© 2016 IJSRSET | Volume 2 | Issue 5 | Print ISSN: 2395-1990 | Online ISSN : 2394-4099
Themed Section: Engineering and Technology

Cancerous/Disease DNA Prediction Using Fixed Length Motifs/Frequent Patterns Matching

Adnan Ferdous Ashrafi, Shah S. Mahin, Tarikuzzaman Emon

Department of Computer Science and Engineering, Stamford University Bangladesh, Dhaka, Bangladesh

ABSTRACT

In the radical field of bioinformatics, one very interesting and rather concerning area of research is predicting cancer infected gene from a set of samples of species DNA. This field is quite a challenging one considering the limited knowledge on how cancers affect gene of species and the pattern of mutation are not always the same. Gene prediction can be effectively done through several techniques like frequent pattern mining, neural networks or sequence alignment. These traditional approaches were able to predict to a very small limit. In this paper a new method using frequent patterns/motifs is shown that can be a new strategy for prediction of gene in a DNA. As the motifs in a DNA are the conserved region, so it's more appropriate to be used for gene predication and alignment. The new method proposed in this paper includes the sampling of fixed length motifs from a sequence of reference genome and finally other samples are aligned against the more frequent motifs to establish their relevancy to the reference genome.

Keywords: gene prediction; cancer cell prediction; motifs; hash table; frequent pattern matching;

I. INTRODUCTION

Motif discovery in genomic sequences is defined as the problem of finding short similar and conserved sequence of elements shared by a set of nucleotide or protein sequences performing a common biological function. The identification of regulatory elements in nucleotide sequences, like transcription factor binding sites (TFBSs), has been one of the most widely studied sectors of the problem, both for its biological significance and for its computational hardness. Motifs are fundamental functional elements in proteins vital for understanding gene function, human disease, and may serve as therapeutic drug targets.

On the other hand frequent pattern matching and data mining for DNA/protein sequence analysis is being considered remarkable among the researchers for its worthy possibilities in the field of gene prediction/protein structure prediction. Normally data sequences are very large but in case of DNA sequence only A,C,T,G makes a nucleotide and so it is very natural that numerous combinations and permutations of A,C,T and G's will be repeated many times[3]. Thus, the importance of recognizing the correct DNA sequence pattern is easily understandable. There are a lot of aspects in DNA sequence that needs to be discovered for proper analysis. Reasons and causes of all diseases can be found if the data are properly analyzed and sorted. Analyzing the sequence we can find similarities and links between two sequences and it is possible to modify them accordingly [4]. Combining these two strategies a new noble approach can be achieved to prioritize cancerous cells of species.

II. METHODS AND MATERIAL

1. Literature Review

A. A Modified algorithm for DNA motif finding and ranking considering variable length motif and Mutation[1]

In this research work the authors proposed a noble approach towards optimizing PSO[3] algorithm for extraction of motifs from a dataset by using a hash table approach.

In this system a DNA sequence will be converted to a subsequent hash table that will hold all information from those DNA and can be used very efficiently to compare the motifs at hand with reference genome.

The next step is to reform the other sequences to their corresponding hash tables. Hash tables are nothing but a two dimensional array which indexes the positions of A, T, G and C's in each DNA sequence. The usefulness of this is the sequence is now automatically sorted and easier to manipulate.

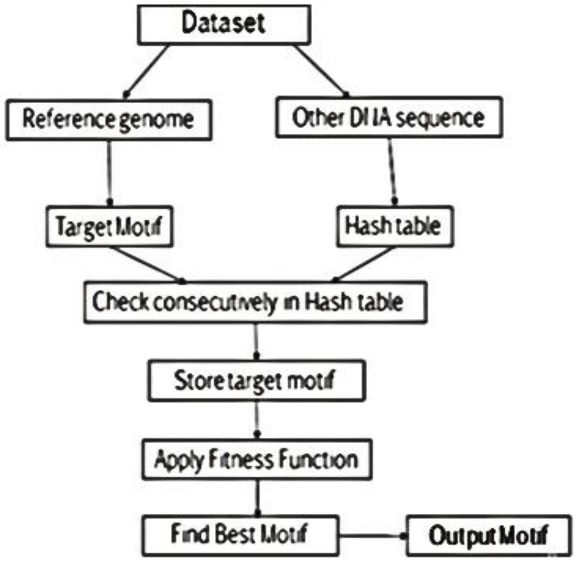


Figure 1: Flow Chart for motif finding and ranking [1]

Finally a more practical approach of fitness function[7] was used to denote the efficiency of the ranking. The fitness function included the following 4 equations/functions:

$$fitness\_score = w_1 * L + w_2 * r\ (k, m)$$

Here w_1 = weight given to length, L = length of motif, w_2 = weight given to similarity of the matched motif and the function *r (k, m)* calculates the relativity of the motif.

$$r(k,m) = \frac{d(k,m)}{d(k, M_{ref}) + c(k)}$$

This equation simply calculates the ratio of the distance from the whole consensus of the target motif samples i.e. $d(k,m)$ and the distance of the motif from known motifs $d(k, M_{ref})$ with its complexity factor score of $c(k)$.

$$d(k,m) = 1 - 1/k \left[\sum_{i=1}^{k} \sum_{\forall b_i \in x} f(b_i, i) * k(b_i, i) \right]$$

Here, the equation simply finds the hamming distance of the motif from the query sequence and normalizes the value to a range of 0 to 1.

$$c(k) = 4/3 \left[1 - \frac{1}{k^2 \left[\sum_{\forall b_i \in x} \left(\sum_{i=1}^{k} k(b_i, i) \right)^2 \right]} \right]$$

This equation finds the complexity of a target sequence and returns a higher score for a sequence that has lower number of repetitions of bases in it.

B. Mining Frequent Pattern within a Genetic Sequence Using Unique Pattern Indexing and Mapping Techniquess[2]

In this work two algorithms were proposed. The first one will simply format the input text file with DNA sequences of a sample species and then encode it using numerical values. The values were dynamically generated in the first run so that no redundant data is stored. Since the unique sequences were nothing more than various permutations and combinations of A, G, C, T (in case of DNA) of length four, a numerical value to each unique sequence was assigned and later replaced that whenever encountered within the complete DNA sequence.

Since at most 256 (44) combinations of A, C, G, T (of length 4) is possible, the ID value went up to 256 (for sequences of length 4) at worst case scenario.

As the whole DNA sequence is divided into sub-sequences of length four, around the end of the sequence there might be some stray sequence of length one, two, or three. For example, for the sequence AAATAGCTTATAGC, the program will extract and id the sub-sequences AAAT (1), AGCT (2), TATA (3). Since the last sequence encountered is GC which is of length two, the program will simply put it into the file as GC and ID it to 4.

The Unique Pattern Indexing[2] Algorithm indexes each unique pattern and puts it into a HashMap[2].

After processing the given input DNA sequence by replacing each unique sequence (primarily of length four) with its corresponding index, the input was then checked for frequent patterns of length 4, 8, 12, and 16 via Searching Frequent Pattern[2] Algorithm.

In order to build the initial database, sample DNA sequences are needed. The sample dataset were taken from NCBI database and are of different lengths. This gave a realistic simulation given the dataset are all obtained from different species of bacteria and viruses.

At first the Unique Pattern Indexing algorithm was applied on the sample data. To do that, first the (unique) DNA sequences were loaded within the database into a HashMap and stored as a key. Then they were numbered sequentially and indices were put into the value field of the HashMap.

After that, the file containing the target DNA sequence was loaded into a StringBuilder. Now subsequences of length 4 were extracted from the beginning of the target sequence and checked if it exists as a key within the HashMap. If it exists, the value was extracted for that key and put it into the output file. Otherwise the subsequence was first put into the HashMap, assigned a new value to it, and then put that value into the output. To exemplify, consider the input string AAGTACTTTATAACTTTATA. Now, the algorithm will automatically assign the index value 1 to AAGT, 2 to ACTT and 3 to TATA etc. Since ACTT already got a value assigned, it will just be replaced with 2. So, the output will be 1 2 3 2 3. In this manner, the sequence was traversed up until the very end. If a sequence of length 3 or less was encountered and it does not exist in the HashMap, it was treated as a new unique sequence and put it into the DNA database.

At this point the Frequent Pattern Searching algorithm was employed. At first, the output file from Unique Pattern

Indexing algorithm was loaded the whole sequence was treated as a continuous string.

2. Proposed Methodology

In our paper, we propose a new searching algorithm based on the PSO [3] where for each cycle only one particle called 'target motif' are selected and compared with the DNA sequences for fitness calculation.

In our system a new searching algorithm[1] was used to compare the target motif with other DNA sequences in effective way. Basically we have introduced the effective use of a 'hash table' in searching process which brings flexibility and simplicity. Each sequence is converted to a corresponding index hash table. So instead of string matching we are trying for integer matching which is faster in all sense. It's also helpful for time consuming. It reduces the redundancy which we face in the linear search. So, we improved the method by effectively using the hash table concept [6] to compare the long DNA sequences in an effective way.

The information in DNA is stored as a code made up of four chemical bases A, T, C, and G. In our proposed method the main focus is on finding the best suited motif sequence from a set of DNA sequences. To do that we assumed the chemical bases as characters and the DNA sequences as strings. Initially we will select some random DNA sequences of same species. Here for our experimentation we chose a specific dataset (GenBank: ACOT2 acyl-CoA thioesterase) for DNA sequences.

A. Target motif extraction[1]

In order to extract motifs for comparison first sequence was chosen as the reference genome and the rest were converted to corresponding hash tables. From the reference DNA sequence motifs were extracted of the exact length 6.

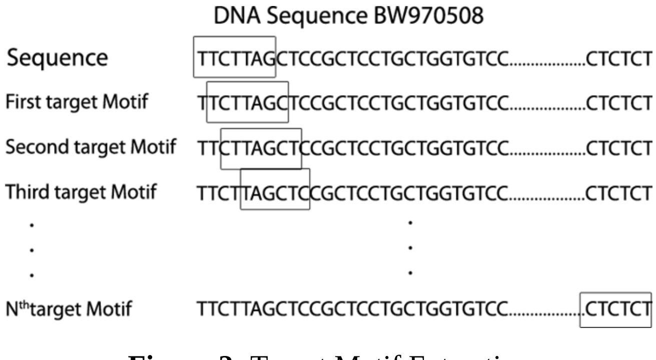

DNA Sequence BW970508
Sequence
TTCTTAGCTCCGCTCCTGCTGGTGTCC..................CTCTCT
First target Motif
TTCTTAGCTCCGCTCCTGCTGGTGTCC..................CTCTCT
Second target Motif
TTCTTAGCTCCGCTCCTGCTGGTGTCC..................CTCTCT
Third target Motif
TTCTTAGCTCCGCTCCTGCTGGTGTCC..................CTCTCT
Nthtarget Motif
TTCTTAGCTCCGCTCCTGCTGGTGTCC..................CTCTCT

Figure 2: Target Motif Extraction

B. Hash table indexing[1]

The next step is to reform the other sequences to their corresponding hash tables. Hash tables are two dimensional array which indexes the positions of A, T, G and C's in each DNA sequence. The usefulness of this is the sequence is now automatically sorted and easier to manipulate.

For example is a DNA sequence is like this

ATCGTCATGCTATGAATGCCATGCGCATT
Then its corresponding hash table will be:

TABLE I
SAMPLE HASH TABLE OF 1ST DNA SEQUENCE

A	1	2	8	13	14	16	17	22
T	3	6	9	12	18	23	29	30
G	5	15	20	21	24	25	26	28
C	4	7	10	11	19	27		

C. Comparison and checking for frequent motifs [1]

Now we need to compare the motifs positions with the hash table entries for each candidate motif and the corresponding hash tables.

Let's say, our reference genome sequence is: **AATCGTAATCCTAAGAATCGGATGGGCGTT**

And 2nd DNA sequence for comparison is: **ATCGTCATGCTATGAATGCCATGCGCATT**

And the hash table for 2nd DNA sequence is thus,

TABLE II
SAMPLE HASH TABLE OF 2ND DNA SEQUENCE

A	1	2	8	13	14	16	17	22
T	3	6	9	12	18	23	29	30
G	5	15	20	21	24	25	26	28
C	4	7	10	11	19	27		

Our target motif is AATCGT and we are going to the 2nd DNA sequence hash table and looking for A's position and we have found it in 1, 2, 8, 13, 14, 16, 17, 22 index position.

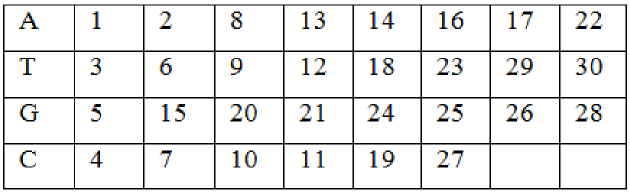
TABLE III
FIRST ITERATION FOR MOTIF MATCHING

A	1	2	8	13	14	16	17	22
T	3	6	9	12	18	23	29	30
G	5	15	20	21	24	25	26	28
C	4	7	10	11	19	27		

Now our second letter, which is also **A**, must be in next of 1st **A** position. And that's why 8 and 22 indices are omitted. So, we will consider only those indices which fulfill the above requirement.

TABLE IV

SECOND ITERATION OF MOTIF MATCHING



A	1	2	8	13	14	16	17	22
T	3	6	9	12	18	23	29	30
G	5	15	20	21	24	25	26	28
C	4	7	10	11	19	27		

Now, our next letter is **T** and its position is 3, 6, 9, 12, 18, 23, 29 and 30. In addition, we need to check whether it's in the next position of the second **A**'s index. And we have found it in the 3rd and 18th position.

TABLE V
THIRD ITERATION OF MOTIF MATCHING

A	1	2	8	13	14	16	17	22
T	3	6	9	12	18	23	29	30
G	5	15	20	21	24	25	26	28
C	4	7	10	11	19	27		

Next we check for positions of **C** that are continuous in the hash table. The positions are 4 and 19.

TABLE VI
FOURTH ITERATION OF MOTIF MATCHING

A	1	2	8	13	14	16	17	22
T	3	6	9	12	18	23	29	30
G	5	15	20	21	24	25	26	28
C	4	7	10	11	19	27		

Next up is **G** in the target motif. And we find that **G** is present at positions 5 and 20.

TABLE VII

FIFTH ITERATION OF MOTIF MATCHING

A	1	2	8	13	14	16	17	22
T	3	6	9	12	18	23	29	30
G	5	15	20	21	24	25	26	28
C	4	7	10	11	19	27		

Now we check for the last position of the motif i.e. the index of **T**. We find that the position 6 fulfills the motif as an exact match but the other one can also be considered as a potential motif with 1 position mutation.

TABLE VIII

SIXTH AND LAST ITERATION OF MOTIF MATCHING

A	1	2	8	13	14	16	17	22
T	3	6	9	12	18	23	29	30
G	5	15	20	21	24	25	26	28
C	4	7	10	11	19	27		

So now we come to a conclusion that the motif in question **AATCGT** is present as exact at the 1st match and as a mutated one in the 2nd match.

TABLE IX

INDICES FOR THE SPECIFIC MOTIF AATCGT

Index	i = 1	i = 2	i = 3	i = 4	i = 5	i = 6
Target motif index	A = 1	A= 2	T=3	C=4	G=5	T=6
Hash table Index (1st occurrence)	A = 1	A= 2	T=3	C=4	G=5	T=6
Hash table index (2nd occurrence)	A= 16	A= 17	T=18	C=19	G=20	T≠21

The same motif has been found in 2 different positions in 2 different forms. Now we can consider that and finally pass these potential motifs for fitness evaluation. B.

D. Fitness function and ranking [7]

In order to formulate a fitness function we need to take into consideration its length, mutation and complexity. In order to fulfil all requirements we propose these new equations:

$$fitness\_score = w1 * L + w2 * r\ (k,m)$$

Here w_1 = weight given to length, L = length of motif, w_2 = weight given to similarity of the matched motif and the function *r (k, m)* calculates the relativity of the motif.

$$r(k,m) = \frac{d(k,m)}{d(k,M_{ref}) + c(k)}$$

This equation simply calculates the ratio of the distance from the whole consensus of the target motif samples i.e. $d(k,m)$ and the distance of the motif from known motifs $d(k,M_{ref})$ with its complexity factor score of $c(k)$.

$$d(k,m) = 1 - 1/k\left[\sum_{i=1}^{k}\sum_{\forall b_i \in x} f(b_i,i) * k(b_i,i)\right]$$

Here, the equation simply finds the hamming distance of the motif from the query sequence and normalizes the value to a range of 0 to 1.

$$c(k) = 4/3\left[1 - \frac{1}{k^2\left[\sum_{\forall b_i \in x}\left(\sum_{i=1}^{k} k(b_i,i)\right)^2\right]}\right]$$

This equation finds the complexity of a target sequence and returns a higher score for a sequence that has lower number of repetitions of bases in it.

1) Example 1

Let's say, K = AATCTC, M = AATGTC and M_{ref} = ACTATC. And $w_1 = w_2 = 1$.

Then $d(k,m)$ = $^5/_6$, $c(k) = 0.72$ and *r (k, m)* = 1.5533

So the fitness_score becomes,

fitness_score = 1*6 + 1*1.5533 = 7.5533

2) Example 2

Let's say, K = TTTTTT, M = AATGTC and $Mref$ = ACTATC. And $w_1 = w_2 = 1$.

Then $d(k,m)$ = $^2/_6$, $c(k) = 0$ and *r (k, m)* = 0.3333

So the fitness_score becomes,

fitness_score = 1*6 + 1*0.3333 = 6.3333

E. Proritazition based on the frequency of small overfrequent patterns overall the sequence

The final step is to prioritize the given sequences based on the reference sequence and finally predict which of the sequences are more prone to similar diseases/cancer.

In order to do that we propose that, if inside a sequence we find the extracted smaller patterns to be over frequent then we can predict that particular specimen to be more prone to similar fate of the reference cancerous cell DNA.

The logic against matching/aligning the whole sequence with the reference genome is that if we try to align the full sequence then it is most likely to show versatile misleading information. Whereas if we can match smaller segments then it is induced that the sequences are matched against gene length and most probable disease causing genes are matched. So it will give us a more credible result giving us the percentage of match with the reference genome sequence.

III. RESULTS AND DISCUSSION

A. Datasets Used

The dataset that was used in this experiment was DNA sequences of humans affected with Alzheimer's disease. The reference sequence with NCBI reference sequence no. NC_000014.9 was the DNA sequence of a practically proven Alzheimer's patient. The other two were potential sequences believed to be affected by the diseases.

TABLE X

DATASETS USED IN EXPERIMENT

Name	NCBI Reference No.	Size
ACOT2 acyl-CoA thioesterase 2	NC_000014.9	9KB
ACOT4 acyl-CoA thioesterase 4	NC_000014.9	5KB
APBB3 amyloid beta (A4) precursor protein-binding, family B, member	NC_000005.10	7KB

B. Experimental Results

The experimental results contain the total fitness function scores against all small frequent terms for each sequence matched against the reference genome. The more the frequent patterns are matched means the more similar the sequences are to the reference one.

TABLE XI

FITNESS VALUES OF MOST FREQUENT PATTERNS

Motif of Seq. No.	Target Motifs	Total Fitness	r(k) value	d(k, m) value	c(k) value
2rd DNA Sequence	AGGCTG	1055.9952	191.9952	96.0048	96.0048
	CTGGAG	1024.8381	166.8381	71.5000	95.3381
	CAGCCT	942.9954	114.9954	23.0046	92.0046
	TTTTTT	870.0000	00.0000	00.0000	00.0000
3rd DNA Sequence	AGGCTG	672.0000	96.0000	31.9968	64.0032
	CTGGAG	724.3298	88.3298	17.6702	70.6702
	CAGCCT	519.3308	63.3308	12.6692	50.6692
	TTTTTT	9566.6200	1166.6200	1166.6200	00.0000

C. Result Analysis

From the experimental results, we can conclude the percentage matching of the sequences with the reference genome. Considering only the given 4 most frequent motifs we can visualize the frequency using a simple graph:

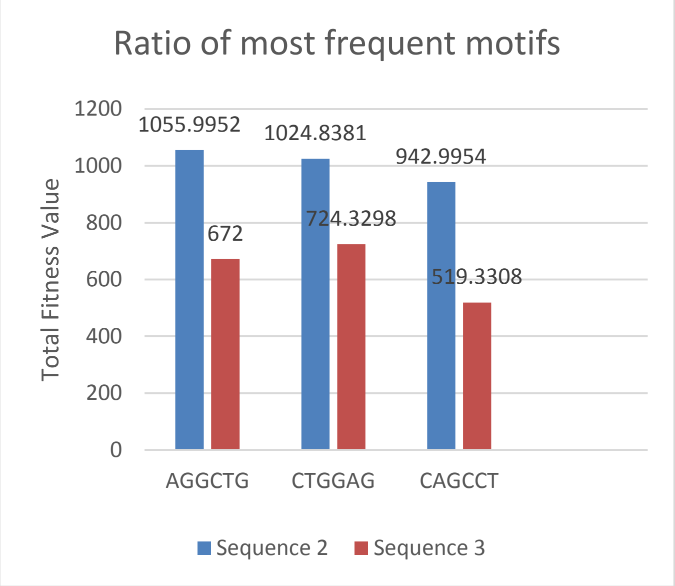


Figure 3: Ratio of most frequent motifs that are present in both sequences.

The last frequent pattern i.e "TTTTTT" has been omitted. The reason behind is that this pattern does not have ant c(k) value confirming that they are more probable stray sequences between actual and meaningful DNA genes.

From the above analysis it is clear that sequence 2 is more similar to the reference sequence than the 3rd sequence. So our analysis predicts that the 2nd DNA sequence is more prone to any disease that the 1st reference DNA sequence suffered from.

IV. CONCLUSION

This research paper focuses on finding similarity between cancerous reference genome and candidate DNA sequences based on motif matching. The main drawback of this work remains that we can only use fixed length motifs to match. But the actual fact is motifs can be of variable lengths and considering those lengths and also mutation the actual result may deviate slightly. This aspect of work remains as our future set of task to modify the system and include all probable anomalies in case of matching motifs.

V. REFERENCES

[1] A. F. Ashrafi, A. K. M. I. Newaz, R. Ajwad, M. M. Tanvee and M. A. Mottalib, "A modified algorithm for DNA motif finding and ranking considering variable length motif and mutation," Recent Trends in Information Systems (ReTIS), 2015 IEEE 2nd International Conference on, Kolkata, 2015, pp. 12-17. doi: 10.1109/ReTIS.2015.7232844, URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7232844&isnumber=7232836

[2] Kazi Mahbub Mutakabbir, S. S. Mahin and Md. Abid Hasan, "Mining frequent pattern within a genetic sequence using unique pattern indexing and mapping techniques," Informatics, Electronics & Vision (ICIEV), 2014 International Conference on, Dhaka, 2014, pp. 1-5. doi: 10.1109/ICIEV.2014.6850729, URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6850729&isnumber=6850678

[3] Sharifa, L., S., A., Harun, H., and Taib, M., N.: A Modified Algorithm for Species Specific Motif Discovery. In International Conference on Science and Social Research (CSSR 2010), Kuala Lumpur, Malaysia, Dec 5-7, 2010.

[4] Sharifa, L., S., A. and Harun, H.: Motif Discovery using Linear-PSO with binary Search. In AWERProcedia Information Technology & Computer Science. Pp 458 – 462. (2012)

[5] Hong Zhou; Zheng Zhao; Hongpo Wang, "A novel parallel motif discovery algorithm based on de Bruijn graph," Industrial Mechatronics and Automation (ICIMA), 2010 2nd International Conference on , vol.2, no.,pp.139,142,30-31May2010 doi: 10.1109/ICINDMA.2010.5538350.

[6] Islam,S.M.S.; Asger, M.R. ; Hasan, M.A. ; Mottalib M.A. : A modified algorithm for variable length DNA motif discovery, Smart Instrumentation, Measurement and Applications (ICSIMA), 2013 IEEE International Conference on 25-27 Nov. 2013, Pages 1-4.

[7] Dianhui Wang, SarwarTapan. : MISCORE: a new scoring function for characterizing DNA regulatory motifs in promoter sequences.From 23rd International Conference on Genome Informatics (GIW 2012) Tainan, Taiwan. 12-14 December 2012.

[8] Chang, B., C., H., Ratnaweera, A., and Halagmuge, S., K.,: Particle Swarm Optimization for Protein Motif Discovery. In Genetic Programming and Evolvable Machine, vol. 5, pp. 203-214. (2004)

[9] Akbari, R., and Ziarati, K.,: An Efficient PSO Algorithm for Motif Discovery in DNA. In IEEE International Conference of Emerging Trends in Computing, Tamil Nadu, India 2009.

[10] Hardin, C., T., and Rouchka, E., C.: DNA Motif Detection Using Particle Swarm Optimization and Expectation-Maximization. In IEEE Symposium on Swarm Intelligence, 2005.

[11] Zhou, W., Zhu, H., Liu, G., Huang, Y., Wang, Y., Han, D., and Zhou C.: A Novel Computational Based Method for Discovery of Sequence Motifs from Coexpressed Genes. In International Journal of Information Technology, vol. 11 (2005).

[12] Lei, C., and Ruan, J.: A Particle Swarm Optimization Algorithm for Finding DNA Sequence. IEEE International Conference on Bioinformatics and Biomedicine, Philadelphia, 2008.

[13] Davila, J., Balla, S., Rajasekaran, S.: Fast and practical Algorithm for Panted (l, d) Motif Search. In: IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 4, pp. 544-552, IEEE Press (2007)

[14] Pradhan, M.: Motif Discovery in Biological Sequences. Master’s Projects (2008)

[15] CompariMotif: quick and easy comparisons of sequence motifs Richard J. Edwards1,2,*, Norman E. Davey1 and Denis C. Shields. Received on February 11, 2008; revised on

March 18,2008; accepted on March 19, 2008.Advance Access publication March 28, 2008.

[16] Kennedy, J., and Ebehart, R.: Particle Swarm Optimization. In: IEEE International Conference on Neural Networks, Perth, Australia (1995).

[17] ShripalVijayvargiya, PratyooshShukla.: A Structured Evolutionary Algorithm for Identification of Transcription Factor Binding Sites in Unaligned DNA Sequences. International Journal of Advancements in Technology. http://ijict.org/ ISSN 0976-4860

[18] Matt Stine, DipankurDashgupta,SurajMukatira. : Motif Discovery in Upstream Sequences of Coordinately Expressed genes. sequences.From 20rd International Conference on Genome Informatics (GIW 2011) Tainan, Taiwan. 11-13 December 2011.

[19] S. Bai, S. X. Bai, “The Maximal Frequent Pattern Mining of DNA Sequence,” GrC, pp 23-26, 2009.

[20] S. F. Zerin, B. S. Jeong, “A Fast Contiguous Sequential Pattern Mining Technique in DNA Data Sequences Using Position Information,” Department of Computer Engineering, Kyung Hee University, 1 Seocheon-dong, Giheung-gu, Yongin-si, Gyeonggi-do, 446-701, Korea. Date of Web Publication 12-Dec-2011.

[21] M. M. Tanvee, S. J. Kabeer, T. M. Chowdhury, “Mining Maximal Adjacent Frequent Patterns from DNA Sequences using Location Information,” Department of CSE, Islamic University Of Technology.

[22] T. H. Kang, J. S. Yoo and H. Y. Kim, “Mining frequent contiguous sequence patterns in biological sequences,” in proceeding of the 7th IEEE International Conference on Bioinformatics and Bioengineering, pp. 723-8, 2007.

[23] R. Agrawal and R. Srikant, “Fast algorithms for mining association rules.” In Proc. 1994 Int. Conf. Very Large Databases (VLDB?94), pages 487–499, Santiago, Chile, Sept. 1994.

[24] R. Srikant and R. Agrwal, "Mining sequential patterns: generalizations and performance improvements", in Proceedings of 5th International Conference on Extending Database Technology (EDBT'96), Avignon, France, pp. 3-17, Mar. 1996.