BIOINFORMATICS APPLICATIONS NOTE Vol. 23 no. 3 2007, pages 372–374
doi:10.1093/bioinformatics/btl592

Sequence analysis

PartTree: an algorithm to build an approximate tree from a large number of unaligned sequences

Kazutaka Katoh[1,*] and Hiroyuki Toh[2]

[1]Digital Medicine Initiative, Kyushu University, Fukuoka 812-8582, Japan and [2]Medical Institute of Bioregulation, Kyushu University, Fukuoka 812-8582, Japan

Received on August 23, 2006; revised on October 30, 2006; accepted on November 17, 2006
Advance Access publication November 21, 2006
Associate Editor: Thomas Lengauer

ABSTRACT

Motivation: To construct a multiple sequence alignment (MSA) of a large number (>∼10 000) of sequences, the calculation of a guide tree with a complexity of $O(N^2)$ to $O(N^3)$, where N is the number of sequences, is the most time-consuming process.

Results: To overcome this limitation, we have developed an approximate algorithm, PartTree, to construct a guide tree with an average time complexity of $O(N \log N)$. The new MSA method with the PartTree algorithm can align ∼60 000 sequences in several minutes on a standard desktop computer. The loss of accuracy in MSA caused by this approximation was estimated to be several percent in benchmark tests using Pfam.

Availability: The present algorithm has been implemented in the MAFFT sequence alignment package (http://align.bmr.kyushu-u.ac.jp/mafft/software/).

Contact: katoh@bioreg.kyushu-u.ac.jp

Supplementary information: Supplementary information is available at *Bioinformatics* online.

1 INTRODUCTION

Most multiple sequence alignment (MSA) programs use a guide tree. An MSA is computed along with the tree using a group-to-group alignment algorithm. When a large number of sequences are aligned, the construction of guide tree is the time- and space-limiting process. A distance matrix is usually calculated before tree building and it requires an $O(N^2)$ memory space, where N is the number of sequences. As for time complexity, MAFFT (Katoh *et al*., 2002, 2005) uses an $O(N^3)$ algorithm for constructing a variant of UPGMA guide tree. MUSCLE (Edgar, 2004a,b) uses a more efficient $O(N^2)$ algorithm. In a context where a large number of sequences are being routinely determined, the scalability of MSA methods is getting important. For instance, a Pfam (Finn *et al*., 2006) alignment of ABC transporter consists of ∼30 000 sequences and Ribosomal Database Project II release 9 (Cole *et al*., 2005) contains over 200 000 SSU rRNA sequences. Here we describe a simple divisive clustering algorithm, PartTree, to construct a rough tree from a set of a large number (more than ∼10 000) of unaligned sequences, with an average time complexity of $O(N \log N)$ and a space complexity of $O(N)$.

2 ALGORITHM

Let $N_{i,j}$ represent the number of sequences belonging to group j at recursive depth i ($i \geq 1$). At the initial cycle ($i = 1$), $j = 1$ and $N_{1,1} = N$. Otherwise ($i > 1$), $1 \leq j$ and $\sum_j N_{i,j} = N$. The sequences are classified into n groups at each cycle, where n is a parameter given by user.

(1) The longest sequence among the $N_{i,j}$ sequences is selected.

(2) The similarities between the longest sequence and the remaining $N_{i,j} - 1$ sequences are calculated.

(3) From the $N_{i,j}$ sequences, n sequences are picked up as ‘seeds’. They include (a) the longest sequence, (b) the sequence with the lowest similarity and (c) randomly selected $n - 2$ sequences.

(4) The similarities among the n seeds are computed. If two seeds are highly similar to each other, shorter one is excluded. The number of the remaining seeds is denoted as n'.

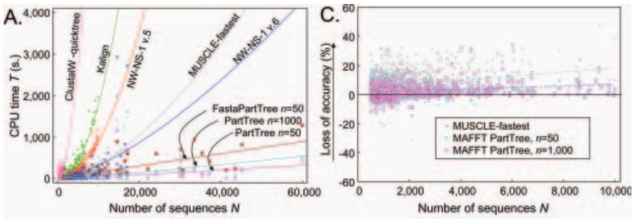
(5) An UPGMA tree is built among the n' sequences. If $n' \geq N_{i,j}$, then the tree is returned to the parent cycle and no further child cycle is carried out.

(6) The similarities between the n' seeds and the remaining $N - n'$ sequences are calculated. Each of the remaining sequences is classified into either of n' groups, according to the similarity. The number of sequences in group j is denoted as $N_{i+1,j}$, and each group is subjected to the child cycle with depth $i + 1$.

(7) The subtrees returned from the n' child cycles are combined into a single new tree along with the UPGMA tree calculated in Step 5. The new tree is returned to the parent cycle.

The number of sequences belonging to group j at depth i is estimated as $N_{i,j} \sim N_{i-1,j}/n \sim N/n^i$ on average, and the cycle is recursively repeated until $N/n^I < n$, where I is the maximum depth. Thus I is proportional to log N on average. At depth i, $O(N)$ sequence comparisons are performed. The overall number of sequence comparisons is therefore proportional to N log N. The time complexity of the entire procedure depends on that for computing the similarities at Steps 2, 4 and 6. This algorithm does not require a standard distance matrix with N^2 elements. Instead, a partial distance matrix, with $nN_{i,j}$

[*]To whom correspondence should be addressed.



B.

	BAliBASE	Pfam	Pfam	
	N=4-142	*N*=500-	*N*=1,001-	
Method		1,000	10,000	Command
One-round progressive methods				
NW-NS-1 v5	78.6**	77.6**	68.5**	
NW-NS-1 v6	79.8**	80.0**	76.5**	`mafft --retree 1 --algq`
PartTree *n*=50	79.7**	77.8**	73.4**	`mafft --retree 1 --parttree --partsize 50`
PartTree *n*=1,000	(79.7**)	(79.1**)	74.1**	`mafft --retree 1 --parttree --partsize 1000`
FastaPartTree *n*=50	82.0	79.9**	76.4**	`mafft --retree 1 --fastaparttree --partsize 50`
MUSCLE-fastest	77.8**	75.9**	68.1**	`muscle -maxiters 1 -diags -sv -distance1 kbit20_3`
Kalign	**82.7**	80.7**		`kalign`
ClustalW-quicktree	74.5**			`clustalw -quicktree`
Two-round progressive methods				
NW-NS-2 v6	81.7	**81.2**	**78.1**	`mafft --retree 2 --algq`
PartTree-2 *n*=50	81.2*	79.2**	74.9**	`mafft --retree 2 --parttree --partsize 50`
PartTree-2 *n*=1,000	(81.2*)	(79.7**)	75.5**	`mafft --retree 2 --parttree --partsize 1000`
FastaPartTree-2 *n*=50	82.2	80.1**	76.6**	`mafft --retree 2 --fastaparttree --partsize 50`
MUSCLE-progressive2	80.7*	77.9**	69.6**	`muscle -maxiters 2`
# of alignments	218	598	567	

Fig. 1. Speed (**A**) and accuracy (**B** and **C**) comparisons of the MAFFT-PartTree and existing progressive methods. In (A), CPU time T required by each method is plotted as a function of the number of sequences N. The relationship was fitted with a regression curve $T = aN^b$. All methods were run on the RedHat Enterprise Linux WS4 on a dual 3.6 GHz Xeon with 4 GB of RAM. (B) Shows the average overlap scores (%) of each method. The symbols '$*$' and '$**$' represent significantly worse results than the method with the highest score at the 0.05 and 0.01 levels, respectively, by the Wilcoxon test. PartTree is virtually equivalent to UPGMA when $N < n$ (shown in parentheses). (C) Shows the differences in accuracy score on individual alignments from NW-NS-1 to each of MUSCLE-fastest and PartTree ($n = 50$ and $n = 1000$) as a function of the number of sequences.

elements, is used at each cycle and is freed before calling the child cycles.

3 APPLICATION

The aforementioned algorithm has been implemented as the PartTree option of an MSA package MAFFT 6.0. See Figure 1B for the command-line usage. In Steps 2, 4 and 6, we use a rapid method to compute a similarity based on the number of shared 6mers (Higgins and Sharp, 1988; Jones *et al*., 1992; Katoh *et al*., 2002), with a length-dependent correction introduced in MAFFT v6 (see the MAFFT page for details). This algorithm requires $O(L)$ steps at every comparison. Thus, the time complexity of the overall procedure is $O(LN \log N)$. We can use more accurate but time-consuming distance measures, such as FASTA (Pearson and Lipman, 1988), instead of the 6mer distance. This strategy is also implemented in MAFFT, as the FastaPartTree option, which requires FASTA v3.4 installed.

Two-round progressive technique (Katoh *et al*., 2002) can be combined with the present method, in which the guide tree is re-calculated from the initial MSA using PartTree and then an MSA is re-constructed. This method is referred to as PartTree-2.

The performances of the present methods were evaluated using a part (1197 entries) of the Pfam 20.0 database (Finn *et al*., 2006) and the full-length set of BAliBASE (Thompson *et al*., 2005). The following progressive MSA programs were compared (see Fig. 1B for detailed list): MUSCLE 3.6 (Edgar, 2004a,b), ClustalW 1.83 (Thompson *et al*., 1994), Kalign 2.01 (Lassmann and Sonnhammer, 2005), and MAFFT v5 and v6. MAFFT v5 uses a UPGMA algorithm with a time complexity of $O(N^3)$, whereas v6 adopted a faster UPGMA algorithm proposed in MUSCLE (Edgar, 2004b). See the mafft page for other differences between v5 and v6. Slower methods were applied to only smaller subsets (with 500–1000 or 500–10 000 sequences) of Pfam. See Figure 1A for the comparison of CPU time. Two-round methods are not shown in Figure 1A but approximately two times slower than the corresponding one-round methods.

Assuming all the Pfam alignments are correct, the accuracy of MSA methods were evaluated with overlap score (Lassmann and Sonnhammer, 2005) between a Pfam alignment and the result of each MSA method (Fig. 1B). The loss of accuracy of an alignment by introducing the present approximation gradually increases with N and was estimated to be $\sim$3% when $N \sim 10\,000$ and $n = 50$ (Fig. 1C). Note that all the progressive methods shown here are much less accurate than more elaborate methods, such as TCoffee [84.6 for overall BAliBASE; Notredame *et al*. (2000)], ProbCons [86.5; Do *et al*. (2005)] and MAFFT-L-INS-i (87.1). As for the topology of guide tree, the loss of accuracy from rigorous UPGMA was estimated to be 10% when $N \sim 2000$ and $n = 50$. See the Supplemental information for detailed discussion on the accuracy of tree topology.

The FastaPartTree option slightly improves the alignment accuracy in comparison with PartTree with 6mer distance, as shown in Figure 1B, because of more accurate guide tree. The Wu–Manber algorithm used in Kalign (Lassmann and Sonnhammer, 2005) might be worth considering as another distance measure. The two-round progressive method is also a practical solution to improve the accuracy of guide tree and alignment, at the cost of roughly doubled CPU time.

ACKNOWLEDGEMENTS

The authors thank Robert C. Edgar for comments on his $O(N^2)$ UPGMA algorighm. The authors also thank Ken Jones, Shiraz Shah and Wataru Nemoto for providing comments and examples. This work was supported by Grant-in-Aid for 'Comparative Genomics' from MEXT of Japan and by JST-PDBj. Funding to pay the Open Access publication charges for this article was provided by Digital Medicine Initiative, Kyushu University.

Conflict of Interest: none declared.

REFERENCES

Cole,J.R. *et al*. (2005) The ribosomal database project (RDP-II): sequences and tools for high-throughput rRNA analysis. *Nucleic Acids Res*., **33**, D294–D296.

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

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

Edgar,R.C. (2004b) MUSCLE: a multiple sequence alignment method with reduced time and space complexity. *BMC Bioinformatics*, **5**, 113.

Finn,R.D. *et al*. (2006) Pfam: clans, web tools and services. *Nucleic Acids Res*., **34**, D247–D251.

Higgins,D.G. and Sharp,P.M. (1988) CLUSTAL: a package for performing multiple sequence alignment on a microcomputer. *Gene*, **73**, 237–244.

Jones,D.T. *et al*. (1992) The rapid generation of mutation data matrices from protein sequences. *Comput. Appl. Biosci*., **8**, 275–282.

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

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

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

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

Pearson,W.R. and Lipman,D.J. (1988) Improved tools for biological sequence comparison. *Proc. Natl Acad. Sci. USA*, **85**, 2444–2448.

Thompson,J.D. *et al*. (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res*., **22**, 4673–4680.

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