Published in final edited form as:
Nat Biotechnol. 2019 December ; 37(12): 1466–1470. doi:10.1038/s41587-019-0333-6.

Large multiple sequence alignments with a root-to-leaf regressive method

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

[1]Centre for Genomic Regulation, The Barcelona Institute of Science and Technology, Barcelona, Spain

[2]Universitat Pompeu Fabra (UPF), Barcelona, Spain

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

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

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

Abstract

Multiple sequence alignments (MSAs) are used for structural[1,2] and evolutionary predictions[1,2], but the complexity of aligning large datasets requires the use of approximate solutions[3], including the progressive algorithm[4]. Progressive MSA methods start by aligning the most similar sequences and subsequently incorporate the remaining sequences, from leaf-to-root, based on a guide-tree. Their accuracy declines substantially as the number of sequences is scaled up[5]. We introduce a regressive algorithm that enables MSA of up to 1.4 million sequences on a standard workstation and substantially improves accuracy on datasets larger than 10,000 sequences. Our regressive algorithm works the other way around to the progressive algorithm and begins by aligning the most dissimilar sequences. It uses an efficient divide-and-conquer strategy to run third-party alignment methods in linear time, regardless of their original complexity. Our approach will enable analyses of extremely large genomic datasets such as the recently announced Earth BioGenome Project, which comprises 1.5 million eukaryotic genomes[6].

Until the first benchmarking of large-scale MSAs, analyses made on smaller datasets suggested that scale-up would result in increased accuracy[7]. However, it has now been established that alignments with more than a thousand sequences are less accurate than smaller alignments[5]. It has been speculated[8] that this fall in accuracy is due to the inability

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

Data availability: All data, analyses and results are available from Zenodo (10.5281/zenodo.3271452).

Author contributions: C.N. designed and implemented the algorithm, E.F., E.G., L.M., A.B., and P.D.T designed the validation procedure and carried out the validation. I.E. performed statistical and CCA analyses. E.F., C.N., E.G., C.M., L.M., A.B., P.D.T., I.E., F.K. and H.L. wrote and edited the manuscript.

Competing interests: Authors declare no competing interests.

Europe PMC Funders Author Manuscripts

of progressive methods to deal with the large number of gaps accumulated during intermediate alignment steps[9]. Recent attempts to address this problem have included SATé[10] and its follow-up PASTA[11,12], 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[13] and MAFFT-Sparsecore[14] (Sparsecore). Both of these methods rely on selecting a subset of 'seed' sequences and turning them into a Hidden Markov Model (HMM) using either PASTA or the slower, more accurate version of MAFFT. The HMM is used to incorporate all the remaining sequences one by one. The downside of this approach is that the seed sequences are insufficiently diverse and therefore preclude the accurate alignment of distantly related homologues to the seed HMM.

We hypothesized that a regressive algorithm would address this problem by combining the benefits of a progressive approach when incorporating distant homologues with the improved accuracy of seeded methods. We needed to fulfil two simple constraints: the splitting of the sequences across sub-MSAs each containing a limited number of sequences and their combination into a MSA without the requirement of an alignment procedure. 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 first used to identify N non-overlapping sequence groups of unspecified size - the children. N defines both the maximum number of children at any level and the maximum size of each sub-MSA. It constitutes the only free parameter of the algorithm. The first sub-MSA - the parent - is computed by selecting a representative sequence from each child group and by aligning theses N representatives with an MSA algorithm such as Clustal Omega (ClustalO), MAFFT or any suitable third-party software. The clustering algorithm is then re-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. In each child group, the N new representatives are selected in such a way that the corresponding child MSA has exactly one sequence in common with its parent - the common representative. 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 common 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. Because of the way they are projected during merging, these insertions and their corresponding gap symbols do not need to be allocated in memory. They can be kept as counts and merely expanded while the MSA is written onto disc, thus dramatically decreasing the memory footprint.

A key step of this recursion is the clustering method and the subsequent selection of the N representative sequences. Our benchmarking suggests N=1,000 to be a sensible choice (Supplementary Fig. S1A and S1B). This value is in agreement with a previous report on the

largest number of sequences that can be directly aligned without accuracy loss[5]. 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 taxon sampling procedure[15]. In our implementation (Supplementary Note 1), every node gets labelled with the longest sequence among its descendants. Given a fully labelled 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 be 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[5] 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[16]. We started by benchmarking the ClustalO and MAFFT-FFTNS-1 (Fftns1) MSA algorithms using two guide-tree methods: ClustalO embedded k-means trees[17] (mBed) and MAFFT-PartTree[18] (PartTree). 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 (Supplementary Tables 1 and 2). Within this first set of analyses, 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[14] and UPP[13]. In both cases their accuracy 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 (Supplementary Tables 1 and 2). 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)[19], 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 (Supplementary Tables 1 and 2).

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 on their own (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 (Supplementary Fig. S2). The improved stability of the regressive combination was especially clear when considering Gins1 (Fig. 2A, Supplementary Fig. S2A) 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)[20], 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).

The most counter-intuitive property of the regressive algorithm is its dependency on an initial parent MSA whose level of identity is imposed by the guide-tree. Given an optimal guide tree, the level of identity of this initial parent is expected to be as low as possible. This first step is central to the algorithm's divide-and-conquer strategy, but it is unclear if so much diversity at this early stage would harm accuracy prospects. We addressed this question by using HomFam to generate several alternative parent MSAs with different levels of identity for each dataset (i.e. same MSA algorithm and dataset but different guide-trees). We then computed the final MSA corresponding to each parent and did not find any significant relationship between parent identity and final MSA accuracy (Supplementary Table 3). By contrast, a similar comparison across datasets (i.e. same MSA algorithm and guide-tree method but different dataset) shows a strong positive dependency between parent identity and final MSA accuracy (Supplementary Table 4). This analysis confirms that, when using the regressive algorithm, the choice of very diverse sequences as a starting point does not

incur a penalty whilst, as one would expect, datasets with lower identity result in MSAs with lower accuracy.

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 favourable when considering CPU intensive MSA algorithms like Gins1 for which the non-regressive computation had been impossible.

We further explored the scaling up capacities of our algorithm using 45 Pfam 28.0 families[21] containing between 100,000 and 1.4 million sequences for the largest (ABC transporter family, PF00005). Although they lack a structural reference, these families were selected among the largest entries so as to provide a biologically realistic benchmark for scalability. When using a standard workstation (48 Gb of RAM, 160 CPU hours), the regressive methods were the only ones able to process all 45 datasets while the non-regressive methods tend to fail above 240,000 sequences and can only align a maximum of 500,000 sequences for the most robust (Supplementary Table 5 and 6).

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 (i.e. N=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 final 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[22].

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[23].

Online Methods

Reference Datasets

The HomFam dataset was downloaded from the Clustal Omega web site (http://www.clustal.org/omega/homfam-20110613-25.tar.gz). It features 94 families which contain homologous sequences extracted from Pfam 25. Each dataset is associated with a smaller set of reference sequences for which a structure-based alignment is available. For each family, the large-scale datasets are produced by merging the reference and the homologous sequences into a single file. The very large datasets were assembled by selecting 45 Pfam families whose sizes range between 100.000 and 1.4 million sequences. Summary statistics of the very large datasets are provided in Supplementary Table 7.

Multiple Alignments and Guide-Trees

The regressive algorithm is implemented in T-Coffee (hash cd5090c in the GitHub repository) and uses third party methods for guide-tree and sub-MSA computation. The sub-MSAs were produced using Clustal Omega (Version 1.2.4), UPP (Version 4.3.4) and MAFFT (Version 7.397) for Gins1, Fftns1 and Sparsecore. The mBed and PartTree guide-trees were estimated using the `--guidetree-out option` of Clustal Omega and the `-parttree` option of MAFFT. Random trees were generated by shuffling the taxa on the original mBed trees (`+newick_randomize` option in T-Coffee/seq_reformat). Parent MSAs were collected using a specific T-Coffee flag triggering their output as intermediate files (`DUMP_ALN_BUCKET=1`).

Benchmarking

Benchmarking was carried out by aligning either the reference or the large-scale datasets, and by comparing the projection of the reference sequences with their reference alignment using the `aln_compare` option of the T-Coffee package. This option supports the Sum-of-Pairs (fraction of pairs of residue in the reference alignment found in the benchmark) and the Total Column score (fraction of columns in the reference alignment found in the benchmark) metrics[24].

Constrained Correspondence Analysis

Each alignment procedure (*e.g.* regressive ClustalO using mBed tTrees) is represented in the form of a string of zeros and ones encoding its categorical variables (guide-tree method, aligner, assembler). Within this string, each variable is encoded in a substring whose length is equal to the number of levels (*e.g.* number of alternative guide-tree methods). These substrings therefore contain a single 1 entry so that the entire string sums to the number of variables for any given procedure (*i.e.* 3 in our case). Once encoded this way alignment procedures become the rows of an indicator matrix that can be analyzed with dimensionality reduction techniques such as multiple correspondence analysis. In Constrained (a.k.a.

Canonical) Correspondence Analysis (CCA) dimensionality reduction is guided by additional information about each observation. In our case, this information is the accuracy (Total Column score) of each alignment procedure averaged across the 20 datasets containing over 10,000 sequences. The application of CCA involves projecting the indicator matrix onto a linear space defined by the accuracy vector[20]. The technique makes it possible to then perform a singular value decomposition and displayed in the form of a biplot as in Figure 2B. Calculations were carried out using the R package Vegan (https://cran.r-project.org/package=vegan). Percent variance explained is obtained by dividing the eigenvalue of the respective axis with the sum of all the eigenvalues, multiplied by 100.

Relationship between parent MSA identity and accuracy

The 75 HomFam datasets containing more than 1,000 sequences were regressively aligned using 3 guide-tree methods (mBed, PartTree and randomized mBed) along with 4 MSA algorithms (ClustalO, Fftns1, UPP and Sparsecore). For a given dataset the use of different guide trees usually results in different parent MSAs. We therefore collected all 900 pairs of combinations involving the same dataset, the same MSA algorithms and two different guide trees. Results were compiled in a contingency table counting increase or decrease of the parent MSA percent identity as well as increase or decrease of the MSA accuracy (as measured by total column score). A two-sided Fisher test (implemented in R) was used to test the Null hypothesis of no association (i.e., odds ratio 1). The ratio of the odds of increasing accuracy vs decreasing accuracy was 0.85 times higher for the cases where identity increased than for the cases where identity decreased ($p<0.29$). In order to do a comparable analysis across different datasets, we collected all the 33,000 pairs of combinations involving a different dataset, the same MSA algorithms and the same guide trees. Here, the odds of increasing vs decreasing accuracy were 4.1 times higher in the cases where identity increased than in the cases where identity decreased ($p<10^{-15}$).

Computation

All computation was carried out on a cluster running Scientific Linux release 7.2 with all guide-trees, alignments and evaluations carried out within a container based on the Debian (Jessie) operating system. The computational pipeline (*c.f.* Code Availability) was implemented in the Nextflow language[25] and was deployed in a containerized form using Singularity. Computation was limited to 48 Gbytes of memory and 160 CPU hours. Given a HomFam family, this pipeline generates the mBed and PartTree guide-trees for both the reference and the large-scale dataset. It then combines the selected aligners (ClustalO, Gins1, Fftns1, Sparsecore and UPP) and the precomputed guide-trees to generate (i) a default alignment of the reference sequences (ii) a default (*i.e.* non-regressive) alignment of the large-scale dataset and (iii) a regressive alignment of the large-scale dataset. Note that UPP and Sparsecore do not support external guide-trees and that their default alignments were therefore produced using the default guide-tree procedures of these methods. A Docker

Code availability: The regressive alignment algorithm has been implemented in T-Coffee and is available at the T-Coffee website (http://www.tcoffee.org) and on GitHub (https://github.com/cbcrg/tcoffee). A GitHub repository containing the Nextflow workflow[25] and Jupyter notebooks[26] to replicate the analysis are available at https://github.com/cbcrg/dpa-analysis (release v1.2).

image has been created which contains all the pipeline dependencies. It is available from DockerHub (https://hub.docker.com) and is available via the command:

```
docker pull cbcrg/regressive-msa:@ cd5090c
```

The Dockerfile is also provided in the Git repository to allow for reuse and addition of new tools. All command lines used by the pipeline are also provided in the dedicated supplementary section (Supplementary Note 2).

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgments

We thank G. Riddihough for revisions and comments on the manuscript and O. Gascuel for suggestions.

Funding: This project was supported by the Centre for Genomic Regulation, the Spanish Plan Nacional, the Spanish Ministry of Economy and Competitiveness, 'Centro de Excelencia Severo Ochoa' (E.G., P.T., C.M., I.E., L.M., A.B., F.K., E.F., C.N.) and an ERC Consolidator Grant from the European Commission, Grant Agreement # 771209 ChrFL (F.K.).

References

1. Uguzzoni G, et al. Large-scale identification of coevolution signals across homo-oligomeric protein interfaces by direct coupling analysis. Proc Natl Acad Sci U S A. 2017; 114:E2662–E2671. [PubMed: 28289198]
2. Mirarab S, Bayzid MS, Boussau B, Warnow T. Statistical binning enables an accurate coalescent-based estimation of the avian tree. Science. 2014; 346:1250463. [PubMed: 25504728]
3. Wang L, Jiang T. On the complexity of multiple sequence alignment. J Comput Biol. 1994; 1:337–348. [PubMed: 8790475]
4. Hogeweg P, Hesper B. The alignment of sets of sequences and the construction of phylogenetic trees. An integrated method. J Mol Evol. 1984; 20:175–186. [PubMed: 6433036]
5. Sievers F, et al. Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. Mol Syst Biol. 2011; 7:539. [PubMed: 21988835]
6. Lewin HA, et al. Earth BioGenome Project: Sequencing life for the future of life. Proc Natl Acad Sci U S A. 2018; 115:4325–4333. [PubMed: 29686065]
7. Katoh K, Misawa K, Kuma K, Miyata T. MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. Nucleic Acids Res. 2002; 30:3059–66. [PubMed: 12136088]
8. Chatzou M, et al. Multiple sequence alignment modeling: methods and applications. Brief Bioinform. 2015; doi: 10.1093/bib/bbv099
9. Breen MS, Kemena C, Vlasov PK, Notredame C, Kondrashov FA. Epistasis as the primary factor in molecular evolution. Nature. 2012; 490:535–538. [PubMed: 23064225]
10. Liu K, Raghavan S, Nelesen S, Linder CR, Warnow T. Rapid and accurate large-scale coestimation of sequence alignments and phylogenetic trees. Science. 2009; 324:1561–1564. [PubMed: 19541996]
11. Mirarab S, et al. PASTA: Ultra-Large Multiple Sequence Alignment for Nucleotide and Amino-Acid Sequences. J Comput Biol. 2015; 22:377–386. [PubMed: 25549288]
12. Collins K, Warnow T. PASTA for proteins. Bioinformatics. 2018; 34:3939–3941. [PubMed: 29931282]
13. Nguyen N-PD, Mirarab S, Kumar K, Warnow T. Ultra-large alignments using phylogeny-aware profiles. Genome Biol. 2015; 16:124. [PubMed: 26076734]

14. Yamada KD, Tomii K, Katoh K. Application of the MAFFT sequence alignment program to large data-reexamination of the usefulness of chained guide trees. Bioinformatics. 2016; 32:3246–3251. [PubMed: 27378296]
15. Minh BQ, Klaere S, von Haeseler A. Phylogenetic diversity within seconds. Syst Biol. 2006; 55:769–773. [PubMed: 17060198]
16. Stebbings LA, Mizuguchi K. HOMSTRAD: recent developments of the Homologous Protein Structure Alignment Database. Nucleic Acids Res. 2004; 32:D203–7. [PubMed: 14681395]
17. Blackshields G, Sievers F, Shi W, Wilm A, Higgins DG. Sequence embedding for fast construction of guide trees for multiple sequence alignment. Algorithms Mol Biol. 2010; 5:21. [PubMed: 20470396]
18. Katoh K, Toh H. PartTree: an algorithm to build an approximate tree from a large number of unaligned sequences. Bioinformatics. 2007; 23:372–374. [PubMed: 17118958]
19. Katoh K, Kuma K-I, Toh H, Miyata T. MAFFT version 5: improvement in accuracy of multiple sequence alignment. Nucleic Acids Res. 2005; 33:511–518. [PubMed: 15661851]
20. Greenacre, MJ. Biplots in Practice. Fundacion BBVA; 2010.
21. Finn RD, et al. Pfam: the protein families database. Nucleic Acids Res. 2013; 42:D222–D230. [PubMed: 24288371]
22. Herlihy, M, Shavit, N. The Art of Multiprocessor Programming, revised first edition. Morgan Kaufmann; 2012.
23. Kahn SD. On the future of genomic data. Science. 2011; 331:728–729. [PubMed: 21311016]
24. Thompson JD, Plewniak F, Poch O. BAliBASE: a benchmark alignment database for the evaluation of multiple alignment programs. Bioinformatics. 1999; 15:87–88. [PubMed: 10068696]
25. Di Tommaso P, et al. Nextflow enables reproducible computational workflows. Nat Biotechnol. 2017; 35:316–319. [PubMed: 28398311]
26. Perkel JM. Why Jupyter is data scientists' computational notebook of choice. Nature. 2018; 563:145–146. [PubMed: 30375502]

Europe PMC Funders Author Manuscripts

Europe PMC Funders Author Manuscripts

Editors summary

Aligning 1.4 million sequences is now possible with T-Coffee's new regressive algorithm.

Europe PMC Funders Author Manuscripts

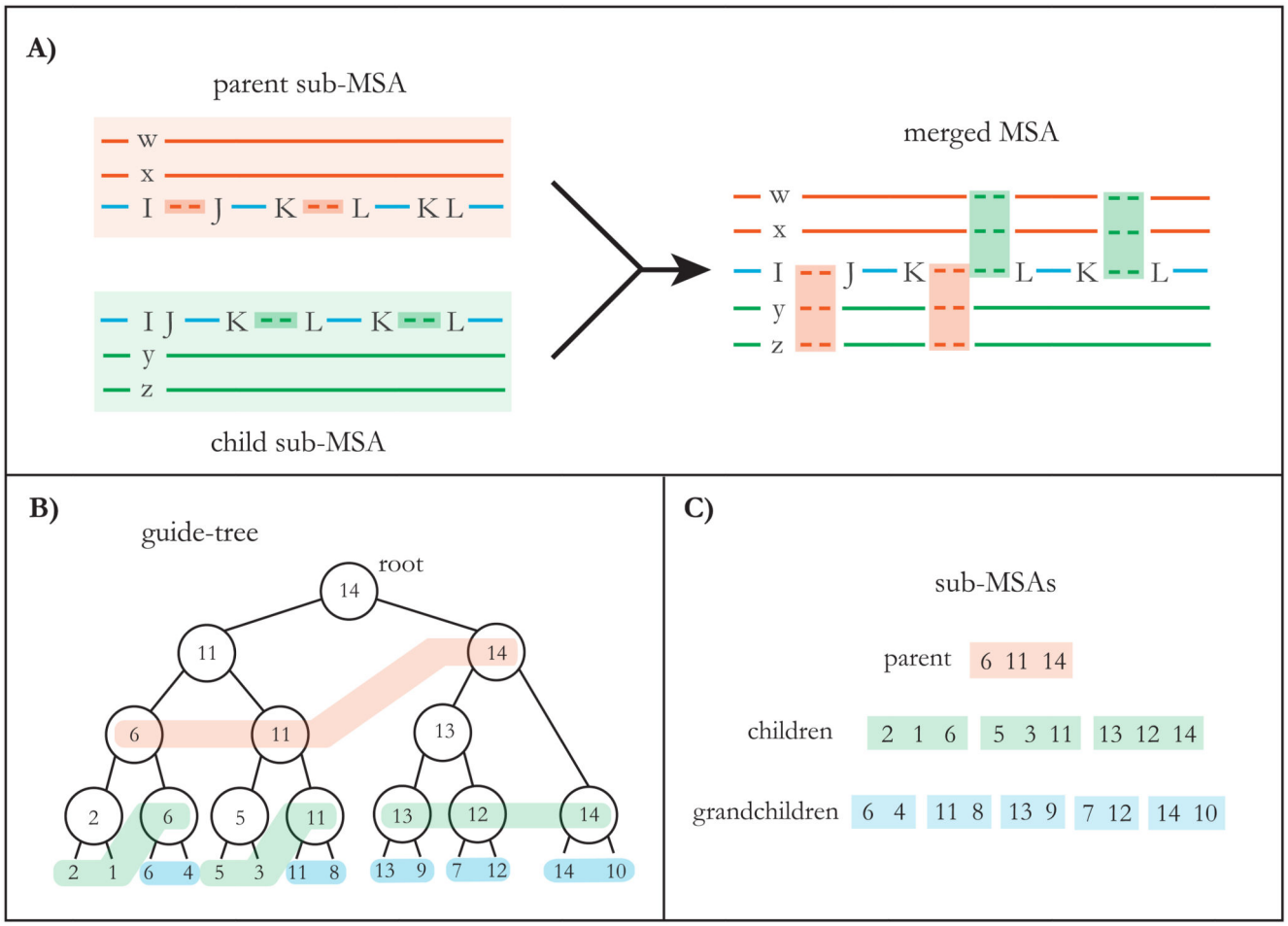

Figure 1. Regressive algorithm overview

(**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 labelled with the name of its longest descendant sequence. Sequences are collected by traversing the tree in a breadth-first fashion. Pale red colour 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.

Europe PMC Funders Author Manuscripts

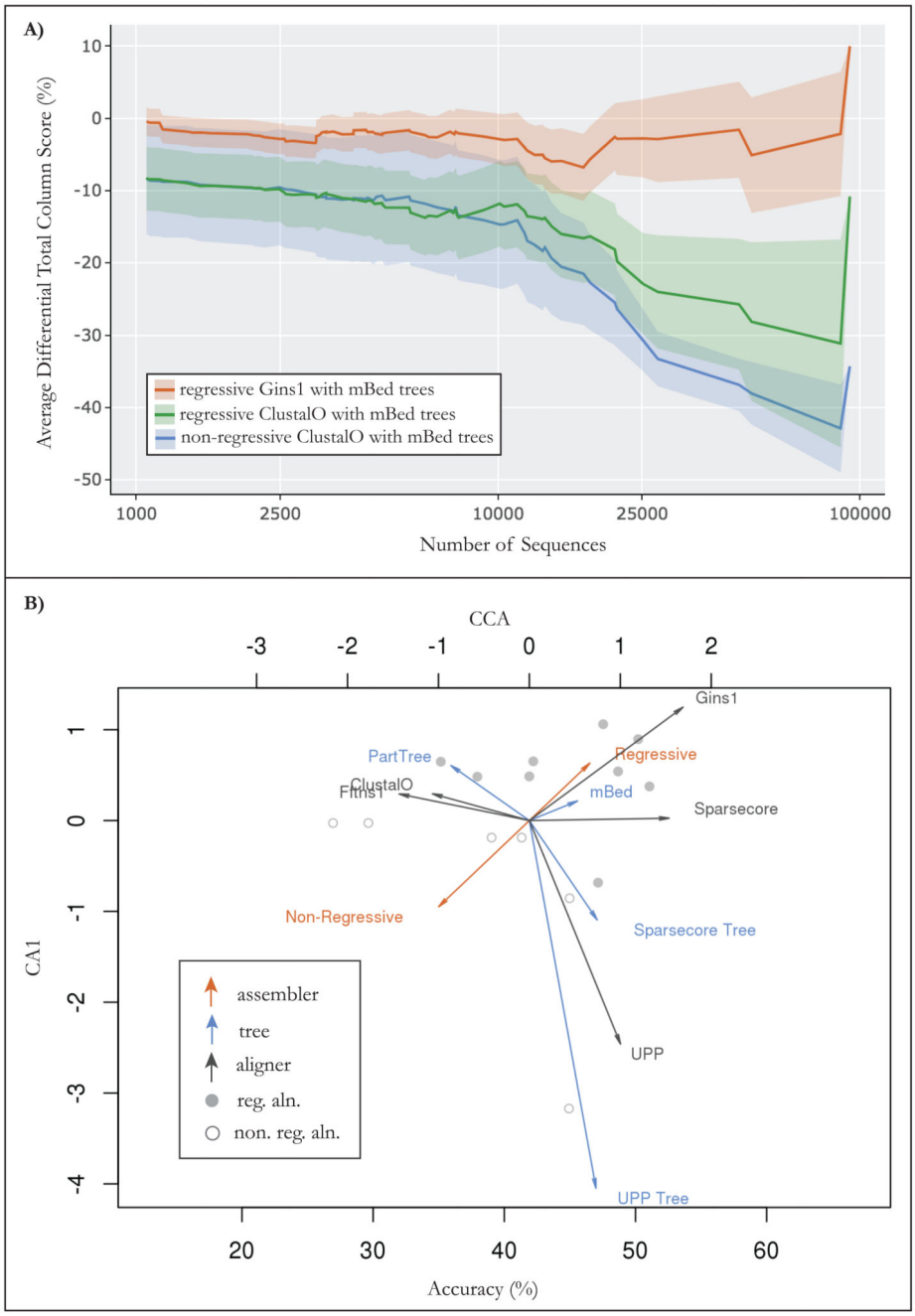

Figure 2. Relative performances of alternative MSA algorithm combinations.

(**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. For each combination, n=75 independent MSA samples. 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. Combinations (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. For each combination, represented as a dot, n=20 independent MSA samples.

Europe PMC Funders Author Manuscripts

Europe PMC Funders Author Manuscripts

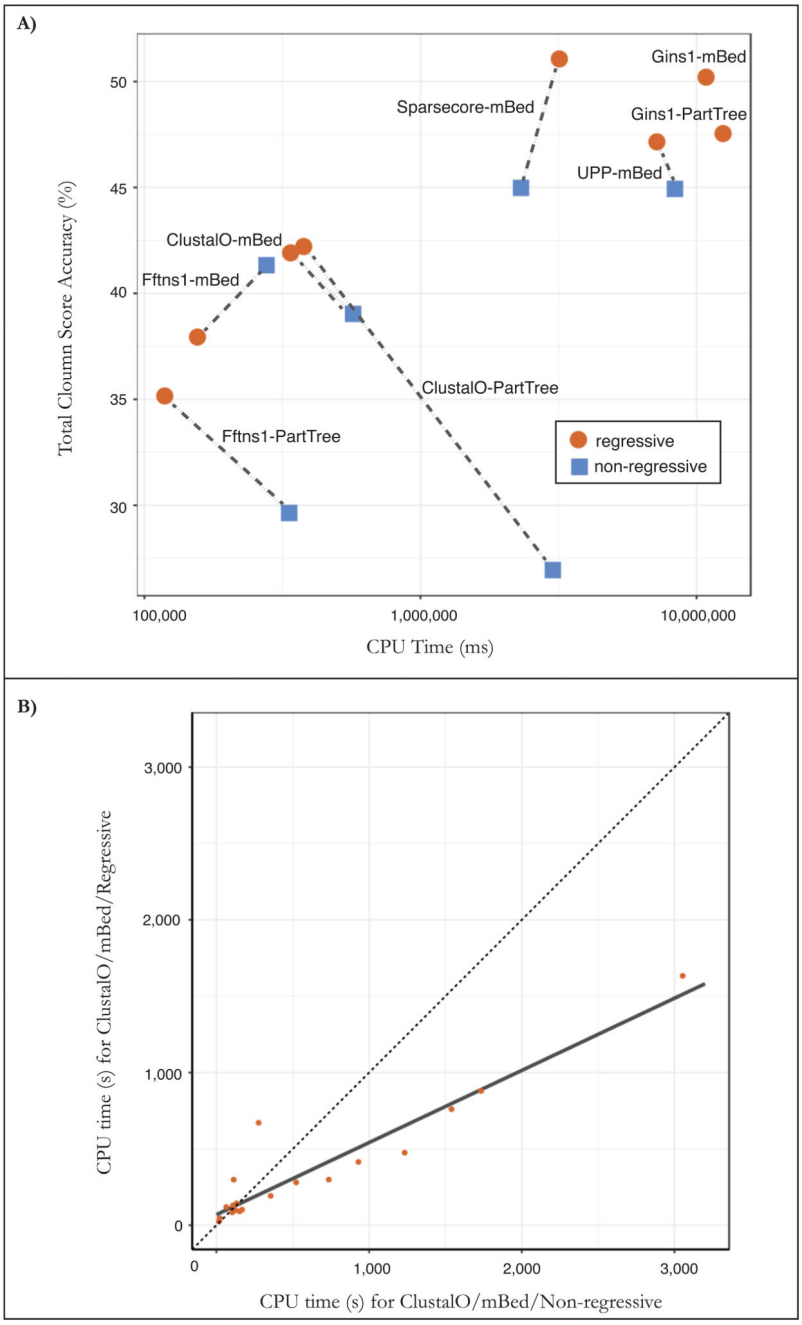

Figure 3. CPU requirements of the regressive algorithm on HomFam datasets containing more than 10,000 sequences.

(**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. For each combination, represented as a circles and squares, n=20 independent MSA samples. (**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. Each point represents an independent MSA. n=20 independent MSA samples. A linear regression (grey) was fitted on the resulting graph (R^2 = 0.89, p-value = $6.9*10^{-10}$).

Europe PMC Funders Author Manuscripts

Europe PMC Funders Author Manuscripts

Table 1

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

			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	156
PartTree	ClustalO		26.94	42.21	50.54		3,017	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		8,354	7,186
default/mBed	Sparsecore		44.98	51.06	53.50		2,313	3,184
PartTree	Gins1		-	47.54	49.46		-	12,478
mBed	Gins1		-	50.20	53.07		-	10,834