Gene 518 (2013) 59–69

Contents lists available at SciVerse ScienceDirect

Gene

journal homepage: www.elsevier.com/locate/gene

Efficient mining differential co-expression biclusters in microarray datasets

Miao Wang [a], Xuequn Shang [a,*], Xiaoyuan Li [a], Wenbin Liu [b], Zhanhuai Li [a]

[a] School of Computer Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China
[b] Department of Physics and Electronic information engineering, Wenzhou University, Wenzhou 325035, China

ARTICLE INFO

Article history:
Accepted 27 November 2012
Available online 28 December 2012

Keywords:
Differential co-expression
Bicluster
Microarray
Gene expression

ABSTRACT

Background: Biclustering algorithm can find a number of co-expressed genes under a set of experimental conditions. Recently, differential co-expression bicluster mining has been used to infer the reasonable patterns in two microarray datasets, such as, normal and cancer cells.
Methods: In this paper, we propose an algorithm, *DECluster*, to mine Differential co-Expression biCluster in two discretized microarray datasets. Firstly, *DECluster* produces the differential co-expressed genes from each pair of samples in two microarray datasets, and constructs a differential weighted undirected sample–sample relational graph. Secondly, the differential biclusters are generated in the above differential weighted undirected sample–sample relational graph. In order to mine maximal differential co-expression biclusters efficiently, we design several pruning techniques for generating maximal biclusters without candidate maintenance.
Results: The experimental results show that our algorithm is more efficient than existing methods. The performance of *DECluster* is evaluated by empirical *p*-value and gene ontology, the results show that our algorithm can find more statistically significant and biological differential co-expression biclusters than other algorithms.
Conclusions: Our proposed algorithm can find more statistically significant and biological biclusters in two microarray datasets than the other two algorithms.

1. Background

DNA microarray techniques have generated a great number of gene expression datasets. Biclustering (Akdes and Martin, 2011) is one of the popular methods for microarray dataset analysis. It can find a number of co-expressed genes under a subset of experimental conditions. There have been many biclustering methods. Such as, the algorithm of Cheng and Church (2000) can find constant value and constant row or column biclusters, Subramanian et al. (2005) focuses on discovering constant value biclusters, coherent evolution biclusters can be found by Ben-Dor et al. (2003), Zhao and Zaki (2005) use a weighted multi-graph to mine scaling biclusters.

Above biclustering methods for gene expression data analysis have focused on the discovery of co-expressed genes under a subset of samples. However, it cannot detect differential co-expression biclusters which show highly corrected co-expression in one dataset but not in another. Mining differential co-expression biclusters is more useful for disease detection. Biologically speaking, using differential co-expression biclusters can indicate the wrong regulation of a pathway (Kostka and Spang, 2004).

The existing differential bicluster mining methods can be classified into two groups. One is to construct a difference matrix to mine discriminative biclusters. Southworth et al. (2009) developed a methodology for differential co-expression network analysis for the comparison of gene co-expression on a global scale. *FDCluster* (Wang et al., 2010) produces discriminative biclusters in differential dataset, which is the difference between *Class A* and *Class B*. The other method for mining differential biclusters is to mine differential co-expression biclusters. Differential co-expression bicluster method aims to find gene sets which are co-expressed under a subset of conditions in one class but not in another class. The recent proposed *DiBiCLUS* (Odibat et al., 2010) algorithm aims to extract differential biclusters from the two gene expression datasets. The genes in the differential biclusters have co-expressed in one class but not in another class. *DiBiCLUS* identifies the differential pairs of genes. Then the differential biclusters are generated using clustering method. However, there exist some drawbacks of *DiBiCLUS*. Firstly, *DiBiCLUS* can only handle one relation between two genes, which may omit some biological information. For example, G_1G_2 is a positive co-expression under samples S_1 and S_2, G_1G_2 is a negative co-expression under samples S_3, S_4 and S_5. According to the procedure of *DiBiCLUS*, G_1G_2 is only a negative co-expression. The reason is that the maximum of the above values is considered the final value for the relation between two genes. Secondly, *DiBiCLUS* cannot be used for mining differential

Abbreviations: SDC, Subspace differential co-expression; DC, Differential co-expression; DWUR graph, Differential weighted undirected sample–sample relational graph; WUR graph, Weighted undirected sample–sample relational graph; GO, Gene ontology.

* Corresponding author.
E-mail addresses: riyushui@gmail.com (M. Wang), shang@nwpu.edu.cn (X. Shang), lixiaoyuan88@gmail.com (X. Li), wbliu6910@126.com (W. Liu), lizhh@nwpu.edu.cn (Z. Li).

0378-1119/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.gene.2012.11.085

biclusters in real-valued gene expression datasets, which may lose some interesting biological results. Thirdly, *DiBiCLUS* needs to be double mining differential co-expression biclusters. One is to produce differential co-expression biclusters from *Class A* to *Class B*. The other is from *Class B* to *Class A*. Such double checking procedure influences the mining efficiency. Finally, *DiBiCLUS* needs to maintain the whole cluster in memory, when the differential gene pairs are huge, the memory may crash.

SDC algorithm (Fang et al., 2010) (in order to distinguish from SDC which is the abbreviation of Subspace Differential Co-expression, we use *SDC* to express SDC mining algorithm proposed in Fang et al.'s algorithm) is another method for mining subspace differential co-expression (SDC) patterns. It can also be used for discovering differential co-expression biclusters. *SDC* algorithm aims to infer the patterns which are co-expressed over a large percent of the conditions in one microarray dataset, but in a much smaller percent of conditions in another microarray dataset. However, *SDC* algorithm has some limitations for mining differential patterns. Firstly, the a-priori framework limits its computing efficiency and scalability. Secondly, *SDC* algorithm also needs to be double mining SDC patterns. One is to mine SDC patterns from *Class A* to *Class B*, the other is to infer from *Class B* to *Class A*. Finally, based on the definition of subspace differential co-expression, *SDC* algorithm cannot find some interesting differential co-expression biclusters. The overview of above double checking method to mine maximal differential co-expression biclusters is shown in Fig. 1. Another recently proposed differentially expressed biclustering algorithm called *DeBi* (Akdes and Martin, 2011) uses frequent pattern mining method to discover maximum size homogeneous biclusters in which all genes are co-expressed under a subset of samples. In fact, *DeBi* aims to find biclusters in one microarray dataset, which is not the same as our proposed differential co-expression bicluster that is produced from two microarray datasets.

In hopes of overcoming the limitations of existing differential biclustering methods, in this paper, we propose an efficient algorithm, *DECluster*, for mining Differential co-Expression biClusters in two discretized microarray datasets. Unlike the traditional double checking method, our approach can produce differential biclusters at one time. Firstly, we mine the differential co-expressed genes in each pair of samples in two microarray datasets, and construct a differential weighted undirected sample–sample relational graph. Secondly, the differential biclusters are generated in the above differential weighted undirected sample–sample relational graph. In order to mine maximal differential co-expression biclusters efficiently, we designed several pruning techniques for generating maximal biclusters without candidate maintenance. The overview of our approach is illustrated in Fig. 2.

The contributions of our *DECluster* framework which distinguish it from existing differential biclustering algorithms are summarized as follows:

1. *DECluster* can identify new types of differential co-expression bicluster which are different from traditional ones. The experimental results show that our method is more effective than others.
2. Instead of using double checking method to mine differential co-expression biclusters, our approach exploits differential co-expression biclusters in the differential weighted undirected sample–sample relational graph.
3. The proposed *DECluster* algorithm can mine maximal differential biclusters without candidate maintenance.

2. Methods

2.1. Preliminaries and definitions

The microarray (also called gene expression dataset) is denoted as $D = S \times G$, where the column S represents the set of different experimental conditions, and the row G represents genes. The element value of D_{ij} is a real value which is the expression level of gene i under condition j. $|D|$ is the total number of experimental conditions in D. A bicluster P is defined as a sub-matrix of D. For simplicity, we denote the gene set of P as $P.Geneset$ and the conditions of P as $P.Sample$. Given two microarray datasets, A and B, where D_1 D and D_2 D. The original microarray data can be discretized each gene expression number into one of the three numbers: 1, -1 and 0, which denotes positive expression, negative expression and non-expression, respectively. The discretized microarray datasets D_1 and D_2 are shown in Tables 1 and 2. The discretization method will be illustrated in Section 3. The three co-expression types of relations between genes G_1 and G_2 in discretized dataset can be respectively defined as follows:

1. G_1 and G_2 are a positive co-expression denoted as (G_1G_2) if $G_1 = 1$ and $G_2 = 1$, or $G_1 = -1$ and $G_2 = -1$.
2. G_1 and G_2 are a negative co-expression denoted as $(G_1 - G_2)$ if $G_1 = 1$ and $G_2 = -1$, or $G_1 = -1$ and $G_2 - 1$.
3. G_1 and G_2 are a non-expression if $G_1 = 1$ and $G_2 = 0$, or $G_1 = -1$ and $G_2 = 0$, or $G_1 = 0$ and $G_2 = 1$, or $G_1 = 0$ and $G_2 = -1$.

Traditional DC bicluster mining methods (Fang et al., 2010; Odibat et al., 2010) aim to find different co-expression types between any two genes under a set of samples. *DiBiCLUS* (Odibat et

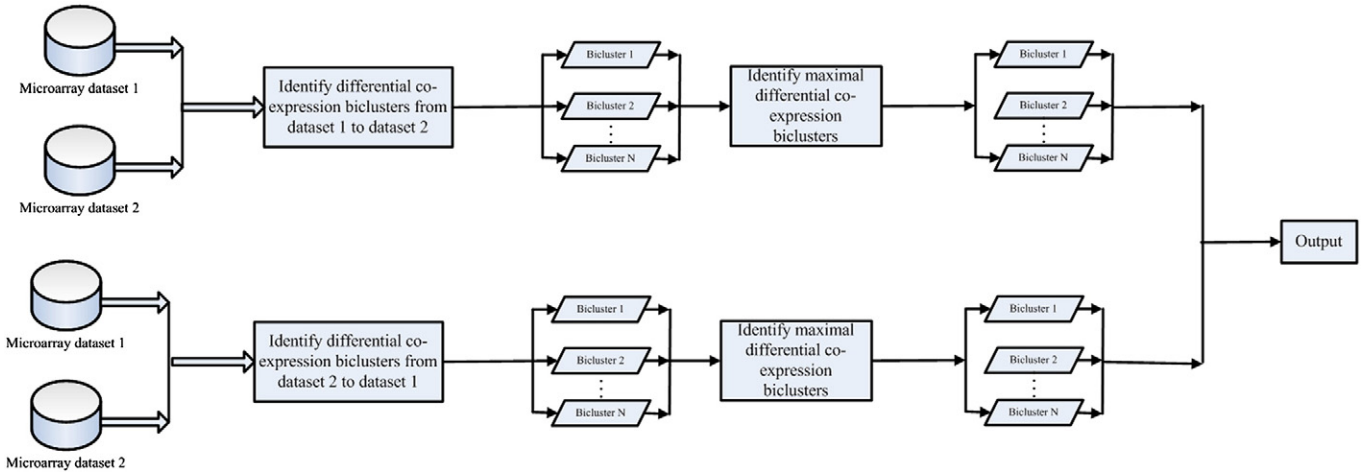

Fig. 1. The overview of traditional double checking method for mining maximal differential biclusters.