An Efficient Mining for Approximate Frequent Items in Protein Sequence Database

J. Jeyabharathi[1], Dr.D. Shanthi[2]
[1] Associate Professor, Department of Computer Science and Engineering, C.R. Engineering College, Madurai, TamilNadu, India.
[2] Professor, Department of Computer Science and Engineering, PSNA College of Engineering and Technology, Dindigul, TamilNadu India.

***Abstract*—The rapid increase of available proteins, DNA and other biological sequences has made the problem of discovering the meaningful patterns from sequences, a major task for Bioinformatics research. Data mining of protein sequence databases poses special challenges, because several protein databases are non-relational whereas most of the data mining and machine learning techniques considers the data input to be a relational database. The existing sequence mining algorithms mainly focus on mining for subsequences. However, a wide range of applications such as biological DNA and protein motif mining needs an effective mining for identifying the approximate frequent patterns. The existing approximate frequent pattern mining algorithms have some delimitations such as lack of knowledge to finding the patterns, poor scalability and complexity to adapt into some other applications. In this paper, a Generalized Approximate Pattern Algorithm (GAPA) is proposed to efficiently mine the approximate frequent patterns in the protein sequence database. Pearson's coefficient correlation is computed among the protein sequence database items to analyze the approximate frequent patterns. The performance of the proposed GAPA is analyzed and tested with the FASTA protein sequence database. FASTA database files hold the protein translations of Ensembl gene predictions. GAPA is compared with the existing methods such as Approximate Frequent Itemsets (AFI) tree and Approximate Closed Frequent Itemsets (ACFIM) in terms of support, accuracy, memory usage and time consumption. The experimental results shows GAPA is scalable and outperforms than the existing algorithms.**

***Index Terms*—Approximate Frequent Patterns, Bioinformatics, Data mining, Generalized Approximate Pattern Algorithm, Pearson's coefficient correlation, Protein Motif, Protein sequence database, and Relational Database.**

I. Introduction

Sequential Pattern Mining is one of the important research areas in the field of data mining as patterns from a sequential database. Due to the massive collection and storage of the data, many industries are emerging interested in mining the sequential patterns from their database. In this case data mining methodology can be used to extract the knowledge from the huge amount of data. Nowadays, the collection of biological data such as DNA sequences, protein sequences etc. are emerging at an explosive rate due to the enhancement of existing methodologies and the introduction of new methods like microarrays. Hence, the data mining techniques are applied on the protein sequence databases to extract the useful and meaningful information from the large amount of data sequences. The approximate pattern mining is important in computational biology [1], where the task is to identify the short sequence, typically of length 6-15. It frequently occurs in the given dataset of protein or DNA sequence. These short set of sequences can deliver the hints about the locations of the regulatory regions. It depends on the repeated patterns of the biological sequences. Those repeated short sequences are not always identical, and some of the copies of the sequences can vary from others in a few positions. The repeated patterns or frequently occurring patterns are called *motifs*.

A protein sequence motif pattern is a short sequence, which is embedded within the sequences of a similar protein family [2]. These frequent patterns are used in the formation of ethnology trees, classification of protein and predicting the structure of the newly identified proteins. By finding the protein sequence motifs, an unknown sequence can be easily classified into its computationally predicted protein family for further biological investigation. Frequent itemset mining is a fundamental for many data mining processes such as mining association rules, classification, clustering etc. Most computations need to be performed to mine frequent itemsets over online data streams. The approximate frequent itemset maintains the frequent itemset, which save much more memory and an effective mechanism of the pruned itemsets to detect the new frequent itemsets.

This paper focusses on the problem of identifying the approximate frequent patterns in the protein sequences. Moreover, the existing algorithms cannot easily incorporate the noise tolerance due to the protein sequences storage nature. The proposed algorithm incorporates the preprocessing stage to remove the noise and irrelevant information presents on the protein sequence database. It helps to extract the approximate frequent patterns from the database in faster and effective manner. The major contribution of this paper is discussed as follows:

doi:10.4304/jetwi.6.3.324-330

a) Support and confidence metrics are used to compute the frequentness after the mining process gets completed. The frequent patterns of the protein sequences are identified based on the support values. The infrequent patterns of the protein sequences are removed based on the user defined threshold values.
b) A probabilistic model is proposed for mining the frequent itemsets in protein sequence databases, which comprising an uncertain sequence based on the possible model. The proposed probabilistic approximate frequent itemset model mine all itemsets that are frequent with a probability of at least the user defined threshold.
c) Association rules are used to mine the frequent patterns of the protein sequence to predict the approximate pattern from the given protein sequence database.
d) Pearson's coefficient correlation computation is applied to identify the k-mismatch, errored and don't care data items on the association rules in order to find out the approximate frequent patterns from the protein sequence database.

The remainder of this paper is organized as follows. Section 2 summarizes the related works in the protein sequence frequent itemset mining, approximate frequent itemset and the association rule mining mechanisms. Section 3 describes about the proposed approximate pattern prediction approach. Section 4 describes the performance analysis. And finally, the paper is ended with the conclusion and future work at section 5.

II. Related Works

There are various amounts of existing methods are available on mining the databases for approximate pattern mining. Previous work focused on mining the association rules. *Avrilia et al* presented a flexible and accurate motif detector to identify the frequent patterns with a various descriptions of motif models. It uses the flexible suffix tree based algorithm to calculate the frequent patterns [3]. *Elayaraja et al* introduced the extraction of motif patterns from protein sequences. The rough k-means clustering algorithm was used to predict the local protein sequence motifs. The structural similarity among the clusters were discovered and analyzed how the recurring patterns correlate with its structure [4]. *Fallahi et al* proposed an algorithm for frequent pattern discovery in protein sequences. It was based on an exhaustive search to compute all possible patterns, which satisfies the constraints of the user. Moreover, a search tree was used to discover the frequent patterns with the hash table to remove the useless searches [5]. *Wong et al* proposed an approximate protein DNA prediction approach. This approach was based on association rule mining. Here, frequent sequence tree and frequent structure class tree were used and generated from the structure to remove the meaningless rules [6]. The existing sequential pattern mining algorithm is inefficient for small and long sequences such as DNA and protein sequences. It takes more time to complete the pattern prediction from the large database.

Tak-ming Chan et al designed an approximate associated sequence pattern predictions for DNA protein interactions *[7]***.** The bindings among transcription factors and transcription factor binding sites were the basic protein-DNA interactions in transcriptional regulation. The exact rules to approximate rules were required for biological variations. *Dan et al* presented an approximate repeating pattern mining from protein sequence data with gap constraints. A data driven pattern generation method was used to avoid the unnecessary candidates. A back tracking pattern search process was used to identify the approximate occurrences of a pattern based on the user defined gap constraints. Finally, an Apriori based pruning approach was used to progressively prune the patterns and attain the search process [8]. *Bernecker et al* designed a model based probabilistic frequent itemset mining. The itemsets are identified based on the frequentness probabilities. Depends on the k-highest frequentness probabilities, the itemsets were extracted. It supports both the tuple and attribute uncertainty systems, and usually used to denote uncertain databases [9].

Matteo et al proposed a parallel randomized algorithm for approximate association rules in map reduce. The random sampling approach was used to cut down the dataset size reliant portion of the cost. Each system extracts a minor random sample of the dataset. The output retrieved from each system were filtered and grouped to reproduce a single output. Association rules were applied to find out the frequentness of the itemsets. The quality of the result was probabilistically guaranteed with the user specified accuracy and error probability metrics [10]. *Sahoo et al* formulated an effective association rule mining mechanism with new generic basis. Bittable was used to denote a compact database form in a single scan of the unique database. An algorithm was developed based on the frequent closest itemsets and their generators in an efficient manner. Lastly, an association rules were generated without accessing the database [11]. The drawbacks in the traditional association rule mining algorithms needs a strict definition of support value, which every item in a frequent itemset to occur in each supporting transactions.

Calders et al formulated an optimized incremental algorithm. This algorithm was used to mine the frequent itemsets in a stream. It manages a compact summary about the selected itemsets. A connection between the results and the size of a summary was established based on number theory [12]. *Deypir et al* proposed a variable size sliding window model for frequent itemset mining. The window size had kept constant by eliminating the existing transactions during the new transactions arrive. To calculate the detailed size of the window, the user should have the prior details about the time and scale of variations within the data stream. The formulated variable size sliding window frequent itemset mining was appropriate for noticing the most recent changes in the frequent itemsets over data streams. The window size was calculated dynamically based on the amount of concept variations [13]. *Li et al* articulated an efficient frequency itemset mining techniques over time sensitive streams. A

timestamp based sliding window model was constructed to further transform into a transaction based sliding window. An extended enumeration tree was used to incrementally manage the necessary information. The transform type bound was used to categorize the itemsets into deferred or ignored [14]. *Oh et al* discovered a frequent patterns based on constructing a frequent pattern network on data streams. In a frequent pattern network, edges and vertices denote a summarized detail about the transaction data. It offers user centered location based on the continuous frequent pattern mining process [15]. The problem of frequent itemsets are necessary to identify all rules from the given protein sequence or some other database D, which have support greater than or equal to the user defined minimum support.

Komate et al proposed a concept of approximate periodicity of an itemset. A tree based data structure called interval transaction ids list tree (ITL tree) was introduced. The tree structure manages an approximation of the occurrence details in an extremely compact manner for the periodic frequent itemsets. A pattern growth mining was used to create all periodic frequent itemsets. It was performed based on the bottom up traversal of the ITL tree for the user given threshold and periodicity [16]. *Li et al* presented an adaptive approximation technique to identify the frequent itemsets across the sliding window based data streams. It predicts the itemsets based on the counts instead of scanning the transaction for them [17]. *Erich et al* discussed about a fast approximation of probabilistic frequent closed itemsets. The frequentness for an itemset was calculated based on Poisson binomial distribution. The probabilistic frequent closed itemsets was used to minimize the number and redundancy of the result [18]. *Pyun et al* introduced a top k frequent pattern mining with combination reduction techniques. Here, the patterns were extracted from single paths was transformed into composite patterns in the mining process. The original patterns were recovered during the top k frequent patterns were extracted. The composite pattern was used to minimize the pattern combinations into a single path. Combination Reduction Method was applied for reduction and Combination Reduction Method for N-itemset was applied to consider the N-itemsets [19]. The fundamental problem of these approximate frequent pattern mining is that the exact match on patterns does not consider the noise present in the database. It causes two kinds of problems. In real time data, long patterns tend to be noisy and it does not achieve the support level with exact matching. In case of moderate noise, a frequent long type patterns can be mislabeled as an infrequent pattern.

III. Generalized Approximate Pattern Algorithm

This section discusses about the proposed approximate pattern prediction approach with the probabilistic approximate frequent itemset. The critical feature of the approximate pattern mining problem is describing the model under that two or more sequences are considered to match approximately. Formulating these models had interesting challenges: a model need to be robust to identify the occurrence of the protein sequence pattern even in the presence of noise and unnecessary itemsets. Another one is to design a model with a few variables to be set based on the user defined thresholds. Here, a Generalized Approximate Pattern Algorithm is proposed to identify the approximate frequent patterns from the given FASTA protein sequence database [20].

Initially the given protein sequence database is loaded and all the items present in the protein sequence database is preprocessed. The structure of the proposed model is depicted in fig.1.

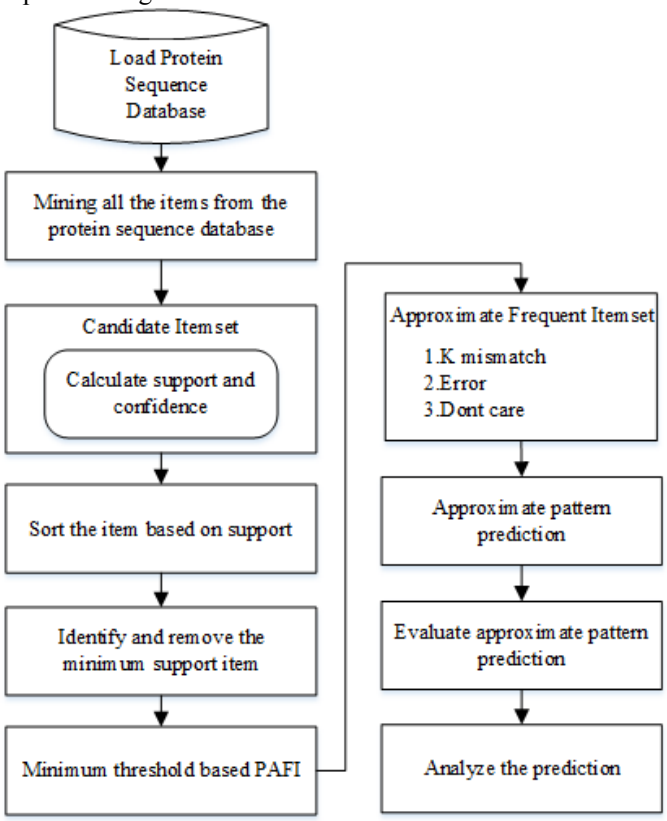

Fig.1. Flow of the proposed generalized approximate pattern mining algorithm

Samples of the FASTA protein sequences

TABLE 1 illustrates the few samples of the given FASTA protein sequence database. It includes the protein name, description and the protein sequence.

TABLE 1
FASTA PROTEIN SEQUENCE SAMPLES

Protein Name	Description	Protein Sequence
SPAC212.08c.1	GPI_anchored_protein_(predicted)"	MSPLIVGTLIIILLSGLATAFYVTWQGRLICAGVGLILEQAYEGGQMFNTLMAHCFETYNGVEKSGTQCV ADWLKVGLLAVTFGAGGPRLVNTLGGTFLTSPTAKRSNLYCDDFTGADYFSCELETLRPYTLMRKSLPY GNIHDVWINTTDTHQMIGVHMTLNGTDMIHYYNKTYVINYSGLKLNSSAINKRSYFYPQDSFLVSHAEW QDGNGIWTDTDYFAAMADCDFLGQNLGFWLASSYPNAYKWETQLWRTVGINLNGNIIYPGQLIMQTFN GS
SPAC212.12.1	S._pombe_specific_GPI_anchored_protein_family"	MSPLIVGTLIIILLSGLATAFYVTWQGRLICAGVGLILEQAYEGGQMFNTLMAHCFETYNGVEKSGTQCV ADWLKVGLLAVTFGAGGPRLVNTLGGSSPTTKRVIYIVMILLVLITLAVNLKH
SPAC212.06c.1	DNA_helicase_in_rearranged_telomeric_region,_truncated"	MGVRLVVHYRLPASSMDYVQETGRAGRDGKYAIAALFYEKYDSTWSSYVEDSMKNFLNDNTMCVRSF LASEMDGECVCYSLLGEESTVSTMYGVKPTLPETPKPAIATHSRYNASFSSSPPPQPGSSSGMSAMNTNTT STTPVSGKT
SPAC212.04c.1	S._pombe_specific_DUF999_family_protein_1"	MSNPESLKKQVEPPGYNELFMVEDVCNVDLEQGLDLCKPEKVNKQSQRSRQSRQSLFTNTIKPQKDKMN IKTNKIKEFLNDLFTEFSKFHNSYYPNGRISTQDKSRWVLLIIWSIITILTIDKKFKIKESYLEWIGENQSHSEI WGPIVIYVGLFILLLSAFNYCSKLIIKALPLISMVIAWVGVVIAAFSVIITATIAGVIAAFSVIITATIAGVIAA MVGILYFGHWLVYKILILAFGFKIVTSGDVCVSNTLPTHNGETALHSDATVGSDIEQIELQNMPTPVKK
SPAC212.03.1	hypothetical_protein"	MSIEFDDSSRHNMNMTQLMQLGAFDRRSGDDFMVQDFKNGIRDCSGIPVNNRNLAFKAYDAVKQKCDS SIKVFNIQDITIKGATWQHHNCQSTGKWYSQLYDYQNTFIGKQEYNILFDCYSYLKYNLNG

A. Preprocessing of Protein Sequence Database

Discovery of frequent itemsets of protein sequences can be a lengthy process due to the huge amount of data available. The aim of this preprocessing stage is to extract the important items within a single family sequence. Hence, the frequent itemsets extraction becomes faster and easier. This stage does not directly provide the result of the frequent pattern, but it removes the less importance itemsets from the protein sequences. After the protein sequence database items are preprocessed, the candidate itemset is determined for all the items. The support values are calculated for all the items. The threshold is fixed to remove the minimum support items from the protein sequence database.

TABLE 2
PROTEIN SEQUENCE AND THEIR SUPPORT VALUES AFTER PREPROCESSING

ID	Protein Name	Protein symbols	Support
1	Methinine	M	0.02108347788821735
2	Serine	S	0.09289522166697355
3	Proline	P	0.04792600082049029
4	Leucine	L	0.09684800363513808
5	Isoleucine	I	0.06121966635548334
6	Valine	V	0.06052461847739829
7	Glycine	G	0.050980765855544175
8	Threonine	T	0.054374709948067916
9	Alanine	A	0.06339579611785341
10	Plenylalanine	F	0.04607451387026993
11	Tyrosine	Y	0.034419137020801904
12	Tryptophan	W	0.011483095423451594
13	Glutamine	Q	0.037726402020861124
14	Arginine	R	0.04913466269988683
15	Cysteine	C	0.014844198369140952
16	Glutamic Acid	E	0.06479935136043105
17	Asparagine	N	0.05153798859281608
18	Histicline	H	0.022842363571953878
19	Lysine	K	0.06477189400815968
20	Aspartic Acid	D	0.053118132297060555

TABLE 2 illustrates the list of protein sequence with their support values.

B. Probabilistic Frequent Itemsets (PFI)

Let $A \subseteq I$ be itemset, the support of A is represented by *s(A),* and is defined as the number of sequences in which *A* looks in a protein sequence database. In a precise databases, *s(A)* is a distinct value. Let $S(g_j, A)$ is the support count of A in probable world g_j. Then the probability that *s(A)* has a value of a denoted by $Pb^A(a)$ is:

$$Pb^A(a) = \sum_{g_j \epsilon W, S(g_j,A)=a} Pb(g_j) \quad (1)$$

$Pb^A(a)$ $(a = 1, \dots, n)$ form a probability mass function of *s(A).* Let min-support ϵ (0, n] be an integer. An itemset A is said to be frequent if s(A) ≥ min-support. For uncertain database, the frequentness probability of A can be represented by $Pb_{freq}(A)$. It is denoted by:

$$Pb_{freq}(A) = \sum_{a \geq min-support} Pb^A(a) \quad (2)$$

Based on frequentness probabilities, it can be observed that an itemset in protein sequence is frequent.

C. Approximate Frequent Itemset Formulation

Once the frequent set of items from protein sequence database had been found out, it is easy to formulate the association rules which satisfy both the minimum support and confidence. Association rules (AR) identify all itemsets that have support greater than the minimum support and then utilizing the huge itemsets to formulate the preferred rules that have confidence greater than the minimum confidence.

From TABLE 2, the set of association predicted results are:

1, 2, 4, 5, 6, 8, 9, 10, 13, 14, 16, 17, 19, 20

Hence, the Frequent Protein Items are:

M, S, L, I, V, T, A, F, Q, R, E, N, K, D

TABLE 3
LEAST FREQUENT ITEMS

ID	Protein symbols	Support
3	P	0.04792600082049029
7	G	0.050980765855544175
11	Y	0.034419137020801904
12	W	0.011483095423451594
15	C	0.014844198369140952
18	H	0.022842363571953878

The set of least frequent values are presented in TABLE 3, those values are lesser than the fixed user defined threshold values.

D. Pearson's Coefficient Correlation

Pearson's coefficient correlation is computed for each item present on the association rule to identify the k-mismatch itemsets to approximately predict the frequent patterns. Pearson's correlation coefficient is defined as a statistical measure of the strength of a linear relationship between the paired items. It is represented by r and the constraint is defined as follows:

$$-1 \leq r \leq 1 \quad (5)$$

The type of correlation can be characterized into three categories:

1. Positive correlation: the other variable has a tendency to also increase
2. Negative correlation: the other variable has a tendency to decrease
3. No correlation: the other variable does not tend to either increase or decrease.

The positive results represent the positive linear correlation, negative values represents the negative linear correlation and a value of 0 represents no linear correlation. The closer value of 1 to -1 denotes the stronger linear coefficient.

IV. Performance Analysis

In this section, the performance of the proposed Generalized Approximate Pattern Algorithm is analyzed and compared with the existing Approximate Frequent Items Tree (AFI Tree) [21] and Approximate Closed Frequent Itemsets (ACFIM) [22]. Here, FASTA protein sequence database [20] is used for predicting the approximate frequent patterns. These database hold the protein translations of Ensembl gene predictions. AFI algorithm uses three datasets to evaluate the performance. The datasets are T5I4D100K and T10I4D100K, the third dataset is the real dataset called BMS-Webview-1 [21]. ACFIM uses the real life sales stream dataset, which includes 515,597 transactions [22]. In this paper, the proposed GAPA and existing AFI and ACFIM are validated based on the FASTA protein sequence database [20].

The performance of the proposed GAPA and the existing AFI and ACFIM are evaluated with the following set of metrics: Support, Memory usage, Execution time and Accuracy. The results shows that the proposed GAPA yields better performance than the existence methods.

A. Support Analysis

The support is calculated based on the equation (3). It is experimented for the proposed GAPA and the exiting AFI tree and ACFIM methods. Fig.2. shows that the proposed GAPA can result better support measure than the existing methods.

$$Support\ (AB) = \frac{Support\ count\ of\ AB}{Total\ number\ of\ transaction\ in\ database} \quad (3)$$

From the above equation, support of an item is denoted as a statistical implication of an association rule.

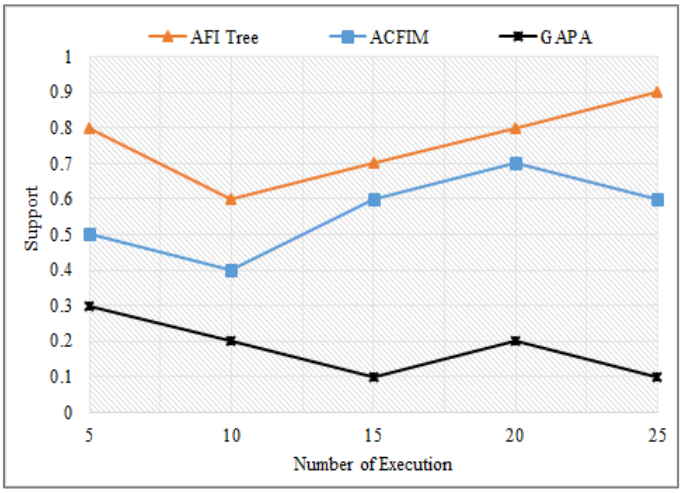

Fig.2. Support analysis between AFI tree, ACFIM and GAPA

B. Memory Usage Investigation

GAPA is implemented and tested up to five times and the results are monitored. The proposed GAPA utilizes less memory usage than the existing AFI tree and ACFIM, which is shown in fig.3.

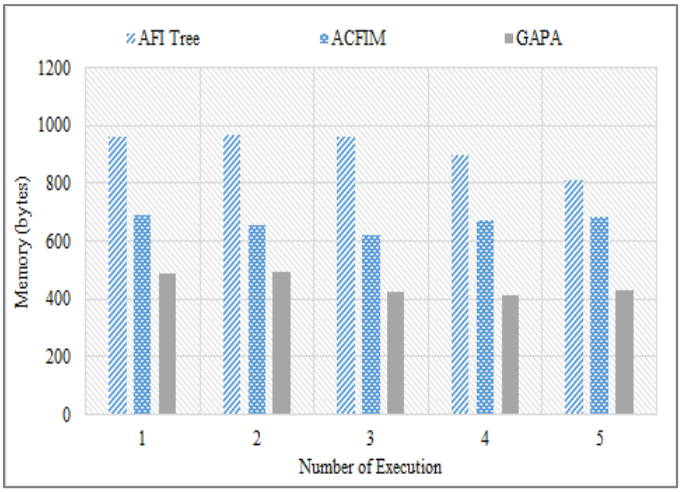

Fig.3. Memory usage analysis

C. Execution Time

The execution time is defined as the time spent by the proposed system to execute the approximate pattern prediction. The proposed system takes lesser time to execute the task than the existing system. Fig.4. shows that the proposed system performs better the existing techniques.

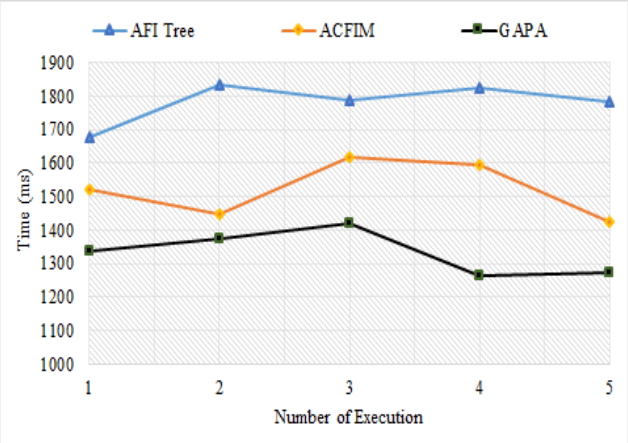

Fig.4. Execution time analysis

D. System Accuracy Examination

The proposed system accuracy is validated based on the precision, recall and F-measure properties. It is validated with the performance of the GAPA and the existing AFI tree and ACFIM methods. Fig.5. shows that the GAPA can yields better results than the existing methods. F-measure computes some average of the information retrieval precision and recall metrics. Precision is the number of correct results divided by the number of all returned results.

$$Precision = \frac{Number\ of\ correct\ results}{Number\ of\ all\ returned\ results}$$

(6) Recall is the number of correct results divided by the number of results that should have been returned.

$$Recall = \frac{Number\ of\ correct\ results}{Number\ of\ returned\ results\ to\ be\ returened}$$

(7) TABLE 4 illustrates the computed values for the mismatch, errored and don't care itemsets values.

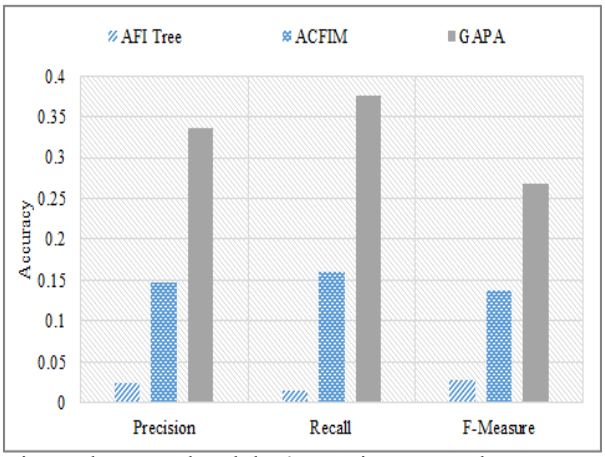

Fig.5. Accuracy analysis

TABLE 4
K-MISMATCH, ERRORED AND DON'T CARE ITEMS FOR GAPA

Itemsets	Precision	Recall	F-measure
K-mismatch	-0.34	-0.22	-0.18
Errored	0.1	0.49	0.65
Don't care	0.25	1.74	1.3

V. CONCLUSION AND FUTURE WORK

Many of the existing approximate algorithms maintain a large number of frequent itemsets of protein sequences, it consumes much memory and time. Hence, it degrades the performance of the approximate algorithms. To improve and enhance the approximation algorithm, this paper introduces a Generalized Approximate Pattern Algorithm (GAPA). It efficiently mines the frequent itemsets from the protein sequence databases. It dramatically improves the accuracy of the retrieved frequent itemsets. The experimental results shows that the proposed GAPA can performs better in terms of accuracy, execution time and memory usage. Also, the results are better than the existing methods like Approximate Frequent Item-Tree (AFI-Tree) and Approximate Closed Frequent Itemsets (ACFIM). Some of the applications includes gene finding, data cleansing, motif detection, protein function domain detection, disease diagnosis etc.

In future, the proposed method includes the pruning techniques to improve the accuracy of the retrieved frequent itemsets.

REFERENCES

[1] T. Calders, C. Garboni, and B. Goethals, "Approximation of Frequentness Probability of Itemsets in Uncertain Data," in *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, 2010, pp. 749-754.

[2] R. Hasan and J. Uddin, "Data Mining Techniques for Informative Motif Discovery," *International Journal of Computer Applications,* vol. 88, pp. 21-24, 2014.

[3] Avrilia Floratou, Sandeep Tata, and Jignesh M Patel, "Efficient and Accurate Discovery of Patterns in Sequence Data Sets," *IEEE Transactions on Knowledge and Data Engineering,* , vol. 23, pp. 1154-1168, 2011.

[4] E. Elayaraja, K. Thangavel, B. Ramya, and M. Chitralegha, "Extraction of Motif Patterns from Protein Sequence Using Rough-K-Means Algorithm," *Procedia Engineering,* vol. 30, pp. 814-820, 2012.

[5] A. Fallahi and M. H. Sadreddini, "An Efficient Algorithm for Frequent Pattern Discovery in Protein Sequences," *International Journal of Biometrics and Bioinformatics (IJBB),* vol. 6, p. 1, 2012.

[6] Po-Yuen Wong, Tak-Ming Chan, Man-Hon Wong, and Kwong-Sak Leung, "Predicting approximate protein-dna binding cores using association rule mining," in *2012 IEEE 28th International Conference on Data Engineering (ICDE),* , 2012, pp. 965-976.

[7] T.-M. Chan, K.-C. Wong, K.-H. Lee, M.-H. Wong, C.-K. Lau, S. K.-W. Tsui, *et al.*, "Discovering approximate-associated sequence patterns for protein–DNA interactions," *Bioinformatics,* vol. 27, pp. 471-478, 2011.

[8] D. He, X. Zhu, and X. Wu, "Mining Approximate Repeating Patterns from Sequence Data with Gap Constraints," *Computational Intelligence,* vol. 27, pp. 336-362, 2011.

[9] T. Bernecker, R. Cheng, D. W. Cheung, H.-P. Kriegel, S. D. Lee, M. Renz, *et al.*, "Model-based probabilistic frequent itemset mining," *Knowledge and Information Systems,* vol. 37, pp. 181-217, 2013.

[10] M. Riondato, J. A. DeBrabant, R. Fonseca, and E. Upfal, "PARMA: a parallel randomized algorithm for approximate association rules mining in MapReduce," presented at the Proceedings of the 21st ACM international

conference on Information and knowledge management, Maui, Hawaii, USA, 2012.

[11] J. Sahoo, A. Das, and A. Goswami, "An effective association rule mining scheme using a new generic basis," *Knowledge and Information Systems,* pp. 1-30, 2014/02/04 2014.

[12] T. Calders, N. Dexters, J. J. M. Gillis, and B. Goethals, "Mining frequent itemsets in a stream," *Information Systems,* vol. 39, pp. 233-255, 1// 2014.

[13] M. Deypir, M. H. Sadreddini, and S. Hashemi, "Towards a variable size sliding window model for frequent itemset mining over data streams," *Computers & Industrial Engineering,* vol. 63, pp. 161-172, 8// 2012.

[14] H. Li, N. Zhang, J. Zhu, H. Cao, and Y. Wang, "Efficient frequent itemset mining methods over time-sensitive streams," *Knowledge-Based Systems,* vol. 56, pp. 281-298, 1// 2014.

[15] K.-J. Oh, J.-G. Jung, and G.-S. Jo, "Discovering Frequent Patterns by Constructing Frequent Pattern Network over Data Streams in E-Marketplaces," *Wireless Personal Communications,* pp. 1-16, 2014/04/20 2014.

[16] K. Amphawan, A. Surarerks, and P. Lenca, "Mining Periodic-Frequent Itemsets with Approximate Periodicity Using Interval Transaction-Ids List Tree," presented at the Proceedings of the 2010 Third International Conference on Knowledge Discovery and Data Mining, 2010.

[17] C.-W. Li and K.-F. Jea, "An adaptive approximation method to discover frequent itemsets over sliding-window-based data streams," *Expert Systems with Applications,* vol. 38, pp. 13386-13404, 2011.

[18] E. A. Peterson and P. Tang, "Fast approximation of probabilistic frequent closed itemsets," presented at the Proceedings of the 50th Annual Southeast Regional Conference, Tuscaloosa, Alabama, 2012.

[19] G. Pyun and U. Yun, "Mining top-k frequent patterns with combination reducing techniques," *Applied Intelligence,* pp. 1-23, 2014/01/22 2014.

[20] *FASTA protein sequence database*. Available: ftp://ftp.ensemblgenomes.org/pub/fungi/current/fasta/schizosaccharomyces_pombe/pep/

[21] Y. Wang, K. Li, and H. Wang, "Maintaining only frequent itemsets to mine approximate frequent itemsets over online data streams," in *Computational Intelligence and Data Mining, 2009. CIDM'09. IEEE Symposium on*, 2009, pp. 381-388.

[22] H. Li, Z. Lu, and H. Chen, "Mining approximate closed frequent itemsets over stream," in *Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2008. SNPD'08. Ninth ACIS International Conference on*, 2008, pp. 405-410.