An Optimized Framework for Precise Motif Discovery by Merging SBF with Pruning Method

R Kumar[1], Capt. Dr. S Santhosh Baboo[2]

[1]*Research Scholar, M.S University, Tamil Nadu, India*
[2]*Department of Computer Science, DG Vaishnav College, Chennai, Tamil Nadu, India*

***Abstract:*-Motif mining became more popular and got more attention towards data mining field due to its real world applications such as health prediction, locating previous patterns in time series database. Motifs are the most correlated pair of subsequences in sequence objects. Motif discovery is hard on emerging applications which have long sequences or applications where queries arrive rapidly. Since correlation computations and prune subsequence techniques requires different ordering on examining subsequence pairs, Existing works cannot bring faster computation of correlations and prune subsequence pairs at the same time. In this paper we propose a new framework called F-Motif (Fast-Motif) which comprises two level approaches for pruning subsequence at outer level and fast correlation computation at the inner level. In our Experimental results, our framework performed 3X faster than existing methods.**

***Keywords* - Motif Discovery, Motif mining, Smart Brute Force, Reference Indices.**

I. INTRODUCTION

Motif discovery has large set utilities on data mining technology which can be used for rules discovery, classifications mining, clustering datasets and sequence summarization. Motifs are the most correlated pair of subsequences in sequence objects. Correlations between two subsequences are measured by correlation metrics. Below figure shows the motif that discovered from power consumption dataset.

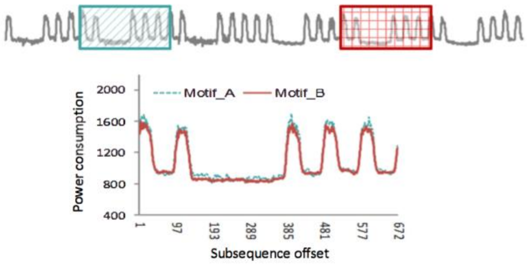


Fig 1: Subsequence offset

Motif Discovery problem has got attention from data mining communities [1-5]. Motif discovery is the backbone of human and animal's activity discovery and also useful with surveillance and sports training [3] etc. Enumerated clustering of motifs is more meaning full than clustering all subsequences in large sequence of datasets [7].

Motif discovery is a time consuming problem for Sequence of object Length **m** contains (**m- l + 1)** subsequences of length **l.** Brute force method will compute all pair of sequences and computes correlation of each pair. This method takes more time which is hard for high length of sequences. Most of the existing algorithms concentrates on fast motif discovery [1],[2],[8] but lacks in accuracy and they not provide guaranteed accurate result.

In Exact discovery of time series motifs [5] paper proposes two better algorithms for motif discovery. The Smart brute force(SBF) [9] method only proposes examining subsequence pairs in a specific order for finding correlation between each pair incrementally in a constant time series. Even though this method examines correlation of subsequence pairs but lacks on pruning those subsequent pairs. In reference indices method(MK) [5] calculates subsequence pair by their order of correlation to their reference and employs pruning to remove inauspicious. Then calculates correlation for remaining pairs. However, it's pruning effect only depends on data distribution. In some cases, the number of remaining pairs may be longer than expected and degrades to the brute force method. In our observation, it is studied that comparing to brute force method MK method calculates exact distances for only 0.013% of all pairs in datasets of ECG results [10]. And also MK only requires reasonable memory for storing all normalized subsequences.

Unfortunately, both MK and SBF methods cannot be readily combined since they both depends on different orderings on examining subsequence pairs. By our literature study we find that there is no previous works are performed on applying both techniques to find motif.

In this paper We propose a brand new two level technique that enables both MK and SBF techniques together for motif discovery. Our idea is to group all subsequence pairs by their offset locality and assigning every successive subsequence to the same group level. In our first level (Outer level) we first examines all pairs in each groups and remove all inauspicious pairs from each group. In second level (inner level), we compute correlation between all subsequence pairs within the group of each pairs. Then we propose two optimization technique for both levels : (i) Proximity based strategy for finding motif quickly and (ii) Group refinement technique for sharing processing cost among surviving groups.

Our proposed method FM (Fast Motif) achieves to get lower time complexity than MK method. In experimental results our proposed solution gives 3X faster results than the existing solutions. For example, FM takes 0.88s to discover motif with length of 900 in TAO dataset where MK and SBF method needs 768s and 1267s respectively. This spectacular performance change makes motif mining in long sequence possible on online processing. Our method will Perform well in both multicore and distributed systems.

Remaining sections of papers is organized as follows. Section II explains about motif discovery and existing solutions. Proposed model described in Section III. Experimental results are discussed in section IV. Conclusion and future work is discussed in section V.

II.PRELIMINARIES

In this section we will briefly discuss about motif discovery problem and existing solutions for this problem.

A. Motif Discovery

Let us assume notation for sequence object as S and its subsequence as follows. These notations are used throughout this works.

Definition 1(subsequence of s): Let us assume length of the sequence object S as M (i.e S=S[0]….S[M-1]).The valid length of the subsequence of length L in S is denoted as S_i=S[i]…..S[i+L-1] where i is the offset of subsequence and 0 <= i < M-L+1.

By Definition 1 it is found that a motif of length L in the sequent object S is the most correlated subsequence pair$(S_i, S_j)^3$. Here we use normalized Euclidean distance as the distance measure which is commonly used in previous Works. There are other distance measures also present but they all are inefficient of normalization renders which makes them unsuitable for subsequences with different scale and offsets [3],[11],[12]. To produce significant results we omit subsequence pair with overlapping as they trivially matched.

Motif Discovery: Motif Discovery is the process of finding pair of subsequences for a given sequence object S and targeted Motif Length L, Where the normalized Euclidean distance of discovered motif should be minimum among all other non-trivial subsequence pairs. Generally Normalized Euclidean distance of Subsequent pairs are calculated by finding Euclidean distance of normalized pairs of them.

B. Related Work

Time series motif was discovered by by Chiu etat [3] which became core subroutines for modern sequence mining techniques [3] [7].The naive solution requires quadratic number of distance calculations which becomes infeasible for large scale data in modern applications.

Castro and Azevedo [1] proposes a solution that converts the sequences into set of symbols by iSax[10].By this we can reduce the dimensionality problem in motif discovery and can compute approximate motif efficiently. Even though this solution doesn't promises the quality of the results. Tao et al.[24] solution proposes a locality based B-Tree to get closest pair query approximately for high dimensional objects. Even though it is one of the best state of the art solution, it doesn't guarantee the exact motif discovery.

C. Brute Force:

Brute Force Algorithm compares all possible subsequence pairs using standard Distance calculation. Algorithm 1 shows pseudo code of BF. Naive approaches used for efficient calculation which normalizes all subsequence pair and store it on memory to prevent multiple time normalization of same subsequence. During execution of algorithm it updates best so-far distance bfs when motif is discovered.

In algorithm 1 brute force technique is defined. Here we takes sequence s of length M and motif length of L as input to the Algorithm to get to get motif offer and best so far as output of the algorithm.

ALGORITHM 1(BRUTE FORCE)

Input: Sequence S of length M, Motif length L

Output: Motif Offset OS and motif distance bsf

```
Normalize all subsequences with L in all
sequence S
BSF <— infinite
For i <— 0 to M-L do
    a. For j <— i+L to M-L do
    b. if(d(subsequence ($S_i$,$S_j$))) < bsf
       then
        i. bsf <— d(subsequence
           ($S_i$,$S_j$)); OS <— (i,j)
```

A. *Smart Brute Force(SBF):*

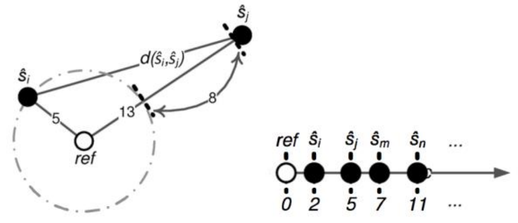
This method is used to calculate distance of subsequence pairs in a specific order ((Si,Sj),(Si+1,Sj+2),…) which reduces the time complexity from offset of motif length O(L) to offer of 1 O(1). Running sum technique is used here. But Lacks in pruning

B. *Reference Indices(MK):*

We can able to derive the distance lower bound between any subsequence pair using given set of subsequences and their distance to reference subsequence.If we have two subsequences and their reference to the distance is ref then the lower bound can be calculated by

$$distance(ref\_Subsequence_1) - distance(ref\_Subsequence_2))$$

If bsf is smaller than the lower bound then the subsequence is removed safely. If we derive lower bound from more reference indices then the lower bind will come more tighter



(a) pruning by reference indices, ℓ (b) execution strategy

Fig 2: Two types of techniques in Reference Indices Method

III. PROPOSED MODEL

From literature study we found that all prior works are focused with narrow aspect which to boost query performance. For example if we take MK technique ,it exploits pruning capability by using reference indices but it takes more time complexity for each distance calculation. SBF supports fast distance calculation method but lacks in pruning unpromising subsequences. None of this method can be applicable for emerging applications with long sequences and application with frequent motif access. Our Proposed framework works with SBF technique and adding batch pruning technique with it.

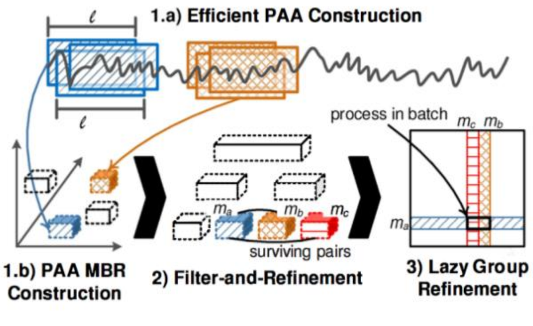


Fig 3: System Architecture

Workflow of our framework is explained in system architecture model Diagram. We transform each subsequence into piecewise aggregate approximation (PAA) representation to reduce dimensionality problem. To reduce distance calculation time we group consecutive PAA into PAA MBR (Minimum bound rectangle) representation. So that pairwise distance of two MBR can be computed in single offset timing. For batch pruning method we use construct MBR into Hilbert R-Tree representation. Then we apply refinement work on it to remove unpromising groups.

A. PAA MBR Construction

Piecewise aggregate approximation Construction:

PAA is the process of reducing logical dimensionality which is more efficient than other dimensionality reduction methods such as SVD ,DFT and DWT [16].More specifically normalized subsequence can be transformed into PAA segments of equal lengths.

Each subsequence can be transformed into PAA by converting subsequence into normalized subsequence. So that each PAA construction needs offset of length O (L) time to transform subsequences into PAA segments. By extending running sum technique we can construct PAA representation which takes only offset of dimensionality timing.

2. PAA MBR Construction:

For batch pruning technique we group the PAA representation of subsequences into minimum bound rectangles (MBRs).MBRs can be calculated only if their minimum distance is smaller than the best so far distance. While grouping PAA representations into MBRs, each group must be tight as possible so that more unpromising MBR pairs can be removed

By literature study [11] [13] normally overlapping subsequences are always similar so that grouping of overlapped subsequences of PAA representation minimizes the number of MBR rectangles. Grouping of PAA representation has another advantage i.e. it reduces the distance calculation time of two remaining MBR into O(1) time by using locality of those elements.

3. Grouping of MBR constructions

In the given sequence S of length M and motif length of L, group of MBRs consists of PAA representation for all subjects should be less than the previous distances. For memory optimization in Implementation PAA representation is discarded after consecutive group of MBR is constructed.

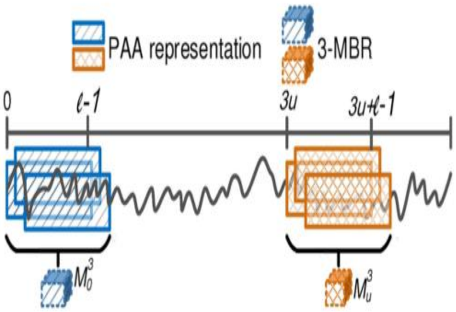


Fig 4: MBR Grouping Technique

MBR group refinement Technique

MBR grouping strategy not only provides reasonable grouping performance and also enables fast distance calculation of subsequence pairs. By using running sum technique we can calculate distance by O(1) time in incremental time order.

ALGORITHM 2

Group Refinement

Given sequence of length M , Motif length of L and MBR group Mn,Mm as **INPUT**

```
Assume BSF as ∞
for distance (Mn-Mm-1) Group index+1
if distance is (Mn-Mm+1) Groupind ex1 then exit
else calculate distance of two subsequence pairs
if distance is already matched with another group then continue
if given subsequence is the first pair then
compute aggregate distance of both pairs entirely
else compute aggregate distance of both incrementally
end
```

Group refinement algorithm provides fast distance calculation to discover motif between two pairs of MBRs. In our method we don't need to check inner subsequence pairs in MBR since if it is already matched.

B. Group Refinement and Filtering

From the Group refinement technique, we studied to find motif from MBR Groups. This section will elaborately tell about how to find surviving MBR pairs. There are many solutions to check MBR pairs. But all solution needs are with higher time complexity. So here we design a hierarchical structure for Group of MBR pairs for betterment of solution which is used to apply refinement process to find surviving pairs.

To construct hierarchical structure, we apply Hilbert R tree since it gives reasonable grouping quality and gives faster construction [18]. Since query may arrive frequently, construction cost also considered as one of the Performance factor. Instead of using page size we place HR tree on the memory. Based on Hilbert curve MBR's groups are sorted. Then with sorted order MBR's are grouped with next Level MBRs to construct each branch. This process will be continued till the last level.

Sorted - List aggregation [21] and best first closest pair, both uses minimum level heaps to prioritize the execution order of tree to reduce MBR pairs. Because of dimensionality problem cost of heap operations in motif discovery is already heavy. Let's take [21], it only calculates up to 6 dimensions. Both of this method pushes excessive level of min-heaps. By the curse of dimensionality problem in worst cases best first search may introduce false MBR groups into min-heaps. Because of this problem we introduce new searching algorithm with minimum heap cost called Locality based search.

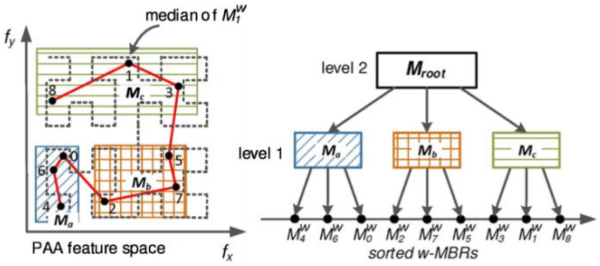


Fig 5: HR tree Construction

Locality Based Search:

For filtering operation, we recommend locality based search strategy which uses Locality of MBRs in the HR tree. Search task is performed by prioritizing the execution order buy exploring the locality of the HR tree. It utilizes queue only during the execution of the process.

C. Final Group Refinement

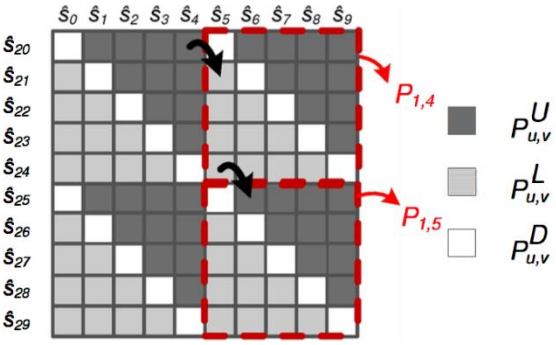


Fig 6: Final Group Refinement

This is the final stage of our framework. Here We Eliminates the large amount of unpromising MBR pairs to find the final Surviving Pairs.

Idea of Refinement Process:

Figure 6 Explains the Refinement process for example we use $P_{u,v\ to}$ indicate MBR pairs (M^w_u,M^w_v).If we have 4 MBR pairs for refinement Process we have to process each pairs individually and each pair has to calculate 9 out of 25 distances. Since we use refine MBR lazily, so distance calculation are sometimes derives incrementally in O(1) time.

D. Grouping All together

We finally process the entire ready to complete our framework, Fast-Motif. Below Algorithm explains the Fast Motif technique. First we are creating sum of two arrays technique then construct MBR for PAA groups. After that we create HR tree for created MBRs. And then we apply Locality based search strategy on HR tree for prioritizing the execution order.

ALGORITHM

FAST-MOTIF

```
1. Sequence s with length of M, Motif length of L as input
2. Compute sum of arrays for execution
3. Apply filter and Refinement
4. Construct HR Tree
5. Construct Group of MBR using PAA
```

IV. RESULT AND DISCUSSION

Here we conducted experiment results on our framework with Prior works. Al works are implemented in java environment and performed in Intel core processors i5 with 8GB RAM and the machine was running in ubuntu 12.04.

A. *Experiment Setup*

We use both synthetically and real datasets for this experiments. For evaluation of scalability we used Synthetic datasets. All sequences are generated by random model adopted in [5] [17] [23].

Below TABLE I show the experimental parameters and their values. In each experiment we changed each value of parameters setting other to its default values. For experiment we use 10 sequence of given dataset. Construction time of MK is 7.17s and for FM technique, it takes only 0.39s under default settings. Below graph state the response time to the variation of sequence length and motif length. Fig 7. Explains aviation of response time to sequence length and compares those to the other competitors.

TABLE I

Parameter	Default	Range
Sequence length	500k	250k,500k,750k,1000k
Motif length	500	3,00,50,07,00,900
Grouping Size	1%	0.5%,1%,2%,3%,4%
Level Factor	4	2,4,6,8
Level Constraint	2	1,2,3,4

Fig. 8 states the response time of our FM frame work according to its procedures.

Respectively fig 9 and fig 10 explains the response time variation when the motif length is varied. Fig. 9 Compares our framework response time with other competitors and fig. 10 explains the response time of 3 procedure in FM technique.

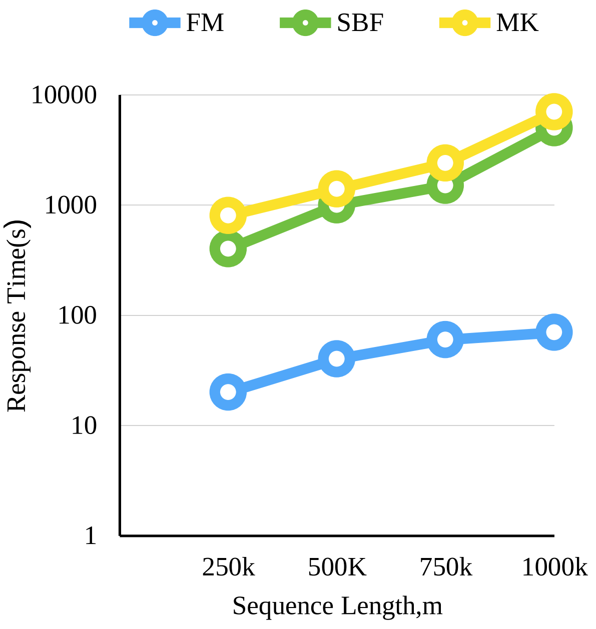


Fig 7: Vary sequence length

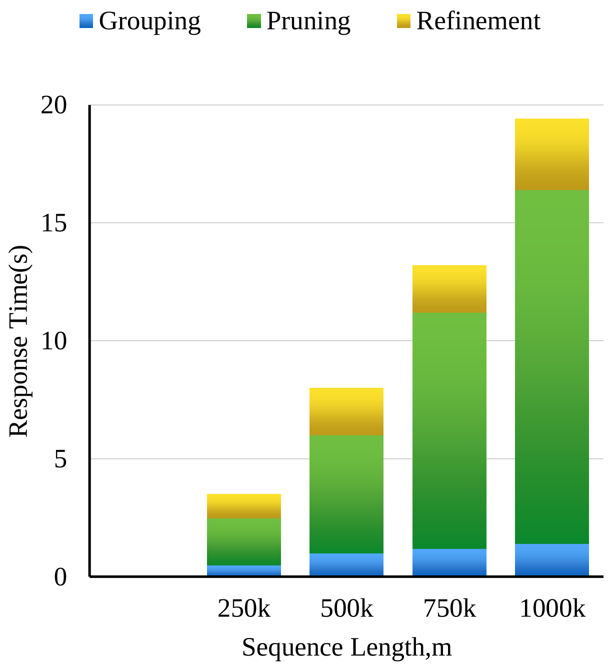


Fig 8. FM Vary Sequence Length

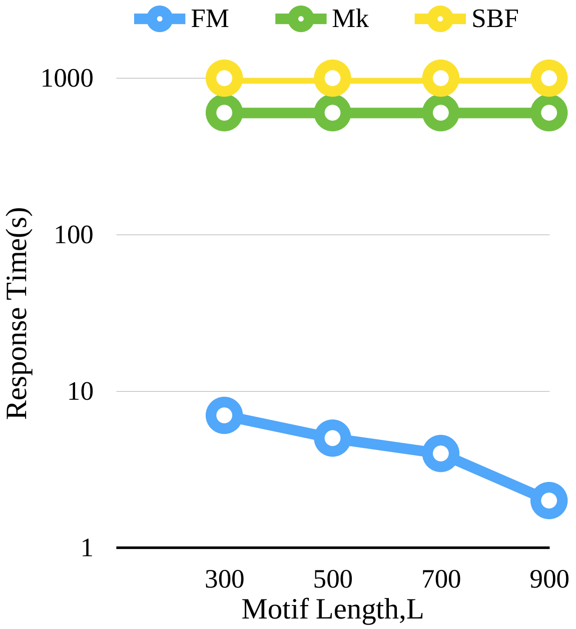


Fig 9 : Vary Motif Length

Real Dataset Experiment Result

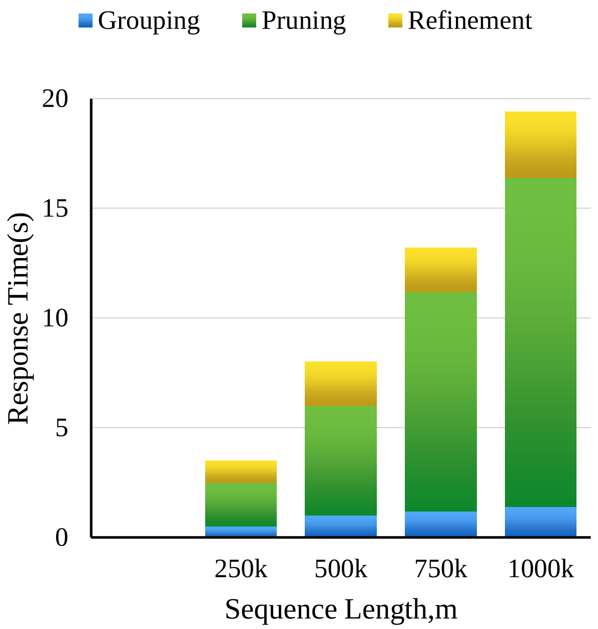


Fig 10 : FM Vary Motif Length

The given Fig. 11 and fig.12 explains the Experiment results of ECG and EEG. Fig. 11 explains results of experience that performed on ECG dataset and fig. 12 gives results EEG Experiments that performed. Both compares Chart gives the comparison of Our framework with other competitors.

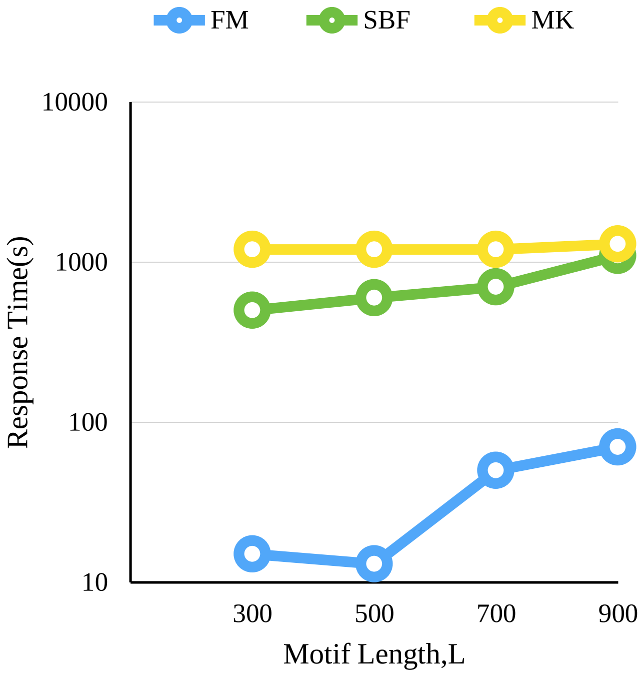


Fig 11: Effect of L on ECG

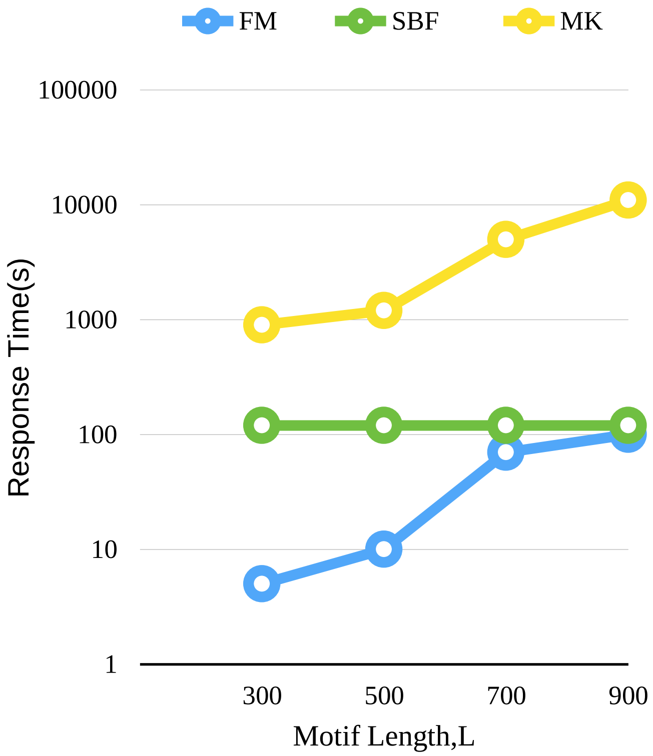


Fig 12. Effect of L on EEG

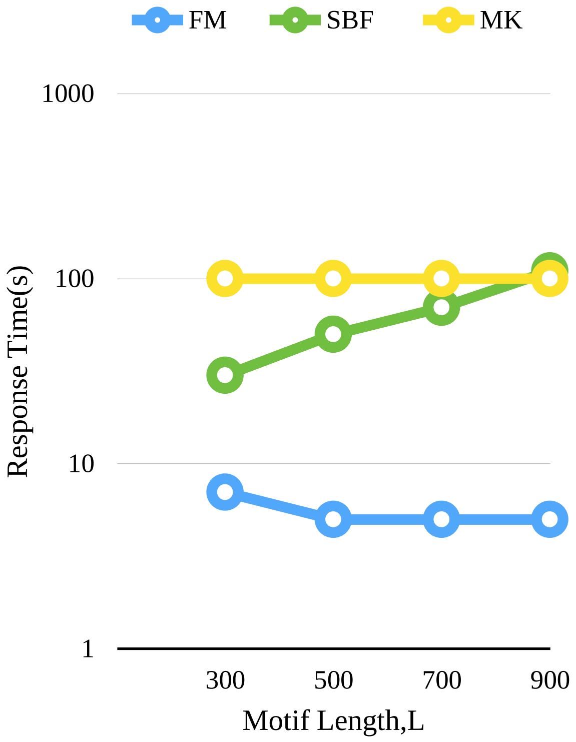


Fig 13. Effect of L on EPG

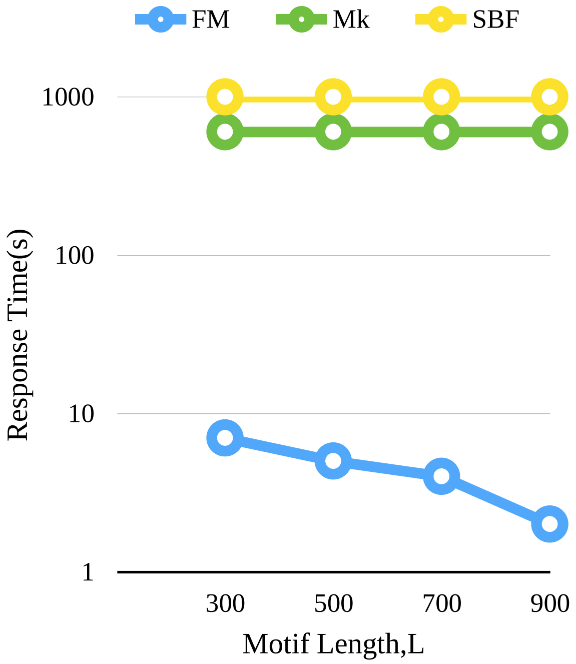


Fig 14. Effect of L on TAO

The other two fig.13 and fig.14 Explains the experiment results of EPG and TAO. Fig 13 Explains results of experience that performed on EPG dataset and fig. 12 Gives results TAO Experiments that performed. Both compares Chart gives the comparison of our framework with other competitors SBF and MK.

V. CONCLUSION

In this paper we presented our new framework called FM (Fast Motif) to handle motif discovery efficiently. Our framework promises to offer efficient motif discovery in online for emerging application where queries arises frequently. Our framework outperforms by discovering motif within ~20s in a sequence of million lengths.

In future work, we can implement this framework in nonparallel hardware architectures. And on the other side we can also improve our framework by discovering motifs based on other distance measurements such as normalized dynamic time wrapping.

REFERENCES

[1]. Castro, N and Azevedo, P L., (2010) "Multi resolution motif discovery in time series," in SDM.
[2]. Li, Lin, J and Oates, T., (2012) Visualising variable-length time series motifs, in SDM.
[3]. Mueen, A and Keogh, E. J.., (2010) Online discovery and maintenance of time series motifs." in KDD.
[4]. Mueen, A, Keogh, E. J. and Shamlo, N. B.,(2009) Finding time series motifs in disk-resident data, in ICDM.
[5]. Mueen, et al., (2009) Exact discovery of time series motifs, in SDM.
[6]. Keogh, E. et. Al., (2005) Hot sax: Efficiently finding the most unusual time series subsequence; in ICDM.
[7]. Rakthanmanon, E. J. Keogh, S. Lonardi, and Evans, S.,(2011) Time series epenthesis: Clustering time series streams requires ignoring some data, in ICDM.
[8]. Lin, J, Keogh, E, Lonardi, S and Patel, P,(2002) Finding motifs in time series, in Proc. of 2nd Workshop on Tempo Data Mining.
[9]. Mueen, A., (2013) Enumeration of time series motifs of all lengths, IEEE 13th International Conference on Data Mining,Dallas.
[10]. Shieh, J. and Keogh, E. J.,(2008) isax: indexing and mining terabyte sized time series, KDD.
[11]. Li, L. H. U, Yiu, M. L, and Gong, Z.,(2013) Discovering longest-lasting correlation in sequence databases, PVLDB, vol. 6, no. 14, pp. 1666- 1677.
[12]. Rakthanmanon, et. al.,(2012) Searching and mining trillions of time series subsequences under dynamic time warping, KDD.
[13]. chi Chiu, B, Keogh, E. J, and Lonardi, S. (2003) Probabilistic discovery of time series motifs, in KDD.
[14]. Keogh, E. L. and Kasetty, S.(2002) On the need for time series data mining benchmarks: a survey and empirical demonstration, in KDD.
[15]. Narang, A and Bhattacherjee, S, (2010) Parallel exact time series motif discovery, in Eu -Par (2).
[16]. Lin, J, Keogh, E. J, Wei, L and Lonardi, S., (2007) Experiencing sax: a novel symbolic representation of time series, Data Min. Knowl. Discov., vol. 15, no. 2, pp. 107-144.
[17]. Yi, B.K. and Faloutsos, C., (2000) Fast time sequence indexing for arbitrary Ip norms, in VLDB.
[18]. Kamel, I and Faloutsos, C.,(1993) On packing r-trees, inIKM.
[19]. Hjaltason, G. R. and Samet, H., (1998) Incremental distance join algorithms for spatial databases, in SIGMOD Conference.

[20]. Corral, A, Manolopoulos, Theodoridis, and Vassilakopoulos, M. (2000) Closest pair queries in spatial databases, in SIGMOD Conference.

[21]. Cheema, M. A, Lin, X, Wang, H, Wang, J and Zhang, (2011) A unified approach for computing top-k pairs in multidimensional space, in ICDE.

[22]. Quick-motif," http://degroup.cis.umac.mo/quickmotifs/.

[23]. Agrawal, R, Faloutsos, C and Swami, A. N.,(1993) Efficient similarity search in sequence databases, in FODO.

[24]. Tao, K. Yi, Sheng, C, and Kalnis, P.,(2010) Efficient and accurate nearest neighbor and closest pair search in high-dimensional space, ACM Transactions Database Systems, vol. 35, no. 3.

.