www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 4 Issue 7 July 2015, Page No. 13011-13016

Predicting Patterns over Protein Sequences Using Apriori Algorithm

Savitri Dhumale

Dept of CSE , B.L.D.E.A's CET, Vijayapur

savitridhumale2011@gmail.com

Abstract*: Data mining is the domain that helps to extract the useful data from large store house of data. It has a large scope in the field of biological science as it can solve the critical problems related to the sequence pattern mining by working on large data sets. It helps in classification of biological sequences. The building blocks of proteins are amino acids. These amino acids play important role. Some are good for health and some are harmful when they come in association with other amino acid. The proposed system is focused on generating the frequent amino acid sets and association rules. We have considered five diseases in our project. If the association rule corresponding to the particular disease occurs in the list of association rules obtained in output, then we can conclude that the disease corresponding to that association rule exists. The measuring parameters are support count and confidence.*

Keywords: Frequent amino acid sets, Association rules, Support count, Confidence, Apriori Algorithm.

1.Introduction

Data mining is a technique that helps to extract important data from a data house. It involves the analysing of data and picking out relevant information through the use of certain sophisticated algorithms. As more data is gathered, with the amount of data doubling every three years, so there is great scope for data miningand is important tool to transform this data into information. Data mining is process of analysing immense data regarding different perspective and summarize into some interested information. It is very important in the information technology field.The most important application of data mining is market basket analysis, fraud detection and analysing of biological data and so on. Bioinformatics generates a large amount of complex data like genomics and proteomics data. Data mining techniques are applied to such biological complex data to discover the hidden knowledge for crucial decision making. Discovered knowledge can be used by medical field to improve the quality of service. Mining frequent pattern and association rule is one of most important task of data mining, Frequent pattern can be defined as the pattern which occurring frequently in database.

Data Mining is a collection of techniques for efficient automated discovery of patterns from large data sets. Data mining can be defined as \the process of discovering meaningful new correlations among interested items, patterns and trends by digging into large amounts of data stored in warehouses. And it may be called Knowledge Discovery in Databases (KDD). In many applications (especially in dense data) with long frequent patterns enumerating all possible subsets of long length pattern is computationally difficult task.Thus, there has been recent interest in mining maximal frequent patterns in these "hard" dense databases [1].

In our project we are considering the database of amino acids. These are building blocks of proteins. Our aim is to find the most dominating amino acid. This is the one which occurs most frequently in the database. In order to find frequently occurred amino acid, we are using the Apriori Algorithm. After finding the frequent amino acids, we need to find frequent amino acid sets. Lastly we need to find the correlation among the subsets of these frequent amino acid sets. The correlation among the subsets is called as the association rule. This rule has two interesting measures; they are support count and confidence. As the confidence increases, the association among the subset increases.

The Section 1 contains the detailed introduction to the data mining and brief introduction to the proposed work and Apriori algorithm. The section 3 contains the literature survey, which describes the work carried out by different authors in brief. The section 4 contains the methodology of proposed work. This section contains the flow chart and algorithm related to proposed work. The section 5 contains the results of experiments conducted on amino acid database. Last section contains conclusion.

2. Literature Survey

In the year 1996, Fayyad et al. [2] proposed an order to discover only those association rules that satisfy the required measures. They have been divided into objective measures and subjective measures. Objective measures depend only on data structure. Subjective measures were proposed to collect the detailed information of decision maker. In the same year Silbershatz et al [3] proposed a classification of subjective measures in unexpectedness a pattern is interesting if it is surprising to the user and

pattern is interesting if it can help the user take some actions. In the year 2000, Zaki et al [4] proposed templates to describe the form of interesting rules (inclusive templates) and not interesting rules (restrictive templates). In 1997, Baralis et al [5] proposed a new query language called Constrained Association Query and they pointed out the importance of user feedback and user flexibility in choosing interestingness metrics. In 2005, Burdick et al [6] proposed algorithm allows grouping the discovered rules that share at least one item in the antecedent and the consequent.In the year 2007, Changchuan and Stephen et al [7] proposed 3-base periodicity. Its limitations are;the strength of 3-base periodicity is measured by computing the Fourier power spectrum.Computational complexity of Fourier transform is expensive when DNA sequences become large.

In the year 2014, Aditi V. Jarsaniya, Shruti B. Yagniket al [8] proposed A Literature Survey on Frequent Pattern Mining for Biological Sequence. Its limitation is, When the database is large, it is sometimes difficult and unrealistic to construct a main memory based FP-tree.In the year 2014, Dr. S.Vijayarani and Ms.S.Deepa et al [9] proposed research works involving different techniques to classify the protein sequences. They worked on GSP Algorithm (Generalized Sequential Pattern algorithm), Spade which utilizes the prefix-based equivalence classes that decompose the original problem in to smaller sub-problems that can be solved independently in main memory using simple join operations.Its limitation is; difficult to classify large amount of biological data and improve database design. In the year 2015, Pieter Meysman, Cheng Zhou, Boris Cule, Bart Goethals and Kris Laukenset al [10] worked on Mining the entire Protein Databank for frequent spatially cohesive amino acid patterns. They characterized protein structures feature that diverse set of frequent amino acid patterns that can be related to the stability of the protein's molecular structure and that are independent from protein function or specific conserved domains .

3. Methodology

Firstly the amino acid data set is taken from NCBI [National Center for Biotechnology Information] database which is available in ncbi website. The part of particular dataset is responsible for particular disease depending on the dominating amino acid. Thisdominating amino acid is obtained from candidate generation process using Apriori Algorithm. The flow diagram of Apriori Algorithm is shown in fig 1.

- ✓ Scan the dataset of amino acid and get the count of each amino acid and compare it with given support count.
- ✓ Generate the candidate amino acid set, this set consists of both types of amino acid i.e. the amino acid which satisfy given support count and other which do not satisfy. This set is frequent 1 amino acid set.
- ✓ Now next is to generate frequent 2-amino acid set of amino acid from frequent 1-amino acid set which satisfy given support count. Then check each frequent 2- amino acid set for its support count. Frequent 2 amino acid set of amino acid which satisfy given condition are again combined into set of 3 amino acids. Again there support counts are compared with given support count. This process continues until there are no frequent amino acid sets to generate next frequent n-amino acid set.

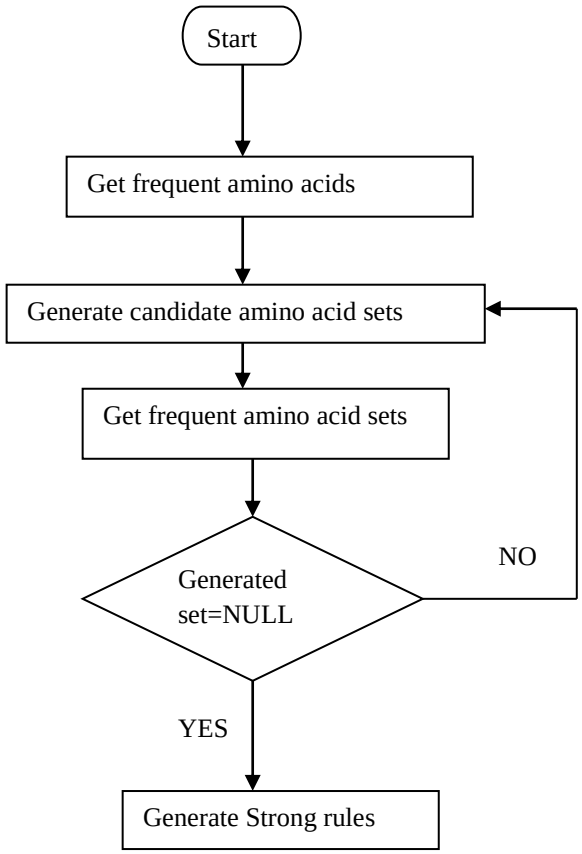

Fig 1: Flow diagram of Apriori Algorithm

- ✓ The confidence of each frequent amino acid set is calculated. If it is greater than or equal to given confidence then it is considered as strong association rule.
- ✓ Firstly take the required frequent amino acid set, and then form its subsets. Next we find the association between two subsets of frequent amino acid set by using formula shown in equation (1).

$$\text{Confidence } (A \Rightarrow B) = \frac{support\ count\ (A\ and\ B)}{support\ count(A)} \rightarrow (1)$$

The association among amino acid subsets is known by value of calculated confidence. The association among the amino acid subsets is strong if their calculated confidence is equal to or greater than given confidence. Hence those amino acids which satisfy given support count and confidence are responsible for causing viral diseases.

APRIORI ALGORITHM

Algorithm (Apriori) Find frequent amino acid sets using an iterative level-wise approach

Input: Database, *D*, of intervals; minimum support threshold, *min_sup*.
Output: *L*, frequent amino acid sets in *D*.

Method:

1) L_1= find_frequent_1-amino acid sets(D);
2) **for**(k = 2; $L_{k-1} \neq \phi$; k++) {
3) C_k= apriori_gen (L_{k-1}, *min_sup*);
4) **for each** interval $t \in D$ { // scan *D* for counts
5) C_t = subset (C_k , *t*); // get the subsets of *t* that are candidates
6) **for each** candidate c $\in C_t$
7) c.count ++;
8) }
9) L_k = { c $\in C_k$| c.count $\geq$ *min_sup* }
10) }
11) **return** $L = \cup_k L_k$;

Procedure apriori_gen(L_{k-1}:frequent *(k-1)*-amino acid sets; min_sup: minimum support)

1) **for each** amino acid set $l_1 \in L_{k-1}$
2) **for each** amino acid set$l_2 \in L_{k-1}$
3) **if** (l_1 [1] = l_2[1]) $\wedge$ (l_1[2] = l_2[2]) $\wedge$ … $\wedge$ (l_1[k-2] = l_2[k-2]) $\wedge$

(l_1[k-1] = l_2[k-1]) then {

4) c = $l_1 \bowtie l_2$; // join step: generate candidates
5) **if**has_infrequent_subset (c, L_{k-1}) then
6) **delete** c; // prune step: remove unfruitful candidate
7) **else add** c **to** C_k;
8) **}**
9) **return** C_k;

Procedure has_infrequent_subset(c: candidate *k*-amino acid set; L_{k-1}: frequent (*k-1*)-amino acid sets); // use prior knowledge

1) **for each** (*k-1*)-subset s **of** c
2) **if** s $\notin L_{k-1}$**then**
3) **return** TRUE
4) **return** FALSE

Step 1: Apriori finds the frequent 1-amino acids sets, L_1.
Step 2-10: L_{k-1} is used to generate candidates C_k in order to find L_k .
Step 3: The **apriori_gen** procedure generates the candidates and then uses the Apriori property to eliminate those having a **subset** that is not frequent.
Step 4: Once all the candidates have been generated, the database is scanned.
Step 5: For each interval, a subset function is used to find all subsets of the amino acid that are candidates.
Step 6-7: The count for each of these candidates is accumulated.
Finally all those candidates satisfying minimum support from the set of frequent amino acid sets, *L*. A procedure can then be called to generate association rules from the frequent amino acids sets.
The prune component (steps 5-7) employs the Apriori property to remove candidates that have a subset that is not frequent.

4. Experimental results

We conducted experiments on dataset which is shown in fig2.

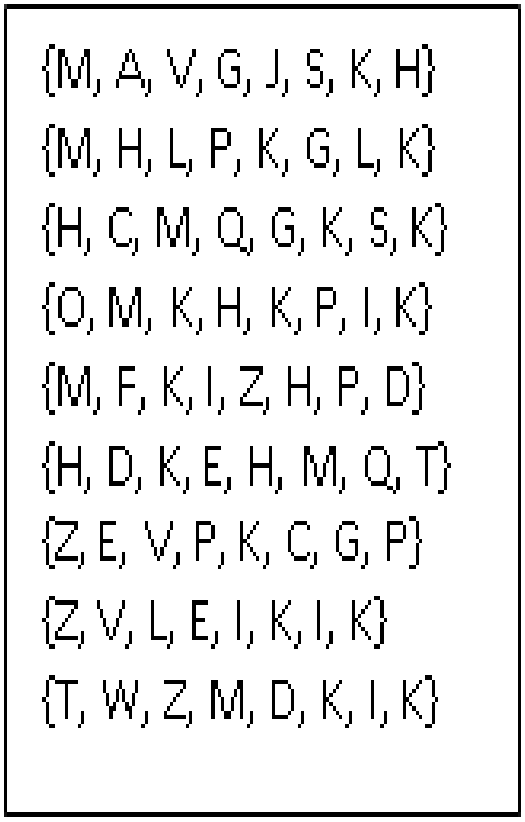

Fig 2: Dataset of amino acids

In this dataset there are 9 intervals of amino acids. First we considered the value of support count as 2, confidence as 90%. Next we considered the value of support count as 3 then 4 and lastly 5. Confidence is set to 90% in all four cases. In order to obtain strong association rules, confidence is set to 90%. Results are displayed clearly in this section using the Apriori algorithm on five diseases that is Epilepsy, Hartnup, Cystinuria, Alzheimer's disease and chrommaffintumor. This is summarizes in table 1. The confidence of all these five association rules is 100%. Epilepsy disease occurs when the {E}→{K} exists with support count 2 in generated list of association rules.Hartnup disease occurs if {D} → {M} occurs with support count 3. Cystinuria disease occurs if {G} → {K} occurs with support count 4 etc.

Table 1: Summarized Table

ASSOCIATION RULES	SUPPORT COUNT	DISEASE NAME
{E}->{K}	3	Epilepsy
{D}->{M}	3	Hartnup
{G}->{K}	4	Cystinuria
{H,G}->{M}	3	Alzheimer
{I,P}->{K}	2	Chromaffin tumor

The resulting patterns illustrate the frequent amino acid sets generated from the transactional protein dataset. These patterns were predicted using an Apriori process with the specified minimum Support count as 2. The generated frequent amino acid sets can be viewed as reports as shown in fig 3 below. The Apriori process maintains list of frequent amino acid sets.

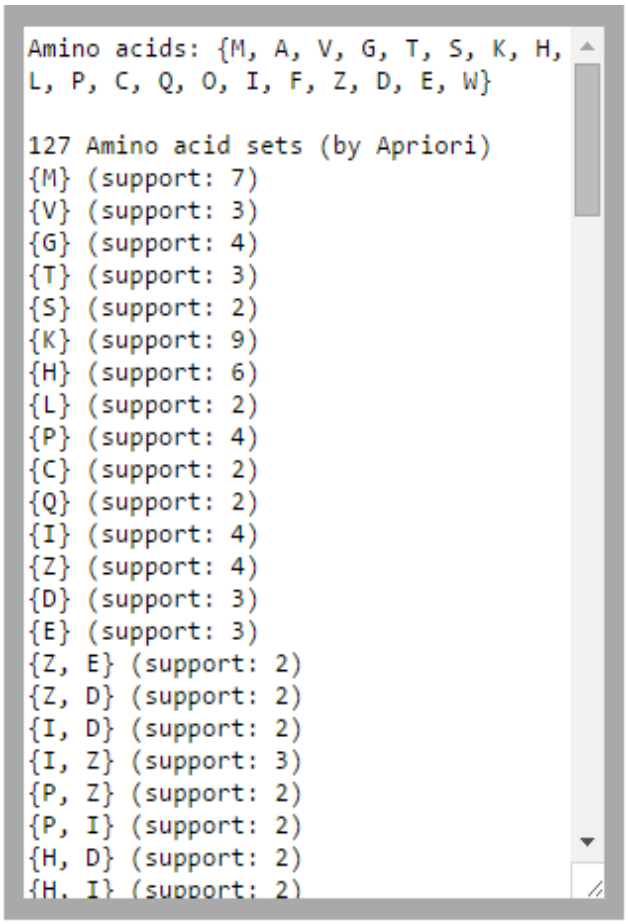

Fig 3: List of frequent amino acid sets obtained when support count = 2

The lists of frequent amino acids that are obtained when support count is set to 2 are shown in fig 3. Here the support count of each amino acid in the list is either equal to 2 or greater than 2. The total numbers of amino acid sets obtained are 127. Hence this is worst case as we are getting unwanted amino acids; though they satisfy given support count (2) along with required amino acids extra amino acids are also obtained which are not necessary.

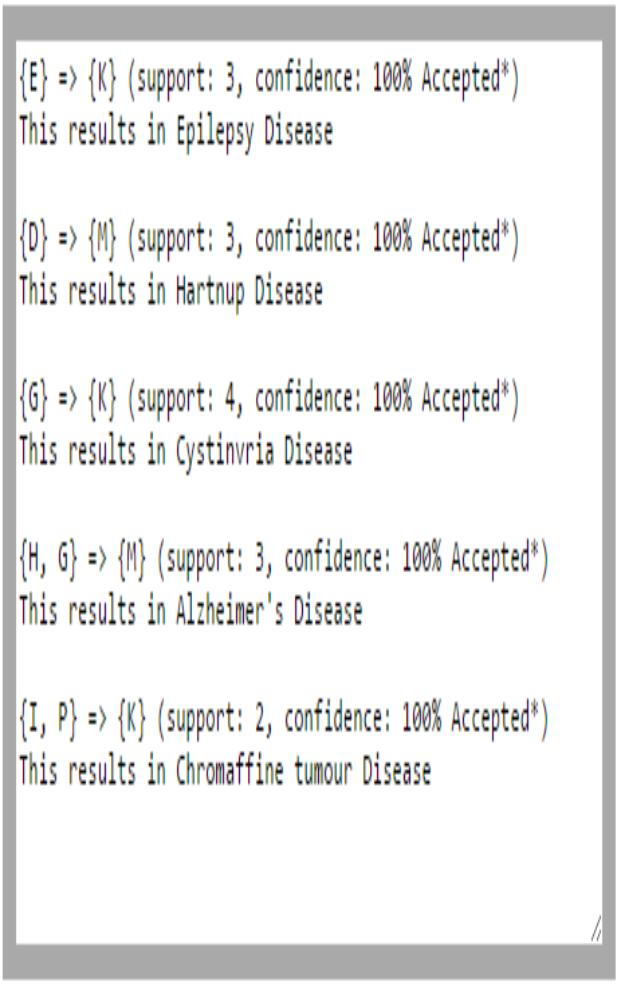

Fig 4: Diseases displayed when the support count is set to 2.

The fig 4 displays all five diseases. This output is obtained when the support count is set to 2 and confidence is set to 90%. If the support count is set to 3 then chrommaffin tumor disease will not be displayed because the support count of association of amino acids that causes chrommaffin tumor is 2. The 2 is less than 3 which is given support count.

If the support count is set to 4 then only Cystinuria disease will be displayed. Because the support counts of rest all association rules is less than 4. This is shown in fig 5. As the confidence increases, the association among the amino acids involved in particular rule will increase.Apriori property: All subsets of a frequent amino acid sets must also be frequent**.**Here "Nil" means that the corresponding association rule is not accepted because it does not satisfy the given support count. Hence it is not the strong rule. Only {G} → {k} is the strong rule and it is accepted because it satisfies the both given support count that is, 4 and confidence that is 90%.

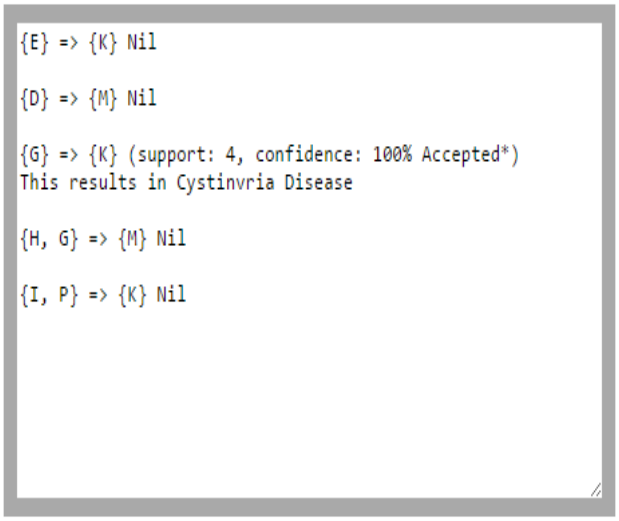

Fig 5: Diseases displayed when the support count is set to 4.

The D-rate is zero. There is no duplication of rules in the generated rule list. Running time (over two runs) is 11seconds. The support count value and the number of frequent amino acid sets generated are inversely proportional.The graph shown in fig 6 is plotted taking Number of frequent amino acid sets versus support count. We can see that as the support count value is increased, the number of frequent amino acid sets obtained goes on decreasing. Hence they are inversely proportional to each other. The number of association rules generated also decreases as the support count increases.

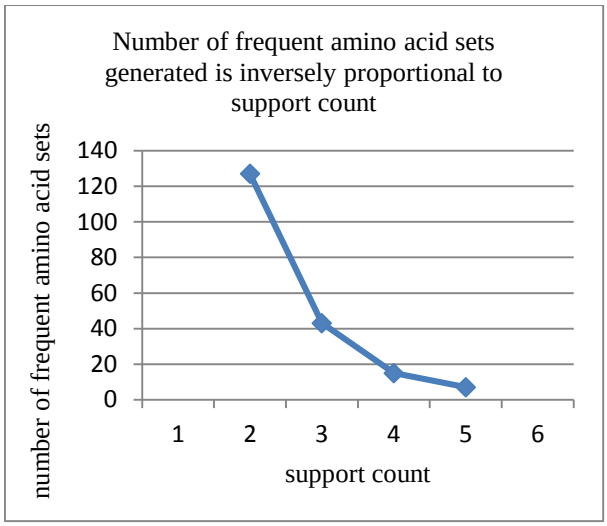

Fig 6: Number of frequent amino acid sets versus support count.

The number of association rules generated is also directly proportional to the number of frequent amino acid sets and inversely proportional to the support count value and confidence value. As the number of frequent amino acid set increases, the number of association rules generated, also increases. When the support count is set to 2, the numbers of association rules generated are 732. For support count 3, 4 and 5, the numbers of association rules generated are 132, 20 and 12. This is shown in graph in fig 7.

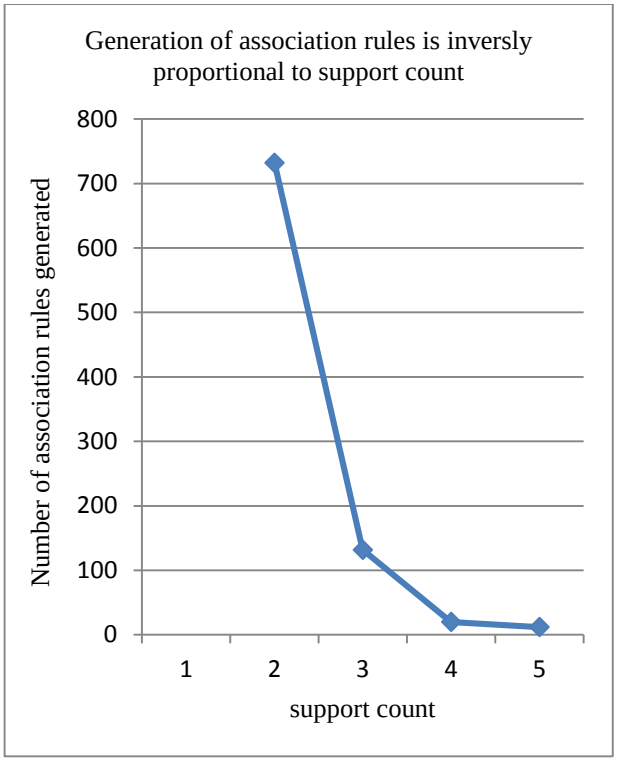

Fig 7: Number of association rules generated is inversely proportional to support count

From graph shown in fig 7, it is clear that there is reduction in number of association rules generated as the support count goes on increasing.

5. Conclusion

The proposed system is focused on finding the most dominating amino acids, which causes the viral disease. Among the generated frequent amino acid sets few amino acids are found to be strongly associated. Using the results retrieved from the protein dataset, a focus is given on which amino acids are more dominating. The result varies with the given support count and confidence. The support count and confidence are the two measures of the Apriori Algorithm.

Acknowledgment The author would like to thank Dr. PushpaPatil, HOD of Dept of Computer science and engineering, BLDEA, CET, Vijaypurfor her helpful suggesstions

References

[1] LakshmiPriya. G. ShanmugasundaramHariharan "A STUDY ON PREDICTING PATTERNS OVER THE PROTEIN SEQUENCE DATASETS USING ASSOCIATION RULE MINING" .Journal of Engineering Science and Technology Vol. 7, No. 5 (2012) 563 - 573 © School of Engineering, Taylor's University

[2] U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, "Advances in Knowledge Discovery and Data Mining". AAAI/MIT Press, 1996.

[3] A. Silberschatz and A. Tuzhilin, "What Makes Patterns Interesting in Knowledge Discovery

Systems," IEEE Trans. Knowledge and Data Eng. vol. 8, no. 6, pp. 970-974, Dec. 1996.
[4] M.J. Zaki and M. Ogihara, "Theoretical Foundations of Association Rules," Proc. Workshop Research Issues in Data Mining and Knowledge Discovery (DMKD '98), pp. 1-8, June 1998.
[5] E. Baralis and G. Psaila, "Designing Templates for Mining Association Rules," J. Intelligent Information Systems, vol. 9, pp. 7-32, 1997
[6] D. Burdick, M. Calimlim, J. Flannick, J. Gehrke, and T. Yiu, "Mafia: A Maximal Frequent Itemset Algorithm," IEEE Trans.Knowledge and Data Eng., vol. 17, no. 11, pp. 1490-1504, Nov. 2005.
[7] Changchuan Yin, Stephen S.-T. Yau. "Prediction of protein coding regions by the 3-base periodicity analysis of a DNA sequence". Journal of Theoretical Biology 247 (2007)
[8] Aditi V. Jarsaniya, Shruti B. Yagnik. "A Literature Survey on Frequent Pattern Mining for Biological Sequence". © 2014 IJIRT | Volume 1 Issue 6 | ISSN: 2349-6002
[9] Dr.S.Vijayarani and Ms.S.Deepa. "An Efficient Algorithm for Sequence Generation in Data Mining". International Journal on Cybernetics & Informatics (IJCI) Vol.3, No.1, February 2014.
[10] Pieter Meysman, Cheng Zhou, BorisCule, Bart Goethals and Kris Laukens. "Mining the entire Protein DataBank for frequent spatially cohesive amino acid patterns". Meysman et al. BioData Mining (2015) 8:4 DOI 10.1186/s13040-015-0038-4

Author Profile

Savitri Dhumale received B.E degree in Computer Science and Engineering from BLDEA CET, Vijaypur. Pursuing M.Tech in Computer Science and Engineering in BLDEA CET, Vijaypur.