

时序微阵列数据中的同步和异步共调控基因聚类

印 莹 赵宇海 张 斌 王国仁

(东北大学信息科学与工程学院 沈阳 110004)

摘 要 基因的共调控可分为同步和异步两种. 文中提出了一种新的聚类模型 Reg-Cluster, 将具有相同编码的同步和异步共调控基因聚集到同一个共调控基因类中. 在此基础上, 提出了一种有效的聚类算法 FBLD, 采用先宽度优先、后深度优先的搜索策略, 并结合高效的削减规则, 挖掘得到所有符合条件的最大 Reg-Cluster. 聚类结果中包含了详细而完备的共调控信息, 有助于基因调控网的研究. 算法可扩展用于三维基因-样本-时间微阵列数据集的分析. FBLD 算法已经应用到真实和人造微阵列数据集中, 其结果被提交到 Gene Ontology, 实验结果证明了算法的高效性和有效性.

关键词 同步/异步共调控; 活化/抑制共调控; 聚类; 时间序列; 基因本体

中图法分类号 TP311

Mining Synchronous and Asynchronous Co-Regulated Gene Clusters from Time Series Microarray Data

YIN Ying ZHAO Yu-Hai ZHANG Bin WANG Guo-Ren

(College of Information Science and Engineering, Northeastern University, Shenyang 110004)

Abstract Gene co-regulation falls into two major categories, i. e. synchronous and asynchronous co-regulation. This paper proposes a new model Reg-Cluster, which groups synchronous and asynchronous co-regulated genes together if they have the same code. Further, an effective clustering algorithm with several efficient pruning rules, namely FBLD, is designed to identify all maximal Reg-Clusters in a "First Breadth-first and Last Depth-first" manner. The resultant clusters contain the detailed and complete co-regulation information, which facilitates the study of genetic regulatory networks. Moreover, the method can be extended to the analysis of 3D gene-sample-time microarray data. The FBLD algorithm has been implemented on both real and synthetic datasets and the results from the real dataset has been submitted to Gene Ontology. Experimental results prove the effectiveness and efficiency of the proposed method.

Keywords synchronous/asynchronous co-regulation; activation/inhibition co-regulation; clustering; time series; gene ontology

1 引 言

生物系统的复杂性使基因/基因聚类之间存在

着多种不同的关系, 大致分为同步共调控和异步共调控. 进一步, 根据调控结果的不同, 每一种共调控又可分为两类: 活化和抑制. 根据对这些调控关系的分析, 人们能够深入地洞察基因/基因聚类之间的相

互影响和作用关系. 发现共调控基因,即具有相同转录调控因子的基因,对揭示基因调控网各成员间的关系具有极其重要的生物意义^[1].

表 1(a)给出了一个由 m 行和 n 列组成的微阵列数据集 D ,其中行代表基因, $G=\{g_1,g_2,\cdots,g_m\}$,列代表实验条件(不同的时间点), $T=\{t_1,t_2,\cdots,t_n\}$. 基因 $g_i(1\leq i\leq m)$ 在某个特定时间点 $t_j(1\leq j\leq n)$ 的表达值,记为 $d_{i,j}$. 为了便于讨论,表中的某些单元格被置空,假定它们对应随机表达值. 表 1(b)是对表 1(a)中的某些行进行置换后得到的,其中有两个不同的共调控基因聚类. 第 1 个共调控聚类对应表 1(b)中的多边形阴影区域,其中的每条基因表达谱如图 1(a)所示;第 2 个共调控聚类,对应表 1(b)中虚线矩形包含的区域,其中每个基因的表达谱如图 1(b)所示. 注意:同一共调控聚类内的每对基因间必定具有已知共调控关系中的一种. 例如:当 $T=$

$\{t_1,t_2,t_4,t_5\}$ 时,图 1(a)中的基因 g_1 和基因 g_4 间存在着平移模式^[2-4],因为 $d_{1,T}=d_{4,T}+25$;当 $T=\{t_1,t_3,t_5,t_6,t_7\}$ 时,基因 g_3 和基因 g_6 间存在着成比例模式^[5],因为 $d_{6,T'}=d_{3,T}\times 3$;基因 g_3 和基因 g_8 间存在着平移且成比例模式^[5-6],因为 $d_{8,T'}=4\times d_{3,T'}+5$. 图 1(b)中的共调控模式都是同步的,因为每个基因的产物会立即影响其它基因的表达. 进一步,同步共调控又可分为同步活化共调控和同步抑制共调控^[7]. 在同步活化过程中,一个基因表达值的增加(或降低)会影响其它基因表达值的增加(或降低),如图 1(a)中基因 g_1 的表达谱随基因 g_4 的表达谱同时“起伏”. 但是,在同步抑制过程中,情况恰恰相反,一个基因表达值的增加(或降低)会引起其它基因表达值的降低(或增加),如图 1(a)中基因 g_1 和基因 g_5 之间的“反转”模式^[7].

表 1 一个简单的微阵列表达矩阵

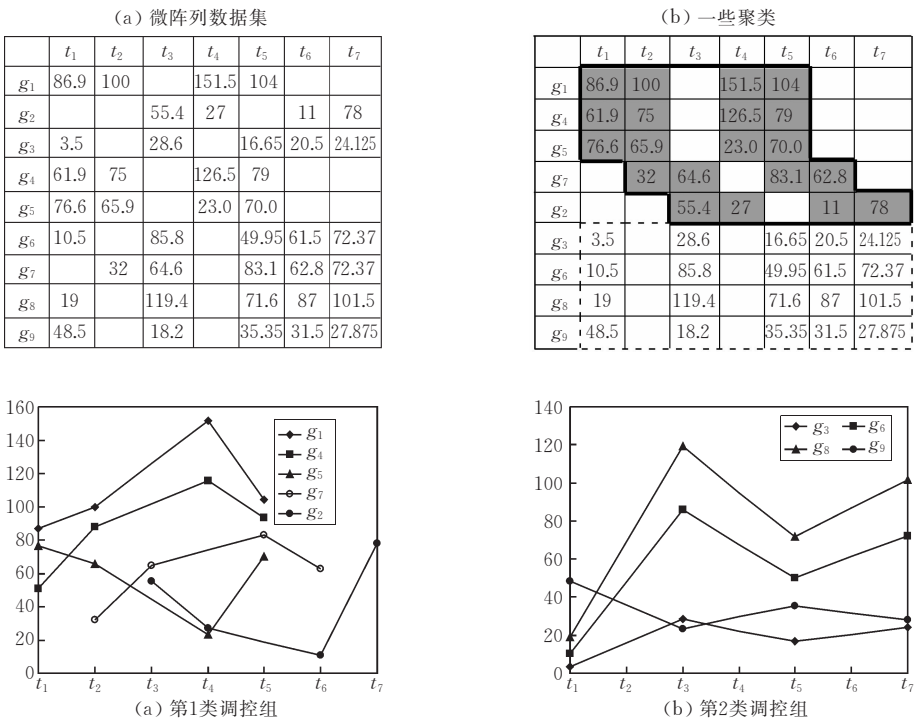


图 1 两类调控组

实际上,在时间序列基因表达数据^[8]中,大多数基因并不是同时(在相同的时间点上)发生共调控,而是在某段时间延迟之后才出现共调控,本文称之为异步共调控^[9]. 与同步共调控类似,异步共调控也分为活化和抑制两种. 如图 1(a)中,基因 g_1 和基因 g_7 间的时间平移模式属于异步活化共调控,因为基因 g_7 的表达谱与基因 g_1 滞后 1 个时间点后的表达谱有相同的“起伏”趋势;基因 g_1 和基因 g_2 间的反转

时间平移模式属于异步抑制共调控,因为基因 g_2 的表达谱与基因 g_1 滞后 2 个时间点后的表达谱有相反的“起伏”趋势. 文献[9]也进一步证实,图 1(a)所示的异步共调控现象在真实世界中是普遍存在的,生物上通常解释为细胞过程通过异步控制某些基因使其它基因得以表达. 发现异步共调控基因对完善调控网的功能,寻找调控网中的新成员有非常大的帮助. 不幸的是,现有的聚类算法大多忽略了这两种

共调控模式的识别.

已有的用于从微阵列数据集中发现共调控关系的方法主要分为以下两类:基于模式/趋势的子空间聚类^[10]和“两个基因,一次比对一种关系^[4]”.前者通常只考虑静态(相同时间点下)的基因表达谱,如单纯的平移模式^[2-4],单存的成比例模式^[5]或二者的简单组合^[6],没有考虑任何异步关系,因此不能用于解释基因调控中的时间延迟现象.许多隐藏在时序微阵列数据中的重要调控关系被忽略了.后者,即“两个基因,一次比对一种关系”,意味着每次比对只能发现两个基因间的一种调控关系.显然,这种计算方法不是高效的.更大的问题在于,这种方法基于所有给定的条件(时间点)评估基因表达谱的相似特性,无法发现那些仅在一小部分相关时间点上共调控,而在其余时间点上不发生共调控的基因.然而,这样的基因在时序微阵列数据中是普遍存在的^[11].

本文深入研究了以上提到的问题,主要贡献点包括:(1)提出并形式化了一个新的子空间共调控基因聚类模型 Reg-Cluster,以整体的方式同时识别时序微阵列数据中所有同步和异步的共调控模式;(2)提出了一种新的编码方法——regCode,使共调控基因具有相同的编码;(3)提出了一个新的基于树的聚类算法——FBLD,并通过有效的削减和优化策略,高效地发现最大同步和异步共调控基因聚类;(4)从聚类结果中,基于提出的编码方案,可以进一步得到更详细的共调控信息,例如:活化共调控模式、抑制共调控模式及活化/抑制共调控模式间滞后的时间点数;(5)基于真实数据集和人工数据集的大量实验进一步证实了提出算法的高效性和有效性.

本文第 2 节给出 Reg-Cluster 模型的定义和问题描述;第 3 节讨论详细的 FBLD 算法及所用的削减规则和优化策略;第 4 节给出实验结果和分析;最后,第 5 节总结全文并提出未来工作.

2 Reg-Cluster 模型

本节将给出同时识别时序微阵列数据集中同步和异步共调控基因的 Reg-Cluster 模型的相关定义和形式化问题描述.

2.1 基本概念

若 $G = \{g_1, g_2, \dots, g_m\}$ 为 m 个基因的集合, $T = \{t_1, t_2, \dots, t_n\}$ 为 n 个时间点的集合,则给定的时序微阵列数据集可以用一个 $m \times n$ 的二维实值矩阵 D 来表示, $D = G \times T = \{d_{i,j}\}$, 其中 $i \in [1, m]$, $j \in [1, n]$. 矩阵 D 的两个维分别代表基因和时间,每个元素

$d_{i,j}$ 代表第 i ($1 \leq i \leq m$) 个基因 g_i , 在第 j ($1 \leq j \leq n$) 个实验条件 t_j 下的表达值. 表 1 为一个简单的 9×7 微阵列表达矩阵,记录了 9 个基因在 7 个不同的时间点下的表达值. 为了方便随后的讨论,首先给出以下几个描述 Reg-Cluster 模型必需的定义.

定义 1. 原型子序列. 设 T 是一组实验条件的集合, T 中的所有条件按照实验顺序构成的序列称为 T , 记为 $T = \langle t_1, t_2, \dots, t_n \rangle$, 存在 $T' = \langle t_{i_1}, t_{i_2}, \dots, t_{i_l} \rangle$ ($1 \leq i_l \leq n$), 如果满足 $T' \subseteq T$ 且 $i_1 < i_2 < \dots < i_l$, 称 T' 是 T 的原型子序列.

例如,令 $T = \langle t_1, t_2, t_4, t_5, t_6, t_7 \rangle$, 则 $T' = \langle t_2, t_4 \rangle$ 和 $T'' = \langle t_2, t_5, t_6, t_7 \rangle$ 分别为 T 的两个原型子序列, 而 $\langle t_4, t_2 \rangle$ 和 $\langle t_5, t_6, t_2, t_7 \rangle$ 不是.

定义 2. l -segment. 给定 T 的一个原型子序列 $T' = \langle t_{i_1}, t_{i_2}, \dots, t_{i_{l+1}} \rangle$, 存在 l 个长度为 2 的相邻子序列, 即 $\langle t_{i_1}, t_{i_2} \rangle, \langle t_{i_2}, t_{i_3} \rangle, \dots, \langle t_{i_l}, t_{i_{l+1}} \rangle$, 则称 T' 是一个与长度为 2 的原型子序列个数相关的 l -segment. T' 中的元素个数, 记作 $|T'|$, 称为 T' 的长度.

例如,令 $T = \langle t_1, t_2, t_3, t_4, t_5, t_6, t_7 \rangle$, 那么 $T' = \langle t_1, t_3, t_4 \rangle$ 和 $T'' = \langle t_1, t_3, t_5, t_7 \rangle$ 分别是 T 的一个 2-segment 和 3-segment.

定义 3. 重要的调控. 给定基因 g_a 和 l -segment, $\langle t_{i_j}, t_{i_k} \rangle, l \in [1, n-1]$, 若条件 $|d_{a,i_k} - d_{a,i_j}| > \delta$ 成立, 则称基因 g_a 从时间点 t_{i_j} 到 t_{i_k} 的调控是重要的调控. 其中, δ 是用户给定的调控阈值.

根据以上调控重要性的定义,可以得到下面两种具体的调控信息:

$$Reg(g_a, (\langle t_{i_j}, t_{i_k} \rangle)) = \begin{cases} \nearrow, & d_{a,i_k} - d_{a,i_j} > \delta \\ \searrow, & d_{a,i_k} - d_{a,i_j} < -\delta \\ \leftrightarrow, & |d_{a,i_k} - d_{a,i_j}| \leq \delta \end{cases} \quad (1)$$

$Reg(g_a, (\langle t_{i_j}, t_{i_k} \rangle))$ 代表 g_a 从 t_{i_j} 到 t_{i_k} 的调控方向.

如果基因 g_a 在时间点 t_{i_j} 和 t_{i_k} 上的表达值满足条件 $d_{a,i_k} - d_{a,i_j} > \delta$, 称之为上调, 简记为 $Reg(g_a, (\langle t_{i_j}, t_{i_k} \rangle)) = \nearrow$; 如果基因 g_a 在时间点 t_{i_j} 和 t_{i_k} 上的表达值满足条件 $d_{a,i_k} - d_{a,i_j} < -\delta$, 称之为下调, 简记为 $Reg(g_a, (\langle t_{i_j}, t_{i_k} \rangle)) = \searrow$. 为了方便讨论, 记 $\nearrow = -\searrow$, $\searrow = -\nearrow$.

注意: 本文只关心调控相关的基因, 因此要求基因 g_a 在条件 t_{i_j} 和 t_{i_k} 上的调控是重要的调控, 即要求 $|d_{a,i_k} - d_{a,i_j}| > \delta$.

定义 4. ‘O’ 操作. 给定一个 2-segment $\langle t_{i_j}, t_{i_k}, t_{i_l} \rangle$, $O(\langle t_{i_j}, t_{i_k}, t_{i_l} \rangle) = O(Reg(\langle t_{i_j}, t_{i_k} \rangle), Reg(\langle t_{i_k}, t_{i_l} \rangle))$, 其中, $Reg(\langle t_{i_j}, t_{i_k} \rangle)$ 和 $Reg(\langle t_{i_k}, t_{i_l} \rangle)$ 分别代

表序列 $\langle t_{i_j}, t_{i_k} \rangle$ 和 $\langle t_{i_k}, t_{i_l} \rangle$ 上的调控趋势, 即上调控“ \nearrow ”或下调控“ \searrow ”. ‘O’操作有下列性质:

- (1) $O(\nearrow, \nearrow) = 1$; $O(\searrow, \searrow) = 1$;
- (2) $O(\nearrow, \searrow) = 0$; $O(\searrow, \nearrow) = 0$.

定义 5. regCode. 给定基因 g_a 和 l -segment $T_l = \langle t_{i_1}, t_{i_2}, \dots, t_{i_l} \rangle$, 以 t_{i_1} 为起点, 依次计算并连接所有 $O(\langle t_{i_k}, t_{i_{k+1}}, t_{i_{k+2}} \rangle)$ 的运算结果 ($1 \leq k \leq l-1$), 得到由 1/0 组成的长度为 $l-1$ 的序列, 称其为基因 g_a 在 l -segment T_l 上的 regCode 编码, 简记为 $regCode(g_a, T_l)$.

注意: g_a 在 l -segment ($l > 1$) 上的 regCode 是基于两个相邻基本调控单元的趋势改变给出的.

例如: 在图 1(a) 中, 基因 g_1 在 3-segment $T = \langle t_1, t_2, t_4, t_5 \rangle$ 上的 $regCode(g_1, T) = O(\langle t_1, t_2, t_4 \rangle) O(\langle t_2, t_4, t_5 \rangle) = 10$.

根据 regCode 的定义, 得到以下同步共调控和异步共调控的形式化定义.

定义 6. 同步共调控. 给定两个基因 g_a, g_b , 如果存在一个 l -segment, $T_l = \langle t_{i_1}, t_{i_2}, \dots, t_{i_{l+1}} \rangle$, 满足 $regCode(g_a, T_l) = regCode(g_b, T_l)$, 那么称基因 g_a 和基因 g_b 在子序列 T_l 上同步共调控. 进一步说, 如果 $Reg(g_a, \langle t_{i_k}, t_{i_{k+1}} \rangle) = Reg(g_b, \langle t_{i_k}, t_{i_{k+1}} \rangle)$, $k \in [1, l]$, 那么基因 g_a 和基因 g_b 之间的同步共调控是活化的. 反之, 如果 $Reg(g_a, \langle t_{i_k}, t_{i_{k+1}} \rangle) = -Reg(g_b, \langle t_{i_k}, t_{i_{k+1}} \rangle)$, 那么基因 g_a 和基因 g_b 之间的同步共调控是抑制的.

定义 7. 异步共调控. 给定两个基因 g_a 和 g_b , 如果存在两个 l -segment, $T_l = \langle t_{i_1}, t_{i_2}, \dots, t_{i_{l+1}} \rangle$ 和 $T'_l = \langle t'_{i_1}, t'_{i_2}, \dots, t'_{i_{l+1}} \rangle$, 满足 $regCode(g_a, T_l) = regCode(g_b, T'_l)$, 且 $t'_{i_1} - t_{i_1} = t'_{i_2} - t_{i_2} = \dots = t'_{i_{l+1}} - t_{i_{l+1}} = d$, ($d > 0$) 是延迟时间点的个数, 则称 g_a 和 g_b 在子序列 T_l 和 T'_l 上异步共调控. 进一步说, 如果 $Reg(g_a, \langle t_{i_k}, t_{i_{k+1}} \rangle) = Reg(g_b, \langle t'_{i_k}, t'_{i_{k+1}} \rangle)$, $k \in [1, l]$, 那么基因 g_a 和基因 g_b 之间的异步共调控是活化的, 基因 g_a 在滞后 d 个时间点后, 异步活化共调控基因 g_b . 反之, 如果满足 $Reg(g_a, \langle t_{i_k}, t_{i_{k+1}} \rangle) = -Reg(g_b, \langle t'_{i_k}, t'_{i_{k+1}} \rangle)$, 那么基因 g_a 和基因 g_b 之间的异步共调控是抑制的, 基因 g_a 在滞后 d 个时间点后, 异步抑制共调控基因 g_b .

例如: 图 1(a) 显示的是基因 g_1, g_4, g_5 在原型子序列 $T = \langle t_1, t_2, t_4, t_5 \rangle$, 基因 g_7 在原型子序列 $T' = \langle t_2, t_3, t_5, t_6 \rangle$, 基因 g_2 在原型子序列 $T'' = \langle t_3, t_4, t_6, t_7 \rangle$ 上的表达谱. 其中, 基因 g_1 和基因 g_4 在原型子序列 $T = \langle t_1, t_2, t_4, t_5 \rangle$ 上是同步活化共调控. 基因 g_4 和基因 g_5 在时间子序列 $T = \langle t_1, t_2, t_4, t_5 \rangle$ 上是同步

抑制共调控. 基因 g_5, g_1 和基因 g_7 在原型子序列 $T = \langle t_1, t_2, t_4, t_5 \rangle$ 和 $T' = \langle t_2, t_3, t_5, t_6 \rangle$ 上异步活化共调控, 基因 g_7 滞后基因 g_1, g_5 1 个时间点. 基因 g_4 和基因 g_2 在原型子序列 $T' = \langle t_2, t_3, t_5, t_6 \rangle$ 和 $T'' = \langle t_3, t_4, t_6, t_7 \rangle$ 上是异步抑制共调控, 基因 g_4 滞后 g_2 2 个时间点. 注意: 如果把异步共调控的延迟时间点 d 置成 0, 就是同步共调控模式, 因此, 同步共调控模式可以看成异步共调控模式的一个特例, 此时时间后滞 $d = 0$.

2.2 模型定义和问题描述

定义 8. Reg-Cluster. 给定 $C = \bigcup_{i=1}^r G_i \times T_j$, 其中 G_i 是一系列基因子集 ($G_i \subseteq G$, $G = \{g_1, g_2, \dots, g_m\}$), T_j 是 T 的原型子序列 ($T_j \subseteq T$, $T = \{t_1, t_2, \dots, t_n\}$), 如果 C 满足以下两个条件, 则称 C 是一个 Reg-Cluster.

- (1) $\forall T_i \subseteq T, T_j \subseteq T, 1 \leq i \leq j \leq r, |T_i| = |T_j|$;
- (2) $\forall g_a \in G_i, \forall g_b \in G_j, 1 \leq i \leq j \leq r$, 满足条件 $regCode(g_a, T_i) = regCode(g_b, T_j)$, 并且 $t_{j_1} - t_{i_1} = t_{j_2} - t_{i_2} = \dots = t_{j_k} - t_{i_k}$, 其中 $T_i = \langle t_{i_1}, t_{i_2}, \dots, t_{i_k} \rangle, T_j = \langle t_{j_1}, t_{j_2}, \dots, t_{j_k} \rangle$.

例如, 图 1(a) 给出了一个 Reg-Cluster $C_1 = \{g_1, g_4, g_5\} \times \{t_1, t_2, t_4, t_5\} \cup \{g_7\} \times \{t_2, t_3, t_5, t_6\} \cup \{g_2\} \times \{t_3, t_4, t_6, t_7\}$, 其中的任何一对基因都表现出同步共调控或异步共调控模式. 显然, 以前的算法不能发现这些不在同一个属性集(不同的时间点)上的共调控信息.

令 \mathcal{C} 是满足定义 8 的所有 Reg-Cluster 集合, $C \in \mathcal{C}$, 如果不存在其它的 Reg-Cluster C' 满足 C' 包含 C , 即 $C' \in \mathcal{C}$, 那么 $C \in \mathcal{C}$ 被称为最大的 Reg-Cluster.

问题描述: 给定: (1) \mathbf{D} , 一个微阵列数据矩阵; (2) δ , 用户指定的最大共调控阈值 ($\delta > 0$); (3) min_t , 最小时间点数; (4) min_g , 最小基因数, 挖掘出所有满足定义 8 要求的最大共调控基因聚类 Reg-Cluster, 并且 $|G_i| \geq min_g, |T_j| \geq min_t$. 真实应用中, 只有当一个共调控基因聚类包含足够多的基因和条件时, 它才是有意义的^[3].

3 算法设计与分析

Reg-Cluster 算法有两个主要步骤: (1) 构建初始 Reg-tree. 在这个步骤中, 所有 1-segment 上的共调控信息和初始 Reg-Cluster 被保存; (2) 递归的发展初始 Reg-tree 来发现所有最大的共调控基因聚

类,不同于以往的算法,提出的算法采用“先宽度优先,后深度优先”的搜索策略(因此又称为 FBLD)来提高搜索效率,其中在宽度优先和深度优先搜索过程中,可以分别采用不同的特定削减规则和优化策略使算法更有效.

算法 1. Reg-Cluster 算法 FBLD.

输入: $D(m \times n$ 的表达矩阵), δ , min_g , min_t

输出: 最大共调控基因聚类集合, M

算法描述:

1. $M \leftarrow \emptyset$; $l = 1$;
2. 为高度为 2 的 1-segment, T_2 创建初始 Reg-tree, R ;
3. 应用削减规则 3,4;
4. if $min_t = 2$ then
5. 插入 T_2 中最大的 Reg-Cluster 到 M , 如果它们满足参数的限定
6. end if
7. while $(l \leq min_t - 1)$ do
8. 基于 min_t -jumping 技术, 构建高度为 l' 的 Reg-tree, 其中 $l' = \min(min_t - 1, 2 \times l)$;
9. $l \leftarrow l'$;
10. 在已建立的 Reg-tree 基础上, 宽度优先构建新的 Reg-tree T_l ;
11. 应用削减规则 1,4;
12. end while
13. Call $DFS(T_l, l_f)$
14. 插入 T_l 中满足条件的最大 Reg-Cluster 到 M ;
15. return M ;

Procedure $DFS(T_l, l_f)$

1. for 对于最左分支 l_f do
2. 扩展最左分支 l_f 到 $l_{f'}$ 如 3.2 节描述;
3. 应用削减规则 2;
4. if 在 $l_{f'}$ 分支上的结果是最大的 then
5. 输出到 M
6. end if
7. $DFS(T_l, l_{f'})$

8. end for

FBLD 算法的伪码如算法 1 所示. 首先,为所有的 1-segment 构建初始 Reg-tree(1~3 行),详细的步骤在 3.1 节中讨论. 其中,削减规则 3 和削减规则 4 被用来削减掉那些没有意义的 Reg-Cluster 和搜索空间中不可能继续发展的分枝(第 3 行). 注意,如果 $min_t = 2$, 算法 1 将为所有的 1-segment 发现最大的共调控聚类(4~6 行);其次,递归的发展初始 Reg-tree 来发现最大共调控基因聚类(7~13 行),该算法在 3.2 节有详细讨论且可以进一步分解为以下两个步骤:(1)(第 7~12 行)在宽度优先的基础上,跳跃式的发展 Reg-tree,利用提出的 min_t -jumping 技术,尽可能快地构建出包含满足最小条件数阈值(min_t)要求的 Reg-tree;(2)以深度优先的方式逐步发现每个 l -segment($l \geq min_t$)上的最大子空间共调控基因聚类(第 13 行). 在这个步骤中,削减规则 2 用来过滤掉那些不可能包含最大 Reg-Cluster 的搜索空间. 函数 $DFS(T_l, l_f)$ 解释了树的高度是 l , 满足阈值 min_t 的 Reg-tree 的发展过程. 最后,返回所有发现的最大 Reg-Cluster(第 14~15 行).

3.1 初始 Reg-tree 的构建

初始 Reg-tree 高度是 1. 图 2 显示了对应于表 1 的初始 Reg-tree. 树中包含了所有符合定义 8 的 1-segment 上的 Reg-Cluster. 每个叶子结点下有两个分枝,一个包含了所有在该分枝对应的 1-segment 上重要上调控的基因集合(用‘ \nearrow ’表示),另一个包含了所有在该分枝上重要下调控的基因集合(用‘ \searrow ’表示). 每个 Reg-Cluster $C = \bigcup_{i=1}^r G_i \times T_i$ 由一个带有标号的桶集组成. 标号为 0 的桶被称为“基桶”,因为每个基桶对应的原型子序列 T_1 由从根节点到 Reg-Cluster C 所连接的结点路径上的时间点组成. 每个桶的标号代表 T_i 滞后于基桶 T_1 的时间点数.

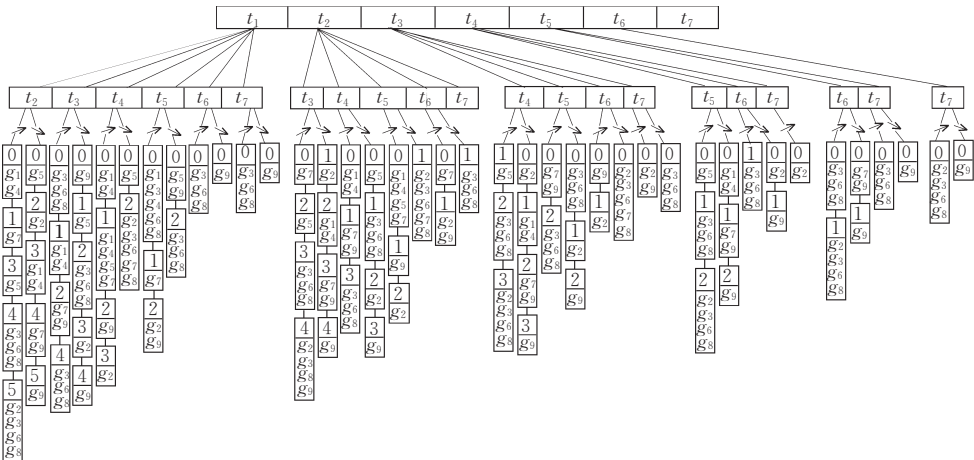


图 2 对应表 1 中的初始 Reg-tree

例如,图 2 中路径 $t_1 t_3$ 下的最左侧 Reg-Cluster 由 5 个桶组成. 其中,基桶 T_1 的原型子序列是 $\langle t_1, t_3 \rangle$, 标号为 0. 第 2 个桶对应的原型子序列是 $\langle t_2, t_4 \rangle$, 因为这个桶的标号是 1. 同理,第 3 个桶对应的原型子序列是 $\langle t_3, t_5 \rangle$, 依次类推. 基因 g_1 和 g_4 在 $t_1 t_2$ 下标记为 ‘ \nearrow ’ 的分枝中, 说明它们在 $\langle t_1, t_2 \rangle$ 上是重要上调控的. g_7 也在同一分枝下, 但对应的桶标号是 1, 说明基因 g_7 是滞后基因 g_1 和 g_4 1 个时间点的异步共调控基因. 同理, 基因 g_2 在 $t_1 t_2$ 下标记为 ‘ \searrow ’ 的分枝中, 对应的桶标号是 2, 说明基因 g_2 是滞后基因 g_1 和 g_4 2 个时间点的异步共调控基因.

3.2 对应于 2-segment 的 Reg-tree

本节, 基于已构建的高度为 1 的 Reg-tree 为所有 2-segment 构建高度为 2 的 Reg-tree.

在高度为 2 的 Reg-tree 中, 一个 2-segment $T_2 = \langle t_i, t_j, t_k \rangle$ 是通过连接初始 Reg-tree 中两个 1-segment $\langle t_i, t_j \rangle$ 和 $\langle t_j, t_k \rangle$ 得到的. 每个 2-segment 下的叶子节点都包含两个基因聚类: L -Cluster 和 R -Cluster. L -Cluster 包含所有在 $\langle t_i, t_j \rangle$ 和 $\langle t_j, t_k \rangle$ 上有相同调控(同升或同降)的基因, 也就是说, L -Cluster 中基因的 regCode 都是 1. 类似的, R -Cluster 包含所有在 $\langle t_i, t_j \rangle$ 和 $\langle t_j, t_k \rangle$ 上有不同调控(上升/下降或下降/上升)的基因, 也就是说,

R -Cluster 中基因的 regCode 都是 0. 注意: 所有的共调控基因或者被聚集到 L -Cluster 中, 或者被聚集到 R -Cluster 中.

下面, 通过一个例子详细说明 2-segment Reg-tree 的构建过程. 图 2 中 1-segment, $\langle t_1, t_2 \rangle$ 下有两个基桶. 一个表示上调控(‘ \nearrow ’), 包含 $\{g_1, g_4\}$, 另一个表示下调控(‘ \searrow ’), 包含 $\{g_5\}$. 同样, 在 1-segment $\langle t_2, t_4 \rangle$ 下也有两个基桶. 一个表示上调控(‘ \nearrow ’), 包含 $\{g_1, g_4\}$, 另一个表示下调控(‘ \searrow ’), 包含 $\{g_5\}$. 据此, 可以产生图 3 中 2-segment $\langle t_1, t_2, t_4 \rangle$ 下的两个聚类: L -Cluster 和 R -Cluster, 其中每个聚类都由具有不同时间滞后 d 的一组子聚类构成. 具体的构建过程如下:

(1) $\langle t_1, t_2, t_4 \rangle$ 下 L -Cluster 中的每个具有时间滞后 d 的子聚类是 $\langle t_1, t_2 \rangle$ 和 $\langle t_2, t_4 \rangle$ 下具有相同时间滞后 d 的 \nearrow -Cluster 交集与 $\langle t_1, t_2 \rangle$ 和 $\langle t_2, t_4 \rangle$ 下具有相同时间滞后 d 的 \searrow -Cluster 交集的并集. 例如: 对应于 $d=0$ 的第 1 个子聚类通过 $(\{g_1, g_4\} \cap \{g_1, g_4\}) \cup (\{g_5\} \cap \{g_5\}) = \{g_1, g_4, g_5\}$ 得到; 对应于 $d=1$ 的第 2 个子聚类通过 $(\{g_7\} \cap \{g_7, g_9\}) \cup (\{\emptyset\} \cap \{g_3, g_6, g_8\}) = \{g_7\}$ 得到. 依此类推, 得到 L -Cluster 下的所有子聚类.

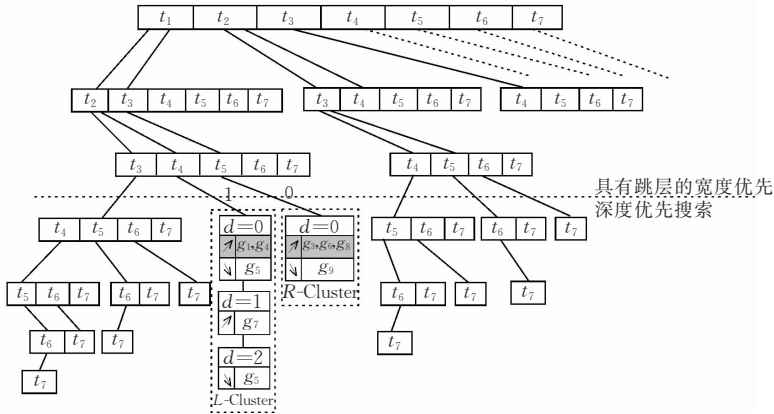


图 3 对应表 1 的 l -segment 对应的 Reg-tree

(2) $\langle t_1, t_2, t_4 \rangle$ 下 R -Cluster 中的每个具有时间滞后 d 的子聚类是 $\langle t_1, t_2 \rangle$ 下具有相同时间滞后 d 的 \nearrow -Cluster 和 $\langle t_2, t_4 \rangle$ 下具有相同时间滞后 d 的 \searrow -Cluster 交集与 $\langle t_1, t_2 \rangle$ 下具有相同时间滞后 d 的 \searrow -Cluster 和 $\langle t_2, t_4 \rangle$ 下具有相同时间滞后 d 的 \nearrow -cluster 交集的并集. 例如, 对应于 $d=0$ 的第一个子聚类通过 $(\{g_1, g_4\} \cap \{g_5\}) \cup (\{g_5\} \cap \{g_1, g_4\}) = \{\emptyset\}$ 得到. 依此类推, 得到 R -Cluster 下所有子聚类.

注意: 一般来说, 在初始 Reg-tree 中, 一个给定的 1-segment $\langle t_i, t_j \rangle$ 下会有多于一个的桶, 因此,

在 2-segment $T_2 = \langle t_i, t_j, t_k \rangle$ 下的 L -Cluster 或 R -Cluster 中上应该有多于一个子聚类.

3.3 l -segment 上的 Reg-tree ($l > 2$)

本节介绍当 $l > 2$ 时, 递归扩展 Reg-tree, 挖掘最大共调控基因聚类的算法 FBLD. 与以前的算法不同, 本文采用“先宽度优先, 后深度优先”的搜索策略(First Breadth-first and Last Depth-first)使提出的算法更有效. 顾名思义, 算法由两个阶段组成: (1) 宽度优先搜索 BFD (Breadth-First Development); (2) 深度优先搜索 DFD (Depth-First Development).

opment).

在 BFD 阶段,不同于以前的算法,在 Reg-tree 的高度达到 $\min_i - 1$ 前,没有必要按这种逐层递增的方式增长 Reg-tree 的高度.可以直接跳过一些不必要的层,以下是提出的 \min_i -based jumping 削减规则.

削减规则 1. \min_i -based jumping 削减. 给定 k -segment $\langle t_{i_1}, t_{i_2}, \dots, t_{i_{k+1}} \rangle$ 和 l -segment $\langle t_{j_1}, t_{j_2}, \dots, t_{j_{l+1}} \rangle$, 可以直接获得 $\min(\min_i, (k+1))$ -segment, 跳过了 $(k+1)$ -segment $\sim \min(\min_i, (k+l-1))$ -segment, 当且仅当 $t_{i_{k+1}} = t_{j_{l+1}}$.

显然, \min_i 越大, 削减规则 1 越有效. 在生物学上, 一个大的 \min_i 意味着统计意义上的聚类^[3]. 此时, 削减规则 1 是非常有效的. 在图 3 中, 若假定 $\min_i = 8$, 那么仅需要创建高度分别为 1, 2, 4, 7 的 Reg-tree T_1, T_2, T_4 和 T_7 , 而不必生成中间高度为 3, 5, 6 的 Reg-tree T_3, T_5, T_6 . 一旦 Reg-tree 的高度达到 $\min_i - 1$, FBLD 算法即转入下一发展阶段, 深度优先发展阶段 (DFD).

与前一阶段不同, 深度优先总是优先扩展当前 Reg-tree 中的最左分枝, 直到 Reg-tree 中的最右分枝扩展结束. 此处, 可以通过下述规则 2 削减不可能包含最大共调控基因聚类的搜索空间, 加快 Reg-tree 的深度优先发展过程.

削减规则 2. 给定 Reg-tree 中的两条路径 X 和 Y , $X = \langle t_{i_1}, t_{i_2}, \dots, t_{i_m} \rangle$, $Y = \langle t_{j_1}, t_{j_2}, \dots, t_{j_n} \rangle$. 若 $X \subseteq Y$ 且 X 下的 Reg-Cluster 与 Y 下的 Reg-Cluster 相同, 则 $\langle t_{i_1}, t_{i_2}, \dots, t_{i_m} \rangle$ 上的 Reg-Cluster 不是最大的, 并且所有路径 $t_{i_1} t_{i_2} \dots t_{i_m}$ 下的搜索都可以被削减, 因为此路径下不可能包含任何最大的 Reg-Cluster (证明略).

FBLD 在两个不同的发展阶段中可以分别使用削减规则 1 和削减规则 2, 所以它优于纯粹的 BFD 或纯粹的 DFD. 它首先以宽度优先的方式发展 Reg-tree, 一旦 Reg-tree 的高度涨到 $\min_i - 1$, 发展转到下一个阶段 DFD.

3.4 更多的削减策略

除了前述的削减规则 1 和削减规则 2, 在 FBLD 上还可以使用以下削减规则, 它们对进一步提高算法效率是非常有帮助的.

削减规则 3. \min_g -based 削减. 如果一个连接到某个结点单元 (v) 的 Reg-Cluster 包含的基因数小于 \min_g 个基因, 那么它可以被削减. 因为在对应原型子序列上的进一步扩展只会减少该 Reg-Cluster 中的基因数. 进一步, 如果连接到 v 的 Reg-Cluster

中的基因总数少于最小基因阈值 \min_g , 则 v 下的所有搜索都可以被削减.

削减规则 4. 削减短序列.

(1) 对于一个长度为 2 的 1-segment $\langle t_i, t_j \rangle$, 令 T' 为任意的扩展 $\langle t_i, t_j \rangle$ 的时间原型子序列, 那么, 从 $\langle t_i, t_j \rangle$ 扩展的最长的时间原型子序列是 $T_{\max} = \langle t_1, \dots, t_{(i-1)}, t_i, t_j, t_{(j+1)}, \dots, t_n \rangle$. 如果 T_{\max} 的长度小于 \min_i (也就是 $i + (j - i + 1) < \min_i$), 那么 $\langle t_i, t_j \rangle$ 序列下就不能产生任何时间点大于或等于 \min_i 个时间点的共调控基因聚类, 因此, 它上面的所有 Reg-Cluster 和其后的所有搜索都可以在构造原始 Reg-tree 的时候削减掉.

(2) 当构造原始 Reg-tree 的时候, 如果符合 $j - i \leq n - \min_i + 1$, 只需要应用削减规则 2(a) 生成 $\langle t_i, t_j \rangle$ 上的 Reg-Cluster. 还可以进一步削减这些时间序列上 Reg-Cluster 中的桶. 令 c 表示一个 $\langle t_i, t_j \rangle$ 上的 Reg-Cluster, c' 表示 c 中的一个桶. 假设 c' 的桶数字为 d , 那么 c' 的时间序列就是 $\langle t_{(i+d)}, t_{(j+d)} \rangle$. 由于从 $\langle t_i, t_j \rangle$ 扩展的最长的时间序列为 $T_{\max} = \langle t_1, \dots, t_{(i-1)}, t_i, t_j, t_{(j+1)}, \dots, t_n \rangle$, 那么从 $\langle t_{(i+d)}, t_{(j+d)} \rangle$ 扩展的最长时间序列就是 $T_{d\max} = \langle t_{(1+d)}, \dots, t_{(i-1+d)}, t_{(i+d)}, t_{(j+d)}, t_{(j+1+d)}, \dots, t_n \rangle$. 如果 $T_{d\max}$ 的长度小于 \min_i (也就是 $i + n - (j + d) + 1 < \min_i$), 那么 $\langle t_{(i+d)}, t_{(j+d)} \rangle$ 就不能产生任何时间点大于或等于 \min_i 的共调控基因聚类, 因此如果 c' 的桶数字 $d > n - \min_i + 1 - (j - i)$, 它就可以被削减掉.

(3) 在 Reg-tree 的发展过程中, 对于深度为 h 的节点中的一个格 v , 假设在其后允许出现的时间点个数是 k . 如果 $h + k < \min_i$, 则削减掉格 v 后面的搜索, 因为在这条路径上不能形成满足 \min_i 要求的 Reg-Cluster.

3.5 Reg-Cluster 结果分析

如前所述, Reg-Cluster 算法不仅能识别所有的同步共调控和异步同调控模式, 而且更加详细的调控信息也可以从聚类结果中很容易的导出, 如活化模式、抑制模式和在活化模式或抑制模式中的滞后时间点数.

图 4 给出了对图 3 中 $\langle t_1, t_2, t_4 \rangle$ 上的聚类结果的进一步分析, 基于该 Reg-Cluster 4 个已知的共调控信息可以很容易的得到. 其中, 同步活化共调控基因是那些具有相同时间延迟 d 且在初始序列 $\langle t_1, t_2 \rangle$ 上具有相同调控趋势 (同上调控或同下调控) 的基因; 同步抑制共调控基因是那些具有相同的时间延迟 d 且在初始序列 $\langle t_1, t_2 \rangle$ 上具有不同调控趋势 (上调控/下调控或下调控/上调控) 的基因; 异步活化共

调控基因是那些具有不同的时间延迟 d ，但在初始序列 $\langle t_1, t_2 \rangle$ 上具有相同调控趋势(同上调控或同下调控)的基因；异步抑制共调控基因是那些具有不同的时间延迟 d 而且在初始序列 $\langle t_1, t_2 \rangle$ 上具有不同调控趋势(上调控/下调控或下调控/上调控)的基因。

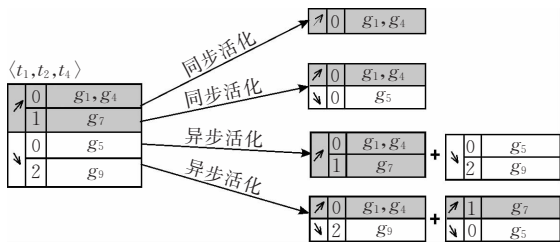


图 4 详细的聚类结果分析

例如，在 $\langle t_1, t_2, t_4 \rangle$ 上，基因 g_1 和 g_4 相互同步活化共调控；基因 g_1/g_4 与基因 g_5 相互同步抑制共调控；基因 g_1/g_4 异步活化共调控基因 g_7 ，时间滞后为 1；基因 g_5 异步活化共调控基因 g_9 ，时间滞后为 2；基因 g_5 异步抑制共调控基因 g_7 ，时间滞后为 1；基因 g_1/g_4 异步抑制共调控基因 g_9 ，时间滞后为 2。注意：所有用于得到这些细节的必需信息都可以在 Reg-tree 构建后得到。

与以前的方法相比，提出的算法在同时识别共调控基因间的同步和异步关系(包括活化和抑制)方面更有效。而且，结果中以简洁得方式包含了所有用户所需的调控信息。基于这些结果，用户能够明确地知道同步共调控或异步共调控模式的起始时间和终止时间，甚至能捕捉到两个基因之间具体的延迟时间滞后数。这在以前的算法中是不能尝试得到的。根据结果传递的信息，用户可以对感兴趣的基因或 bicluster 进行更深层的探究。

4 实验

本文从聚类结果的真实生物意义，模型的敏感性，参数的有效性，算法的可伸缩性和可扩展性等方面对三个算法进行分析比较。为了简单起见，基本的

宽度优先算法被称为 BBFS，基本的深度优先算法被称为 BDFS，先宽度优先后深度优先算法简记为 FBLD。实验环境为 2.4GHz DELL PC, 512MB 内存, Window XP 操作系统。文中算法均用 C++ 实现。实验数据包括真实微阵列数据集和人工合成数据集。对于真实数据集，本文使用了两个 Yeast 基因表达数据集，其中一个可以在 <http://genome-www.stanford.edu/cellcycle/data/rawdata/> 上获得，包含了 6178 个基因在 35 个时间点上的表达值，另一个也是 Yeast 数据集，可以在 <http://arep.med.harvard.edu/biclustering/> 上获得，包含了 2884 个基因在 17 个时间点上的表达值^[6]，人工合成数据集由一个特定的数据集产生器生成^[12]，并受以下参数控制：(1) 基因的数目；(2) 时间点的数目；(3) 嵌入的聚类个数。

4.1 性能分析

本节，首先通过分别增长人工合成数据集的基因数、时间点数和嵌入到数据集中聚类个数的变化评估三个算法 BBFS, BDFS, FBLD 的可伸缩性，如图 5 所示。默认的参数分别为：最小基因数 $min_g = 30$ ，最小时间点数 $min_t = 6$ ，调控阈值 $\delta = 0.01$ 。

图 5(a)显示了 3 个算法随基因数变化的响应时间比较，其中 $min_t = 6, \delta = 0.01$ 。图 5(b)显示了 3 个算法在随时间点数变化情况下的响应时间比较。其中 $min_g = 30, \delta = 0.01$ 。图 5(c)显示了 3 个算法随嵌入的聚类个数变化情况下的响应时间比较，其中 $min_g = 30, min_t = 6, \delta = 0.01$ 。可以看出，随着基因数和时间点数的增加，Reg-tree 渐宽渐深。因此，响应时间必然会变长。可以看到，FBLD 算法要优于 BBFS 和 BDFS，而且数据集越大，嵌入的聚类数越多，优势越明显。FBLD 的特殊搜索算法和有效的削减规则极大的削减掉了搜索空间，所以它的响应时间最短。在发展 Reg-tree 的过程中，BBFS 需要决定哪些桶要进行连接来生成新的桶，BDFS 不必要。所以 BBFS 相对 BDFS 要花费更多的搜索时间。

下面，分别在 3 个数据集(两个 Yeast 数据集和

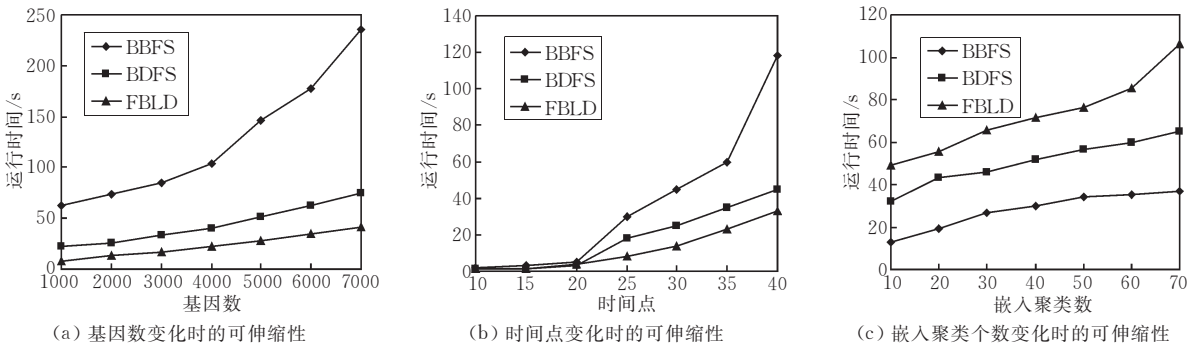


图 5 有效性评估

一个人工数据集)上观察参数(min_g 和 min_t)的变化对算法响应时间的影响,结果如图 6 和图 7 所示.可以看出,随着 min_g 和 min_t 的增加,响应时间变短.因

为大的 min_g (或 min_t)会过滤掉许多不合格的聚类,并因此导致 Reg-tree 的规模变小.

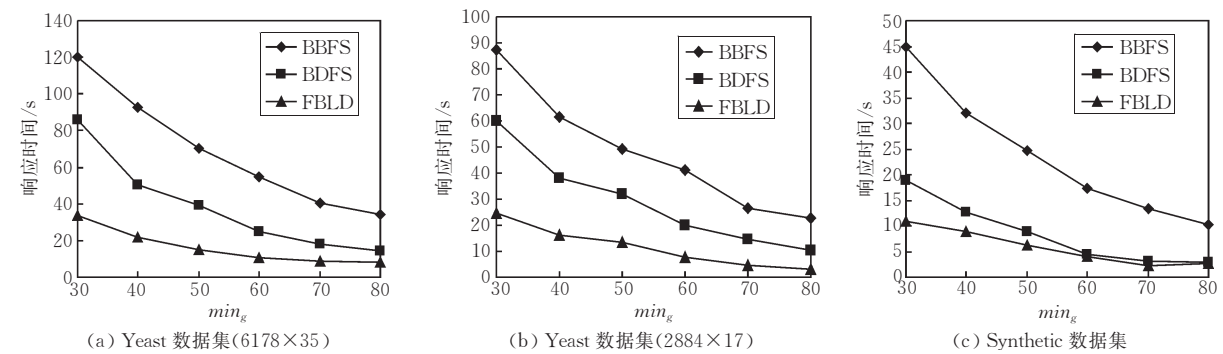


图 6 响应时间随 min_g 变化

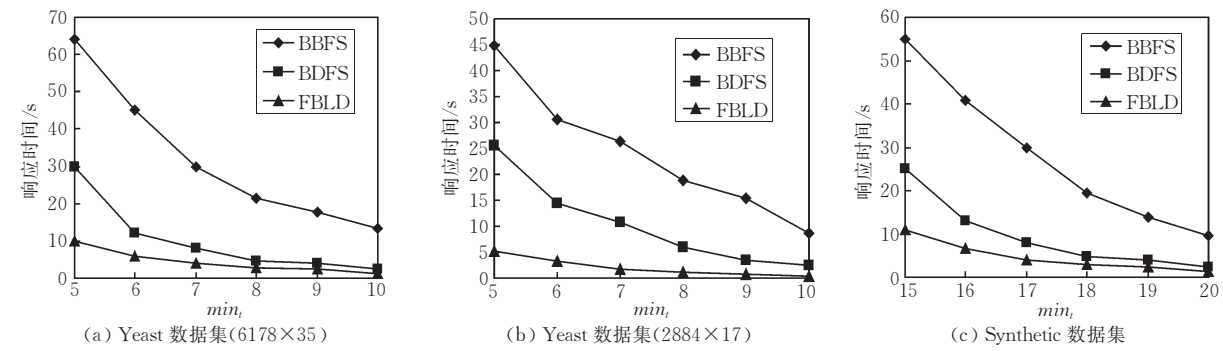


图 7 响应时间随 min_t 变化

4.2 结果的生物意义

FBLD 算法在真实数据集上发现了一些令人感兴趣的共调控基因聚类结果. 为了更好地理解结果的生物意义,GO 和 p -值被用来进一步评估聚类结果的生物兴趣,即是否在一个或多个功能组内有重要的丰度. 把结果提交到在线分析工具 Ontology Term Finder (<http://db.yeastgenome.org/cgi-bin/GO/goTermFinder>)对每个提交的结果都可以根据 Gene Ontology 获得一个层次的功能注释来分别计算他们在联合生物过程、细胞成分以及基因功能各方面的生物学意义. 表 2 为每个基因在 Reg-Cluster 聚类中构建了层次 GO. 并且给出了与 C2,

C8,C10 三个聚类结果中较低 p 值对应的顶层 GO 术语,这些是以前的工作中没注意到的. 例如,对于聚类 C10,发现基因主要与 positive regulation of transcription 有关. 元组 ($n=15, p=9.20E-5$)意味着在 75 个基因中,15 个已被证实属于这个过程,统计重要性通过 p 值= $9.20E-5$ 给出. 由于空间限制,虽然还有一些聚类对应的术语和 p -值,只在表 2 中列出了最重要的共有 GO 术语及相关的最低 p -值. 从表中可以看出,聚类在每个类别中是唯一的. 较低的 p 值表示聚类在生物过程、细胞成分以及基因功能中有重要的生物学意义.

表 2 某些发现的 Reg-Cluster 的 GO 术语

聚类号	基因数	生物过程	基因功能	细胞成分
C2	43	mRNA splice site selection ($n=2, p=0.00045$), nuclear mRNA 3'-splice site ($n=2, p=0.00113$)	Specific RNA polymerase II transcription factor activity ($n=4, p=0.00078$)	nucleus ($n=11, p=0.00596$)
C8	52	response to DNA damage stimulus ($n=12, p=0.000517$)	double-stranded DNA specific 3'~5' exodeoxyribonuclease ($n=4, p=0.00113$)	cell fraction ($n=3, p=0.00953$)
C10	75	positive regulation of transcription, DNA-dependent ($n=15, p=0.000920$)	RNA polymerase II transcription factor activity ($n=6, p=0.00129$)	intracellular membrane-bound organelle ($n=32, p=0.00104$)

如前所述,对每个得到的 Reg-Cluster 都可以进行更深入的分析,以探究共调控基因间更深层的关系.图 8 给出了从 Spellman 的数据集中发现的共调控基因聚类 C18 的层次结构.根节点包含了 Reg-Cluster C18 中所有 179 个基因在 35 个时间点上的表达谱.注意:regCode 编码方案能够保证在给定原型子序列上一个 Reg-Cluster 包含且仅包含同步共调控或异步共调控的基因.根据时间滞后数 d 可以导出一些更详细信息的子聚类,如第 2 层所示, $d=1$ 时,共 29 个子图,分别是 $\langle t_1, t_2, \dots, t_7 \rangle$ 上的子聚类, $\langle t_2, t_3, \dots, t_8 \rangle$ 上的子聚类,一直到 $\langle t_{29}, t_{30}, \dots, t_{35} \rangle$ 上的子聚类,每个子图都依次滞后了一个时间点,每个子图内部都包含两类调控信息(活化和抑制),第 2 层中同一子图中的基因必定相互同步共调控,或者活化或者抑制,不同子图的基因必定异步共

调控,或者活化或者抑制.而且,时间滞后数可以通过两个子图中 Reg-Cluster 的不同起始时间之差得到.对更详细的共调控信息,如活化或抑制可以从第 3 层的子图得到.此时,同一子图内的基因必定相互同步活化共调控,不同子图但是具有相同父亲结点的基因之间必定相互同步抑制共调控.不同子图且具有不同父亲结点的基因之间一定是异步抑制共调控.如果他们都是父亲节点的左(右)孩子,那么他们的关系肯定是活化关系,否则,一个是左孩子,另一个是右孩子,或恰恰相反,那么他们一定是抑制的关系.很明显,起伏趋势相反,相互交叉的基因表达模式在图中频繁出现,这说明 FBLD 算法能同时有效地发现同步和异步共调控模式.相反,以往的基于模式/趋势方法不能发现如图 9 所示的共调控基因聚类.

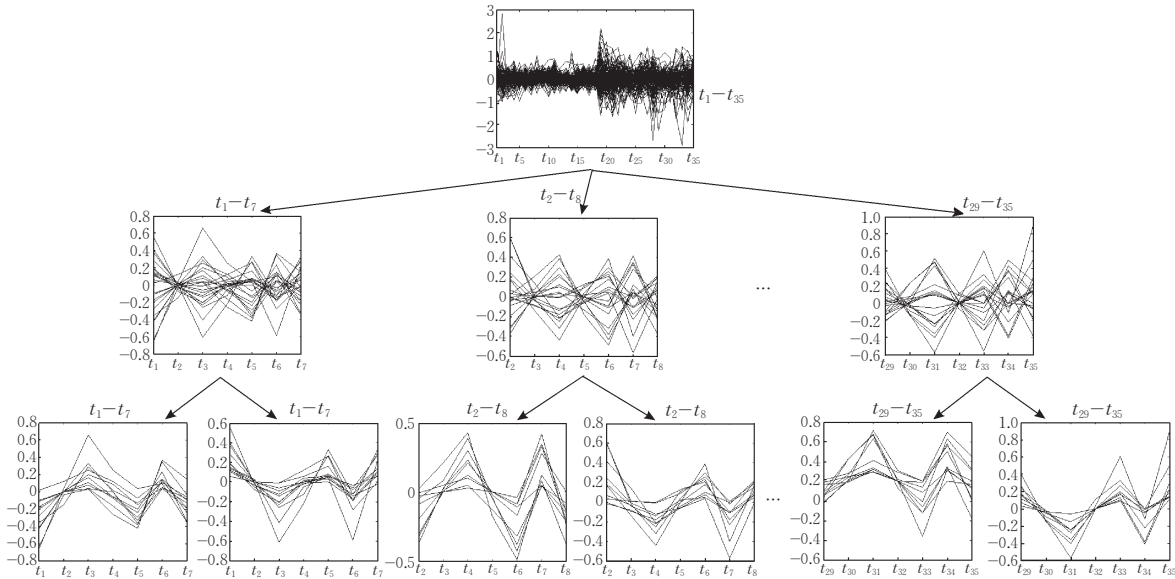


图 8 一个 Reg-Cluster 的层次结构

4.3 参数的有效性

本节讨论不同参数对 Reg-Cluster 模型输出的影响,输出结果主要受参数 min_g , min_t 和调控阈值

δ 的影响.缺省的参数设置为 $min_g = 30, min_t = 5, \delta = 0.01$.图 9 显示了在保持其中两个参数不变,而改变另一个参数值的情况下,在真实的 Yeast GT

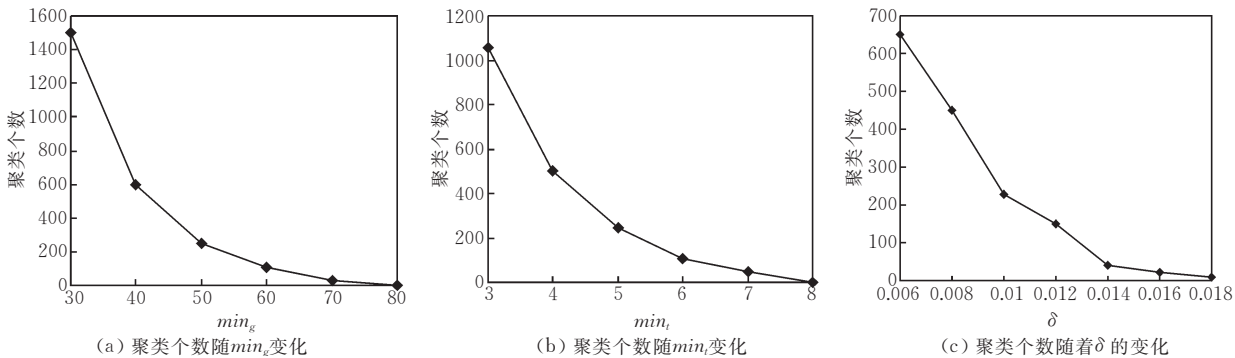


图 9 Reg-Cluster 个数随着 min_g, min_t, δ 变化

数据集上的测试结果. 可以看出, 随着 min_g 的增加, 共调控基因聚类的个数下降, 这是因为相对较少的基因聚类会满足较大的基因值. 图 9(b) 显示了给定 $min_g=30$ 时, 共调控基因聚类个数随 min_t 变动的情况, 结果充分说明了算法的有效性.

更令人感兴趣的是, 图 9 中的 3 条曲线有相似的特性. 也就是说, 在曲线上有“拐点(knots)”. 注意前两条曲线在遇到拐点之前下降的很快, 然后向右平稳的前进. 而第 3 条曲线在拐点之前平稳的上升, 然后急剧的上升. 例如, 可以看到图 9(a) 中的拐点 $min_g=50$, 图 9(b) 中的拐点 $min_t=5$, 图 9(c) 中的拐点 $\delta=0.012$. 这些拐点说明在真实数据集中存在

稳定的和重要的 Reg-Cluster. 它们高度相关, 包含统计重要数目的基因和时间点. 拐点还表明了为了避免一致基因聚类由于偶然而生成的最佳的参数设置.

4.4 扩展到 3 维数据集

Reg-Cluster 挖掘算法可以很容易地扩展到基因-样本-时间三维微阵列数据集上^[5], 构建的三维基因表达矩阵数据集为基因-样本-时间 = $7679 \times 13 \times 24$, 这些原数据是开放的 <http://genome-www.stanford.edu/celcycle/data/rawdata/individual.html>, 通过选择 13 个属性作为样本, 选择 14 个时间点 (0min, 30min, ..., 290min) 作为实验数据得到三维的 Reg-Cluster (见图 10). 其中最小基因

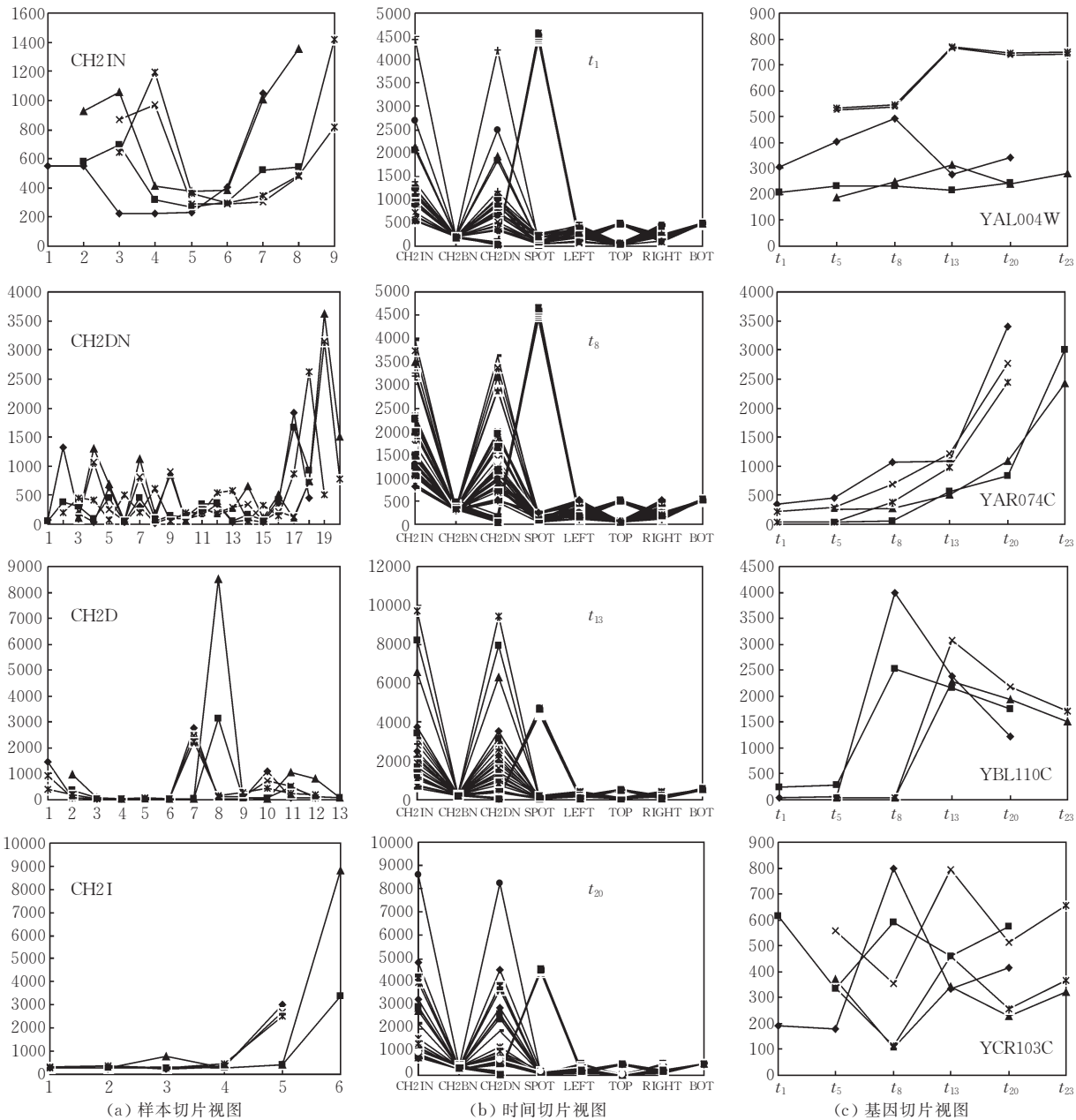


图 10 三维 Reg-Cluster

数、最小样本数和最小时间点数的设置分别为 $min_g=40, min_s=6, min_t=5, \delta=0.01$.

图 10 所示, 为扩展到三维微阵列数据集上的 Reg-Cluster 算法识别的一个 Reg-Cluster, 它的规模是 $32 \times 5 \times 6$. 其中样本集合为 $\{CH2I, CH2D, CH2IN, CH2DN\}$, 时间集合为 $\{t_1, t_5, t_8, t_{13}, t_{20}, t_{23}\}$ 和一系列基因集合 $\{YAL004W, YAR074C, YBL110C, YCR103C, \dots\}$ (因基因数太多, 不一一列举). 该三维 Reg-Cluster 在各个维上的切片分别如图 10(a), 图 10(b) 和图 10(c) 所示, 分别对应样本切片、时间切片和基因切片. 实验结果说明了 Reg-Cluster 算法对扩展到三维基因-样本-时间数据集也是非常有效的.

5 结 论

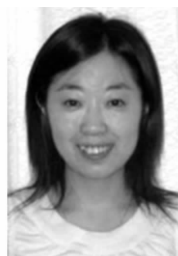
本文提出了一种新的符合生物意义的最大子空间共调控基因聚类模型 Reg-Cluster, 用于同时识别时序微阵列数据集中同步和异步的共调控. 基于提出的编码方案 regCode, 具有任意已知共调控关系 (如同步活化、同步抑制、异步活化、异步抑制) 的基因被聚集在同一类中. 一种“先宽度优先, 后深度优先”的搜索策略和若干有效的削减规则被用来使最大 Reg-Cluster 挖掘过程更高效. 进一步, 更详细和完整的共调控信息 (如活化共调控、抑制共调控及共调控之间的时间滞后数) 可以很容易地从聚类结果中得到, 有助于基因调控网的研究. 而且, 该算法可以被扩展到三维基因-样本-时间微阵列数据集分析. 实验结果证实了提出算法的有效性和高效性.

参 考 文 献

- [1] Liu J Z, Wang W. Op-cluster: Clustering by tendency in high dimensional space//Proceedings of the IEEE International Conference on Data Mining. Melbourne, Florida, USA, 2003: 187-194
- [2] Wang H X, Wang W, Yang J, Yu P. S. Clustering by pat-

tern similarity in large data sets//Proceedings of the ACM SIGMOD Conference Madison. Wisconsin, USA, 2002: 394-405

- [3] Pei J, Zhang X L, Cho M, Wang H X, Yu P S. Maple: A fast algorithm for maximal pattern-based clustering//Proceedings of the IEEE International Conference on Data Mining. Melbourne, Florida, USA, 2003: 259-266
- [4] Wang H X, Chu F, Fan W, Yu P S, Pei J. A fast algorithm for subspace clustering by pattern similarity//Proceedings of the 16th International Conference on Scientific and Statistical Database Management (SSDBM). Santorini Island, Greece, 2004: 51-62
- [5] Zhao L Z, Zaki M J. Tricluster: An effective algorithm for mining coherent clusters in 3D microarray data//Proceedings of the ACM SIGMOD Conference. Baltimore, Maryland, USA, 2005: 694-705
- [6] Xu X, Lu Y, Tung A K H, Wang W. Mining shifting-and-scaling co-regulation patterns on gene expression profiles//Proceedings of the 22nd International Conference on Data Engineering. Atlanta, Georgia, USA, 2006: 89
- [7] Zhao Y H, Wang G R, Yin Y, Yu G Y. A novel approach to revealing positive and negative co-regulated genes//Proceedings of the 6th IEEE Symposiums on Bioinformatics and Bio-Engineering. Arlington, Virginia, USA, 2006: 86-93
- [8] Erdal S, Ozturk K, Armbruster D, Ferhatosmanoglu H, Ray W. A time series analysis of microarray data//Proceedings of the 6th IEEE Symposium on Bioinformatics and Bio-Engineering. Taichung, China, 2004: 366-378
- [9] Yu H, Luscombe N M, Qian J, Gerstein M. Genomic analysis of gene expression relationships in transcriptional regulatory networks. Trends Genet, 2003, 19(8): 422-427
- [10] Cheng Y D, Church G M. Bicustering of expression data//Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology. La Jolla/San Diego, CA, USA, 2000: 93-103
- [11] Zhang Y, Zha H Y, Chu C H. A time-series biclustering algorithm for revealing co-regulated genes//Proceedings of the International Symposium on Information Technology: Coding and Computing. Las Vegas, Nevada, USA, 2005: 32-37
- [12] Jiang D X, Pei J, Zhang A D. Interactive exploration of coherent patterns in time-series gene expression data//Proceedings of the 9th International Conference on Knowledge Discovery and Data Mining. Washington, DC, USA, 2003: 565-570



YIN Ying, born in 1980, Ph. D. candidate. Her research interests include data mining and bioinformatics.

His research interests include data mining and bioinformatics.

ZHANG Bin, born in 1964, professor, Ph. D. supervisor. His research interests include data mining and Information integration.

WANG Guo-Ren, born in 1966, professor, Ph. D. supervisor. His research interests include bioinformatics and XML.

Background

The complexity of a biological system provides a great diversity of correlations among genes/gene clusters, including synchronous and asynchronous co-regulations, each of which can be further divided into two categories: Activation and inhibition. Most existing methods can only identify the synchronous activation patterns, such as shifting, scaling and shifting-and-scaling, however, few focus on capturing both synchronous and asynchronous co-regulations. This paper focuses on identifying synchronous and asynchronous co-regulation patterns simultaneously. Furthermore, the detailed and complete co-regulation information including synchronous/asynchronous activation co-regulation, inhibition co-regulation and the number of time-lag points in genes/gene clusters, which facilitates the study of genetic regulato-

ry networks, can be easily derived from the resulting clusters analysis. This research is supported by National Natural Science Foundation of China (60573089) and the National Key Technologies Research and Development Programming (2004BA721A05). One mission of all these projects is to mine interesting patterns of significant meaning from bio/medical data. The research work of this paper is encouraged by this background and is considered as a significant part of pattern discovery. Bioinformatics is a research direction of data mining group. What they are interested in includes clustering, classification, and association analysis on bio/medical data. Several related papers have been published or accepted by some journals or international conferences.