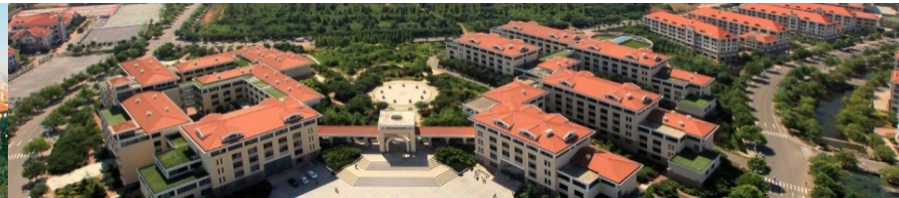
学术论文写作

论文的构成与基本表达

2020 / 09



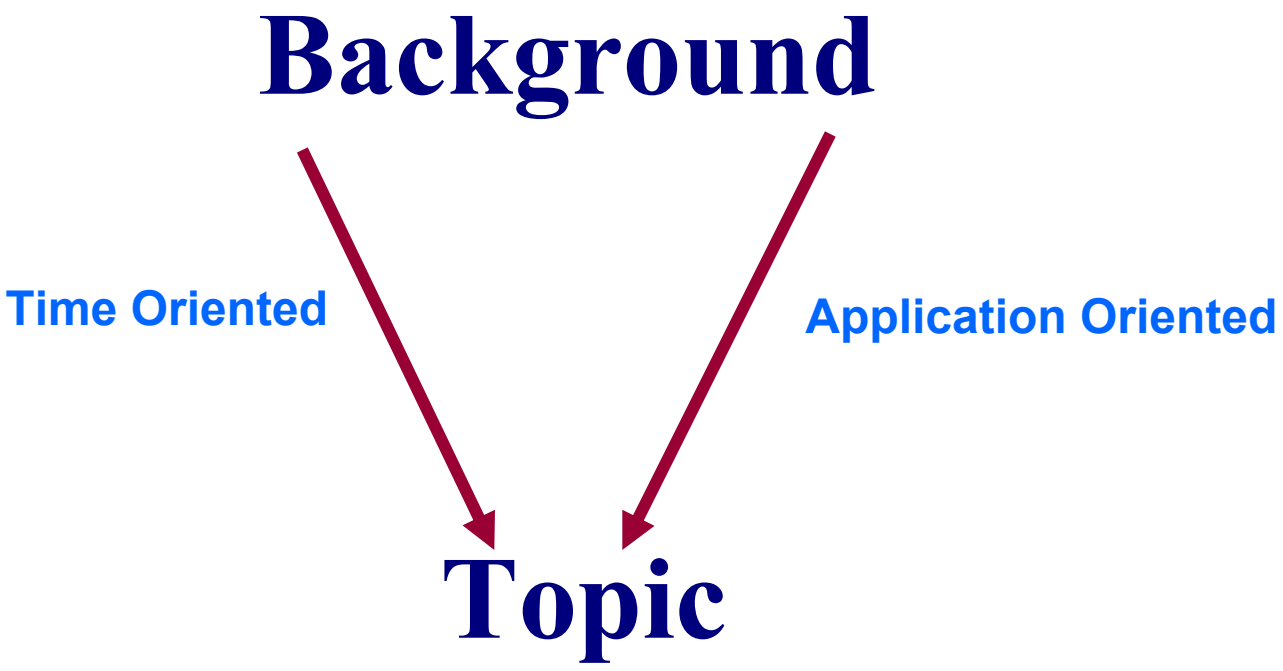
Five Main Sections

- **Abstract, Keywords**
- **Introduction**
- **Related Work (Literature Review)**
- ***Preliminary***
- **Algorithm (Method)**
- **Experimental Results**
- **Conclusion & Future Work**

Introduction

- **Background→ Topic (1-2 paragraphs)**
- **Literature Review→ Motivation (1 paragraph)**
- **Algorithm Overview (1 paragraph)**
- **Contributions (1 paragraph)**
- ***Experiment Overview***
- **Roadmap (1 paragraph)**

Background → Topic



- Concept, Characteristics, Application, Importance, and etc.

Background → Topic

Clustering is a widely used technique in identifying co-expressed gene patterns from microarray data. Traditional "global" clustering algorithms either group genes according to their expression under all conditions or group conditions based on the expression of all genes. However, in cellular processes, subsets of genes are usually co-expressed only under certain experimental conditions, but behave almost independently under other conditions. Hence, discovering such local co-expressed patterns may be the key to uncovering many genetic pathways that are not apparent otherwise. Therefore, researchers are motivated to extract a subset of genes whose expression levels rise and fall coherently under a subset of conditions, that is, they exhibit fluctuation of a similar shape when conditions change, which is called "consistent trends". Such patterns are referred to as "biclusters". As highlighted in [9], discovery of biclusters is essential in revealing the significant connections in gene regulatory networks.

Background → Topic

FREQUENT pattern (FP) mining [1], [5], [13], [9] is a fundamental step to several essential data mining tasks, including association analysis, correlation analysis, causality analysis, association-based classification, and clustering. However, the number of FPs can be too large for them to be of practical use, especially for dense data sets and/or when low support thresholds are used. To reduce the number of FPs, frequent closed pattern (FCP) mining has been introduced and successfully adopted for data analysis in many domains. In particular, FCPs mined from gene expression data have been used to build association rules to uncover gene regulation networks [3], [16] and to build classifiers for diagnosis [17].

Introduction

- **Background→ Topic (1-2 paragraphs)**
- **Literature Review→ Motivation (1 paragraph)**
- **Algorithm Overview (1 paragraph)**
- **Contributions (1 paragraph)**
- ***Experiment Overview***
- **Roadmap (1 paragraph)**

Literature Review → Motivation

- **Notable Method Summarization (focus on nearest neighbors)**
- **Comments (weakness)**
- **Motivation (against the weakness)**

While existing algorithms can generate biclusters with similar trends, they are limited in several ways. First, these schemes typically employ a similarity score (e.g., *MSR and* pScore) to determine the quality of biclusters. However, most of these similarity scores do not adequately capture the trend consistency of biclusters. In other words, patterns with higher *MSR* score or *pScore* could have more consistent trends than those with lower *MSR* score or *pScore*. Second, these algorithms usually generate biclusters based on selected "seeds" that cover only a small part of the whole dataset. As such, interesting patterns may be missed and result in loss of relevant information. Third, the seed improvement process follows the hill-climbing paradigm and can involve significant amount of computation. This often results in a long processing time before any acceptable result is returned to the user. Finally, very few inter-bicluster relationships are delivered by previous framework (e.g., which biclusters are closer to each other, which biclusters are remote from each other, and which bicluster is superset/subset of another bicluster). A biclustering algorithm that (bi)clusters a gene expression dataset and provides a graphical representation of the inter-bicluster relationships would be more favored by the biologists. To the best of our knowledge, no previous work has established a clear relationship between biclusters.

Literature Review → Motivation

Some notable FCP mining schemes include CLOSET+ [10], CHARM [12], CARPENTER [6], REPT [3], and D-miner [2]. Although these algorithms have been shown to perform well in their respective context, it turns out that they are not suited for applications that involve data sets with very high density, where nearly 50 percent or more of the cells contain ones (as we shall see, all the real data sets that we used in the performance study are dense): they are either very inefficient (that is, take hours or even days to produce patterns, even with *high* minimum support threshold) or may even fail (that is, run out of memory). In addition, these methods are nonprogressive; that is, the users are swarmed with *all* the answer patterns (after a very long wait) at a single time when the algorithm completes.

Introduction

- **Background→ Topic (1-2 paragraphs)**
- **Literature Review→ Motivation (1 paragraph)**
- **Algorithm Overview (1 paragraph)**
- **Contributions (1 paragraph)**
- **Experiment Overview**
- **Roadmap (1 paragraph)**

Algorithm Overview & Contribution

- **Method Definition**
- **Principle Description (How to make it better?)**
- **Contribution**

Algorithm Overview & Contribution

In this paper, we propose an efficient top-down hierarchical biclustering algorithm called Quick Hierarchical Biclustering (QHB), to mine biclusters with consistent trends. QHB continuously partitions the whole dataset into subsets such that genes with more consistent trends during condition transitions are grouped together while genes with inconsistent trends are set apart. To measure the trend consistency of a bicluster, we define a new score that reflects the similarity of fluctuating degrees in the changing trends. Compared with previous biclustering models, we have made five main contributions:

Efficiency. QHB adopts a partition based refinement that can simultaneously processed several rows/columns. This is much more efficient than existing techniques.

Inter-bicluster Relationships. QHB provides a very clear hierarchical inter-bicluster relationships. Such graphical representation of the relationships among biclusters provides more valuable knowledge to the biologists.

Algorithm Overview & Contribution

In this paper, we tackle the problem of mining FCC from 3D datasets. Our contributions are as follows. First, we introduce the notion of FCC and formally define it. Second, we propose two approaches to mine FCCs. The first approach is a three-phase framework, called Representative Slice Mining algorithm (RSM) that exploits 2D FCP mining algorithms to mine FCCs. The basic idea is to transform a 3D dataset into a set of 2D datasets, mine the 2D datasets using an existing 2D FCP mining algorithm, and then prune away any frequent cubes that are not closed. The second method is a novel scheme, called CubeMiner, that operates directly on the 3D dataset to mine FCCs. Third, we also show how RSM and CubeMiner can be easily extended to exploit parallelism. Finally, we have implemented RSM and CubeMiner, and conducted experiments on both real and synthetic datasets. To our knowledge, there has been no prior work that mine FCCs.

Contributions

- New Concept
- New Model
- New Algorithm (More Efficient, Less Memory, Parallelism and etc.)
- New Result (Significant in Application Domain)

Introduction

- **Background→ Topic (1-2 paragraphs)**
- **Literature Review→ Motivation (1 paragraph)**
- **Algorithm Overview (1 paragraph)**
- **Contributions (1 paragraph)**
- ***Experiment Overview***
- **Roadmap (1 paragraph)**

Experiment Overview

- **Implementation**
- **Data**
- **Result**

We have implemented C-Miner and B-Miner and experimented with synthetic data sets and three real microarray data sets. Our results show that our C-Miner and B-Miner are superior to CLOSET+, REPT, and D-Miner on dense data sets. We also report results on parallel versions of our proposed schemes.

Introduction

- **Background→ Topic (1-2 paragraphs)**
- **Literature Review→ Motivation (1 paragraph)**
- **Algorithm Overview (1 paragraph)**
- **Contributions (1 paragraph)**
- ***Experiment Overview***
- **Roadmap (1 paragraph)**

Roadmap

The rest of this paper is organized as follows: In the next section, we summarize some previous works. In Section 3, we present some preliminaries. Section 4 presents the proposed C-Miner and B-Miner algorithms. In Section 5, we report experimental results obtained from comparing C-Miner and B-Miner against some existing schemes. Finally, we conclude in Section 6.

The rest of this paper is organized as follows. Section 2 reviews some related works. In Section 3, we formally define the FCC mining problem. Section 4 presents the proposed RSM framework, while Section 5 presents the proposed CubeMiner algorithm. In Section 6, we show how RSM and CubeMiner can be extended to exploit parallelism. Section 7 reports experimental results on RSM and CubeMiner, and finally, we conclude in Section 8.

Introduction

- **Background→ Topic (1-2 paragraphs)**
- **Literature Review→ Motivation (1 paragraph)**
- **Algorithm Overview (1 paragraph)**
- **Contributions (1 paragraph)**
- ***Experiment Overview***
- **Roadmap (1 paragraph)**

Five Main Sections

- **Abstract, Keywords**
- **Introduction**
- **Related Work (Literature Review)**
- ***Preliminary***
- **Algorithm (Method)**
- **Experimental Results**
- **Conclusion & Future Work**

Related Work

- Detail some notable works
- Give comments individually or by group
- Summarize irrelevant works in a few words
- Head and tail

In time or advancing order

Put to the bottom the methods to be compared

Related Work

Traditionally, frequent pattern mining algorithms [1, 18, 14] typically generate a large number of patterns and many of them are redundant. To reduce the number of frequent patterns, frequent closed pattern (FCP) mining algorithms have been proposed. A-close [10] uses a breadth-first search to find FCPs. CLOSET [11] and CLOSET+ [16] adopt a depth-first, feature enumeration strategy. CLOSET uses a frequent pattern tree for a compressed representation of the dataset. CLOSET+, an enhanced version of CLOSET, uses a hybrid tree-projection method to build conditional projected table in two different ways according to the density of the dataset. Both MAFIA [4] and CHARM [17] use a vertical representation of the datasets. MAFIA adopts a compressed vertical bitmap structure while CHARM enumerates closed itemsets using a dual itemset-tidset search tree and adopts the *Diffset* technique to reduce the size of the intermediate tidsets. Since these methods adopt a feature enumeration strategy, they cannot efficiently handle datasets with a large number of features (columns).

Related Work

More recently, several schemes have been designed to handle "large columns small rows" datasets. In [8], the scheme CARPENTER combines depth-first, row enumeration strategy with some efficient search pruning techniques. In [9], COBBLER dynamically switches between feature enumeration and row enumeration depending on the data characteristic in the process of mining. Both schemes, however, cannot handle dense datasets. In [3], D-miner was proposed to identify closed sets of attributes (or items) for dense and highly-correlated boolean contexts. D-miner generates and employs a set of cutters (containing "0" information) to divide the whole dataset into small dense spaces.

Although the above-mentioned algorithms perform well in their respective application domains in 2D datasets, they cannot mine FCCs in 3D context.

Related Work (supplement)

Add the head

- There are plenty of covert channel methods in literature. Here, we will only review some notable work due to space limitation.
- Now (Here/ In this section), we will review some notable methods (previous work)……

Related Work (supplement)

- **Add the tail**

patterns as a result of running out of memory. Since CLOSET+ [10], REPT [3], and D-Miner [2] represent the state-of-the-art efficient FCP mining algorithms for relatively dense microarray data, we conduct experiments to compare our proposed schemes against them.

Hence, we are motivated to……

Five Main Sections

- **Abstract, Keywords**
- **Introduction**
- **Related Work (Literature Review)**
- ***Preliminary***
- **Algorithm (Method)**
- **Experimental Results**
- **Conclusion & Future Work**

Preliminary

- Applying to paper with lots of definitions
- Three Components:
  - Overview
  - Definitions
  - Problem Definition

We shall first define some notations that we will be using throughout this paper, and then give the problem description.

Let $R = \{r_1, r_2, \ldots, r_n\}$ denote a set of rows, $C = \{c_1, c_2, \ldots, c_m\}$ denote a set of columns, and $H = \{h_1, h_2, \ldots, h_l\}$ denote a set of heights. Then a three-dimension dataset can be represented by a $l \times n \times m$ binary matrix $O = H \times R \times C = \{O_{k,i,j}\}$ with $k \in [1, l]$, $i \in [1, n]$ and $j \in [1, m]$. Each cell o_{kij} corresponds to the relationship among height h_k, row r_i, and column c_j. The value true (i.e., "1") denotes the relationship that any two dimensions are "simultaneously contained (S-contained)" in the third one.

Definition 3.1 Height Support Set and H-Support: *Given a set of rows $R' \subseteq R$ and a set of columns $C' \subseteq C$, the maximal set of heights that simultaneously contain R' and C' is defined as the Height Support Set $H(R' \times C') \subseteq H$. The number of heights in $H(R' \times C')$ is defined as the H-Support of $(R' \times C')$, denoted as $|H(R' \times C')|$.* For example, in Table 1, let $R' = \{r_1, r_2\}$ and $C' = \{c_1, c_2, c_3\}$, then $H(R' \times C') = \{h_1, h_3\}$ since both h_1 and h_3 simultaneously contain $\{r_1, r_2\}$ and $\{c_1, c_2, c_3\}$, and no other heights contain them simultaneously.

Problem Definition: Given a three-dimension dataset O, our problem is to discover all frequent closed cubes with respect to the user support thresholds *minH*, *minR*, and *minC*.

Five Main Sections

- **Abstract, Keywords**
- **Introduction**
- **Related Work (Literature Review)**
- ***Preliminary***
- **Algorithm (Method)**
- **Experimental Results**
- **Conclusion & Future Work**

Algorithm (Method)

- Overview
- Preliminaries (Definitions, Model Description)
- Put the whole idea in Phases or Steps
- Emphases which phase/step/part makes the algorithm outstanding
- Use running examples, figures, tables, pseudo-code.....
- Supplementary (limitations, suggestions, future work…)

Algorithm (Overview)

In this section, we first present the basic framework for compressed hierarchical FCP mining. We then present the two schemes, C-Miner and B-Miner, that are based on the framework. Finally, we show how the framework can be easily adapted for parallel FCP mining.

In this section, we present the proposed QHB framework. The QHB algorithm comprises 3 phases. In the first phase, the original matrix is transformed into a binary matrix that captures the changing trend of the gene expression value between each consecutive conditions. This trend could either be a rising trend, a falling trend or one that is considered to have no significant change. In the second phase, an iterative partitioning procedure is applied to the

Algorithm (Main Content)

Phase 1

- Step 1
- Step 2

Phase 2

- In the 1st step
- In the 2nd step

Phase 3

……

Algorithm (Try to Make It Clear)

Alice and Bob would take the following five steps to transmit the secret message:

Step 1: Alice and Bob communicate normally. They both record the message lengths sent by Alice, and make the record as *Reference*.

Step 2: Alice and Bob select a length *l* from the *Reference* by the same random algorithm.

Step 3: In the *i*th sending, Alice sends to Bob a message of length *lnext* =*l*+*SUMi*. The *Reference* is updated by appending *lnext*.

Step 4: Bob decodes the *i*th message into *Wi* by subtracting *l* from *lnext*.

In our method, the Step 2 and 3 ensure that each sending message would learn from the normal network traffic (*Reference*) so that messages from our channel have similar length distribution as normal network messages. To better resist the detection, our scheme could be further enhanced through the way that Alice periodically sends to Bob redundant normal network messages during the transmission.

Five Main Sections

- **Abstract, Keywords**
- **Introduction**
- **Related Work (Literature Review)**
- ***Preliminary***
- **Algorithm (Method)**
- **Experimental Results**
- **Conclusion & Future Work**

Experimental Results

- Overview: tool, data, methods to compare
- Data Description
- Data Preprocessing
- Compare against other algorithms (methods)
- Parameter Studies (Optimization)

Experimental Results

Experimenting on the Clartnet dataset [9], we compared our scheme with Girling's[3] and Yao's[8] models, and estimated the upper bound of our model's bandwidth.

4.1 Dataset

Clarknet dataset [5] contains the logs of 1.6 million HTTP queries from two weeks. We take the 7239 queries from Client 'piweba5y.prodigy.com' as the messages for our experiments. In each query, the average size of transmitted message is 489 bytes (39 bytes message + 450 bytes header).

Experimental Results

4.3 Efficiency

In this group of experiments, we set $maxMFD = 0.15$ and vary the $minGene$ and $minCon$ thresholds and compare the execution time of QHB against DBF. The execution time for QHB includes processing all seeds while the execution time for DBF only includes processing the top 100 seeds ranked by the algorithm. However, compared with DBF, QHB is still much more efficient - while DBF takes at least 1000 sec in all our experiments, QHB is no more than 100 sec (figure not shown due to space constraint). This is because QHB simultaneously groups several genes and conditions at the same time and the grouping (submatrix partition) process is oriented by bins. This makes the whole processing very efficient. However, while refining the seeds, DBF tends to randomly try the row/column one by one to decide which row/column to add. This process is

Five Main Sections

- **Abstract, Keywords**
- **Introduction**
- **Related Work (Literature Review)**
- ***Preliminary***
- **Algorithm (Method)**
- **Experimental Results**
- **Conclusion & Future Work**

Conclusion & Future Work

- **Summarize what we did in the paper**
  - **Method**
  - **Experimental Results**
  - **Benefits (Application)**
- **Future Work**

Conclusion & Future Work

CONCLUSION

In this paper, we revisit the problem of analyzing gene expression data for time-lagged gene co-regulation relationships. We have presented a localized algorithm to identify the time-lagged gene clusters based on the concept of q-clusters. Genes with a similar pattern over a subset of q consecutive time points (conditions) are grouped into the same q-cluster. In this way, we can easily determine the co-regulations of genes within each q-cluster and between q-clusters. We have experimented on a real time-series gene expression dataset and compared our method and results with the Event Method. Our study shows that our approach is efficient at detecting both activation and inhibition time-lagged co-regulations, and our results can draw relationships between both genes and gene clusters and provide more detailed information. We believe that our approach delivers valuable information and provides an excellent tool that facilitates more detailed exploration for gene network research.

cantly. As future research, we plan to study 3D association rule analysis and classifier based on frequent closed cubes.

Five Main Sections

- **Abstract, Keywords**
- **Introduction**
- **Related Work (Literature Review)**
- ***Preliminary***
- **Algorithm (Method)**
- **Experimental Results**
- **Conclusion & Future Work**

Abstract

- **Motivation**
  - **Topic Importance**
  - **Literature Review: However, existing work…**
- **Method: In this paper, we….**
- **Experimental Results**
- ***Application***

Abstract

Mining biclusters that exhibit both consistent trends and trends with similar degrees of fluctuations is vital to bioinformatics research. However, existing biclustering methods are not very efficient and effective at mining such biclusters. Moreover, few inter-bicluster relationships are delivered to biologists. In this paper, we introduce a quick hierarchical biclustering algorithm (QHB) to efficiently mine biclusters with both consistent trends and trends with similar degrees of fluctuations. Our QHB produces not only biclusters but also a hierarchical graph of inter-bicluster relationships. We experimented with the Yeast dataset and compared QHB against an existing biclustering scheme, DBF. Our results show that QHB identifies biclusters with better quality. In addition, QHB shows the relationships among biclusters. Moreover, compared with DBF, QHB is much more efficient and offers users a progressive way of bicluster exploration.

Abstract VS Conclusion

ABSTRACT

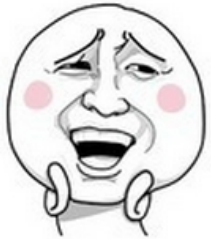
Motivation: Analysis of gene expression data can provide insights into the time-lagged co-regulation of genes/gene clusters. However, existing methods such as the Event Method and the Edge Detection Method are inefficient as they compare only two genes at a time. More importantly, they neglect some important information due to their scoring criterian. In this paper, we propose an efficient algorithm to identify time-lagged co-regulated gene clusters. The algorithm facilitates localized comparison and processes several genes simultaneously to generate detailed and complete time-lagged information for genes/gene clusters.

Results: We experimented with the time-series Yeast gene dataset and compared our algorithm with the Event Method. Our results show that our algorithm is not only efficient, but also delivers more reliable and detailed information on time-lagged co-regulation between genes/gene clusters.

CONCLUSION

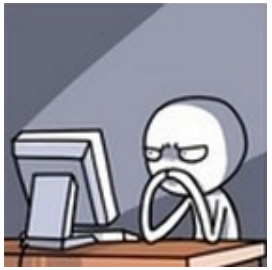
In this paper, we revisit the problem of analyzing gene expression data for time-lagged gene co-regulation relationships. We have presented a localized algorithm to identify the time-lagged gene clusters based on the concept of q-clusters. Genes with a similar pattern over a subset of q consecutive time points (conditions) are grouped into the same q-cluster. In this way, we can easily determine the co-regulations of genes within each q-cluster and between q-clusters. We have experimented on a real time-series gene expression dataset and compared our method and results with the Event Method. Our study shows that our approach is efficient at detecting both activation and inhibition time-lagged co-regulations, and our results can draw relationships between both genes and gene clusters and provide more detailed information. We believe that our approach delivers valuable information and provides an excellent tool that facilitates more detailed exploration for gene network research.

However … … …

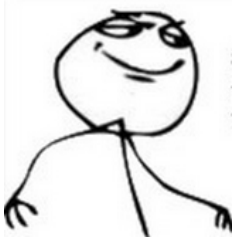


Amazing!!! 我要开始写论文了

构思ing … 写一篇高大上的论文



创作中，请勿打扰 … …



我好机智，这么快就写好了

和小伙伴聊聊天，刷刷朋友圈，上上网

论文好像忘记保存了 … …

一切都得靠自己去探索

在不断的reject中成长

预祝同学们顺利的开启科研之旅！