International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Constraint-Based Measures for DNA Sequence Mining using Group Search Optimization Algorithm

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

[1]*VIT University, India*
[2]*VIT University, Vellore, Tamil Nadu, India*
*Corresponding author's Email: kuruvalakshmanna0783@gmail.com

Abstract: In this paper, we propose a 3-step DNA sequence mining algorithm, called 3s-DNASM, incorporating prefix span, length and width constraints and group search optimization. The complete mining process is comprised into following vital steps: 1) applying prefix span algorithm, 2) length and width constraints, 3) Optimal mining via group search optimization (GSO). We first present the concept of prefix span, which detects the frequent DNA sequence. Based on this prefix tree, length and width constraints are applied to handle restrictions. Finally, we adopt the group search optimization (GSO) algorithm to completeness of the mining result. The experimentation is carried out using DNA sequence dataset, and the evaluation with DNA sequence dataset showed that the 3s-DNASM system is good for sequence mining. The simulation results illustrated that when *min_support=4*, the number of DNA sequence mined only 29 patterns by 3s-DNASM system, and in this case prefix span mined about 2168 patterns.

Keywords: DNA sequences; group search optimization; constraint; prefix span; mining.

1. Introduction

Data mining techniques are employed to make calculations and naturally employing only current static data. Sequence mining is a unique case of structured data mining and concerned with discovering statistically related patterns among data examples where the values distributed in a sequence. These values distributed and next accumulated in vast collections of data; Examples of such collections comprise transaction databases where the DNA sequence databases and web site usage logs are accessible [1]. The attitude of sequence mining is to find out constructive sequential knowledge. Knowledge attains the shape of insight into the structure of the data, "as it is structure that formulates things expected, and it is predictability that can be used" [2]. In addition, the frequent pattern mining of the DNA sequence is an imperative mean to revise the structure and function of the DNA sequence [3]. The Sequential Pattern Mining is a significant branch of data mining, which is one of the significant methods of data mining [4]. The sequential pattern mining is splitted into two categories: The initial kind, the frequent patterns mining base on sequence alignment, Such as FASTA, [5] etc. The next is the pattern mining base on the frequent pattern mining algorithm in the field of data mining.

Frequent pattern mining performs an important role in mining associations [6], correlations [7], causality, sequential patterns [8], episodes, multidimensional patterns, max-patterns [9], partial periodicity, emerging patterns [10], and several other significant data mining tasks. Moreover, Data mining [11] can be considered as an algorithmic process that takes sample as an input and give up patterns such as classification rules, association rules, or clustering as an output. The dissimilar soft computing methodologies are emphasized along with a recently introduced paradigm such as soft set theory [12]. In addition, data mining is employed in the calculation of gene relations in a genome, understanding of relations for region activation in the brain, and the prediction of protein folding effecting from changes in the DNA. Sequence study is a most important area in bioinformatics encompassing the techniques for learning the biological sequences, DNA, RNA, and proteins, on the linear structure level [13]

One of the most studied data mining tasks is Classification. The purpose is to forecast the value (the class) of a user-specified goal attribute based on the values of other attributes, called the predictive (feature) attributes. With regards to DNA, clustering is broadly employed in genome database. Several techniques were suggested to cluster genome sequences and DNA microarrays [14]. On the other hand, there is very minute research in the area of employing DNA computation for clustering. A few plans to employ DNA computing to work out clustering problems [15]. In addition, a small number of decades witnessed the individual and joined attempts of data mining and soft computing in the realm of bioinformatics [16]. In the DNA sequence mining Soft computing methodologies (involving fuzzy sets, neural networks, genetic algorithms, rough set, and soft sets) can be broadly employed. There are numerous general classification models such as Naive Bayesian Network [17, 18, 19], Decision Tree, Neural Networks, and Rule Learning using Evolutionary Algorithm [20].

The main intension of this paper is to, constrain based DNA sequence using prefix span, length and width constraints and group search optimization. This process is consisting of mainly three important steps such as (i) apply prefix span algorithm (ii) length and width constrains and (iii) optimal mining via group search optimization algorithm. Here, at first we apply the prefix span algorithm to the dataset which detects the frequent DNA sequence patterns. After that, we apply the length and width constrains to the frequent dataset. Finally, we adopt the group search optimization (GSO) algorithm for completeness of the mining result. The rest of the paper organized as follows: the recent research works is analyzed in section 2; the 3rd section describes the basic explanation of GSO; the proposed work is briefly explained in section4; in the 5th section, the experimental results are depicted and the section 6 represents the summary of the paper.

2. Related Works

For DNA sequence mining literature presents several theories. Now we assess some of the works associated to it**;** The CLIQUE algorithm using DNA computing techniques based on closed-circle DNA sequences has been described by H. Zhang and X. Liu [21]. DNA computing has been used in broad fields such as graph theory, finite state problems, and combinatorial problem. The CLIQUE (Clustering in QUEst) algorithm was one of the gird-based clustering techniques for spatial data. It was the combinatorial difficulty of the density cells. Hence they employ DNA computing by means of the closed-circle DNA sequences to implement the CLIQUE algorithm for the two-dimensional data.

Moreover, the recognition of promoters in DNA Sequences Using Weightily Averaged One-dependence Estimators has been made cleared by Z. Z. Htike and S. L. Win [22]. This paper explains a state-of-the-art machine learning based approach called weightily averaged one-dependence estimators to deal with the problem of identifying promoters in genetic sequences. To lower the computational complexity and to raise the generalization capability of the system, they use an entropy-based feature extraction approach to choose related nucleotides that are openly liable for promoter recognition.

Additionally, K. S. Leung et al [23] have made cleared the data mining of DNA sequences of Hepatitis B Virus. Their work group gathered HBV DNA sequences, either genotype B or C, from over 200 patients particularly for the project. In the molecular evolution study and clustering, three subgroups were recognized in genotype C and a clustering method was improved to detach the subgroups. In the feature selection process, potential markers were chosen based on Information Gain for further classifier learning. Next, meaningful rules were studied by our algorithm called the Rule Learning, which was based on Evolutionary Algorithm.

Likewise, in multiple biological sequences L. Chen and W. Liu [24] have made cleared the frequent patterns mining. Now they initially made cleared the idea of primary pattern, which was expanded to form larger patterns in the series. A prefix tree was erected to identify frequent primary patterns. A. Nakamura et al [6] have explained the Mining approximate patterns with frequent locally optimal occurrences. Here, they explained a candidate patterns are generated without duplication using the suffix tree of a given string.

In, X. Wu et al [25] have made cleared the problem of frequent pattern mining without user-specified gap constraints and brought in PMBC (namely Pattern Mining from Biological sequences with wildcard Constraints) to work out the problem. Specified a sequence and a support threshold value (i.e. pattern frequency threshold), PMBC proposes to discover all subsequences with their support values equal to or greater than the specified threshold value. Jerry et al [5] have explained the efficient algorithms for mining up-to-date high-utility patterns. It considers not only utility measure but also timestamp factor to discover the recent HUPs. The UDHUP-apriori was first introduced to mine UDHUPs in a level-wise way.

Moreover, U. Kamath et al [26] have made cleared the Evolutionary Algorithm Approach for Feature Generation from Sequence Data and its Application to DNA Splice Site Prediction. Now, they made cleared evolutionary algorithm to successfully investigate a huge feature space and produce predictive features from sequence data. The efficiency of the algorithm was shown on an imperative component of the gene-finding problem, DNA splice site prediction.

3. Terms Related to Group Search Optimization (GSO)

Group search optimizer (GSO) [27] is a population based optimization algorithm, motivated by animal foraging behavior. GSO Optimization uses the producer–scrounger (PS) model and the animal scanning mechanism. Scanning is the important element of search criterion and it can be completed through physical contact. In GSO, vision as the fundamental scanning approaches introduced by white crappie is used. The population of the GSO algorithm is named a group and each individual in the population is named a member. During each iteration, a member is described by its position and head angle. Every member has its present position $Y_i^k \in R^n$, a head angle $\theta_i^k = \left(\theta_i^k, \ldots, \theta_{i(n-1)}^k\right) \in R^n$. The search direction of the i^{th} member, $G_i^k(\theta_i^k) = \left(g_{i1}^k, \ldots, g_{in}^k\right) \in R^n$ that can be computed from θ_i^k through polar to Cartesian coordinate transformation [28],

$$g_{i1}^{k} = \prod_{q=1}^{n-1} \cos\left(\phi_{iq}^{k}\right) \tag{1}$$

$$g_{ij}^{k} = \sin\left(\phi_{i(j-1)}^{k}\right) \bullet \prod_{q=i}^{n-1} \cos\left(\phi_{iq}^{k}\right) \tag{2}$$

$$g_{in}^{k} = \sin\left(\phi_{i(j-1)}^{k}\right) \tag{3}$$

(i) **For producer**: The producer scrutinizes ate zero degree and checks three points toward its position by means of equations (4)-(6),

$$Y_z = Y_q^k + r_1\, l_{\max}\, G_q^k(\theta^k) \tag{4}$$

$$Y_r = Y_q^k + r_1\, l_{\max}\, G_q^k\left(\theta^k + \frac{r_2 \phi_{\max}}{2}\right) \tag{5}$$

$$Y_l = Y_q^k + r_1\, l_{\max}\, G_q^k\left(\theta^k - \frac{r_2 \phi_{\max}}{2}\right) \tag{6}$$

Where $r_1 \in R_1$ a normally distributed arbitrary number with is mean 0 and standard deviation 1 and $r \in R_{n-1}$ is an arbitrary sequence in the range (0, 1). If the producer finds the best point with the best resource from above three points than its present position, it will fly to this point, or else it will stay in its present position and turn its head to a novel angle:

$$\theta^{k+1} = \theta^k + r_2\, c_{\max} \tag{7}$$

(ii) Where, $c_{\max}$ is the maximum turning angle. If the producer can never find a better area after c iterations, it will turn its head back to zero degree:

For scroungers: The area replicating behavior of scroungers can be modeled as a random walk towards the producer:

$$Y_i^{k+1} = Y_i^{k} + r_3\left(Y_d^k - Y_i^k\right) \tag{8}$$

Where, $r_3 \in R_n$ is a uniform random sequence in the range (0, 1).

(iii) **For rangers**: Rangers move to the novel point through generate a random head angle and select a random distance:

$$\theta^{k+1} = \theta_i^k + r_2\, c_{\max} \tag{9}$$

$$l_{\max} = c.r_1\, l_{\max} \tag{10}$$

$$Y_i^{k+1} = Y_i^{k} + l_i G_i^k(\theta^{k+1}) \tag{11}$$

4. Proposed Algorithm for DNA Sequence Mining

With novel features the growth of bioinformatics has effected in datasets. The DNA sequences classically enclose a bigger number of items. According to this, the frequent pattern mining of the DNA sequence is an important mean to learn the structure and function of the DNA sequence. From them biologists pull together a whole genome of species based on frequent sequences. These frequent sequences normally have hundreds of items. How to competently find out long frequent sequence creates a great dispute for presented sequential pattern discovery algorithms. In current years, there are huge numbers of sequence pattern mining algorithms in the field of bioinformatics. In this revise, we offer 3-step DNA sequence mining algorithm, called 3s-DNASM combining prefix span, length and width constraints and group search optimization. The specified DNA sequence mining process is comprised into following significant steps: 1) applying prefix span algorithm, 2) length and width restrictions, 3) Optimal mining through group search optimization (GSO). In figure 1, the overall diagram of the suggested approach is offered:

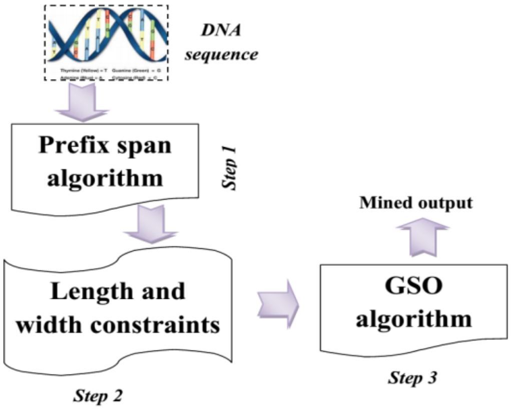


Figure.1 Flow diagram of the proposed approach

Here, we look at different mining algorithms and constrains on the DNA sequences, which include prefix span, length and width constraints and GSO based mining.

4.1. Prefix-Span Algorithm

The CGSO-DNASM approach begins with prefix span (Prefix-projected Sequential pattern mining) algorithm, which presents ordered growth and decreased projected databases. In the initial step, this prefix span algorithm is carried out to mine the initial level mining process. Let our running database be DNA sequence database DB specified in Figure 2 and min_support =2.

Id	Sequence
10	<A(AGC)(AG)T>
20	<(AG)T(CT)(AT)>
30	<(GC)(AT)(CT)>
40	<A(AT)>

Figure.2 DNA Sequence database

4.1.1. Find length 1 sequential patterns

Find length of sequence patterns for the DNA sequence database DB considering the minimum support that has been given. Initially, scanning process is prepared on database once to find all the frequent items in sequences. Each of these frequent items is a length-1 sequential pattern is revealed in figure 3. They are $\langle A\rangle:4,\langle G\rangle:3,\langle C\rangle:3$ and $\langle T\rangle:4$. Where the notation of $"\langle pattern\rangle:count"$ symbolizes the pattern and its related support count.

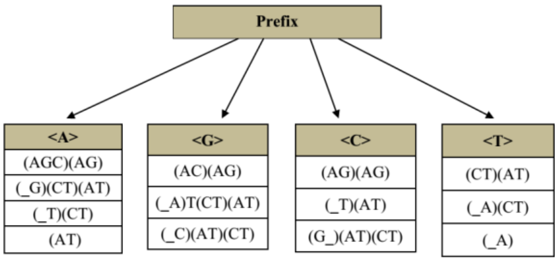
<A>	<G>	<C>	<T>
4	3	3	4

<A><G><C><T>

Figure.3 Obtained 1-length frequent itemsets

4.1.2. Divide the search space

Divide the search space into the prefixes whose support is greater than the minimum support. i.e., the complete patterns can be divided into the subsequent four prefixes: the ones with prefix $\langle A\rangle,\langle G\rangle,\langle C\rangle$ and $\langle T\rangle$. Figure 4 demonstrates the projected data base of specified DNA database.



<A>	<G>	<C>	<T>
(AGC)(AG)	(AC)(AG)	(AG)(AG)	(CT)(AT)
(_G)(CT)(AT)	(_A)T(CT)(AT)	(_T)(AT)	(_A)(CT)
(_T)(CT)	(_C)(AT)(CT)	(G_)(AT)(CT)	(_A)
(AT)			

Figure.4 Projected database

4.1.3. Find subsets of sequential patterns

The subsets of sequential patterns can be looked for by building projected databases of the prefixes supported. The subsets of sequential patterns can be excavated by constructing the related set of projected databases and mining each recursively. In figure 5, the left side table demonstrates projection database and right side table symbolizes mined sequence of prefix <A>. The above process is replicated for all prefix <G>, <C>, and <T> and those are tabulated in table 1.

<A>
(AGC)(AG)
(_G)(CT)(AT)
(_T)(CT)
(AT)

A	~~G~~	C	T
3	~~2~~	3	3

<AA><AC><AT>

Figure.5 Prefix of <A> and mined sequence

Table 1. Projected database and mined sequence

refix	Projected database	Sequential database
$\langle A \rangle$	$\langle AGC\rangle\langle AG\rangle\langle \_G\rangle$ $\langle T\rangle\langle CT\rangle\langle AT\rangle\langle \_T\rangle$ $\langle CT\rangle\langle AT\rangle$	$\langle AA\rangle\langle AC\rangle\langle AT\rangle\langle AG\rangle$
$\langle G \rangle$	$\langle AC\rangle\langle AG\rangle\langle \_A\rangle$ $\langle T\rangle\langle CT\rangle\langle AT\rangle\langle \_C\rangle$ $\langle \_C\rangle\langle AT\rangle\langle CT\rangle$	$\langle GA\rangle\langle GC\rangle\langle GT\rangle$
$\langle C \rangle$	$\langle AG\rangle\langle AG\rangle\langle \_T\rangle$ $\langle AT\rangle\langle G\_\rangle\langle AT\rangle\langle CT\rangle$	$\langle CA\rangle\langle C\Gamma\rangle\langle CT\rangle$
$\langle T \rangle$	$\langle CT\rangle\langle AT\rangle\langle \_A\rangle$ $\langle CT\rangle\langle \_A\rangle$	$\langle TA\rangle\langle TC\rangle\langle TT\rangle$

4.2. Handling Length and Width Restrictions

Constraint-based mining may overcome over the difficulties of efficiency and competence as constraints frequently symbolize user's interest and focus, which limits the patterns to be found to a specific subset satisfying some strong conditions. In this document, we just add a check to see if $width\ (S) \leq \max_w$ and $length\ (S) \leq \max_l$, where $\max_w$ and $\max_l$ are the user specified restrictions on the maximum permissible width and length of a DNA sequence. Incorporating restrictions on the length and width of the sequences is clear-cut.

4.3. Optimal Mining Via GSO

The fullness of the mining process is prepared through group search optimization after length and width restrictions. The optimization is prepared to decrease the redundancy and duplication the sequences from the DNA sequence data. The GSO functions under a particular procedure, such as producer, scrounger and ranger. A DNA sequence obtains optimized according its fitness function by GSO algorithm.

- **Fitness function for GSO**

The fitness for the functions is planned based on the support, confidence, frequency and lift parameters of the suggested approach. The minimum support is the support necessary to maintain a sequence regarding the DNA sequence database. The minimum support is symbolized as main support and is described as,

$$\min\_\sup port = \frac{T(U,V)}{T_n} \qquad (12)$$

Where $T(U,V)$ is the number of transactions, which enclose the sequences and T_n is the total number of transactions. The other characteristics, which is referred in the fitness function are confidence and lift parameters. The parameters confidence and lift are obtained from the parameter support. The parameter can be obtained as,

$$confidence = \frac{Support(U \cup V)}{Support(U)} \qquad (13)$$

$$lift = \frac{Support(U \cup V)}{Support(U) \times Support(V)} \qquad (14)$$

Hence based on these parameters, we expand a fitness function for the suggested method for optimizing the sequences. A sequence obtains optimized based on the relevance of the demand of the situation. In the proposed approach, the demand is getting the maximum revenue from the financial transactions. Thus the sequences should be particular to avail maximum revenue from the item sets which comprises the sequences. Hence the fitness should enclose the minimum support value of the sequences, as it provides the relevance of accepting a sequence through the item sets. The fitness is described by the relating the confidence, support and the lift value.

$$\begin{gathered} fitness = conf(s) + \log(Support(s) \times \\ (\min\_Suppot(l) + \min\_Support(s)) * lift(s)) \end{gathered} \qquad (15)$$

For obtaining the maximum revenue out from sequences the fitness value specifies the sequences to be significant. The fitness values are chosen to differentiate between the sequences, those with high possibility of getting maximum revenue and those with low possibility of getting maximum revenue. The difference is made by applying a threshold value with the fitness values, the fitness value, which is higher than that of threshold value are chosen and rest are thrown away.

$$opt\_seqs = \begin{cases} select,\ fitness > threshold \\ discard,\ fitness < threshold \end{cases} \qquad (16)$$

Where, opt_seqs represents the sequences which are getting optimized according the GSO algorithm.

- **Optimization of sequences via GSO Algorithm**

In the field of data optimization, the group search optimization algorithm is one of the most generally employed optimization technique. In the present scenario, we employ group search optimization algorithm to optimize the sequences attained from the mining process. The GSO considers the extracted sequences as the first population. The first population is subject for fitness calculation based on the fitness function. The population is practiced through different stages like, producer, scrounger and ranger as described by the GSO algorithm. Let us reflect on the first population be as follows,

$$s_p = [s_1, s_2, ..., s_n] \quad (17)$$

The set s_p represents the population of the extracted sequence and the individuals in the population are represented with s_1 to s_n . The individuals in the population are then formatted with the fitness derived in the above section.

$$\begin{aligned} & fitness = conf\,(s) + \log(Support(s) \times \\ & (\min\_Support(i_{left}) + \min\_Support(i_{right})) \times lift(s)) \end{aligned} \quad (18)$$

Now, i_{left} and i_{right} are the item sets in left side and right side of a sequence correspondingly. Once all the fitness values are computed, the fitness values are supplied to a fitness set, which encloses the fitness of the sequences.

$$f = [f_1, f_2, ..., f_n] \quad (19)$$

Now, the fitness between the old sequence and novel sequence are compared and the one with higher fitness is maintained. If the novel sequence has better fitness, it will be substituted with the old sequence. Alternatively, if the old sequence has higher fitness, it will be subjected for development in the next iteration of genetic algorithm. Likewise, the processes are prolonged up to each sequence is being revised. The last step of the genetic algorithm is optimizing the sequences based on the fitness threshold. A set for optimized sequence is produced for storing the optimized sequences from the extorted sequences based on the fitness, defined by S_{op} . Reflect on the set of sequences be S , and S_d be the rejected sequences.

$$s_i \in S = \begin{cases} r_i \in S_{op}, \ fitness > threshold \\ r_i \in S_d, \ fitness < threshold \end{cases} \quad (20)$$

The above expression specifies, 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 GSO algorithm as follows:

1) Generate an initial population of member randomly (described in equation (17)).
2) Evaluate the fitness of each member in the population using equation (15).
3) Create a new population by replacing the updation steps (a, b and c) until the new population is complete.
   a. Perform producer operation using equation (4, 5, and 6)
   b. Perform scrounger operation using equation (8)
   c. Perform ranger operation using equation (11)
4) If the test condition is satisfied, stop and return the best solution in the current population.
5) Repeat step 3 until the target is met.

Pseudo code of 3s-DNASM

Input: *DNA sequence* S *; min_support =2.*

Output: *Complete set of sequential DNA patterns* S_3

Assumptions

α → *Sequential pattern*

α' → *Projected database*

L → *Length of* α

$S|\alpha$ → *The* α *projected database, if* $\alpha \neq <>$ *, otherwise the sequence database* S

k → *Particular frequent item*

Y_p → *Producer*

Lg → *Length constraint*

Wg → *Width constraint*

T → *Iteration*

i → *Member*

Y_i → *Position*

ϕ_i → *Head angle*

$f(Y_i)$ → *Fitness function of position* Y_i

S_1 → *First step mined result*

S_2 → *First step mined result*

S_3 → *Third step mined result*

Pseudo code:

```
Begin
    Scan $S|\alpha$ once
    Find the set of frequent items $k$ such that:

    $k$ can be assembled to the last element of α to form a    sequential pattern; or
    $< k >$ can be appended to α to form a sequential pattern.
    For each frequent item $k$, append it to $\alpha$ to
generate a sequential pattern $\alpha'$ and show $\alpha'$
    as output.

    For each $\alpha'$ construct $\alpha'$-projected
database $S|\alpha'$ and repeated the steps.
   Obtain first step mined set $S_1$
   Apply length constraint $Lg$
     Apply width constraint $Wg$
    Obtain second step mined set $S_2$

     Set T=0
    Randomly initialize the position $Y_i$ and head
angle $\phi_i$ of all members
    Calculate the fitness value of the initial members:
$f(Y_i)$
        While (the termination conditions are not met)
          for (each members $i$ in the group)
         Choose producer: Find the producer $Y_p$ in
the group;
        Perform scrounging: Randomly select 80%
from the rest members to perform
                                    scrounging.
           Perform ranging: For the rest members they
will be carry out ranging
           Verify feasibility: Check weather each
member of the group violates the constraints.
                                   If it does, it will move
back to the previous position to guarantee a
                                   feasible Solution.
                       end for
     Set t=t+1;
                  end while
     Obtain third step mined set $S_3$
end
```

5. Result and Discussion

In this section, the experimental results of the proposed approach for DNA sequence mining by effectual mining of sequential patterns using 3s-DNASM algorithm is described. We evaluate the efficiency and performance of algorithm 3s-DNASM and compare it with the traditional algorithms prefix span algorithm. In this approach, we use two set of DNA sequence dataset such as AF008216.1 (dataset 1) and AF348525.1 (dataset 2) [9]. The DNA to be sequenced is prepared as a single strand. The DNA sequence presents the dideoxy nucleotides (A, G, C and T). The proposed approach has been programmed using JAVA (jdk 1.6) and the experimentation is performed on a 3.0 GHz Pentium PC machine with 2 GB main memory.

5.1. Experimental Result Analysis

The basic idea of our research is to DNA sequence mining using 3s- DNASM algorithm. Here, at first we apply the prefix span algorithm to the DNA sequence. In our proposed approach the patterns are mined into three stages by using 3s-DNASM algorithm. In the first stage, the DNA sequence database is constructed for the input database and then, we mined sequential patterns from the DNA sequence database based on the prefix span algorithm. After that we add the two constrains such as weight and length, based on the constrained the sequence are mined. Finally, we apply the GSO optimization algorithm for the obtained mined sequence. The following graphs show our proposed approach experimental result outputs.

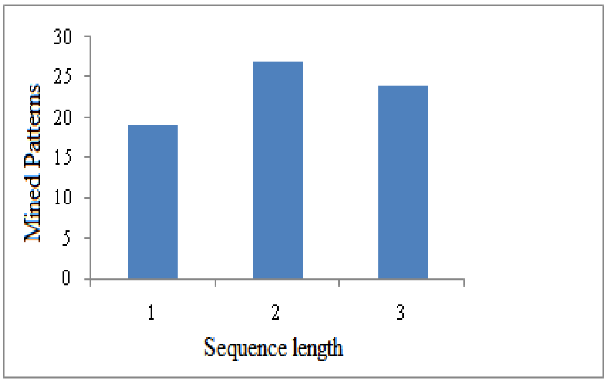


Figure.6 Number of patterns mined for varying length in dataset 1

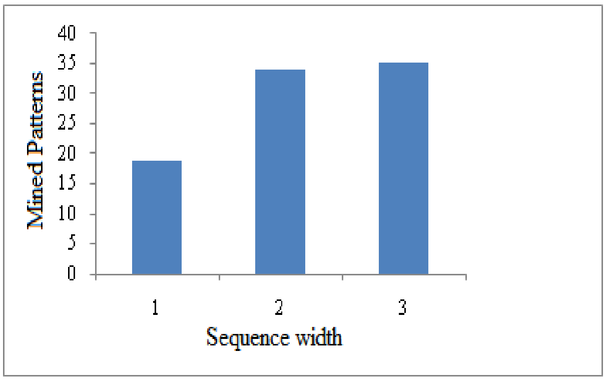


Figure.7 Number of pattern mined for varying width in dataset 1

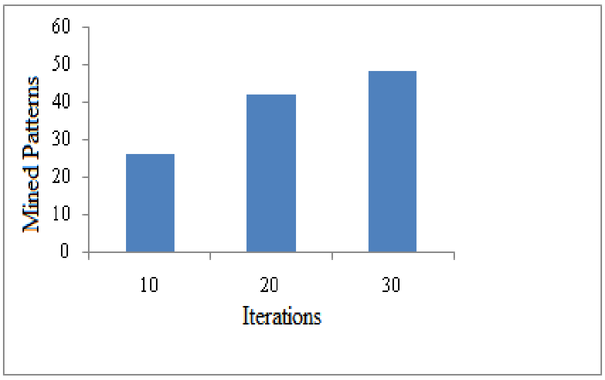


Figure.8 Number of pattern mined for varying iterations in dataset 1

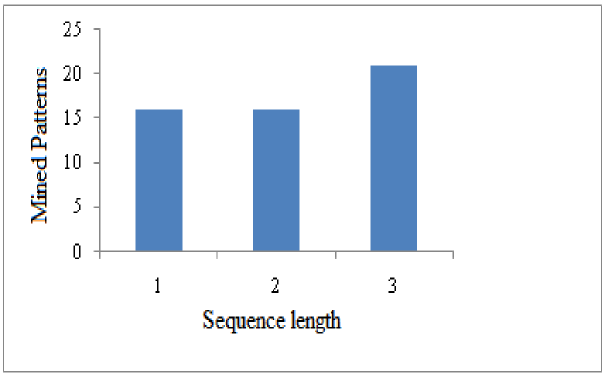


Figure.9 Number of pattern mined for varying length in dataset 2

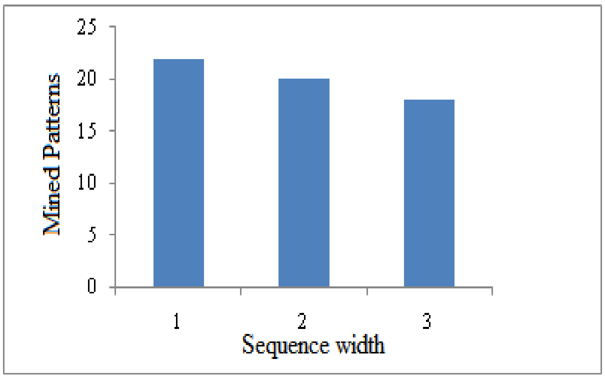


Figure.10 Number of pattern mined for varying width in dataset 2

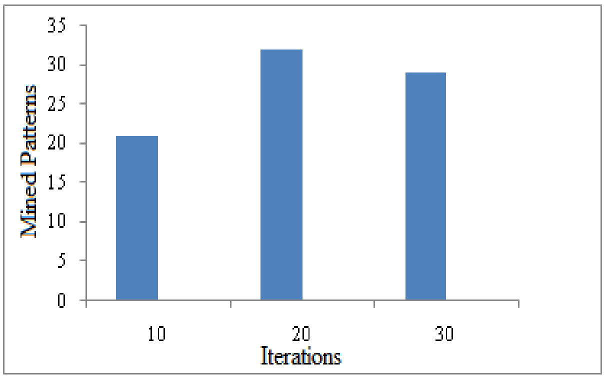


Figure.11 Number of pattern mined for varying iteration in dataset 2

The experimental results obtained from the proposed approach with the 2 types of DNA sequence datasets are described in the above figures. Initially, the input dataset is given to the proposed approach of 3-stage DNASM algorithm to mine the sequence. The mining performance with respect to the mined sequence is given in the above graphs shown in Figures 6 to 11. Figure 6 shows the total number of mined patterns by varying the length of the sequence. We obtain the minimum sequence pattern 19 for the length of the sequence is 1. In the same figure 6, we obtain the mined sequence is 30 for the length of the sequence is 3. Figure 7 shows the total number of mined patterns by varying the width of the sequence. When the width value is 1, we obtain the minimum mined sequence and when the width is 3 we obtain the maximum mined sequence which is 30 using 3s- DNASM algorithm. Figure 8, shows that we obtain the total number of mined patterns by varying the iteration. From the figure, we understand the number of iteration increase as the total number of mined sequence also increases. Similarly, figure 9, 10 and 11 shows the experimental result based on the dataset 2.

5.2. Comparative analysis of proposed approach

This section describes the comparative analysis of the proposed approach 3s- DNASM to prefix span method. The comparative result clearly ensures that the proposed approach provides optimal order of sequential patterns compared to Prefix Span algorithm. In the proposed approach, we use length is 3 and the width is 2. The table 2 shows the comparative result of DNA sequence mining.

Table 2. Comparative analysis of the proposed approach

Dataset 1			Dataset 2		
Min-support	Mined patterns		Min-support	Mined patterns	
	3s-DNASM	Prefix Span		3s-DNASM	Prefix Span
2	30	13118	2	15	8072
3	27	6142	3	21	3636
4	32	3768	4	29	2168

With the same viewpoint, two comparable datasets are taken and the format examples are created from the first dataset. At that point, the second dataset is given to the mining strategy such a way; the comparable examples mined from these two datasets are assessed. Actually, the second dataset is synthetically generated with the view of installation which will be done in respect to the first patterns. The support of the mined patterns from first dataset is relatively high in the second dataset as per the table given in Table 2. Here, we use the constrains length as 3 and width as 2. From the table 2, our proposed approach achieves the minimum mined pattern compared to the Prefix span method. When the minimum support is 2 we obtain the mined pattern of 30 for proposed approach and 13118 for prefix span approach which is associated to the dataset 1. Similarly, when using the minimum support is 3 we obtain the mined pattern of 27 for proposed approach and 6142 for using prefix span approach which is associated to the dataset 1. In the same way, we obtain the mined pattern is minimum when using the minimum support is 4 using 3s DNASM. Similarly, dataset 2 also we obtain the better result compare to the prefix span approach. From the comparison results, our approach is identified and mined minimum patterns, which proved that constraint, useful and meaningful patterns only mined.

6. Conclusion

We developed a 3-step DNA sequence mining algorithm, called 3s-DNASM, incorporating prefix span, length and width constraints and group search optimization. The detail DNA sequence mining process contains following vital steps: 1) applying prefix span algorithm, 2) length and width constraints, 3) Optimal mining via group search optimization (GSO). In the first step, the concept of prefix span is presented, which detects frequent DNA sequence pattern using prefix tree. After prefix tree construction, in the second step, Length and width constraints were applied to handle restrictions. Finally, optimized mining result was obtained through group search optimization. The experimentation was carried out on the standard DNA sequence data set, and the evaluation with DNA sequence dataset showed that the 3s-DNASM system was good for DNA sequence mining. The experimentation results demonstrated that proposed 3s-DNASM system achieved higher quality results compared with other methods. In future I will plan to add multiple constraints in sequential pattern mining process. Also we may apply hybrid optimization algorithm in future.

Reference

[1] Z. S. Zubi1 and M. A. Emsaed, “Sequence Mining in DNA chips data for Diagnosing Cancer Patients”, *applied computer science*, Vol. 4, No. 4, pp. 139-151, 2010.

[2] P. Hingston, “Using Finite State Automata for Sequence Mining”, *Australian Computer Society*, Vol. 24, No. 1, pp. 105-110, 2001.

[3] S. Bai and X. Dai, "An Efficiency apriori Algorithm: P_Matrix Algorithm", *First International Symposium on Data, Privacy, and E-Commerce*, pp. 101-103, 2007.

[4] R. Srikant and R. Agrawal, “Mining sequential patterns: Generalizations and performance improvements”, *springer*, pp. 1-17, 2005.

[5] J. C. W. Lin, W. Gan, T. P. Hong and V. S. Tseng, “Efficient algorithms for mining up-to-date high-utility patterns”, *Journal of Advanced Engineering Informatics*, Vol. 29, No. 3, pp. 648-661, 2015.

[6] 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

[7] S. Brin, R. Motwani, and C. Silverstein, “Beyond market basket: Generalizing association rules to correlations”, *In SIGMOD'97*, Vol. 26, No. 2, pp. 265-276, 1997.

[8] R. Agrawal and R. Srikant, “Mining sequential patterns”, *In ICDE*, pp. 3-14, 1995.

[9] Site: http://www.ncbi.nlm.nih.gov/

[10] G. Dong and J. Li, “Efficient mining of emerging patterns: Discovering trends and differences”, *In KDD*, pp. 43-52, 1999.

[11] Han and Kamber, “Data Mining: Concepts and Techniques”, *Morgan Kaufmann*, 2006.

[12] D. Molodtsov, “Soft set theory first results”, *Computers and Mathematics with Applications*, Vol. 37, No. 4, pp. 19-31, 1999.

[13] S. K. Pal, “Soft data mining, computational theory of perceptions, and rough-fuzzy approach”, *Information Sciences*, Vol. 163, No. 1, pp. 5-12, 2004.

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

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

[16] S. Mitra and Acharya, “Data Mining: Multimedia, Soft Computing, and Bioinformatics”, *John Wiley & Sons*, 2003.

[17] C. Eugene, “Bayesian Network without Tears”, *AI Magazine*, Vol. 12, No. 4, pp. 50-63, 1991.

[18] D. M. Chickering, D. Heckerman, and D. Geiger, “Learning Bayesian Networks: The Combination of Knowledge and Statistical Data”, *Machine Learning*, Vol. 20, No. 3, pp. 197-243, 1995.

[19] 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*, pp. 83-90, 1997

[20] W. Banzaf, P. Nordin, R. Keller and F. Francone, “Genetic Programming - An Introduction”, *Morgan Kaufmann*, 1997.

[21] H. Zhang and X. Liu, “A CLIQUE algorithm using DNA computing techniques based on closed-circle DNA sequences”, *Biosystems*, Vol. 105, No. 1, pp. 73-82, 2011.

[22] Z. Z. Htike and S. L. Win, “Recognition of Promoters in DNA Sequences Using Weightily Averaged One-dependence Estimators”, *Procedia Computer Science*, Vol. 23, pp. 60-67, 2013.

[23] K. S. Leung, K. H. Lee, J. F. Wang, E. Y. T. Ng, H. L. Y. Chan, S. K. W. Tsui, T. S. K. Mok, P. C. Hang Tse, and J. J. Y. Sung, “Data Mining on DNA Sequences of Hepatitis B Virus”, *IEEE/ACM transactions on computational biology and bioinformatics*, Vol. 8, No. 2, pp. 428-440, 2011.

[24] L. Chen and W. Liu, “Frequent patterns mining in multiple biological sequences”, *computers in biology and medicine*, Vol. 43, No. 10, pp. 1444-1452, 2013

[25] X. Wu, X. Zhu, Y. He and A. N. Arslan, “PMBC: Pattern mining from biological sequences with wildcard constraints”, *computers in biology and medicine*, Vol. 43, No. 5 pp. 481-492, 2013.

[26] U. Kamath, J. Compton, R. I. Dogan, K. D. Jong and A. Shehu, “An Evolutionary Algorithm Approach for Feature Generation from Sequence Data and its Application to DNA Splice Site Prediction”, *IEEE transactions on computational biology and bioinformatics*, Vol. 9, No. 5, pp. 1387-1398, 2012.

[27] S. He, Q. H. Wu, and J. R. Saunders, “Group Search Optimizer: An Optimization Algorithm Inspired by Animal Searching Behavior”, *IEEE Transactions On Evolutionary Computation*, Vol. 13, No. 5, pp. 973-990, 2009.

[28] D. Mustard, “Numerical integration over the n-dimensional spherical shell”, *Math Computing*, Vol. 18, No. 88, pp. 578-89, 1964.