Multiple Sequence Alignment based on Enhanced Brainstorm Optimization Algorithm with dynamic population size(EBSODP)

Jeevana Jyothi Pujari^{1*}, Dr.Kanadam Karteeka Pavan²

¹Research Scholar,Nagarjuna University,Andhra Pradesh ²R.V.R &J.C.College of Engineering,Andhra Pradesh,India *jeevanajyothi.pujari@gmail.com

Abstract

Multiple Sequence Alignment is a significantresearch problem in the feild of Bioinformatics.Variousmethodshave been developed for computing optimal sequence alignment, But deriving optimum accuracy is still a challengein multiple alignments. One of the new meta heuristic approach is Brain storm Optimization which can efficiently solve more optimization applications. However premature convergence occurs due to the inability in maintaining the diverging populations and reaching local optima in BSO.In order to address this shortfall in premature convergence, we proposed a new adaptive dynamic population size BSO in our paper. This enhanced mechanism will dynamically increase or decrease the solution set in the search space for every iteration to maintain population diversity. We intend to use Enhanced Brain Storm Optimization Algorithm with dynamic population (EBSODP-MSA) to explore more optimality in alignments of multiple sequences. The experiments derived with the datasets shows that the proposed algorithm performs well in obtaining the nearest and optimal fitness score compared to the original BSO and other evolutionary approaches.

1.Introduction:

Multiple sequence alignment(MSA) is an essential prerequisite in bioinformatics to identify the most similar regions and their evolutionary constraints by aligning two or more nucleotide/amino acid sequences. The resulting sequences can also be used for performing evolutionary based studies like construction of phylogenetic tree of life [1], predicting the structures of newly synthesized sequences and to deduce the homology between the existing and newly derived families of sequences [2], analysis of biological functions, sequence database searching [3]. The extended areas of multiple sequence alignment are structure comparisons in proteins [4], reconstruction of phylogenetic trees [5], and forecasting mutations in proteins [6]. MSA is a NP- complete optimization problem [7] which is computationally complex to obtain global optimal alignments with increase of number sequences. Many techniques have been developed to solve MSA problem from afew decades, even though they remain unsolved [8]. Various algorithms exist to perform MSA and those are categorized into exact, progressive algorithms and iterative algorithms.

One of the exact approaches used to align sequences is Dynamic Programming method [9]. This approach optimally aligns two sequences by maximizing the objective function using trace back method, but the complexity may increase exponentially for aligning more than three sequences [10]. The second technique to align sequences is progressive alignment developed by Feng and Doolittle [11] which aligns closest sequences first and subsequently adds distant ones by iteratively using dynamic programming technique to obtain optimal global alignments and also used to construct phylogenetic trees for depicting the association among the sequences. The algorithms proposed based on progressive approach are, MULTALIGN [13], MULTAL[14], PILEUP[15], Clustal Series(X,W,V)[16]. Clustal Series use the same technique but they are different in user interaction. Progressive based algorithms are very fast but easily get converge in local minima. This situation arises, due to the pairwise sequence misalignments in initial stages and these misalignments cannot be corrected at a later date as well. Hence it leads the progressive based techniques into the situation of local optima [16]. To defeat such type of limitations, it is recommended to use an alternative technique as iterative technique which improve the obtained solutions iteratively and refines it until no improvements can be made on the given objective function. Many techniques which depend on iterative approach used for aligning sequences have been proposed. Those are simulated annealing [17] which aligns sequences simultaneously, HMMT [18], evolutionary computation techniques [19], and swarm Intelligence algorithms [20] are population based techniques.

One of the young swarm intelligence and socially inspired technique is Brain Storm Optimization(BSO) algorithm [21] motivated by the process of human brainstorming sessions for dealing with the issues which are not

solved by individuals, where a horde of individuals meets up to advance the generation of novel ideas by emulating current ideas. In order to generate new ideas, BSO needs more tedious conversations and discussions. Due to this elaborated search space and simplistic nature of implementation, BSO has gained more attention and it is one of the prominent technique among swarm intelligence algorithms, and has made incredible progress in different real world applications within a short period [22]. In BSO, individual ideas are grouped into clustersinitially, the best individual idea can be selected as the center of the cluster. There are three preset probabilities that affect the selection of existing ideas from one group or two groups to generate each new idea.Regardless of its performance and efficacy the BSO still needs a few enhancements. To address the challenges of exploration and exploitationvariant versions of BSO are proposed to optimally solvenumerous scientific and engineering problems such as feature selection [23-24], hybrid Fuzzy BSO for classifying Brain Tumor Images [25], Agglomerative Greedy BSO for Travelling Salesman Problem [26] and so on. The efficacy of BSO appears in the above applications and in numerous areas. However, BSO faces loss of population diversity due to exploitation thereby leading local optima in the early stages itself. Local optima alsodiminish and deteriorate the performance of getting optimal solutions in different applications. Many enhancements research can be done to maintain the diversity of population in BSO [27-29].

In this research paper, we proposed Enhanced BSO by changing the number of ideas dynamically (increasing or decreasing) in every iteration to align sequences optimally in order to solve multiple sequence alignment problem efficiently. To evaluate the efficacy of the proposed method, experiments were performed by changing number of ideas or population dynamically in BSO (EBSODP) and compared with the performance over GA and BSO [30] algorithms.

The rest of the paper is classified as follows: MSA problem Description can be given in section 2. Section 3 presents the introduction to BSO technique.Section 4 presents the proposed Enhanced BSO with Dynamic pop size for solving MSA, problem encoding, and its objective scoring function explained in detail. Section 5 concentrates in details about the experiments performed on the datasets with the proposed and comparative results. Finally, the conclusions with the conducted experiments and the future directions for improvement as mentioned in section 6.

2. Multiple Sequence Alignment:

Multiple Sequence Alignment(MSA) is a tool for predicting and revealing the function and structure of different species of sequence families and is also a crucial step for finding motifs evolved. Regions of similarity in various sequences are identified by Multiple Sequence Alignment. Aligning two sequences is termed as pair wise alignment whereas multiple sequence alignment problem is aligning more than two sequences. Consider a set of *n* input sequences $S=\{s_1,s_2,\ldots,s_n\}$ over an alphabet $\sum U\{-\}$ and the length of sequences are $\{ln_1,ln_2,\ldots,ln_n\}$ respectively. \sum consists of 4 characters in DNA, whereas \sum consists of 20-characters in proteins. Taking the given sequences as input and finding optimally aligned sequences including gaps $S'=\{s_1',s_2',\ldots,s_n'\}$ in order to maximize the similarity score objective function. Gap '- ' in aligned sequences is the representation of indels (insertion or deletion) in biological system. The methods employing an objective scoring function that assigns a score for matches, mismatches and penalizes the gaps with the objective of maximizing matches score and minimizing number of insertedgaps in MSA.

3.Original BSO:

BSO as an algorithm has gained more attention due to its solution accuracy. In BSO model each individual is an idea akin to brainstorming process of human beings to solve optimization problems. All the ideas from individuals of different kinds are gathered as possible solutions to a specific problem. BSO consists of following steps.

A.Individual Generation :

Initially generating N solutions or individualideas randomly within a selected search space. All the randomly solutions are evaluated by a given scoring or objective function.

B.Clustering:

Next step is disrupting the ideas into C number of clusters using the algorithm named k-means. In all the clusters ideas are sorted and most fitted solutions can be recorded as the cluster centers. Generate a value randomly using rand(0,1), if rand(0,1) < Pr_replace then select one of C cluster centers and replace it as a randomly generated idea.

C.Selection of Individuals for mutation:

Selection of individuals for generating new ideas based on Pr_one , if $rand(0,1) < Pr_one$, then select only one clusterotherwise two clusters are selected for mutation. Again selection of cluster center or cluster individuals can be

chosen based on the probabilities of Pr_onecenter and pr_twocenter. if random value is smaller than Pr_onecenter or Pr_twocenter then one cluster center or two cluster centers are selected for mutation, otherwise select one or two individual solutions are chosen for generating new ideas.

D. New Individual Generation:

The new idea can be generated using the equation(1).

 $X_{newidea} = X_{sel} + \xi \cdot N(0, 1)(1)$

 $X_{newidea}$ and X_{sel} are the newly generated and selected ideas respectively. N(0,1) is the guassian distribution with mean 0 and variance 1. and ξ is step length which adds weights to Guassian distribution calculated using the formula (2).

 $\xi = logsig((0.5 * MaxItr - CurrentItr)/k) \cdot Rand(0,1) \quad (2)$

where MaxItrstores the value of maximum iterations and CurrentItr stores current iteration number respectively. A sigmod transfer function logsig() is used. K is the slope adjustment function. Rand(0,1) is the random value lies in the range 0 and 1. The current idea can be updated only when the objective value of the new idea is superior than the old idea's score.

The procedure from (c) to (d) is repeated for all individuals in that iteration. After updating the pool by fittest individuals then go to step (b). The entire process is repeated until *MaxIter* reached.

4. MSA using Enhanced BSO with Dynamic population size: Proposed Algorithm (EBSODP-MSA):

In this section, we describes the application of enhanced BSO to MSA problem. To enhance the search space capabilities population diversity mechanisms has been exploited in nature inspired techniques. In [31-33]proposed a variabe and dynamic population size mechanisms in Differential Evolutionary algorithm. The selection of new population size increases population diversity in turn increasing the searching capabilities in global space. Inspired by the above process, we proposed an enhanced BSO(EBSODP) with dynamically adjusting the population size for obtaining optimal alignments in multiple sequence alignment. The realtive growth rate of number of ideas parameter is controlled by the uniformly distributed random value between [-0.5,0.5] which can diminish or increment the population size in every iteration.

4.1 Individual representation:

Each individual in BSO algorithm is called an idea and considered to be a candidate solution for the given problem. Take the sequences as input and determine the lengths of sequences. $n_i dea_i = \{i=1,2,3,...,N\}$, Initially 'N'number of ideas or population is randomly generated by encoding the gap occurrence positions in each sequence. Determining number of gaps in each sequence requires to calculate the length of largest sequenceplus allow the gaps up to 20% of the maximum length [35]depends on the solutions sets produced in MSA. Depending on the sequence length, the number of gaps and their positions are varied from one sequence to the other.

In order to store and determine the number of gaps in each sequence a vector vg_j is used where $\{1,2,3,...n\}$ for n sequences. For each sequence, the number of gaps varies in between $j=(rand((max(ln_m) - ln_i), (max(ln_m) + [20\%max^{[n]}(lnm-lni])), m=\{1,2,3,...n\}$ and ln_m represents the length of the largest sequence and ln_i is the length of the ith sequence. After that all the gaps are inserted into their positions in the sequences. The total length $Totlen_{align}$ of aligned sequences is same for all the sequences and it lies in the range given in (3). $k=max(ln_m) \leq Totlen_{align} \leq max(ln_m) * 20\%$ ---- (3)

A candidate solution is a matrix of size(n,j), *n* represents number of sequences and *j* represents the maximum column size specified in equation 3. Each solution holds the gap positions for the input sequences in a matrix of n rows and j columns. ith row in the matrix represents the gap occurrence positions of ith sequence with maximum number of columns *j*.insert all gaps into the sequences of alignment matrix S' which consists of aligned sequences in rows as *n* and columns of aligned length ask.

4.2 Scoring Objective Function:

Validating MSA requires use and selection of objective functions as primitive in evolutinary algorithms. Various functions to evaluate MSA are developed. The scoring objective functions are Sum-of-pairs score (SoP), Column

http://annalsofrscb.ro

Score(CS), Weighted Sum-of-Pairs Score (WSP), Coffee, T-Cofee etc. In this work SoP is used to assess the alignment of sequences due to its simplistic nature. Our aim is to maximize the SoP objective function for assessing the alignments in each iteration of produced candidate solutions.

The most popular one used to measure the similarity of MSA is sum-of-pair function and formulated as

$$SOP = \sum_{i=1}^{\kappa-1} \sum_{j=i+1}^{\kappa} Score(S_i, S_j)$$

Subject to

$$Score(S_{i}, S_{j}) = \begin{cases} p, & \text{if } S_{i} = S_{j} \\ q, & \text{if } S_{i} \neq S_{j} \text{ and } S_{i} \neq -' \& S_{j} \neq ' - ' \\ r, & \text{if } S_{i} \neq S_{j} \text{ and } S_{i} = -' \text{ or } S_{j} = -' \end{cases}$$

Where k represents the number of sequences, p is the score assigned to match, q is the score for mismatch and r assigned as gap score. In this technique we assigned the scores as p = +20, q = -1, r = -2. For getting positive score values we choose a highest value for matched sequences.

4.3 Methodology:

The proposed BSO algorithm is enhanced by dynamicallyadjusting the size of individuals or population to avoid the situation of local optima and the search space can be explored efficiently to solve MSA problem optimally. The size of population affects the performance of BSO. The steps involved are described in the algorithm: initially taking all the sequences as input and generating ' n_idea ' number of solutions. each candidate solution is a matrix having *n* number of rows and each row dedicated to one sequence and consists of occurence positions that are randomly generated in the respective sequence and these candidate solutions are evaluated using the sum-of-pairs (SoP) objective function. In MSA the first step is to maximize the score of objective function SoP the second step is to disrupt all the individuals into clusters and sort the ideas based on their SoP score values to make the highest scored idea as centre of the cluster. Third step is, based on the selection of probabilities, create new individuals with new individual. The third step is repeated for all solutions in that iteration. After that step, select a random number $P_rand(0,1)$, which introduces randomness into the algorithm so as to give more optimal results. if $P_rand(0,1)$ is less than the pre specified decision variable pp_r randomly chosen from (0,1) then the size of the population will change that is either incremented or diminished based on the distributed random value rd lies between [-0.5,0.5].

The new population size can be calculated as specified in the equation (4) and the percentage of growth rate of population can increase or decrease according to the choosen value of rd.

 $n_{idea_{new}} = round(n_{idea} + rd*n_{idea}) -----(4)$

If the current population size n_idea is less than the generated population size n_idea_{new} , then all the current individuals will be propated into next generation as well as the remaining $X_{new} = (n_idea_{new} - n_idea_i)$ new individuals are added by taking the current best individual and a random individuals from the current iteration. Otherwise the select the best fitted n_idea_{new} individuals from the current generation to make up the next generation individuals, and the remaining $X_{worst} = (n_idea_i - n_idea_{new})$ worst ranked individuals are eliminated from the next iteration. If $n_idea_{new} = n_idea$ then no more individuals are added and eliminated.

4.4 Proposed Algorithm:

Enhanced Brain Storm Optimization Algorithm with Dynamic Population size for Multiple Sequence Alignment

- 1. Read Input sequences
- 2. n_idea=N, n_cluster=C; maxIter /* initializing the number of solution Ideas, number of clusters & maximum number of iterations*/

Annals of R.S.C.B., ISSN:1583-6258, Vol. 25, Issue 4, 2021, Pages. 10033 - 10042 Received 05 March 2021; Accepted 01 April 2021.

- 3. Initializing N Candidate Ideas as solutions randomly
- 4. **While** (iter<= maxIter)

do

- a. Clustering N solutions into C clusters and keep the best solution as cluster centres
- b. Determine the fitness for each feasible solution by aligning the sequences using SoP scoring function.
- c. $If(rand(0,1) \le Pr_replace)$

Randomly choose one cluster centre and Replace with the randomly selected solution from the cluster.

End if

d. For(i=1 to n_idea)

 $If(rand(0,1) < Pr_one)$

Randomly choose one cluster among C clusters

If(rand(0,1)<Pr_onecenter)

New_idea=Gen_New_Idea(); /*Generate new individual idea by adding information to the selected cluster centre */

Else

Sel_one_Individual; /* Randomly Select one individual idea from the chosen cluster $\!\!\!\!*/$

New_idea=Gen_New_Idea(); /*Generate New Individual by adding information to randomly selected individual of the selected cluster.

End If

Else

Randomly choose two clusters among C clusters

If(rand(0,1)<Pr_twocenter)

New_idea=Gen_New_Idea(); /*Generate New Individual by adding Information to the Selected two cluster centres

Else

Sel_Two_Individuals; /* Select individual ideas from each of two clusters

New_Idea=Gen_New_Idea(); /* Generate New Individual by adding to the two individuals

End If

End If

Fit_New_Idea=Evalfit_SoP(New_Idea) /* Evaluate the Idea using their Fitness Function If(Fit_Old_Idea<Fit_New_Idea)

Replace the Old_idea with New_idea

Else

Keep the Old_idea

End If

End For

e. Dynamically changing n_ideas based on the decision variable

If(P_Rand(0,1)<ppr)

 $n_idea_{new} = round(n_idea + (\pi * n_idea))$

If $n_{\text{new}} \leq n_{\text{idea}}$ then :

carry forward only the best n_idea current individuals into the next generation delete the lowest scored individuals;

Else

carry forward all current individuals into the next generation and add the remaining(n_*idea* new $-n_idea$) individuals by randomly initiating the solutions.



n_idea_{new}=n_idea

```
end If
```

iter=iter+1

End While

5. Experiment results:

This section discusses about the benchmark dataset and the parameter settings and the optimized score for EBSODP for MSA. The Performace of EBSODP algorithm is tested on the sequence alignment benchmark dataset is SABmark [34]. SABmark contains a sequences related to MSA problems and those are divided into two sets of sequences as Superfamily and Twilight sequence sets. We performed tests on some of the randomly choosen Superfamily sets of sequences. The results of EBSODP is compared with Genetic Algorithm(GA) and BrainStorm Optimization(BSO) algorithms for MSA.

Parameter Settings: Throghout the entire expriment the values pertaining to parameter are same. Since these settings yields better score for the sequences tested in the experiment. The values are as follows:Number of ideas or Population(n_idea_i):3000, Number of iterations(max_iter):5, Number of clusters(C):10 are used to find optimal alignment score.

| Table 1:Performance | comparison of SoP | score for EBSODP in | n difeerent iterations as | termination criteria |
|---------------------|-------------------|---------------------|---------------------------|----------------------|
|---------------------|-------------------|---------------------|---------------------------|----------------------|

| Iteration | sup_371 | sup_417 | sup_008 | sup_381 | sup_416 |
|-----------|---------|---------|---------|---------|---------|
| 1 | 72 | 125 | 359 | 590 | 1477 |
| 2 | 105 | 219 | 428 | 701 | 1534 |
| 3 | 117 | 364 | 546 | 746 | 1603 |
| 4 | 258 | 397 | 648 | 746 | 1627 |
| 5 | 258 | 397 | 648 | 746 | 1627 |

Table 1 and Fig 1 shows the performance of the proposed EBSODP at different iterations on the datasets from SABmark. To determine the maximum number of iterations as termination condition for score convergence in EBSODP, we have taken the score for each iteration number as 1,2,3,4,5 in the experiment and depicted in Table 1 and Figure 1. Results from table.1 and Fig.1 shows that, The best score obtained at iteration 4 in majority of the cases, as such the algorithm reached the optimal solution at iteration 4 with population size 3000. The results obtained at Iteration number 4 and 5 are similar in all the cases. So stopping criteria can be taken as maximum number of iterations as 5.

Table 2:Avg SoP score for SABmark database sequences for EBSODP along with GA and BSO approaches

| S.no | No.of Sequences | Seq_Number | GA | BSO | EBSODP |
|------|-----------------|------------|-----|-------|--------|
| 1 | 3 | sup_009 | 104 | 122 | 167 |
| 2 | 3 | sup_017 | 245 | 187.1 | 226 |
| 3 | 3 | sup_043 | 2 | 5 | 11 |
| 4 | 3 | sup_019 | 264 | 254.4 | 291.4 |
| 5 | 3 | sup_030 | 260 | 298.2 | 345.78 |
| 6 | 4 | sup_360 | 274 | 185 | 309 |
| 7 | 4 | sup_364 | 100 | 123 | 165 |
| 8 | 3 | sup_370 | 22 | 59 | 105 |
| 9 | 3 | sup_304 | 343 | 258.2 | 365.9 |

Annals of R.S.C.B., ISSN:1583-6258, Vol. 25, Issue 4, 2021, Pages. 10033 - 10042 Received 05 March 2021; Accepted 01 April 2021.

| 10 | 3 | sup_024 | 1324 | 1331.4 | 1331.4 |
|----|---|---------|------|--------|--------|
| 11 | 8 | sup_008 | 517 | 589.6 | 635.76 |
| 12 | 9 | sup_371 | 287 | 174 | 183 |
| 13 | 4 | sup_393 | 14 | 56.4 | 68.2 |
| 14 | 3 | sup_399 | 159 | 161.5 | 190.5 |
| 15 | 3 | sup_406 | 136 | 205 | 259 |
| 16 | 5 | sup_416 | 993 | 1309.5 | 1600.3 |
| 17 | 3 | sup_417 | 222 | 286.8 | 393.7 |
| 18 | 3 | sup_372 | 479 | 523.6 | 569.2 |
| 19 | 6 | sup_381 | 501 | 625.1 | 725.9 |
| 20 | 4 | sup_401 | 495 | 356 | 477 |

Table 3. Best SoP score for SABmark database sequences for EBSODP along with GA and BSO approaches

| SNO | No_of Sequences | Seq_Number | GA | BSO | BSO_DP |
|-----|-----------------|------------|------|------|--------|
| 1 | 3 | sup_009 | 104 | 122 | 167 |
| 2 | 3 | sup_017 | 245 | 189 | 226 |
| 3 | 3 | sup_019 | 264 | 257 | 292 |
| 4 | 3 | sup_030 | 260 | 314 | 347 |
| 5 | 4 | sup_041 | -241 | -351 | -225 |
| 6 | 3 | sup_043 | -95 | -73 | 11 |
| 7 | 3 | sup_107 | -143 | -21 | -43 |
| 8 | 4 | sup_360 | 274 | 185 | 309 |
| 9 | 4 | sup_364 | 100 | 123 | 165 |
| 10 | 3 | sup_370 | 22 | 59 | 105 |
| 11 | 3 | sup_304 | 343 | 263 | 374 |
| 12 | 5 | sup_379 | -410 | -504 | -409 |
| 13 | 3 | sup_024 | 1324 | 1335 | 1387 |
| 14 | 8 | sup_008 | 517 | 596 | 648 |
| 15 | 9 | sup_371 | 287 | 174 | 183 |
| 16 | 4 | sup_393 | 14 | 56.4 | 69 |
| 17 | 3 | sup_399 | 159 | 169 | 193 |
| 18 | 3 | sup_406 | 136 | 205 | 259 |
| 19 | 5 | sup_416 | 993 | 1323 | 1627 |
| 20 | 3 | sup_417 | 222 | 293 | 397 |
| 21 | 3 | sup_372 | 479 | 535 | 571 |
| 22 | 3 | sup_094 | -103 | -23 | -2 |
| 23 | 7 | sup_034 | -419 | -85 | -24 |
| 24 | 6 | sup_381 | 501 | 625 | 746 |
| 25 | 4 | sup_401 | 495 | 356 | 477 |

Fig 1:Performance comparison of EBSODP approach alignment results with different iterations as termination criteria



Fig 2: Performance comparison of Alignment Results for EBSODP along with GA and BSO Algorithms



The avg SoP Scores of comparative algorithms with randomly chosen sequences from SABmark dataset are tabulated in Table 2 and can be depicted as graphically in Figure 2. It is observed from the Table 2, the proposed EBSODP algorithm obtained more optimal or similar alignment scores in 17 cases compared to other algorithms. For the sequence datasets sup_017,sup_371 and sup_401 GA performs better than BSO and EBSODP. It can be observed from the figure.2 the EBSODP has remarkable performance over compared algorithms in most of the cases. Table 3 shows the Best SoP score values obtained by EBSODP, BSO and GA Algorithms. It can also be observed from the Table 3 and Figure 2 by changing the population size dynamically at each iteration the results shown that the proposed EBSODP algorithm performs far better than other specified approaches. So the local optima and premature convergence can be avoided by maintaining the balance between exploration and exploitation.

6.Conclusion:

Our proposed work presents a novel Brain storm Optimization Algorithm (Multiple Sequence Alignment) enhanced with Dynamic population size(EBSODP) hereby termed as MSA to address the challenging areas in Bioinformatics.MSA plays a major role in predicting the structure and functionality of secondary and tertiary sequences and is also used for phylogenetic Tree construction. The proposed algorithm is enhanced by adapting dynamic population size mechanism to increase the search space and to also maintain a good balance between exploration and exploitation so that it reduces the probability of solutions getting into local optimum. The EBSODP Algorithm is tested against the conventional GA and BSO algorithms with fixed, static population sizes on different protein sequences in benchmark dataset. The results showed that the enhanced BSO with Dynamic Population size yielded better score when compared to conventional GA and BSO algorithms. Moreover based on the best alignment score for the sequences there was a significant improvement in alignment score, that is more optimal solutions can be found in the final iterations. We concluded that use of dynamic population size in BSO enhances the performance getting more optimal alignment score compared to the existing conventional algorithms. In future the comparision between competitive algorithms with respect to the computational complexities could be an interesting area. The future scope of this work also involves improving the algorithm performancefor producing better alignment results while taking multiple objective functions for lengthy sequences.

References:

- [1] Dunn CW, Hejnol A, Matus DQ, Pang K, Browne WE, et al. (2008) Broad phylogenomic sampling improves resolution of the animal tree of life. Nature 452: 745–749.
- [2] Mount, D. W., & Bioinformatics, S. (2001). Genome analysis. Bioinformatics: Sequence and Genome Analysis. Cold Spring Harbor Laboratory Press, Cold Spring Harbor, NY, 479-532.
- [3] Pei, J. (2008). Multiple protein sequence alignment. Current opinion in structural biology, 18(3), 382-386.
- [4] Gelly, J. C., Joseph, A. P., Srinivasan, N., & de Brevern, A. G. (2011). iPBA: a tool for protein structure scomparison using sequence alignment strategies. *Nucleic acids research*, *39*(suppl_2), W18-W23.
- [5] Wang, L. S., Leebens-Mack, J., Wall, P. K., Beckmann, K., DePamphilis, C. W., & Warnow, T. (2009). The impact of multiple protein sequence alignment on phylogenetic estimation. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 8(4), 1108-1119.
- [6] Hicks, S., Wheeler, D. A., Plon, S. E., & Kimmel, M. (2011). Prediction of missense mutation functionality depends on both the algorithm and sequence alignment employed. *Human mutation*, *32*(6), 661-668.
- [7] Wang, L., & Jiang, T. (1994). On the complexity of multiple sequence alignment. *Journal of computational biology*, 1(4), 337-348.
- [8] Morrison, D. (2018). Multiple Sequence Alignment is not a Solved Problem. arXiv: Populations and Evolution.
- [9] Needleman, S. B., & 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*(3), 443-453.
- [10] Notredame, C. (2002). Recent progress in multiple sequence alignment: a survey. *Pharmacogenomics*, 3(1), 131-144.
- [11] Feng, D. F., & Doolittle, R. F. (1987). Progressive sequence alignment as a prerequisitetto correct phylogenetic trees. *Journal of molecular evolution*, 25(4), 351-360.
- [12] Li, K. B. (2003). ClustalW-MPI: ClustalW analysis using distributed and parallel computing. *Bioinformatics*, 19(12), 1585-1586.
- [13] Barton, G. J., & Sternberg, M. J. (1987). A strategy for the rapid multiple alignment of protein sequences: confidence levels from tertiary structure comparisons. *Journal of molecular biology*, *198*(2), 327-337.
- [14] Taylor, W. R. (1988). A flexible method to align large numbers of biological sequences. *Journal of Molecular Evolution*, 28(1-2), 161-169.
- [15] Devereux, J., Haeberli, P., & Smithies, O. (1984). A comprehensive set of sequence analysis programs for the VAX. *Nucleic acids research*, *12*(1Part1), 387-395.
- [16] Chenna, R., Sugawara, H., Koike, T., Lopez, R., Gibson, T. J., Higgins, D. G., & Thompson, J. D. (2003). Multiple sequence alignment with the Clustal series of programs. *Nucleic acids research*, 31(13), 3497-3500.
- [17] Kim, J., Pramanik, S., & Chung, M. J. (1994). Multiple sequence alignment using simulated annealing. *Bioinformatics*, 10(4), 419-426.
- [18] Eddy, S. R. (1995, July). Multiple alignment using hidden Markov models. In Ismb (Vol. 3, pp. 114-120).
- [19] Cai, L., Juedes, D., & Liakhovitch, E. (2000, July). Evolutionary computation techniques for multiple sequence alignment. In *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No. 00TH8512)* (Vol. 2, pp. 829-835). IEEE.

- [21] Shi, Y. (2011, June). Brain storm optimization algorithm. In *International conference in swarm intelligence* (pp. 303-309). Springer, Berlin, Heidelberg
- [22] Cheng, S., Qin, Q., Chen, J., & Shi, Y. (2016). Brain storm optimization algorithm: a review. Artificial Intelligence Review, 46(4), 445-458.
- [23] Zhang, W. Q., Zhang, Y., & Peng, C. (2019). Brain storm optimization for feature selection using new individual clustering and updating mechanism. *Applied Intelligence*, 49(12), 4294-4302.
- [24] Pourpanah, F., Shi, Y., Lim, C. P., Hao, Q., & Tan, C. J. (2019). Feature selection based on brain storm optimization for data classification. *Applied Soft Computing*, 80, 761-775.
- [25] Narmatha, C., Eljack, S. M., Tuka, A. A. R. M., Manimurugan, S., & Mustafa, M. (2020). A hybrid fuzzy brainstorm optimization algorithm for the classification of brain tumor MRI images. *Journal of Ambient Intelligence and Humanized Computing*, 1-9.
- [26] Wu, C., & Fu, X. (2020). An Agglomerative Greedy Brain Storm Optimization Algorithm for Solving the TSP. *IEEE Access*, 8, 201606-201621.
- [27] Cheng, S., Shi, Y., Qin, Q., Zhang, Q., & Bai, R. (2014). Population diversity maintenance in brain storm optimization algorithm. *Journal of Artificial Intelligence and Soft Computing Research*, 4(2), 83-97
- [28] Cheng, S., Shi, Y., Qin, Q., Ting, T. O., & Bai, R. (2014, July). Maintaining population diversity in brain storm optimization algorithm. In 2014 IEEE Congress on Evolutionary Computation (CEC) (pp. 3230-3237). IEEE.
- [29] Yu, Y., Gao, S., Wang, Y., Lei, Z., Cheng, J., & Todo, Y. (2019). A multiple diversity-driven brain storm optimization algorithm with adaptive parameters. *IEEE Access*, 7, 126871-126888.
- [30]Jeevana Jyothi, Pujari and K, Karteeka Pavan and Raiyyan, S M and Rajasekhar, T, MSA: An Application of Brain Storm Optimization Algorithm (April 29, 2020). 2nd International Conference on Communication & Information Processing (ICCIP) 2020,dx.doi.org/10.2139/ssrn.3646174
- [31] Teo, J. (2006). Exploring dynamic self-adaptive populations in differential evolution. *Soft Computing*, 10(8), 673-686.
- [32] Wang, H., Rahnamayan, S., & Wu, Z. (2011, June). Adaptive differential evolution with variable population size for solving high-dimensional problems. In 2011 IEEE Congress of Evolutionary Computation (CEC) (pp. 2626-2632). IEEE.
- [33] Poláková, R., Tvrdík, J., & Bujok, P. (2017). Adaptation of population size according to current population diversity in differential evolution. In 2017 IEEE Symposium Series on Computational Intelligence (SSCI) (pp. 1-8). IEEE.
- [34] Van Walle, I., Lasters, I., & Wyns, L. (2005). SABmark—a benchmark for sequence alignment that covers the entire known fold space. *Bioinformatics*, 21(7), 1267-1268.
- [35] Asraf Yasmin, B., Latha, R., & Manikandan, R. (2019). Implementation of Affective Knowledge for any Geo Location Based on Emotional Intelligence using GPS. International Journal of Innovative Technology and Exploring Engineering, 8(11S), 764–769. <u>https://doi.org/10.35940/ijitee.k1134.09811s19</u>