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

Kazutaka Katoh\textsuperscript{a}\textsuperscript{*} and Hiroyuki Toh\textsuperscript{b}

\textsuperscript{a}Digital Medicine Initiative, Kyushu University, Fukuoka 812-8582, Japan and \textsuperscript{b}Medical Institute of Bioregulation, Kyushu University, Fukuoka 812-8582, Japan

Associate Editor: Thomas Lengauer

\textbf{ABSTRACT}

\textbf{Motivation:} To construct a multiple sequence alignment (MSA) of a large number (\(\geq 10,000\)) 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.

\textbf{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 several percent in benchmark tests using Pfam.

\textbf{Availability:} The present algorithm has been implemented in the MAFFT sequence alignment package, http://www.biophys.kyoto-u.ac.jp/~katoh/programs/align/mafft/

\textbf{Contact:} katoh@bioreg.kyushu-u.ac.jp

\textbf{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 about 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 \(\sim 10,000\)) of unaligned sequences, with an average time complexity of \(O(N \log N)\) and a space complexity of \(O(N)\).

\textbf{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 \(N_{1,j} = N\). Otherwise (\(i > 1\)), \(1 \leq j\) and \(\sum_{1}^{j} N_{i,j} = N\). 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: (1) the longest sequence, (2) the sequence with the lowest similarity, (3) 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. A 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 \(N_{i,j} \sim N_{i+1,j}/n \sim N/n'\) on average, and the cycle is recursively repeated until \(N/n' < n\), 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 \(nN_{i,j}\) elements, is used at each cycle and is freed before calling the child cycles.

\textbf{3 APPLICATION}

The aforementioned algorithm has been implemented as the PartTree option of an MSA package MAFFT 6.0. See Fig. 1A 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 & 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
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 $T = a N^b$. All methods were run on the RedHat Enterprise Linux WS4 on a dual 3.6GHz Xeon with 4GB 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.

Assuming all the Pfam alignments are correct, the accuracy of MSA methods were evaluated with overlap score (Lassmann & 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 approximately 3% when $N = 10,000$ and $n = 50$ (Fig. 1C). Note that all the progressive methods shown here are much less accurate than 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 approximately 10% when $N = 2,000$ 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 Fig. 1B, because of more accurate guide tree. The Wu-Manber algorithm used in Kalign (Lassmann & 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 rougher doubled CPU time.

ACKNOWLEDGEMENT

The authors thank Robert C. Edgar for comments on his $O(N^2)$ UPGMA algorithm. 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.

REFERENCES


