JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Enhancement of the accuracy of protein Multiple Sequence Alignment by Hybrid-Genetic Algorithm

Narayan Behera[1,2] and Jeevitesh.M.S[2]

[1]Department of Applied Physics, School of Applied Natural Science, Adama Science and Technology University, Adama, P O Box 1888, Ethiopia

[2]Institute of Bioinformatics and Applied Biotechnology Electronics City, Phase 1, Bengaluru 560100, INDIA

ABSTRACT

We propose a stochastic computation approach for a meta-analysis of protein Multiple Sequence Alignments (MSA) obtained from different methods. The outputs of four individual MSA programs – ClustalW, Mafft, Muscle and T-Coffee – are combined and a Genetic Algorithm (GA) is used as an optimizer to find a better MSA. This GA basically uses the mutation, selection and recombination principles of evolution to discover the optimized sequence alignment. The performance of this Hybrid-Genetic Algorithm (HGA) is tested on the Homstrad and the Balibase version 3 benchmark reference protein datasets. The efficiency of protein sequence alignments is evaluated in terms of the Total Column (TC) score which is equal to the number of correctly aligned columns between the test alignment and the reference alignment divided by the total number of columns in the reference alignment. In terms of the TC scores, the HGA optimizer achieves, on an average, 3-16% better alignment over the above individual methods on the Balibase benchmark and 2-7% better on the Homstrad benchmark. The present HGA is a simple method that efficiently combines the outputs of various MSA programs and then creates a more accurate optimized alignment by evolutionary principles.

Key words: multiple sequence alignment, genetic algorithm, BaliBase and Homstrad benchmark, evolutionary principles, optimized alignment.

*corresponding author, Email: nbehera321@gmail.com

INTRODUCTION

Given a set of biological sequences, a Multiple Sequence Alignment (MSA) provides a way of identifying sequence conservation. In particular, the conserved amino acid stretches in alignments are strong

indicators of preserved three dimensional structural domains. They also provide information on evolutionary, molecular and functional relationships among the proteins. The MSA is one of the most commonly used techniques in sequence analysis in building profiles, finding phylogenetic reconstructions, making function assignment, generating structure prediction and discovering single nucleotide characterization. So the underlying quality of protein alignments depends on the computational methods used. But the accurate MSA algorithm is still a difficult task in terms of computational cost (1) and lack of proper objective function to assess the alignment quality (2). An exact solution is possible only when the number of sequences is small and related (3). Therefore, most alignment packages use heuristic progressive alignment algorithm that doesn't necessarily provide an optimal solution (4). Most sequence alignment programs use Blosum62/PAM matrix which are derived from high percentage sequence identity information. For an example, Blosum62 matrix is derived from many (more than 2000) protein patterns called as blocks. These blocks are composed of sequence segments that are identical in residues beyond
62%.

We know the fact that structures are more conserved than sequences, which means that at sequence level we may not find any identity but structurally the residues may be conserved. As most of the above matrix elements are derived from high percentage sequence identity, it is obvious to fail when sequence identity is low. When sequence identity falls below 30%, called as the "twilight zone" of protein alignments, alignment accuracies of the most reliable sequence alignment methods drop considerably. So the rational approach to this problem has been to replace sequence similarity with structural information. A number of attempts have been made to include structural information for sequence alignment, such as secondary structure (5-6) and tertiary structure (7-10). For large evolutionary distances, the sequence alignment algorithms that depend on the sequence similarity matrix become less reliable. That is why structure based information is necessary. Here a dataset containing the protein pairs having high structural similarity are superimposed for the derivation of the amino acid substitution matrix. Instead of sequence based matrix, we use structure based matrix derived on superimpositions from protein pairs of similar structures, but of low or no sequence similarity (11). This can enhance the sequence alignment procedure.

Some well known sequence alignment algorithms are ClustalW, Mafft, Muscle and TCoffee. The traditional ClustalW uses the progressive alignment approach (12) to align protein sequences. This involves making a preliminary estimation of how the sequences are related using pair-wise alignments to obtain a similarity matrix. Then, using the knowledge of this similarity matrix, a guide tree is generated beginning with the most closely related sequences and concluding with the most distant. But it suffers from the problem that alignment errors made early in the process can never be rectified.

The M-Coffee (13) program is a meta-method that combines the output of various MSA programs into one single better alignment. It is a continuation of the T-Coffee (14) method that blends a set of alignment programs. The T-Coffee method uses progressive alignment and a consistency-based objective function that is optimized. An information library is used in the alignment procedure. The library is constructed from pairwise

residue-by-residue scores from many global and local alignments. This method tries to maximize the alignment score between the final multilple alignment and the above library of alignments. The Mafft program (15) is based on progressive alignment algorithm that uses Fast Fourier Transform to find the guide tree and subsequently obtain the required alignment. The Muscle method of sequence alignment (16) is based on progressive alignment algorithm that uses the log expectation score to align sequences by a guide tree. The recently develped one that is practically the best for multiple sequence alignment is the ProbCons (17). It is a pair-hidden Markov-model based progressive alignment algorithm that uses probabilistic consistency transformation to incorporate multiple sequence conservation information during pairwise alignment. The ProbCons achieves statistically significant improvement over the other methods. However, most of the algorithms perform poorly when applied to the analysis of sequences with low sequence similarity. The high-throughput technologies such as genome sequencing and structural proteomics have created the necessity to develop of very high quality protein alignments.

The high quality protein alignment benchmarks are needed to compare the effectiveness of various MSA tools. The Balibase 3.0 benchmark alignment database is a collection of 386 structural protein alignments which are manually verified alignments (18). They are based on three dimensional structural superpositions with correct alignments of conserved residues. The benchmark is organized into five different categories, each category represents some characterstics, such as high or low sequence identity, long or short sequences, and large insertions or deletions. The HOMologous STRucture Alignment Database (Homstrad) is a curated database of structure-based alignments (19). Here the structures of proteins are aligned using the programs COMPARER. Alignments having less than 4 sequences are not considered, thus resulting in 233 alignments. The basic purpose of a benchmark is to give a set of tests to compare the efficiencies of alternative computational tools. So the idea behind this benchmarking is that the average best performing package will be able to find the best alignment of uncharacterized protein sequences.

Here we propose a scheme that can use the sequence alignment outputs of different programs and effectively re-combine them and then use an optimization technique to create a better alignment. The opimization technique used is the Genetic Algorithm which is an adaptive heuristic search algorithm designed to simulate processes in natural systems necessary for evolution. It is modelled on the principles of evolution via natural selection, employing a population of individuals (multiple alignments) that undergo selection in the presence of variation inducing operators such as mutation and recombination. Holland first introduced this algorithm (20) and later it has been applied to many optimization problems in finding optimal and near optimal solutions. Genetic algorithm technique has been successfully implemented to multiple sequence alignment problems (21, 22, 23). Genetic algorithm as an alignment optimizer has been studied by taking Clustalw as the initial seeding alignment (24).

In this method a better solution is allowed to evolve over many generations starting from a set of approximate initial solutions (populations) until the most opimized solution is obtained. A fitness function is used to evaluate the success of individuals that survive the selection pressure. The creation of individuals in

successive generations depends on a fitness function. In our present work, new kinds of operators, such as block crossover and mutation are studied. The initial population of MSAs from four different program outputs - ClustalW, Mafft, Muscle, T-Coffee – are constructed with equal probability. The Hybrid-Genetic Algorithm (HGA) combines and improves the alignments until the most optimized alignment is obtained. In addition to this, the effect of using structural information via a structure based amino acid matrix (instead of BLOSUM62) is also considered. Our HGA method is tested on the most accepted benchmarks like the Balibase version 3 and the Homstrad. It achieves a statistically significant enhancement of alignment over the other methods.

MODEL

Objective function

Evaluation of a sequence alignment is made by using an Objective Function (OF) which is a measure of the multiple alignment quality. The fitness value is connected to the OF and reflects the solution's biological relevance and should ideally provide an important clue about the implicit structural and evolutionary relationships that exist among the aligned sequences. The approach in this model is to use a measure of multiple alignment quality and to optimize it using a Genetic Algorithm. We are implementing the sum of pairs method as a measure for alignment quality. The aim is to maximize the score of alignment. This sum of pair scores (S) is defined as ,

$$S = \sum_i \sum_j \mathbf{S}(i,j) \qquad \mathbf{1}$$

where i = 1, 2, .., n-1 (where n = number of sequences in the alignment), j = i +1, i+2, .., n and S(i,j) is the value obtained using structure based matrix. The overall alignment score of a MSA is the sum of each pair of rows. The alignment score of a pair of rows is the sum of the alignments of the individual pair of residues. We have implemented affine gap penalty. In this scheme two types of penalties are used for the score calculation: one for gap opening and the second for gap extension. The gap opening penalty is applied only once when a gap is introduced into the sequence and the gap extension penalty is added to the standard gap penalty for each additional gap. Optimum gap opening penalties are tested in the range from 5 and 20 and extension penalties between 0 and 2, observed that a gap opening penalty of 15, gap extension penalty of 0.9 yielded higher accuracy and terminal gaps are not scored.

Weights

Sequence weights are incorporated in a multiple sequence alignment in order to correct the unequal representation. For instance, consider a sequence profile that is derived from an alignment of 20 hemoglobin sequences and one myoglobin sequence. Once converted to a profile, the information from one myoglobin is not going to contribute much. It is important to avoid complete domination by hemoglobin sequence. So,

weights are used to avoid this type of bias. It is necessary to include the myoglobin sequence in alignment process. This method can be separated into two groups that are based on an alignment or on a phylogenetic tree.

Alignment based weighting method requires pair wise distances between sequences (25). Therefore, complex issues of tree topology and root placement are avoided.

Steps involved in alignment method:

1 .Count the number of non-identical residues (C) for each pair of sequences (ignoring gaps).

2 Divide by the length of the sequence (L) and obtain the result of pairwise distances $\frac{C}{L}$.

3 Generate a distance matrix for all pairs of sequences as specified in 1 and 2.

4 Generate the inverse of this distance matrix.

5 Compute the sum of each row of the matrix to get the weight corresponding to each individual sequence.

6 Compute the product of any two individual sequence weights and get the pair sequence weight (W_{ij}), then multiply this weight (W_{ij}) with the corresponding pair score (S_{ij}) obtained by sum of pair method (which includes gap opening and extension penalties).

7 Calculate the final score by taking the sum of all the products together,

$$\text{i.e., Final score} = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} W_{ij} \; S_{ij} \,, \qquad \mathbf{2}$$

Where n is the number of sequences.

Altschul Carroll Lipman (ACL) Method

The tree-based weights assume that the sequences are related through evolution. A reasonably correct tree can be deduced from pair wise distances. ACL method produces pair wise weights while the above methods give individual sequence weights (26).

Steps involved in ACL method are:

1.Construct a tree for the given alignment.

2.Calculation of variance-covariance matrix:

Consider the figure 1, there are six leaf-pairs (AB, AC, AD, BC, BD, CD) and the matrix is of dimension 6×6. It means that the covariance of the pair AC and pair BD is $\frac{6\times6}{10\times10}$ = 0.36. The covariance of pair AB and pair AC is $\frac{2\times2}{4\times10}$ = 0.10. The covariance is calculated using the formula $\frac{(L\_XY)(L\_XY)}{(L\_X)(L\_Y)}$ where L_X is the corresponding path length of the pair X, L_Y is the corresponding path length of the pair Y and L_XY is the shared path within the tree corresponding to pairs X and Y. This means that the variance or diagonal

terms in the variance-covariance matrix are all 1.0.

3.Find the inverse of the above variance–covariance matrix.

4.Once the inverse matrix is obtained, calculate the weights for all sequence-pairs by taking sum of the corresponding rows of the inverse matrix.

5.Calculate the final score by multiplying the score of each pair wise sequence alignment by its corresponding pair sequence weight, and sum all these products together.

It as been observed inclusion weighting scheme as shown a small improvement in accuracy of 1% on Balibase benchmark (27). So HGA is not implementing any kind of weighting scheme.

Genetic Algorithm operators

Mutation

Mutation is an important part of the search for the best solution as it helps to prevent the population from stagnating at any "local optima". Mutation is a genetic operator that alters one or more positions in the sequence from its initial state. Mutation is performed by inserting a gap randomly in a sequence. This can result in an entirely new alignment. With these new sequence alignments, the genetic algorithm may be able to arrive at a better alignment.

For each alignment in the population of alignments, gaps are inserted randomly with a fixed probability (p) given by the following formulae.

$$p = \frac{\ln(xy)}{I \times 10} \qquad \mathbf{3}$$

where x is the maximum length of a sequence in the multiple sequences, y is the number of sequences and I is the number of columns with identical residues (ignoring gaps)**.** If the calculated identity is zero, it is reassigned to one (for computational purpose). The equation (3) has been empirically obtained after analyzing a set of alignment data. For a gap insertion in a MSA, a random number r is generated in the range of 0 to 1 and for $r < p$, a gap is inserted at a random position else no gap is inserted in that alignment. After insertion of a gap, the remaining sequences of the MSA are padded with gaps so that all sequences are of the same length.

In the Hill Climbing mutation, for a current solution, a new solution is obtained by mutation. If the new solution is fitter, it is accepted. Otherwise the current solution is retained. The Hill Climbing algorithm works as follows:

1. For each alignment $\boldsymbol{a}$ of the population of alignments, its current fitness $f(\boldsymbol{a})$ is calculated.
2. Mutate $\boldsymbol{a}$ to produce a mutant $\boldsymbol{m}$ by inserting a gap randomly in one of the sequences and then gaps are padded at the end of the other sequences.
3. If $f(\boldsymbol{m})$ is fitter than $f(\boldsymbol{a})$ then, replace $\boldsymbol{a}$ with $\boldsymbol{m}$ else $\boldsymbol{a}$ is retained.

The fundamental idea behind this local search is that the good solutions tend to cluster together.

Selection operator

Those individual alignments that survive the process of selection serve as parental alignments for the next generation. The next generation is obtained from the current generation using the selection operators. The current population of alignments is sorted (in descending order) according to their alignment scores. The parental alignments are selected from the current population of alignments using the "Roulette wheel with replacement" method. This technique selects parental alignments (for recombination purpose) in proportion to their relative fitnesses. Here, the scores of parental alignments are sorted in descending order and normalized to positive values with the minimum value being 0. A random number r is generated between 0 to 100. The scheme of Roulette wheel selection is as follows. For all the normalized scores in the present generation, the scores are iteratively added for each MSA till the sum of scores becomes greater than or equal to r. The alignment, at which this condition is met, is selected as the parental alignment for recombination as shown in figure 2. Similarly, the second parental alignment is also selected. It is ensured that the two parental alignments are distinct.

Recombination operators

Recombination operators generate new alignments by combining the two existing parental alignments. We have implemented two types of operators.

Point crossover

The point crossover combines the two parental alignments through a single exchange. Here the first parent alignment is cut straight at some randomly chosen position. The second parent alignment is cut in such a way that the number of residues on either side of the cut exactly match those of the first parental alignment. This is done in order to conserve the number and order of residues in the original alignment as shown in figure 3.

Block crossover

Under this scheme the parental alignments are converted into blocks of varying sizes such that each block in both the parental alignments contain equal number of residues in each sequence. The blocks formed must contain at least one gap in any of the sequences of the alignment. Also, blocks consisting of only gaps are not permitted. Hence padding gaps that are present at the end of the sequences are removed before generating the blocks. This has no effect on the alignment score as a gap aligned with another gap doesn't contribute to the score.

Each block is compared sequentially from both the parental alignments. The blocks with higher scores are selected to form the recombinant. After each comparison the best block is appended to the recombinant being generated. Consider the figure 4 where parental alignments have been divided into 3 blocks each, conserving the number of residues in each block. Suppose, parent ***i*** contains blocks (i_1, i_2, i_3) and parent ***j*** contains blocks (j_1, j_2, j_3). To generate a recombinant the alignment scores of each block of parent ***i*** is compared with the corresponding blocks of parental alignment ***j*** i.e.,i_1 with j_1, i_2 with j_2 and so on. Suppose, score(i_1) > score(j_1), score(i_2) < score(j_2), and score(i_3) > score(j_3). The resultant recombinant now consists of the blocks (i_1,

j_2, i_3).

Elitism selection

Only a portion of the population of alignments is to be replaced during each generation. This means that the half of the high scoring alignments will survive unchanged while the other half is replaced by the alignments generated by recombination.

Termination of the program

The simulation terminates when the difference of the best fitness for ten consecutive generations is less than 1%. At the n^{th} (n>10) generation, the percentage differences between the best fitness of $(n-i)^{th}$ generation and $(n-10)^{th}$ generation are found, where i varies from 0 to 9. If all these ten differences are less than 1%, the program is terminated, else it proceeds to the next generation.

RESULTS

To assess the efficiency of the HGA model, two different protein benchmark suites: the Balibase 3.0 and the Homstrad are used. The program is implemented on a 3 GHz Intel Xeon Dual core processor with 8 GB RAM. Fedora core 6 is used as the operating system. The HGA program is compared with the ClustalW version 1.83, the T-Coffee version 4.96, the Mafft version 5.861 and the Muscle version 3.6. All the above programs are exceuted on default modes.

Alignment accuracy measurement

The alignment quality of each method is determined by measuring: Quality (Q) and Total Column (TC) scores. Q is the number of correctly aligned residue pairs between test alignment and reference alignment divided by the total number of aligned residue pairs in the reference alignment. TC is the number of correctly aligned columns between test alignment and reference alignment divided by the total number of columns in the reference alignment. In general, the TC score is lower than the Q score. However, the TC score provides a more important measure to evaluate the efficiency of a sequence alignment. So we are using TC score to find the best alignment for analysis purposes. Then the corresponding Q for that alignment is determined. The TC and the Q scores are calculated using a software QSCORE from the developers of Muscle.

Hybrid-Genetic Algorithm Parameters

The following model parameters are used in the HGA. The population of alignments is equal to 80 (20 from each MSA program). 50% of the alignments are generated from the block crossover and the point crossover with equal probability. The rest 50% alignments are chosen through elitism. The overall schema of HGA is shown in figure 5.

Since the HGA is a stochastic method, the output TC score for the best alignment depends on the initial conditions (the way gap insertions are distributed, recombination and selection procedures are carried out).

That is why six different simulations are carried out relaxing the above initial conditions. Finally the best TC score and the corresponding alignment is obtained out of the six different simulations. Then the Q score corresponding to that best alignment is calculated.

The HGA model is built using ANSI C Version 4.1 as a programming environment. Agglomerations of various MSA method outputs and analysis of the results are done by using Perl.

Statistical analysis

In order to compare the efficiencies of various MSA programs with the HGA method, we conduct the Friedman rank test. This is basically a non-parametric test. It makes no assumption about the distribution of alignment scores across different pairs of MSA programs. Here, instead of using alignment score directly, the ranking of the score across pairs of programs is used for finding the efficiency of a MSA method. The higher the alignment score of an alignment program, the better is its rank. Then the Ranksum is calculated as the sum of ranks for a given MSA program.

The concept of null hypothesis is used to compare the efficiencies of the two MSA programs in terms of the TC and the Q values. The null hypothesis says that a pair of programs is equally good. The Ranksum is further used to calculate the P-value which measures a probability factor for rejecting the null hypothesis. If the P-value is very small (say, ≤ 0.05), the above null hypothesis is rejected. Hence the higher the Ranksum, the better is the program. If the P-value is greater than 0.05, there is no statistically significant difference between the efficiencies of the two MSA programs. For a set of scores (say, Q and TC) the P-values are obtained using the Friedman rank test from the statistical analysis package R (http://www.r-project.org/).

The TC, the Q scores and the statistical significances of the alignments are summarized in tables 1-11. All the results shown are for six simulations, so there are six pairs of Q and TC for an alignment. But while deciding the best alignment among the six, the best TC is considered. Then the corresponding Q is considered for that alignment. As there is no straightforward relationship between Q and TC, a higher Q score might have been lost in some cases. On all the test sets and quality measures, HGA model achieves the highest ranking as well as a statistical significant enhancement over the well known alignment methods. The results of testing on the Homstrad benchmark alignment database are shown in Table 1. In the Ranksum of Friedman rank test, the program with the highest Ranksum means the program most often constructs the most accurate alignment. The HGA achieves improvement of 6.12% improvement over the ClustalW, 7.23% over the Mafft, 2.37 % over the Muscle and 5% over the T-Coffee in terms of TC on Homstrad. However, the HGA achieves enhancement of 3.5% over the ClustalW and 3.9% over the Mafft in terms of Q on Homstrad as shown in Table 2. To assess the significance of the differences in the overall Q and TC scores, we have performed a Friedman rank test for all pairs of programs. These results are summarized in Table 3. HGA achieves a statistical significant result across programs. To understand the role of structure based matrix in sequence alignment, we constructed alignment by using sequence similarity based matrix (Blosum62). The rest of the parameters are chosen by default. The analysis shows an enhancement of 1.3% over sequence based matrix as shown in Table 4. The results of performance of alignment program on Homstrad are shown in Table 5. HGA

again shows a strong lead in TC score although their running time prolonged than the other alignment program expect for SAGA (28). Is due to the nature of Genetic algorithm approach. But compared to conventional Genetic algorithm program like SAGA HGA attains lead in context of speed and accuracy. The results of comparison of sum-of-pair score across Homstrad are shown in Table 6. In which HGA attain the highest score compared to other program alignment output this proves HGA as an excellent optimizer. The result of comparison of SAGA and HGA program are shown in Table 7. The HGA achieves strongest performance in terms of Q and TC when compared to SAGA on Homstrad and Balibase benchmark while sustaining practical speed. The results on Balibase benchmark alignment database are shown in Table 8-11. The HGA achieves improvements of 8.87% over the ClustalW, 3.98% over the Mafft in terms of Q on Balibase benchmark as shown in Table 8. The HGA has enhancements of 16.24% over the ClustalW, 12.01% over the Mafft, 3.54% over the Muscle and 12.58% over the TCoffee in terms of TC on Balibase as shown in Table 9-10. The statistically significant differences in the overall Q and TC scores is shown in Table 11.

DISCUSSION AND CONCLUSION

In this current work we describe the HGA model which can efficiently combine the outputs of various MSA methods and pick one more accurate alignment. We show that the HGA model is able to improve sequence alignment by 3-16% in terms of TC on the Balibase benchmark and by 2-7% on the Homstrad benchmark. By using the HGA, we show that structural information makes it possible to improve alignment accuracy by 1.3% on the Homstrad.

It requires delicate analysis to obtain the best alignment. A number of operators, such as block insertion, block shifting, block searching in terms of the gaps and different types of block crossover have been tried. Most of those operators have improved the alignment scores but in terms Q and TC assessment they have failed. Finding the highest alignment score of a multiple protein alignment is an open field of research that is evolving rapidly. We have used a simple idea of evolutionary optimization and a genetic algorithm model to start the initial population of alignments as various MSA program outputs and then let the alignments evolve. Eventually we have obtained significant enhancement of alignment in terms of the Q and the TC scores in comparison to the individual MSA methods. We have implemented our model in Perl language. But later it has been rewritten in C language because the execution time taken by C language is much less. It is very interesting to note that this HGA alignment program structure is such that the program running time reduces by a factor of about 10 when the codes are written in C language instead of using Perl.

It has been reported that 11 – 19% of the core residues are misaligned by the structural alignment programs (29). And majority of the benchmark alignments are obtained by using the structural alignment programs. So there is a concern over the benchmark alignments. In that case, we suspect that our HGA alignment program will still provide better alignment accuracy. There is an interesting review on the MSA methods and applications (30). In another method, the pair-wise sequence alignment method was used for benchmark study by calculating the cluster validity score (31).

It is a fact that a genetic algorithm program takes more run time compared to conventional algorithms because of its stochastic nature. The traditional genetic algorithm approach towards sequence alignment like SAGA tends to build alignment from the initial sequences. But in our current approach, initial alignment solutions are near to global optimum as they are the outputs of other important programs. So it takes very less time compared to the conventional GA programs as shown in table (Table 5, 7). This proves that the GA can be used as an excellent optimizer. In finding a better sequence alignment, generally it is a tough task to choose the right MSA program over several programs available in the literature. So the present HGA is a better alternative to combine the individual methods and further improve them to find still better alignment. Furthermore, the HGA is quite robust with respect to the evolution of novel individual methods. The HGA has incorporated biological knowledge such as structure based derived matrix and some novel GA operators and has shown to generate good results even below the twilight zone of sequence similarity.

The ProbCons program mostly gives outputs of protein alignments whose aligned lengths are statistically more than the corresponding reference benchmark alignments. Therefore, the mutation operator must have gap elimination and gap shifting mechanism to get better alignment if used in the HGA like schemes (where gap insertion in hill climbing manner is an important factor). The alignment outputs of the HGA program are statistically comparable to the output of ProbCons method in alignment accuracy when applied to the Homstrad dataset (result not shown). We are working on a model called as ProbCons Genetic Algorithm (PGA) that uses the initial population of alignments as outputs of ProbCons program and then optimizes by the GA operators that uses gap elimination and gap shifting as mutation operators. We are hoping to obtain better alignment accuracy than that of the ProbCons. These results could be more applicable to protein families with low sequence similarity.

Another direction of research would be to combine the principles of Hidden Markov Model (HMM) and GA to create a hybrid model. The idea is to use the HMM solutions of sequence alignment and then incorporate the appropriate GA optimizing operators to create better alignment. The more accurate protein sequence alignment algorithms will help in predicting protein structures more accurately and, in turn, will be useful in the drug discovery process in the biotechnology industries.

ACKNOWLEDGEMENT

Funding from Department of Information Technology, Government of India (DIT R&D)/BIO/15(5) /2006 to NB) is acknowledged.

REFERENCES

1. Wang,L. and Jiang,T. (1994) On the complexity of multiple sequence alignment. *J. comput. Biol*., **1**, 337-348.
2. Thompson,J.D., Plewniak,F., Ripp,R., Thierry,J.C. and Poch,O. (2001) Towards a reliable objective function for multiple sequence alignments. *J. Mol. Biol.*, **314**, 937-951
3. Lipman,D.J., Altschul,S.F. and Kececioglu,J.D. (1989) A tool for multiple sequence alignment. *Proc. Natl Acad. Sci.*, **86**, 4412-4415.
4. Hogeweg,P. and Hesper,B. (1984) The alignment of sets of sequences and the construction of phylogenetic trees. An integrated method. *J. Mol. Evol.*, **20**, 175-186**.**
5. Simossis,V.A. and Heringa,J. (2005) PRALINE: a multiple sequence alignment toolbox that integrates homology-extended and secondary structure information. *Nucleic Acids Res*., **33**, W289–W294.
6. Heringa,J. (1999) Two strategies for sequence comparison: profile-preprocessed and secondary structure-induced multiple alignment. *Comput. Chem*., **23**, 341–364.
7. Zhang,Z., Lindstam,M., Unge,J., Peterson,C. and Lu,G. (2003) Potential for dramatic improvement in sequence alignment against structures of remote homologous proteins by extracting structural information from multiple structure alignment. *J. Mol. Biol*., **332**, 127–142.
8. Ren,T., Veeramalai,M., Tan,A.C. and Gilbert,D. (2004) MSAT: a multiple sequence alignment tool based on TOPS. *Appl. Bioinformatics*., **3**, 149–158.
9. Kleinjung,J., Romein,J., Lin,K. and Heringa,J. (2004) Contact-based sequence alignment. *Nucleic Acids Res.*, **32**, 2464–2473.
10. O'Sullivan,O., Suhre,K., Abergel,C., Higgins,D.G. and Notredame,C. (2004) 3DCoffee: combining protein sequences and structures within multiple sequence alignments. *J. Mol. Biol*., **340**, 385-395.
11. Prlic,A., Domingues,F.S. and Sippl,M.J. (2000) Structure-derived substitution matrices for alignment of distantly related sequences. *Protein Eng*., **13,** 545-550.
12. Thompson,J.D., Higgins,D.G. and Gibson,T.J. (1994) CLUSTALW: 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.
13. Wallace,I.M., O'Sullivan,O., Higgins,D.G. and Notredame,C. (2006) M-Coffee: combining multiple sequence alignment methods with T-Coffee. *Nucleic Acids Res*., **34,** 1692-1699.
14. Notredame,C., Higgins,D.G. and Heringa,J. (2000) T-Coffee: a novel method for fast and accurate multiple sequence alignment. *J. Mol. Biol*., **302**, 205-217.
15. Katoh, K., Misasa, K., Kuma, K. and Miyata,T. (2002) MAFFT: A novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic Acids Res*., **30**, 3059-3066.
16. Edgar,R.C. (2004) MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res*., **32**, 1792-1797.
17. Do,C.B., Mahabhashyam,M.S., Brudno,M., Batzoglou,S. (2005) ProbCons: Probabilistic consistency-

based multiple sequence alignment. *Genome Res*., **15**, 330-340.

18. Thompson,J.D., Koel,P., Ripp,R. and Poch,O. (2005) Balibase 3.0: Latest Developments of the Multiple Sequence alignment Benchmark. *Proteins*., **61,** 127-136.
19. Mizuguchi,K., Deane,C.M., Blundell,T.L. and Overington,J.P. (1998) HOMSTRAD: a database of protein structure alignments for homologous families. *Protein Sci*., **7**, 2469–2471.
20. Holland, J.H. (1975) Adaptation in natural and artificial systems. Univ of Michigan press, Ann Arbor, MI.
21. Zhang, C and Wong A.K. (1997)A Genetic algorithm for multiple molecular sequence alignment. CABIOS,13(6): 565-581.
22. Cai, L, Juedes, D and Liakhovitch, E. (2000) Evolutionary computation techniques for multiple sequence alignment. Proceedings of the second congress on evolutionary computation. 829-835.
23. Anbarasu, L. A., Narayanasamy and Sundararajan, V. (2000) Current Science Vol. 78 858-863.
24. Thomsen, R., Fogel, G.B and Krink, T. A (2002) Clustal alignment improver using evolutionary algorithms. Proceedings of the fourth congress on evolutionary computation 1 121-126.
25. Vingron, M. and Sibbald, P. (1993) Weighting in sequence space: a comparison of methods in terms of generalized sequences. *Proc. Natl Acad. Sci.*, **90** 8777−8781.
26. Altschul, S. F., Carroll, R. J., and Lipman, D. J. (1989) Weights for data related by a tree. *J. Mol. Biol*., **207**, 647 − 653.
27. Edgar, R.C. (2004) MUSCLE: a multiple sequence alignment method with reduced time and space complexity. *BMC Bioinformatics*., **5**, 113.
28. Notredame, C., Higgins, D.G. (1996) SAGA: sequence alignment by genetic algorithm. *Nucleic Acids Res*., **24**, 1515-1524.
29. Kim,C., Lee,B. (2007) Accuracy of structure-based sequence alignments of automatic methods. *BMC Bioinformatics*., **20**, 8355.
30. Chatzou, M et al (2016) Multiple sequence alignment modeling: methods and applications. *Briefings in Bioinformatics*, 17, 1009.
31. Wang Y et al (2018) A benchmark study of sequence alignment methods for protein clustering. *BMC Bioinformatics*, 19, 529

Figure legends

Figure 1: pictorial representation of un-rooted tree.

Figure 2: shows a pictorial representation of roulette wheel selection.

Figure 3: describes the operation of point crossover between two alignments that produce two recombinant alignments. The arrow indicates the way the parental alignments are spliced.

Figure 4: depicts the way the Block crossover recombinants are produced by swapping blocks between two parental alignments.

Figure 5: Pseudocode of the HGA.

Table legends

Table 1: Average of Q and TC on the Homstrad benchmark

The columns represent the average of Q and TC scores for all the alignments. The Ranksum values are obtained from the Friedman test for all the alignments. A higher rank sum represents better alignment accuracy. The highest score on each benchmark is highlighted in bold. All the scores have been multiplied by 100.

Table 2: Comparison of Q and TC on the Homstrad benchmark in terms of percentage

The first and second columns represent the increase of alignments in terms of Q and TC scores respectively. The performance of the HGA model against other programs is shown.

Table 3: Statistical analysis on Homstrad benchmark

Each value in the table contains the P-value assigned by the Friedman rank test, indicating the significance of difference of alignments between the programs. The upper triangle of the matrix values are derived from the TC scores on the Homstrad. The signs + and - represent that a program for a row performs significantly better and worse respectively than that of a column program. If the P-value is greater than 0.05, the difference is not significant and is shown in parentheses. For example, the HGA ranks higher than the ClustalW with a P-value of 5.2×10^{-6}.

Table 4: Average of Q and TC scores on Homstrad benchmark using the sequence similarity based matrix (Blosum62)

The **c**olumn represents the average of TC scores for all the alignments. The Ranksum values are obtained from the Friedman test for all the alignments on the Homstrad benchmark. See Table 1 for comparison when structure based matrix is used.

Table 5: Performance of alignment program on HOMSTRAD protein reference alignment benchmark.

Entries reveals the running time for programs over the entire HOMSTRAD protein reference alignment benchmark for each program in hours, minutes and seconds. Shows the average of TC achieved on HOMSTRAD benchmark. All scores have been multiplied by 100. The best in each column is shown in bold.

Table 6: Comparison of sum of pair score across different programs on HOMSTRAD protein reference alignment benchmark.

Entries show the average of sum of pair score on HOMSTRAD protein reference alignment benchmark. The best result is shown in bold.

Table 7: Comparison of programs SAGA and HGA in terms of Q and TC score on Homstrad and Balibase protein reference alignment.

Entries show the average Q and TC score attained by SAGA and HGA program on Homstrad and Balibase database. All scores have been multiplied by 100. Running times for programs are given in hours, minutes and seconds. The best results between the rows are shown in bold. Result of SAGA on other sets of Balibase benchmark set is not shown due to fact that SAGA alignment program took enormous amount of time.

Table 8: Average of Q on the Balibase benchmark

The column represents the average of Q score for all the alignments. The Ranksum values are obtained from the Friedman test for all the alignments. The highest score in each benchmark set is highlighted in bold.

Table 9: Average of TC scores on the Balibase benchmark

The column represents the average of TC score for all the alignments. The Ranksum values are obtained from the Friedman test for all the alignments. The highest score in each benchmark set is highlighted in bold.

Table 10: Comparison of TC score on the Balibase in terms of Percentage

The column represents the increase of alignment in terms of the TC score. The HGA model is compared against all the programs with respect to each group.

Table 11: Statistical analysis on the Balibase benchmark

Each value in the table contains the P-value assigned by the Friedman rank test, indicating the significance of difference of alignments between the programs. The upper triangle of the matrix values are derived from the TC scores on the Balibase. The signs + and - represent that a program in a row performs significantly better and worse respectively than that of a program in a column. If the P-value is greter than 0.05, the difference is not significant and is shown in parentheses. For example, the HGA ranks higher than the ClustalW with a P-value of 2.2×10^{-16}.

Figures

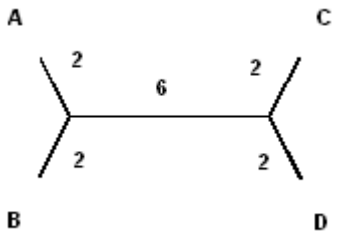

Figure 1

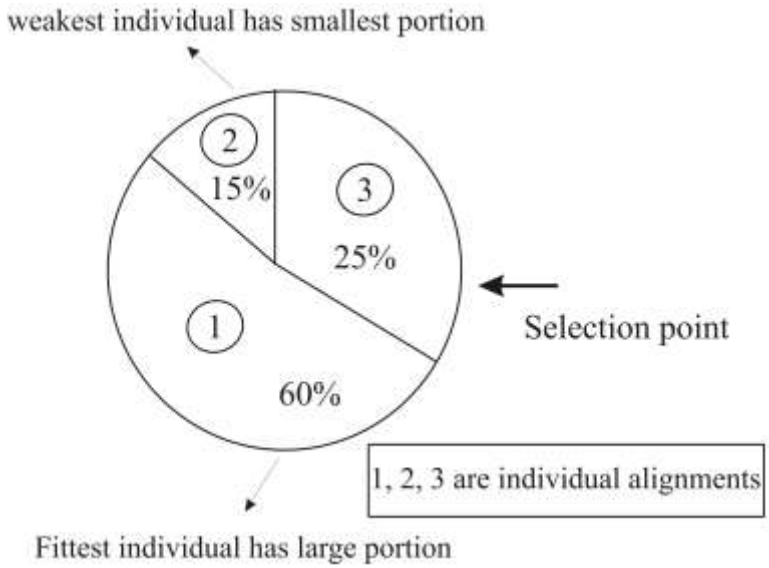

Figure 2

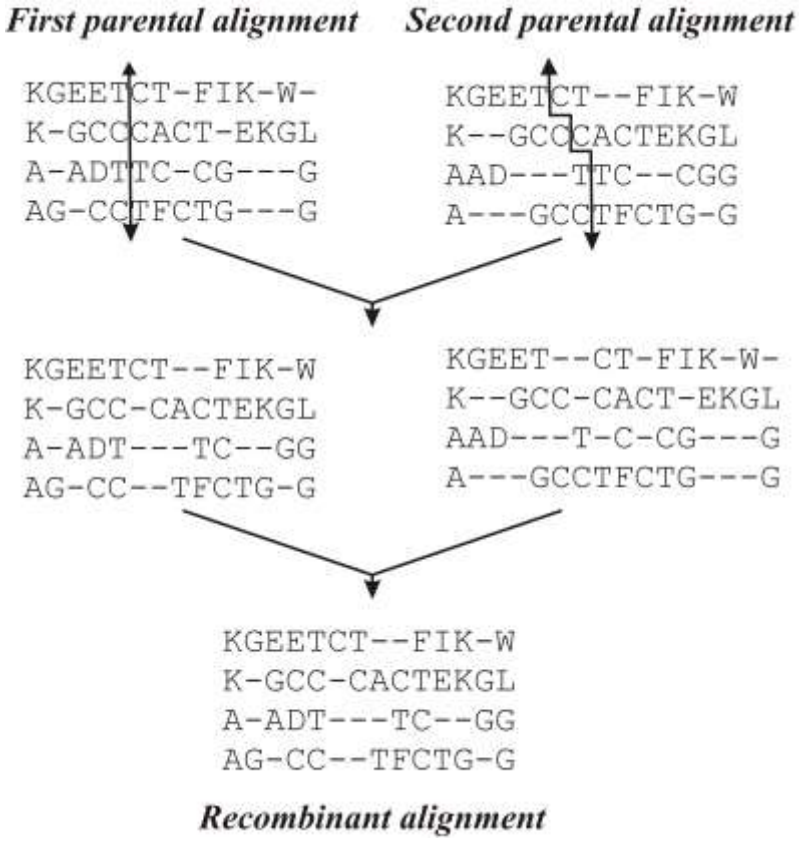

Figure 3

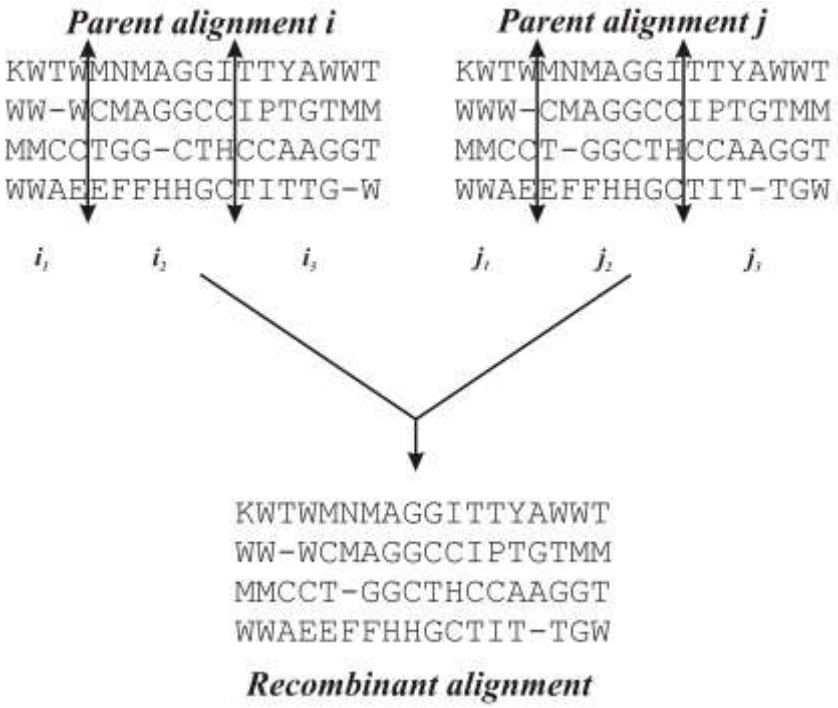

Figure 4

Schema HGA

Start

Initialization of the population

Assessment of the population

Loop

Crossover operator

Elitism selection

Mutation operator

Assessment of the population

End loop (if termination condition reached)

End

Figure 5

Tables

Table 1:

Methods	Q	TC	Ranksum of Q	Ranksum of TC
HGA	**82.28**	**67.71**	**858.5**	**874**
ClustalW	79.43	63.80	635.5	673
Mafft	79.19	62.81	535.5	557.5
Muscle	81.51	66.14	764.5	748
TCoffee	80.65	64.26	701	642.5

Table 2:

	Q	TC	Methods
HGA	3.5	6.12	**ClustalW**
HGA	3.9	7.23	**Mafft**
HGA	0.94	2.37	**Muscle**
HGA	2	5.36	**TCoffee**

Table 3:

Methods	HGA	ClustalW	Mafft	Muscle	T Coffee
HGA		$+5.2 \times 10^{-6}$	$+2.2 \times 10^{-16}$	$+2.3 \times 10^{-7}$	$+1.5 \times 10^{-9}$
ClustalW	-2.4×10^{-8}		$+7.2 \times 10^{-3}$	-2.2×10^{-2}	$+6.3 \times 10^{-1}$
Mafft	$-< 2.2 \times 10^{-16}$	-2.7×10^{-2}		-2.0×10^{-8}	-1.0×10^{-2}
Muscle	-1.4×10^{-4}	$+9.8 \times 10^{-4}$	$+7.0 \times 10^{-11}$		$+1.7 \times 10^{-4}$
TCoffee	-3.0×10^{-3}	$+4.5 \times 10^{-2}$	$+1.9 \times 10^{-4}$	$- 1.0 \times 10^{-2}$	

Table 4:

Methods	Q	TC	Ranksum of Q	Ranksum of TC
HGA	81.17	**66.8**	**803.5**	**813.5**
ClustalW	79.43	63.8	643.5	683
Mafft	79.19	62.81	543	564
Muscle	**81.51**	66.14	796	779.5
TCoffee	80.65	64.26	709	655

Table 5:

Program	Time	TC
Clustalw	1 Min, 22 sec	63.80
Mafft	**49 sec**	62.81
Muscle	1 Min, 25 sec	66.14
TCoffee	14 Min, 18 sec	64.26
HGA	50 Min, 16 sec	**67.71**
SAGA	29 Hrs, 21 Min	57.19

Table 6:

Program	Sum of pair score
Clustalw	5842.99
Mafft	5936.65
Muscle	6076.91
TCoffee	5740.17
HGA	**6198.53**
SAGA	5329.88

Table 7:

	Q	TC	Time Taken	Benchmark system
SAGA	74.20	57.19	29 hrs 5 sec	**Homstrad**
HGA	**82.24**	**67.71**	**50 min 5 sec**	**Homstrad**
SAGA	33.96	19.63	9 hrs 50 min	**Balibase RV11(76 alignments)**
HGA	**53.84**	**30.67**	**26 Min 41 sec**	**Balibase RV11(76 alignments)**
SAGA	70.42	48.61	55 hrs 10 sec	**Balibase RV12(88 alignments)**
HGA	**84.17**	**64.51**	**74 Min 50 sec**	**Balibase RV12(88 alignments)**

Table 8:

Methods	Ref 1.1 (76)	Ref 1.2 (88)	Ref 2 (82)	Ref 3 (60)	Ref 4 (49)	Ref 5 (31)	Overall (386)	Ranksum
HGA	**53.84**	**84.17**	82.98	**73.14**	**67.53**	70.43	**72.02**	**1452**
ClustalW	46.82	79.64	79.70	65.86	61.73	63.17	66.15	715

Mafft	47.08	80.58	81.77	72.04	65.22	68.91	69.26	948.5
Muscle	53.23	83.31	82.99	72.16	66.80	69.61	71.35	1321.5
TCoffee	50.08	83.94	**83.72**	70.24	66.68	**70.46**	70.85	1353

Table 9:

Methods	**Ref 1.1 (76)**	**Ref 1.2 (88)**	**Ref 2 (82)**	**Ref 3 (60)**	**Ref 4 (49)**	**Ref 5 (31)**	**Overall (386)**	**Ranksum**
HGA	**30.67**	**64.51**	**30.02**	**34.52**	**20.76**	**29.92**	**35.06**	**1459**
ClustalW	23.11	58.70	26.03	29.48	18.58	25.07	30.16	955.5
Mafft	23.46	57.25	28.53	32.48	18.68	27.44	31.30	980.5
Muscle	30.07	62.36	28.87	32.87	20.37	28.64	33.86	1215.5
TCoffee	25.89	62	28.23	27.88	19.02	23.83	31.14	1179.5

Table 10:

	RV11	**RV12**	**RV20**	**RV30**	**RV40**	**RV50**	**Over all**	**Methods**
HGA	32.71	9.9	15.32	17.09	11.73	19.34	16.24	**ClustalW**
HGA	30.73	12.68	5.22	6.28	11.13	9.03	12.01	**Mafft**
HGA	1.99	3.44	3.98	5.02	1.91	4.47	3.54	**Muscle**
HGA	18.46	4.04	6.34	23.81	9.14	9.96	12.58	**TCoffee**

Table 11:

Methods	**HGA**	**ClustalW**	**Mafft**	**Muscle**	**T Coffee**
HGA		$+<2.2 \times 10^{-16}$	$+<2.2 \times 10^{-16}$	$+2.8 \times 10^{-14}$	$+2.0 \times 10^{-7}$
ClustalW	$-<2.2 \times 10^{-16}$		(0.60)	-2.1×10^{-10}	-2.4×10^{-6}
Mafft	$-<2.2 \times 10^{-16}$	$+6.6 \times 10^{-9}$		-9.0×10^{-10}	-3.0×10^{-6}
Muscle	-2.2×10^{-4}	$+<2.2 \times 10^{-16}$	$+6.5 \times 10^{-15}$		(0.34)
TCoffee	(0.40)	$+<2.2 \times 10^{-16}$	$+3.1 \times 10^{-13}$	(0.44)	