NOVEL ALGORITHM FOR DETECTION OF MOTIFS IN SEQUENCE DATA SETS

Venu Madhav Kuthadi[*]

Rajalakshmi Selvaraj[**]

Abstract---

The basis of bioinformatics is the extraction of motifs from the sequences. Pattern emerging continuously either over a string or inside the same string are the significant objects for identifying. Emerging continuous patterns are called as motifs and their detection is referred as motif extraction or motif interference. The most significant problem in biology is the motif search. This issue normally needs a voluminous data for detecting the short patterns of interest. Basically, we look at the issue of mining structured motifs which can allow the variable length gaps between simple motif components. Here in this research paper, we present a novel algorithm which is mainly used to detect the continuous pattern with a diversity of definitions of motif model and it is referred as Pattern Shift MatchingAlgorithm.Thus our proposed algorithm performance is analyzed by using both the synthetic and real dataset and compared with the existing techniques.

***Keywords*---sequence mining, motifs**.

[*] Department of AIS, University of Johannesburg, South Africa

[**] Department of CS, BIUST, Palaype, Botswana

1.INTRODUCTION

One of the serious features of interpretation is to take out the important patterns from the sequence datasets. The most important task in bioinformatics is to analyze and interpreting the sequence data. There are two challenges occurs while extracting the motif first challenge is for enumerating the frequent motifs one is to design a flexible algorithm secondly Statistically legalize the motif that are extracted and report the important pattern.

In general, motifs[1][4] are basically classified in to two categories as simple motifs and structured motifs. In the motif, if there is no variable gaps are allowed then it is referred to as a single motif and whereas if any variable gaps are allowed then it is referred to as the structured motifs. A structured motif can be stated as a planned set of simple motifs with gap limitation among each pair of adjacent simple motifs.

In most application of the sequential data mining, the purpose is to detect the continuously occurring patterns. For detecting the matching process[5][8][9] initially a set of noise patterns are allowed. At the heart of such method, is the definition of pattern and similarity among a pair of patterns. This definition of similarity will vary from one application to the other. In computational biology, the estimated subsequence mining issue is the significant one and these confront is to detect the short sequence pattern basically of length 6 – 15 which occur regularly in a given set of protein sequence or DNA. This short sequence pattern can offer evidence regard the location called as regulatory region. These repeated happening of the short sequence dataset will not be identical always and a few of them differs from other. A complex similarity metrics should be utilized in order to find the distances. In computational biology, the frequently occurring patterns are referred to as motifs[2][3]. The motif mining algorithm wants to be able to deal with a variety of notions of similarity. The problem of finding patterns from large databases has been studied[6][7][8][9].

For satisfying our model we have also presented a new novel motif mining algorithm called as PSM (Pattern Shift Matching). It is noted that the problem of pattern mining is related to the problem of frequent sub sequences and the frequent item sets. Conventionally, let us assume Q is a sub sequence of P, if Q can be built by prognostic out few of the elements from sequence P. elements of sequence P is “TCAGCCAACG” and its sub sequence is constructed by choosing only the selective elements from the sequence P and the sub sequence Q is formed. This paper highlights only the continuous sub sequences mining issues. It is motivated by the issue of detecting the frequent motifs in DNA sequences which has philosophical significance in the computational biology community and life sciences that has created number of algorithms for finding the frequent patterns using the hamming distance[16][17] notion of similarity. Some of the example algorithms in this category are MITRA, YMF, Random projections[14] and Weeder[11][12][13].

On comparing the entire algorithm PSM is more powerful and flexible one. Here the PSM model is compared with the Weeder model in order to show the scalability and faster magnitude of both the algorithms. Further more, there are various applications are available for motif mining. Foremost thing is the detection of association rules in the sequence data and also used to discover best seeds for clustering the sequence data sets. From patients, records of medical signals such as respiratory data or ECG are mined to detect the signals which are mainly used to represent the possible dangerous conditions. The chief part of the gene rule is to arbitrate through exact proteins known as transcription factors which manipulate the transcription of a specific gene by binding to exact sites on DNA sequences which are referred to as transcription factor binding sites.

It is noticed that the binding sites frequently materialize as a mixture of one or more simple motifs divided by variable length spaces particularly in eukaryotic organisms. The process of mining the combination of patterns is known as structured motifs which is the challenging issue. PSM is developed efficiently in order to mine the combination of simple motif with out the need of any additional information about distance among the simple models.

II MODEL

One of the major issues in pattern mining is the model defining in which various numbers of sequence patterns are considered and used to compute the matching. Constructing such model create an interesting challenge. In fact the model should be enough to find the happening of motif even in the presence of noise and also it must not be in general which matches not related subsequences. Here we construct a new novel model for patterns which is used for mining issue in multiple domains. Assume that the input sequence is made of symbols from a distinct alphabet set. Though our input sequence is made of symbols it can also be applied for continuous time series datasets. This is done by changing those datasets into a symbolic sequence data by one of the judgment data of the numeric data which is often carried out by using the mining process of continuous time series datasets. The motif model can be called as (L, M, S, K) model.

The four parameters that should be determined are

L – Length of the motif,

M – Distance matrix for computing the similarity among two strings,

S – The maximum threshold in which the strings are taken and

K – Minimum support needed for a pattern to become a motif.

The (L, M, S, K) model is more spontaneous and authorize the user to achieve a more flexibility in doing the right tradeoff among noise tolerance and specificity model. As here the parameter M is used hence it solves for a complex datasets also. Both FLAME as well as PSM uses the same model for computation. For the FLAME computations refer paper [1].

Let us assume T which has [1…n] of length n and pattern P [1…m] of length m. From a finite alphabet set Σ the elements of T and P are drawn. The strings of characters are nothing but the character arrays of T and P. Pattern P is said to be occur with shift s in text T

If 0<= s<=n-m and

T[s+1…s+m] =P [1…m]

Hence these shifts are generally referred to as a valid shift.

To determine the entire valid shift or the possible values of S so that

T[s+1…s+m] =P [1…m];

There are n-m+1 possible values of s.

IIIRESULT & DISCUSSION

In the simulation, we examine a input dataset by applying both the FLAME algorithm as well as proposed PSM algorithm in order to show that the newly proposed PSM algorithm is more fast efficient and accurate in determining the matching motifs in the sequence datasets. Simulation is carried out in java. The following steps are followed and the results are obtained.

Step 1: The first input of the DNA set is uploaded (i.e., say the father data set)

Step 2: (L, M, S, K) motifs are extracted

Step 3: suffix tree construction

Step 4: Pruning process

Step 5: New DNA Extraction i.e., the second input (it may be the input dataset of son)

Step 6: Step 2 to step 4 is followed again

Step 7: Now the matching is calculated using FLAME algorithm and the corresponding percentage of matching is shown

Step 8: Now to the output of the pruning technique for the second input dataset the pattern shift matching is done and then the matching percentage is computed.

Result: For the same set of input data, on applying FLAME and PSM algorithm it is proven that the result percentage of PSM is obviously high when compared to FLAME.

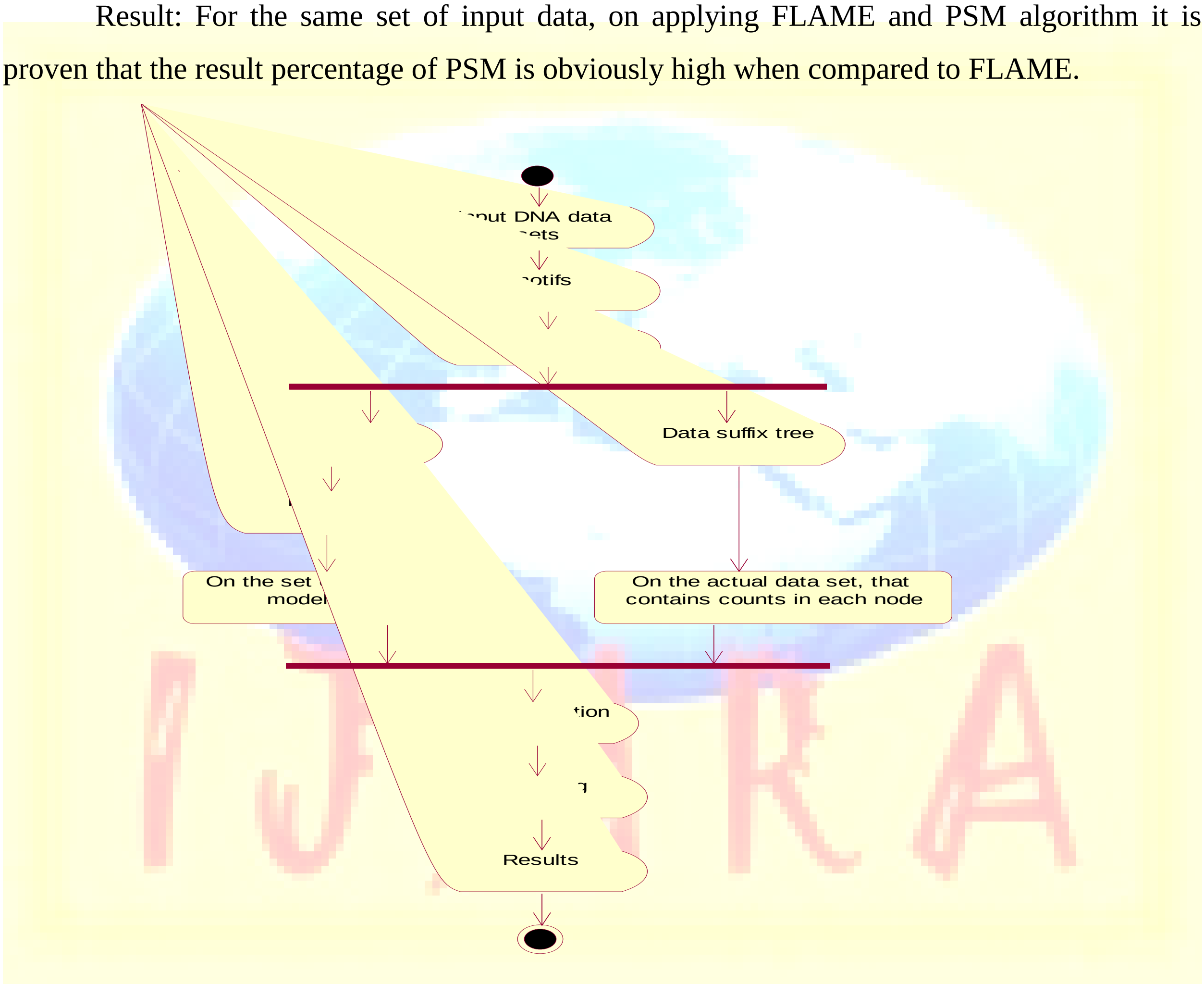


Fig.1: shows the general steps for extracting the DNA dataset

For example, consider the following calculation which illustrates you how PSM is efficient than Flame.

Let us assume two input datasets of a father and son. Dataset of father is “TCAGCCAACG” and for son it is “ATGTCCATAC”. The input we illustrate here is a sample not the original datasets. Both the father and son will be have the same string but not in the same order. As per above mentioned steps we can prove,

Step 1: The first input of the DNA set is uploaded (i.e., say the father data set)

“TCAGCCAACG”

Step 2: (L, M, S, K) motifs are extracted.

Length of the dataset = 10

Data = T, C, A, G

Length of the motifs (L) 4

Character = T distance (M)0

Character = Cdistance (M)1

Character = C distance (M)2

Character = C distance (M)0

Character = C distance (M) 2

. . . . . .

.. . .

Character = G distance (M)5

Character => T Maximum distance (S) = 0

Character => C Maximum distance (S) = 2

Character => A Maximum distance (S) =2

Character => G Maximum distance (S) = 4

Character => T total count (K) =1

Character => C total count (K) =4

Character => A total count (K) =3

Character => G total count (K) =2

Now, L, M, S, K is calculated like this.

Step 3: Data Suffix tree formation

T C A G

Step 4: Pruning

T -------→ CAGCCAACG

C -------→ AGCCAACG

→CAACG

→AACG

→G

A-------→GCCAACG

→ACG

→CG

G-------→CCAACG

→ Null

Step 5: new DNA set is extracted i.e) the son data set

“ATGTCCATAC”

As mentioned above, step 2, 3, 4 are carried out and the result is computed. The result of son after pruning will be like this

A-------→TGTCCATAC

→TAC

→C

T-------→GTCCATAC

→CCATAC

→AC

G-------→TCCATAC

C-------→CATAC

→ATAC

→ Null

Step 6: now, matching is done as per FLAME algorithm. i.e) the output of pruning technique from both the father and son is taken and compared, and then finally the output percentage is displayed.

According to FLAME, matching percentage is “72%”.

Step 7: To prove our proposed technique we take the output of the pruning process of the son and shift them till the same sequence occurs for the next time. For example,

A-------→TGTCCATAC

Shifting is carried out and the result will be

CTGTCCATA

ACTGTCCAT

TACTGTCCA

ATACTGTCC

CATACTGTC

CCATACTGT

TCCATACTG

GTCCATACT

Now this will be taken as a input for the son and it is compared with the output of the father and the percentage of matching is noted as ”80”.

From the above example, it is clear that the PSM algorithm is more efficient, accurate and faster to detect the motif when compared to the FLAME algorithm.

Likewise, we took a data from the “TRANSFAC, http://www.gene-regulation.com/pub/ databases.html” and examined the result. The input and the output are shown below; here we use the original dataset of a father and son in order to check our efficiency:

<u>Input1:</u>

TCAGCCAACGGGTACACCTCATCCTCGACGCTAAAAAATTCCCCATCACAA
AATCTACATCGATTTGATTGACAAGATGTCCATACCGCAAAAACACTTTAAGAAAGG
CTCCACCAAACCGCAGCTGCCCGACGACGGGGTATTGCGAGTCTACTCGATGAGATT
CTGTCCCTCCCGAATGGTACAAGGATTACAGTCCCCTAGGAAAGGTACCCGCCTTGC
AGCTGACGGATGTAAAAGACCAGCCAACACTTGTGGAATCGATGATCATAGCCGAA
TTTCTCGATGAGCAGTATCCCGAGTTGCGGCTCTTTCCCAGT

<u>Input2:</u>

ATGTCCATACCGCAAAAACACTTTAAGAAAGGCTCCACCAAACCGCAGCTGC
CCGACGACGGGGTATTGCGAGTCTACTCGATGAGATTCTGTCCCTTCAGCCAACGGG
TACACCTCATCCTCGACGCTAAAAAATTCCCCATCACAAAATCTACATCGATTTGA
TTGACAAGCCCGAATGGTACAAGGATTACAGTCCCCTAGGAAAGGTACCCGCCTTG

CAGCTGACGGATGTAAAAGACCAGCCAACACTTGTGGAATCGATGATCATAGCCGA ATTTCTCGATGAGCAGTATCCCGAGTTGCGGCTCTTTCCCAGTGA

Result:

According to FLAME the output percentage is only 72% but it more when using PSM and it is nearly 80%.

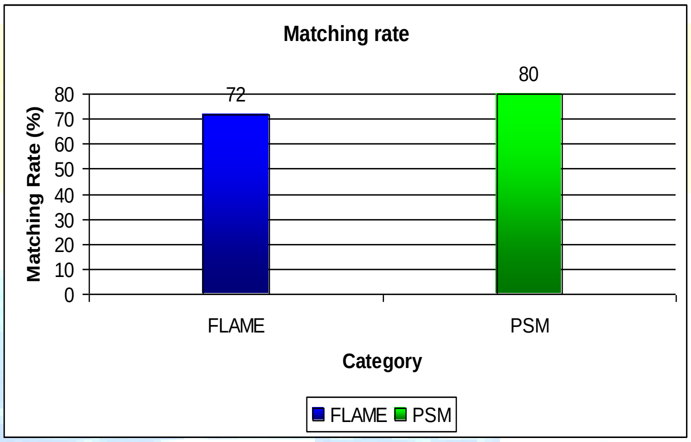


Fig.2: Result of FLAME verses PSM

Here the result from one experiment is analyzed for testing the performance and the evaluation of the PSM algorithm and also compared the PSM with pattern mining algorithm of various application domains. Many of the algorithms which are existing nowadays operates with the parameter (L, D, K) motifs and those algorithm does not maintain the more general model (L, M, S, K). Hence here we compared the PSM with the one of the existing algorithm which has only the parameter (L, D and K). There are plenty of different data sets are available for the experiments here we examine a data set of a snake and the result are computed.

Data set of Snake

This is the protein data set which was considered for mining sub sequences. The data sets of snake consist of 352 various venom protein sequences of snakes in a varying length. The data set consist of a size of about 28000 symbols. 20 is the size of alphabet of amino acids. These protein data sets are habitually investigated for detecting the similar patterns which offer imminent in to their function.

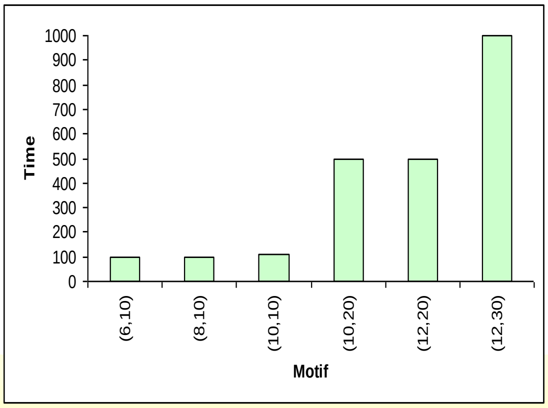
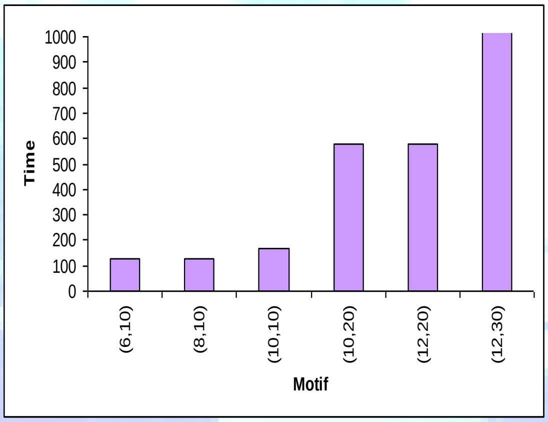


Fig .3 motifs using flame.



4a 4b

Fig. 4: motifs using PSM

Comparing PSM with Weeder

The algorithm of weeder is a fast discovering which is designed particularly to detect the motifs that are present in the DNA data sets. Weeder algorithm is restricted to the (L, D, K) model and it is not suited to operate in the (L, M, S, K) model. This algorithm is enormously fast as it considers that the mismatches are spread homogeneously diagonally to the motif length. Thus the outcome of his method can violently reduce the space of search fast but it is not sure to be exact.

For determining the exactness of Weeder algorithm a simple experiment is performed. Here consider the data set of snake which is depending on the real motif determined in the database. By using different models both the algorithms on the data sets are analyzed. The percentage of motifs present from the analysis is presented in the figure but FLAME as well as PSM finds all the motifs present in the data set. The Weeder algorithm presents the output in a faster way but it fails to collect all the motifs present in the data sets. In case of a (12, 2) motif, weeder determines only the 5% of the total amount of motif determined by the PSM but the time for computing is more for weeder that is it takes only one second to determine the motif where as the PSM takes nearly 40 seconds. For exhibiting the efficiency of the PSM in determining the real biological motifs that are not found by the weeder, an experiment is conducted. The candidate motifs listed from the PSM algorithm in the data set of snake and then a ranking is done. As a result, it is observed that the PSM is capable of determining exactly all the motifs present in the dataset which are missed by the Weeder algorithm.

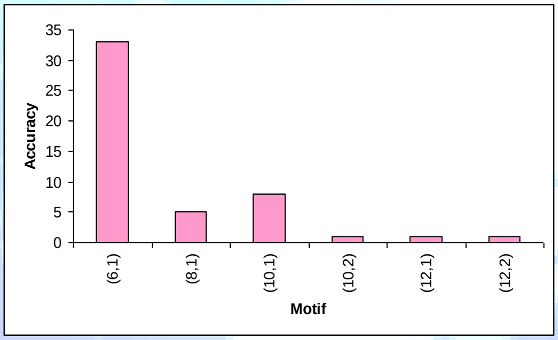
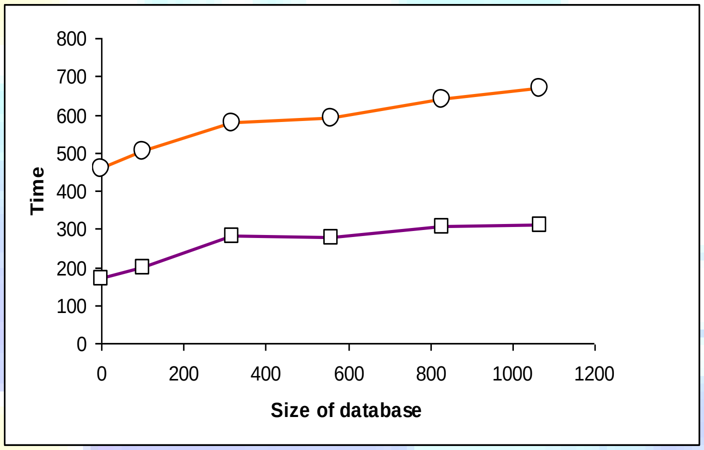


Fig. 5: Shows the accuracy of dataset

Scalability

One of the critical tasks is Motif mining and the present available algorithms looks in to small data sets only (nearly 10000 symbols). But PSM is capable to balance as much large size of database. By generating the synthetic data set and entrench a motif of length selected arbitrarily among 8 and 14 in 10% of sequences. Length of 1000 sequences is placed in the datasets and the sequence counting is raised steadily to make database of increasing size. The entire database is varied between 20000 symbols to 1 million symbols. The PSM is across the data sets to detect (8 – 14, 1) and (8 – 14, 2) models by 10% support. The result is displayed in the below figure. If the size of the database is increased then obviously the execution time also increases. In the case of (8 – 14, 1, 10%) motifs the time raises from 7 to 55 secs whereas in the case of (8 - 14, 2, 10%) motifs the time raises from 290 to 5900 secs. The time for mining the more complicated motifs are increased more because of the number of models taken before pruning started in the model tree is exponential in d. the running is related to the total number of candidate motifs taken and the number raises exponentially with d. for instance, the database size may be of 20000 symbols. In the case of (8 – 14, 1, 10%) the number of model before pruning is 13240 and the number of candidate motifs are 1323 whereas in the case of (8 – 14, 2, 10%) the number of model discovered is 200620 and the number of candidate motifs are 20061. though the pattern which has a length of 14 can be mined in certain time. To my knowledge, no other present algorithms exactly scale those large sizes of databases.

Fig. 6: Shows the scalability



IVCONCLUSION

In this paper a significant new model for motif mining in sequence database is presented. This model considers various existing models and present extra flexibility which makes the model valid in a large diversity of data mining applications. In addition an algorithm named PSM is also developed in order to attain accuracy and flexibility for detecting the motifs. By conducting a sequence of experiments in synthetic and real date sets, it is concluded that the PSM algorithm is an adaptable one which is mainly used in various real motif mining tasks. PSM algorithm is better when compared to other algorithm used in the computational biology for detecting the motifs. It is also proven that the algorithm can scale to hold the larger size datasets than the other algorithms. At last, the PSM algorithm is considered to be a best one for detecting the motif mining.

REFERENCES

[1] P. Patel, E. Keogh, J. Lin, and S. Lonardi, "*Mining Motifs in Massive Time Series Databases*," in *ICDM*, 2002, pp. 370–377.

[2] AvriliaFloratou, Sandeep Tata, and Jignesh M. Patel, "*Efficient and Accurate Discovery of Patternsin Sequence Data Sets Member"*, IEEE Transactions on knowledge and data engineering,vol 23,no.8,pp 1154-1168,2011.

[3] Kuo-Ching Liang, Xiaodong Wang and Dimitris A *" A sequential Monte carlo method for motif discovery*",IEEE transactions on signal processing, vol.56, no.9, pp 4496-4507,2008.

[4] M. J. Zaki, "SPADE*: An Efficient Algorithm for Mining Frequent Sequences,*" Machine Learning, vol. 42, no. 1/2, pp. 31–60, 2001.

[5] X. Yan, J. Han, and R. Afshar, "*CloSpan: Mining Closed Sequential Patterns in Large Datasets,"* in SDM, 2003.

[6] J. Wang and J. Han, "*BIDE: Efficient Mining of Frequent Closed Sequences,"* in ICDE, 2004, pp. 79–90

[7] J. Yang, W. Wang, P. S. Yu, and J. Han, "*Mining Long Sequential Patterns in a Noisy Environment*," in SIGMOD, 2002, pp. 406–417.

[8] H. Wu, B. Salzberg, G. C. Sharp, S. B. Jiang, H. Shirato, and D. Kaeli,"*Subsequence Matching on Structured Time Series Data,"* in *SIGMOD*,2005, pp. 682–693.

[9] B. Y.-C. Chiu, E. J. Keogh, and S. Lonardi, "Probabilistic Discovery of Time Series Motifs," in *KDD*, 2003, pp. 493–498.

[10] I. Jonassen, J. F. Collins, and D. G. Higgins, "*Finding Flexible Patterns in Unaligned Protein Sequences,"* Protein Science, vol. 4, no. 8, pp.1587–1595, 1995.

[11] G. Pavesi, P. Mereghetti, G. Mauri, and G. Pesole, "*Weeder Web: Discovery of Transcription Factor Binding Sites in a Set of Sequences From Co-Regulated Genes,"Nucleic Acids Research*, vol. 32(Web Serverissue), pp. W199–W203, 2004.

[12] L. Chen, M. Tamer Ozsu, and V. Oria, "*Robust and Fast Similarity Search for Moving Object Trajectories*," in SIGMOD, 2005, pp. 491–502.

[13] A. Udechukwu, K. Barker, and R. Alhajj, "*Discovering all frequent trends in time series,"* in Proc. of Winter Int. Sym. on Information and Comm. Tech., vol. 58, 2004, pp. 1–6.

[14] M. Tompa et al., “*Assessing Computational Tools for the Discovery of Transcription Factor Binding Sites*,”Nature Biotechnology, vol. 23, pp.137–144, 2005.

[15] M. Vlachos, G. Kollios, and D. Gunopulos, “*Discovering Similar Multidimensional Trajectories*,” in ICDE, 2002, pp. 673–684

[16] A. W.-C. Fu, E. J. Keogh, L. Y. H. Lau, and C. A. Ratanamahatana,“*Scaling and Time Warping in Time Series Querying*,” in VLDB, 2005,pp. 649–660.

[17] M. J. Zaki, “*Sequence Mining in Categorical Domains: Incorporating Constrains*,” in CIKM, 2000, pp. 442–429.