


Improved Genetic Algorithm for Multiple Sequence Alignment Using Segment Profiles (GASP)

View metadata, citation and similar papers at core.ac.uk

brought to you by  CORE

provided by Xiamen University Institutional Repository

Yanping Lv¹, Shaozi Li¹, Changie Zhou¹, Wenzhong Guo², and Zhengming Xu¹

¹Intelligent Information Technology Lab., Department of Computer Science, Xiamen University, Xiamen, 361005, China

Catlet.lyp@gmail.com, {szlig, dozero}@xmu.edu.cn

²Department of Computer Science, Fuzhou University, Fuzhou, 350002, China
guowenzhong@fzu.edu.cn

Abstract. This paper presents a novel genetic algorithm (GA) for multiple sequence alignment in protein analysis. The most significant improvement afforded by this algorithm results from its use of segment profiles to generate the diversified initial population and prevent the destruction of conserved regions by crossover and mutation operations. Segment profiles contain rich local information, thereby speeding up convergence. Secondly, it introduces the use of the norMD function in a genetic algorithm to measure multiple alignment. Finally, as an approach to the premature problem, an improved progressive method is used to optimize the highest-scoring individual of each new generation. The new algorithm is compared with the ClustalX and T-Coffee programs on several data cases from the BALiBASE benchmark alignment database. The experimental results show that it can yield better performance on data sets with long sequences, regardless of similarity.

1 Introduction

Multiple sequence alignment (MSA) has become an essential tool in molecular biology. It has been used for the analysis of protein families, comprehension of their evolutionary trends and detection of remote homologues, genome annotation and analysis and a host of other tasks. When sequences are similar to each other, virtually any alignment method will produce good results. However, evolutionary divergence in families can result in the pair similarity between family members being so low as to be indistinguishable from chance [1]. The development of accurate, reliable multiple alignment programs capable of handling large numbers of very divergent sequences, is therefore of major importance.

Unfortunately, accurate multiple alignments can be difficult to build. The optimization algorithms largely fall into two categories: progressive and iterative algorithms. In progressive methods, an MSA is built up gradually by aligning the closest se-

* Corresponding author.

This work is supported by the Natural Science Fund, Science & Technology Project of Fujian Province (Project Number: A0310009, 2001J005), China, the 985 Innovation Project on Information Technique of Xiamen University(2004-2007), China.

quences first and successively adding in the more distant ones. A typical program is ClustalX [2]. It constructs a global alignment over the entire length of the sequences. It has the advantages of speed and simplicity. However, due to its 'greediness', errors made in the first alignments cannot be rectified later as the rest of the sequences are added in.

Iterative strategies have been applied to refine and improve the initial alignment. DIALIGN [3] constructs multiple local alignments based on segment-to-segment comparisons. Other iterative algorithms aim at building global alignments, two examples are SAGA [4], based on a genetic algorithm, and HMM [5]. For low-identity (low-similarity) sequences, DIALIGN will produce low quality MSAs due to its local nature. HMM does not correctly align structurally similar regions existing in some, but not all, sequences.

SAGA has been demonstrated to obtain better MSAs than other programs for divergent sequences [1]. It succeeds in aligning critical motifs and conserved core structure of protein families. However, the length and size of sequences it can handle is restricted due to its limit speed and it may sometimes tend to diverge away from the correct alignment in the presence of an 'orphan' sequence aligned to a family of closely related sequences, as in ref2 of the BALiBASE database.

The paper is organized as follows. Section 2 proposes an improved genetic algorithm for multiple sequence alignment. Results of experimental evaluation are given in Section 3, which contains the description of benchmark database used for comparison of algorithms, the experimental setting for each algorithm, and discussions about the results. Section 4 gives conclusions and future work.

2 The GASP Algorithm

We call our algorithm GASP, for alignment based on a genetic algorithm using segment profiles. The outline of the procedure follows.

2.1 Encoding and Initialization

For genetic algorithms, each individual in the population is a possible solution to the problem. Different encoding methods can be chosen for different problems. Here, each individual is an alignment, in SAGA. Intuitively, an alignment of the population is expressed as a string matrix consisting of characters from a given alphabet.

The challenge in initialization is to generate an diverse initial population. However, a diversified population simultaneously increases computational complexity. In existing GA-based methods, individuals in the initial population are constructed randomly. The lengths of initial alignments are bounded by a value. For highly similar sequences, it is reasonable to limit the number of gaps. For divergent sequences, however, it is likely to result in the optimal alignment being missed. In our algorithm, the diversified initial population is generated, centered on different SPs.

We have designed a simple and efficient method for finding SPs. Here, a SP is defined as a string set in which every string from every sequence is highly similar. The first step is to find all segment pairs of equal length within a finite position distance d , with sum-of-pairs score (using the PAM250 substitution matrix [6]) higher than a

threshold T_{sp} . Two similar segments always get a higher score, since PAM250 considers the similarity of residue pairs. Therefore, it is necessary to set this threshold. The position distance restricts the number of gaps.

Next we construct an SP whose segments are from different sequences. The number of segments in the SP must be greater than half of the sequence size and its norMD [7] score must be higher than a cutoff T_{md} . (The sum-of-pairs score is sensitive to the length and size of sequences, whereas the norMD score is not affected by these factors.) Finally, we extend the SP to both sides until its norMD score is less than the cutoff mentioned above, since when a SP corresponds to a structurally similar region of alignment, there is a high probability that there will be another SP located nearby.

To create one of these individuals, we randomly align two substring sets on both sides of an SP, then build up an MSA by integrating the two subalignments and the SP. As a result, each MSA is centered on a different SP. If the number of MSAs is less than the population size, the remaining individuals are randomly generated as in other GA-based methods. The final result is a diversified initial population with different individuals, most of which are centered on different SPs.

2.2 Fitness Function and Its Scaling

In this algorithm, norMD is introduced to measure the quality of an MSA. The goal of MSA is to align structurally similar regions of all sequences and to succeed in aligning regions that are structurally similar in some sequences. Sum-of-pairs can't reasonably evaluate the quality of an MSA, for it is sensitive to the length and size of sequences. NorMD was therefore suggested here for comparison purposes. Simulation experiments show that it is not sensitive to the factors mentioned above and delineates an MSA better, since it combines column scores with residue similarity scores.

Because the variance of the fitness value given by norMD is so low, a corresponding function scaling method has been used in this algorithm. The NorMD scores of most alignments obtained during the iterative procedure range between 0 and 1. This algorithm also calculated the expected offspring (EO) of an alignment on the basis of the fitness value.

$$EO_i = \frac{f_i}{\sum_j f_j / Num}$$

Here, f_i is the norMD score of the i^{th} individual and Num is the population size.

2.3 Operators

Selection: In this algorithm, an individual is selected as a parent simply based on the proportional probability of its EO.

$$P_i = EO_i / \sum_j EO_j$$

One-point crossover: The crossover can be very disruptive at the junction point. Positions in SPs are chosen as crossover sites on the basis of zero probability, to prevent destruction due to crossover. As a result, SPs as conserved regions in the initial population will be kept down until the iterative process terminates. If an SP is an excellent gene, a MSA which contains it will get a high norMD score. Otherwise, it will get a low score and be abandoned in a later generation. However, SPs also bring the problem of premature convergence. To overcome this problem, we optimize the highest-scoring MSA by rearranging it after crossover and mutation operations.

Mutation: Some positions are conserved more than others during the process of generation [8]. For this reason, we found it useful to bias the choice of the mutation site. In this algorithm, the positions in SPs are chosen for zero probability and other positions are selected as mutation sites for equal probability.

2.4 Rearrangement

A very stable local minimum makes it difficult for operators to generate an optimal MSA. To avoid being trapped in local minima resulting from SPs, we rearrange the highest-scoring MSA of every generation. During the iterative procedure, we extract all substrings from two adjacent SPs in the MSA, align them using a progressive method and incorporate all subalignments and SPs into it. We align pair sequences using an improved SPA [9] for proteins, if substrings are long or the sequence size is large; otherwise, traditional dynamic programming is used here. We simply need to rearrange one MSA. As a result, most of the conserved residues can be aligned in the same columns, without sacrificing too much time.

3 Experimental Results

3.1 Reference Alignments

In order to demonstrate the feasibility of our algorithm, we used version 3 of the BALiBASE benchmarks database [10]. BALiBASE is designed for the evaluation and comparison of MSA programs. The alignments in BALiBASE are divided into eight reference sets. Here, we used only the first two reference sets. Ref1 contains alignments of a small (<6) number of sequences which are equidistant, meaning that the percent identity between two sequences is within a specified range. Alignments in Ref2 combine three ‘orphan’ sequences (<25% identical) from ref1 with a family of at least 15 closely related sequences. Ref1 and Ref2 are divided into groups of short, medium and long sequences. For clarity of comparison, a single ‘orphan’ sequence is aligned to a family in ref2.

3.2 Alignment Quality Scoring

BALiBASE provides a module (BaliSore) that defines two scores. The sum-of-pairs score, SPS, is the ratio of the number of correctly aligned pairs of positions in the test alignment to the number of aligned pairs in the reference alignment structurally informed. The column score, CS, is the ratio of the number of correctly aligned columns in the test alignment to the number of aligned columns in the reference alignment.

Both SPS and CS range from 0.0 for no agreement to 1.0 for perfect agreement. The designers recommend SPS as the best quality score for Ref1,2,3.

While the BALiBASE scores are useful, they have limitations as measures of alignment quality. On the one hand, they only take core blocks into account and give no credit for positions between core blocks. Neither of them penalizes columns between core blocks in the test alignment that are not structurally aligned. On the other hand, neither of them measures the alignments excluded from BALiBASE benchmarks. (In Ref2, we only align one orphan to a family.) As a complementary measure of alignment quality, we also evaluate alignments using the norMD measure, where both each residue pair and each column are compared between the two alignments.

3.3 Algorithm Parameters

GASP has the following parameters: Num , the population size; P_c , the probability of crossover; P_m , the probability of mutation. Three additional parameters are d , the maximal position distance between segment pairs; T_{sp} , the sum-of-pair score threshold and T_{md} , the NorMD score cutoff. The parameter d involves the size of the alignment search space. If the d value is too small, a segment profile containing rich local sequence information cannot be found. If it is too large, too many gaps must be inserted into an alignment centered on the SP found and more time is required to find the optimal alignment. The value of d used in these experiments depends on the variance in the length of the sequences. For sequences of similar length, it is set to one-quarter the sequence length, to avoid having to insert too many gaps into an optimal alignment; otherwise, the proportion is one-half. T_{sp} and T_{md} are obtained empirically. For most experiments, good results can be produced with $T_{sp}=0$ and $T_{md}=0.5$. P_c and P_m are two main parameters in GAs. We performed some experiments to find the optimal values of these parameters. In our experiment, we empirically chose 15 values of P_c and 20 values of P_m . For each parameter composition, we ran the program 30 times and the average SPS (ASPS) of the results was obtained. We first determined the optimal P_c value as follows. For each of the 15 values, we averaged all ASPS values with different P_m values. We selected the optimal result of the 15 results and subsequently the optimal P_c could be chosen. Good performance is obtained with $Num=91$ (more individuals increases the computer load without reducing the number of algorithm iterations before convergence), and with $P_c=0.7$ and $P_m=0.065$. With these parameters, the iteration terminated at a point beyond which no better solution would usually be found.

3.4 Experiments

Using the parameters set above, GASP constructed the alignments by extracting sequences from the BALiBASE reference alignments, (60 out of 123 cases, 45 in ref1 and 15 in ref2). For the alignments selected, we also downloaded ClustalX and, T-Coffee [11] for comparison. ClustalX is one of the most commonly used tools; we used version 1.83. T-Coffee is one of the most recent tools, which generates a MSA faster and sacrifices less accuracy than SAGA, which runs too slowly for long sequences. Figures 1 and 2 show the SPS and NorMD scores, respectively, for DASP, ClustalX, and T-Coffee on benchmarks with low, medium, and high similarity.

From Figure 1, when medium or long sequences are considered regardless of similarity, GASP outperforms other tools. For medium benchmarks, on average, GASP gets a score of 80.3%, which is better than 79.1% for T-Coffee and 76.2% for ClustalX. GASP finds the best results for 11 out of 15 medium reference benchmarks. For long benchmarks, GASP is again superior to the other two tools. Its average score of 85.5% is the highest of the three, and it performs best in 13 cases out of 15. On short sequences, however, it gets an average accuracy of 73.5%, worse than 84.7% for T-Coffee and 79.2% for ClustalX, since it constructs the initial population with few alignments centered on segment profiles, ultimately resulting in premature convergence. In conclusion, for medium and long benchmarks regardless of similarity, our method performs best.

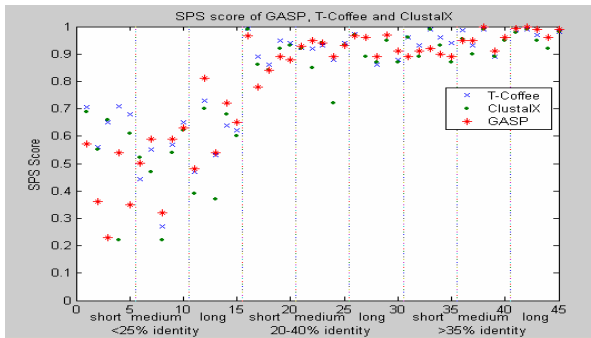


Fig. 1. The SPS scores of GASP and the other tools. Here, the five test cases are chosen from ref1 with varied percent identity and varied length of equidistant sequences.

Figure 2 shows the norMD scores of GASP and the two other programs on ref2 alignments. Neither of the BALiBASE scores measure the alignments excluded from the BALiBASE benchmarks. Here, for clearer comparison, we align only one ‘orphan’

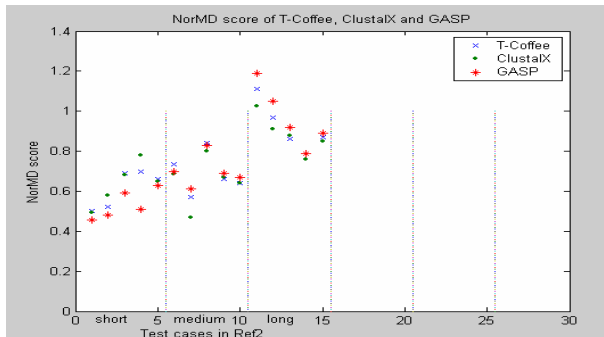


Fig. 2. The norMD scores of GASP and the other tools on test cases. Here, each of the five test sets is chosen from ref2 with varied length of sequences but only one, rather than three ‘orphan’ sequences, is aligned to a family of at least 15 closely related sequences on ref2 alignments.

sequence to a family of closely related sequences extracted from ref2 benchmarks. Figure 2 shows that, on average, GASP has comparable norMD scores to the other two programs for medium and long sequences but still obtains the worst norMD score for short sequences. However, the superiority is more pronounced for medium and long sequences, and the difference less for short ones, using the norMD measure. This means there are few gaps inserted into columns between core blocks in GASP, since norMD also considers the columns between core blocks.

Figure 3 shows one subalignment of GASP and the corresponding one for T-Coffee. Here, an ‘orphan’ sequence, *lgowA*, is aligned to a family of 15 closely related sequences. In this alignment, 8 of 16 aligned subsequences are laid out below. In GASP, an alpha helix from each sequence is aligned in common columns, for it was first found as a segment profile and then kept down. The corresponding alpha helix diverges by some gaps in T-Coffee.

```

      2myr .FESDGGDGSSNIYYYPKGIYSVMDYFKNKYYN----PLIYVTENGISTPG---:
bgl2 trirp ----PRAASIWIYVYPYMFIQEDFEIFCYILKINITILQFSITENGMNEFNDAT
bgl2_maize ----PPMGNPWYMYPEGLKDLLMIMKKNYGN----PPIYITENGIGDVDTKE
bgl2_bacsu -PHLITSNWDW-TIDPIGLRIGLRRITSRYQ-----LPVFITENGLGEFDK--
lacg_staau t-VDVPRTDWDW-MIYPQGLYDQIMRVVKDY---PNYHKIYITENGLGYKDEFI
lacg_lacac PDGIETTDWDW-LIYPQGLYDQIMRVKNDY----PNIHKVYITENGLGFKDIVP
lacg_lacca .PDGIETTDWDW-SIYPRGMYDILMRIHNDY----PLVPVYVTENGLGLKESLP
lgowA PTSDFG----WEFF-PEGLYDVLTKYWNRYH---L--YMYVTENGIADDAD--
--PLF-ESDGGDGSSNIYYYP---KGIYSVMDYF-KNKYYN-PLIYVTENGISTP
-----PRAASIWIYVYPYMFIQEDFEIFCYILKINITI-LQFSITENGMNEF
-----PPMGNPWYMYP---EGLKDLLMIM-KNKYGN-PPIYITENGIGDV
KTKKN-PHLITSNWDW-TIDP----IGLRIGLRRI-TSRYQ---LPVFITENGLGEF
QREFD-VDVPRTDWDW-MIYP---QGLYDQIMRV-VKDYPNYHKIYITENGLGYK
EEKLP-DGIETTDWDW-LIYP---QGLYDQIMRV-KNDYPNIHKVYITENGLGFK
EEKLP-DGIETTDWDW-SIYP---RGMYDILMRI-HNDYPLVPVYVTENGLIGLK
-NSVSLAGLPTSDFGW-EFFP---EGLYDVLTKY-WNRYH---LYMYVTENGIADD

```

Fig. 3. A subalignment of GASP and the corresponding one for T-Coffee. Here, the red region is a secondary structure (an alpha helix); the green, a beta strand.

The average running time of GASP and the other programs for long sequences is calculated in Figure 4. Here, time is measured in milliseconds and short sequences are not taken into consideration, for the alignments constructed by GASP are less accurate than those yielded by the other two. GASP is not suitable for building alignments for short sequences. In Figure 4, we conclude that ClustalX performs best for long sequences. However, ClustalX achieves this at the expense of low accuracy. Our algorithm is slower than ClustalX, but faster than T-Coffee.

Test case	GASP	ClustalX	T-Coffee
Ref1 long<25% identity	1881	752	3109
Ref1 long<20-40% identity	2263	869	4187
Ref1 long >35% identity	2902	915	4984
Ref2 long	3859	2073	6735

Fig. 4. Average running time of GASP and the other tools for long sequences

4 Conclusion and Future Work

GASP was developed to structurally align similar regions of multiple long proteins. GASP is based on a genetic algorithm, but differs from existing GA-based multiple sequence alignment methods in that it builds up the initial population by SPs. It first constructs the initial population in which the most individuals are centered on different SPs, then keeps SPs down, finally rearranges the highest-scoring individual of each new generation to avoid being trapped in local minima. The experimental results show that GASP achieves high accuracy and still maintains a competitive running time. For medium and long sequences, GASP yields the best result with appropriate parameters and the running time of GASP is comparable to that of representative tools. For short sequences, GASP can be improved by incorporating other computational methods during the iterative procedure.

References

1. Thompson, J.D., Plewniak, F.: A comprehensive comparison of multiple sequence alignment programs. *Nuc. Acids. Res.*, 1999, 27:2682–2690.
2. Thompson, J.D., Gibson, T.J.: The CLUSTAL_X windows interface: flexible strategies for MSA aided by quality analysis tools. *Nuc. Acids. Res.*, 1997, 25(24):4876-82.
3. Brudno, M., Chapman, M.: Fast and sensitive multiple alignment of large genomic sequences. *Bioinformatics*, 2003, 4:66.
4. Notredame, C., Higgins, D.G.: SAGA: sequence alignment by genetic algorithm. *Nuc. Acids. Res.*, 1996, 24:1515-1524.
5. Eddy, R.: *Biological Sequence Analysis: Probabilistic models of proteins and nucleic acids*. Cambridge University Press, 1998, pp: 51-68.
6. Dayhoff, M., Schwartz, R.M.: A model of evolutionary change in proteins. *Atlas of Protein Sequence and Structure*, 1978, 5:345–352.
7. Thompson, J.D., Plewniak, F.: Multiple Sequence Alignment Objective Function. *J. Mol. Biol.*, 2001, 314(4):937-951.
8. Benner, S.A., Cohen, M.A.: Amino acid substitution during functionally constrained divergent evolution of protein sequences. *Protein Eng.*, 1994, 7:1323–1332.
9. Shiyi, Shen., Jun, Yang.: Super Pairwise Alignment (SPA): An Efficient Approach to Global Alignment for Homologous Sequences. *J. Com. Biol.*, 2002, 9(3):477-486.
10. Thompson, J.D.: BALiBASE: A benchmark alignment database for the evaluation of multiple alignment programs. *Bioinformatics*, 1999, 15:87-88.
11. Notredame, C., Higgins, D., Heringa, J.: T-Coffee: A novel method for multiple sequence alignments. *J. Mol. Biol.* 2000, 302:205-217.