Phylogenetic diversity within seconds. Syst. Biol. 55, 769–773 (2006).

13. L. A. Stebbings, K. Mizuguchi, HOMSTRAD: recent developments of the Homologous Protein Structure Alignment Database. Nucleic Acids Res. 32, D203–7 (2004).

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

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

16. K. Katoh, K.-I. Kuma, H. Toh, T. Miyata, MAFFT version 5: improvement in accuracy of multiple sequence alignment. Nucleic Acids Res. 33, 511–518 (2005).

17. M. J. Greenacre, Biplots in Practice (Fundacion BBVA, 2010).

18. M. Herlihy, N. Shavit, The Art of Multiprocessor Programming, revised first edition (Morgan Kaufmann, 2012).

19. S. D. Kahn, On the future of genomic data. Science. 331, 728–729 (2011).

Acknowledgments: We thank Guy Riddihough for revisions and comments on the manuscript and Olivier Gascuel for suggestions. **Funding**: This project was supported by the Center for Genomic Regulation and the Spanish Plan Nacional and the Spanish Ministry of Economy and Competitiveness, 'Centro de Excelencia Severo Ochoa'. **Author contributions**: C.N. designed and implemented the algorithm, E.F., E.G. and P.D.T carried out the validation, I.E. performed the CCA analysis and all the authors designed the validation procedure and wrote the manuscript. **Competing interests:** Authors declare no competing interests. **Data and materials availability:** The regressive alignment algorithm has been implemented in T-Coffee and is available on GitHub for immediate use (https://github.com/cbcrg/tcoffee). All data, analyses and results are available from Zenodo (https://doi.org/10.5281/zenodo.2025847). A GitHub repository containing the Nextflow workflow and Jupyter notebooks to replicate the analysis in full is available at https://github.com/cbcrg/dpa-analysis (release v1.0). See supplementary materials for a written description of methods.

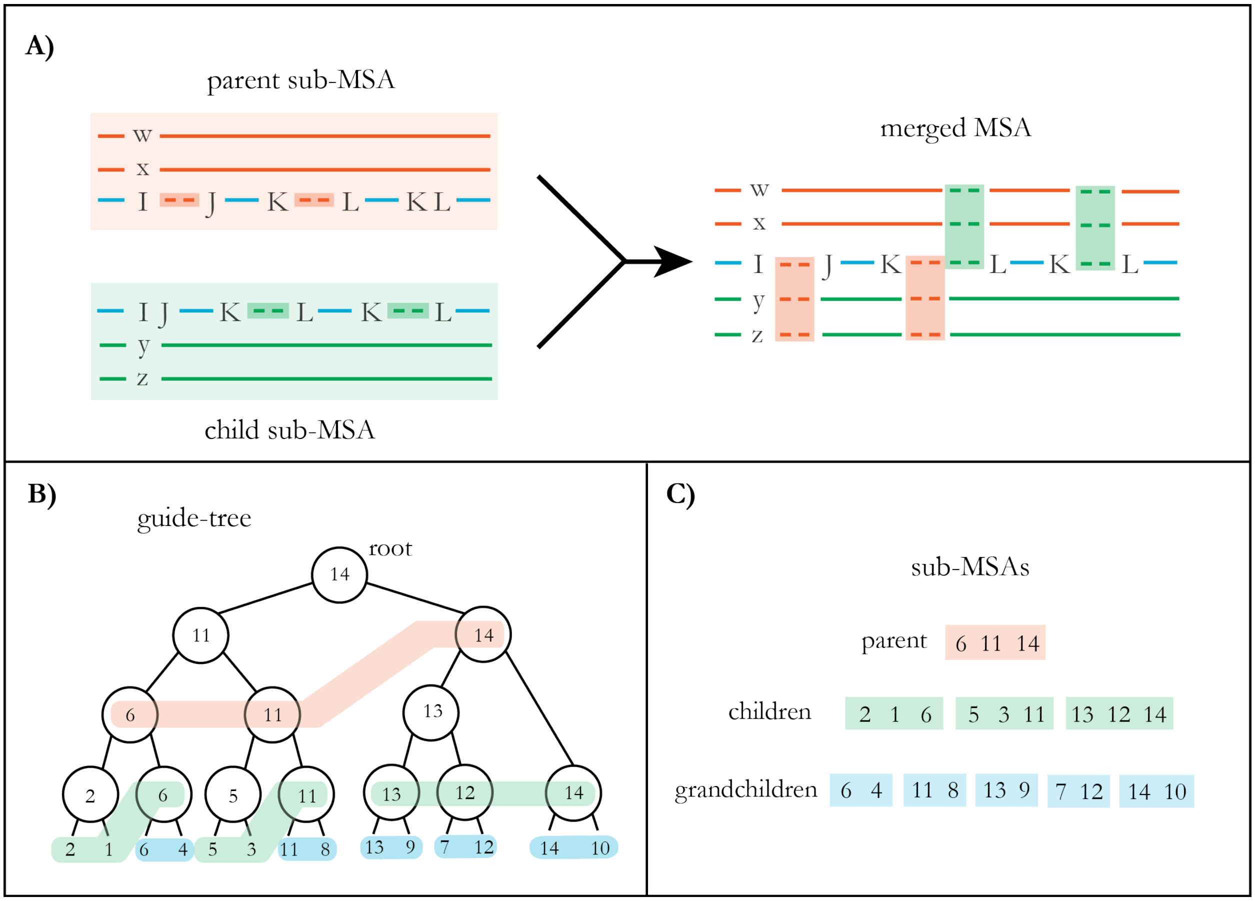

Fig. 1.

Regressive algorithm overview (**A**) Parent and children sub-MSAs are merged via their common sequence (blue) whose indels are projected from child to parent (green) and parent to child (red). (**B**) The sub-MSAs are produced after collecting sequences from a binary guide tree with each node labeled with the name of its longest descendant sequence. Sequences are collected by traversing the tree in a breadth-first fashion. Pale red color blocks indicate how the N parent sequences (N=3) are collected by recursively expanding nodes. The same process is then applied to gather the children (green) and the grandchildren (blue). (**C**) In the nine resulting sub-MSAs that are displayed, one should note the presence of a common representative sequence between each child and its parent.

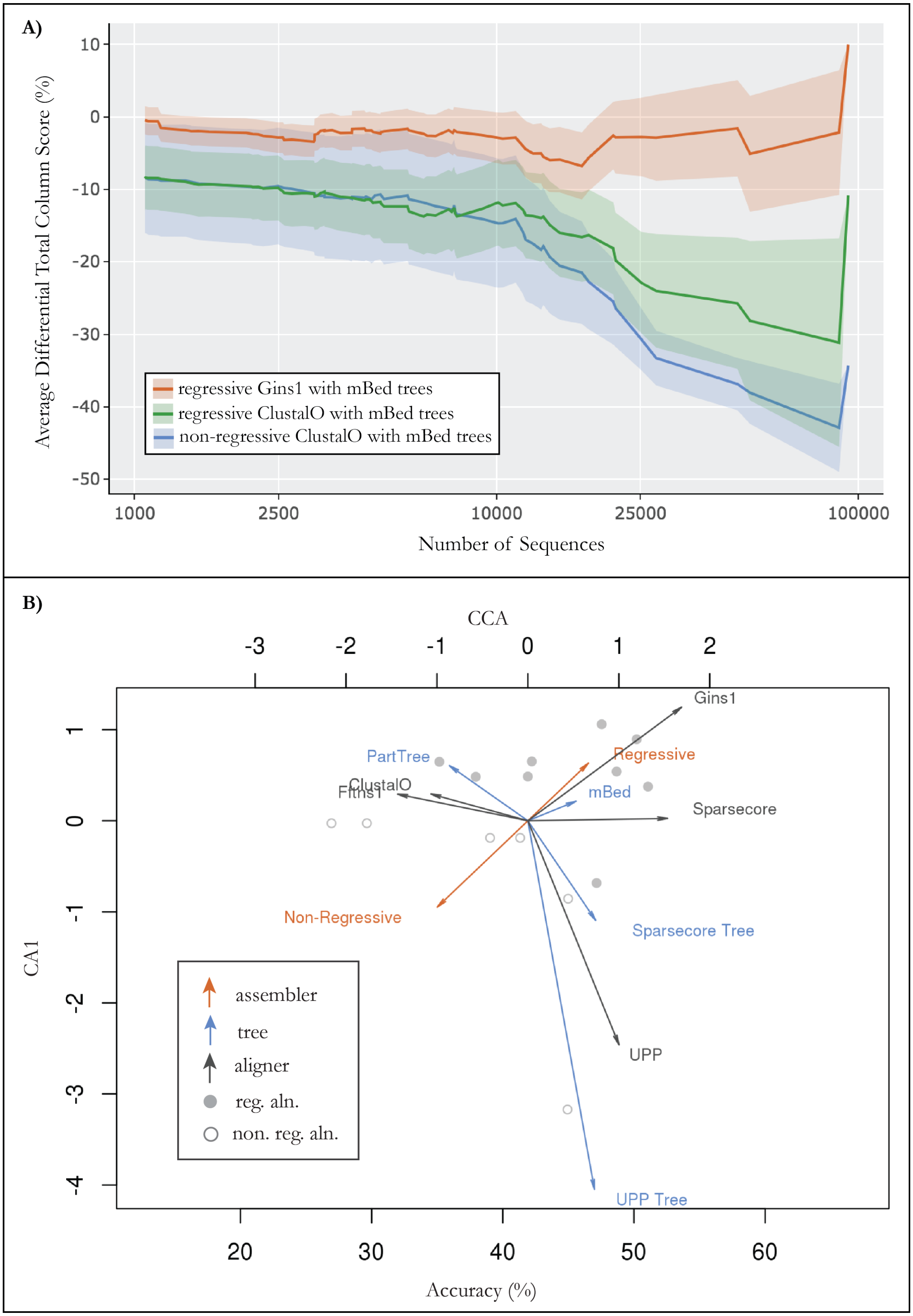
A)
Average Differential Total Column Score (%)
10
0
-10
-20
-30
-40
-50
1000
2500
10000
25000
100000
Number of Sequences
regressive Gins1 with mBed trees
regressive ClustalO with mBed trees
non-regressive ClustalO with mBed trees
B)
CCA
-3
-2
-1
0
1
2
CA1
1
0
-1
-2
-3
-4
20
30
40
50
60
Accuracy (%)
Gins1
PartTree
Regressive
ClustalO
Fftns1
mBed
Sparsecore
Non-Regressive
Sparsecore Tree
UPP
UPP Tree
assembler
tree
aligner
reg. aln.
non. reg. aln.

Fig. 2.

Relative performances of alternative MSA algorithm combinations. (**A**) Average differential accuracy of datasets larger than *Number of Sequences* (horizontal axis). The differences of accuracy are measured between the reference sequence MSAs and their embedded projection in the large datasets. The envelope is the standard deviation. (**B**) In this constrained correspondence analysis (CCA) the first component (horizontal axis, 14.1% of the variance) is constrained to be the total column score accuracy as measured on datasets larger than 10,000. The best unconstrained component (vertical axis) explains 20.8% of the remaining variance. Methods (grey dots with their accuracy on the lower horizontal axis) are categorized by their guide-tree (blue), MSA algorithms (grey) and regressive/non-regressive procedure (red). Vectors indicate the contributions to variance of each category from the three variables. Their projection onto the upper horizontal axis quantifies the contribution to variance of overall accuracy.

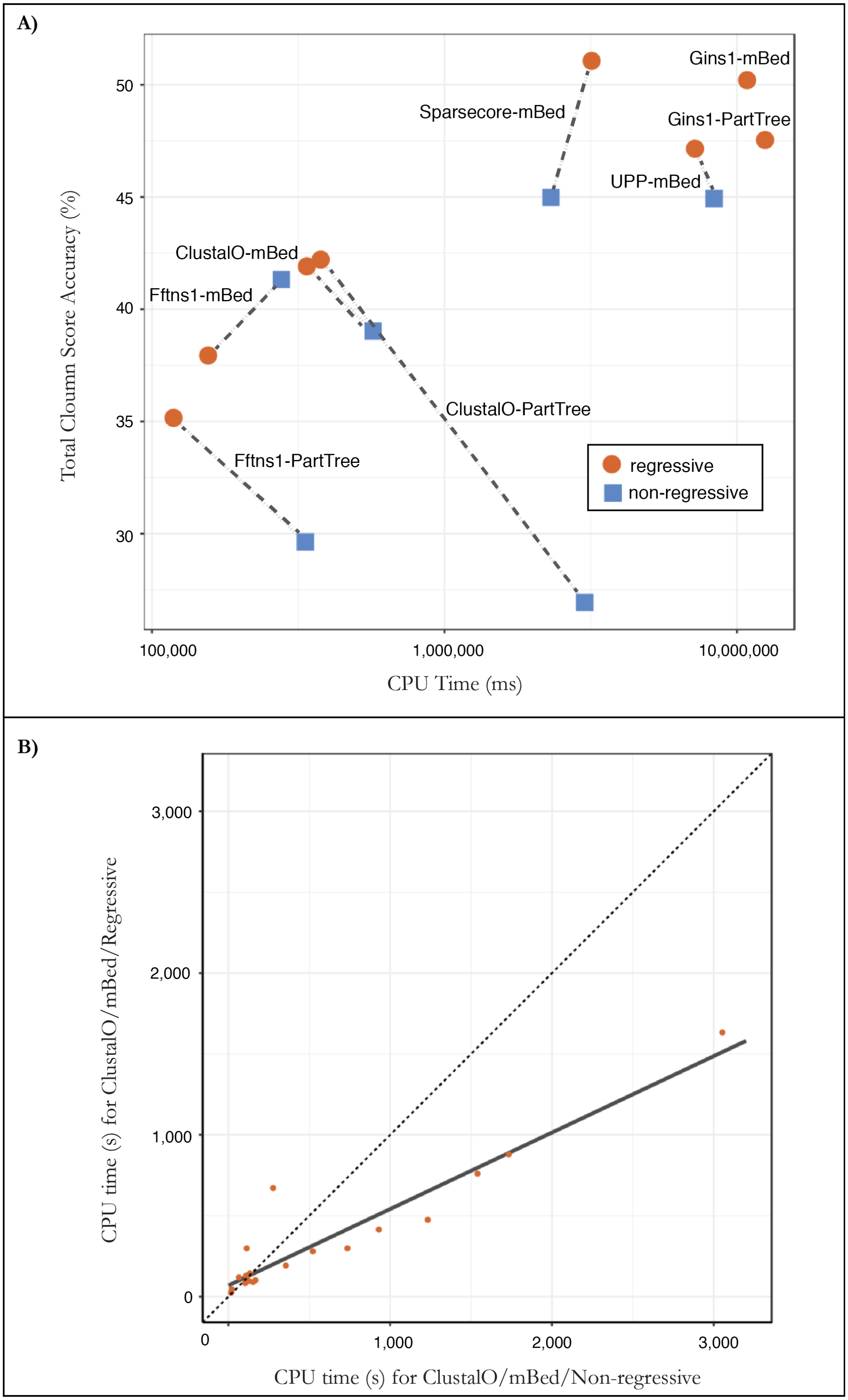
A)
Total Cloumn Score Accuracy (%)
50
45
40
35
30
Sparsecore-mBed
Gins1-mBed
Gins1-PartTree
UPP-mBed
ClustalO-mBed
Fftns1-mBed
ClustalO-PartTree
Fftns1-PartTree
● regressive
■ non-regressive
100,000
1,000,000
10,000,000
CPU Time (ms)
B)
CPU time (s) for ClustalO/mBed/Regressive
3,000
2,000
1,000
0
0
1,000
2,000
3,000
CPU time (s) for ClustalO/mBed/Non-regressive

Fig. 3.

CPU requirements of the regressive algorithm on HomFam datasets containing more than 10,000 sequences. (**A**) The total CPU requirements (horizontal axis) and average total column score accuracies (vertical axis). The corresponding non-regressive (blue square) and regressive (red circles) combinations are connected by a dashed line with the exception of Gins1 for which the non-regressive computation costs are prohibitive. (**B**) Comparison of CPU time requirements for ClustalO using mBed trees using a regressive and a non-regressive procedure on HomFam datasets containing more than 10,000 sequences. A linear regression (grey) was fitted on the resulting graph ($R^2 = 0.89$, p-value $< 6.9*10^{-10}$).

		total column score %			cpu time (s)	
tree method	**MSA algorithm**	**non-regressive**	**regressive**	**reference**	**non-regressive**	**regressive**
PartTree	Fftns1	29.64	35.16	47.84	334	118
mBed	Fftns1	41.33	37.94	52.03	277	155
PartTree	ClustalO	26.94	42.21	50.54	3,016	377
mBed	ClustalO	39.03	41.91	53.71	570	338
average		**34.24**	**39.31**	**51.03**	**1,050**	**247**
default/mBed	UPP	44.93	47.15	49.78	8,353	7,186
default/mBed	Sparsecore	44.98	51.06	53.5	2,3125	3,184
PartTree	Gins1	-	47.54	49.46	-	12,477
mBed	Gins1	-	50.2	53.07	-	10,833

Table 1.

Total Column Score and average CPU time (s) on the 20 HomFam datasets containing over 10,000 sequences.