

# Pairwise Sequence Alignment by Differential Evolutionary Algorithm with New Mutation Strategy



Lakshmi Naga Jayaprada.Gavarraju, K. Karteeka Pavan

**Abstract:** Sequence alignment is a significant facet in the bio-informatics research field for the molecular sequence analysis. Arrangement of two biological sequences by maximizing the similarities between the sequences by incorporating and adjusting gaps is Pairwise Sequence Alignment (PSA). Arrangement of multiple sequences is Multiple Sequence Alignment (MSA). Though Dynamic programming can produce optimal sequence alignment for PSA it suffers from a problem when multiple optimal paths are present and trace back is required. Back tracking becomes complex and it is also not suitable for MSA. So many meta-heuristic algorithms like Genetic Algorithm (GA) and Differential Evolutionary Algorithm (DE) are developed in the recent years to resolve the issue of optimization. Both GA and DE are used to produce optimal sequence alignment. But Compared to GA, DE is able to produce more optimal sequence alignment. To further enhance the performance of DE a new mutant is proposed by considering best, worst and a random candidate solution and applied on DE. It is named as New Differential Evolutionary Algorithm (NDE). By taking the test sequences from a bench mark data set “prefab4ref” tests are performed on GA, All DE mutants and NDE and it is observed that the proposed algorithm NDE outperformed all the other algorithms.

**Keywords:** Sequence Alignment, Biological Sequences, Pairwise Sequence Alignment, Multiple Sequence Alignment, Genetic Algorithm, Differential Evolutionary Algorithm.

## I. INTRODUCTION

Biological Informatics, in brief Bioinformatics is a combination of Biology, Computer Science and Information Technology. Now-a-days it is called as Computational Biology by many scientists. In order to solve so many biological problems, Scientists are continuously striving to design new algorithms [1]. Many bioinformatics tools and databases were designed and developed by scientists to analyse the biological data and to store the biological information. Bioinformatics covers so many areas like genetics, proteomics etc. One of the most prominent applications of the Bioinformatics is Sequence Analysis and Sequence Alignment. Sequence Alignment is mainly for the determination of analogous regions with in the specified

biological sequences like nucleotide or protein sequences. Identifying the analogous areas within the specified biological sequences is for the purpose of finding functional similarity or structural similarity or to evolve evolutionary relationships among the specified sequences. Sequence alignment is majorly of two varieties based on number of sequences: Pair-wise and Multiple Sequence Alignments. Aligning two biological sequences is called PSA and Aligning multiple biological sequences is called MSA. Sequence Alignment is divided into two categories depending on the type of alignment: Global [2] and Local [3] Sequence Alignments. Performing sequence alignment on amino acid sequences is more appropriate than performing the sequence alignment on nucleotide sequences. It is because amino acid sequences (protein sequences) consist of functional and structural information [4]. Many scoring functions are utilized to find the similarity or identity among the sequences. When comparing the nucleotide sequences a simple scoring function called Identity Score (IS), can be used, where similar nucleotide bases is assigned a positive score, dissimilar a negative score and for a gap less negative score is assigned. Another Scoring function called Column Score (CS) can also be used, in which identical nucleotide bases are present in a single column a value of ‘1’ is assigned otherwise a value of ‘0’ is assigned. For protein sequences another scoring function called Similarity Score (SS) can also be used along with scoring functions like IS and CS. In SS, amino acids with analogous physiochemical properties are assigned a value based substitution matrices like Point Accepted Mutation (PAM) [5] and BLOcked Substitution Matrix (BLOSUM) [6]. A variety of PSA techniques are available to produce best alignment to two given sequences both local and global. To produce optimal alignment of the two given sequences, previously Dynamic programming was used. A Dynamic programming algorithm “Smith-Waterman algorithm” [7] is used for local sequence alignment and “Needleman-Wunsch algorithm” [8] is used for global sequence alignment. Both the algorithms suffer from a drawback specifically when two or many optimal paths are generated and trace back is needed. Back tracking becomes complex [9, 10]. So, many scientists tried to use nature inspired optimization algorithms. Genetic Algorithm (GA) is an optimization algorithm to solve the problem of sequence alignment to produce the optimal alignment. A multi objective GA was developed by Taneda for PSA of RNA sequence alignment [11]. Notredame et.al., developed an algorithms for the optimal alignment of RNA sequences by GA called RAGA and another algorithm Parallel GA (PRAGA) [12]. Cedric Notredame and Desmond G.

Revised Manuscript Received on December 30, 2019.

\* Correspondence Author

**Lakshmi Naga Jayaprada. Gavarraju**, Assoc.Prof, Dept. of Computer Science & Engineering, Narasaraopeta Engineering College [Autonomous], Narasaraopet, Guntur(Dt), A.P., India.

**Kanadam Karteeka Pavan**, Professor & Head Department of Computer Applications, R.V.R.& J.C.College of Engineering [Autonomous], Chowdavaram, Guntur, A.P., India.

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](http://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

# Pairwise Sequence Alignment by Differential Evolutionary Algorithm with New Mutation Strategy

Higgins developed a package called SAGA: sequence alignment by genetic algorithm [13] which is useful for finding globally optimal multiple alignments in reasonable time. In 1995, R. Storn and K.

V. Price developed an evolutionary algorithm called Differential Evolutionary Algorithm (DE) [14-16]. The algorithm consists of two major steps known as mutation and cross over. Biologically, Mutation is an unexpected modification in the chromosome with respect to gene characteristics. The crossover operation plays an important role to improve the potential diversity of the population [17]. Storn and Price specified 5 mutants to perform mutation operations and 2 types of cross over operations [18-19]. Many scientists used DE to solve the problem of PSA and MSA. This work recommends an innovative mutant to DE. The mutant is specified using best, worst chromosomes and a random chromosome in mutation process from the existing population to maintain the convergence speed by making best use of the search space and named as New Differential Evolutionary Algorithm (NDE). PSA is done with NDE, GA and all other DE mutants with different Fitness functions: SS, CS and IS and recorded the average fitness scores. In terms of average fitness scores NDE performed well than all the other algorithms for all the three fitness functions SS, CS and IS. ADE performed well for 34% times with fitness function "SS", 39% times with fitness function "CS" and 38% times with fitness function "IS". To perform experiments on the above specified algorithms bench mark datasets from "prefab4ref" are taken into account. The work is organized in 6 segments. Segment I is introduction. Segment II explains Pair-wise Sequence Alignment and Fitness functions. Segment III enlightens proposed DE mutant and Differential Evolutionary Algorithm. Segment IV demonstrates the Results and Discussions. Segment V focuses on Limitations and at the end Segment VI comprises conclusions and future accomplishments.

## II. PAIRWISE SEQUENCE ALIGNMENT AND FITNESS FUNCTIONS

Pair-wise sequence alignment is the problem of aligning two protein sequences or two nucleotide sequences by incorporating gaps at some random positions for maximizing the similarity in the given two sequences. Major application of the sequence alignment is to find evolutionary relationships among the sequences. The subsequent sections will explain how the sequence alignment can be achieved and also the scoring functions available to find the fitness values.

### A. Alignment of Sequences

To perform pair-wise sequence alignment a test data set is considered from a bench mark data set prefab4ref. Figure 1 shows the input test data set, one test sequence "1tafA" sequence is having the length 66 characters and the second sequence "1a7w" sequence is having the length of 68 characters.

```
>1tafA
KDAQVIMSILKELNVQEYEPVVNQLEFFRYVTSILDDAKV
YANHARKKTTIDLDDVRLATEVTL
>1a7w
MELPIAPIGRRIKDAGAERVSSDARITLAKILEEMGRDIASEAI
KLARHAGRKTIKAEDIELAVRRFK
```

Figure 1: Test Sequences

For the alignment sequences Gaps are inserted at some random positions. When the population size is considered as 5 the population is as shown in Figure 2 below. Gaps are inserted in the test sequences at the specified positions in the population.

```
population(:,1) =
78 52 41 60 63 40 17 61 23 6 49 36 31 28 58 26
51 42 38 74 41 57 44 65 39 23 53 82 70 9 0 0
population(:,2) =
18 65 12 64 50 16 73 11 33 39 56 58 47 28 61 74
75 37 82 42 67 26 10 29 16 61 52 6 30 77 0 0
population(:,3) =
9 25 22 56 75 23 78 19 1 52 24 48 10 7 34 15
77 18 72 76 5 66 60 14 8 41 48 23 33 70 0 0
population(:,4) =
29 31 3 72 45 6 77 69 37 52 67 17 74 56 58 15
69 8 49 32 40 71 36 68 81 35 6 46 47 56 0 0
population(:,5) =
50 37 19 13 73 56 76 8 44 55 22 34 61 80 72 58
68 11 24 23 67 56 22 29 58 12 69 41 76 49 0 0
```

Figure 2: Population

Figure 3 is the sample alignment of the two input test sequences.

```
KDAQV-IMS-I-LKELN---VQEY-EPVVNQLEF-TFR-YVT-SILDD-AKVYANHARKK--TIDLD-DVRLATEV-TL
MELPIAPIGRRIKDAGAER-----RVSSDARITLAKI-LEEMG-RDIASEA-IKL-AR-HA-GRKTIKAED-IELAVRR-FK
```

Figure 3: Alignment of Sequences

### B. Fitness Functions

Many fitness functions or objective functions can be used to calculate the scores while performing sequence alignment. They are Identity Score, Similarity Score and Column Score.

#### Identity Score (IS):

$$IS = \sum_{b=1}^P \sum_{c=1}^{Q-1} \sum_{a=c+1}^Q Score(Xcb, Xab) \quad (1)$$

Where

$$Score(Xcb, Xab) = \begin{cases} \text{match if } Xcb = Xab \text{ and } Xcb \neq '-' \text{ and } Xab \neq '-' \\ \text{gap if } Xcb = '-' \text{ or } Xab = '-' \\ \text{mismatch if } Xcb \neq Xab \end{cases} \quad (2)$$

where '-' means a gap; match is given value of '+2', gap is given value of '-2' and mismatch is given value of '-1'.

#### Similarity Score (SS):

$$SS = \sum_{b=1}^P \sum_{c=1}^{Q-1} \sum_{a=c+1}^Q Score(Xcb, Xab) \quad (3)$$

where

$$Score(Xcb, Xab) = \begin{cases} \text{match if } Xcb = Xab \text{ and } Xcb \neq '-' \text{ and } Xab \neq '-' \\ \text{gap if } Xcb = '-' \text{ or } Xab = '-' \end{cases} \quad (4)$$

where similarity score is generated depending on PAM score matrix or BLOSUM score matrix and gap=0.

#### Column Score (CS):

$$CS = \sum_{a=1}^Q Ca \quad (5)$$

If all the nucleotide residues or amino acids in the a<sup>th</sup> column are aligned after sequence alignment then Ca=1 otherwise Ca=0

III. PROPOSED DE MUTANT

By the inspiration of development of new optimization algorithms to generate more optimal alignment a new algorithm called New Differential Evolutionary algorithm (NDE) was proposed built on a new DE mutant. NDE is explained in the next sub section.

A. DE Mutant

The proposed DE mutant is as shown below and is named as NDE.

$$Y_{k,G} = U_{r1,G} + p(U_{b,G} - U_{k,G}) + q(U_{w,G} - U_{k,G}) \quad (6)$$

where  $U_{b,G}$  is best fit chromosome,  $U_{w,G}$  is worst fit chromosome,  $U_{k,G}$  the present chromosome,  $U_{r1,G}$  is the random chromosome considered from the current population with random number  $r1$ ,  $Y_{k,G}$  is the newly generated chromosome and  $p$  and  $q$  are two random numbers between 0 and 1.

B. Differential Evolutionary Algorithm

1. Control parameters population size (S), scale factor (F) and cross-over rate (Cr) is to be initialized.
  2. Based on population size initialize population.
  3. Repeat the successive steps until the stated number of generations is completed.
- For  $i=1$  to S
4. Find the fitness of every chromosome in the population with a fitness function.
  5. Mutation: In mutation the variance of two vectors is added to the third vector.

$$V_{i,G+1} = X_{r1,G} + F(X_{r2,G} - X_{r3,G})$$

where F is the mutation factor. It is between 0 and 1.

6. Crossover: In crossover find a trial vector  $P_{i,G}$  for the  $i^{th}$  target vector  $Q_{i,G}$  by applying binomial crossover.
7. Selection: Assess trial vector  $P_{i,G}$

$$\begin{aligned} &\text{if } f(P_{i,G}) \leq f(Q_{i,G}), \text{ then } Q_{i,G+1} = P_{i,G} \\ &\text{else } Q_{i,G+1} = Q_{i,G} \\ &\text{end if} \end{aligned}$$

end for

IV. RESULTS AND DISCUSSIONS

The performance of proposed algorithm “NDE” is examined using three different fitness functions like SS, CS, and IS. To evaluate the performance the proposed algorithm is compared with all DE mutants DE/rand1, DE/best1, DE/rand2, DE/best2, DE/current-to-best1 and GA. In this work, all the above specified algorithms are implemented in MATLAB and checked on the System with Intel’s i3 processor with 4 Giga Byte RAM with 3.70 GHz and Windows 8.1 Operating System. Tests are conducted for pairwise sequence alignment using bench mark data sets available in prefab4ref.

A. PSA Using Similarity Score

Similarity Scores of the NDE, DE with all its mutants and GA is calculated using prefab4ref bench mark data sets. 56 data sets are considered to find Similarity Score for all the above specified algorithms. Average Similarity Scores are presented in table 1 as demonstrated below.

Table 1: Performance Comparison for PSA based on Average Similarity Scores

S.No	Name of the Sequences	GA	DE/ rand1	DE/ best1	DE/ rand2	DE/ best2	DE/ current-to-best1	NDE
1	1d2zA-1alz	43.2	41.9	39.8	36.2	39	35.8	<b>43.7</b>
2	1a3aA-1a6jA	58	76.5	<b>79.8</b>	70.6	79.2	61.6	72
3	1a3k-1c11A	12.2	50.9	49.2	44.6	43.8	38.4	<b>59.2</b>
4	1a3k-1lcl	30	87.9	95.8	92.8	82.4	<b>105.8</b>	91.4
5	1a6m-1ash	17.1	20.5	<b>22.6</b>	14.4	20.4	24	19.8
6	1a6m-1cg5B	35.1	54.1	57	46.2	49.8	<b>59.6</b>	52.1
7	1a6m-1ewaA	-13.9	13.2	23.2	<b>33</b>	29	19.2	27.6
8	1a6m-1flP	<b>49.2</b>	36.1	37.6	32	33.6	30.6	32.8
9	1a6m-1h97A	68.9	63.9	66.4	49.4	60.2	67.8	<b>71.3</b>
10	1a6m-1h1b	<b>76.4</b>	55.2	47.8	55.6	61.4	50.4	50.8

## Pairwise Sequence Alignment by Differential Evolutionary Algorithm with New Mutation Strategy

11	1a6m-1ithA	9	30.3	35.4	33.4	35.6	<b>36.4</b>	31.3
12	1a6m-2fal	37.3	61.2	65.6	53.6	54.8	<b>78</b>	55.7
13	1a6m-2gdm	30.7	51.6	<b>104.4</b>	81	79	86.2	50.9
14	1a6m-2hbg	39.4	67.9	75.2	66.8	78.6	<b>81.2</b>	66.9
15	1a6m-2vhbA	13.4	36.3	33.2	31.6	31	29	<b>38.6</b>
16	1a6m-3sdhA	40.4	49.3	48.6	39.8	48.4	48.4	<b>50.8</b>
17	1a7tA-1sm1A	-12.8	13.4	13.4	16.6	20.2	9.4	<b>22.7</b>
18	1a0iA-1a7w	6.1	33.1	33.8	<b>37.8</b>	34.8	30	34.3
19	1a0iB-1a7w	29.4	40.3	35	30.8	29.8	32.8	<b>40.6</b>
20	1a7w-1b67B	144.8	160.8	<b>179.2</b>	148.4	161.6	154	153.6
21	1a7w-1bh9B	14.8	28	27	21.6	25.4	22.2	<b>28.1</b>
22	1tafA-1a7w	36.7	50.5	<b>58.2</b>	46.6	55.8	50.8	52.7
23	1a7w-1tafB	43.9	61	<b>77.6</b>	56.2	56.6	56.8	61.1
24	1a8l-1hyuA	8.5	47.2	45	35.8	<b>54.4</b>	37	51.3
25	1a9nA-1dceA	16.6	20.8	18.8	20	19	20	<b>21.8</b>
26	1a44-1qouB	-26.8	15.3	9	6.4	12.6	8.4	<b>19.6</b>
27	1a53-1dvjA	-22.9	14.9	5.8	11.8	11.2	22	<b>20.7</b>
28	1a62-1mjc	11.1	24.4	19.8	21.2	25.3	25.3	<b>25.4</b>
29	1a75A-1b8cB	152.5	216	<b>251.2</b>	213.4	211.8	220.4	219.9
30	1aba-1h75A	23.9	59.9	<b>60.6</b>	60.4	50.6	49.4	57.6
31	1aba-1kte	23.2	32.4	31.8	26.8	31.2	25.4	<b>33.4</b>
32	1aba-1qfnA	19.3	48.7	<b>54.6</b>	40	45.2	38	47.1
33	1aba-3grx	17.7	54.4	49.4	48	52.6	45.6	<b>54.8</b>
34	1aerA-1aerB	<b>386.9</b>	186.8	233	198.2	227.8	240.6	177.7
35	1agdB-1iakA	89.8	122.3	145.6	111.2	117.8	110	<b>122.7</b>
36	1agqA-1vpfA	-18.8	11.5	11.8	5.4	1.2	<b>12.8</b>	10.6
37	1ahl-1cid	-13.6	<b>-1.1</b>	-9	-8.4	-10.4	-1.4	-8.1
38	1ahl-1f97A	-7.5	10.7	<b>11.4</b>	6.4	11.2	7.6	6.6
39	1ah1-1kb5B	-12.6	-5	-3.6	-11.6	-86	-10.4	<b>-1.7</b>
40	1ah1-1qfoA	-4.5	4.1	4.4	0.8	6.8	6.2	<b>7.5</b>
41	1ah1-1t1k	3.8	<b>7.2</b>	2.6	-1.8	-1.4	2.8	5.3
42	1ah1-2ncm	-9	6.3	17	17.2	8.8	<b>17.4</b>	2.6
43	1ajqA-1cp9A	119.8	182.2	<b>443.4</b>	202.2	214.6	220.2	191.2
44	1alu-1bgc	9.5	42.7	46.8	36	40.6	<b>51.6</b>	46.7
45	1alu-1ent3	22.3	28.4	<b>33.4</b>	25.6	25.8	25.8	27.5
46	1alu-1ilrB	58.5	<b>82.3</b>	28.4	34.8	26.4	27.8	79.4
47	1aly-1d4rB	84.4	107.3	<b>117.8</b>	81	90.2	91.2	95.9
48	1aly-1dyoA	17.5	17.8	<b>20.2</b>	6	16	7.6	<b>20.2</b>
49	1aly-2tnfA	54.9	81	<b>107</b>	75.6	90.8	86	82.2
50	1am7A-31zt	-22.4	<b>-5</b>	-10.2	-11.6	-9.2	-7.6	-9
51	1an8-1aw7A	-4.9	13.4	<b>27.4</b>	13.2	8.4	18.6	13.6
52	1an8-1eu4A	10.5	67	64	69	<b>75.8</b>	67.4	67.3
53	1an8-1f77B	68.5	99.4	<b>125.4</b>	87.8	75.4	95.6	96.2

54	1aohA-1anu	191	234.7	<b>242.2</b>	200	220	207.6	230.4
55	1aohA-1g1kB	152.1	139.5	144.4	135.8	144.8	<b>164.6</b>	135.1
56	1aoiA-1bh9B	31.4	38.1	27	26.4	32	34.4	<b>38.2</b>

above specified algorithms. Average Column Scores are presented in tables 2 as demonstrated below.

**B. PSA Using Column Score**

Column Scores of the NDE, DE with all its mutants and GA is calculated using prefab4ref bench mark data sets. 56 data sets are considered to find Column Score for all the

**Table 2: Performance Comparison for PSA based on Average Column Scores**

S.No	Name of the Sequences	GA	DE/ rand1	DE/ best1	DE/ rand2	DE/ best2	DE/ current-to-best1	NDE
1	1d2zA-1alz	11.3	<b>14.2</b>	13.8	13.8	13.4	13.5	13.9
2	1a3aA-1a6jA	22.9	25.7	25.5	24.5	25.3	24.1	<b>26.1</b>
3	1a3k-1c11A	17.5	<b>20.2</b>	20.1	18.9	19.7	18.7	19.8
4	1a3k-1lcl	20.5	<b>24.8</b>	24.4	23.8	23.9	24.3	24.5
5	1a6m-1ash	<b>20.1</b>	19.6	19.1	19.4	18.7	18.2	18.8
6	1a6m-1cg5B	20.9	22.8	22.1	21.7	21.7	21.6	<b>22.9</b>
7	1a6m-1ewaA	14.1	16.1	15.6	16.2	16.2	15.5	<b>16.3</b>
8	1a6m-1flP	17.1	20	<b>20.6</b>	19.3	19.8	20.5	19.6
9	1a6m-1h97A	19.1	23.8	24.1	23	24.1	23.6	<b>24.5</b>
10	1a6m-1h1b	16.9	18.7	18.5	18	18.4	18.1	<b>18.8</b>
11	1a6m-1ithA	15.7	18	18	18.1	<b>18.5</b>	18.1	17.9
12	1a6m-2fal	25.1	23.7	27.4	26.7	26.6	<b>27.7</b>	26
13	1a6m-2gdm	21.8	22.8	22.7	22.7	22.1	22.5	<b>23</b>
14	1a6m-2hbg	21.1	23.6	24.3	22.5	24.1	<b>24.7</b>	23.8
15	1a6m-2vhbA	19.7	<b>21.6</b>	20.4	20.5	20.4	20	20.6
16	1a6m-3sdhA	18.4	19.5	19.8	19.1	19.3	19.6	<b>19.9</b>
17	1a7tA-1sm1A	18.3	22.1	21.6	21	21.6	22.1	<b>22.2</b>
18	1a0iA-1a7w	13	<b>15.4</b>	14.4	13.3	14.4	14.3	14.5
19	1a0iB-1a7w	12.8	13	13.3	12.7	12.8	12.6	<b>13.9</b>
20	1a7w-1b67B	29.8	37.4	<b>42.5</b>	35.1	38	36.7	35.9
21	1a7w-1bh9B	7.8	10.5	9.8	10.8	10.2	10.1	<b>11.6</b>
22	1tafA-1a7w	14.1	15.2	<b>16.2</b>	14.9	15.7	15.5	15.3
23	1a7w-1tafB	11.6	14.9	<b>16.3</b>	14.4	15.1	14.2	15.1
24	1a8l-1hyuA	22.2	27.2	27.5	25.9	26.9	26.4	<b>28.2</b>
25	1a9nA-1dceA	17.2	18.4	17.3	17.2	17.8	17.3	<b>18.9</b>
26	1a44-1qouB	15.8	19.8	19.4	19.2	19.8	19.6	<b>19.9</b>
27	1a53-1dvjA	18.1	22.4	<b>22.9</b>	22.2	22.5	22.3	22.4
28	1a62-1mjc	11.3	11.7	11.5	11	11.7	11.8	<b>12</b>
29	1a75A-1b8cB	40.3	51.1	53	48.5	<b>50.7</b>	49	49.4
30	1aba-1h75A	10	13.7	<b>14.5</b>	13.8	14	13.6	14.2
31	1aba-1kte	<b>15</b>	14.9	14.2	14	14.1	14.2	14.5
32	1aba-1qfnA	12	14.6	<b>15.7</b>	13.2	15.3	13.3	14.2
33	1aba-3grx	13.6	<b>15.9</b>	15.8	14.7	15.6	14.6	15.3



## Pairwise Sequence Alignment by Differential Evolutionary Algorithm with New Mutation Strategy

34	1aerA-1aerB	<b>78.6</b>	63.25	66.9	59.6	65.8	68.5	61.4
35	1agdB-1iakA	19.9	21.2	<b>21.7</b>	20.1	21.3	21.3	20.9
36	1agqA-1vpfA	8.8	11.1	11.1	10.6	11	10.4	<b>11.2</b>
37	1ahl-1cid	14.1	15.9	15.6	15.2	14.7	14.9	<b>16.2</b>
38	1ahl-1f97A	14.7	17.3	17.2	17	17.1	16.7	<b>17.7</b>
39	1ah1-1kb5B	11	<b>14.1</b>	13.4	13.4	13.1	13.5	13.4
40	1ah1-1qfoA	11.2	14.2	<b>14.9</b>	14	14.41	14.8	14.3
41	1ah1-1t1k	10.8	13.3	13.1	12.7	13.1	12.8	<b>13.5</b>
42	1ah1-2ncm	11.6	13.1	13.1	12.6	12.9	12.9	13.3
43	1ajqA-1cp9A	<b>66.8</b>	53.4	56.1	50.7	53.6	57.6	53.2
44	1alu-1bgc	18.3	22.3	22.4	21.8	22.1	22.7	<b>23</b>
45	1alu-1ent3	18.4	18.7	19.2	18.3	19.1	<b>19.6</b>	18.8
46	1alu-1ilrB	17.5	25.1	25.4	24.2	24.4	24.5	<b>26</b>
47	1aly-1d4rB	20.4	<b>28.3</b>	26.9	25.9	26.2	25.4	26.6
48	1aly-1dyoA	15.8	18.2	18.3	17.4	17.9	17.4	<b>18.4</b>
49	1aly-2tnfA	19.4	<b>25.6</b>	25	24.8	25.1	25.4	24
50	1am7A-31zt	13.1	<b>17.5</b>	16.8	17.3	16.8	16.6	17.2
51	1an8-1aw7A	24.2	<b>24.3</b>	23.9	23.4	23.9	24.1	23.7
52	1an8-1eu4A	29.3	<b>31.6</b>	30.3	28.9	30	30.6	31.2
53	1an8-1f77B	27.4	34	<b>34.2</b>	33	32.9	32.7	33.4
54	1aohA-1anu	44.9	52.1	55.2	48.6	51.5	50.8	<b>53.8</b>
55	1aohA-1g1kB	28.9	28.5	29.1	27.4	29	<b>30.5</b>	29.2
56	1aoiA-1bh9B	11.5	<b>13.1</b>	12.8	12.3	12.4	12.8	12.6

### C. PSA Using Identity Score

Identity Scores of the NDE, DE with all its mutants and GA is calculated using prefab4ref bench mark data sets. 56 data sets are considered to find Identity Score for all the

above specified algorithms. Average Identity Scores are presented in tables 3 as demonstrated below.

**Table 3: Performance Comparison for PSA based on Average Identity Score**

S.No	Name of the Sequences	GA	DE	DE/ best1	DE/ rand2	DE/ best2	DE/ current- to-best1	NDE
1	1d2zA-1alz	-138.3	-128.4	-129.5	-131.8	-129.7	-131.1	<b>-128.2</b>
2	1a3aA-1a6jA	-155.4	-154	-152.6	-154.7	-151.9	-155	<b>-151.3</b>
3	1a3k-1c11A	-174.5	-159.2	-162.1	-165	-162.8	-163.1	<b>-161.1</b>
4	1a3k-1lcl	-156.2	-142.2	-143.3	-146.6	-144.9	-144.3	<b>-140.8</b>
5	1a6m-1ash	-190.4	-189.1	-189.5	-191.6	-187.9	-187.9	<b>-187.8</b>
6	1a6m-1cg5B	-187	-176.8	-176.5	-181.4	-180.4	<b>-175.2</b>	-177.2
7	1a6m-1ewaA	-208.4	-202	-202.3	-205.2	-204.5	-203.8	<b>-201.9</b>
8	1a6m-1flP	-196	-186.4	-186.3	-186.7	-190.5	-187.5	<b>-186.1</b>
9	1a6m-1h97A	-178.6	-170.1	<b>-167.4</b>	-174.8	-172.9	-170.8	-173.8
10	1a6m-1h1b	-184.1	-182.9	-183.8	-184.2	-185.8	-183	<b>-181.4</b>
11	1a6m-1ithA	-197.2	-188.7	-193	-192.5	-189.1	-190	<b>-188.1</b>
12	1a6m-2fal	-170.1	-162.4	-161	-163	-160.4	<b>-153.5</b>	-160.6
13	1a6m-2gdm	-179.3	-170.6	-177	-172	-170.7	-169.9	<b>-169.5</b>

14	1a6m-2hbg	-189.9	-172.2	-178	-174.5	-174.5	<b>-171.2</b>	-175
15	1a6m-2vhbA	-194.2	<b>-186.9</b>	-196	-192.4	-189.9	-191.4	-188.2
16	1a6m-3sdhA	-193.7	<b>-186</b>	-192	-189.3	-188.5	-187	-188.8
17	1a7tA-1sm1A	-325.1	<b>-311.1</b>	-320	-313.5	-313.4	-312.9	-312.6
18	1a0iA-1a7w	-139.2	-129.1	-135	-130.6	-130.2	-129.9	<b>-127.9</b>
19	1a0iB-1a7w	-107.4	-100.4	-102	-100.9	<b>-99.1</b>	-100.7	-101.3
20	1a7w-1b67B	-12.5	3.7	<b>7</b>	-3.5	5.4	5.2	2.5
21	1a7w-1bh9B	-127	-121.2	-126	-123	-123.7	-123.8	<b>-120.4</b>
22	1tafA-1a7w	-64.6	-55.8	<b>-50.4</b>	-60.1	-56.8	-56.5	-58
23	1a7w-1tafB	-73.7	-58.6	<b>-54.2</b>	-62.4	-59.6	-61.9	-60.1
24	1a8l-1hyuA	-300.1	-295.1	-293.9	-297.3	-295.9	-294.5	<b>-293.8</b>
25	1a9nA-1dceA	-234.3	-228	-230.8	-260.5	-229.9	-230.9	<b>-225.9</b>
26	1a44-1qouB	-265.6	-259	259.3	-258.5	-257.7	-260.4	<b>-256.6</b>
27	1a53-1dvjA	-371.4	-365.9	-371.3	-368.2	-369.1	<b>-365.8</b>	-366.2
28	1a62-1mjc	-213.8	<b>-203.9</b>	-207	-206.2	-205.1	-207.9	-204.7
29	1a75A-1b8cB	-12.6	-16.1	<b>-5.3</b>	-25.3	-18.4	-16.5	-16.4
30	1aba-1h75A	-103.9	-101	-102.3	-105.5	<b>-96</b>	-100	-99.7
31	1aba-1kte	-134	-131.3	-132.6	-134.1	-132.5	-134.7	<b>-129.3</b>
32	1aba-1qfnA	-102.1	<b>-93</b>	-93.2	-97.4	-95.3	-98.5	-93.9
33	1aba-3grx	-100.1	-90.2	-90.1	-94.2	-92.4	93.4	<b>-89.9</b>
34	1aerA-1aerB	-114.9	-145.4	<b>-110.7</b>	-145.2	-117.8	-124.2	-137.7
35	1agdB-1iakA	-91.2	-88.7	<b>-85.7</b>	-93.7	-92	-89.2	-91.9
36	1agqA-1vpfA	<b>-115.1</b>	-117.6	-116.3	-117.9	-118.4	-116.2	-116.2
37	1ahl-1cid	-274.5	-267.9	-269.6	-269.3	-269.5	-269.3	<b>-265.8</b>
38	1ahl-1f97A	-333	<b>-330.5</b>	-333.1	-333.8	-333.6	-332.9	-331.6
39	1ah1-1kb5B	-170.3	<b>-166.5</b>	-167.9	-169.7	-170	-170.2	-167
40	1ah1-1qfoA	-172	-165.5	-166.7	-168.5	-165.5	<b>-165.1</b>	-167.8
41	1ah1-1tlk	-186.9	<b>-180.6</b>	-183.8	-185.6	-183.5	-184.8	-182.6
42	1ah1-2ncm	-192.4	<b>-183.2</b>	-187.3	-187.6	-188	-189	-186.7
43	1ajqA-1cp9A	<b>-141.9</b>	-168.7	-152.6	-178.7	158.3	-153.2	-163.1
44	1alu-1bgc	-201.1	-191.1	-188.2	-191.8	-188.4	<b>-186.5</b>	-190.2
45	1alu-1ent3	-207.7	-210.2	-209.3	-208.7	-209.2	<b>-206.8</b>	-207.6
46	1alu-1lirB	-181.4	-170.8	<b>-169</b>	-173.1	-171.4	-170.6	-170.4
47	1aly-1d4rB	-178.4	-172.1	<b>-157.2</b>	-159.3	-157.4	-160	-168.5
48	1aly-1dyoA	-203.4	<b>-193.1</b>	-195.8	-198.2	-198.3	-197.6	-195.3
49	1aly-2tnfA	-169.5	-154.6	-157.1	-159.4	-159.5	-160.6	<b>-154.1</b>
50	1am7A-31zt	-216.1	<b>-208.7</b>	-208.8	-211.9	-211.5	-211	-209.8
51	1an8-1aw7A	-272.2	-265.2	-265.3	-267.4	-267.1	-265.4	<b>-264.3</b>
52	1an8-1eu4A	-243.9	<b>-237.1</b>	<b>-237.1</b>	-244.2	-240.9	-242.6	-237.9
53	1an8-1f77B	-247	-238.2	-228.4	-233.3	<b>-227.2</b>	-227.3	-235.4
54	1aohA-1anu	-70.8	<b>-60.9</b>	-65.4	-73.1	-73.5	-71.4	-71.3
55	1aohA-1glkB	-155.7	-140.4	-138.7	-144.2	-140.9	<b>-135.6</b>	-142
56	1aoiA-1bh9B	-122.7	-118.3	-119.7	-120.1	-119.9	-120.3	<b>-116.9</b>

#### D. Ddiscussions

NDE, DE with all mutants and GA are applied on 56 bench mark data sets available in prefab4ref and calculated the Average Similarity Scores of the respected data sets and tabulated in Table 1. It is observed that NDE performed better than all the other DE mutants and GA. NDE has shown better

performance 19 times among 56 test cases. i.e., 34% times NDE is showing better performance. Frequency of best performance for PSA based on Average Similarity Scores is shown in Table 4 and depicted Graph 1.



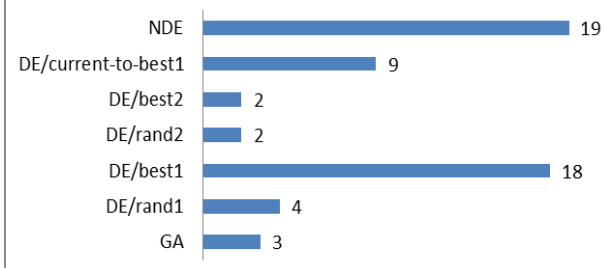
**Table 5: Frequency of best performance for PSA based on Average Column Scores**

GA	4
DE/rand1	13
DE/best1	10
DE/rand2	0
DE/best2	2
DE/current-to-best1	4
NDE	22

**Table 4: Frequency of best performance for PSA based on Average Similarity Scores**

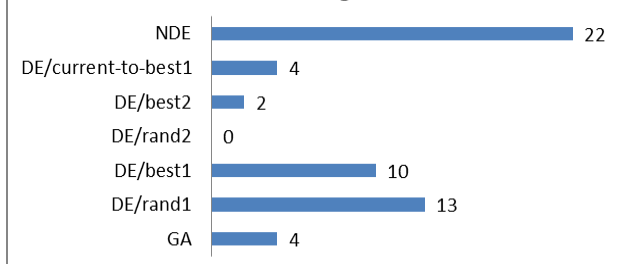
GA	3
DE/rand1	4
DE/best1	18
DE/rand2	2
DE/best2	2
DE/current-to-best1	9
NDE	19

**Graph 1: Frequency of best performance for PSA based on Average Similarity Scores**



NDE, DE with all mutants and GA are applied on 56 bench mark data sets available in prefab4ref and calculated the Average Column Scores of the respected data sets and tabulated in Table 2. It is observed that NDE outperformed all the other algorithms. NDE has shown better performance 22 times among 56 times. i.e., NDE has shown better performance 39% times. Frequency of best performance for PSA based on Average Column Scores is shown in Table 5 and depicted in Graph 2.

**Graph 2: Frequency of best performance for PSA based on Average Column Scores**

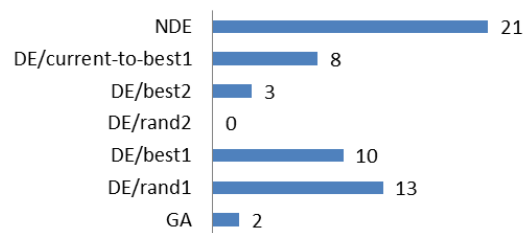


NDE, DE with all mutants and GA are applied on 56 bench mark data sets available in prefab4ref and calculated the Average Identity Scores of the respected data sets and tabulated in Table 3. It is observed that NDE performed better than all the other DE mutants and GA. NDE has shown better performance 21 times among 56 test cases. i.e., 38% times NDE is showing better performance. Frequency of best performance for PSA based on Average Identity Scores is shown in Table 6 and depicted Graph 3.

**Table 6: Frequency of best performance for PSA based on Average Identity Scores**

GA	2
DE/rand1	13
DE/best1	10
DE/rand2	0
DE/best2	3
DE/current-to-best1	8
NDE	21

**Graph 3: Frequency of best performance for PSA based on Average Identity Scores**



In general Average Fitness values are considered for all the optimization algorithms. In this work the pair-wise sequence alignment is performed using three fitness functions: Similarity Score, Column Score and Identity Score. When pair-wise sequence alignment is done using Similarity Score fitness function on 56 test cases, 34% times the proposed algorithm NDE performed well for Average Similarity Scores. In case of Column Score fitness function also NDE performed well 39% times for Average Column Score. In case of Identity Score fitness function also NDE performed well 38% times for Average Identity Score.

**V. LIMITATIONS**

This work is on PSA only. All experiments are conducted by considering only two sequences and a maximum length of 300 characters. If it is applied on multiple sequences and long length sequences may be it consumes more execution times. The work can also be extended to MSA.

**VI. CONCLUSIONS AND FUTURE ACCOMPLISHMENTS**

In computational biology sequence alignment is having many applications. Alignment of two biological sequences is known as PSA and alignment of three or more biological sequences is known as MSA. Initially the alignment of sequences is done manually.



Later Dynamic Programming was used. It is really capable of producing optimal alignment. But when two are more optimal paths are produced trace back is required. Then back tracking becomes cumbersome. Then scientists tried to use nature inspired optimization algorithms to solve the problem of sequence alignment. GA is one such algorithm which is used to solve the problem of sequence alignment and is able to produce optimal alignment. Later another optimization algorithm DE came into existence and it a real parameter optimization method with few control parameters. The two major steps of DE are mutation and cross over. Storn and Price proposed 5 mutants to perform mutation operation and 2 cross over operators.

This paper proposes a new DE mutant and its performance is checked by making use test sequences from familiar benchmark data set called “prefab4ref” Tests are conducted on GA, All DE mutants and the proposed DE mutant. Three objective functions or fitness functions are used to calculate the performance of all the above specified algorithms.

The proposed algorithm “New Differential Evolutionary Algorithm” outperformed all the other algorithms. This study is done on PSA. Later this can be extended to MSA. In this paper three fitness functions IS, SS and CS are chosen to evaluate the performance. There are so many other fitness functions. So, the work may be extended with other fitness functions also. To further improve the performance with respect to time parallelization can be applied on the above algorithms

## REFERENCES

1. Kumar A, Chordia N. (2017), “Role of Bioinformatics in Biotechnology”. Res Rev Biosci. 2017;12(1):116. © 2017 Trade Science Inc.
2. S.B. Needleman, C.D.Wunsch, (1970), “A general method applicable to the search for similarities in the amino acid sequence of two proteins”, J. Mol. Biol. 48, 443–453.
3. T.F. Smith, M.S. Waterman, (1981), “Identification of common molecular sequences”, J. Mol.Biol. 147, 195–197.
4. J. Xiong, (2006), “Essential Bioinformatics”, Cambridge University Press, NY, 2006.
5. M.O. Dayhoff, R.M. et.al., (1978), “A model of evolutionary change in proteins”, in: M.O. Dayhoff (Ed.) Atlas of Prot. Seq. and Struct., vol. 5, National Biomedical Research Foundation, Washington, DC 1978, pp. 345–352.
6. S. Henikoff, J.G. Henikoff, (1992), “Amino acid substitution matrices from protein blocks”, Proc. Natl. Acad. Sci. 89 (1992) 10915–10919.
7. Smith, T.F. and Waterman, M.S. (1981), “Identification of common molecular subsequences”, Journal of Molecular Biology, 147, 195-197. doi:10.1016/0022-2836(81)90087-5
8. Needleman, S.B. and Wunsch, C.D., (1970), “A general method applicable to the search for similarities in the amino acid sequence of two proteins”, Journal of Molecular Biology, 48, 443-453. doi:10.1016/0022-2836(70)90057-4
9. Othman, M.B.et.al., (2008), “Genetic algorithms and scalar product for pairwise sequence alignment”, International Journal of Computers, 2, pp. 134-147.
10. Gautam Garai, Biswanath Chowdhury, (2012), “A novel genetic approach for optimized biological sequence alignment”, Journal of Biophysical Chemistry, Vol.3, No.2, 201-205 Journal of Biophysical Chemistry <http://dx.doi.org/10.4236/jbpc.2012.32022>.
11. Taneda, A., (2010), “Multi-objective pairwise RNA sequence alignment”, Oxford Journals, Bioinformatics, 26, 2383-2390, doi:10.1093/bioinformatics/btq439.
12. Notredame, C. et.al., (1997), “RAGA: RNA sequence alignment by genetic algorithm”, Nucleic Acids Research, 25, 4570-4580, doi:10.1093/nar/25.22.4570.
13. Cedric Notredame and Desmond G. Higgins, (1996), “SAGA: sequence alignment by genetic algorithm”, Nucleic Acids Research, Vol. 24, No. 8 1515–1524.
14. R. Storn and K. V. Price, (1995), “Differential evolution: A simple and efficient adaptive scheme for global optimization over continuous

- spaces,” ICSI, USA, Tech. Rep. TR-95-012, <http://icsi.berkeley.edu/~storn/litera.html>.
15. R. Storn and K. V. Price, (1996), “Minimizing the real functions of the ICEC 1996 contest by differential evolution,” in Proc. IEEE Int. Conf. Evol.Comput., pp. 842–844.
  16. R. Storn, (1996) “On the usage of differential evolution for function optimization,” in Proc. North Am. Fuzzy Inform. Process. Soc., pp.519–523.
  17. Swagatam Das, Member, IEEE, and Ponnuthurai Nagaratnam Suganthan, Senior Member, IEEE, (2011), “Differential Evolution: A Survey of the State-of-the-Art, IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 15, NO. 1, FEBRUARY 2011.
  18. K. Price et.al., (2005), “Differential Evolution—A Practical Approach to Global Optimization”, Berlin, Germany: Springer, 2005.
  19. K. V. Price, 1999, “An introduction to differential evolution,” in New Ideas in Optimization, D. Corne, M. Dorigo, and V. Glover, Eds. London, U.K.: McGraw-Hill, pp. 79–108.