基于离散时序基因表达数据的双聚类算法*

许 涛，尚学群，杨蜜静，王 淼
（西北工业大学 计算机学院 计算机软件与理论系，西安 710129）

摘 要：目前应用于基因表达数据上的双聚类算法大多是基于真实数据提出的，因此易受噪声干扰，且这些算法很少考虑样本间的时序性。提出了一种有效的时间点连续的双聚类挖掘算法 DTCB，从离散的时序基因表达数据中挖掘出时间点连续的最大共表达双聚类。该算法使用了一种新的数据离散化方法，同时提出了三种在离散数据集下基因间的共表达关系；为了提高挖掘效率，DTCB 使用了有效的剪枝和输出策略，可以在不产生候选集的情况下一次性挖掘出所有的最大共表达双聚类。通过实验分析，证明 DTCB 具有高效的性能和良好的鲁棒性，且结果具有较好的统计和生物意义。

关键词：时序基因表达数据；双聚类；共表达；时间点连续；离散化
中图分类号：TP311；TP301.6 **文献标志码**：A **文章编号**：1001-3695(2013)12-3551-06
doi:10.3969/j.issn.1001-3695.2013.12.008

Bicluster algorithm on discrete time-series gene expression data

XU Tao, SHANG Xue-qun, YANG Mi-jing, WANG Miao
(*School of Computer Science & Technology, Northwestern Polytechnical University, Xi'an* 710129, *China*)

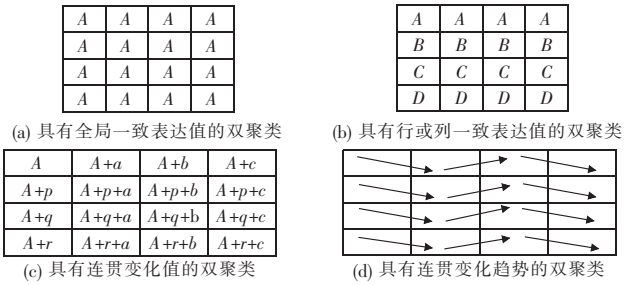
Abstract: At present, the bicluster algorithms applied to the gene expression data were mostly based on real data. Therefore, they were susceptible to noise interference, and these algorithms rarely considered the time sequence between samples. This paper proposed an efficient time-continuous bicluster algorithm DTCB to mine the maximal time-continuous biclusters from the discrete time-series gene expression data. It used a new discretization method on gene expression data and defined three co-expression relations between genes in the discrete dataset. DTCB adopted several pruning and output techniques to improve the efficiency. It could produce maximal co-expression biclusters without candidate maintenance. The experimental results show that DTCB has efficient performance and better robustness. Simultaneously, the results can be of more statistical and biological significance.

Key words: time-series gene expression data; bicluster; co-expression; time-continuous; discretization

0 引言

近年来，基因芯片作为一种新型的高通量检测技术与方法，它可以同时测量成千上万个基因的表达水平，因而已成为“后基因时代”研究基因间相互关系的一个强有力的工具。不过，如何对该技术产生的海量实验数据进行准确而合理的分析，已成为有效应用该项技术的主要瓶颈。因此，基因表达数据的分析已成为当前生物信息学和数据挖掘领域的一个重要研究内容和主要研究方向。

聚类分析[1]是分析基因表达数据的一种常用方法。通过聚类分析，能够有效地预测未知基因或者蛋白质的功能和作用。传统的基因表达数据聚类算法是一种全局意义上的聚类，即在全局数据空间中进行挖掘，其存在着一些缺陷：a）不能实现交叉聚类，从生物学意义上来讲，一个基因可能参与多个生物过程或功能，也就是基因应该存在于多个聚类结果簇中，但传统聚类方法不能实现这种交叉聚类；b）不能实现局部空间聚类，两个基因可能只是在某些条件下具有相似的表达模式，但传统聚类方法不能实现这种局部条件下的聚类；c）不能发现聚类间的关系。因此，双聚类的概念应运而生。双聚类是指同时在基因和实验条件两个维度上对基因表达数据进行聚类，目的是找出哪些基因在哪些实验条件下具有相似的表达水平或者模式。如图 1 所示，根据双聚类中基因表达值的特征，Madeira 等人[2]将双聚类分为以下几种类型：具有全局一致表达值的双聚类（图 1(a)）；具有行或列一致表达值的双聚类（图 1(b)）；具有连贯变化值的双聚类，即行之间（或者列之间）表达值差值为常数（图 1(c)）；具有连贯变化趋势的双聚类（图 1(d)）。



(a) 具有全局一致表达值的双聚类 (b) 具有行或列一致表达值的双聚类
(c) 具有连贯变化值的双聚类 (d) 具有连贯变化趋势的双聚类

图 1 双聚类的类型

随着双聚类挖掘方法研究的不断深入，许多双聚类算法，

收稿日期：2013-04-15；**修回日期**：2013-05-26 **基金项目**：国家“973”计划资助项目(2012CB316203)；国家自然科学基金资助项目(61272121)

作者简介：许涛(1989-)，男，河南信阳人，硕士研究生，主要研究方向为生物信息学、数据挖掘(2008_xutao@163.com)；尚学群(1973-)，女，教授，博士，主要研究方向为数据库技术、数据挖掘、生物信息学等；杨蜜静(1988-)，女，硕士研究生，主要研究方向为生物数据挖掘、属性图的模式研究；王淼(1981-)，男，博士研究生，主要研究方向为数据挖掘、生物信息学.

如 SAMBA[3]、OPSM[4]、xMotifs[5]等被提出,并用来挖掘不同类型的双聚类模式。Cheng 等人[6]提出了一种基于贪心策略的双聚类算法——CC 算法,该算法采用一个较低的均方误差作为双聚类的得分来逐步删除冗余节点,每产生一个双聚类,就用一个随机数来代替其原始值,该迭代过程反复执行。但是,因为随机数的代替使得结果产生很大的随机性,同时其算法效率不高。Zhao 等人[7]提出了一种新的双聚类算法 MicroCluster,该算法适用于挖掘行列之间表达值之间呈比例变化的双聚类。MicroCluster 的核心思想是通过权值图来挖掘双聚类,而不是按照分步骤挖掘再比较的方法。实验结果表明,这种通过权值图来挖掘双聚类的方法使得算法效率有了较大的提高。Pandey 等人[8]提出了一种行常量双聚类算法 RAP。该算法提出了范围支持度这一新颖概念,采用广度优先策略从真实的基因表达数据中直接挖掘出行常量双聚类,但基因扩展的算法效率不高。然而,对于时序基因表达数据来说这些双聚类算法忽略了样本时间点的内在顺序关系,所以基于时间序列基因表达数据的时序性,学者们又提出了一些新方法。Zhang 等人[9]改进了 CC 算法从而提出了 CC-TSB 算法,该算法考虑了时间序列基因表达数据的样本的时序性,挖掘出具有时间连续特征的双聚类。Wang 等人[10]提出了 TD-cluster 算法,该算法也是在时间序列基因表达数据下的挖掘,它定义了一个带时间延迟的同比例的双聚类,使得结果更具有意义。

通过上述分析可以看到,已有方法从基因表达数据中挖掘共表达双聚类存在一定的弊端,为了具有更好的可扩展性、更加有效地进行挖掘,本文提出了一种基于权值图扩展的算法 DTCB(discrete time-continuous co-expression bicluster),从离散的时序基因表达数据中挖掘时间点连续的最大共表达双聚类。该算法的特点有:a)为了从离散的时序基因表达数据中产生共表达基因集,本文定义了三种基因间的共表达关系;b)构造以时间点为顶点、以连续时间点间满足定义的基因集为边的基因共表达权值图;c)使用有效的剪枝和输出策略,并以时间点连续扩展的方式从基因共表达权值图中挖掘出所有满足定义的最大共表达双聚类。本文的主要创新点还包括:

a)提出了一种数据离散化方法,在传统数据离散化方法的基础上,将离散化后的数值从 -1、0、1 扩展成多值,并重新定义了基因之间的共表达关系。

b)在基因共表达权值图上采用了样本扩展的方法来挖掘时间点连续的最大共表达双聚类,提高了算法效率。

c)可以在不产生候选集的情况下一次性挖掘出所有满足定义的最大共表达双聚类,同时使用了有效的剪枝和输出策略。

1 问题描述及相关定义

时间序列基因表达数据通常被定义为一个矩阵 $M = G \times S, G = \{G_1, \cdots, G_n\}, S = \{S_1, \cdots, S_m\}$。其中行集合 G 表示基因,列集合 S 代表不同的实验条件。矩阵中每一个元素 $M_{i,j}$ 是基因 G_i 在实验条件 S_j 下的真实表达水平值。为了提高程序的可扩展性和数据的使用范围,本文采用离散的时序基因表达数据集来进行有效的挖掘。由于基因表达数据存在明显的噪声数据,采用真实值进行挖掘时容易受噪声影响;将真实数据离散化可以降低噪声、提高数据处理效率,而传统的数据离散化方法将时序基因表达数据离散化成 -1、0、1 三个值,使得离散后的数据丢失了一些重要信息。本文在传统的离散化方法上进行扩展,将时序基因表达数据离散化为 $-k/2, -k/2+1, \cdots, 0, 1, \cdots, k/2-1, k/2$ 这样的多值(通常 k 取奇数),相比传统离散化方法的三个值更好地量化了基因表达水平,这样既降低了噪声的影响又减少了信息的丢失,从理论和实验结果上来看都取得了较好的效果。表 1 为一个离散后的时序基因表达数据集($k = 5$)。

表 1 时序基因表达数据集

基因	实验条件					
	S_1	S_2	S_3	S_4	S_5	S_6
G_1	1	0	2	1	0	0
G_2	1	-2	-1	-2	-1	-2
G_3	-1	2	1	2	1	2
G_4	1	-2	0	-2	-1	-2
G_5	2	1	2	1	-2	1
G_6	2	1	2	1	-2	0
G_7	2	1	1	2	0	0
G_8	2	1	2	1	-2	1
G_9	0	1	2	1	-2	1

根据传统的数据离散化方法的共表达定义:1 和 1、-1 和 -1 是正共表达;1 和 -1 是负共表达;数值 0 表示不表达。本文在新的离散化方法上重新定义了基因之间的共表达关系(离散化后的时序基因表达数据矩阵用 D 表示)。

定义 1 共表达关系。G_1 和 G_2 是离散时序基因表达数据矩阵 D 中任意两个基因,在任意实验条件 S 下的表达值分别用 $S.G_1$ 和 $S.G_2$ 表示,G_1 和 G_2 在 S 下共表达关系有三种情况:

a)如果 G_1 和 G_2 的表达值的绝对值相等,且符号相同,即 $|S.G_1| = |S.G_2| = n$ 且 $S.G_1 \times S.G_2 > 0$ $(1 \leqslant n \leqslant k/2)$,那么 G_1 和 G_2 是正共表达。

b)如果 G_1 和 G_2 的表达值的绝对值相等,且符号相反,即 $|S.G_1| = |S.G_2| = n$ 且 $S.G_1 \times S.G_2 < 0 (1 \leqslant n \leqslant k/2)$,那么 G_1 和 G_2 是负共表达。

c)不满足上述两种情况时,G_1 和 G_2 不共表达(其中表达值为 0 时基因不表达)。

定义 2 共表达基因集。G_1、G_2、G_3 是离散时序基因表达数据矩阵 D 中任意三个基因,S 是 D 中的任意一个实验条件,如果基因 G_1、G_2、G_3 的表达值的绝对值相等,即 $|S.G_1| = |S.G_2| = |S.G_3| = n (1 \leqslant n \leqslant k/2)$,那么称 $G_1G_2G_3$ 是 S 下的共表达基因集。为了表明基因间的正负共表达关系,如果基因表达值为负,那么在该基因前加“-”。若 $S.G_2 < 0$,则记 $G_1 - G_2G_3$ 是 S 下的表达值为 $\pm n$ 的共表达基因集。

根据上述离散基因表达数据下基因共表达关系,从表 1 中可得知,在 S_1 下 G_1、G_2、G_4 是正共表达,G_3 和 G_1、G_2、G_4 是负共表达,则在 S_1 下,记 $G_1G_2 - G_3G_4$ 是表达值为 ± 1 的共表达基因集。由此可以得到每个时间点下的共表达基因集集合,然后构造基因共表达权值图,挖掘出所有满足定义的时间点连续的最大共表达双聚类。

定义 3 双聚类定义。在时序基因表达数据中,一个时间点连续的共表达双聚类必须满足以下几个条件:

a)双聚类中的样本集必须是时间点连续的实验条件集合;

b)双聚类中的每个基因在连续实验条件下是呈趋势变化的,即每个基因在相邻的时间点下表达值不同;

c)双聚类中的基因集在任意实验条件下都满足定义 2,即在任意时间点下都是共表达基因集;

d)双聚类中的基因个数和时间点个数分别满足设定的

minGene 和 minSample 阈值。

根据上述双聚类定义,当双聚类中所有的基因呈一致趋势变化时,将它称为正共表达双聚类;当双聚类中存在某些基因呈相反趋势变化时,称它为负共表达双聚类。

定义 4　最大双聚类。设定 E 是 D 中所有满足定义的双聚类的集合,其中 $N=P\times Q\subseteq E$。如果 N 是最大双聚类,则它满足条件:不存在任意双聚类 $M=X\times Y$,使得 $X\subseteq P$ 且 $Y\subseteq Q$。

2　DTCB 算法

2.1　构造基因共表达权值图

在基因表达数据中,基因数量要远远多于样本数量,如前文分析,直接使用基因扩展方式挖掘双聚类的效率比较低下。因此,本文首先构造出时间点连续的基因共表达权值图,里面包含了每对时间点的共表达基因集信息,然后使用样本扩展的方式来挖掘满足定义的共表达双聚类。

本文构造的是以时间点为顶点、以相邻时间点之间满足定义的共表达基因集为边的时间点连续的基因共表达权值图。a)通过定义 1 和 2 产生出每个时间点下的满足 MinGene 阈值的共表达基因集的集合;b)通过如下定义 5 产生出相邻两个时间点下的满足趋势变化的共表达基因集作为基因共表达权值图的边。

定义 5　相邻时间点下的趋势变化共表达基因集。S_1 和 S_2 是离散时序基因表达数据矩阵 D 中任意两个相邻时间点,gene1 和 gene2 分别是 S_1 和 S_2 下满足 minGene 阈值的共表达基因集。当 gene1 和 gene2 的表达值的绝对值不相等且都不为 0 时,取 gene12 = gene1 ∩ gene2,称 gene12 为 S_1S_2 下的趋势变化共表达基因集(gene12 的基因个数必须满足 minGene 阈值)。

根据定义 1 和 2 可知,共表达基因集中基因的正负号代表基因表达值的正负,也代表了基因间的正负共表达关系。所以对两个共表达基因集 gene1 和 gene2 取交集时将得到两个基因集,一个记录符号相同的基因交集,另一个记录符号相反的基因交集,新产生的基因集中的符号均按 gene1 记录。例如 S_1 下共表达基因集 gene1 = $G_1-G_2G_3G_4-G_5$ 和 S_2 下共表达基因集 gene2 = $G_1G_2G_3-G_4-G_5$ 取交将得到 S_1S_2 下趋势变化共表达基因集 sub1 = $G_1G_3-G_5$ 和 sub2 = $-G_2G_4$。

基因共表达权值图的定义如下所示:

定义 6　基因共表达权值图 $R=\{E,V,W\}$,图中每个顶点 V_i 表示一个时间点,如果两个相邻顶点 V_i 和 V_{i+1} 之间存在满足定义 5 的趋势变化共表达基因集时,这两个顶点之间构建一条无向有权边 $E_{i,i+1}$,边上的权值 $W_{i,i+1}$ 表示在这两个相邻时间点下的趋势变化共表达基因集。

DTCB 算法使用的是基于时间点连续的样本扩展的方法,为了高效地挖掘出满足定义的双聚类,本文构造的是具有时序性的基因共表达权值图,即在权值图中只有相邻时间点间才存在边,非相邻时间点间没有边。如图 2 所示,根据表 1 的时序基因表达数据集分析得到的时间点连续的基因共表达权值图。

定理 1　样本扩展所得到的连续时间点间共表达基因集的交集在每个时间点下都是共表达基因集,即在时间点间具有传递性。

证明　设 gene1、gene2、gene3 分别是时间点 S_1、S_2 和 S_3 下的共表达基因集,扩展 $S_1S_2S_3$ 得到共表达基因集的交集 gene。由于在权值图上进行样本扩展时,是对趋势变化共表达基因集进行取交操作,因此,gene 是 gene1、gene2 和 gene3 的子集,gene 是 S_1S_3 下的趋势共表达基因集 gene13 的子集,即 gene 在每个时间点下均为共表达基因集。

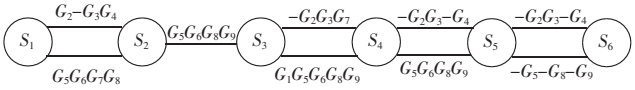


图 2　基因共表达权值图

2.2　挖掘时间点连续的最大共表达双聚类

本节将详细介绍 DTCB 算法如何从时间点连续的基因共表达权值图中有效挖掘出时间点连续的最大共表达双聚类。DTCB 算法采用了不产生候选集的方式挖掘最大共表达双聚类。在介绍剪枝策略和输出策略之前,先分析一下如何在基因共表达权值图上通过样本扩展的方式挖掘双聚类。

为了一次挖掘出所有的时间点连续共表达双聚类,基因共表达权值图中包含了每对相邻时间点下的趋势变化共表达基因集。本文从第一个时间点开始,以时间点连续的方式依次将当前时间点扩展所得的共表达基因集和候选时间点上的共表达基因集取交集,得到新的双聚类;接着扩展下一个候选时间点,直到扩展所得的双聚类不满足 minGene 和 minSample 阈值或者到达最后一个时间点为止,同时在此过程中将满足 minGene 和 minSample 阈值的最大共表达双聚类输出。再以第二个时间点为初始节点进行上述的扩展过程,依次进行下去,直到以某个时间点为初始节点,剩下的时间点个数无法产生满足 minSample 阈值的双聚类时算法结束。

DTCB 算法使用了一种有效的剪枝策略,使用该策略可以在不产生候选集的情况下有效挖掘出最大共表达双聚类。为了减少扩展中的冗余结果,采用前驱检验且不保存候选集的挖掘方法[11]是目前比较有效的方法。即如果当前扩展的双聚类是其前驱节点的某个双聚类的子集,那么当前扩展的双聚类可以被剪枝,因为它接着扩展产生的所有结果都可以由其前驱节点的双聚类扩展产生。基于上述分析,定理 2 可以保证在不产生候选集的情况下正确挖掘出所有的共表达双聚类。

定理 2　剪枝策略。设 sampleSet = $\{S_i,\cdots,S_j\}$ 为当前扩展下的时间点集合,geneSet 是当前扩展下的某一共表达基因集,当前扩展的唯一前驱时间点为 S_{i-1}。时间点对 $S_{i-1}S_i$ 下的所有趋势变化共表达基因集的集合为 preSets,如果在 preSets 中存在一个基因集 preGeneSet,它与 geneSet 的交集为 interSet,使得 interSet = geneSet,则 sampleSet 下的基因集 geneSet 将被剪枝。

证明　可知 preSampleSet = $\{S_{i-1},S_i,\cdots,S_j\}$ 下的基因集 preGeneSet 是 sampleSet = $\{S_i,\cdots,S_j\}$ 下的基因集 geneSet 的超集,因此由 sampleSet(geneSet)继续扩展得到的双聚类,也必然存在于 preSampleSet(preGeneSet)继续扩展得到的双聚类中。按本文描述的时间点连续的扩展方式,sampleSet(geneSet)及由它扩展所得的双聚类已经在初始时间点为 S_{i-1} 时就扩展过了,所以可以将 sampleSet 下的基因集 geneSet 进行剪枝。以图 2 为例,当前扩展为 S_3S_4 时,其候选时间点是 S_5,扩展 S_5 后得到双聚类 $S_3S_4S_5(G_5G_6G_8G_9)$,即 geneSet = $G_5G_6G_8G_9$;当前的唯一前驱时间点为 S_2,geneSet 与 S_2S_3 上的趋势变化共表达基因集取交后得到 interSet = $G_5G_6G_8G_9$,可得 interSet 与 geneSet 相等,则此双聚类 $S_3S_4S_5(G_5G_6G_8G_9)$ 被剪枝。从图 3 中可以明显地看出基因集 $G_5G_6G_8G_9$ 已经在 $S_2S_3S_4S_5$ 下扩展产生,因此

需要将其剪枝。

同时,DTCB 算法也使用了一种有效输出策略,使用该策略可以在挖掘过程中直接输出最大共表达双聚类。下述定理 3 详细地描述了如何直接输出最大共表达双聚类。

定理 3 输出策略。设 sampleSet = $\{S_i,\cdots,S_j\}$ 为当前扩展下的时间点集合,geneSet 是当前扩展下的某一共表达基因集,当前扩展的候选时间点为 S_{j+1},时间点对 S_jS_{j+1} 下的所有趋势变化共表达基因集的集合为 canSets。在对 S_{j+1} 扩展的过程中,将 geneSet 与 CanSets 取交得到所有交集的集合 subSets,若 subSets 中任意一个基因集都是 geneSet 的真子集,即都不等于 geneSet,且当双聚类 sampleSet(geneSet)满足 minGene 和 minsample 阈值时,输出最大共表达双聚类 sampleSet(geneSet),否则不输出。

证明 如果 geneSet 与 canSets 中任意一个基因集的交集 interSet 都为 geneSet 的真子集,由于在时间点连续的样本扩展方式下,对共表达基因集都是取交集的操作,则扩展候选时间点产生的基因集不会为此基因集 geneSet 的超集,即后续扩展过程中不会得到双聚类 sampleSet(geneSet)的超集,所以 sampleSet(geneSet)是满足极大性的最大共表达双聚类,且当此双聚类满足 minGene 和 minSample 阈值时,应将其输出。如果 canSets 中存在一个基因集 geneSet1 使得它与 geneSet 的交集和 geneSet 相等,而时间点集合 sampleSet 是 sampleSet1 = $\{S_i,\cdots,S_j,S_{j+1}\}$ 的子集,则当前双聚类 sampleSet(geneSet)是扩展后的新双聚类 sampleSet1(geneSet1)的子集,因此不满足极大性,不需要输出。以图 3 为例,当前扩展为 $S_2S_3S_4S_5$ ($G_5G_6G_8G_9$),其唯一候选时间点为 S_6,扩展 S_6 时基因集取交集得 $G_5G_8G_9$,而 $G_5G_8G_9$ 是 $G_5G_6G_8G_9$ 的真子集,则 $S_2S_3S_4S_5$ ($G_5G_6G_8G_9$)是满足阈值的最大共表达双聚类,所以将它输出。另外,当扩展到最后一个时间点时,扩展得到的双聚类若满足 minGene 和 minSample 阈值,则此双聚类必然是最大共表达双聚类并将其输出。

基于上述的剪枝策略和输出策略,本文提出的 DTCB 算法不需要在内存中保留候选集而直接挖掘出最大共表达双聚类。图 3 用表 1 的数据举例说明了 DTCB 算法的具体挖掘过程,对剪枝策略和输出策略都有特别的说明。

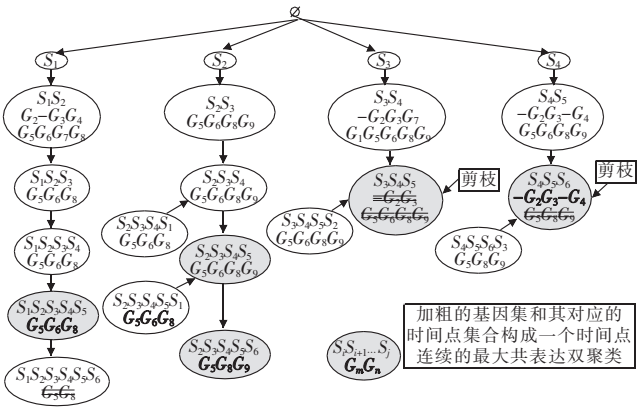


图 3 基因共表达权值图扩展过程(minGene=3,minSample=3)

2.3 算法流程

输入:离散后的基因表达数据集 D;D 的样本数 sampleNumber;D 的基因数 geneNumber;双聚类中最小时间点个数 minSample;双聚类中最小基因个数 minGene;离散化的数值个数 k。

输出:时间点连续的最大共表达双聚类集合 B。

初始化:每个时间点下的共表达基因集的集合 $C_i=\varnothing(1\leqslant i\leqslant$ sampleNumber);基因共表达权值图 R 中的每个相邻时间点对 S_iS_{i+1} 上趋势变化共表达基因集的集合 $R_i=\varnothing(1\leqslant i\leqslant$ sampleNumber -1);$B=\varnothing$。

算法描述:DTCB(D, sampleNumber, geneNumber, minSample, minGene, k)

a) 对每个时间点 $S_i(1\leqslant i\leqslant$ sampleNumber),产生满足定义 2 的共表达基因集的集合 C_i;

b) 对每个相邻时间点对 $S_iS_{i+1}(1\leqslant i\leqslant$ sampleNumber -1),产生满足定义 5 的趋势变化共表达基因集集合 R_i,即构建时间点连续的基因共表达权值图 R;

c) for 每个时间点 $S_i(1\leqslant i\leqslant$ sampleNumber $-$ minSample)

d) $B_{old}.S=\{S_i,S_{i+1}\}$;$B_{old}.G=R_i$;/ * B_{old}为当前扩展双聚类,B_{new}为扩展候选时间点后的双聚类,其中. S 和. G 分别表示双聚类的时间点集合和基因集的集合 * /

e) for 每个候选时间点 $S_{i+j}(2\leqslant j\leqslant$ sampleNumber $-i$)

f) $B_{new}.S=\{S_i,S_{i+1},\cdots,S_{i+j}\}$,并扫描当前扩展的前驱时间点 S_{i-1};

g) for $B_{old}.G$ 中的每个基因集 BoldGeneSet

h) if 前驱时间点 S_{i-1}不为空

i) 将 BoldGeneSet 和候选基因集 R_{i+j-1}中基因集取交得 interSet,再将 interSet 和前驱基因集 R_{i-1}中基因集取交得 subInterSet,如果满足定理 2 中的剪枝条件,则将剪枝 interSet,否则 $B_{new}.G=$ interSet$\cup B_{new}.G$;

j) else

k) 将 BoldGeneSet 和候选基因集 R_{i+j-1}中基因集取交得 interSet,$B_{new}.G=$ interSet$\cup B_{new}.G$;

l) if 双聚类 $B_{old}.S$(BoldGeneSet)满足定理 3 的输出条件 then $B=B_{old}.S$(BoldGeneSet)$\cup B$;

m) end for

n) if B_{new}为空则 break;else $B_{old}=B_{new}$;

o) end for

p) end for

2.4 算法复杂度分析

根据上述的算法流程,本节进行复杂度分析。记 $m=$ sampleNumber,$n=$ geneNumber。a)求每个时间点下的共表达基因集,为了描述方便,设将时序基因表达数据离散化成$2k+1$ 个值,根据定义可知每个时间点下有 k 个共表达基因集,故最坏情况下时间复杂度为 $O(kmn)$;b)构建基因共表达权值图,设两个共表达基因集取交集的时间为 mixtime,则在时间点连续的情况下只需要构建 $m-1$ 对时间点上的边,故最坏情况下时间复杂度为 $O((m-1)k(k-1)\times$ mixtime),即 $O(mk^2\times$ mixtime);c)在权值图上进行样本扩展。在时间点连续的情况下,最坏情况进行 $m(m+1)/2$ 次扩展,同时在每次扩展中要进行前驱检验,故最坏情况下时间复杂度为 $O(m^2k^2\times$ mixtime)。DTCB 算法使用了有效的剪枝策略,减少了扩展中的双聚类数量;在本文中,共表达基因集间取交时不产生重复结果,因此无冗余结果;在输出策略下,不需要在内存中保存过多的双聚类结果,便可直接输出最大共表达双聚类。根据上述分析,DTCB 算法在挖掘双聚类的过程中有较高的效率。

3 实验及分析

本文采用的时序基因表达数据是广泛使用的酵母时序基因表达数据集 cdc28。其中,cdc28 的数据来自 Cho 等人[12]的

实验,使用的是 cdc28 突变的温度敏感的酵母细胞,采样间隔为 10 min,共有 17 个采样点,记录了 6000 多个基因的表达水平。另外,对于上述数据集中缺失的少量采样值,本文采用 Oba 等人[13]2003 年提出的贝叶斯缺失值估算方法对其进行了填补。实验的硬件环境是 Intel Core™2 Duo 2.53 GHz CPU,4 GB内存的台式电脑;软件环境是微软 Windows 7 操作系统,算法编程及运行环境为 Microsoft Visual C++ 6.0。

本文提出的 DTCB 算法将与 RAP-TSB 算法进行比较。RAP[7]算法提出了范围支持度的概念,利用 Apriori-like 框架从真实的基因表达数据中挖掘出那些具有行一致表达值的双聚类。在行常量双聚类挖掘算法中,RAP 公认是比较好的算法之一,相比传统的双聚类算法 CC[6]和 ISA[15],RAP 能够挖掘出规模更小、关联度更高、更有意义的功能团。它采用的是基因扩展广度优先的挖掘策略,通过范围支持度来判断挖掘出的双聚类是否具有行一致基因表达值。但 RAP 算法是不考虑时间点序列的连续性的,为了更公平地比较,在 RAP 算法得到的双聚类结果中,将那些在连续时间点下满足范围支持度阈值且满足最小时间点个数阈值的双聚类无冗余地加到具有极大性的时间点连续共表达双聚类集合 RAP-TSB 中,把这种得到 RAP-TSB 双聚类集合的方法称为 RAP-TSB 算法(范围支持度阈值记做 rs)。

3.1　基因表达数据离散化方法

本文提出的 DTCB 算法只能对离散化的时序基因表达数据进行挖掘,下面介绍将真实数据离散化的方法。本文采用 Odibat 等人[14]提出的基于 K-means 的方法对时序基因表达数据进行离散化。在本文的实验中,将全部基因的基因表达值作为一个整体来进行离散化,而不是对每个基因或每个样本进行离散化。

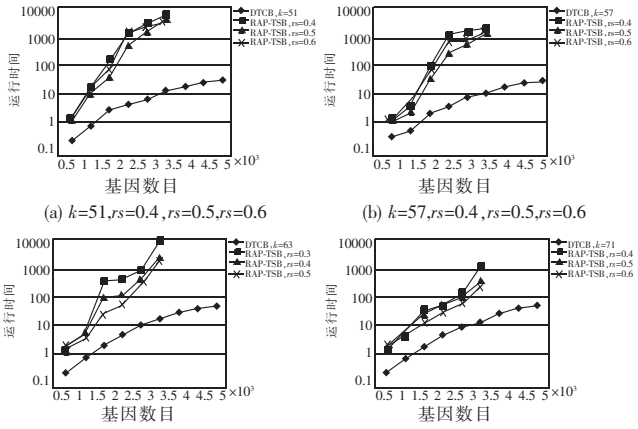
该 K-means 聚类算法中,聚类个数 k 取奇数,中间的聚类用 0 表示,余下的聚类一半用 $1,2,\cdots,k/2$ 表示,另一半用 $-k/2,-k/2+1,\cdots,-1$ 表示。该算法首先随机选取 k 个点作为起始中心点进行聚类,然后计算聚类结果的质心作为下一次聚类的中心点,如此进行 m 次聚类得到结果。由于选取起始中心点的随机性,两次运行可能产生不同的结果,因此本文运行 n 次该算法,然后计算每次结果的误差平方和用来评价聚类结果,最后选取最好的聚类结果对时序基因表达数据进行离散化。在本文的实验中,$n=10$,$m=100$。

在传统的数据离散化方法中将数据集离散为三个值($k=3$),进而挖掘的大多都是具有全局一致表达值的双聚类、具有行或列一致表达值的双聚类、具有连贯变化值的双聚类和差异共表达双聚类等。本文提出的离散为多值的方法更好地量化了基因表达水平,从而可以挖掘出更有意义的双聚类,如具有连贯变化趋势的双聚类。在 DTCB 算法中 k 是个参数,通过下面的实验分析,可以看出离散为多值的方法比传统离散化方法具有更好的效果和意义。

3.2　算法效率评价

本节比较 DTCB 算法和 RAP-TSB 算法的执行效率。为了充分对比两种算法的执行效率,使用了不同大小的数据集进行测试,数据集中基因的选取是根据 cdc28 中基因出现的先后顺序截取的。为了保证执行效率对比的公平性,选取了三组不同参数下的 RAP-TSB 算法和 DTCB 算法进行比较;在同一大小数据集下,本文选取两种算法挖掘出的双聚类结果数量相近时的运行时间。

图 4 显示了上述两种算法在不同参数、不同大小数据集下挖掘结果数目相同时的运行时间对比,本文分别取 $k=\{51,57,63,71\}$ 时 DTCB 算法与 RAP-TSB 算法的对比。从图 4 中可以看到,DTCB 算法在不同数据集和不同双聚类结果数目下运行时间都最小,当基因个数较少时两种算法运行时间相差较小;当基因个数增多时 RAP-TSB 算法的运行时间远远大于 DTCB 算法,当基因个数超过 4000 时,RAP-TSB 算法运行 24 h 后依然没有产生出结果。由于 RAP-TSB 算法采用的是基于 Apriori 原理的扩展算法,可以看出 RAP-TSB 算法的运行时间随基因个数增加成指数级增长,当基因个数超过一定规模后,可能会因为内存耗尽使得程序无法运行出结果。由此可见,DTCB 算法的执行效率和可扩展性优于 RAP-TSB 算法。



(a) k=51,rs=0.4,rs=0.5,rs=0.6　(b) k=57,rs=0.4,rs=0.5,rs=0.6

(c) k=63,rs=0.3,rs=0.4,rs=0.5　(d) k=71,rs=0.4,rs=0.5,rs=0.6

图 4　两种算法运行时间的对比

3.3　统计学意义评价

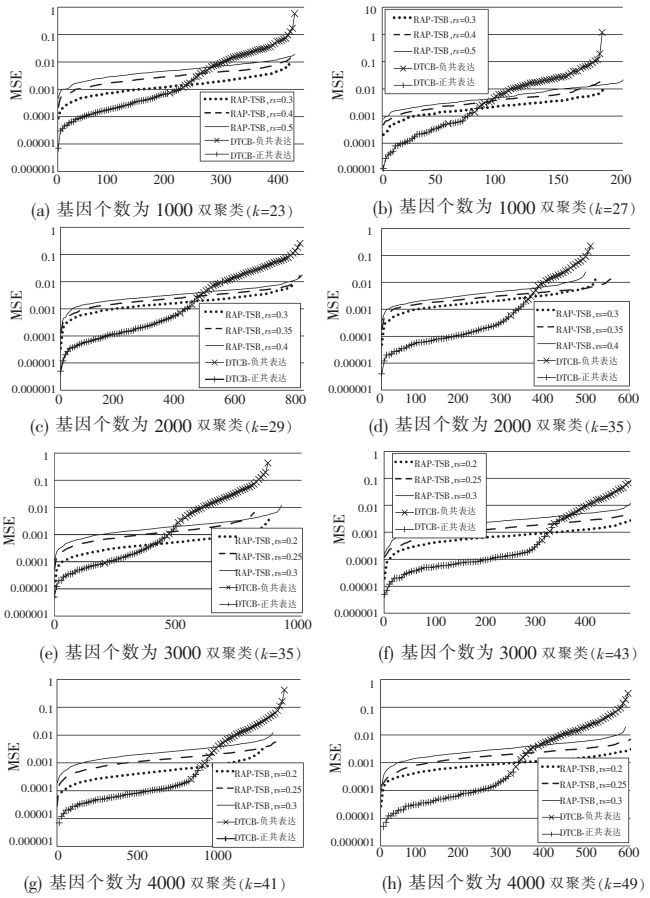
下面将比较算法的挖掘结果质量,首先使用均方误差 MSE 来衡量双聚类的差异度。均方误差用来衡量一组基因在某一组实验条件下的关联度,均方误差得分越小,则说明这组基因在这组实验条件下的表达值差异度越小,即具有高关联度。假设 I 和 J 分别是数据集中基因和时间点的集合,M_{ij} 是第 i 个基因在第 j 个时间点下的真实表达值,那么这组基因在所有时间点下的均方误差得分可由下面公式进行计算:

$$\mathrm{MSE}(I,J)=\frac{1}{|I||J|}\sum_{i\in I,j\in J}(M_{ij}-M_{iJ}-M_{Ij}+M_{IJ})^2$$

其中:$M_{iJ}=\frac{1}{|J|}\sum_{j\in J}M_{ij}$ 和 $M_{Ij}=\frac{1}{|I|}\sum_{i\in I}M_{ij}$ 分别是第 i 行和第 j 列的均值;$M_{IJ}=\frac{1}{|I||J|}\sum_{i\in I,j\in J}M_{ij}$ 是基因集在时间点集合下所有表达值的均值。

图 5 为 DTCB 算法和 RAP-TSB 算法在不同参数下和不同大小数据集下挖掘出的结果双聚类的 MSE 值分布情况。根据定义 3 可知,DTCB 算法挖掘出的双聚类结果中存在正共表达双聚类和负共表达双聚类。正共表达双聚类是所有基因呈一致趋势变化的双聚类,具有高关联度;负共表达双聚类是基因呈相反趋势变化的双聚类,因而关联度较低。因此在使用 MSE 值来评价的时候,把正共表达双聚类和负共表达双聚类分开进行对比,RAP-TSB 挖掘的是具有较高关联度的共表达双聚类。在图 5 中,可以看到 DTCB 算法挖掘出的正共表达双聚类 MSE 值大多数都比 RAP-TSB 算法低,DTCB 算法挖掘出

的负共表达双聚类 MSE 值普遍都比 RAP-TSB 算法高，即 DTCB 算法得到的结果中正共表达双聚类具有较高的关联度，而负共表达双聚类具有较大的差异度。



(a) 基因个数为 1000 双聚类(k=23)　(b) 基因个数为 1000 双聚类(k=27)

(c) 基因个数为 2000 双聚类(k=29)　(d) 基因个数为 2000 双聚类(k=35)

(e) 基因个数为 3000 双聚类(k=35)　(f) 基因个数为 3000 双聚类(k=43)

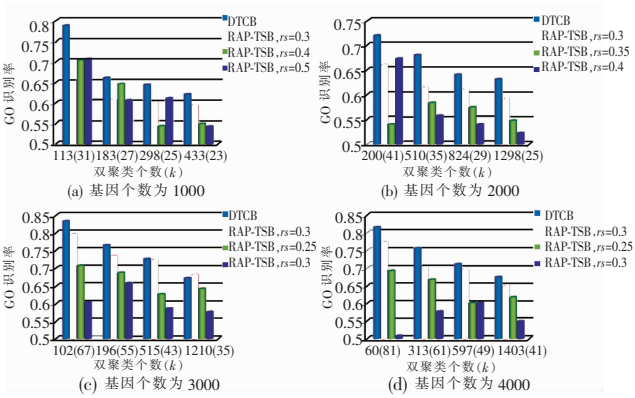
(g) 基因个数为 4000 双聚类(k=41)　(h) 基因个数为 4000 双聚类(k=49)

图 5　两种算法挖掘结果的 MSE 分布

3.4　生物学意义评价

使用均方误差来评价双聚类只是从数学统计方面来验证实验结果，但这并不能说明结果是否真的具有生物学意义。因此，本文使用另一种验证方法 GO(gene ontology)[16] 来评价实验结果。GO 是生物信息学的一项重大工程，它提供了标准的词汇和术语来描述基因产物的性质和注释数据，对每个生命载体，如基因、cDNA 或蛋白质等，都有一个或一组 GO 类别，但并不是所有的载体在 GO 数据库中都有对应的 GO 类别。本文采用的评价方法是使用 GO 类别的识别比值进行评价，即该双聚类中可以被同一个 GO 识别的基因个数与该双聚类中所有基因个数的比值大于设定的基因同源率阈值时，认为该双聚类具有生物意义。在本文中，基因同源率设置为 0.6。

图 6 是在不同规模的数据集下 DTCB 算法和三组参数下 RAP-TSB 算法的 GO 识别率的对比。在实验中，本文取两种算法挖掘出的双聚类个数相近时进行对比，以图 6 中(a)为例，纵坐标是 GO 识别率，横坐标是双聚类个数和参数 k 的取值。从图 6 中可以看到绝大多数情况下 DTCB 算法的 GO 识别率高于 RAP-TSB 算法，个别情况如(c)中 k = 35 时 GO 识别率略低于 RAP-TSB 算法(rs = 0.2)；随着双聚类个数的增大 GO 识别率在降低，但可以被识别的双聚类个数在增加，即具有生物意义的双聚类个数在增大。因此，DTCB 算法可以挖掘出更多具有生物意义的双聚类，同时也说明本文提出的离散化方法有较好的效果。



(a) 基因个数为 1000　(b) 基因个数为 2000

(c) 基因个数为 3000　(d) 基因个数为 4000

图 6　两种算法挖掘结果的 GO 识别率对比

4　结束语

本文提出了一种有效的时间点连续的共表达双聚类挖掘算法 DTCB，从离散的时序基因表达数据中挖掘出时间点连续的最大共表达双聚类。该算法提出了一种新的数据离散化方法，更好地量化了基因表达水平，并提出了离散数据集下三种基因间的共表达关系。该算法是基于权值图扩展的，采用了有效的剪枝和输出策略，使算法可以在不产生候选集的情况下，通过基因共表达权值图挖掘出所有最大共表达双聚类。同时，本文对 DTCB 算法的运行效率和挖掘结果进行了分析。a)在不同参数、不同规模数据集下运行时间的对比，可以得到 DTCB 算法的运行效率和可扩展性优于 RAP-TSB 算法；b)从统计学角度分析 DTCB 算法挖掘出的正共表达双聚类和负共表达双聚类的 MSE 值分布，根据关联度分析可得出具有更好的统计意义；c)通过 GO 评价得到 DTCB 算法可以挖掘出更多具有生物学意义的双聚类。但是，本文提出的方法是针对离散的基因表达数据，并不适用于真实数据。另外，考虑到基因相互调控的影响可能需要一定的时延才能表达出来，同时鉴于 Wang 等人[10] 提出的 time-delayed 模式可以挖掘出更有意义的双聚类，下一步的研究工作是进行带时延的共表达双聚类挖掘，以及在真实基因表达数据上进行共表达双聚类挖掘。

参考文献：

[1] RAMONI M, SEBASTIANI P, KOHANE I. Cluster analysis of gene expression dynamics [J]. PNAS, 2002, 99(14): 9121-9126.

[2] MADEIRA S C, OLIVEIRA A L. Biclustering algorithms for biological data analysis: a survey [J]. IEEE/ACM Trans on Computational Biology and Bioinformatics, 2004, 1(1): 24-45.

[3] TANAY A, SHARAN R, SHAMIR R. Discovering statistically significant biclusters in gene expression data[J]. Bioinformatics, 2002, 18(S1): 136-144.

[4] BEN-DOR A, CHOR B, KARP R, *et al*. Discovering local structure in gene expression data: the order-preserving submatrix problem [J]. Journal of Computational Biology, 2003, 10(3-4): 373-384.

[5] MURALI T M, KASIF S. Extracting conserved gene expression motifs from gene expression data[C]//Proc of the 8th Pacific Symposium on Biocomputing. 2003: 77-88.

[6] CHENG Y, CHURCH G M. Biclustering of expression data[C]//Proc of International Conference on Intelligent Systems for Molecular Biology. New York: ACM Press, 2000: 93-103.

[7] ZHAO Li-zhuang, ZAKI M. MicroCluster: efficient deterministic biclustering of microarray data[J]. IEEE Intelligent Systems, 2005, 20(6): 40-49.

(下转第 3567 页)

3　实验结果与分析

为了评估算法的性能,对不同 DAG 的虚拟企业应用进行模拟测试,并与分支定界法的性能和效率进行比较。实验仿真环境为 PIV 2.6 GHz,512 MB,操作系统为 Windows 2000。虚拟企业过程模型采用与文献[11]类似的 DAG 自动生成器,节点数由用户输入。虚拟伙伴数采用固定值 5。子项目 v_i 在候选企业 u 的加工成本 c_{ik} 与时间 t_{ik} 分别是种子为 20 和 10 的随机数;子项目 v_i 在伙伴企业 u、子项目 v_j 在伙伴企业 v 加工时工件在两个伙伴之间的转运成本和时间 $c_{iu,jv}$ 和 $t_{iu,jv}$ 分别是种子为 10 和 5 的随机数。给定的完工时间取 2 × MCP。每种节点数的 DAG(分别为 10、15、20、25、30)各运行 10 次,取其平均的成本耗费和运算时间进行比较。

图 5 描述了不同节点下两种算法最终调度方案的成本耗费和运算时间。其中圆点线是本文算法结果,从低到高依次为 10、15、20、25、30 五种 DAG 节点时的最优成本耗费和运算时间;方框线是分支定界法的结果,从低到高依次为 10、15、20、25、30 五种 DAG 节点时的最优成本耗费和运算时间。

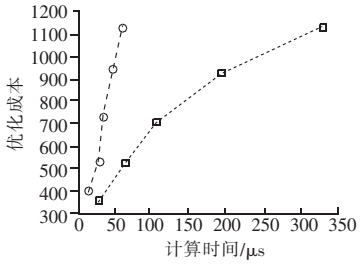


图 5　两种算法获得最终方案的运算时间与成本耗费

从图中可以看到两种算法的比较效果:

a)随着 DAG 节点数的增加,本文算法的计算时间呈线性递增,而分支定界法的计算时间呈指数递增。这是因为本文算法在完工时间超过截至期时,总是选择成本下降最快的虚拟伙伴加入;在完工时间低于截至期时,总是选择加工时间增长最慢的虚拟伙伴加入,并将对应任务的原伙伴替换出去。

b)对于同样节点数的 DAG 而言,分支定界法得到的最终成本均优于本文算法,但随节点数的增加,两者之间的差距越来越小。这是因为在不同的 DAG 下,本文算法最终方案的完工时间与给定的截至期有一定的时间碎片,无法以成本进行弥补;随着 DAG 节点数的增加,这个空闲的时间碎片变得很小,成本弥补的效果也越来越低。

4　结束语

存在工件转运时间和费用的虚拟企业伙伴选择是典型的 NP-hard 问题。本文将虚拟企业伙伴优选问题建模为项目配置图,将任务—资源分配图作为求解方案,给出了基于相对费效比的启发式算法。算法在伙伴优选过程中围绕给定的截止期逐步调整任务—资源分配图,实现优化求解。通过实例验证了模型及其求解算法的有效性。

参考文献:

[1] ZENG Zhi-bin, LI Yan, ZHU Wen-xing. Partner selection with a due date constraint in virtual enterprises[J]. Applied Mathematics and Computation, 2006, 175(2): 1353-1365.

[2] 贾瑞玉,潘雯雯,刘范范. 粗糙集与遗传算法的虚拟企业伙伴选择[J]. 哈尔滨工程大学学报, 2012, 33(6): 730-734.

[3] 张以文,倪志伟,宋捷,等. 云计算环境下动态虚拟企业伙伴选择模型[J]. 计算机科学, 2011, 38(7): 212-215.

[4] 王正成,潘晓弘, 潘旭伟. 基于蚁群算法的网络化制造资源服务链构建[J]. 计算机集成制造系统, 2010, 16(1): 174-181.

[5] 苏兆品,蒋建国,夏娜,等. 一种基于免疫的敏捷虚拟企业伙伴选择算法[J]. 中国机械工程, 2008, 19(8): 925-928.

[6] NIU S H, ONG S K, NEE A Y C. An enhanced ant colony optimiser for multi-attribute partner selection in virtual enterprises[J]. International Journal of Production Research, 2012, 50(8): 2286-2303.

[7] 尚耀华,万威武. 基于图论的虚拟企业制造伙伴选择优化算法[J]. 系统工程学报, 2006, 21(4): 375-380.

[8] 孙雪冬,李中华,刘晓锋,等. 支持个性化优化的业务过程建模[J]. 计算机集成制造系统, 2013, 19(1): 137-145.

[9] WU Nai-qi, MAO Ning, QIAN Yan-ming. An approach to partner selection in agile manufacturing[J]. Journal of Intelligent Manufacturing, 1999, 10(6): 519-529.

[10] LP W H, YUNG K L, WANG Ding-wei. A branch and bound algorithm for sub-contractor selection in agile manufacturing environment[J]. International Journal Production Economics, 2004, 87(2): 195-205.

[11] 龙浩,邸瑞华,梁毅. 基于相对效费比的网格工作流调度算法[J]. 计算机集成制造系统, 2010, 16(3): 589-597.

[12] 陈廷伟,张斌,郝宪文. 基于任务—资源分配图优化选取的网格依赖任务调度[J]. 计算机研究与发展, 2007, 44(10): 1741-1750.

(上接第 3556 页)

[8] PANDEY G, ATLURI G, STEINBACH M, *et al*. An association analysis approach to biclustering[C]//Proc of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2009: 677-686.

[9] ZHANG Ya, ZHA Hong-yuan, CHU C H. A time-series biclustering algorithm for revealing co-regulated genes[C]//Proc of IEEE International Conference on Information and Technology: Coding and Computing. 2005: 32-37.

[10] WANG Guo-ren, YIN Lin-jun, ZHAO Yu-hai, *et al*. Efficiently mining time-delayed gene expression patterns[J]. IEEE Trans on IEEE Systems, Man, and Cybernetics, Part B: Cybernetics, 2010, 40(2): 400-411.

[11] 王淼,尚学群,谢华博,等. 行常量差异共表达基因模式挖掘算法研究[C]//第 29 届中国数据库学术会议论文集 (B 辑). 2012: 228-234.

[12] CHO R, CAMPBELL M, WINZELER E, *et al*. A genome-wide transcriptional analysis of the mitotic cell cycle[J]. Molecular Cell, 1998, 2(1): 65-74.

[13] OBA S, SATO M A, TAKEMASA I, *et al*. A Bayesian missing value estimation method for gene expression profile data[J]. Bioinformatics, 2003, 19(16): 2088-2096.

[14] ODIBAT O, REDDY C, GIROUX C. Differential biclustering for gene expression analysis[C]//Proc of the 1st ACM International Conference on Bioinformatics and Computational Biology. New York: ACM Press, 2010: 275-284.

[15] IHMELS J, BERGMANN S, BARKAI N. Defining transcription modules using large-scale gene expression data[J]. Bioinformatics, 2004, 20(13): 1993-2003.

[16] HARRIS M, CLARK J, IRELAND A, *et al*. The gene ontology database and informatics resource[J]. Nucleic Acids Research, 2004, 32(database issue): 258-261.

[17] 杨蜜静,尚学群,许涛,等. 面向时序基因表达数据的双聚类算法[J]. 计算机应用研究, 2013, 30(8): 2308-2314.