International Journal of Pure and Applied Mathematics
Volume 120 No. 6 2018, 4223-4236
ISSN: 1314-3395 (on-line version)
url: http://www.acadpubl.eu/hub/
Special Issue

http://www.acadpubl.eu/hub/

DIFFERENTIAL EVOLUTION WITH NEW MUTANT - A CASE STUDY ON SEQUENCE ALIGNMENT

Lakshmi Naga Jayaprada.Gavarraju[1],
Kanadam. KarteekaPavan[2],
[1]Asso.Professor
Department of Computer Science and Engineering
Narasaraopeta Engineering College
Narasaraopet-522 601, Andhra Pradesh, India
gjayaprada74@gmail.com
[2]Professor & Head
Department of Computer Applications
R.V.R & J.C. College of Engineering
Guntur - 522 019, Andhra Pradesh, India
karteeka@yahoo.com

June 22, 2018

Abstract

Sequence Alignment problem is modelled as an optimization problem that optimize the gap positions. Differential Evolution is a simple, efficient evolutionary algorithm which require less number of algorithmic control parameters compared to other evolutionary algorithms. Thepaper proposed a variant to the Differential Evolution by defining a novel mutation operator DE/current-to-best-worst/1. The proposed mutation operator is used to find new offsprings using best and worst candidates solutions.The performance of the proposed method is evaluated by conducting experiments

with several prefabref4 benchmark datasets and results are compared with the existing mutation operators. It is also compared with so many existing sequence alignment tools like CLUSTALW, DIALIGN-TX v1.0.2, FSA v1.14.5, FSA with -maxsn, KALIGN v2.03, MAFFT v6.603 using EINSI, MUMMALS, MUSCLE v4.0, PROBCONS v1.12 and PROBALIGN v1.0 with the help of prefabref4 benchmark data sets. The results of the experiments revealed the supremacy of the new mutant.

***Keywords*:**Sequence Alignment, Mutation Operator, Differential Evolution, Sequence Alignment Tools, Benchmark data sets.

1 INTRODUCTION

In the field of Bio-informatics Sequence Alignment plays a vital role. It is the concept of aligning DNA sequences or protein sequences. Optimal alignment of two sequences maximizes the number of identical of similar bases by introducing gaps [1]. Pairwise sequence alignment maximizes similarities or identities between two sequences and multiple sequence alignment(MSA) maximizes similarities or identities among three or more sequences[2, 3]. Pairwise sequence alignment aligns the two sequences considering a match, mismatch or a gap. Gaps are used to specify indels (insertions and deletions) in a sequence alignment by aligning them in two parallel rows.Asimple scoring process is used to calculate the fitness for DNA sequences having nucleotide bases i.e., a simple identification scheme. A simple identification scheme is used to find the fitness where the match=+2, mismatch=-1 and gap=-2. Point Accepted Mutation (PAM) [4] is the substitution matrices used to find similarity score in protein sequence alignment.Blocked Substitution Matrix (BLOSUM) [5] is also used to find similarity score. In sequence alignment there are three major categories like global, local and semi global. During the process of global alignment the entire length of each sequence is considered. Local sequence alignment considers portions of sequences with high similarity score. Semi global alignment finds the best possible alignment which includes the start and end of one or the other sequence[6].

For pairwise sequence alignment Dynamic programming (DP) is used where a matrix is created with two axes representing two sequences. DP aligns all possible pairs in a systematic way based on matches, mismatches and gaps. By using back tracking [7, 8] the optimal alignment is determined by calculating the score. In 1970 Needleman-wunschhave proposed an algorithm for global alignment [9]. Commonsub sequences can be identified using local alignment as proposed by Smith-waterman [10]. The two algorithms cannot be used when the length of the sequence is very high because computational complexity is high [11, 12]. To reduce the computational complexity metaheuristics were developed. BLAST[13] and FASTA[14] are heuristic algorithms. Major types of metaheuristic methods for MSA are progressive methods, iterative methods and consistency based methods [15, 16]. ClustalW[17, 18] is the most familiar progressive alignment algorithm used for MSA. It uses a distance matrix to construct guide tree and gets the alignment. Another iterative alignment algorithm is MUSCLE[19] which solves the problem of alignment. One more alignment technique T-Coffee[20] avoids the greedy nature present in progressive alignment. In this technique initial alignment is taken from ClustalW and Lalign[21] alignments.

Differential evolution (DE) [22] is the recent successful heuristic real coded parameter optimization method. DE is a simple, efficient evolutionary algorithm which require less number of algorithmic control parameters compared to other evolutionary algorithms. DE provides prominent results in a wide variety of applications.

A new mutation operator is proposed for DE and named as Improved DE (IDE). The performance of IDE is evaluated using various benchmark data sets for sequence alignment. Basically there are 5 mutants in DE. Among the 5 mutants performance wise DE/best/1 was proved as best. It was observed that performance improvement ofIDE over DE/best/1 is observed with respect to time. IDE is taking less time than DE/best/1. In view of fitness IDE is working slightly better than DE/best/1.The efficacy of the IDE is compared with CLUSTALW, DIALIGN-TX v1.0.2, FSA v1.14.5, FSA with -maxsn, KALIGN v2.03, MAFFT v6.603 using EINSI, MUMMALS, MUSCLE v4.0, PROBCONS v1.12 andPROBALIGN v1.0 with the help of prefab4ref. i.e., the pair-wise reference pairs in PREFAB v4 benchmark data sets and

this proposed mutant outperformed all the above said methods. This paper is consisting of 4 sections. Section 1 is introduction. Section 2 is explains the new mutant IDE. Experimental results are discussed in section3 and section 4 contains conclusions and future enhancements.

2 NEW MUTANT IDE

Differential evolution (DE) [22] is very efficient evolutionary algorithm for solving many real-world applications. It is a population-based optimisation method developed by Storn and Price in 1996. In evolutionary computation, DE solves a real world application by iteratively trying to develop candidate solutions. These methods are normally named as metaheuristics as they create few or no assumptions about the problem being optimized and can search very large spaces of candidate solutions.

2.1 Differential Evolutionary Algorithm

The Pseudo-code for Differential Evolutionary algorithm with Binomial Crossover is as shown below.
1: Control parameters of DE like scale factor F, crossover rate Cr, and the population size NPare taken as input.
2: The generation number 'g' is taken as '0' and initialize a population of NP individuals randomly.
$PG = X_{1,g}, ..., X_{NP,g}$ with $X_{i,g} = [x_{1,_{i,g}}, x_{2,_{i,g}}, x_{3,_{i,g}}, ..., x_{D,_{i,g}}]$ and each individual uniformly distributed in the range $[Xmin, Xmax]$, where $X_{min} = \{x_{1,min}, x_{2,min}, ..., x_{D,min}\}$ and $X_{max} = \{x_{1,max}, x_{2,max}, ..., x_{D,max}\}$ with $i = [1, 2, ..., NP]$.
3. until the stopping criterion is not reached
repeat the following steps
3.1 Compute a donor vector $S_{i,g} = s_{1,i,g}, ...,$
$s_{D,i,g}$ corresponding to the ith target vector Xi,gvia the differential mutation scheme of DE as
$S_{i,g} = X_{r1,g} + F(X_{r2,g}X_{r3,g})$.
3.2 Compute a trial vector $T_{i,g} = \{t_{1,i,g}, ..., t_{D,i,g}\}$ for the ith target vector $X_{i,g}$ through binomial crossover in the following way:
$t_{j,i,g} = s_{j,i,g}$, if $(rand_{i,j}[0, 1] \leq Cr or j = j_{rand})xj, i, g$, else,
3.3 Compute the trial vector $T_{i,g}$ IF $f(T_{i,g}) \leq f(X_{i,g})$, THEN

$X_{i,g+1} = T_{i,g}$
ELSE $X_{i,g+1} = X_{i,g}$.

Existing mutation operators for DE are shown here.

"DE/rand/1". $V_{i,G} = X_{r_1,G} + F \cdot (X_{r_2,G} - X_{r_3,G})$

"DE/best/1". $V_{i,G} = X_{best,G} + F \cdot (X_{r_1,G} - X_{r_2,G})$

"DE/current-to-best/1". $V_{i,G} = X_{i,G} + F \cdot (X_{best,G} - X_{i,G}) + F \cdot (X_{r_1,G} - X_{r_2,G})$

"DE/best/2". $V_{i,G} = X_{best,G} + F \cdot (X_{r_1,G} - X_{r_2,G}) + F \cdot (X_{r_3,G} - X_{r_4,G})$

"DE/rand/2". $V_{i,G} = X_{r_1,G} + F \cdot (X_{r_2,G} - X_{r_3,G}) + F \cdot (X_{r_4,G} - X_{r_5,G})$

Where

$$r_1, r_2, r_3, r_4, r_5 \in [1, \cdots, NP]$$

are randomly chosen integers, and .

$$r_1 \neq r_2 \neq r_3 \neq r_4 \neq r_5 \neq i$$

Where 'F' is a scaling factor that takes a random value in between 0 and 1. In all these mutants"DE/rand/1"is proved as best with respect to time and "DE/best/1" was proved as best with respect to performance. Now this paper proposes a new mutation variant "DE/current-to-best-worst/1" (IDE)by considering best and worst candidate solutions of current population. This mutant IDE is found as slightly better than "DE/best/1" in view of performance and it is found better than "DE/best/1" in view of execution time.

2.2 New mutant "DE/ current-to-best-worst/1" for Improved DE

Improved Differential Evolution is equipped with new mutant **"DE/ current-to-best-worst /1" (IDE)**.

$S_{i,g} = X_{best,g} + p(X_{worst,g} - X_{i,g})$

$X_{best,g}$ is best fit chromosome, $X_{worst,g}$ is worst fit chromosome, p is a random number in between 0 and 1.

2.3 Genome Representation

The Genome is a vector that represents position of the gaps in the resultant alignment. The gap locations of all the sequences are collectively known as single genome. The genome for the given two sequences is as shown in Fig1. Fig2 is a snapshot for population.
Input Sequences:
>MEVKKTSWTEEEDRILYQAHKRLGNRWAEIAKLLPGRTDNAIKNHWNSTMRRKV
>PRGSALSDTERAQLDVMKLLNVSLHEMSRKISRSRHCIRVYLKDPVSYGTS
Genome Representation for the chosen example :

```
population(:,:,1) =

    27    30     9    35    46    40    63     1    47     4    37     0     0     0
    17    48     7    39    58    20    65    28    51     6    36    24    59    26

population(:,:,2) =

    17     9    26    36    21    44    35    15    50     5    19     0     0     0
    48    37    39    23    46    47    55    54    13    58    61    51    28     9

population(:,:,3) =

    64     6    28    56    60    50    39     4     9     5    63     0     0     0
    36    21     3    53    25    44    33    51     5     9    61    12    50    32

population(:,:,4) =

    45     8    49    58    36     6    39     4    44    51    27     0     0     0
    13    14    63    17    50    35    49    15    26    57    48    65    52    16
```

Fig 1: Genome Representation

Gap positions in the resultant alignment is figured out as in fig 2.

2.4 Objective Function

The objective function is used to find quality of the candidate solution. The candidate solution with maximum score is considered as best solution. In this work "Total Column Score" (TC-Score) is used as objective function.

2.5 Stopping Criteria

The algorithm is repeated for maximum number of generations and outputs the best solution.

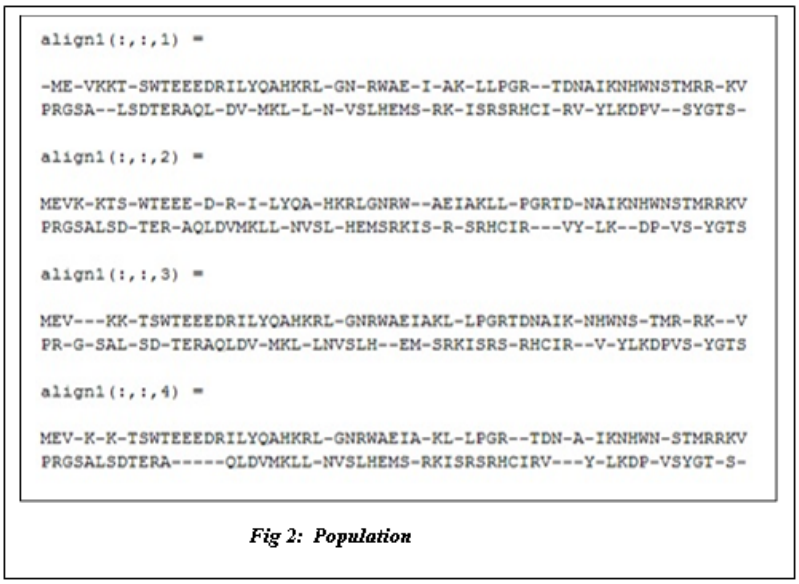


Fig 2: Population

3 EXPERIMENTAL RESULTS AND DISCUSSIONS

The efficacy of the new mutant IDE is evaluated by considering prefab4refbench mark data sets. Each dataset contains two variable length sequences. The performance of the IDE is compared with all the mutants of DE and some predefined tools CLUSTALW, DIALIGN-TX v1.0.2, FSA v1.14.5, FSA withmaxs-option, KALIGN v2.03, MAFFT v6.603 using EINSI, MUMMALS, MUSCLE v4.0, PROBCONS v1.12 and PROBALIGN v1.0 demonstrated in the form of Tableas shown below.

3.1 Performance of IDE with respect to time

The performance of the IDE is proved as better than "DE/best/1" with respect to time is demonstrated with table and graph.

S.No	Sequences	IDE	DE/ rand1	DE/ best1	DE/ current -to- best1	DE/ rand2	DE/ best2	CLUS TALW	DI ALIGN-TX v1.0.2	FSA v1.14.5	FSA with-maxsm	KALIGN v2.03	MAFFT v6.603 using ENS1	MUM MALS	MUS CLE v4.0	PROB CONS v1.12	PROB ALIGN v1.0
1	1a3k-1tcl	0.9489	0.9489	0.9489	0.9489	0.957	0.956	**0.985**	0.792	0.969	0.969	0.969	0.969	**0.985**	**0.985**	0.969	0.969
2	1a0fA-1hqoA	**0.92**	0.7194	**0.92**	0.9105	0.9047	0.916	0.807	0.679	0.722	0.754	0.701	0.791	0.834	0.802	0.588	0.765
3	1a0fA-1lprA	0.9731	0.9701	0.9701	0.9701	**0.975**	**0.975**	0.913	0.723	0.872	0.882	0.903	0.821	0.908	0.913	0.908	0.882
4	1a0fA-1pd2I	**0.925**	0.8557	0.8575	0.8553	0.8598	0.856	0.739	0.41	0.335	0.372	0.548	0.495	0.638	0.654	0.66	0.383
5	1a1z-1d2zA	0.4623	0.2925	0.434	0.3113	0.4335	0.298	0.434	0.237	0.25	0.289	0.303	0.303	0.329	0.303	0.289	**0.513**
6	1a3aA-1a6jA	**0.959**	**0.959**	**0.959**	**0.959**	**0.959**	**0.959**	0.899	0.863	0.921	0.921	0.921	0.849	0.95	0.921	0.914	0.863
7	1a3c-1tclA	**0.916**	0.8313	**0.916**	0.9017	0.9123	0.895	0.791	0.739	0.588	0.627	0.791	0.83	0.804	0.784	0.81	0.797
8	1a61-2fdn	0.3208	0.2075	0.2453	0.2358	0.2446	0.215	**0.927**	0.782	0.873	0.873	0.873	0.8	0.873	0.873	0.818	0.873
9	1a6m-1ash	0.8344	0.7483	**0.914**	**0.914**	0.7846	0.836	0.891	0.464	0.254	0.514	0.623	0.507	0.884	0.746	0.572	0.63
10	1a6m-1cg5B	0.9073	0.7818	0.9139	0.9205	0.9149	0.904	0.914	0.555	0.36	0.561	0.921	0.906	**0.957**	0.95	0.942	0.95
11	1a6m-4b97A	**0.914**	0.894	0.9038	0.9038	0.9045	0.905	0.794	0.142	0.411	0.532	0.787	0.34	0.801	0.525	0.525	0.56
12	1a6m-1h1b	**0.96**	**0.96**	**0.96**	**0.96**	**0.96**	**0.96**	0.945	0.938	0.959	0.952	0.959	0.959	0.959	0.952	0.952	0.952
13	1a6m-1ithA	**0.881**	0.8543	0.8445	0.8445	0.8532	0.853	0.662	0.655	0.676	0.676	0.669	0.669	0.734	0.741	0.676	0.669
14	1a6m-2fxl	**0.934**	**0.934**	**0.934**	**0.934**	0.9279	0.915	0.811	0.769	0.643	0.643	0.629	0.909	0.811	0.643	0.643	0.783
15	1a6m-2gdm	**0.974**	**0.974**	**0.974**	**0.974**	**0.974**	**0.974**	0.953	0.306	0.401	0.469	0.898	0.68	0.769	0.905	0.673	0.803
16	1a6m-2hbg	0.8808	0.8079	0.8543	0.8654	0.8035	0.855	0.906	0.587	0.543	0.551	0.928	0.336	0.819	0.891	0.594	**0.942**
17	1a6m-2hbbA	**0.775**	0.6225	0.7035	0.6935	0.7546	0.765	0.326	0	0	0	0.729	0	0.116	0.426	0.194	0.14
18	1a6m-1ffp	0.8742	0.8654	0.8554	0.8651	0.8703	0.85	0.934	0.438	0.343	0.35	0.934	0.81	**0.949**	0.818	0.81	0.723
19	1a7w-1b67B	0.9559	0.9347	0.9147	0.8947	0.9246	0.915	**1**	0.954	0.954	0.985	**1**	0.985	0.954	**1**	0.954	**1**
20	1a7w-1tafA	0.9851	0.9732	0.9732	0.9851	0.9851	0.975	**1**	0.533	0.477	0.492	**1**	**1**	**1**	**1**	0.969	0.769

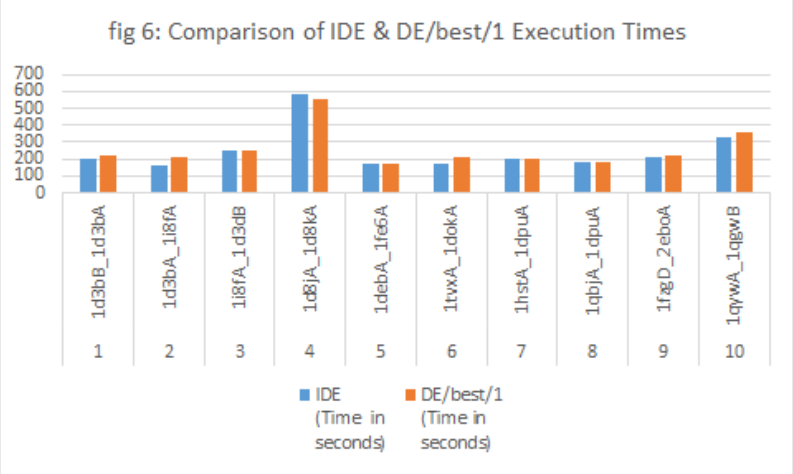
Figure 1: Performance comparison of IDE with other DE mutants and tools

S.No	Sequences	IDE	DE/ rand1	DE/ best1	DE/ current -to- best1	DE/ rand2	DE/ best2	CLUS TALW	DI ALIGN-TX v1.0.2	FSA v1.14.5	FSA with-massa	KALIGN v2.03	MAFFT v6.603 using EINSI	MUM MALS	MUS CLE v4.0	PROB CONS v1.12	PROB ALIGN v1.0
21	1a7w-1nzB	0.9155	0.9155	0.9155	0.9155	0.9155	0.916	1	0.846	0.954	0.954	1	0.969	1	1	0.954	0.415
22	1agdB-1iakA	0.9899	0.979	0.9809	0.9735	0.9749	0.974	0.574	0.959	1	1	1	1	1	1	0.98	1
23	1aly-2mfA	**0.954**	0.9247	0.9434	0.9247	0.9446	0.95	0.833	0.703	0.659	0.667	0.674	0.688	0.768	0.746	0.739	0.725
24	1a0iB-1a7w	0.6265	0.5301	0.5301	0.5301	0.6075	0.613	0.882	0.588	0.603	0.603	**0.912**	0.574	**0.912**	**0.912**	0.897	0.691
25	1b64-1fjfJ	0.2719	0.2719	**0.343**	0.3345	0.2946	0.278	0.319	0.236	0	0.125	0.25	0.181	0.222	0.208	0.208	0.194
26	1ash-2fal	0.9592	0.9494	0.9432	0.9494	0.9416	0.944	**0.972**	0.312	0.837	0.851	0.872	0.851	0.894	0.844	0.816	0.773
27	1b5l-1mi	0.9382	0.9211	0.9211	0.9302	0.9279	0.928	0.796	0.718	0.937	0.937	0.831	0.831	0.803	0.937	**0.958**	0.824
28	1b6tA-1f9aA	**0.892**	0.8344	0.8445	0.8649	0.8778	0.865	0.754	0.5	0.535	0.656	0.613	0.528	0.732	0.725	0.627	0.5
29	1d1bB-1i8fA	0.4523	0.4432	0.4432	0.4329	0.4335	0.443	0.836	0.642	0.836	0.836	0.836	0.836	**0.925**	0.91	0.896	**0.925**
30	1qbjA-1dpuA	**0.914**	0.9085	0.8965	0.88	0.8865	0.833	0.883	0.783	0.05	0.783	0.867	0.783	0.867	0.867	0.867	0.783
31	1ah1-1qfoA	0.5839	0.4161	0.4234	0.4375	0.4935	0.457	0.583	0.34	0.534	0.553	0.544	0.534	0.592	**0.621**	0.553	0.553
32	1ayfA-1put	0.9352	0.9259	0.9352	0.9085	0.9246	0.925	0.98	0.959	0.959	0.959	0.98	0.969	0.99	1	0.98	1
33	1aohA-1anu	0.965	0.965	0.965	0.965	0.965	0.965	0.964	0.956	0.949	0.949	0.964	**0.971**	0.956	0.964	0.956	0.964
34	1as8-1rcb	**0.364**	0.2876	0.3401	0.3075	0.3246	0.335	0	0.198	0	0	0	0	0.308	0.253	0.253	0
35	1azpA-1bf4A	0.9254	0.9254	0.9254	0.9254	0.9354	0.915	0.919	0.871	0.903	0.903	0.903	0.903	0.871	0.935	0.919	1
36	1awe-1pls	0.2411	0.0709	0.1702	0.1647	0.0898	0.155	0	**0.562**	0.385	0.458	0.396	0.427	0.448	0.417	0.417	0.448
37	1cc8A-1awv0	**0.907**	0.88	**0.907**	0.8905	0.8775	0.887	0.765	0.471	0.329	0.632	0.5	0.603	0.515	0.529	0.529	0.515
38	1apyB-2gawD	0.7703	0.7432	0.7635	0.7489	0.7545	0.753	0.909	0.894	0.917	0.917	0.917	0.871	**0.932**	**0.932**	0.924	0.924
39	1bb9B-1gfB	0.034	0.0272	0.0272	0.0272	0.0135	0.023	0	0.603	0.781	0.808	0.863	0.781	0.904	0.849	0.822	**0.89**
40	1am7A-31zt	**0.429**	0.1099	0.1648	0.1648	0.254	0.295	0.126	0	0	0	0.0118	0.141	0.141	0.141	0.129	0.0118
	Average	**0.784**	0.7291	0.7595	0.7531	0.7585	0.756	0.7431	0.59168	0.598	0.6501	0.750495	0.69	0.7714	0.7656	0.7221	0.7025

Figure 2: Performance comparison of IDE with other DE mutants and tools

S.No	Sequences	IDE (Time in seconds)	DE/best/1 (Time in seconds)
1	1d3bB_1d3bA	201.4719	224.8816
2	1d3bA_1i8fA	159.5878	214.4048
3	1i8fA_1d3dB	246.4165	251.3185
4	1d8jA_1d8kA	586.4361	556.8759
5	1debA_1fe6A	168.7117	168.22
6	1tvxA_1dokA	176.1288	209.1142
7	1hstA_1dpuA	198.9522	202.3093
8	1qbjA_1dpuA	178.5006	180.2985
9	1fzgD_2eboA	206.2716	218.2812
10	1qywA_1qgwB	330.2048	362.1007

Table2: Comparison of Execution times of IDE & DE/best/1



4 CONCLUSION & FUTURE ENHANCEMENTS

This paper addressed a new IDE which is DE with mutant "DE/current-to-best-worst/1". The performance of evolutionary algorithms depend upon crossover and mutation strategies. This paper aimed to suggest a new mutant for improved efficacy of DE for sequential alignment problem. The efficacy of the operator is studied in terms of total column score (TC-Score) and compared with 5 other DE mutant variants and with another 10 well known sequential alignment methods. The experimental results shows the efficiency of the proposed IDE compared to all other mutants & sequential algorithms.

The work can be extended to MSA problem to discover the previously unknown evolutionary relationships among sequences and to predict the functional & structural similarity of proteins.

References

[1] D.W. Mount, Bioinformatics: Sequence and Genome Analysis, Cold Spring Harbor Laboratory Press, Cold Spring Harbor, NY, 2004.

[2] J. Xiong, Essential Bioinformatics, Cambridge University Press, NY, 2006.

[3] C. Notredame, Recent progresses in multiple sequence alignment: a survey, Pharmacogenomics 3 (1) (2002) 131144.

[4] M.O. Dayhoff, R.M. Schwartz, B.C. Orcutt, A model of evolutionary change in proteins, in: M.O. Dayhoff (Ed.), Atlas of Protein Sequence and Structure, vol. 5, National Biomedical Research Foundation, Washington, DC, 1978, pp. 345352.

[5] S. Henikoff, J.G. Henikoff, Amino acid substitution matrices from protein blocks, Proc. Natl. Acad. Sci. 89 (1992) 1091510919.

[6] Brudno M, Malde S, Poliakov A, Do CB, Couronne O, Dubchak I, Batzoglou S (2003). "Glocal alignment: finding

rearrangements during alignment". Bioinformatics. 19. Suppl 1 (90001): i5462.

[7] R. Durbin, S. Eddy, A. Krogh, G. Mitchison, Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids, Cambridge University Press, Cambridge, UK, 1998.

[8] C. Zhang, A.K. Wong, A genetic algorithm for multiple molecular sequence alignment, Comput. Appl. Biosci. 13 (1997) 565581.

[9] S.B. Needleman, C.D. Wunsch, A general method applicable to the search for similarities in the amino acid sequence of two proteins, J. Mol. Biol. 48(1970) 443453.

[10] T.F. Smith, M.S. Waterman, Identification of common molecular subsequences, J. Mol. Biol. 147 (1981) 195197.

[11] C. Shyu, L. Sheneman, J.A. Foster, Multiple sequence alignment with evolutionary computation, Genet. Prog. Evol. Mach. 5 (2004) 121144.

[12] L. Wang, T. Jiang, On the complexity of multiple sequence alignment, J. Comput. Biol. 1 (1994) 337348.

[13] S.F. Altschul, W. Gish, W. Miller, E.W. Myers, D.J. Lipman, Basic local alignment search tool, J. Mol. Biol. 215 (1990) 403410.

[14] W.R. Pearson, Rapid and sensitive sequence comparison with FASTP and FASTA, Methods Enzymol. 183 (1990) 6398.

[15] C. Kemena, C. Notredame, Upcoming challenges for multiple sequence alignment methods in the high-throughput era, Bioinformatics 25 (2009) 24552465.

[16] C. Notredame, Recent progress in multiple sequence alignment: a survey, Pharmacogenomics 3 (2002) 131144.

[17] J.D. Thompson, D.G. Higgins, T.J. Gibson, CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap

penalties and weight matrix choice, Nucl. Acids Res. 22 (1994) 46734680.

[18] S.B. Needleman, C.D. Wunsch, A general method applicable to the search for similarities in the amino acid sequence of two proteins, J. Mol. Biol. 48 (1970) 443453.

[19] R.C. Edgar, MUSCLE: multiple sequence alignment with high accuracy and high throughput, Nucl. Acids Res. 32 (2004) 17921797.

[20] C. Notredame, D.G. Higgins, J. Heringa, T-coffee: a novel method for fast and accurate multiple sequence alignment, J. Mol. Biol. 302 (2000) 205217.

[21] X. Huang, W. Miller, A time-efficient, linear-space local similarity algorithm, Adv. Appl. Math. 12 (1991) 337357.

[22] Swagatam Das, Member, IEEE, and PonnuthuraiNagaratnamSuganthan, Senior Member, IEEE Differential Evolution: A Survey of the State-of-the-Art, IEEE transactions on evolutionary computation, vol. 15, no. 1, february 2011.