The First International Symposium on Optimization and Systems Biology (OSB'07)
Beijing, China, August 8–10, 2007
, pp. 309–323

# Two Phase Evolutionary Method for Multiple Sequence Alignments

Farhana Naznin[1,*] Morikazu Nakamura[1,†]
Takeo Okazaki[1,‡] Yumiko Nakajima[2,§]

[1] Department of Information Engineering, University of the Ryukyus, Okinawa, Japan
[2] Center of Molecular Biosciences, University of the Ryukyus, Okinawa, Japan

**Abstract** This paper presents a new evolutionary method, namely, a Two Phase evolutionary algorithm for multiple sequence alignments. This method is composed of different types of evolutionary algorithms, that is, an evolutionary progressive multiple sequence alignment method (abbreviated to ET) and Sequence Alignment by Genetic Algorithm (abbreviated to SAGA). The former is employed to obtain efficiently good quality of multiple alignments and the latter to improve them much better. The basic idea was obtained from analysis of the characteristics of the two evolutionary methods, that is, the ET and SAGA can compensate each other's weak points. Experimental evaluation shows that the proposed Two Phase method can generate good quality of multiple alignments.

## 1 Introduction

The multiple alignment of nucleotide or amino acid sequences is an essential problem since multiple alignments are used to detect homology between new sequences and existing sequences [1]. The rate of appearance of new sequence data is continuously increasing. Therefore, the development of efficient and accurate methods for multiple alignments is always required. The majority of multiple alignment algorithms is based on the 'progressive' approach of Feng and Doolittle [2] or its variations [3–5].

One major well known problem with the progressive approach is trapping at local minimums. This problem is due to the 'greedy' nature in constructing the guide tree. There is no guarantee that good quality of solutions will always be found. That is, any wrong choice made at early stage in the alignment process cannot be corrected later. The only way to correct this is to use an iterative or stochastic procedure [6-8]. SAGA (Sequence Alignment by Genetic Algorithm) [9] is one of the stochastic procedure, which is developed based on GAs [10] and it is capable of finding good multiple alignment. And it needs high computation time to get good results. We have

*Email: fnrbitr@yahoo.com
†Corresponding author, Email: Morikazu@ie.u-ryukyus.ac.jp
‡Email: okazaki@ie.u-ryukyu.ac.jp
§Email: yumiko28@comb.u-ryukyu.ac.jp

developed another evolutionary approach, called evolutionary tree-base method (ET). Computer Experiment was performed for some datasets obtained from BALiBASE (Benchmark Alignment dataBASE) [11] and showed that our ET method was superior to well-known methods such as SAGA, T-Coffee [12], MUSCLE [13], MAFFT [14], and ProbCons [15].

The objective of this research is to improve furthermore the accuracy of the ET method. To achieve this goal, we need to overcome the weak points of the ET method. That is, in the ET there is no one-one correspondence between the genotype and phenotype. Therefore, lots of alignments (phenotype) can not be generated from any chromosome (genotype). The other weak point is its fast convergence at local minimum solutions.

By overcoming the weak points, in this paper, we propose a Two Phase evolutionary multiple alignment based on the ET and SAGA in which the application of SAGA as the second phase searching redeems the weak points of the ET method used in the first phase. Experimental evaluation shows our achievement by comparing with SAGA, T-Coffee, MUSCLE, MAFFT, ProbCons and the ET.

# 2 Evolutionary Multiple Alignments

Here we explain two kinds of evolutionary multiple alignment approaches used in the new method; SAGA (Sequence Alignment by using Genetic Algorithm) and the ET (Evolutionary Tree-base) method. The former was developed in [9] and it is the frontier research of evolutionary computation for the multiple alignment problems. In SAGA, chromosomes represent directly multiple alignments and the genetic operators are performed directly on the alignments. That is, the phenotype and genotype are the same, the one-one correspondence. Therefore, flexible searching of alignments is possible since all the alignments can be represented as a chromosome, however, it is not so efficient to find good quality solutions. On the other hand, in the ET, chromosomes correspond to guide trees which are transformed into multiple alignments by progressive alignment and genetic operators are carried out on guide trees. That is, there is no one-one correspondence between the phenotype and genotype. Lots of multiple alignments can not be generated from a chromosome (guide tree). However, the ET can find very efficiently good quality of alignments since the searching space is drastically reduced by the guide tree representation of the genotype.

## 2.1 Sequence Alignment by using Genetic Algorithm (SAGA)

SAGA, which is developed based on the simple genetic algorithm, is starting from completely unaligned sequences. In SAGA, the initial population is randomly generated. In order to generate new population, evaluation, selection and genetic operators (crossover or a mutation) are used.

## Genetic Operators

Two types of operators are used in SAGA, one is crossover and the other is mutation. Two types of crossover are also implemented in SAGA: one-point and uniform crossover.

*(i) One point crossover:* Figure 1 shows the one point crossover between two selected parents [9]. In Fig.1, the arrow position on Parent Alignment 1 is randomly selected, which is cut at that position. The left part of the cut position has four residues. Parent Alignment 2 is cut such that the left part of the cut position must have four residues. Before connecting the discarded parts of both parents, the left and right parts of Parent Alignment 2 are made equal length by inserting null signs. After that the right part of Parent Alignment 1 and the left part of Parent Alignment 2 are combined to generate Child Alignment 1. Child Alignment 2 is produced by combining the left part of Parent 1 and the right part of Parent 2. According to the fitness, only one of the two generated children is selected for the next generation.

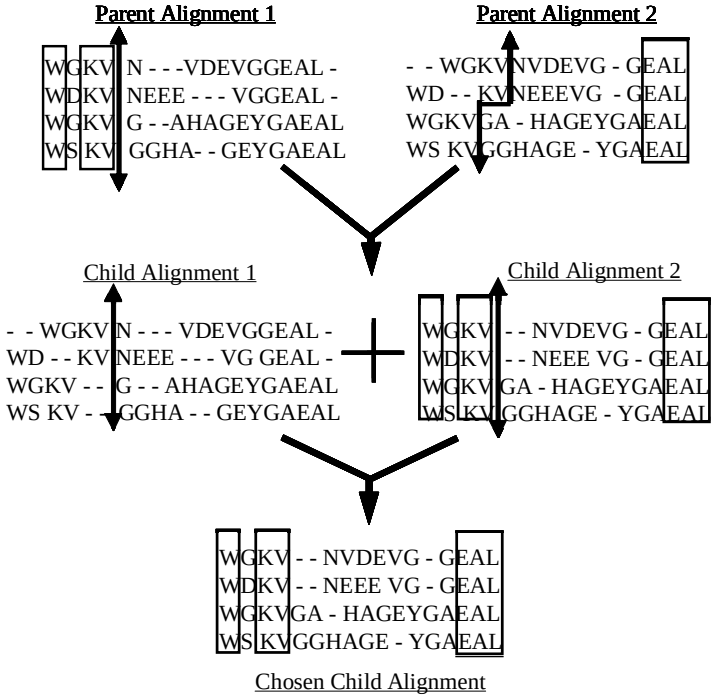


Figure 1: One point crossover between two parent alignments [9]

*(ii) Uniform crossover:* Figure 2 shows the uniform crossover process [9]. In this figure "*" mark indicates the common residues between two selected parents. Child Alignment from these two parents is generated by swapping blocks between them where each block is randomly chosen (or selected based on the best score).

*Mutation (Gap insert):* In this case, one parent is randomly selected. In order to

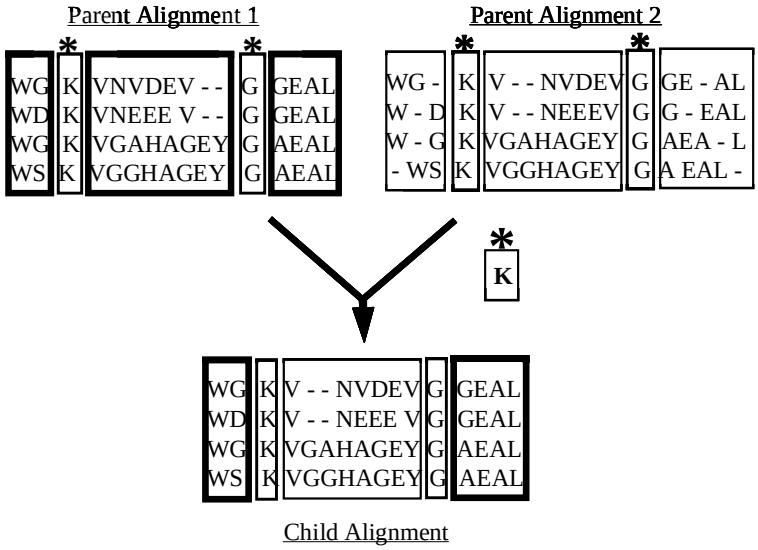


Figure 2: Uniform crossover [9]

generate a child, the sequences of the parent are split into two groups. The sequence numbers of each group are randomly selected. The length of the gaps and the positions of each group where gaps will be inserted are randomly selected. In each group the same length of gaps will be inserted in the same position of all sequences.

The genetic operators of this method are primitive in which the operators can change small parts of parents and all possible solutions can be probabilistically obtained. Though SAGA is not so efficient to find good quality of solutions but the one-one correspondence and the primitiveness of the genetic operators are important characteristic.

## 2.2 Evolutionary Tree-base Method

The progressive alignment algorithm utilizes the pair-wise alignment algorithm of Needleman and Wunsch [16] iteratively in order to obtain a multiple sequence alignment and to construct a guide tree to depict the relationship between sequences. The major problem of the progressive approach is trapping at local minimums and a way to overcome this problem is to use iterative procedure with progressive approach.

In order to make the tree-base method as iterative process, GA is introduced to find good guide trees. For this purpose, guide trees are represented by chromosomes in the ET method. And then the initial chromosomes from the same sequences are generated by using Dynamic Programming (DP) with random selection. The random selection mechanism is used for generating various chromosomes (guide trees). Note that the DP generates only one guide tree in the conventional tree-base method. That is, a number of chromosomes can be generated by using the DP with the random selection. After generating initial chromosomes, evaluations, selection and operators

(crossover and mutation) are iteratively performed to generate new populations. The algorithm of this process is shown in Fig. 3.

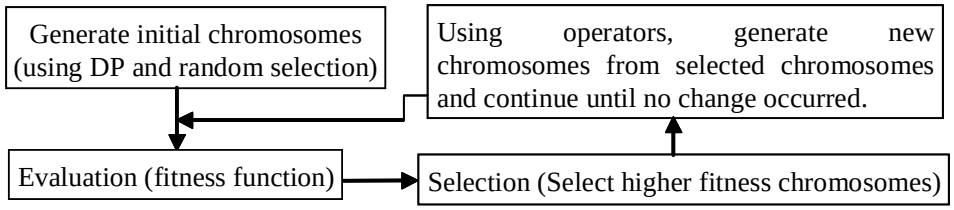


Figure 3: Evolutionary Tree-base algorithm

**Fitness Evaluation**

In order to evaluate the fitness value of a chromosome, the multiple alignment is constructed from the chromosome. For example, Figure 4 shows the process of generating the multiple alignment for the following five sequences. G1 corresponds to the alignment between *Seq* 2 and *Seq* 3, G2 between *Seq* 1 and *Seq* 4, and G3 between *Seq* 5 and the result of G1. Finally the multiple alignment is obtained by aligning the results of G3 and G2. The order of consecutive pair-wise alignments is determined by the guide tree shown at the left side of the figure.

*Seq* 1: NFS; *Seq* 2: NYLS; *Seq* 3: NKYLS;

*Seq* 4: NFLS; *Seq* 5: NKLS

The fitness of the chromosome is measured by the following SPM (Sum of pair) method.

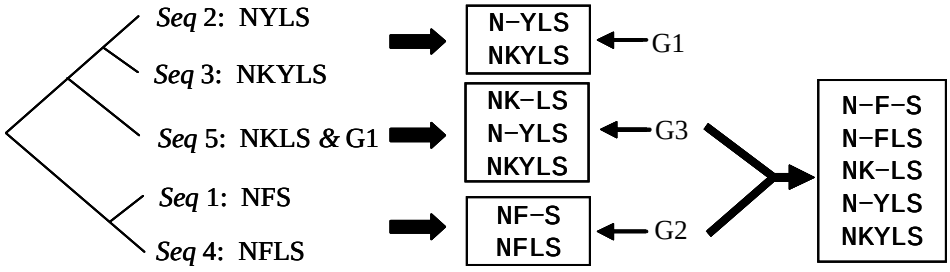


Figure 4: Multiple alignment of guide tree (chromosome)

## Sum of Pair Method (SPM)

By using the SPM the fitness of a chromosome can be determined by using (1), (2) and (2c).

$$S = \sum_{l=1}^{L} S_l \tag{1}$$

where,

$$S_l = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \cos t(A_i A_j) \quad (2)$$

Here, $S$ is the cost of the multiple alignment. $L$ is the length (columns) of alignment; $S_l$ is the cost of $l$-th column of $L$ length. $N$ is the number of sequences, $A_i$ ($A_j$) the aligned sequence $i(j)$ and $cost(A_i, A_j)$ is the alignment score between the two aligned sequences$A_i$ and $A_j$. When $A_i \neq$ '-' and $A_j \neq$'-' then $cost(A_i, A_j)$is calculated from PAM250[17], mutation probability matrix. The cost function includes the sum of the substitution costs of the insertion/deletions using a model with affine gap penalties shown in (2c).

$G = g + nx$ (2c)

Here, $G$ is gap penalty, $g$ is the cost of opening a gap, $x$ is the cost of extending gaps by one and $n$ is the length of the gap. By this way, the fitness of a chromosome is calculated. This scoring method is applied on all chromosomes and determines their scores. All the chromosomes are ranked according to their fitness.

## Selection

To select chromosomes for the next generation, the roulette wheel method is used. The number of occurrences of a chromosome is proportional to its fitness.

## Genetic Operators

*Crossover:* In order to generate new chromosomes by using crossover, two chromosomes are randomly selected. In the evolutionary tree-base method, there are two ways to do crossover between selected chromosomes.

*(i) Subtree selection method*: A subtree is randomly selected from one parent chromosome. The sequences included in the subtree are removed from the other parent chromosome and a rooted tree is reconstructed by connecting the remaining parts according to the relation in the original tree. That is, the nearest nodes are always connected to be a rooted tree. Then the reconstructed tree from the second parent and the selected subtree from the first parent are connected together to make a new guide tree. Note that all the sequences should be always in the new guide tree. This process is shown in Fig. 5. In this figure, (a) and (b) show two selected parents, (c) shows subtree which is selected from (a) {1 0}, here "1" means the right branch of (a) and "0" means the left side of the right branch of (a), (d) shows the remaining tree after the sequences of the subtree are discarded from (b), and (e) shows new chromosome after adding (c) and (d).

*(ii) Tree uniform order method:* In this crossover, firstly some sequences are randomly selected from one parent. The selected sequences are connected according to the relation in the original tree to construct a rooted tree. The selected sequences are removed from the other parent and a rooted tree is reconstructed by connecting the remaining parts according to the relation in the original tree. That is, the nearest nodes are always connected. Then the reconstructed tree from the second parent and

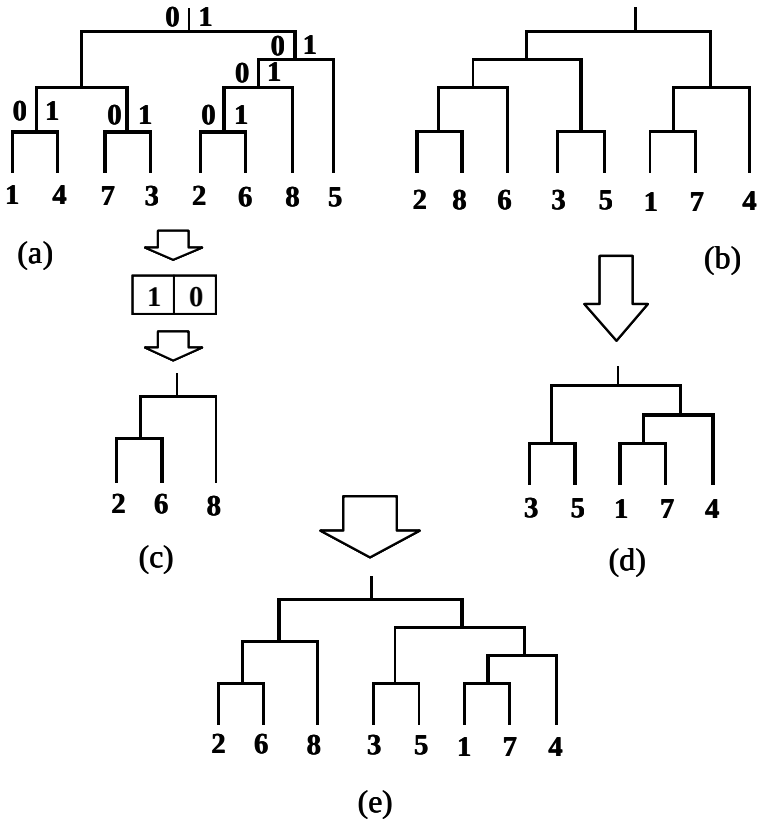


Figure 5: Subtree selection method

the selected subtree from the first parent are connected together to make a new guide tree. Note also that all the sequences should be always in the new guide tree. This process is shown in Fig. 6. In this figure, (a) and (b) show two selected parents, (c) shows the new tree which is randomly selected sequences from (a) according to the randomly selected mask pattern {0 1 1 0 1 1 0 0 }, (d) shows the remaining tree after the sequences of the subtree are removed from (b), (e) shows a new chromosome after adding (c) and (d).

*Mutation:* In the mutation, one parent is randomly selected. Then the two sequences of the selected parent are randomly selected and exchange their position, and then a new chromosome is generated. Figure 7 is an example of the mutation: (a) shows the selected parent in which P1 and P2 show the randomly selected two sequences. (b) shows a new chromosome generated by exchanging the positions of these two sequences.

The steps of the ET method are repeated iteratively, generation after generation. During these generation cycles, new pieces of alignment appear because of the operators. The selection makes sure that the good pieces survive and the dynamic

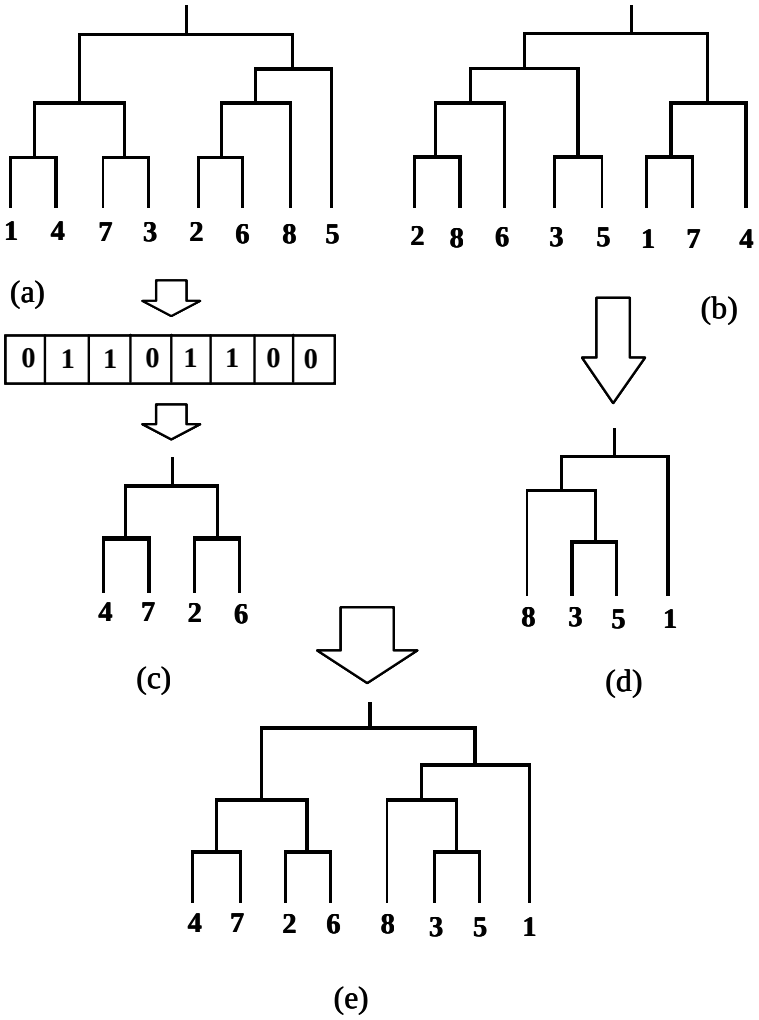


Figure 6: Tree uniform order method

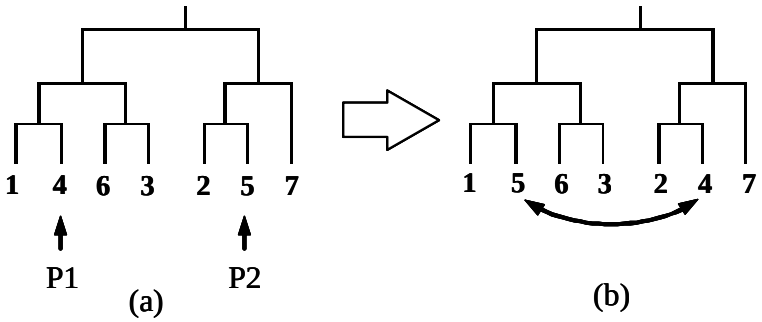


Figure 7: Mutation

setting of the operators helps the chromosomes to be improved by creating the children it needs. This process will continue until the fitness value of the chromosome per generation is reached to the saturation. Though the ET method reaches to the local minimum very quickly, however, the solution at the local minimum is better than the final solution of other methods. The ET method is capable of finding better quality solutions within small number of generations than other methods. This is an important characteristic of the ET method.

# 3 Two Phase Evolutionary Multiple Alignments

## 3.1 Reinforcement of Evolutionary Tree-base Method

The authors developed the evolutionary tree-base method. As explained in the previous section, the ET can efficiently obtain good quality of multiple alignments. Table 3.1 shows some results of experimental evaluation. We can observe that the ET can search better quality of multiple alignments comparing with well-known methods. However, the ET can be easily trapped at local minimums comparing with SAGA. Figure 8 shows the solution curves of the ET and SAGA. The ET can improve very quickly solutions but has converged at early stage, while SAGA is slow to improve solutions but the convergence is also slow. From the enlarged view of Fig. 8, we observed that the ET has converged after 200 generations and SAGA has converged after 500 generations. The convergent point of the ET is quite earlier than SAGA. Therefore, the ET and SAGA may compensate for each other's weak points if we combine both evolutionary approaches.

Table 3.1. Summary of BAliBASE Test results

| Data set | BAliBASE (score) | SAGA (score) | T-Coffee (score) | MUSCLE (score) | MAFFT (score) | ProbCons (score) | ET (score) |
|---|---|---|---|---|---|---|---|
| Ref. 1 (1aboA) | -512 | -762 | -679 | -291 | -462 | -494 | -220 |
| Ref. 2 (1aboA) | 8054 | 7067 | 7912 | 8549 | 8587 | 8106 | 9988 |
| Ref. 3 (1idy) | 7855 | 1070 | 1372 | 6830 | 7404 | 7992 | 9173 |
| Ref. 4 (1mfa) | -7484 | -10955 | -8806 | -6698 | -6579 | -7226 | -5381 |
| Ref.5 (Kinase3) | 53435 | 32305 | 54496 | 56315 | 56238 | 54935 | 59377 |

## 3.2 Two Phase Approach

In this paper, we propose a Two Phase evolutionary method. The first phase is to search efficiently and quickly a set of good solutions with the ET and the second phase with SAGA is to improve the searched solutions. This combination is quite

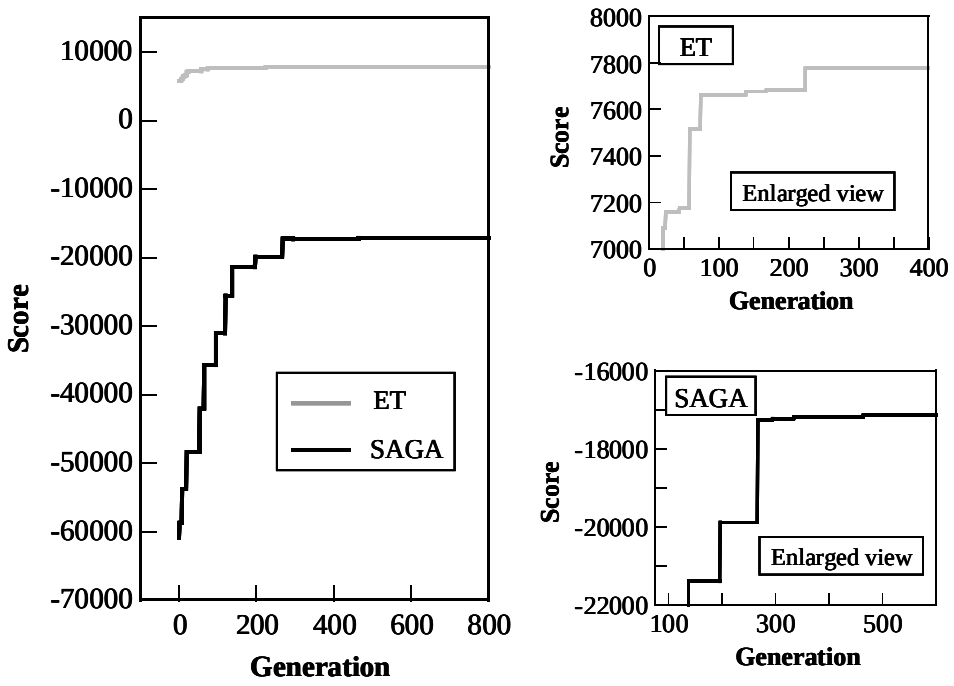


Figure 8: The solution curves of the ET and SAGA

reasonable since the ET is good at searching good quality solutions but its convergence is fast relatively while SAGA is slow to find good solutions but its convergence is also slow.

Our proposed Two Phase method starts from completely unaligned sequences. At first a number of guide trees (chromosomes) are generated by the DP and random selection and the ET starts searching from the initial population and continues until the new generation is failed to update the highest fitness chromosomes for long time. That is, the ET method is converged at some local minimums. The multiple alignments found by this convergence should be good quality but we need more accurate solutions. And then the second phase starts from the multiple alignments with SAGA. The multiple alignments generated in the first phase are transferred into the initial generation of the second phase. Note that a chromosome in SAGA represents a multiple alignment itself. The second phase should be stopped when the evolution in SAGA is saturated.

# 4 Experimental Evaluation

In order to evaluate our proposed method, we did experiments for input datasets obtained from two resources; BAliBASE and the NCBI database [18]. For comparison, we employed softwares such as SAGA, T-Coffee, MUSCLE, MAFFT, ProbCons, and the ET. Among these softwares, source programs of only SAGA and the

ET are available to the authors and the remaining ones are utilized through the online services [19]. On using the online softwares, we got the final multiple alignment results corresponding to each input sequence set and the multiple sequence alignment was evaluated by the same objective function (SPM).

We experimented for five sequence sets from BAliBASE. The results are shown in Table 4.1. The rows Ref.1, Ref.2, ..., Ref.5 represent the input sequence sets and the columns correspond to the methods. Therefore, each cell shows the SPM score of the method specified by the column for the input data of the row. From Table 4.1, we observed that our proposed Two Phase method obtains better quality of solutions than the others. Except for the Two Phase method, our ET was superior to the others.

Table 4.1. Summary of Test results of BAliBASE Datasets

| Data set | BAliBASE (score) | SAGA (score) | T-Coffee (score) | MUSCLE (score) | MAFFT (score) | ProbCons (score) | ET (score) | Two Phase (score) |
|---|---|---|---|---|---|---|---|---|
| Ref. 1 (1aboA) | -512 | -762 | -679 | -291 | -462 | -494 | -220 | -202 |
| Ref. 2 (1aboA) | 8054 | 9067 | 7912 | 8549 | 8587 | 8106 | 9988 | 10133 |
| Ref. 3 (1idy) | 7855 | 1070 | 1372 | 6830 | 7404 | 7992 | 9173 | 9285 |
| Ref. 4 (1mfa) | -7484 | -10955 | -8806 | -6698 | -6579 | -7226 | -5381 | -5164 |
| Ref.5 (Kinase3) | 53435 | 32305 | 54496 | 56315 | 56238 | 54935 | 59377 | 59399 |

We carried out the experiments for ten sequence sets with different lengths taken from the NCBI database. The score values of the experiments were also calculated by the SPM, and are summarized in Table 4.2. From the comparisons of the experimental results, we observed that the Two Phase approach has the best performance for the accuracy of solutions. Except for the Two Phase method, the ET could obtain better solutions than any other methods for each experiment.

## Computation Time

The computation time of our proposed Two Phase method is quite longer than the other methods. However, practically we can relax the problem of the computation time. Figure 9 represents an example of increasing highest fitness value (score) in searching with the ET. We can see that the final highest fitness could be obtained much earlier than the time the searching stopped. We can see in Table 4.3 that the computation time in the case of 2,054 generations was 4,690.05 seconds but the highest score was obtained at generation 1,054, which is observed in the enlarged view of the graph in Fig. 9. In such case, the computation time to reach the highest score was about 2406.94 seconds. Moreover, such a situation can be seen in the first phase. In the Two Phase method, the ET method can obtain much earlier the final fitness value of the first phase. For example, the computation time of the ET method for 1,015 generations was 2004.00 (CPU time) seconds, but the highest score was obtained at

Table 4.2. Summary of Test results of NCBI Datasets

| Data set | No. of sequences | Sequence length | SAGA (score) | T-Coffee (score) | MUSCLE (score) | MAFFT (score) | ProbCons (score) | ET (score) | Two Phase (score) |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 7 | 153 | -1826 | 1598 | 2419 | 2447 | 2225 | 2496 | 2546 |
| 2 | 12 | 74 | 2740 | 6634 | 7105 | 6846 | 6513 | 7742 | 7831 |
| 3 | 15 | 603 | -40098 | -38411 | -28423 | -28297 | -50741 | -3782 | -3488 |
| 4 | 21 | 996 | -115453 | -140166 | -100802 | -104781 | -171058 | -39573 | -39053 |
| 5 | 18 | 495 | -40554 | -51367 | -34221 | -35742 | -56067 | -10867 | -10708 |
| 6 | 20 | 849 | -74373 | -88668 | -62143 | -62821 | -100389 | -15521 | -15503 |
| 7 | 16 | 440 | -29564 | -17229 | -7800 | -8461 | -27832 | 14579 | 14765 |
| 8 | 26 | 1012 | -196529 | -158592 | -120571 | -124287 | -172690 | -63737 | -63495 |
| 9 | 18 | 841 | 11231 | 53046 | 102465 | 111289 | 60099 | 141388 | 141649 |
| 10 | 19 | 155 | -17120 | -3448 | 3635 | 5009 | 6950 | 7778 | 8702 |

generation 15. Therefore, the ET needed only 30.06 seconds to reach the final solutions of the first phase. Table 4.3 shows the computation time of SAGA, T-Coffee, the ET, and the Two Phase method when we execute them for the NCBI dataset shown in Table 4.3. In the experiment, we used Windows XP environment on Pentium M processor with 1.40 GHz. The programs for SAGA, the ET, and the Two Phase method were developed with C language. The computation time of T-Coffee was provided from the server. We could not measure the computation time for MUSCLE, MAFFT, and ProbCons.

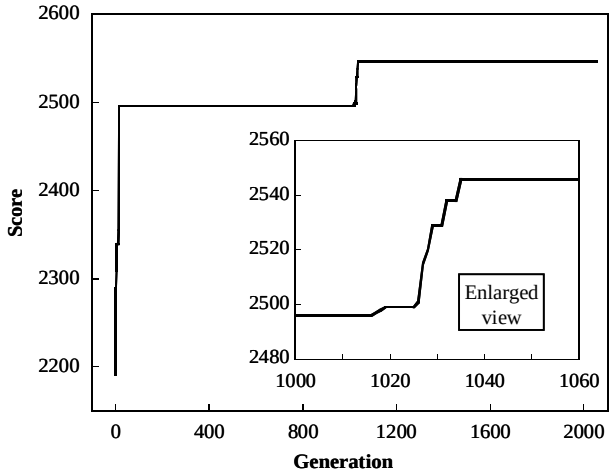


Figure 9: Score-Generation output by Two Phase method 1st experiment

The number of generations required to obtain the highest score value can not be determined before searching. However, searching programs may judge the situation

Table 4.3. Summary of Computation Time of Second Group Test Results

| Data set | SAGA | | T-Coffee | ET | | Two Phase | |
|---|---|---|---|---|---|---|---|
| | CPU time for total computation (sec.) | Effective computation time for good solution (sec.) | CPU time for computation (sec.) | CPU time for total computation (sec.) | Effective computation time for better solution (sec.) | CPU time for total computation (sec.) | Effective computation time for optimum solution (sec.) |
| 1 | 208.50 | 57.192 | 1.88 | 2004.00 | 30.06 | 4690.05 | 2406.94 |
| 2 | 200.40 | 47.415 | 3.90 | 1087.00 | 47.83 | 1535.00 | 803.42 |
| 3 | 3166.80 | 1482.38 | 30.28 | 36365.00 | 8305.77 | 118654.46 | 67775.42 |
| 4 | 11260.80 | 5979.49 | 94.24 | 32811.00 | 4104.67 | 102243.55 | 56458.89 |
| 5 | 3061.56 | 1481.80 | 29.97 | 28406.00 | 2556.54 | 60223.03 | 31683.34 |
| 6 | 5534.56 | 2792.19 | 69.84 | 19259.07 | 1209.47 | 58521.20 | 30360.8 |
| 7 | 1032.24 | 72.05 | 27.80 | 11249.00 | 1132.77 | 38223.45 | 20216.38 |
| 8 | 10534.08 | 3516.28 | 72.11 | 35546.24 | 4528.59 | 134137.49 | 75479.17 |
| 9 | 8069.35 | 3214.02 | 700.09 | 59031.28 | 14173.41 | 142876.55 | 81596.8 |
| 10 | 918.49 | 153.11 | 10.87 | 3446.36 | 633.10 | 10076.55 | 5707.36 |

of the convergence. For example, we can set the condition of the convergence: the searching program can judge its convergence if the predetermined number of generations has passed since the last update of the best fitness. So, if we introduce a proper predetermined number for judging of the convergence, we can reduce drastically the computation time of the Two Phase method.

Actually, the main goal of the Two Phase method is to find out a better quality of multiple alignments, which is verified through the experimental evaluation. However, the computation time of the Two Phase method is quite longer than others even though we can reduce drastically the computation time by setting the condition of the convergence. The time problem may be solved by parallelizing the Two Phase method though this issue is beyond the scope of this paper. There are lots of researches on parallel processing of evolutionary computation.

An example of an alignment obtained by proposed Two Phase method is shown in Fig. 10. This figure shows the multiple alignment for 1aboA_ref2 from BAliBASE, Reference 2. The score of this alignment according to SPM is 10,133.

# 5 Conclusions

This paper proposed a new evolutionary method, namely, a Two Phase evolutionary algorithm for multiple sequence alignments. This method is composed

```
1aboA   -----NLF---V-ALYDF-VASGDN-TLS----I--TKGE-K-L--RV-LGY-NHNG-EWCEA-QTKNGQ-GWVP---SNYI--TPV-N
1ark    TAG--KIF----RAMYDY-MAADAD-EVS----F--KDGDAI-I--NV--Q--AIDE-GWMYGTVQRTGRTGMLP---ANYV--EAI--
1gbq    --M--EA----I-AKYDF-KATADD-ELS----F--KRGD-I-L--KV-LN-EECDQ-NWYKA-ELN-GKDGFIP---KNYI--EMKP-
1ckb    --A--E-Y---VRALFDF-NGNDEE-DLP----F--KKGD-I-L--RI-RD--KPEE-QWWNA-EDSEGKRGMIP---VPYV--EKY--
1gfc    --G--STY---VQALFDF-DPQEDG-ELG----F--RRGD-F-I--HV-MD--NSDP-NWWKG-ACH-GQTGMFP---RNYV--TPV--
1hsp    G-SP-T-FKCAVKALFDY-KAQRED-ELT----F--IKSA-I-I--QN-VE--KQEG-GWWRG-DYGGKKQLWFP---SNYV--EEM-V
1aey    --GK-E-L---VLALYDY-QEKSPR-EVT----M--KKGD-I-L--TL-LN--STNK-DWWKV-EVN-DRQGFVP---AAYV--KKL--
1csk    --G--TEC---I-AKYNF-HGTAEQ-DLP----F--CKGD-V-L--TI-VAV-TKDP-NWYKA-KNKVGREGIIP---ANYV--QKR--
1ad5    --E--DII---VVALYDY-EAIHHE-DLS----F--QKGD-Q-M--VV-LE--ESG--EWWKARSLATRKEGYIP---SNYV--ARV-D
1awj    RRSFQEPEETLVIALYDY-QTNDPQ-ELA----L--RCDE-E-Y--YL-LD--SSEI-HWWRV-QDKNGHEGYAP---SSYL-VEKS--
1efn    -----ALF---V-ALYDY-EAITED-DLS----F--HKGE-K-F--QI-LN--SSEG-DWWEARSLTTGETGYIP---SNYV--APV--
1sem    --E--TKF---VQALFDF-NPQESG-ELA----F--KRGD-V-I--TL-IN--KDDP-NWWEG-QLN-NRRGIFP---SNYV--CPY--
1ycsB   -KG--VIY-----ALWDY-EPQNDD-ELP----M--KEGD-C-M--TI-IHREDEDEIEWWWA-RLN-DKEGYVP---RNLL--GLYP-
1pht    --GY-Q-Y----RALYDYKKEREEDIDLHLGDILTVNKGSLVALGFSDGQEA-RPEEIGWLNGYNETTGERGDFPGTYVEYIGRKKISP
1vie    --D--R-----VRKKSG--AAWQGQ-IVG---WY---CTN---L---T-PE---GYA-V--ES-EAHPGSVQIYP---VAAL--ERI-N
1ihvA   -----N-F----R-VY-Y---RDSR-D-P----V--WKGP-AKL---L-W---KGEG-A-VVI-QDNSDIK-VVP-RRKAKI--IRD--
```

Figure 10: An example of multiple alignment by Two Phase method

of different types of evolutionary algorithms: an evolutionary progressive multiple sequence alignment method (ET) and Sequence Alignment by Genetic Algorithm (SAGA). The former is employed to obtain efficiently good quality of multiple alignments and the latter to improve them much better. Experimental evaluation showed that the proposed Two Phase method can generate good quality of multiple alignments.

As future works, we need to reduce the computation time by tuning genetic parameters such as population size, crossover rates, and by investigating a proper condition and its implementation for switching the phases. Moreover, parallel processing of the Two Phase method should be also effective.

# References

[1] B. Rost, and C. Sander, "Prediction of protein secondary structure at better than 70% accuracy," *J. Mol. Biol.*, vol. 232, pp. 548-599, 1993.

[2] D. F. Feng, and R.F. Dolittle, "Progressive sequence slignment as a prerequisite to correct phylogenetic trees," *J. Mol. Evol.* vol. 25, pp. 351-350,1987.

[3] W. R. Taylor, "A flexible method to align large number of biological sequences," *J. Mol. Evol.* vol. 28, pp. 151-159, 1988.

[4] J. D. Thompson, D. G. Higgins, and T. J. Gibson, "CLUSTAL W: Improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice, *Nucleic Acids Res.*, vol. 22, pp. 4673-4680, 1994.

[5] G. J. Barton, and M. J. E. Sternberg, "A strategy for the rapid multiple alignment of protein sequences," *J. Mol. Biol.* Vol. 198, pp. 327-337, 1987.

[6] O. Gotoh, "Optimal alignment between groups of sequences and its application to multiple sequence alignment," *Comput. Appl. Biosci.* vol. 9, pp. 361-370, 1993.

[7] A. V. Lukashin, J. Engelbrecht, and S. Brunak, "Multiple alignment using simulated annealing: branch point definition in human mRNA splicing," *Nucleic Acids Res.* vol. 20, pp. 2511-2515, 1992.

[8] C. E. Lawrence, S. F. Altschul, M. S. Boguski, J. S. Liu, A. F. Neuwald, and J. C. Wooton, "Detecting subtle sequence signqls: q Gibbs sampling strategy for multiple alignement," *Science,* vol. 262, 208-214, 1993.

[9] C. Notredame, and D. G. Higgins, "SAGA: Sequence Alignment by genetic algorithm," *Nucl.Acids Res.* vol. 24, pp. 1515-1524, 1995.

[10] D. E. Golberg, "Genetic Algorithms in Search, Optimization and Machine Learning," Addison-Wesley, New York, 1989.

[11] http://bips.u-strasbg.fr/fr/Products/Databases/BAliBASE/

[12] C. Notredame, D. G. Higgins, & J. Heringa, "T-Coffee: A novel method for fast and accurate multiple sequence alignment," *J. Mol. Biol.* vol. 302, pp. 205-217, 2000.

[13] R. C. Edgar, "MUSCLE: Multiple sequence alignment with high accuracy and high throughput," *Nucleic Acids Res.* vol. 32, pp. 1792-1797, 2004.

[14] K. Katoh, K. Kuma, H. Toh, and T. Miyata, "MAFFT version 5: Improvement in accuracy of multiple sequence alignment," *Nucleic Acids Res.*, vol. 33, pp. 511-518, 2005.

[15] C. B. Do, M. S. Mahabhashyam, M. Brudno, S. Batzoglou, "ProbCons: Probabilistic consistency-based multiple sequence alignment," *Genome Res.*, vol. 15(2), pp. 330-340, Feb., 2005

[16] S. B. Needleman, and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *J. Mol. Biol.*, vol. 48, pp. 443-453, 1970.

[17] http://www.icp.ucl.ac.be/∼opperd/private/pam250.html

[18] http://www.ncbi.nlm.nih.gov/

[19] http://www.answers.com/topic/sequence-alignment-software