International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

FDSMO: Frequent DNA Sequence Mining Using FBSB and Optimization

Kuruva Lakshmanna[1*] **Neelu Khare**[2]

[1]*VIT University, Vellore, Tamil Nadu, India*
[2]*VIT University, Vellore, Tamil Nadu, India*
* neelu.khare@vit.ac.in

Abstract: DNA Sequence mining helps in discovering the patterns which can occur frequently, structures of DNA in DNA data sets. Frequent pattern mining is a central strategy for affiliation guideline discovery, but existing calculations experience the ill effects of low effectiveness or poor error rate on the grounds that natural groupings vary from general successions with more attributes. In our last work, we proposed Prefix Span with Group Search Optimization (PSGSO) to optimize the mined results from the Prefix Span method. We propose a new method called Frequent DNA Sequence Mining using Optimization (FDSMO) which combines Frequent Biological Sequence based on Bitmap (FBSB) and Hybrid of Firefly and Group Search Optimization (HFGSO) in this paper. The FDSMO process includes three stages: (i) applying the Frequent Biological Sequence based on Bitmap (ii) calculate length, width and regular expression (iii) optimization using HFGSO. The exploratory results demonstrate that FDSMO performs better than the existing methods, both in terms of running time and scalability.

Keywords: FBSB, DNA sequence, mining, HFGSO, FDSMO, bitmap, Biological sequence

1. Introduction

In this computerized world, where the gigantic measure of information is accessible in advanced structure, a substantial measure of information contains both critical and non-noteworthy patterns. Here the principle test is to discover intriguing examples that are useful to decide, which is an exceptionally dull and time taking task. So there emerges the requirement for a robotized innovation, which does this job proficiently and successfully. Successive example, mining strategy is very valuable for this reason [1]. Different scientists have proposed diverse, continuous pattern mining methods. These methods are considered into biological sequence mining, condition based pattern mining, closed pattern mining, and so on [2]. Sequential pattern mining is a crucial job in extensive applications. Their tasks include examining net access patterns, user obtaining patterns, DNA sequences [3], estimation of diseases etc. [4]. In addition, Sequence pattern mining [5, 6] is one of the many fundamental subjects in data mining and is an additional perspective required in association principle mining [7, 8]. The sequential pattern mining algorithm [9], takes action on solving the problem of determining the frequent sequences in a given database [10]. The sequential pattern mining algorithm [9], deals with the issue of deciding the sequences which occur a number of times in a given database [10]. In association rule mining, the mined result is termed as the items that are purchased together regularly in a single transaction [11].

Concerning DNA, clustering is extensively utilized in genome database. In spite of the fact that few methods were proposed already to cluster genome alignments and DNA microarrays [12], there is exact moment research in the region by utilizing DNA calculations for clustering. A couple arrangements are advanced to utilize DNA calculations to work out clustering issues [13]. In addition to this, very few eras observed the individual and joined tries of data mining and soft computing in the domain of Bioinformatics [14]. In the sequence mining of DNA, Soft computing procedures (including neural networks, fuzzy sets, genetic procedures, soft set and rough sets) etc. which can be most utilized. There are various general classification models, like, Naive Bayesian Network [15], [16], [17], Neural Networks, Decision

Tree, and Rule Learning using evolutionary Algorithm which are put to use here [18].

Recent research works has proved that it is not the best to determine the significance of a pattern with respect to different applications based on its frequency.. The importance of condition pattern mining has improved due to its incompetence [19]. As of late, the condition based sequential pattern mining algorithms [20], have drawn much consideration among scientists. The thought process of condition based sequence mining is to decide the whole arrangement of successive patterns that can fulfil a Regular Expression (RE) conditions.

In this work, we explain a novel approach called Frequent DNA Sequence Mining using Optimization (FDSMO) which combines Frequent Biological Sequence based on Bitmap (FBSB) and a Hybrid of Firefly and Group Search Optimization (HFGSO). Here, at first we apply the FBSB algorithm to the dataset which decides the frequent DNA sequence pattern. Then we apply the length, width, regular expression constrains to the frequent dataset. At last, we receive the HFGSO calculation for culmination of the mining result. The remaining section of this paper is planned as follows: Division 2 gives a brief report on the literature survey. Division 3 explains the proposed DNA sequence mining and division 4 clarified an Outcome and discourse part. Conclusion summed up in division 5.

2. Literature survey

For DNA sequence mining, writing presents a few hypotheses. Presently we survey a portion of the works related to it; Bhawna Mallicket al. [21] clarified the Constraint Based Sequential Pattern Mining. Here, fiscal and compactness constraints were included. Moreover, a CFML-Prefix Span was explained by mixing these constraints with the original Prefix Span algorithm. This allows learning all CFML sequential patterns from the sequence database. The CFML-Prefix Span algorithm was validated on synthetic sequential databases. Dipak R. Kawade and Kavitaet al. [22] explained the Frequent Sequential Pattern Mining with Weighted Regular Expression and Length Limitation. With a specific end goal to accomplish productivity and for powerful execution of the algorithm, the present study makes utilization of regular expression condition which spares time and memory. Many researcher's uses support count technique [SCT], but the main problem they had to encounter related to SCT was finding ideal support value. To resolve this issue, present work uses weight constraint. Essentially, in numerous biological sequences Ling Chen and WeiLiu [23] illuminated frequent pattern mining. Primarily, they elucidated the possibility of essential pattern example, which was extended to frame bigger patterns in the arrangement. In, Xindong Wu et al. [24] elucidated the issue of pattern mining, which is done frequently without client determined gap requirements and also he got PMBC (Pattern Mining from Biological sequences with wild card constraints) to work out the issue. In [36], K. Lakshmanna et al. explained the enhanced method of two heuristic methods (one-way vs. two-way scans) to find out subsequence's which are frequent and to calculate their frequency in the sequences. Limitations of the PMBC and enhanced PMBC is scanning the database many times. Likewise, Jerry et al. [33] have clarified the effective methods for mining up-to-date high-utility patterns (UDHUP). It considers the utility measure as well as timestamp variable to find the latest high utilization patterns (HUPs). In [34], Ke-Chung et al. have clarified the frequent item set mining method in view of the Rule of Inclusion–Exclusion and exchange mapping.

Also, the acknowledgment of promoters in DNA Sequences Utilizing Weightily Averaged One-dependence Estimators are clarified by ZawHtike and Shoon Lei Win [25]. To reduce the time complexity and to raise the efficiency of the system, an entropy-based feature extraction approach used to choose related nucleotides that are liable for promoter recognition. In [26], George Aloysius and D. Binu have clarified a methodology for items arrangement in grocery stores utilizing Prefix Span algorithm. Masseglia et al. [31] have clarified the productive mining of sequential patterns with time requirements. They presented a method GTC (Graph for Time Constraints) for mining patterns in huge databases. It depended on the idea that taking care of time limitations in the prior phase of the mining procedure was highly useful. In [35], K Lakshmanna and N. Khare proposed a method which called 3s-DNASM, incorporating prefix span, length and width constraints and group search optimization. The drawback of this paper is achieving the less patterns. In addition, Atsuyoshi et al. [32] have clarified the Mining inexact patterns with continuous locally optimal existences. Here, candidate patterns were produced using the postfix tree of a given string without repetition. Further, they define a k-gap constrained setting, in which the no. of gaps in the alignment between a pattern and an occurrence is limited to at most k.

In [37], Q. Wang et al. FBSB method utilizes bitmaps to record the arrangement position in every

activity, and a quick sort list QS-list is made for speedy arrangement connection. They show a pressure trademark what's more, proficient computation amid the mining procedure, and FBSB won't extensively join frequent arrangements to produce hopeful sequences. Every one of the successions tried truly do happen in light of the fact that FBSB method utilizes position value to direct the connection and it guarantees that the subsequence's of the trade arrangements are frequent. So less computing time and memory space are required. In the wake of examining the exploratory results, it is demonstrated that FBSB computation is more viable with a decent versatility.

3. Proposed DNA Sequence Mining

The essential thought of our proposed procedure is to mining DNA sequential pattern with imperatives utilizing hybridization of firefly and group search optimization algorithm. Generally, the DNA database having a greater number of items. The big sequence makes an extraordinary debate for introducing sequential pattern mining algorithms. As indicated by this, we are mining the frequent patterns. Essentially, this paper comprises of three modules, like, (i) FBSB module (ii) constraint module and (iii) hybrid optimization module. Every module the repeated DNA sequences are mined. The overall diagram of the proposed DNA sequence mining is shown in figure 1.

Module 1: Mining based on FBSB algorithm

To plan a successful mine frequent biological sequences and information structure precisely, the FBSB method [37], can be carried out in light of the accompanying perceptions. Initially, the supports of all items are calculated by scanning the databank once, all of the location values of the 2-sequences are placed in arrays and the initial bitmap is formed. Repetitive items are extracted directly. On 2nd step,

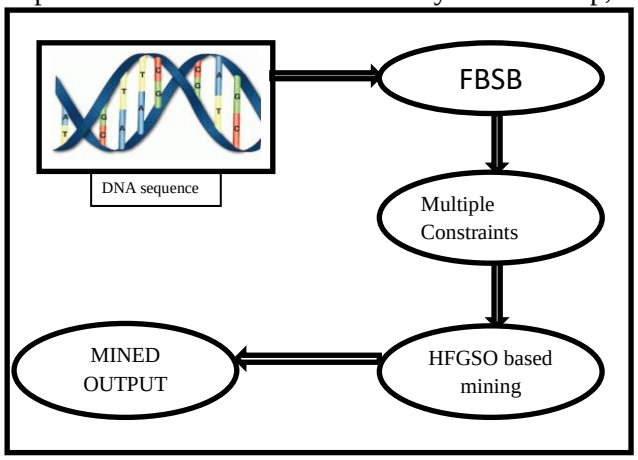

Figure.1 Overall Architecture of FDSMO method

frequent k length sequences are mined and (k+1) - sequences are produced by joining frequent k length sequences in the same activity with allocation value arising order. The two k length sequences can be associated just when the position estimation of the second sequence is 1 more than the first, then frequent (k+1) length sequences are gotten and bitmap is redesigned. In the 3rd step, candidate sequences are not produced, every one of the sequences truly do happen and 100% precision.

Algorithm 1: Bitmap initiation and frequent item generation

Input: A sequence database D, threshold T

Output: Bitmap Pos and frequent item-set L1

Step 1: Min_sup = T *|D|

Step 2: Pos = null

Step 3: Evaluate the supports of the items, regardless of the fact that an item happens many times in one exchange, its backing can be included at most 1 for one exchange.

Step 4: Frames the bitmap for all the 2-sequences, the position estimation of the last item of a sequence is utilized as the position estimation of the sequence.

Step 5: Mine frequent items whose backings are not less than the Min-sup and push the frequent items into the frequent item-set L1.

Step 6: Produce the frequent1-itemset L1and bitmap Pos.

Algorithm2. Frequent k length sequence mining and (k+1) length sequence generation

Input: Bitmap Pos for k length sequences, minimal support min-sup

Output: Bitmap NewPos for (k+1) length sequences

Step 1: Mine the frequent k length sequences whose support not less than Min-sup.

Step 2: Frame the QS-list for the ith row in Pos. If the QS-list is valueless or its length |QS-list| is one, remove the ith row of the Pos and frame the QS-list for the following row in Pos.

Step 3: Join frequent k length sequences to create (k+1) length sequences, double sequences can be joined only when the location value of the 2nd sequence is one more than the location value of the 1st sequence

The newly joined (k+1) length sequence can be produced by adding the most recent item of the 2nd sequence to the 1stsequence.

Step 4: Yields the updated bitmap for(k+1) length sequences &the frequent k length sequence set Lk.

❖ An example

The way toward mining frequent bio-groupings from database D in Table1 is as per the following. Let T=50%, then Min-sup=T*|D|=50% *4=2. The primary bitmap for 2-sequences are formed in Table2. All the backings of the items can be evaluated. sup(A)=4, sup(C)= 4, sup(G)=4. Items A,C,G are all frequent items. Frequent2-sequences are mined, as sup(AC)=4, sup(CG)=4, sup(GC)=4, sup(CA)=3, sup(AG)=3 and sup(GA)=1. The sequence GA and Pos(GA) are deleted because of Sup(GA) is less than Min-sup. Frequent 2 length sequences are AC, CG, GC, CA, AG.

The QS-list of the repetitive 2 length sequences in the next transaction (GA has been removed). 3-length sequences are easy to be attained. Add the last item of GC to AG, and add the last item of CG to GC. Then Pos(AGC)=3, Pos(GCG)=4, AC cannot be linked because of the position value of AC is not next to CG, and link stops for there are no more sequences after AC. New bitmap for 3-sequences are exposed in Table3. sup(ACG)=3, sup(CGC)=2, sup(GCA)=3,sup(CAG)=1,sup(AGC)=2,sup(GCG) =1,sup(CAC)=2. Frequent 3-sequences are ACG, CGC, GCA, AGC, CAC. Next bitmap for 4-sequences are displayed in Table4. The concluding results for all the frequent sequences are in Table5.

Table 1. Sequential database D

ID	S_i
1	<ACGCAG>
2	<AGCGAC>
3	<CGCACG>
4	<AGCACG>

Table 2. Bitmap of database D

S ID	AC	CG	GC	CA	AG	GA
1	2	3	4	5	6	ɵ
2	6	4	3	ɵ	2	5
3	5	2,6	3	4	ɵ	ɵ
4	5	6	3	4	2	ɵ

Table 3. Bitmap for 3-sequences

S ID	ACG	CGC	GCA	CAG	AGC	GCG	CAC
1	3	4	5	6	ɵ	ɵ	ɵ
2	ɵ	ɵ	ɵ	ɵ	3	4	ɵ
3	6	ɵ	4	ɵ	ɵ	ɵ	5
4	6	ɵ	4	ɵ	3	ɵ	5

Table 4. Bitmap for 4-sequences

S ID	ACGC	CGCA	GCAG	AGCG	GCAC	CACG	AGCA
1	4	5	6	ɵ	ɵ	ɵ	ɵ
2	ɵ	ɵ	ɵ	4	ɵ	ɵ	ɵ
3	ɵ	4	ɵ	ɵ	5	6	ɵ
4	ɵ	ɵ	ɵ	ɵ	5	6	4

Table 5: Final results for all the frequent sequences

Frequent sequence set	Patterns
L1	A, C, G
L2	AC, CG, GC, CA, AG
L3	ACG, CGC, GCA, AGC, CAC
L4	CGCA, GCAC, CACG
L5	GCACG

Module 2: DNA Sequential Pattern mining based on constraints

The mined DNA sequence got from FBSB is given in this segment. Here, we utilize three sorts of constraints, for example, length, weight and regular expression constraints (RE) to find the continuous patterns. In this method, takes input as the weight of each item obtained from the FBSB algorithm, regular expression (RE) and Min_length and Max_length. This method test database once and find frequent sequential patterns which fulfill given Min_weight, Min_length and Max_length. In this method, we have taken the regular expression is < A*C*G*T* >. Moreover the sequences [ACGT], [AGT], [CG] are the valid sequence. Here, we additionally discover the aggregate number of patterns which fulfil the regular expression. After that we check these achieved sequences are fulfilling the length and weight obliges. At long last, we check what are the sequences fulfil all the three limitations, which are taken as the mined patterns, the remaining patterns are wiped out. This method works in the following way which is given in Algorithm 3.

Algorithm 3: steps involved in DNA Sequential Pattern mining based on constraints

Step 1: This algorithm inspects the database (output of the FBSB algorithm) only once and find patterns which are satisfy the regular expression constraints (REC).

Step 2: Once the REC is completed, we delete the repeated patterns and store the count of patterns

Step 3: After that we calculate the average weight of the sequence

Step 4: We check which are the sequences satisfy min_length and max_length and delete remaining sequences which does not satisfy the length constraint.

Step 5: After that we select those sequences that satisfy the weight constraint.

Step 6: Finally display the frequent sequential patterns which satisfy given regular expression, given weight and given length constraints.

Module 3: Optimizing mining via HFGSO algorithm

The completeness of the mining process is organized through hybridization of group search and firefly algorithm (HFGSO) after length, width and RE Constraints. The optimization is used to decrease repetition of the sequences and the redundancy from the DNA database. First, we apply the GSO algorithm [27] to DNA sequence pattern to mine the efficient patterns. GSO algorithms have the three operators such as producer, scrounger and ranger. To increase the efficiency of the method, we have hybrid the Firefly algorithm [29] with GSO algorithm. By hybridized these two classifiers, we expect that mining execution will be expanded keeping in mind, which will be enhanced the precision of mined pattern. The result section shows that the proposed optimization algorithm of HFGSO attained healthy performance than the separate optimization algorithm.

Step 1: Initialize the search solution as well as the head angle:

The important stage of the optimization algorithm is Solution encoding. Here, we produce the solution for hybrid GSO and FA algorithm. Let the first population be as follows,

$$P^S = [\, P_1, P_2, \ldots . , P_n \,] \quad (1)$$

Where set P^S denotes the population of the mined sequence. P_1 to P_n are the individuals in the population. The head angle (Ψ) can be expressed as shown in Equation (2).

$$\Psi_i^s = \left(\Psi_{i1}^s \ldots\ldots\ldots \Psi_{i(n-1)}^s\right) \quad (2)$$

The direction of the search of the member $L(\Psi)$ is dependent on head angle, as in equation (3).

$$L_i^s(\Psi_i^s) = \left(l_{i1}^s \ldots\ldots\ldots l_{i(n)}^s\right) \quad (3)$$

The Polar and Cartesian coordinate transformation is powerfully structured to estimate the direction of search based on the head angle.

$$L_{i1}^s = \prod_{p=1}^{n-1} \cos(\Psi_{ip}^s) \quad (4)$$

$$L_{ij}^s = \sin(\Psi_{i(j-1)}^s \prod_{p=j}^{n-1} \cos(\Psi_{ip}^s) \quad (5)$$

Where j = 2 to n-1

$$L_{in}^s = \sin(\Psi_{i(n-1)}^s) \quad (6)$$

Step 2: Fitness calculation:

On the basis of the support, confidence, lift and frequency parameters of the suggested approach, the fitness function is calculated. The support of the sequence is defined as the significance of a particular sequence in the DNA datasets and the support is the ratio of the presence of a particular sequence in the transactions to the total number of transactions in the DNA data sets. The minimum support is indicated as min_support and is designated as,

$$\min\_ \sup port = \frac{T(X,Y)}{T_n} \quad (7)$$

Here, $T(X,Y)$ is the number of transactions, which enfold the sequences and T_n is the total number of transactions. The parameters, confidence and lift are obtained from the parameter support. The parameter can be obtained as,

$$confidence = \frac{Support(X \cup Y)}{Support(X)} \quad (8)$$

$$lift = \frac{Support(X \cup Y)}{Support(X) \times Support(Y)} \quad (9)$$

So, based on the lift and confidence parameters, expand a fitness function for the suggested method for optimizing the sequences.

$$fitness = conf(s) + \log(Support(s) \times (\min\_ Suppot(i_{left}) + \min\_ Support(i_{right})) * lift(s)) \quad (10)$$

Now, i_{left} and i_{right} are the item sets in left side and right side of a sequence correspondingly. After evaluating all the fitness values, the fitness values are supplied with a fitness set, which enfolds the fitness of the sequences.

$$F^C = [\, f_1^c, f_2^c, \ldots . f_n^c] \quad (11)$$

Step 3: Find the producer(Z_p)of the group:

The member with the top fitness of Zi is known as the producer and indicated as *Zp*.

- Producer performance

In the functioning of the GSO technique, the action of the producer *Zp* at the s^{th} iteration explained as,

(i) At zero degree, it brings the scanning assignment

$$Z_z = Z_p^s + \varepsilon_1 d_{\max} L_p^s(\Psi^s) \quad (12)$$

d_{max} denotes the maximum search distance.

(ii) Accomplishes the scanning function at the right hand side hypercube

$$Z_r = Z_p^s + \varepsilon_1 d_{\max} L_p^s\left(\Psi^s + \varepsilon_2 \Phi_{\max}/2\right) \quad (13)$$

(iii) It executes the scanning task at the left hand side hypercube

$$Z_l = Z_p^s + \varepsilon_1 d_{\max} L_p^s\left(\Psi^s - \varepsilon_2 \Phi_{\max}/2\right) \quad (14)$$

Here, $\varepsilon 1$ - normally distributed random number with zero mean and unity standard deviation

$\varepsilon 2$ - stands for a uniformly distributed random sequence which has values within the range 0 and 1.

The maximum search angle Φ_{max} is effectively represented as:

$$\Phi_{\max} = \frac{\pi}{c^2} \tag{15}$$

The constant c can be furnished as:

$$c = round(\sqrt{n+1}) \tag{16}$$

Where, n corresponds to the dimension of the search space.

$$\therefore \Phi_{\max} = \frac{\pi}{n+1} \tag{17}$$

The evaluation of maximum search distance d_{max} includes the ensuing equations.

$$d_{\max} = \left\| d_U - d_L \right\|$$
$$d_{\max} = \sqrt{\sum_{i=1}^{n} (d_{Ui} - d_{Li})^2} \tag{18}$$

Here, d_{Li} and d_{Ui} represent the lower and upper limits of the ith dimension, correspondingly.

The best location consisting of the most beneficial resource may be achieved by means of Equations (8), (9) and (10). The current location gives the way for a new better location, if its existing resource is found to be poorer for that in the new location. Otherwise, the producer preserves its location and turns its head as per the head angle direction which is randomly formed by means of Equation (19).

$$\Psi^{s+1} = \Psi^{s} + \varepsilon_2 \tau_{\max} \tag{19}$$

Here, τ_{max} corresponds to the maximum turning angle which is evaluated with the help of the equation given below.

$$\tau_{\max} = \frac{\Phi_{\max}}{2} \tag{20}$$

When the producer is incapable to recognize a healthier position even after the conclusion of m iterations, its head would assume its initial position as given in equation (21).

$$\Psi^{s+c} = \Psi^{s} \tag{21}$$

Step 4: Scrounger performance

The scrounging action of the GSO usually comprises the area copying task. During the s^{th} iteration, the function of area copying which the i^{th} scrounger carries out may be shaped as a movement to inch near the producer in an friendly manner which is illustrated as:

$$Z^{s+1} = Z_i^{\ s} + \varepsilon_3 O(Z_p^{s} - Z_i^{s}) \tag{22}$$

Where, O specifies the Hadamard product which determines the product of the two vectors in an entry-wise manner and $\varepsilon 3$ denotes a uniform random sequence lying in the interval of (0, 1). The i_{th} scrounger remains to be in its searching task so as to make a selection of the superior chance for the purpose of linking.

Step 5: Solution updating via Firefly operator

The firefly algorithm works based on the brightness of the birds. The firefly updating is based on equation (23).

$$Z_{i+1} = Z_i + B_0\, e^{-\gamma r^2} \left(Z_j - Z_i\right) + \alpha \left(rand - \frac{1}{2} \right) \tag{23}$$

Here, B_0→ The Degree of attractiveness of the firefly at distance $r = 0$

r → The Distance between any two fireflies

γ → The Coefficient of light absorption

After total iteration, the fitness between the old and novel sequence are compared and the one with higher fitness is maintained. If the new sequence has better fitness, it will be replaced with the old sequence. Otherwise, it will be subjected for development in the next iteration of HFGSO. The latest step of the HFGSO algorithm is optimizing the sequences based on the fitness threshold. A set for optimized sequence is created for packing the optimized sequences from the squeezed sequences based on the fitness, defined by S_{op}. Reflect on the set of sequences be S, and S_d be the rejected sequences.

$$s_i \in S = \begin{cases} r_i \in S_{op}, if\ \ fitness > threshold \\ r_i \in S_d, if\ \ fitness < threshold \end{cases} \tag{24}$$

Here, the set of sequences S_i in S is passed to either to the set of optimized sequences and either to the set of discarded sequences. The procedure for DNA sequence mining scheme using the HFGSO algorithm as follows in Algorithm 4.

Step 6: Termination criteria

This algorithm suspends its execution only if the extreme number of iterations are attained and the solution which is holding the best fitness value is selected. Once the best fitness is accomplished by means of HFGSO algorithm, the selected sequences are mined DNA sequences.

Algorithm 4: Optimal mining via HFGSO algorithm

Input: Parameter of the GSO and the Firefly algorithm, DNA sequence

Output: Mined sequence

Step 1: call equation (1) to create an initial solution

Step 2: perform evaluating the fitness function

Step 3: copy the solution directly to the next new population

Step 4: repeat

Step 5: call step 2 to choose the producer

Step 6: call step 3 to the scrounger operation

Step 7: call equation (23) to solution updating based on the firefly algorithm
Step 8: call equation (24) to check the test condition
Step 9: if condition is satisfied, stop and return the best solution in the current population. Otherwise, repeat step 2 to 8 until the target is met.

4. Result and discussion

The experimental results of the proposed approach for DNA sequence mining are explained. We evaluate the performance and efficiency of our proposed approach. We compare it with the traditional algorithms FBSB algorithm. In this approach, we use two set of DNA sequence dataset such as AF008212.1 (dataset 1) and AF348520.1 (dataset 2) [30]. The DNA sequence presents the deoxy nucleotides (A, G, C and T).

➢ Experimental result analysis

The main concept of our research is to Mining DNA sequence patterns with constraints using hybridization of firefly and Group search optimization (HFGSO). The FBSB algorithm to the DNA sequence to mine the pattern is applied at first. After that, some of the sequence is mined using the Frequent Sequential Pattern mining with Weighted Regular Expression and Length Constraint Algorithm. Finally, we obtain the optimization algorithm to mine the sequential pattern. In Table 6, the sample data sequences are shown. The parameter used in this HFGSO algorithm is shown in table 7. We choose 3000 sequences from the DNA sequence dataset, and divide the selected sequences into five groups so as to test the scalability of the algorithms. In each group, we use protein sequences with similar length to form 1 or 2 test data sets. In different groups, data sets have different numbers of sequences from 100 to 500.Table 8, shows the number of data sets, the number of sequences in each data set.

Table 6. Sample DNA sequence database

ID	Sequence
10	ACTATTGTAGAGTA
20	AGTATTAATCGAT
30	ACTAGTCGATCG
40	CTAGTGCGATCTATGCTTAA
50	GAGTGCTTAATCG

Table 7. Parameters of HFGSO algorithm

Step size factor α	Population size N	Absorption coefficient γ	Initial head angle	Producer	Scrounger	Ranger	ϕ_{max}
0-1	100	1	45	1	16	3	$\pi/6$

Table 8. Groups of sequences tested

Group id	Total number of sequences	Number of datasets	Number of sequences in each dataset
1	200	2	100
2	400	2	200
3	600	2	300
4	800	2	400
5	1000	2	500

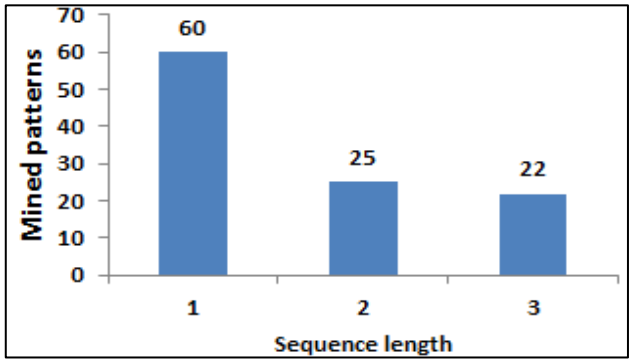

Figure.3 Patterns mined based on length in dataset 1

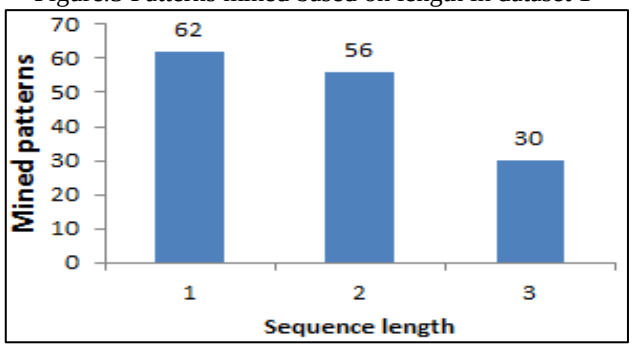

Figure.4 Patterns mined based on width in dataset 1

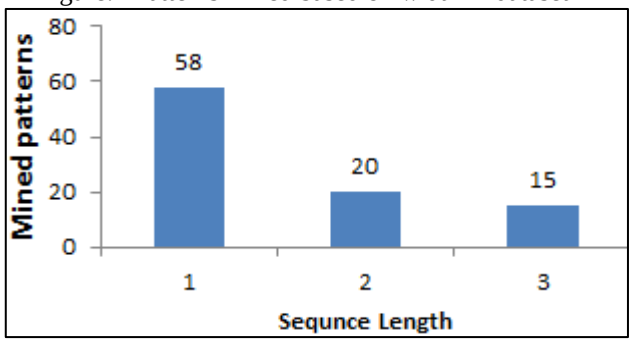

Figure.5 Patterns mined based on length in dataset 2

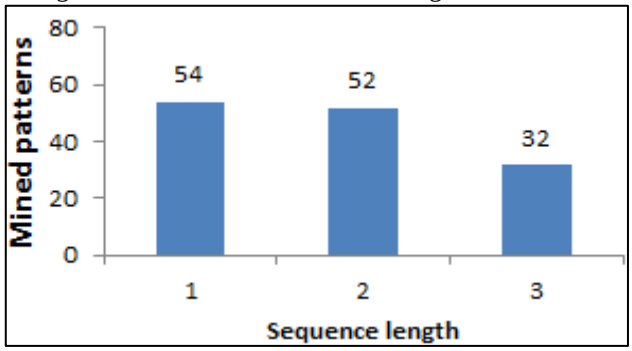

Figure.6 Patterns mined based on width in dataset 2

The above figures show the experimental results obtained from the proposed approach with the 2 types of DNA sequence data sets. To mine the sequence, the input dataset is given to the proposed approach of 3-module FDSMO algorithm. Figure 3 shows the total number of mined patterns by varying the length of the sequence. If the length of the sequence is 3 we obtain the mined pattern of 22. Likewise, figure 4 shows the performance of proposed method for dataset 1 by varying the width of the sequence. Regular expressions are used by algorithm for discovering user interested patterns. Length constraint is used to restrict the length of the pattern so as to reduce the search space and find a user interested pattern efficiently and effectively. Weights are used to discover pattern according to importance of the items. Similarly, figure 5 and 6 shows experimental result based on the dataset 2.

- **Comparative analysis of proposed approach**

In this section the comparative analysis of the proposed approach to prefix span (PS), PS+GSO, PS+FA and PS+HFGSO is described. The proposed approach provides an optimal order of sequential patterns when compared to existing algorithm is ensured by the comparison result. In this proposed approach, we use length is 3 and the width is 2. Hybrid optimization algorithm which increases the effectiveness of the approach is used in this DNA sequence mining method. The hybrid firefly and group search optimization for the mined pattern are also used. Two different datasets are used for comparison to prove the effectiveness of this approach. The second dataset is synthetically generated with view of installation which will be done in respect to the first patterns. Comparative analysis of the proposed approach for dataset 1 & 2 showed in Table 9 & 10 respectively. When the minimum support is 2 we obtain the mined pattern of 55 for proposed approach, 62 for PS+HFGSO, 13218 for prefix span approach, 5298 for PS+GSO approach and 7954 for PS+FA which are associated to the dataset 1. Similarly, we choose the minimum support at 4 means our proposed approach achieves the mined pattern of 60. In Table 10, our proposed approach achieves the minimum mined pattern of 54. Three types of constraints which is increasing the effectiveness of the system are used in this. The RE constraints used to select the regular expression of the sequence other patterns are eliminated. Moreover, figure 7 and 8 shows the comparative analysis based on computation time. We compare our proposed work with PS, PS+GSO, PS+HFGSO and PS+FA. When analyzing figure 9, our proposed work takes minimum time when compared to other approaches. Similarly, the comparative analysis based on computation time is illustrated by Figure 10. When analyzing figure 10, when compared to

Table 9. Comparative analysis of the proposed approach for dataset 1

Min_ support	Mined patterns				
	FDSMO	PS+HFGSO	PS	PS+GSO	PS+FA
2	55	62	13218	5298	7954
3	54	67	6342	4521	6743
4	60	69	3968	2767	5447

Table 10. Comparative analysis of the proposed approach for dataset 2

Min_ support	Mined patterns				
	FDSMO	PS+HFGSO	PS	PS+GSO	PS+FA
2	54	56	8072	6543	7932
3	48	55	3636	5643	6893
4	44	53	2168	4562	5342

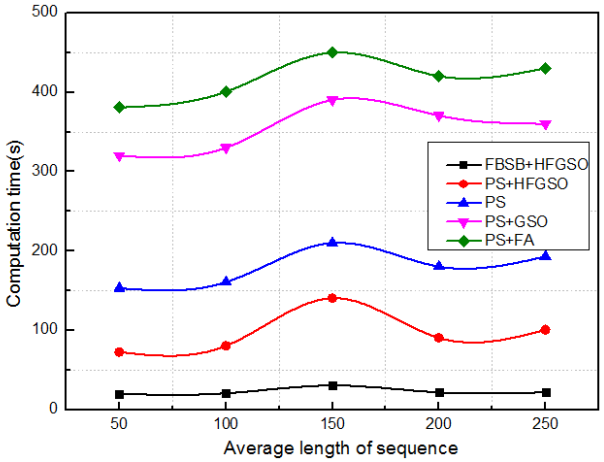

Figure.7 Comparison of computation times of five algorithms on data sets 1 with different length of sequences (min_support 4)

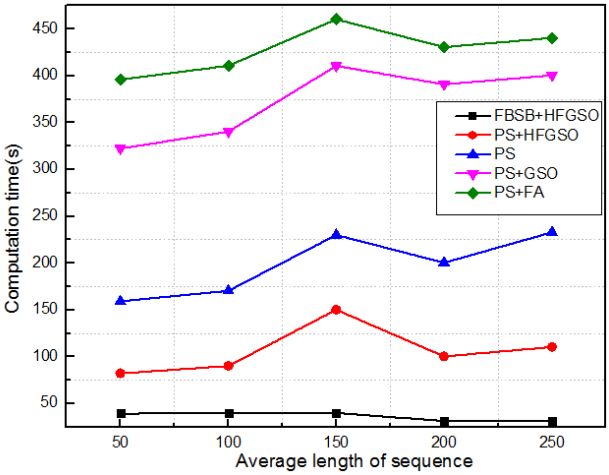

Figure.8 Comparison of computation times of five algorithms on data sets 2 with different length of sequences (min_support 4)

other techniques PS based DNA sequence approach takes maximum time. From all of the above tables we clearly understand that better performance is achieved by our proposed method when compared to the existing approaches.

5. Conclusion

In this article, we explained an algorithm that permits to accomplish efficiently the constant-based DNA sequence mining task. The detail DNA sequence mining procedure comprises mainly three modules: 1) mining based on FBSB algorithm 2) constrained based mining, 3) HFGSO optimization. The concept of FBSB is applied on DNA data sets to mine the frequent DNA sequence pattern in the first module. Next, we used three different types of constraints such as weight, length and Regular expression. The suggested RE constraints in mining the pattern method to generate repetitive pattern of customer interest. Also, we utilized the weight of every item which is an important constraint of each and every item. Equally, our proposed algorithm uses length constraints which restrict the length of the pattern. FDSMO algorithm produces a repetitive pattern which satisfies min weight, length and regular expression constraints. Finally, the optimized mining result was obtained through HFGSO algorithm. The experimental results proved that FDSMO achieved higher quality results when compared to the existing methods. In future I will plan to consider candidate generation in the intial step. Also we may try with the suffix tree concept in future.

References

[1] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules", Proc. of the 20th Int'l Conference on Very Large Databases, Santiago, Chile, Sept. 1994., Expanded version available as IBM Research Report RJ9839, June 1994. |

[2] Yun and Unil. "WIS: Weighted Interesting Sequential Pattern Mining with a Similar Level of Support and/or Weight", ETRI Journal, vol. 29, no. 3, 2007.

[3] Yun, Unil and K. Ryu, "Discovering Important Sequential Patterns with Length-Decreasing Weighted Support Constraints", International Journal of Information Technology & Decision Making, vol. 9, no. 4, pp.575-599, 2010.

[4] D. R. Kawade and S. Kavita Oza, "Exploration of DNA Sequences Using Pattern Mining", International Journal of Emerging Technologies in Computational and Applied Sciences, vol.2, no.6, pp. 144-148, 2013

[5] Julisch, "Data Mining for Intrusion Detection - A Critical Review, Application of Data Mining in Computer Security", Kluwer Academic Publisher, Boston, 2002.

[6] W.Frawley, G. Shapiro, and Matheus, "Knowledge Discovery in Databases: An Overview," AI Magazine, vol. 13, no. 3, pp. 213- 228, 1992.

[7] S. Hou and X. Zhang , "Alarms Association Rules Based on Sequential Pattern Mining Algorithm", in Proceedings of the 5th International Conference on Fuzzy Systems and Knowledge Discovery, Shandong, vol. 2, pp. 556-560, 2008.

[8] F. Masseglia , P. Poncelet, and M. Teisseire , "Incremental Mining of Sequential Patterns in Large Databases," Data & Knowledge Engineering, vol. 46, no.1, pp. 97-121, 2003.

[9] R. Agrawal and R. Srikant, "Mining Sequential Patterns," in Proceedings of the 11th International Conference on Data Engineering, Taiwan, pp. 3-14, 1995.

[10] J. Parmar and S. Garg , "Modified Web Access Pattern (mWAP) Approach for Sequential Pattern Mining," Journal of Computer Science, vol. 6, no. 2, pp. 46-54, 2007.

[11] E. Yafi, A. Al-Hegami., A. Afsar, and Ranjit B., "YAMI: Incremental Mining of Interesting Association Patterns", International Arab Journal of Information Technology, vol. 9, no. 6, pp. 504-510, 2012.

[12] Lu, X., Lin, Y., Li, X., Yi, Y., Cai, l., Wang, H., "Gene cluster algorithm based on most similarity tree", In: Proceedings of the Eighth International Conference on High-performance Computing in Asia-Pacific Region,2005.

[13] Bakar, R.B.A., J. Watada, W. Pedrycz, "A DNA computing approach to data clustering based on mutual distance order", In: Proceedings 9th Czech–Japan Seminar , pp. 139–145, 2006.

[14] S. Mitra and Acharya, "Data Mining: Multimedia, Soft Computing, and Bioinformatics", John Wiley & Sons, 2003.

[15] C. Eugene, "Bayesian Network without Tears," AI Magazine, vol. 12, no. 4, pp. 50-63, 1991.

[16] D.M. Chickering, D. Heckerman, and D. Geiger, "Learning Bayesian Networks: The Combination of Knowledge and Statistical Data", Machine Learning, vol. 20, pp. 197-243, 1995.

[17] W. Liu, J. Cheng, and A.B. David, "An Algorithm for Bayesian Belief Network

Construction from Data", Proc. Sixth Int'l Workshop Artificial Intelligence and Statistics, 1997

[18] W. Banzaf, P. Nordin, R. Keller, and F. Francone, "Genetic Programming—An Introduction", Morgan Kaufmann, 1997.

[19] Hu Y., "The Research of Customer Purchase Behavior using Constraint-Based Sequential Pattern Mining Approach," Thesis Report, National Central University Library Electronic Theses and Dissertations System, 2007.

[20] S. Orlando, R. Perego, and C. Silvestri, "A New Algorithm for Gap Constrained Sequence Mining," in Proceedings of the ACM Symposium on Applied Computing, Cyprus, pp. 540-547, 2004.

[21] B. Mallick, D. Garg and P. Singh Grover, "Constraint-Based Sequential Pattern Mining: A Pattern Growth Algorithm Incorporating Compactness, Length and Monetary", the International Arab Journal of Information Technology, vol. 11, no. 1, 2014

[22] D.R. Kawade and S. Kavita Oza, "Frequent Sequential Pattern Mining With Weighted Regular Expression and Length Constraint", international journal of scientific research, volume : 4 | Issue : 5 | May 2015

[23] ZawZawHtike and S. Lei Win, "Recognition of Promoters in DNA Sequences Using Weightily Averaged One-dependence Estimators", Procedia Computer Science, Vol. 23, pp. 60-67, 2013.

[24] L. Chen and WeiLiu, "Frequent patterns mining in multiple biological sequences", computers in biology and medicine, vol.43, pp.1444-1452, 2013

[25] X. Wu, X. Zhu, Yu He and A.N. Arslan ,"PMBC: Pattern mining from biological sequences with wildcard constraints", computers in biology and medicine, vol.43, pp. 481-492, 2013

[26] G. Aloysius and D. Binu, "An approach to products placement in supermarkets using Prefix Span algorithm", Journal of King Saud University - Computer and Information Sciences, Vol. 25, no. 1, pp.77–87, 2013

[27] S. He, Q. H. Wu and J. R. Saunders, "A Group Search Optimizer for Neural Network Training" Lecture Notes in Computer Science, vol.3982,pp.934-943,2006.

[28] I. D. Couzin, J. Krause, N. R. Franks, and S. A. Levin, "Effective leadership and decision-making in animal groups on the move," Nature, vol. 434, pp. 513–516, 2005.

[29] X. Yang and X. He, "Firefly Algorithm: Recent Advances and Applications", nt. J. of Swarm Intelligence, 2013 Vol.1, No.1, pp.36 - 50

[30] J. Pei, Jiawei Han, BehzadMortazavi-Asl and H. Pinto, "PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected PatternGrowth", 2002.

[31] F. Massegliaa, Poncelet and M. Teisseire, "Efficient mining of sequential patterns with time constraints: Reducing the combinations", Expert Systems with Applications, vol. 36, no. 2, pp. 2677–2690, 2009.

[32] A. Nakamura, I. Takigawa, H.Tosaka, M. Kudo and H. Mamitsuka, "Mining approximate patterns with frequent locally optimal occurrences", Journal of Discrete Applied Mathematics, vol. 200, pp. 123–152, 2016

[33] J. Chun-Wei Lin, WenshengGan, Tzung-Pei Hong and S. Vincent Tseng, "Efficient algorithms for mining up-to-date high-utility patterns", Journal of Advanced Engineering Informatics, vol. 29, no.3, pp. 648–661, 2015.

[34] K. Lin , I-En Liao , T. Chang and S. Lin, "A frequent itemset mining algorithm based on the Principle of Inclusion–Exclusion and transaction mapping", journal of Information Sciences, vol. 276, pp.278–289, 2014.

[35] K. Lakshmanna, N. Khare. "Constraint-Based Measures for DNA Sequence Mining using Group Search Optimization Algorithm." International Journal of Intelligent Engineering & systems 9.3 (2016): 91-100.

[36] K. Lakshmanna, K. Rajesh , G. Thippa Reddy, G. Nagaraja, D.V. Subramanian. "An Enhanced Algorithm For Frequent Pattern Mining From Biological Sequences" International Journal Of Pharmacy & Technology 8.2 (2016): 12776-12784.

[37] Q.Wang, C. DarrylNDavis, JiadongRen, "Mining frequent biological sequences based on bitmap without candidate sequence generation "Computers in Biology and Medicine 69 (2016) 152–157