A Faster Fitness Calculation Method for Genetic Algorithm Based Multiple Protein Sequence Alignment

Sagnik Banerjee, Tamal Chakrabarti, Devadatta Sinha

Abstract— Sequence alignment is a method to establish similarity between sequences. Sometimes it is necessary to compare many sequences simultaneously to establish evolutionary relationship between them. The process of aligning multiple sequences simultaneously to achieve maximum similarity is called Multiple Sequence Alignment problem. Multiple Protein Sequence Alignment is an NP Hard problem. So there have been several attempts made to approximate the solution using genetic algorithm, where it is necessary to calculate the fitness of each chromosome in the population for every generation. In this paper we have presented a scheme that will help calculate the fitness of the chromosomes faster, thereby reducing time of the alignment process.

Index Terms— Proteins, Bioinformatics, Genetic Algorithm, Fitness Function, Multiple Sequence Alignment.

1 INTRODUCTION

Sequence alignment [6] is a method of arranging the sequences of DNA, RNA or proteins to identify regions of similarity. When two sequences are aligned to locate regions of similarity between them, the problem is called Pairwise Alignment. When such alignment is done for more than two sequences at one go, it is called multiple sequence alignment (MSA) [8], [9]. Similarities between sequences arise due to the functional, structural or evolutionary relationship among them. Applications of sequence alignment are commonly observed in several fields, such as molecular biology, speech recognition and computer science. Multiple DNA sequence alignment has been applied to develop phylogenetic trees. Sequence alignment has played a major role in molecular sequence analysis, such as that of proteins. In bioinformatics multiple sequence analysis is important in the study of evolution, control of gene expression and also in protein structure/function relationships. Fig. 1 shows an example of such an alignment.

- *Sagnik Banerjee is currently pursuing master's degree program in Computer Science and Engineering from Jadavpur University, Kolkata, India, PH-919038749129. E-mail: sagnikbanerjee15@gmail.com*
- *Tamal Chakrabarti is currently working as an Assistant Professor in the department of Computer Science & Engineering at Institute of Engineering & Management, Kolkata, India, PH-919836237632. E-mail: tamalc@gmail.com*
- *Devadatta Sinha is currently working as a Professor in the department of Computer Science & Engineering at Calcutta University, Kolkata, India, PH-919830269278. E-mail: devadatta.sinha@gmail.com*


```
MVTISCTGSSSNIGAG-NHVKWYQQLPG
FVTISCTGTSSNIGS--ITVNWYQQLPG
PLRLSCSSSGFIFSS--YAMYWVRQAPG
SLSLTCTVSGTSFDD--YYSTWVRQPPG
GPEVTCVVVDVSHEDPQVKFNWYVDG--
HATLVCLISDFYPGA--VTVAWKADS--
EAALGCLVKDYFPEP--VTVSWNSG---
DVSLTCLVKGFYPSD--IAVEWWSNG--
```

Fig 1. An example of Multiple Sequence Alignment of eight protein sequences

The gap characters '-' are inserted in sequences to enforce regions of similarity to come together and form a good alignment. The quality of the alignment needs to be measured using some heuristic. The most commonly used method is sum-of-pairs score. The score of each amino acid pair is often determined from the PAM250 matrix.

The PAM (Point Accepted Mutation) matrix was calculated by observing differences in closely related proteins. The PAM1 matrix estimates what rate of replacement would be expected if 1% of the amino acids had changed.

The problem of finding the best possible alignment for multiple sequences simultaneously is NP-Hard. Therefore instead of finding the best alignment there has been extensive research done to find a near optimal solution using soft computing techniques such as Genetic Algorithm (GA) [1], Simulated Annealing (SA) [3], [4], Particle Swarm Optimization

(PSO) [2] etc. In GA the chromosomes are a prospective solution and they encode a particular alignment. It is necessary to determine the strength/fitness of each chromosome. The sum-of-pairs technique to compute scores has a complexity which is quadratic in the number of sequences to be aligned. In this paper we have presented a scheme to calculate the sum-of-pairs much faster, thereby reducing the time requirement for the complete algorithm.

2 Related Work

There has been a lot of research done to find alignments that are almost as good as the optimal alignment. In [1] Zhang et. al. have proposed a novel method of population initialisation and of crossover. Chang et. al. in [5] has successfully combined fuzzy arithmetic with GA to arrive at better alignments. In [7] Nguyen et al. presents a hybrid scheme where they convert the MSA problem to the problem of finding the shortest path in a weighted directed acyclic *k*-dimension graph (where k is the number of sequences). In a new approach taken by Hu in [10], the author has combined chaos and differential evolution to overcome the local optima of chaotic local search. Lai et. al. in [11] have suggested new genetic operators that direct the GA towards better solutions.

In all the above methods, the fitness of chromosomes needs to be calculated. Also after each generation, the fitness of the new chromosomes must be calculated. Therefore it is highly desirable to reduce the time required for calculating the fitness of chromosomes. This situation gains more importance in a multiple sequence scenario where calculating fitness alone can take up a lot of time.

3 Alignment Scores and Fitness Function

3.1 Alignment Scores

In order to calculate the fitness of each chromosome, which represents an alignment, a scoring function is required. Most often the PAM250 matrix has been used for the scores.

3.2 Sum-of-pairs score

Suppose a chromosome (an alignment) has N rows and M columns, i.e. the chromosome represents an alignment of N sequences consisting of M columns. The fitness of the chromosome is the sum of the sub-scores obtained from each column. Let us consider any such column j and represent the alignment in the j^{th} column as $(A_{j1}, A_{j2}, \ldots, A_{jN})^T$. The sub-score of the j^{th} column can be represented as

$$SB(j) = \sum_{p=0,q=1}^{N-1} f(A_{jp}, A_{jq}) \quad p<q \qquad (1)$$

Here $F(A_{jp}, A_{jq})$ represents the score/penalty of aligning amino acid A_{jp} with amino acid A_{jq}. This score is obtained from the PAM250 matrix.

Our attempt in this paper has been to reduce the computations required to compute SB(j) for all $0 \le j \le M$. In general there are ${}^{N}C_2 = \frac{N(N-1)}{2}$ computations needed for each j. With increasing value of N this computation gets expensive. Moreover for a single chromosome, fitness calculation requires summation of SB(j) over all *j*'s.

$$S(A) = \sum_{j=0}^{M-1} SB(j) \qquad (2)$$

Moreover there is the requirement to impose affine gap cost for each of the gaps introduced. The penalty for gap initiation is 5 and for gap extension is 2. Gap extension has been penalised less than gap initiation because gap extension is biologically more favourable than gap initiation. Hence the equation for fitness calculation becomes

$$S(A) = \sum_{j=0}^{M-1} SB(j) + gap\_costs \qquad (3)$$

This fitness S(A) needs to be computed for every chromosome in the population and for all the generations. Therefore any reduction of time requirement achieved in calculating the chromosome fitness can be extremely beneficial.

4 Pre-Calculations

In MSA we need to calculate the sub-scores for each column in any given alignment. This task of calculating the scores is repetitive. Therefore it is beneficial to pre-calculate a part of the data once and use that pre-computed data for calculating sub-scores. Fig. 1 gives a possible alignment of eight protein sequences. In order to calculate the sub-score of the 0^{th} column we need to perform the following computation:

```
SB(0)=f(M,F)+f(M,P)+f(F,P)+f(M,S)+f(F,S)+f
(P,S)+f(M,G)+f(F,G)+f(P,G)+f(S,G)+f(M,H)+f(
F,H)+f(P,H)+f(S,H)+f(G,H)+f(M,E)+f(F,E)+f(P
,E)+f(S,E)+f(G,E)+f(H,E)+f(M,D)+f(F,D)+f(P,
D)+f(S,D)+f(G,D)+f(H,D)+f(E,D)
```

This involves adding up 28 numbers. But if some of these additions are pre-computed and stored in a file for later use then it will speed up the process. We have prepared 3 such files which contain pre-computed data. In one file we have considered triads of amino acids. This file contains the scores that will be obtained by aligning all possible triads of amino acids. For example if MFP is one such triad, then we compute f(M,F) + f(M,P) + f(F,P) and store it in a file. Later this value was read from the file and used. This is computationally effective because we no longer need to calculate 3 values.

IJSER

Instead we read only one value from the file. Similarly two more files were created which contained alignment scores obtained by aligning 4 amino acids and 5 amino acids respectively. For example let us calculate the sub-score for column 0. We make use of the files we created. We need to compute the sum of the following 28 pairs:

```
MF
MP  FP
MS  FS  PS
MG  FG  PG  SG
MH  FH  PH  SH  GH
ME  FE  PE  SE  GE  HE
MD  FD  PD  SD  GD  HD  ED
```

Now we read from the file, the value of aligning amino acids MFPS, which has been pre-computed. It should be noted that in the file the value corresponding for MFPS is actually the sum of the scores obtained by aligning the pairs,

```
MF
MP  FP
MS  FS  PS
```

Reading this value from the file is basically a single operation as opposed to 6 calculations required in sum-of-pairs method. The following figures illustrate the procedure of considering triads of amino acids or quartet of amino acids to reduce computations.

```
MF                  MFPS
MP  FP
MS  FS  PS
MG  FG  PG  SG
MH  FH  PH  SH  GH
ME  FE  PE  SE  GE  HE
MD  FD  PD  SD  GD  HD  ED
```

Fig 2. Pairs of amino acid alignment computation reduction achieved by considering the quartet **MFPS**.

```
MF                  MFPS, MGH
MP  FP
MS  FS  PS
MG  FG  PG  SG
MH  FH  PH  SH  GH
ME  FE  PE  SE  GE  HE
MD  FD  PD  SD  GD  HD  ED
```

Fig 3. Pairs of amino acid alignment computation reduction achieved by considering the triad **MGH**.

```
MF                  MFPS, MGH, FED
MP  FP
MS  FS  PS
MG  FG  PG  SG
MH  FH  PH  SH  GH
ME  FE  PE  SE  GE  HE
MD  FD  PD  SD  GD  HD  ED
```

Fig 4. Pairs of amino acid alignment computation reduction achieved by considering the triad **FED**.

```
MF                  MFPS, MGH, FED, SHE
MP  FP
MS  FS  PS
MG  FG  PG  SG
MH  FH  PH  SH  GH
ME  FE  PE  SE  GE  HE
MD  FD  PD  SD  GD  HD  ED
```

Fig 5. Pairs of amino acid alignment computation reduction achieved by considering the triad **SHE**.

```
MF                  MFPS, MGH, FED, SHE,
MP  FP              PGE
MS  FS  PS
MG  FG  PG  SG
MH  FH  PH  SH  GH
ME  FE  PE  SE  GE  HE
MD  FD  PD  SD  GD  HD  ED
```

Fig 6. Pairs of amino acid alignment computation reduction achieved by considering the triad **PGE**.

```
MF                  MFPS, MGH, FED, SHE,
MP  FP              PGE,
MS  FS  PS
MG  FG  PG  SG
MH  FH  PH  SH  GH
ME  FE  PE  SE  GE  HE
MD  FD  PD  SD  GD  HD  ED
```

Fig 7. Pairs of amino acid alignment computation reduction achieved by considering the triad **PHD**.

```
MF                  MFPS, MGH, FED, SHE,
```

```
MP  FP          PGE, SGD
MS  FS  PS
MG  FG  PG  SG
MH  FH  PH  SH  GH
ME  FE  PE  SE  GE  HE
MD  FD  PD  SD  GD  HD  ED
```

Fig 8. Pairs of amino acid alignment computation reduction achieved by considering the triad **SGD**.

We find that the pairs FG, FH, ME and MD are not considered by any triad or quartet. So we have to add the scores corresponding to these pairs separately.

```
MF                  MFPS, MGH, FED, SHE,
MP  FP              PGE, SGD,
MS  FS  PS          FG, FH, ME, MD
MG  FG  PG  SG
MH  FH  PH  SH  GH
ME  FE  PE  SE  GE  HE
MD  FD  PD  SD  GD  HD  ED
```

Fig 9. Completion of calculation of sub-score by considering pairs **FG**, **FH**, **ME**, **MD**.

So these figures illustrate how 28 computations can be reduced to merely 11 computations resulting in 60.71% improvement. This could be achieved by keeping the triads, quartet and quintet of amino acids pre-computed.

5 Observations

We conducted an experiment to analyse the improvement achieved by performing fitness value computations using the method proposed by us. Chromosomes were created randomly for this purpose. The largest chromosome consisted of 3245 amino acids and the smallest chromosome was 260 amino acids long. In few cases the number of chromosomes used was less than 500, but later while comparing the two techniques, the results were adequately extrapolated. The environment for the experiment was:

- Hardware
  - Processor - Intel® Core™ i7-3610QM CPU @ 2.30GHz × 8
  - RAM - 8 GB
  - Disk - 1000 GB
- Software
  - Operating system – Windows 7 Ultimate
  - OS type – 64-bit
  - Programming Language – Python version 3.2
  - IDE: Eclipse

The results are summarised below in tabular format.

TABLE 1
COMPARISON BETWEEN SUM-OF-PAIRS AND PRE-CALCULATION METHOD

No. of Sequences	Time Taken by Sum-of-Pairs Method (in seconds)	Time Taken by Pre-Calculation Method (in seconds)
3	7.167	11.031
4	7.901	17.577
5	15.044	26.74
6	38.882	42.589
7	43.04286	52.48571
8	64.97143	66.32143
9	105.716	143.278
10	105.716	175.768
11	115.078	209.614
12	116.346	242.326
13	274.72	403.8767
14	316.3467	471.96
15	300.4033	538.91
16	307.1333	611.44
17	1093.09	2072.9
18	1093.09	2336.66
19	1103.68	2579.87
20	1143.51	2840.04

From the results obtained in Table 1, it is evident that with increase of number of sequences the pre-calculation method performs better than sum-of-pairs method, as is depicted in the graph below.

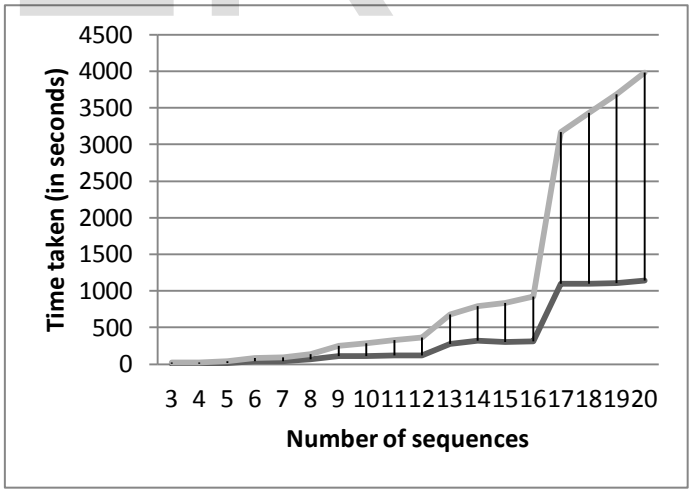

Fig 10. Comparison between sum-of-pairs method and pre-calculation method

The percentage improvement of the time of the pre-calculation method as compared to the sum-of-pairs method is displayed in the histogram below.

IJSER

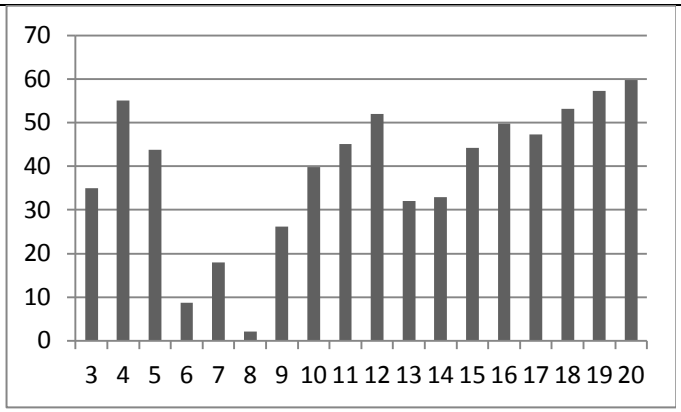

Fig 11. Percentage of improvement of time of pre-calculation method with respect to the sum-of-pairs method

6 CONCLUSIONS

From Table I and the graphs it is clear that the Pre-Calculation method performs better than the Sum-of-Pairs method. Typically for finding the MSA the GA was initialized with 10 chromosomes and executed for 500 generations. Though we did not introduce any alteration to the procedure of GA, there was significant improvement of time because the finesses of the chromosomes are being computed much faster.

REFERENCES

[1] Guang-Zheng Zhang; De-Shuang Huang; "Aligning multiple protein sequence by an improved genetic algorithm," Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on , vol.2, no., pp. 1179- 1183 vol.2, 25-29 July 2004.

[2] Hai-Xia Long; Wen-Bo Xu; Jun Sun; Wen-Juan Ji; "Multiple Sequence Alignment Based on a Binary Particle Swarm Optimization Algorithm," Natural Computation, 2009. ICNC '09. Fifth International Conference on , vol.3, no., pp.265-269, 14-16 Aug. 2009.

[3] Omar, M.F.; Salam, R.A.; Rashid, N.A.; Abdullah, R.; "Multiple sequence alignment using genetic algorithm and simulated annealing," Information and Communication Technologies: From Theory to Applications, 2004. Proceedings. 2004 International Conference on , vol., no., pp. 455- 456, 19-23 April 2004.

[4] Hongwei Huo; Stojkovic, V.; , "A simulated annealing algorithm for multiple sequence alignment with guaranteed accuracy," Natural Computation, 2007. ICNC 2007. Third International Conference on , vol.2, no., pp.270-274, 24-27 Aug. 2007.

[5] Ping-Teng Chang; Lung-Ting Hung; Kuo-Ping Lin; Chih-Sheng Lin; Kuo-Chen Hung; , "Protein Sequence Alignment Based on Fuzzy Arithmetic and Genetic Algorithm," Fuzzy Systems, 2006 IEEE International Conference on , vol., no., pp.1362-1367, 0-0 0.

[6] Agrawal, A.; Xiaoqiu Huang; , "Pairwise DNA Alignment with Sequence Specific Transition-Transversion Ratio Using Multiple Parameter Sets," Information Technology, 2008. ICIT '08. International Conference on , vol., no., pp.89-93, 17-20 Dec. 2008

[7] Hung Dinh Nguyen; Yoshihara, I.; Yamamori, K.; Yasunaga, M.; , "A parallel hybrid genetic algorithm for multiple protein sequence alignment," Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on , vol.1, no., pp.309-314, 12-17 May 2002

[8] Agrawal, A.; Khaitan, S.K.; , "A new heuristic for multiple sequence alignment," Electro/Information Technology, 2008. EIT 2008. IEEE International Conference on , vol., no., pp.215-217, 18-20 May 2008

[9] Carrillo H.; Lipman D. "The Multiple Sequence Alignment Problem in Biology," SIAM Journal on Applied Mathematics, vol. 48 No. 5, October 1988

[10] Gui-wu Hu; "Chaos-differential Evolution for Multiple Sequence Alignment," *Intelligent Information Technology Application, 2009. IITA 2009. Third International Symposium on* , vol.2, no., pp.556-558, 21-22 Nov. 2009

[11] Chih-Chin Lai; Chih-Hung Wu; Cheng-Chen Ho; "Using Genetic Algorithm to Solve Multiple Sequence Alignment Problem ", International Journal of Software Engineering and Knowledge Engineering Vol. 19, No. 6 (2009)

IJSER