从基因芯片数据快速有效地挖掘共调控基因

赵 倩,尚学群
ZHAO Qian,SHANG Xue-qun

西北工业大学 计算机学院,西安 710129
School of Computer,Northwestern Polytechnical University,Xi'an 710129,China
E-mail:zhaoqian_qiezi@126.com

ZHAO Qian,SHANG Xue-qun.Mining co-regulated genes from microarray data quickly and effectively.Computer Engineering and Applications,2010,46(9):33-37.

Abstract: Microarray data sets typically contain strong noise and an order of magnitude more genes than experiments.To further reduce the running time and improve the validity of co-regulated genes mined from microarray data,a new method is proposed which firstly groups all genes according to the Pearson correlation coefficient between every two genes,then uses column(gene) enumeration to mine closed frequent patterns as positive or negative co-regulated genes for each group.The experimental results show that the proposed approach can quickly and effectively mine two kinds of co-regulated genes from microarray data.
Key words: microarray data;co-regulated genes;Pearson correlation coefficient;closed frequent pattern

摘 要:针对基因芯片数据高噪音、列(基因)数比行(实验条件)数多几个数量级的特殊性,为了进一步提高从基因芯片数据挖掘共调控基因的时间效率和挖掘结果的有效性,首先根据所有两两基因对之间的Pearson相关系数对原始完整数据集进行分组,然后使用列(基因)枚举方法对各组数据分别进行闭合频繁模式挖掘,并对活化和抑制共调控关系的挖掘分别进行处理。实验结果证明:算法快速有效地挖掘出了两种共调控基因。
关键词:基因芯片数据;共调控基因;Pearson相关系数;闭合频繁模式
DOI:10.3778/j.issn.1002-8331.2010.09.011 **文章编号**:1002-8331(2010)09-0033-05 **文献标识码**:A **中图分类号**:TP311

1 引言

微阵列(microarray),又称基因芯片,是将大量生物分子样品微缩排布在一块载体上而制成的。微阵列可检测DNA、RNA或蛋白质分子的变化情况,记录不同条件下基因表达水平的变化。近年来,基因芯片数据大量涌现,为人们研究细胞的生理功能提供了重要资源[1]。基因调控网络是指一组调控因子如何调控一套基因表达的过程。利用基因芯片数据构建基因调控网络是研究调控关系的重要方法之一。

基因表达是指结构基因在生物体内的转录、翻译以及所有加工过程。任何影响基因开启与关闭、转录和翻译速率的直接因素统称为对基因表达的调控。共调控基因是受某些转录因子调控的一组基因,是建立基因调控网络的基础。并且,根据调控结果的不同,基因间的共调控关系可分为活化和抑制两种。基于对细胞生命过程的理解, 当细胞处于不同的周期或状态,或外界环境发生改变时,基因表达也会发生改变,而功能相似或相关的基因其表达模式往往呈现一定程度的相似性。有大量的文献证明:存在调控关系的基因,其表达谱存在着相关性[2]。

基因芯片数据有着完全不同于传统数据的特点:列(基因)数比行(实验条件)数多几个数量级;由于生物样本对象和实验因素的影响,大量噪音数据的存在。这些特点使得传统的数据挖掘方法并不适用于基因芯片数据,也使计算机人员面临了新的挑战。但是,自从基因芯片技术问世以来,人们一直不断地进行探索和研究,到目前为止已有很多方法被提出。

聚类是对基因芯片数据进行信息挖掘的一种常用的基本手段,它通过各种不同的数学模型将表达规律相似的基因聚为一类,并在此基础上寻找相关基因以分析基因的功能。常用的聚类方法有:层次聚类、K-means聚类、自组织特征映射网络和遗传算法等。聚类是将数据对象按相似性划分到一个个相互独立的组(Cluster)的过程,通常采用数据对象属性值间的距离进行相似或不相似的度量。因此一个Cluster中的数据对象彼此之间高度相似,不在一个Cluster的数据对象之间不相似[3]。从数学的角度,聚类得到的基因分组,一般是组内成员在数学特征上彼此相似,但与其他组中的成员不同。但是从生物学角度看,某些基因可能与多个类别高度相关,对其进行非此即彼的严格划分不符合自然规律。

频繁模式挖掘是基因芯片数据的另一重要分析方法,尤其是闭合频繁模式,因其较之全部频繁模式更少的冗余越来越受到人们的关注。但是由于一次芯片实验可以同时测量成千上万

基金项目:陕西省自然科学基金(the Natural Science Foundation of Shaanxi Province of China under Grant No.2007F27)。
作者简介:赵倩(1985-),硕士研究生,研究方向数据管理技术;尚学群(1973-),工学博士,硕士研究生导师,主研方向数据库技术,数据挖掘,生物信息学等。
收稿日期:2009-04-08 **修回日期**:2009-06-12

基因的表达水平,要从大量的基因在几十甚至上百个实验条件下的表达水平矩阵中挖掘出基因间的频繁模式,巨大的时间、空间代价仍是频繁模式挖掘算法面临的主要问题。

针对基因芯片数据高噪音、列(基因)数比行(实验条件)数多几个数量级的特殊性,以及传统基因芯片数据挖掘方法各自存在的不足,为了更快速有效地从基因芯片数据中挖掘出存在共调控关系的基因,先根据所有两两基因对之间的 Pearson 相关系数对原始完整数据集进行分组,然后对各组数据使用列枚举方法进行闭合频繁模式的挖掘。此外,利用分组之便,对基因的活化和抑制两种调控关系分别进行挖掘。Pearson 相关系数反应了两个变量之间的变化趋势是否一致,可以避免频繁模式挖掘时某些表达水平值总是很显著基因的频繁出现而导致错误的频繁模式挖掘结果。而对离散化后的基因芯片数据进行频繁模式挖掘可以剔除掉高度相似、表达水平值在所有的实验条件下总是不显著的基因。

2 知识点介绍

2.1 基因芯片数据

基因芯片技术可以在一次生物实验中同时测量一个细胞内成千上万个基因的表达水平,从而使得生物学家能够全局性地分析一个组织内基因组的行为,彻底改变了传统生物研究进行的方式。但同时基因芯片实验产生的大量数据也对如何有效地对其进行分析提出了挑战。从基因芯片实验得到的基因表达数据集可以用 $N\times M$ 的实值表达水平矩阵来表示,N(行)表示不同的实验条件个数(一般<500),M(列)表示基因个数(一般>>6 000)[4]。通常需要对得到的原始实验数据进行对数变换,经过变换后,上调的基因具有正值、下调的为负值。例如:表 1 是一组离散化后的基因芯片数据,Sj 为实验条件,Gi 为基因,N 的值为 4,M 的值为 6。

表 1 离散化后的基因芯片数据

	G0	G1	G2	G3	G4	G5
S0	0	−1	0	−1	1	1
S1	−1	−1	0	1	1	1
S2	1	1	1	1	−1	1
S3	1	0	−1	−1	0	−1

目前主要有两类基因芯片数据:一种是扰动试验(perturbation experiment)芯片数据,通过敲除特定基因,研究其下游效应,以确定与该基因存在调控关系的基因;另一种是时间序列(time-series)芯片数据,可以反映一组基因在生命活动周期的时间序列条件下的表达水平的变化,其表达水平变化的时间延迟关系可反映基因调控关系。由于时间序列芯片数据可以反映基因表达水平的连续变化,进而提示蛋白质活性和生物学网络的动态变化情况,因而近年来被广泛地用于基因之间转录调控关系的寻找和基因调控网络的构建[1]。

2.2 Pearson 相关系数

Pearson 相关系数是聚类时经常采用的一个相似度计算公式,其定义形式为:给定两个向量数据 $\boldsymbol{O}_i$ 和 $\boldsymbol{O}_j$,

$$Pearson(\boldsymbol{O}_i,\boldsymbol{O}_j)=\frac{\sum_{d=1}^{p}(O_{id}-\mu_{oi})(O_{jd}-\mu_{oj})}{\sqrt{\sum_{d=1}^{p}(O_{id}-\mu_{oi})^2}\sqrt{\sum_{d=1}^{p}(O_{jd}-\mu_{oj})^2}}$$

其中 μ_{oi} 和 μ_{oj} 分别是 $\boldsymbol{O}_i$ 和 $\boldsymbol{O}_j$ 的均值。Pearson 相关系数通过计算两个随机变量分布的线性关系测量它们变化趋势的相似度。Pearson 相关系数可以反映变量间的线性协同关系,这与生物学上基因共表达(co-expression)的概念相一致,因此可以作为表达模式相似基因的一个理想指标。

$0<Pearson(\boldsymbol{O}_i,\boldsymbol{O}_j)\leqslant 1$ 称为正相关,表示基因之间表达谱变化趋势相似,同增同减,越接近 1 相似程度越大;$-1\leqslant Pearson(\boldsymbol{O}_i,\boldsymbol{O}_j)<0$ 称为负相关,表示基因之间表达谱变化趋势相反,一个增加,另一个减少,越接近于−1 则这种相反的程度越大;$Pearson(\boldsymbol{O}_i,\boldsymbol{O}_j)=0$ 称为零相关,表示基因之间的变化趋势完全没有规律可循。

在基因芯片数据分析中,可以通过 Pearson 相关系数度量两个基因的表达谱之间的变化趋势是否一致。一般来说,如果两个基因表达谱的变化趋势越相似,则表明它们在一系列的生物过程中具有相似的行为(表达模式),即表现为共表达,这有利于揭示被共同的转录因子调控的基因,对于探索一些基因的功能有很好的指导意义。

2.3 闭合频繁模式挖掘

频繁模式挖掘,尤其是闭合频繁模式挖掘是一种重要的数据挖掘方法,同时也是基因芯片数据的主要分析方法之一。大部分频繁模式挖掘算法处理传统事务型数据库时性能较好,但对于包含大量长模式的高维基因芯片数据集,所挖掘的频繁模式完全集数目随维数(基因)的增加呈指数增长,算法性能急剧下降。闭合频繁模式以其含模式少且信息完整等优点取代了频繁模式挖掘,也越来越受到人们的关注。

已有的频繁模式挖掘算法可分为两类:列(基因)枚举和行(样本或实验条件)枚举。传统的频繁模式挖掘算法是列枚举,适用于事务型数据库,基因芯片数据集行少列多的特点对于传统的基于列枚举搜索空间的(闭合)频繁模式挖掘算法来说是一个极大的挑战,主要体现在算法的搜索空间上,如当基因数目达到 10 000 时,传统算法的搜索空间可达 $2^{10\,000}$,这是非常巨大的。文献[5]提出的 CARPENTER 算法首先提出行枚举树的概念,这便大大减少了枚举树的搜索空间,也通过实验证明了 CARPENTER 算法在时间效率上的优势。另外,文献[6]提出的 RERII 算法是结合了数据库垂直布局格式以及高效的剪枝策略的经典行枚举算法。文献[7]提出的 COBBLER 算法是一种行、列枚举结合的频繁模式挖掘算法,在枚举树的每个节点都可以根据当前数据特征选择两者中效率更好的枚举方法进行频繁模式的挖掘。

3 算法设计与分析

根据特定的应用背景,设计快速有效挖掘潜在信息的方法。针对基因芯片数据高噪音、列(基因)数比行(实验条件)数多几个数量级的特殊性,以及传统基因芯片数据挖掘方法各自存在的不足,为了更快速有效地从基因芯片数据挖掘存在共调控关系的基因,首先根据所有两两基因对之间的 Pearson 相关系数对原始连续数据集进行分组,然后对离散化的各组数据分别进行闭合频繁模式挖掘,并对基因间的活化和抑制两种类型的共调控关系分别进行挖掘。算法的总流程如图 1 所示,其中GroupByPearson 是设置 Pearson 阈值对原始数据集进行分组的过程,具体流程见图 2;FrequentPPatternMine 和 FrequentNPatternMine 是分别挖掘表示活化和抑制共调控关系的闭合频繁模式挖掘过程,该章后续描述中会给出其主要思想。

Input 基因芯片数据集 D：$N\times M$ 的实值矩阵；Pearson 阈值 *MinPearson*；最小支持度阈值 *MinSupport*
Output 闭合频繁模式集合
Algorithm
$\{D_0,D_1,\cdots,D_{n-1},D_n\}$=GroupByPearson($D$,*MinPearson*)；
for each positive group of genes D_i
进行数据离散化；
FrequentPPatternMine(D_i,*MinSupport*)；
for each negative group of genes D_i
进行数据离散化；
FrequentNPatternMine(D_i,*MinSupport*)；

图 1　算法总流程图

Procedure GroupByPearson(D,*MinPearson*)
for each gene $G_i(0\leqslant i<M)$ in D
为 G_i 创建两个组 $PGrp_i$(*positive group*)和 $NGrp_i$(*negative group*)
for each gene $G_j(i<j<M)$ in D
if Pearson(G_i,G_j)≥MinPearson，then 把 G_j 放入为 $PGrp_i$
if Pearson(G_i,G_j)≤−MinPearson，then 把 G_j 放入为 $NGrp_i$
for 每个基因 G_i 的对应组 $PGrp_i(0\leqslant i<M)$
for $PGrp_i$ 中所有 $G_j(j\neq i)$
删除基因 G_j 的对应组 $PGrp_j$ 中所有在 $PGrp_i$ 中出现过的基因(G_j 除外)
for 每个 $PGrp_i(0\leqslant i<M)$
if $PGrp_i$ 含有的基因个数为 1，then 删除分组 $PGrp_i$

图 2　分组流程图

对于某些表达水平值总是很显著的基因，如果直接进行频繁模式挖掘，其出现的频率会相当高，从而导致错误的频繁模式挖掘结果。Pearson 相关系数反应了两个变量之间的变化趋势是否一致，设置 Pearson 阈值分组，每个组里的基因都与所在组第一个基因的 Pearson 相关系数大于(小于)最小(大)阈值(见图 2)，就可避免这些表达总是很显著基因的干扰，从而提高频繁模式挖掘结果的准确率。同时，分组过程中，若没有任何基因和某基因的 Pearson 相关系数大(小)于阈值，那么这个基因将不会出现在后续的闭合频繁模式挖掘过程中，即：可以删除一些噪音数据。并且，对分组后的各组数据分别进行频繁模式挖掘可以将每次的搜索空间限制在本组少量数据内，避免对原始大数据集的反复搜索，从而提高运行时间。

分组时会有这样的情况出现，虽然某些基因被分到一组，可它们的表达水平值在所有实验条件下总是不显著。通过紧接其后的频繁模式挖掘，支持度阈值的设置可以消除这些基因的干扰，进一步提高挖掘结果的准确率。

对基因的活化和抑制共调控关系分别进行挖掘，即：分别对所有的 *PGrp* 和 *NGrp*(见图 2)进行闭合频繁模式挖掘。对于所有 *PGrp* 的挖掘，除支持度值的求解外，与传统挖掘闭合频繁模式的列枚举方法完全相同，下文会以表 1 中的数据为例说明该文算法支持度值的求解方法；对于所有 *NGrp* 的挖掘，将每组中第一个基因的表达值求反，然后用对 *PGrp* 进行挖掘时同样的方法挖掘出所有含有第一个基因的闭合频繁模式。基于抑制共调控基因的生物学意义，对于 *NGrp* 进行这样的特殊处理，其根据为：若有这样一组共调控基因 $G_i,G_j,-G_m,-G_n$，其含义为 G_i，和 G_j 被某组调控因子活化的同时，G_m 和 G_n 被这组调控因子抑制，这样的情况完全可以由 $G_i,-G_m,-G_n$ 和 $G_j,-G_m,-G_n$ 两种情况推导得出。

对各组数据挖掘闭合频繁模式前，需对所有数据进行离散化。将数据离散化为三元值(如表 1 所示)即 1(活化)、−1(抑制)和 0(既不表达也不抑制)，这是经过实验验证的比较好的离散化数量，过多就会失去离散化的优势，过少就会丢失更多信息。

支持度是一组基因(模式)同时出现的次数(即：Support−Count 值)与总实验条件个数的比值。使用列枚举方法进行闭合频繁模式挖掘时，以表 1 的数据为例 Support−Count 值的计算如下：$G_0\cap G_1$={*S*1,*S*2}，并且 G_0 和 G_1 在实验条件 *S*1 下表达值都为−1，在 *S*2 下表达值都为 1；对于这种情况，认为模式{*G*0*G*1}的 Support−Count 值为 2，即：对两组基因求交集时只要在同一实验条件下的表达值相同(都为 1 或都为−1)，就将 Support−Count 值加 1。整个枚举过程以深度优先搜索进行。按照上述 Support−Count 值的求解方法，若对表 1 的数据进行行(实验条件)枚举，要想这样求解支持度，直接对原始数据集的各行求交集无法实现。所以，采用列枚举进行闭合频繁模式挖掘。

4　实验与结果分析

实验的硬件环境是台式电脑：Intel ® 2CPU 2.53 GHz，2 GB 内存；软件环境是：微软 Windows XP SP2 操作系统，算法编程及运行环境为 Microsoft Visual C++6.0 SP6。

4.1　实验数据

实验使用了 Spellman[8]等的酵母的细胞循环数据，数据集包含 77 次基因表达测试，6 178 个基因，可以从 http://genome−www.stanford.edu/cellcycle/data/rawdata/combined.txt 下载得到。之所以选择这个数据集有 3 个原因：(1)目前对酵母细胞中调控关系的研究已相对完整，可以方便进行挖掘结果生物学意义的验证；(2)目前主要有两类基因芯片数据：扰动试验芯片数据和时间序列芯片数据，前者可以识别被调控基因的直接调控子，但对必须(essential)基因无能为力，而后者可以反映基因表达水平的连续变化，进而提示蛋白质活性和生物学网络的动态变化情况，因而被广泛用于基因之间调控关系的寻找[1]；(3)文献[3]通过其实验结果说明了：当酵母数据的实验条件个数介于 50 到 100 之间时挖掘出的共调控基因准确率最高。

由于下载得到的数据含有缺失值，在对其进行分析之前，使用 Oba[9]等的 BPCAfill(http://hawaii.aist−nara.ac.jp/~shige−o/tools/)进行缺失值的填充。对分组后的数据进行闭合频繁模式挖掘前，对连续数据的离散化设定了两个阈值 0.3 和−0.3：若某一表达水平值大于 0.3，标记为 1，即被活化；若小于−0.3，标记为−1，即被抑制；若介于两者之间，标记为 0，即不表达也不抑制。

4.2　实验结果

表 2　挖掘活化共调控基因的运行时间　　s

最小支持度/(%)		40	45	50	55	60	65
旧算法		7 213.469	407.359	76.765	12.828	2.875	0.812
新算法	*PH*=0.4	39.484	4 303.250	242.875	79.485	24.579	13.110	10.094
	PH=0.5	37.906	2 135.985	91.000	33.234	11.219	5.969	4.375
	PH=0.6	37.703	240.797	24.265	11.062	4.782	2.609	1.796

旧算法是指没有进行分组，直接对原始完整数据集用列枚举进行闭合频繁模式挖掘的方法；新算法是指该文方法。

表 3：不同最小支持度阈值下新算法和旧算法分别对实验数据挖掘表示活化共调控关系的闭合频繁模式的运行时间，新算法有分组时 Pearson 阈值(*PH*)分别为 0.4、0.5 和 0.6 的 3 组运行时间结果；

表 4：不同最小支持度阈值下新算法在分组时 Pearson 阈值分别为 0.4、0.5 和 0.6 时，挖掘表示抑制共调控关系的闭合频繁模式的 3 组运行时间结果。

表 3 挖掘抑制共调控基因的运行时间 s

最小支持度/(%)	新算法		
	PH=0.4	*PH*=0.5	*PH*=0.6
40	99.625	10.094	0.687
45	80.328	8.203	0.610
50	77.469	7.734	0.593

表 4 新算法挖掘出的闭合频繁模式个数

最小支持度/(%)	*PH*=0.4		*PH*=0.5		*PH*=0.6	
	活化	抑制	活化	抑制	活化	抑制
40	198 544	7 344	211 122	3 437	105 714	622
45	28 176	939	27 901	593	17 947	157
50	9 119	283	8 712	216	6 352	75

4.3 性能分析

对比表 4 中分组时 Pearson 阈值分别为 0.4 和 0.5 时挖掘出的闭合频繁模式个数:最小支持度阈值为 40%时前者少于后者;最小支持度阈值为 45%和 50%时前者多于后者,但相差不多,推断 Pearson 阈值设置为 0.5 接近最合适的选择。当 Pearson 阈值为 0.5 时,在最小支持度阈值分别为 40、45、50 和 55 时新算法的运行时间大大少于对原始完整数据集直接使用列枚举进行闭合频繁模式挖掘的时间,而当最小支持度阈值为 60 和 65 时,虽然新算法比旧算法运行时间长,可两者都在极短的时间内完成了闭合频繁模式的挖掘,新算法并不存在明显劣势,并且当最小支持度阈值增加时可以适当选择更大的 Pearson 阈值以进一步减少新算法的运行时间。即使当 Pearson 阈值降为 0.4,在最小支持度阈值为 40 和 45 时新算法的时间性能仍远远优于旧算法。以上说明,当 Pearson 阈值为 0.5 时新算法在绝大多数最小支持度阈值下的时间性能明显优于旧算法,为了在较低最小支持度阈值下进一步提高新算法性能可以适当增大分组时的 Pearson 阈值。

为了挖掘抑制共调控基因,算法的特殊处理也在较短的时间内挖掘出了结果(见表 3)。如表 4 所示,新算法得到的表示抑制共调控关系的闭合频繁模式明显少于表示活化共调控关系的闭合频繁模式。但是,得到的表示抑制共调控关系的闭合频繁模式不会丢失信息。

4.4 生物学意义分析

使用 YEASTRACT 数据库(http://www.yeastract.com/)验证闭合频繁模式是否存在共调控关系。

YBR088C 和 YCL040W 没有共同的调控因子,不是共调控基因,两者之间的 Pearson 相关系数为 0.090 505 6,新算法不会得到这个频繁模式,可直接在原始完整数据集上进行列枚举的传统闭合频繁模式挖掘,算法将 YBR088C 和 YCL040W 作为一个结果挖掘了出来,并且其支持度值为 40%。图 3 所示为 YBR088C 和 YCL040W 的表达水平值,由此图 3 可以看出:两者表达水平值的变化趋势并不相似,在某些时间段还存在相反的变化趋势(时间点 60~77),但是,两者在很多实验点表达都很显著,所以旧算法将它们挖掘出来了。

同样道理:YBL032W 和 YGR279C(图 4)、YBR088C 和 YGR044C(图 5)没有共同的调控因子,不是共调控基因,两组基因的 Pearson 相关系数分别为 0.361 191 和 0.333 410,不是新算法的结果,但却被传统方法挖掘出来,两组支持度值都是 40%。

由图 5 可以看出:YBR088C 和 YGR044C 在时间点 15 到 47 的变化趋势基本吻合,但在其他时间点情况却截然不同,甚至有完全相反的变化趋势。若将实验条件个数降为 33,则极有可能将 YBR088C 和 YGR044C 作为共调控关系基因而挖掘出来。这也再次验证了文献[3]中提到的基因芯片实验条件个数对挖掘共调控基因的影响。

图 6 为 YBL002W 和 YGR189C 的表达水平值,经查询两者有 3 个共有的调控因子,存在共调控关系。由于其 Pearson 相关系数为 0.365 623,没有被新算法挖掘出来,但传统算法得到了这两个基因组成的闭合频繁模式,支持度值为 45%。仔细观察发现,两者的变化趋势存在延迟相似关系,属于异步共调控基因,异步共调控基因是指一个基因在另一基因的某段时间延迟之后两者才出现共调控。这说明:新算法不能挖掘出异步共调控基因。

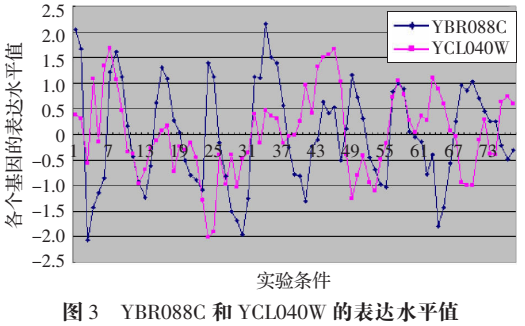

图 3 YBR088C 和 YCL040W 的表达水平值

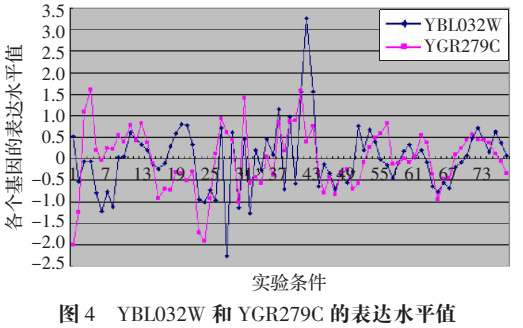

图 4 YBL032W 和 YGR279C 的表达水平值

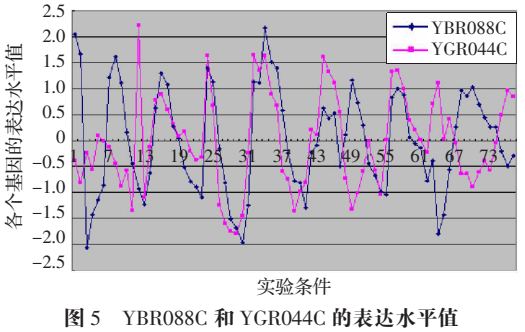

图 5 YBR088C 和 YGR044C 的表达水平值

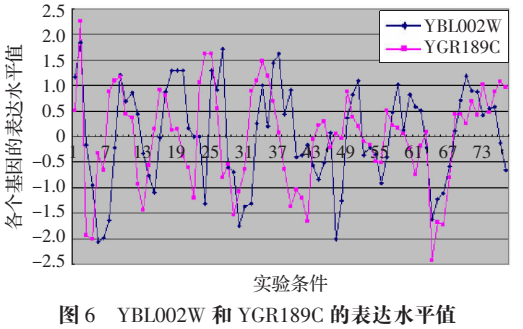

图 6 YBL002W 和 YGR189C 的表达水平值

5　结束语

针对基因芯片数据的高噪音、列(基因)数远远大于行(实验条件)数的特点,以及传统基因芯片数据挖掘方法各自存在的不足,为了更快速有效地从基因芯片数据中挖掘出存在共调控关系的基因,设计实现了一种先用两两基因间的 Pearson 相关系数进行分组,然后再对每组数据使用列枚举方法分别进行闭合频繁模式的挖掘。并且,分组后,对基因间的活化共调控和抑制共调控关系分别进行挖掘。实验结果证明:新算法不仅在时间性能上优于直接在原始完整数据集进行列枚举的闭合频繁模式挖掘算法,通过对已有调控关系数据库的查询也验证了其在剔除一些不存在共调控关系的闭合频繁模式上的优势,共调控基因挖掘的准确率得到进一步提高。

虽然算法剔除了一些不是共调控关系的结果,但也删除了少量确实存在共调控关系的结果。究其原因,可能是以下两方面造成的:(1)芯片实验过程中造成的误差;(2)挖掘出的只是共表达基因,而共表达并不能完全等同于共调控,两者之间的关系复杂多变,各种各样,如异步共调控。

参考文献:

[1] 刘万霖,李栋,朱云平,等.基于微阵列数据构建基因调控网络[J].遗传,2007(12).

[2] 李传星,李霞,郭政,等.调控通路内基因表达的相关性分析[J].遗传,2004,26(6):929-933.

[3] Ka Y Y,Mario M,Roger E B.From co-expression to co-regulation:How many microarray experiments do we need?[J].Genome Biology,2004,5.

[4] Mclntosh T,Chawla S.High-confidence rule mining for microarray analysis[J].IEEE/ACM TCBB,2007,4(4):611-623.

[5] Feng Pan,Gao Cong,Yang Jiong,et al.CARPENTER:Finding closed patterns in long biological datasets[C]//Proc ACM SIGKDD Int'l Conf on Knowledge Discovery and Data Mining(KDD),2003:637-642.

[6] Gao Cong,Tan Kian-Lee,Tung A K H,et al.Mining frequent closed patterns in microarray data[C]//Proceedings of the 4th IEEE International Conference on Data Mining(ICDM'04),2004:363-366.

[7] Feng Pan,Gao Cong,Xu Xin,et al.COBBLER:Combining column and row enumeration for closed pattern discovery[C]//Proc of the 16th Int Conf on Scientific and Statistical Database Management,2004:21-30.

[8] Spellman P T,Sherlock G,Zhang M Q,et al.Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization[J].Mol Biol Cell,1998,9:3273-3297.

[9] Oba S,Sato M,Takemasa I,et al.A Bayesian missing value estimation method[J].Bioinformatics,2003,19:2088-2096.

[10] Dominic J A,Isaac S K,Atul J B.Quantifying the relationship between co-expression,co-regulation and gene function[J].BMC Bioinformatics,2004,5.

[11] Jiang Da-xin,Tang Chun,Zhang Ai-dong.Cluster analysis for gene expression data:A survey[J].IEEE Transactions on Knowledge and Data Engineering,2004,16(11):1370-1986.

[12] Creighton C,Hanash S.Mining gene expression databases for association rules[J].Bioinformatics,2003,19(1):79-86.

(上接 30 页)

[10] Cai Zi-xing,Wang Yong.A multiobjective optimization-based evolutionary algorithm for constrained optimization[J].IEEE Trans on Evol Comput,2006,10(6).

[11] Runarsson T P,Yao X.Stochastic ranking for constrained evolutionary optimization[J].IEEE Trans on Evol Comput,2000,4(3):284-294.

[12] Eppstein D.Quasiconvex programming[J].ACM Computing Research Repository,cs.CG/0412046,2004.

[13] Powell D,Skolnick M M.Using genetic algorithm in engineering design optimization with nonlinear constraint[C]//Forrest S.Proc 5th Int Conf Genetic Algorithms,San Mateo,CA,1993:424-431.

[14] Kita H.A comparison study of self-adaptation in evolution strategies and real-coded genetic algorithms[J].Evol Comput,2001,9(2):223-241.

(上接 32 页)

因此,$f|_{(s)}=g|_{(s)}$。

定理 3.13　设 Q 是 Quantale,S 是 Q 上的基。如果 $f,g:Q\to K$ 是 Quantale 同态,且 $f|_s=g|_s$,则 $f=g$。

命题 3.14　设 Q 是 Quantale,S 是 Q 上的基(或稠密子集)。如果 $f,g:Q\to K$ 是 Quantale 满同态,则 $f(S)$也是 K 上的基(或稠密子集)。

参考文献:

[1] Rosenthal K I.Quantales and their application[M].New York:Longman Scientific and Technical,1990.

[2] Rosenthal K I.A general approach to Gabriel filters on Quantales[J].Communications in Algebra,1992,20(11):3393-3409.

[3] 王国俊.L-fuzzy 拓扑空间论[M].西安:陕西师范大学出版社,1988.

[4] Resende P.Quantales,finite observations and strong bisimulation[J].Theoretical Computer Science,2001,254:95-149.

[5] 李永明.非可换线性逻辑及其 Quantale 语义[J].陕西师范大学学报:自然科学版,2000,29(2):1-5.

[6] 韩胜伟.几类 Quantale 结构及其 Quantale 矩阵的研究[D].西安:陕西师范大学,2005.

[7] 王顺钦,赵彬.Girard Quantale 若干性质的研究[J].陕西师范大学学报:自然科学版,2007,35(2):10-13.

[8] Yetter D.Quantales and(noncommutative)linear logic[J].Journal of Symbolic Logic,1990,55:41-64.

[9] 韩胜伟,赵彬.单纯 Quantale 及其 Quantale 商[J].模糊系统与数学,2005,19(4):28-33.