Engineering Sciences International Research Journal
ISSN (Print) : 2330 – 4338 ; Volume 1 Issue 1 ‖ February. 2013 ‖

Implementing Efficient DNA Matching using Suffix Tree

U. Vignesh[1]

Abstract: In this paper I propose a new data structure for an aspect of matching two DNA datasets resulting in an efficient and accurate pattern. The new data structure called FLAME-ST which includes suffix tree formation as its major role and also includes sequential datasets, extended structured motif extraction built in to it. Motif is defined as repeated combinations of patterns that occur in a datasets. FLAME-ST algorithm can be used in both real and synthetic datasets, which is of either large class of applications or small class of applications. To illustrate this pattern extraction algorithm FLAME-ST without loss of generality, I choose a local database scenario as most primitive data model. FLAME-ST is very appropriate for many applications such as DNA, in which it is used for mining noncontiguous subsequences. FLAME-ST algorithm is compared against several existing mining algorithms and a competing solution presented in the literature. FLAME-ST also outperforms existing mining algorithms on fast evaluation, scalable and performance metrics. To end, I evaluate FLAME-ST on two DNA datasets and show that it produce an result with much efficient and accuracy for DNA matching.

Keywords: Pattern extraction, suffix tree formation, structured motif extraction, motif.

1. INTRODUCTION

Implementation of efficient and accurate DNA matching using suffix tree formation includes these steps in an order for the mining aspect of patterns. The input is given as DNA datasets. Thus, the data representation does the process of identifying a most regard definitions in the actual dataset that they had given for an extraction process by FLAME-ST algorithm. FLAME-ST algorithm, which identifies the most matching motifs, i.e. frequently occurring patterns in the sequential dataset that suffix tree formulation resulted it. From this referential view of methodologies the required aspect can be extracted from the original dataset that it has been given. First identifies the length, distance formation, threshold level that it has to reach minimum support count that the data should taken for consideration in to the suffix tree formation from the original given dataset. Frequent combinations of repeatedly occurring patterns are extracted without omitting their levels and patterns are discovered with their accurate level. After data representation from that motif, two suffix trees are derived as Model suffix tree, first performs pruning, then set data on set of all possible model strings and Data suffix tree, sets the data on actual dataset, which contains the minimum support count of each node.

Suffix tree formation, which results an sequential dataset, this is an dataset there which an motifs are to be combined on with it, from this dataset extraction of motifs has to be done and problems are to be overcome during extraction. From the suffix tree, extended structured motif extraction is performed, which is the P-structured occurrence. The resulting array is an F-Existential array. P-structured occurrence, that is an existing step completion that the need to go through a frequent combinations of frequently occurring patterns in the sequential datasets. The out coming resulting array format is known as an F-Existential array and the array result to be noted. Thus, this is a motifs group that the FLAME-ST algorithm has given the result, which is to be compared with already extracted patterns that are stored in a database. Suffix tree formation done on an dataset with two types of suffix tree that to be required for extraction.

2. RELATED WORK

EXMOTIF is the mining algorithm; the problem formulation is to identify two tasks, An efficient algorithm is to be designed to calculate frequent motifs. The resultant motifs are to be evaluated on basis of statistics, in which the significant ones are to be reported. Thus, these two problems are also on basic consideration for an aspect of bioinformatics to analyze and interpret data in dataset. Here the algorithm is described by a known transcription factor in an every extracted patterns and the variance in motifs are to be noted. The proposed algorithm here is EXMOTIF, which includes its major components as sequences and structured motif template and their characteristics is to mine repeated motifs that have quorum. It deals with the mining of single or composite binding of DNA sequences but takes larger time. When EXMOTIF compared with state-of-the-art algorithm, it is efficient in terms of both time and space. Here EXMOTIF deals with structured motifs within numerous biological sequences. This system assumes gap range between each pair of motifs is known, but it fails in case of which motifs are unknown within the gap range. **Box-links** algorithm, which includes factor tree and k-factor tree as its data structure. It extracts structured motifs, when it has been built over on generalized factor tree. The box-links algorithm achieves time and space exponential gain. It requires O(N^2np) time and

$O(Nb_p(k,d))$ space. It considers a pk+(p-1)d- factor tree instead of a full suffix tree for the aspect of saving space but it fails in an efficiency and accuracy of an reported result.

3. MY PROPOSAL

3.1 ARCHITECTURE

The user 'A' gives an input dataset to the system to which extracting patterns has to be done on through it. To the given dataset, FLAME-ST algorithm has to be applied and based on this algorithm, the identification has to be done from a dataset such as the length, distance, threshold level and a minimum support count that has been estimated over a given dataset. There are two types of suffix trees. Data suffix tree which contains the counts in each node. Model suffix tree which is a tree on the set of all possible model strings. Suffix tree formation, which results an sequential dataset, this is an dataset there which an motifs are to be combined on with it, from this dataset extraction of motifs has to be done and problems are to be overcome during execution.

3.2 FLAME-ST ALGORITHM

FLAME-ST algorithm, which identifies the most matching motifs, i.e. frequently occurring patterns in the sequential dataset that suffix tree formulation resulted it. From this referential view of methodologies the required aspect can be extracted from this original dataset that it has been given. FLAME-ST algorithm follows these steps in an order for the mining aspect of patterns.

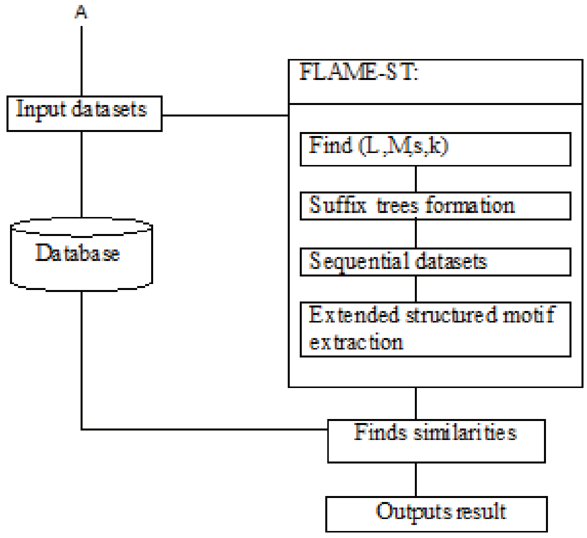
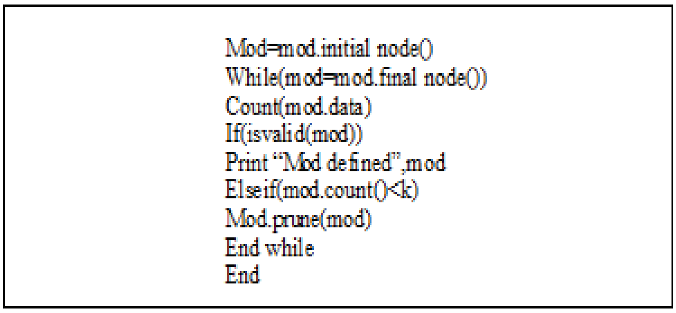


Fig. 1. FLAME-ST architecture

FLAME-ST algorithm follows these steps in an order for the mining aspect of patterns. First identifies the Length, distance formation, threshold level that it has to reach, minimum support count that the data should taken for consideration in to the suffix tree formation from the original given dataset. Thus the shown fig. 3 is the FLAME-ST algorithm.



```
Mod=mod.initial node()
While(mod=mod.final node())
Count(mod.data)
If(isvalid(mod))
Print "Mod defined",mod
Elseif(mod.count()<k)
Mod.prune(mod)
End while
End
```

Fig. 2. FLAME-ST algorithm

4. FLAME-ST & CONTROL FLOW

The overall process and control flow of the FLAME-ST is shown in fig. 3 implemented by using jdk 1.6. Flow diagram is a graphical representation of the "flow" of data through an information system, modeling its process aspects. Often they are a preliminary step used to create an overview of the system which can later be elaborated. It is also useful for visualization of data processing. Input dataset has been given. From the given dataset length of data has to be identified with their individual reference of a data in dataset. Distance matrix has to be formulated to compute a similarity between patterns of dataset. Maximum distance threshold to be identified to find a data's of repetition in a dataset. Data suffix tree considers an original given dataset that contains count in each node. Model suffix tree formed on set of all possible model strings and further pruning to be done. From these suffix tree calculation, sequential set of data are resulted and for further evaluation, sequential dataset to be used

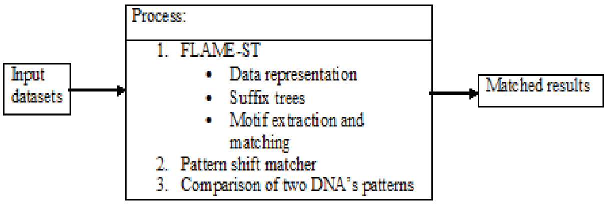


Fig. 3. Control flow of DNA matching

5. PERFORMANCE ANALYSIS

Thus, the use of FLAME-ST algorithm achieves a time and space exponential gain. Efficiency improvement is obtained because the data from the dataset are undergone a structured motif evaluation with the repeated combinations also to be noted. Thus, the graph shows the difference in efficiency of the executed result on variance of two DNA datasets by using FLAME and FLAME-ST algorithms.

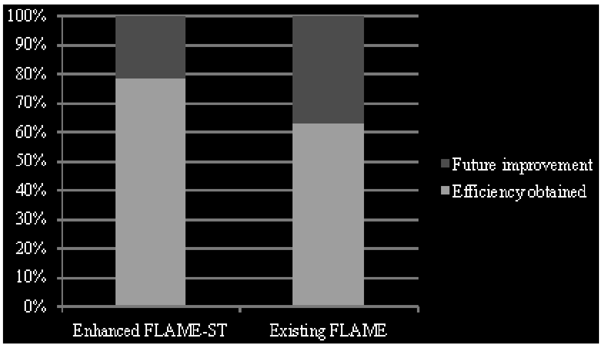


Fig. 4. DNA matching efficiency(FLAME vs FLAME-ST)

6. CONCLUSION

In this paper I have shown that DNA matching is efficiently achieved by using FLAME-ST algorithm. Also the pattern shift matcher technique used greatly influences the resource requirements of matching properties by providing the extracted patterns from the database to be matched with the patterns that has been extracted by using FLAME-ST algorithm. But the pattern shift matcher being in a separate module can be extended to construct patterns matcher in to FLAME-ST algorithm.

7. REFERENCES

[1] Avrilla Floratou, Sandeep Tata and Jignesh M. Patel, “Efficient and Accurate Discovery of Patterns in Sequence Data Sets”, vol. 23, 2011.

[2] Y. Zhang and M.J. Zaki, “EXMOTIF: Efficient Structured Motif Extraction,” Algorithms for Molecular Biology, vol. 1, pp. 21-38, 2006.

[3] J. Davila, S. Balla, and S. Rajasekaran, “Space and Time Efficient Algorithms for Planted Motif Search,” Proc. Int’l Conf. Computational Science, pp. 822-829, 2006.

[4] S. Rajasekaran, S. Balla, and C.-H. Huang, “Exact Algorithms for Planted Motif Challenge Problems,” Proc. Asia-Pacific Bioinformatics Conf. (APBC), pp. 249-259, 2005.

[5] A.M. Carvalho et al., “An Efficient Algorithm for the Identification of Structured Motifs in DNA Promoter Sequences,” IEEE/ACM Trans. Computational Biology and Bioinformatics, vol. 3, no. 2, pp. 126-140, Apr.-June 2006. N. Pisanti, A.M. Carvalho, L. Marsan, and M.-F. Sagot, “Risotto: Fast Extraction of Motifs with Mismatches,” Proc. Seventh Latin Am. Theoretical Informatics Symp. (LATIN), pp. 757-768, 2006.K.Sophos, “Security Threat Report”, July – 2008.

[6] L. Wenyin, G. Huang, L. Xiaoyue, Z. Min, X. Deng, “Detection of Phishing Webpages based on Visual Similarity”, 14th international conference on World Wide Web, Chiba, Japan, 2005, pp: 1060-1061.

[7] N.Wiliam Robinson and Sandeep Purao, “Monitoring Service Systems from a Language Action Perspective, March 2011.

[8] Y. Zhang, S. Egelman, L. Cranor, J. Hong, “Phinding Phish: Evaluating Anti-phishing Tools”, Annual Network and Distributed System Security Symposium, USA, February 2007.

* * *

***Mr. U.Vignesh** has been serving as a Assistant Professor with the Department of Information Technology (IT), Mookambigai College of Engineering affiliated to Anna University, Chennai. He obtained his B.Tech and M.Tech degree in Information Technology from Anna University, Chennai. His research interest includes Data mining, Distributed systems, Image Processing.*