A Doubleton Pattern Mining Approach for Discovering Colossal Patterns from Biological Dataset

K.Prasanna
JNIAS Research Scholar
Assistant Professor, Dept of CSE
AITS, RAJAMPET

M.Seetha, PhD
Professor
Dept of CSE
GNITS, HYDERABAD

ABSTRACT

The running time of existing algorithms in Frequent Pattern Mining (FPM) increases exponentially with increasing average data size. The existing algorithms on high dimensional datasets create large number of frequent patterns of small and mid sizes which are ineffective for decision making and shows deficiency on mining process. To discover large patterns or Colossal Patterns Doubleton Pattern Mining (DPM) is considered as very constructive for analyzing these datasets.

In this paper, DPM, An integrated approach for discovering Colossal Pattern from Biological datasets is discussed. DPM effectively discovers a set of Colossal Patterns using vertical top-down column intersection operator. DPM makes use of a data structure called 'D-struct', as combination of a doubleton data matrix and one dimensional array pair set to dynamically discover Colossal Patterns from Biological datasets. D-struct has a diverse feature to facilitate is, it has extremely limited and accurately predictable main memory and runs very quickly in memory based constraints. The algorithm is designed in such a way that it enumerates D-struct matrix iteratively and constructs a phylogenetic tree to discover colossal patterns and takes only one scan over the database. The empirical analysis on DPM shows that, the proposed approach attains a better mining efficiency on various Biological datasets and outperforms Colossal Pattern Miner (CPM) in different settings.

General Terms

Data Mining, Bioinformatics Frequent Pattern Mining.

Keywords

Colossal pattern, Doubleton Pattern Mining, Gene Association Analysis, Biological data, Colossal Pattern Miner (CPM).

1. INTRODUCTION

After its introduction in Data Mining, Frequent Pattern Mining (FPM) gained as a prominent data mining paradigm that assists to extract patterns that conceptually symbolize associations amongst discrete attributes and performs an imperative role in information mining and data exploration tasks as well as applications. Based on the intricacy of those relations, different types of patterns can occur. The most commonest kind of patterns tend to be mining association rules [1],[2], episodic and correlations [3], sequential patterns [4],[5], maximal patterns and frequent closed patterns [6],[7],[8] , classification [9],[11], and clustering [10].

There are numerous algorithms developed for fast and efficient mining of frequent patterns, such as Apriori [2] and its subsequent studies are in view of Apriori property [2]. The Apriori based algorithm achieved good diminution around the sized candidate sets. Nevertheless, when there are quite a few frequent patterns, it will take multiple scans over large databases to build candidate sets. In *pattern-growth approach*, including *FP-growth* [2] also uses the *Apriori* property. However, it recursively partition the database into sub databases to generating candidate sets. It makes limited scans over the database.

Modern Computational Biology known as Bioinformatics exploration is gaining much importance in the extraction of knowledge from biological data sets. The best part is its strong relationship with Medicine. The development in Medical Technology in last decade has introduced a new form of datasets called Biological datasets well known as Gene Expression Datasets and Microarray Datasets. Unlike transactional datasets, these High Dimensional Databases usually have few rows (samples) and a huge number of columns (genes). It is broadly believed that in a living organism, the accumulating of genes and their substances like DNA, RNA and protein sequences are usually activity in a complicated and orchestrated way. With the progression of DNA Microarray Data, it has presented a mixture of information and data analysis issues which are not discussed under conventional molecular biology. The data extracted from different microarray studies is represented normally in matrix form *N X M* of articulation levels, where N rows relate to different experimental conditions and M columns relate to genes under study. Due to this very high dimensionality, it needs efficient data mining methods to discover interesting knowledge from datasets including the exploration of DNA sequences for scientific or medical process.

In the literature, several algorithms were developed under pattern growth approach for discovering frequent patterns and closed patterns [7], [14], [15]. It uses enumeration based approaches [7], [15], [16], [17] in which item combinations are searched for frequent colossal patterns. In view of this, their running time increases exponentially with increase the average length of the records and makes minimum two scans over the Data Base. These will consume large extent of memory usage and predictably takes enough time when memory based constrains are present. These algorithms are rendering to be impractical on high dimensional microarray datasets. The complete set of frequent closed patterns are obtained using row enumeration space was first shown in [16], which was also observed in [12].

Nevertheless, the existing frequent pattern mining approaches still encounter the following difficulties.

- All item enumeration based mining methods are based on singleton patterns and take much time to compute these patterns.

- *Huge main memory is required for effective mining.* When memory constraints are present, an *Apriori*-like algorithm will not be effective since it produces enormous candidates for long patterns. Enough memory space is required to store candidate sets for discovering frequent patterns of different size. *FP-growth* [6] evades candidate generation by condensing into an *FP-tree.*
- *Real time applications require high dimensional and scalable. S*everal existing approaches are efficient for smaller size data sets. However as the dataset size increases the existing methods shows fit falls on core data structures and requires enough memory.
- *Multiple scans over the Database.* Most of the existing apriori and FP-growth approaches make several scans over the databases. Efficient data storage structures are needed to store intermediate results.

To do this colossal pattern are very useful. These sequences are useful to infer human behaviors from other species and also disclose the biological relevant information between gene associations and tend to discover gene networks [12] and bi-clustering of expressions [13].

Row enumeration search can be explored by constructing projected database recursively [16]. However, there is a need to consider column enumeration algorithms specified in many algorithms are proposed to mine frequent Colossal patterns. However for high dimensional datasets the pattern mining problem consumes more time and space. If a dataset is with 100 rows and 1000 columns, the existing enumeration algorithms works well if threshold is set to low while discovering Colossal patterns and often generates huge number of discovered patterns with no suitable information. However, traditional FPM methods are having fit falls in dealing with high dimensional datasets because of its dimensionality, size and main memory utilization. These pretenses a novel challenge on design and developing a new method which is *efficient in pattern mining on large databases where space requirement is limited.* For this reason Doubleton Pattern Mining (DPM) is considered for analyzing biological datasets.

In this paper, we study an efficient new algorithm DPM that is specially designed to discover colossal pattern over biological datasets are described. DPM makes use of a new data structure called doubleton data matrix D-struct which can be used to discover colossal pattern by performing attribute enumeration as depth first row wise enumeration, and efficiently reduces the searching time over the dataset. DPM has the following stages; first, a doubleton pattern discovery algorithm is proposed for the reduced datasets using D-struct that can fit into the memory. Second, DPM uses a new attribute enumeration column vector based intersection operator to discover Colossal pattern efficiently by reducing the search time and database scans. The experimental results show that this approach produces better results when mining biological datasets and outperforms Colossal Pattern Miner (CPM) on different settings.

2. BASIC PRELIMINARIES

In this chapter, first we present the basic concepts of Doubleton Pattern Mining and a formal problem statement and in the second one we describe the related work of the mining task by using an example.

2.1 BASIC CONCEPTS

Let G = $\{g_1, g_2 ....g_m\}$ be a set of m gene attributes, also called gene variables. An attribute *X* is a subset of attributes such that *X*, an attribute *G*= $\{g_1, g_2....g_m\}$ is also denoted as *G*= $g_1, g_2....g_m$. Let C= {C_1, C_2, . . C_n} be a set rows representing experimental conditions defied over biological dataset, where each ci is a set of n subsets called genes. Each row in C identifies a subset of items. C= (Rid, X) is a 2-tuple, where Rid is a row-id and X an attribute. A row C= (Rid, X) is said to contain attribute Y if and only if Y$\subseteq$ X. Table 1 shows an example of the dataset in which the genes are represented from g1 to g11. Let the first two columns of Table 1 be our sample data set. Table1 shows a dataset DB is the set of experimental conditions. Each C_i contains a subset of genes represented in lexicographic order.

Table 1: **sample transaction database DB**

Rid	Gene attributes
c1	g1 ,g2,g3,g5,g7,g8,g9
c2	g1,g3,g4,g5,g6,g8,g10
c3	g2,g5,g6,g7,g8
c4	g1,g2,g3,g4,g5,g6,g7
c5	g1,g2,g4,g6,g7
c6	g2,g5,g7,g8,g9,g10

Definition 1: A support(s) is defined as the number of transactions in *DB*, that contains both *X* and *Y*, represented as its frequency.

$$Support\ (X \rightarrow Y) = P\ (X\ U\ Y).$$

Definition 2: The ***Relative frequency*** of an attributes, *X*, *Y* is contained in Database, the relative frequency is defined as

Relative frequency (RF) = $\frac{support(X,Y)}{support(X)}$

Definition 3: Confidence (c) of the Rule X→ Y is true in the Database DB, if it contains the number of transactions containing X that also contains Y, represented as

$$Confidence\ (X \rightarrow Y) = P\ (Y\ |\ X) = P\ (X\ U\ Y)|\ P\ (X)$$

Definition 4: Colossal pattern: An attribute set *X∊I*, is a pattern sequence, if and only if *sup(X)* ≥ *minsup* and must be a *doubleton pattern.*

Definition 5: Doubleton pattern: A doubleton pattern can be frequent if both items in the set are frequent by themselves. A doubleton pattern set *(X, Y)* is frequent if both *X* and *Y* in the set are also frequent and it is true in Database DB, if it is having its *minsup* above 2.

2.2. Problem statement:

The problem of doubleton pattern mining is to find the complete set of colossal pattern in a given biological data set. The main objective is to discover all colossal pattern in a given biological dataset D with regard to user minimum support threshold.

3. DOUBLETON PATTERN MINING (DPM)

In this section, we study efficient mining of colossal patterns from biological dataset. We first illustrate the mining process of DPM with an example. Then, we present the DPM algorithm. The following subsections will describe the framework approach as shown in Figure1.

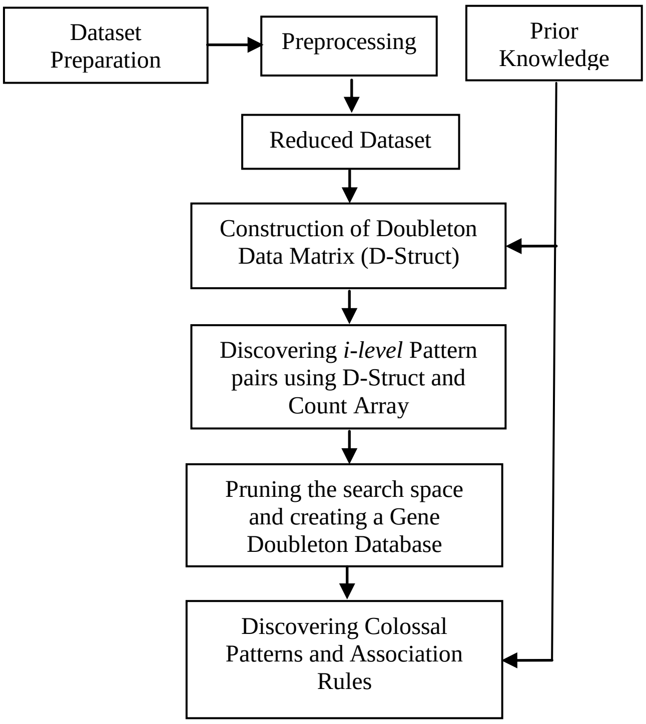

Dataset Preparation
Preprocessing
Prior Knowledge
Reduced Dataset
Construction of Doubleton Data Matrix (D-Struct)
Discovering *i-level* Pattern pairs using D-Struct and Count Array
Pruning the search space and creating a Gene Doubleton Database
Discovering Colossal Patterns and Association Rules

Fig. 1. Framework for Discovering Colossal Patterns from Biological Dataset using DPM

For a given set features in biological dataset, we define a Gene Expression matrix (M) with m rows and n columns in such way that experimental conditions on rows and genes on columns. Table 2 represents a bit matrix M, which is the equivalent gene Expression Matrix of the database D, where 1-means ‘overexpressed’ and 0-means ‘underexpressed’. A transaction in gene expression data is associated with ‘overexpressed’ data.

Table 2: Gene expression matrix (M) of the sample database (DB)

Rid	g1	g2	g3	g4	g5	g6	g7	g8	g9	g10
c1	1	1	1	0	1	0	1	1	1	0
c2	1	0	1	1	1	1	0	1	0	1
c3	0	1	0	0	1	1	1	1	0	0
c4	1	1	1	1	1	1	1	0	0	0
c5	1	1	0	1	0	1	1	0	0	0
c6	0	1	0	0	1	0	1	1	1	1
Support Count	4	5	3	3	5	4	5	4	2	2

Column wise pruning will be performed on gene matrix M based on minsup and eliminate the columns whose total occurrences are less than minsup. The pruned gene Expression matrix is shown Table 3 with the minsupp is 3.

Table 3: pruned gene matrix with support is 3

Rid	g1	g2	g3	g4	g5	g6	g7	g8
c1	1	1	1	0	1	0	1	1
c2	1	0	1	1	1	1	0	1
c3	0	1	0	0	1	1	1	1
c4	1	1	1	1	1	1	1	0
c5	1	1	0	1	0	1	1	0
c6	0	1	0	0	1	0	1	1
Support Count	4	5	3	3	5	4	5	4

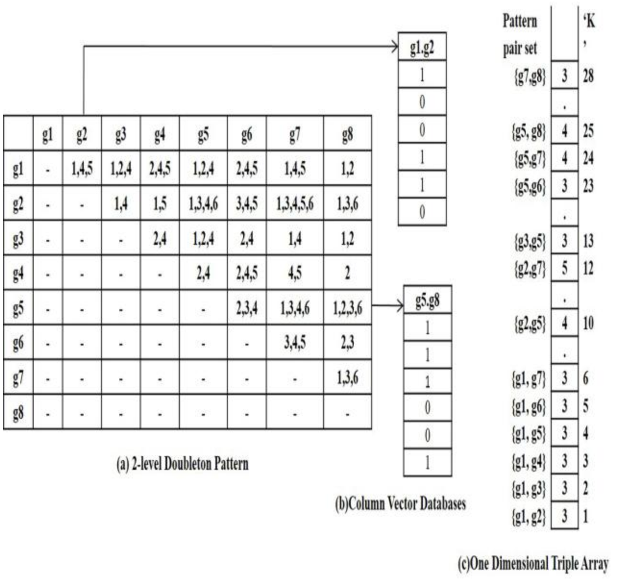
The support is given as the frequency of the rows in the dataset that contain a set of features G’. By definition **Support** is defined as the maximal frequency set of the transaction that contains *X*. The relative frequency of rows in the dataset that contain *X* is called its support of *X*, for a given set of items $X \subseteq I$. For set patterns, there exists a colossal pattern with a maximum length. The Colossal pattern are discovered using Doubleton Pattern Mining.

3.1. Discovering Colossal Patterns

In this section, we describe the process of discovering colossal patterns using an example dataset described above. High dimensional Databases characterized as experimental conditions as rows and large gene variables as columns. This distinctive characteristic will minimize the number of experimental conditions in pattern mining process by constructing a doubleton data matrix with vertical search strategies. Row enumeration algorithms works well when the dataset size is low dimensions. Horizontal search strategy cannot do efficient mining of patterns since the possibility of discovering exponential order of items. In this paper, we used vertical search strategies along with vector intersection operator to generate colossal patterns from biological datasets. D-struct matrix is constructed using vertical search strategy as shown in Figure 2. For the same gene expression data in Table1 with minimum support =3, we introduce a doubleton pattern mining method for mining Colossal pattern. This method explores the concept of vector databases as shown in Figure 2.

3.2. Finding doubleton frequent patterns:

“A doubleton pattern can be frequent if the items in the set are also frequent by themselves”. Using vertical search strategies, construct a doubleton data matrix such that each attributes is a bitwise column vector and their corresponding genes are those which there are in the all rows of this column vector. Now scan the dataset and mark the row number corresponding to each row and column. Each entry in the matrix is a column vector contains set of bit fields and storing with binary values. Its support values are stored in triple count array as shown in the Figure 2.



	g1	g2	g3	g4	g5	g6	g7	g8
g1	-	1,4,5	1,2,4	2,4,5	1,2,4	2,4,5	1,4,5	1,2
g2	-	-	1,4	1,5	1,3,4,6	3,4,5	1,3,4,5,6	1,3,6
g3	-	-	-	2,4	1,2,4	2,4	1,4	1,2
g4	-	-	-	-	2,4	2,4,5	4,5	2
g5	-	-	-	-	-	2,3,4	1,3,4,6	1,2,3,6
g6	-	-	-	-	-	-	3,4,5	2,3
g7	-	-	-	-	-	-	-	1,3,6
g8	-	-	-	-	-	-	-	-

(a) 2-level Doubleton Pattern

g1,g2
1
0
0
1
1
0

g5,g8
1
1
1
0
0
1

(b)Column Vector Databases

Pattern pair set		'K'
{g7,g8}	3	28
	.	
{g5, g8}	4	25
{g5,g7}	4	24
{g5,g6}	3	23
	.	
{g3,g5}	3	13
{g2,g7}	5	12
	.	
{g2,g5}	4	10
	.	
{g1, g7}	3	6
{g1, g6}	3	5
{g1, g5}	3	4
{g1, g4}	3	3
{g1, g3}	3	2
{g1, g2}	3	1

(c)One Dimensional Triple Array

Fig. 2. Doubleton data matrix with vector database and Count Array

3.3. One Dimensional Triple Array Pair Set

In General, the Association Rule mining algorithms maintain different item count frequency values throughout a scan over database. For instance, it is essential to have adequate main memory to hoard each pattern count that the number of times a pattern pair sets occurs in the transaction database. It is hard to update a 1 to a count set where the counting sequences are stored in different memory locations and difficult in loading the page to main memory. In such cases, these algorithms will be slow in finding that pattern pair count in main memory as it takes extra overhead on processing time and increases the time to discover frequent pattern set.

When it comes to high dimensional datasets, it is difficult to maintain all in one memory. So a new one dimensional triple array set is used to count all the pattern occurrences in the given database. To optimize main memory, a pattern pair (i,j) occurrence in the dataset should be counted in one place. This approach makes half of the array as useless. Count Array (CA) is a more efficient way to store colossal pattern in memory. A count array is defined as a one-dimensional triple array set which will store a count as CA[k] for the pair (i,j), with $1 \leq i < j \leq n$, where

$$K = (i-1)\left(n - \frac{i}{2}\right) + (j - i)$$

To discover colossal pattern, DPM performs a iterative depth first search (DFS) on column enumeration strategy. By imposing backtracking search order on column sets, we are able to perform a systematic search over colossal pattern. A 2-level doubleton pattern pairs using vector databases and triple array are discovered as shown in Table 4.

Table 4: a 2- level patterns pairs discovered using doubleton matrix

1	{g1, g2}, {g1, g3},{g1, g4}, {g1, g5}, {g1, g6} , {g1, g7}
2	{g2, g5}, {g2, g6} , {g2, g7}, {g2, g8}
3	{g3,g5}
4	{g4,g6}
5	{g5,g6}, {g5,g7}, {g5, g8}
6	{g6,g7}
7	{g7,g8}

3.4. Pruning the search space and creating a doubleton database

Each Colossal pattern corresponds to a unique set of genes. By enumerating all possible combinations on genes, we discovered all patterns sequences in the dataset. However, pruning the search space must be introduced to minimize unnecessary exploring on set of genes.

Let R be the gene sequence discovered from doubleton matrix, a Colossal pattern are identified as R-gene doubleton database which exclusively contains a particular gene and its Rid count must be above the min support threshold as shown in Table 5. All the discovered doubleton pattern pair sets can be divided into 7 non-overlap subsets based on the doubleton data matrix:

- The ones containing gene g1,
- The ones containing g2 but not g1,
- The ones containing item g3 but no g1 nor g2,
- The ones containing g4 but no g1, g2 nor g3,
- The one containing g5 but no g1,g2,g3 nor g4
- The one containing only g6, g7
- The one containing only g7 and g8.

Once all the pattern pair sets are found, the complete set of vector database is done.

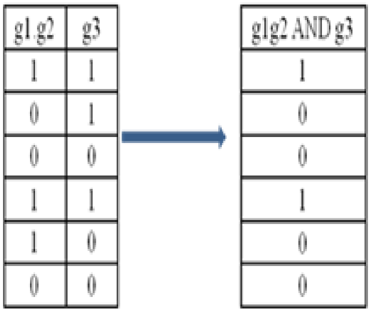
Table 5: A Gene Doubleton Database.

S.no	Gene	Conditioned on	Rid numbers
1	g1	{ g2, g3, g4, g5, g6, g7}	1,2,4,5
2	g2	{ g5, g6, g7, g8}	1,3,4,5,6
3	g3	{g5}	1,2,4
4	g4	{ g6}	2,4,5
5	g5	{g6,g7, g8}	1,2,3,4,6
6	g6	{g7}	3,4,5
7	g7	{g8}	1,3,6

3.5. Discovering Colossal pattern

From the discovered gene doubleton databases, we can expand each i-level pattern pair sets to form a new bitwise column vector to determine the equivalent Colossal pattern which is frequent or not. In column enumeration search strategy, each pattern is a column set and its adjunct gene is those which there are all rows of this column set. A column bit vector, which is the result of performing intersection operation on column vectors to determine the corresponding pattern. The discovered doubleton pattern pair sets can be mined to discover colossal pattern by construing a one dimensional triple array pair set and mine each pattern recursively

The left over mining process can be performed on D-struct, only without referring the original Database. Pattern are discovered one by one using triple array pair set values. For every doubleton pattern there is a k value. Using this k values recursively over the doubleton patterns and vector intersection operator collectively discovers pattern as shown in Figure 3.



g1.g2	g3
1	1
0	1
0	0
1	1
1	0
0	0

g1g2 AND g3
1
0
0
1
0
0

Fig. 3. Column vector intersection operator

In the above example the pair set g1.g2 can be explored on g3 to create a new pattern pair set as g1.g2 AND g3. It performs bitwise vertical intersection on g1g2 and g3 and discovers a new doubleton pattern pair set. Its corresponding count value is stored on a 3-level triple count array. g1g2 AND g3 is not equal to either g1g2 or g3. Hence it is also called as colossal patterns and its count is 2, which is stored in triple array. By performing the above process recursively we can discover colossal patterns. Clearly it takes one scan over the dataset to discover set of colossal patterns are shown in Table 6.

Table 6: Colossal patterns

S.No	COLOSSAL PATTERNS
1	{g1,g2,g3,g5,g7}
2	{g1,g2,g4,g6,g7}
3	{g1,g3,g4,g5,g6}
4	{g2,g5,g6,g7}
5	{g2,g5,g7,g8}
6	{g1,g3,g5,g8}
7	{g1,g2,g7}
8	{g1,g4,g6}
9	{g2,g5,g7}
10	{g2,g6,g7}
11	{g5,g6,g8}
12	{g2,g7}
13	{g5,g8}

3.6. DPM Algorithm

In a given gene database, a relevance frequency(RF) , the problem of mining the set of doubleton patterns can be considered as partitioning into n-sub problems. The problem of partitioning can be performed recursively that is each subset of DPM can be further divided when necessary. This forms a divide and prune framework. The mine the subsets of DPM, we construct corresponding doubleton Databases.

Given a *DB*, let i be the frequent attribute or gene constituted as a column vector in *DB*. The *i-level* doubleton *DB* denoted as $DT|_i$ is the subset of *DB* containing *i*, and all occurrences are stored in count array (CA). All the genes that precedes *i* can be formed as Colossal pattern.

Let *j* be a frequent attribute in i-Doubleton Database and *i* is a Doubleton pattern set. The *ji-DB* is the set of transactions containing *i* and *j* denoted as $DB|_{ij}$ by performing bitwise intersection operation on column vector *DB* and all the occurrences are stored in its *i-level* count array crated dynamically. Thus patterns sequences are discovered by repeating the process for all attributes.

DPM Algorithm

Input: Gene Database and relevance frequency as min support threshold.

Output: the complete set of Colossal Pattern.

```
1. Initialize CP←0, let CP as set of Colossal Patterns
2. Scan the database and compute doubleton data matrix and discover all doubleton patterns pair set and create a doubleton database $DT|_i$.
3. Let E: the set of doubleton patterns on DT|i: Doubleton Database and $A_i$: Attribute list.
4. for i = o to n do
   for each subset $A_i$ of A of size i, do if DP_Col_set(A,E) then CP← $A_i$
```

Procedure DP_Col_set(A,E)

Let ***E***: the set of doubleton patterns on ***DT|i***: Doubleton Database

A_i : Attribute list.
h is the local variable in Doubleton Database.

```
1. For each row in 'e' in E do
2. Set $DT|_i$ ←0,
3. If subset in h has same value as e for the attributes Ai but a different core pattern value then return false.
4. Let j be the set of attributes in DB, such that they appear in every transactions of DB, then insert $i \otimes j$ to CP and create its column vector DB's, count array and verify $i \otimes j$ count should be above 3.
5. For each remaining attribute i in Ai, starting from the Recursively call DP_Col_set($A_i$,E) to build its i-level doubleton Database DT|i and discover all its patterns using dynamically created count array.
```

4. PERFORMANCE ANALYSIS

In this section we will revise the performance of our algorithm with Colossal Pattern Miner. In our performance study we used the variant size of the datasets; it is difficult to assess the minsup threshold as an absolute number. Instead, minsup threshold is determined by relative frequency (RF). Experiments are performed on a synthetic dataset generated using a synthetic data generator and two real datasets obtained from UCI [18] to compare the algorithm. Table 7 contains the dataset descriptions.

Table 7: Dataset Descriptions

Dataset name	Attributes/Genes	Samples/ Experiments
T100-AT10-I100-P50-AP5	100	1000
SAGE	11082	207
Lung Cancer (LC)	12533	181

Table 8 and Table 9 shows the result of running two algorithms DPM and Colossal pattern miner on a Synthetic Dataset and LC. It is notices that with increasing minimum support (RF) all the algorithm performance in data set will be decreased.

Table 8: performance on Synthetic Dataset in sec

Support	Colossal Pattern Miner(CPM)	DPM
0.03	32	16
0.05	14	6
0.06	6	4
0.08	3	2
0.09	2	2

Table 9: Performance on LC in sec

Support	Colossal Pattern Miner(CPM)	DPM
0.03	110	60
0.05	92	48
0.06	60	28
0.08	54	20
0.09	35	16

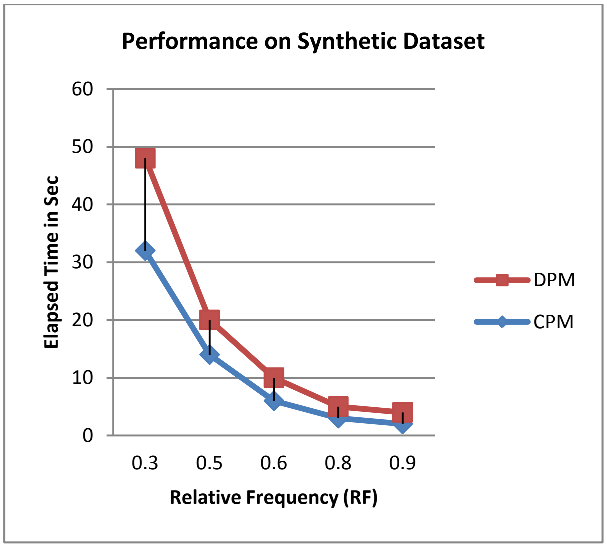


Fig. 4. Performance on Synthetic Dataset (sec)

Often with frequent pattern mining algorithms, it is observed that the performance is poor when minsup is small than the large because with smaller minsup the algorithms will find more frequent items and the searching time will be increased dramatically. However in DPM, when the minsup decrease, the level of data matrix will be decreased and searching time also minimized. Therefore when the minsup is small, DPM has a good mining efficiency.

From the Figure 4and Figure 5 it is observed that the difference of efficiency of DPM with Colossal Pattern Miner (CPM) is very much when the minsup is small.

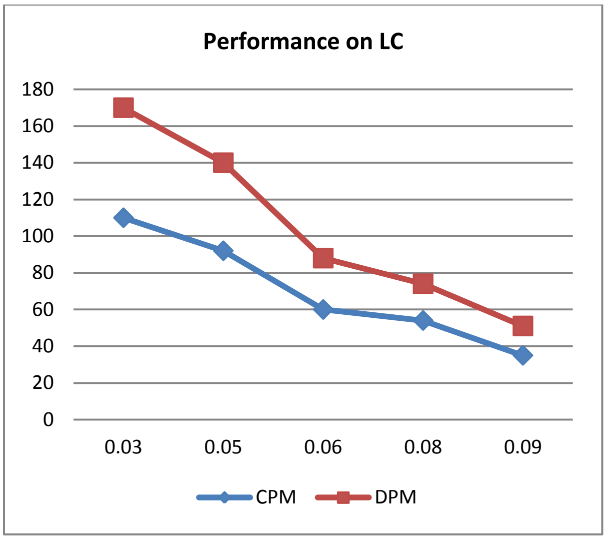


Fig. 5. Performance on LC (sec)

The number of interesting Colossal Patterns and Association Rules pair of 3 genes generated with varying Relative Frequency (RF) and confidence values on SAGE dataset are shown in the Table 10. As the Relative Frequency (RF) values increases the number of rules and elapsed time are decreasing.

Table 10: Colossal Patterns and Association Rules generated on SAGE Dataset

Relative Frequency(RF)	Colossal Patterns	Association Rules Pair of 3 genes
0.3	79945	96082
0.5	9689	83945
0.6	8673	10689
0.8	909	9673
0.9	146	909

5. CONCLISION

According to Doubleton Pattern Mining "A doubleton pattern is frequent only it all items in the set are also frequent" This property leads to a generation large number of redundant patterns of small and midsized patterns. Doubleton Pattern Mining is a relatively new approach applied on high dimensional datasets which enhance the process time than Frequent Pattern Mining.

In this paper, a new algorithm DPM is presented to mine long biological datasets. In our algorithm we iteratively constructed a data matrix D-Struct, a bitwise representation of the dataset for effective discovery of doubleton patterns. We used a vector column intersection bitwise operation to facilitate the algorithm to extract colossal pattern. We also used Triple count array along with D-Struct to improve the efficiency of the mining process when memory constraints are present. The empirical study shows that our algorithm has attained good mining efficiencies under different settings. Furthermore, our performance analysis demonstrate that this algorithm also attains highest Accuracy and F-measure in discovering Colossal pattern and significantly the best compared to formerly developed algorithms.

6. REFERENCES

[1] R. Agrawal and R. Srikant. “Fast algorithms for mining association rules” Proceedings of Internatinonal Conference on VLDB, pp 487–499 in 1994.

[2] H. Mannila, H. Toivonen, and A. I. “Verkamo. Efficient algorithms for discovering association rules”. Proceedings of Internatinonal Conference on KDD in 1994.

[3] H.Manila,H.Toivonen, and A.I. verkamo”Discovery of frequent episodes in event sequences” , journal of Data Mining and Knowledge Discovery. pp 259-289. 1997.

[4] R. Srikant and R. Agrawal. “Mining sequential patterns: Generalizations and performance improvements” Proceedings of International Conference on EDB’, pages 3–17 in 1996.

[5] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen,U. Dayal, and M.-C. Hsu. “PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth” . Proceedings of International Conference on Data Engineering (ICDE) in 2001,

[6] R. J. Bayardo ”Efficiently mining long patterns from databases” . Proceedings of International conference ACM SIGMOD, pages 85–93 in 1998.

[7] J. Pei, J. Han, and R. Mao. “CLOSET: An efficient algorithm for mining frequent closed itemsets” Proceedings of International conference ACM SIGMOD and International Workshop Data Mining and Knowledge Discovery (DMKD’00), pages 11–20 in 2000.

[8] M. Zaki. “Generating non-redundant association rules” . Journal of Knowledge Discovery in Databases pages 34–43 in 2000.

[9] Y. Cheng and G. M. Church. “Biclustering of expression data” . Proceedings of International Conference on Intelligent Systems for Mocular Biology, 2000.

[10] J. Yang, H. Wang, W. Wang, and P. S. Yu. “ Enhanced Biclustering on Gene Expression data” . Proceedings of 3rd IEEE International Symposium on Bioinformatics and Bioengineering (BIBE), Washington DC, Mar. 2003

[11] G. Cong, A. K. H. Tung, X. Xu, F. Pan, and J. Yang. “FARMER: Finding interesting rule groups in microarray datasets” . Proceedings of 23rd ACM International Conference on Management of Data, 2004.

[12] C. Creighton and S. Hanash. “Mining gene expression databases for association rules” Journal of Bioinformatics, volume 19, 2003.

[13] Z. Zhang, A. Teo, B. Ooi, and K.-L. Tan. “Mining deterministic biclusters in gene expression data” . In 4th Symposium on Bioinformatics and Bioengineering, 2004.

[14] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. “Discovering frequent closed itemsets for association rules” . Proceedings of International Conference on Database Theory (ICDT), 1999.

[15] M. J. Zaki and C. Hsiao. “ CHARM: An efficient algorithm for closed association rule mining” . Procedings of International Conference SIAM on Data Mining (SDM), 2002.

[16] F. Pan, G. Cong, A. K. H. Tung, J. Yang, and M. J. Zaki. “CARPENTER: Finding closed patterns in long biological datasets” . Procedings of Internationa Conference ACM SIGKDD and International Conference on Knowledge Discovery and Data Mining (KDD), 2003.

[17] D.Madhavi, M.Shashi. “An Efficient Approach to Colossal Pattern Mining” International Journal of Computer Science and Network Security(IJCSNS) Volume 6 304-312. 2010.

[18] UCI machine learning data sets http://archive.ics.uci.edu/ml/datasets/.