An Efficient Progressive Alignment Algorithm for Multiple Sequence Alignment

P.V.Lakshmi [†], Allam Appa Rao [† †], GR Sridhar [† † †]
[†] Gitam Institute of Technology, GITAM University, Andhra Pradesh, India
[† †] Jawaharlal Nehru Technological University, Andhra Pradesh, India
[† † †] Endocrine and Diabetes Centre, Andhra Pradesh, India

Summary
Analyzing and comparing the string representations of sequences reveals a lot of useful information about the sequences. As new biological sequences are being generated at exponential rates, sequence comparison is becoming increasingly important to draw functional and evolutionary inference of proteins. This paper presents a partitioning approach for biomolecular sequence alignment that significantly improves the solution time and quality of the problem. The algorithm solves the multiple sequence alignment in three stages. First, an automated and suboptimal partitioning strategy is used to divide the set of sequences into several subsections. Then a multiple sequence alignment algorithm based on progressive method is used to align the sequences of each subsection. Finally, the alignment of original sequences can be obtained by assembling the result of each subsection. Test was conducted on two sets of sequences, namely BChE sequences of mammals and bacteria. Experimental results show that the algorithm can significantly reduce the running time and improve the solution quality of multiple sequence alignment

***Keywords*:** bioinformatics, multiple sequence alignment, partitioning, progressive alignment algorithm, biomolecular sequences.

1. Introduction

Alignment of nucleotide or protein sequences is a fundamental process in molecular biology. Biomolecules, such as deoxyribonucleic acid (DNA), ribonucleic acid (RNA), and protein sequences, which are of interest to computational biologist are made out of many small units of nucleotides molecules strings. The string representation of biomolecules allows for a wide range of algorithmic techniques concerned with strings to be applied for analyzing and comparing biological data. Aligning the string representations of biomolecules can reveal a lot of useful information about the biomolecules. Pairwise alignment attempts to compute the similarity between two sequences to determine maximum alignment. Smith-Waterman [1] and Needleman-Wunsch [2] algorithms are sequence alignment algorithms that determine the maximal alignment of two sequences based on certain parameters, and determine a score that represents the quality of the alignment and the degree of similarity of the two sequences. Smith-Waterman generalized the method of Needleman and Wunsch to satisfy all the metric conditions[3].For pairwise alignment analysis, well-known systems such as BLAST [5] and FASTA [6] work well when comparing sequences within ten of thousands of nucleotides. The definition of sequence alignment for a pair of sequences can be generalized to the multiple sequence alignment problem. Multiple sequence alignment of k sequences involves a k-row alignment matrix, and there are various scoring schemes assigning a weight to each column. Among all the methods for multiple sequence alignment, progressive alignment is the most popular technique because of its simplicity and efficiency. Multiple sequence alignment refers to the search for maximal similarity in three or more sequences [4]. In this paper, based on the observation of aligned sequences, we propose an efficient partitioned algorithm for MSA. The algorithm solves the multiple sequence alignment in three stages. First, an automated partitioning strategy is used to divide the set of sequences into several subsections. Then, a multiple sequence alignment algorithm based on progressive alignment is used to align sequences of each subsection. Finally, an alignment of the original sequences is obtained by assembling the results from multiple partitions. Experimental results showed that the algorithm can significantly reduce the running time and improve the solution quality.

2. Multiple sequence alignment : definition and previous work.

Multiple sequence alignment is an extension of pair wise alignment which align multiple related sequences to achieve optimal matching of sequences. There is a unique advantage of multiple sequence alignment because it reveals more biological information than pair wise alignment. For example, it allows the identification of conserved sequence patterns and motifs in the whole sequence family, which are not obvious to detect by comparing only two sequences. Many conserved and

Manuscript received October 5, 2008
Manuscript revised October 20, 2008

functionally critical amino acid can be identified in a protein multiple alignment. Multiple sequence alignment is also pre-requisite to carrying out phylogenetic analysis of sequence families and prediction of protein secondary and tertiary structures. Multiple sequence alignment also has applications in designing degenerate polymerase chain reaction (PCR) primers based on multiple related sequences.

2.1 Estimation of the matching score

For a given two sequences $x = x_1\ x_2\ x_3 \ldots x_i \ldots x_n$ and $y = y_1\ y_2\ y_3 \ldots y_j \ldots y_m$. Construct an (n+1) x (m + 1) matrix F. Its $(i, j)^{th}$ element F (i,j) for i = 1,…,n. j = 1,…,m is equal to the score of an optimal alignment between $x_1\ x_2\ x_3 \ldots x_i$ and $y_1\ y_2\ y_3 \ldots y_j$. The element F (i, 0) for i = 1,…,n is the score of aligning $y_1\ y_2\ y_3 \ldots y_j$ to a gap region of length j. Build F recursively initializing it by the condition F (0,0) = 0 and then proceeding to fill the matrix from the top left corner to the bottom right corner. If F (i – 1,j-1), F(i-1, j) and F(i , j-1) are known, F (i ,j) is clearly calculated as follows

$$F(i,j) = \max \begin{cases} F(i-1, j-1) + S(x_i, y_j) \\ F(i-1, j) - d \\ F(i, j-1) - d \end{cases}$$

Where $S(x_i,y_j)$ is a substitution matrix. We considered blosum 62 in our implementation. There are three possible ways to obtain the best score F (i ,j): x_i can be aligned to y_j, or x_i is aligned to gap or y_j is aligned to a gap. Calculating F(i ,j), we keep a pointer to the option from which F(n,m) was produced. When we reach F(m,n), we trace back the pointers to recover the optimal alignments. The value F (n, m) is exactly their score. The general idea is to first select a sequence which is most similar to all the other sequences and then to use it as center of star aligning all the other sequences to it.

3. Automated Partitioning Strategy of sequences set

Our approach is to partition the overall problem into smaller subproblems that are exponentially easier to solve. Suppose we are given a family $S = (S_1,...,S_N)$ of N sequences, we partition the sequences into several subsets of segments vertically. If we cut each sequence S_i at a suitable position c_i near to the midpoint, we obtain two new families of shorter sequences, one family consisting of the prefixes $S^p = (S^p{}_1(c_1), S^p{}_2(c_2)..., S^p{}_N(c_N))$ and suffixes $S^s = (S^s{}_1(c_1), S^s{}_2(c_2)., S^s{}_N(c_N))$.If we can align these two new families of sequences optimally, we can concatenate our resulting alignments to obtain an alignment of the original sequences. Such a partitioning method can be applied recursively to reduce the original problem to multiple smaller MSA problems until the lengths of the subsequences are all less than an acceptable threshold.

4. Progressive alignment for MSA subproblems

Progressive alignments are by far the most widely used multiple sequence alignment method. This approach has the great advantages of speed and simplicity combined with reasonable sensitivity. Among the progressive algorithms, Clustal-W [7] is the most popular program based on the improved algorithm presented by Feng and Doolittle [8]. Progressive alignment produces multiple alignments from number of pair wise alignments. First two sequences are chosen and aligned, then third sequence is chosen and aligned to alignment of first two sequences and the process continues until all the sequences have been used. We assumed same scoring scheme as pairwise alignment. Suppose we have n sequences $x^1 = x^1{}_1 \ldots x^1{}_{m1}$, $x^2 = x^2{}_1 \ldots x^2{}_{m2}$, …, $x^n = x^n{}_1 \ldots x^n{}_{mn}$. Let $i_1,\ldots,i_n$ be integers with $0 \le i_j \le m_j$, j = 1, …,n, where at least one number is non- zero. Define $F(i_1,\ldots,i_n)$ as the maximal score of an alignment of the subsequences ending with $x^1{}_{i1} \ldots x^n{}_{in}$ (if for some j we have $i_j = 0$, then the other subsequences are aligned to gap region). Blosum 62 substitution matrix was taken into consideration. Gap penalty, distance is taken as -1 and 2.

5. Divide Progressive alignment MSA

After the set S of N sequences of a maximum length L being partitioned into sets of short sequences of a maximum length l, we can align these sets of subsequences separately. we developed a Progressive alignment algorithm as the basic solver for subproblems. The algorithm of multiple sequences alignment based on our partitioning strategy and Progressive alignment algorithm is described below. Both algorithm shows less time than progressive sequence alignment.

Divide_Progressive _MSA(*S,,l, S'*)
Input: S: set of sequences to be aligned;
Output: S': aligned sequences;
Begin
1.partition(*S,l*, P,*N*);
2.for i=1,2..num
3.Progressive-msa (Pi , P'i)
4. end for .
5. assemble Pi (i=1,2,…,N) to form S' .
6. end partition(*S,l*, P,*N)*
Input: set of sequences to be partitioned;
l: limit of length of subsequences.

Output: P: set of subsequence derived from S
N : number of sets in P.
Begin
partition(S^p, l, P^p ,N1)
partition(S^s,*l*,P^s,N2);
$P = P^p \cup P^s$
Divide_Progressive_MSA(S,l,S')
Input: S: set of sequences to be aligned;
Output: S': aligned sequences;
Begin
if the length of the sequences in S are larger than *l*
then
1. Partition (S,S^p,S^s);
2.Progessive-msa (S^p,*l*,P^p);
3.Progessive-msa (S^s,*l*,P^s);
4.$P = P^p \cup P^s$;
end if
end.

6. Experimental results

We tested the Divide-Progressive-MSA using a dataset of BChE sequences (http://www.ncbi.nlm.nih.gov/) we also tested the algorithm of Progressive-msa alignment algorithm as an independent sequence alignment algorithm to compare their performance with Divide-Progressive-MSA. Compared to Progressive alignment , the proposed partitioning method is faster. For example, for 9 sequences with length of 603, the running time of Progressive alignment is 100 to 150 seconds. while the running time of Divide-Progressive-MSA is 50 to 80seconds. Experimental results also show that the proposed Divide-Progressive-MSA algorithm improves the alignment accuracy for long sequences and requires less computational time than Progressive alignment without partitioning. A comparison of the running time of Divide-Progressive-MSA and Progressive alignment on sequences with 602-850 characters and with an increasing number of sequences is shown in Figure 2. We can see that the performance gain of using the partitioning strategy increases as the number of sequences increases.

6.1 Sample data

Selected amino acid sequences of BChE protein: Dataset of BChE Sequences was collected from (http://www.ncbi.nlm.nih.gov/). accession number and name of sequences are P21927: Chle_rabit, NP000046: Homo sapiens, XP516857: Pan troglodytes CAH92116: Pongo pygmaeus, NP_001070374 : Bos taurus, NP_001075319: Equus caballus, NP_001009364 : Felis catus, O62761: Chle_pantt, XP_545267: Canisfamiliaris, NP_033868: Mus musculus, XP_001505841: Ornithorhynchus anatinus, NP_075231: Rattus norvegicus, NP_989977 : Gallus Gallus. BChE Bacteria sequences: Thiocapsaroseopersicina, Chlorobium tepidum, Rhodobacter sphaeroides, Rhodobacter capsulatus, Synechocystis sp
pcc6803, Rosebacter denitrificans, Heliobacillus mobilis, Bradyrhizobium japonicum, Rhodopseudomonas palustris, Rhizobium etli cFN42, Lawsonia intracellularis, Rubrivivax gelatinosus, Candidatus Kuenenia stuttgartiensis, Bradyrhizobium sp, Chloroflexus aggregans.

7. Conclusions

The multiple sequence alignment based on partition and progressive alignment was proposed. We have designed an automated partitioning algorithm that can suboptimally divide a MSA set into multiple subproblems. To solve each subproblem efficiently, procedure of partitioning is applied to prefix family in recursive manner until the lengths of subsequences are equal to *l*. Experimental results demonstrated the proposed algorithm requires less time than progressive alignment algorithm without partitioning.

Sequence1	:MVTEIHF-L-LWILLL-C------MLFG--K-SH-T-EEDVI-ITTK-T
Sequence2	:M-------------------------------H-----S------KVT
Sequence3	:M—SVQSNLQAGAAAASCISPKYYMIFTPCKLYHLCCRESEINMHSKVT
Sequence4	:M---------------------------------R-S------KVT
Sequence5	:MV-T-RSS-H-TE---DVI------I-TT-KNGRI--R-G-INL----P
Sequence6	:MQSRSTV-IYIRFVLWFL-LL---WV-LFE--K-SH-T-EE-DIIITTK
Sequence7	:M------QS-W--GTII--CI--RILLRF--L-LLWVLI--G—NSHTE
Sequence8	:EEDIIITTKNGKVR-G------MN-----L--P--V--L----G--G—
Sequence9	:MQSKGTIISIQFLLRF-LL----L---WVLI-G--K-SH-T-EE-DI-I
Sequence10	:MRSKGTILSIRL-L-LWFLL---L---WVLI-G--K-SH-T-EE-DIVI
Sequence11	:MQTQHTKVTQTHFL-LWILLL-CM-P-----FG--K-SH-T-EEDFI-I
Sequence12	:MWSTGTSVPCQFLTWFL-LSCMLVSKSYMEEDF----V--I---TTK-K
Sequence13	:MVTEIHFLLWILLLCMLFGKSHTEEDVIITTKTGRVRGLSMPILGGTVT
Sequence14	:MVWANGMSICARFLMWLLLLFMFIRKVVPEDNV-ITTE-K-GRV—RGT
Sequence15	:E--
Sequence1	:-G-R-VRG-LSM-PILG-G-T---VT-AF---LGIPYAQPPLGSLRFKK
Sequence2	:ICIRFLFWFLLLCMLIGKSHTEDDIIIATKNGKVRGMNLTVFGGTVTAF
Sequence3	:ICIRFLFWFLLLCMLIGKSHTEDDIIIATKNGKVRGMNLTVLGGTVTAF
Sequence4	:ICIRFLFWFLLLCMLIGKSHTEDDIIIATENGKVRGMNLTVLGGTVTAF
Sequence5	:-----VF---------G-G-T---V---T--AFL-GIPY-AQPPLGRLR
Sequence6	:GKV--RGMH--LPV--L-G-G-T---V----TAFLGIPYAQPPL-G—R
Sequence7	:DII-IT--------TKN-GKVRGMN—LPVLGGTVTAFLGIPYAQPPLG
Sequence8	:-VT--A--FL----------G---------I-----P-YAQP-PL-G-R
Sequence9	:--ITT-KNGKVRG-MNL-PVL------DG---TVTAFLGIPYAQPPL-G
Sequence10	:TK-N----GKVRG-MNL-PVL------DG---TVTAFLGIPYAQPPL-G
Sequence11	:TKNGKV--RGMN--LPV--L-G---G------TVTAFLGIPYAQPPL-G
-	
-	
sequence1	
sequence2	
-	
-	

Fig1 Results from the alignment of BChE sequence by progressive alignment algorithm

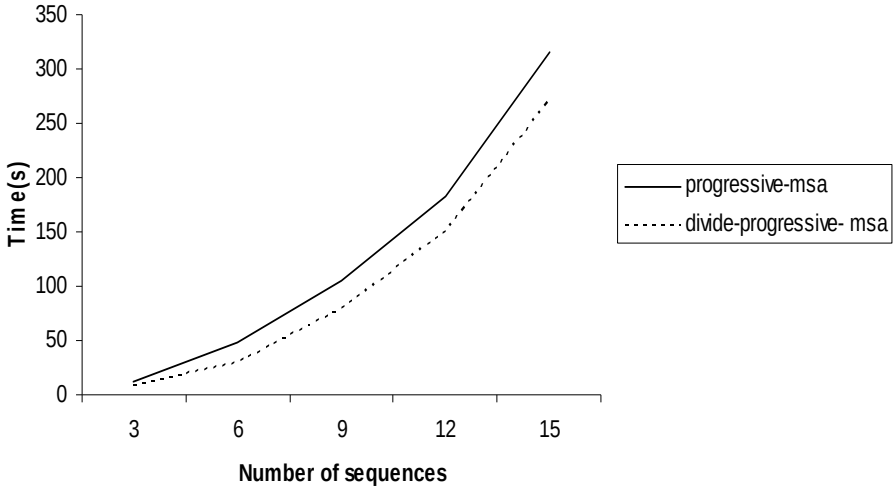

Fig 2 Time taken verses number of sequences

Comparison of Progressive-MSA and Divide-Progressive –MSA on amino acid sequence .

Reference

[1] T. Smith and M. Waterman, "Identification of common molecular subsequences, "*Journal of Molecular Biology*, vol. 147, pp. 195–197, 1981.
[2] S.B.Needleman, and C.D.Wunsch, "A General method applicable to the search for similarities in amino acid sequence of two proteins", *Journal of Molecular Biology*, 48pp.443-453, 1970.
[3] Sellers, P. (1974) An algorithm for the distance between two finite sequences. Comb. Theory 16:253-258.
[4] Chan, S.C., Wong, A.K.C., and Chiu, D.K.Y. (1992) A survey of multiple sequence comparison methods, Bull. Math. Biol.
[5] R.C.Gonzalez and R.E.Woods ,Digital image processing, second ed. Prentice Hall,2001.
[6] W.R.Pearson, "Comparison of methods for searching protein sequence database, "Protein Sciences,vol 4, no.6, pp.1145-1160, june 1995.
[7] Thompson, JD, Higgins, DG and Gibson, TJ(1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position specific gap penalties and weight matrix choice. Nucleic Acids Research,(1994), vol.22,No.22.4673-4680.
[8] Feng D-F, Doolittle RF: Progressive sequence alignment as a prerequisite to correct phylogenetic trees. J. Mol. Evol. 25,(1987)351-360

Authors:

P V Lakshmi received M.Phil degree in Mathematics from Andhra Universityin 1997,M.Tech degree from Andhra University in 2001. Working as Associate professor in the Department of Computer Science and Engineering, Gitam Institute of Technology, GITAM University, Visakhapatnam, Andhra Pradesh, India. Working on computational analysis of BChE sequences. Research interest : Bioinformatics, Cryptography and Network Security, Theory of computation.

Dr. Allam Appa Rao has received PhD in Computer Engineering from Andhra University,Visakhapatnam Andhra Pradesh, India. Currently he is Vice chancellor of JNT University, Kakinada and Professor in Bioinformatics & Computational Biology. Dr.Allam Appa Rao is a member of professional societies like IEEE, ACM and a life member of CSI and ISTE .He is International Editorial Board member. Organized International Workshops / Seminars / Symposia. Areas of Specializations: Bioinformatics and Computational Biology, Knowledge Management, Agents, Machine Vision, Software Engineering, Network Security.

Dr G R Sridhar, an endocrinologist, is Adjunct Professor, Bioinformatics, Andhra University College of Engineering. He was Chairman, Scientific Committee Annual Conference of RSSDI (2005).He is currently Chairman, Indian Chapter, American Association of Clinical Endocrinologists (2005-7). Dr Sridhar was the founder Editor, Indian Journal of Endocrinology and Metabolism, (1997-2000),' Widely published, he contributed chapters to 'RSSDI Textbook of Diabetes' and to 'API Textbook of Medicine'. A fellow of Madras Science Foundation, he was honored with RSSDI Oration, 2007, the Hoechst Senior lecturer ship in diabetes (2002) and Boehringer Knoll lecturer ship in Diabetes (1997). Dr Sridhar's major areas of research interest are in Clinical informatics, computational biology and bioinformatics, psychosocial aspects of diabetes.