DENGENE: 一种高精度的基于密度的适用于基因表达数据的聚类算法*

孙 亮[1,4], 赵 芳[2], 王永吉[1,3]
(1. 中国科学院 软件研究所 互联网软件技术实验室, 北京 100080; 2. 香港理工大学 计算学系 生物识别中心, 香港; 3. 中国科学院 软件研究所 计算机科学重点实验室, 北京 100080; 4. 中国科学院 研究生院, 北京 100049)

摘 要: 根据基因表达数据的特点, 提出一种高精度的基于密度的聚类算法 DENGENE。DENGENE 通过定义一致性检测和引进峰点改进搜索方向, 使得算法能够更好地处理基因表达数据。为了评价算法的性能, 选取了两组广为使用的测试数据, 即啤酒酵母基因表达数据集对算法来进行测试。实验结果表明, 与基于模型的五种算法、CAST 算法、K-均值聚类等相比, DENGENE 在滤除噪声和聚类精度方面取得了显著的改善。

关键词: 基因表达数据; 聚类分析; 基于密度的聚类; 一致性检测; 峰点
中图分类号: TP18; TP301.6 **文献标志码**: A **文章编号**: 1001-3695(2007)04-0058-04

DENGENE: High Accurate Density-based Clustering Algorithm for Gene Expression Data

SUN Liang[1,4], ZHAO Fang[2], WANG Yong-ji[1,3]
(1. *Laboratory for Internet Software Technologies, Institute of Software, Chinese Academy of Sciences, Beijing* 100080, *China*; 2. *Biometrics Research Centre, Dept. of Computing, Hongkong Polytechnic University, Hongkong China*; 3. *Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing* 100080, *China*; 4. *Graduate School, Chinese Academy of Sciences, Beijing* 100049, *China*)

Abstract: According to the characteristics of gene expression data, a high accurate density-based clustering algorithm called DENGENE was proposed. DENGENE achieves good clustering by defining homogeneity test and peak points. To evaluate the performance of DENGENE, two budding yeast Saccharomyces cerevisiae data sets, which are widely used as test data sets, were used to validate the effectiveness of DENGENE. The experiment results show that compared with five model-based clustering algorithms, CAST and K-means clustering, DENGENE filters noises effectively and produces more accurate clustering results.

Key words: gene expression data; cluster analysis; density-based clustering; homogeneity test; peak point

0 引言

DNA 微阵列技术(DNA Microarray Technology)的迅速发展导致了基因表达数据(Gene Expression Data)的爆炸性增长。因此有效地分析这些基因表达数据成为当前生物信息学面临的主要问题之一。一般来讲, 基因表达数据是一个矩阵。矩阵的每一行代表一个基因的表达向量, 即该基因在多种实验条件下测得的表达数据; 每一列对应于一种实验条件下测得的所有基因的表达值。

目前, 聚类分析已广泛应用于基因表达数据分析中[1]。聚类分析将表达模式相似的基因聚在一起, 从而有助于基因的分析。①表达模式相似的基因具有相同的功能。从而能够从属于同一聚类的某些已知功能的基因推断该聚类中其他基因的功能[2]。②具有相似表达模式的基因更有可能处于相同的细胞过程中。因此聚类结果有助于基因转录调控网络中调控机理的研究[1]。

在对基因表达数据聚类时, 必须充分考虑基因表达数据本身的特点。一般来讲, 对基因表达数据进行聚类分析时有如下特点:

(1) 与数据挖掘中的聚类算法相比, 在对基因表达数据进行处理时更关注于算法的有效性(Effectiveness)而非效率(Efficiency)。基因表达数据的规模一般为数千级或数万级, 完全可在主存中处理。注意, 并不是说忽视效率, 而是指在效率允许的情况下尽量提高算法精度。

(2) 基因数据中存在大量噪声, 所以好的基因聚类算法必须能对噪声具有很强的鲁棒性。在基因表达数据分析中, 聚类的目的并不是为了得到所有对象的相互关系(如层次聚类所作), 而是取得那些基本有用的数据, 所以噪声必须去

收稿日期: 2006-02-24; 修返日期: 2006-04-15 基金项目: 国家自然科学基金资助项目(60373053); 中国科学院"百人计划"基金资助项目; 中国科学院与英国皇家学会联合资助项目(20030389, 20032006); 留学回国人员科研启动基金项目([2003]406)

作者简介: 孙亮(1981-), 男, 湖北天门人, 硕士研究生, 主要研究方向为生物信息学、数据挖掘(sunliang@itechs.iscas.ac.cn); 赵芳(1970-), 女, 浙江兰溪人, 博士研究生, 主要研究方向为生物信息学、分子生物学; 王永吉(1962-), 男, 辽宁盖州人, 研究员, 博导, 主要研究方向为生物信息学、数据挖掘、实时系统、人工智能、网络优化.

除。

对基因表达数据进行处理时, 要求算法能够有效地滤除噪声并取得高精度的聚类结果。

根据基因表达数据的以上特点, 基于密度的聚类算法(Density-based Clustering) 是一种较好的解决方案。在数据挖掘领域中, 已有大量的基于密度的聚类算法。其中以 DBSCAN[3] 最为经典。此类方法能够得到任意形状的聚类, 并且可以很方便地处理噪声数据。但是 DBSCAN 最大的缺陷是要求指定全局密度阈值。当阈值过大时, 仅能算出最主要的几个聚类而将其他点视为噪声; 阈值过小时, 又易于合并不应合并的聚类。在基因表达数据分析中, 经常存在聚类相交或嵌入的情况, 使得直接将 DBSCAN 应用于基因表达数据的效果并不理想。当使用 DBSCAN 处理一组该领域内经典的啤酒酵母基因表达数据[2] 时, 如果密度阈值太小, 则两个主要的聚类, 即具有糖酵解(Glycolysis) 功能的聚类和具有蛋白质合成(Protein Synthesis) 功能的聚类则会合并在一起; 如果密度阈值较大, 虽然能够分开上述两个聚类, 但是大量其他基因均被当成噪声滤除了。

本文根据基因表达数据的特点, 提出了一种高精度的基于密度的聚类算法(Density-based Clustering Using Homogeneity Test, DENGENE)。其基本思想如下: ①用密度(Density) 来勾勒数据分布的疏密程度, 使之能够合理地处理噪声。②规定密度大于某一阈值的点为核心点(Core Point)。因为核心点是聚类的骨架, 这样算法能够得到任意形状的聚类。③ DENGENE 通过使用一致性检测(Homogeneity Test) 有效提高了聚类结果的精确度, 克服了 DBSCAN 关于全局阈值的缺陷。④定义了峰点(Peak Point), 也就是密度比给定邻域内点都大的核心点。聚类时不像 DBSCAN 一样可从任意一个核心点出发, 而是只能从峰点出发, 且聚类的每次扩展均只能沿密度最大的方向进行扩展。

为了评价算法性能, 选取了两组在此领域内非常经典的啤酒酵母基因表达数据集[2,4] 来进行测试。由于啤酒酵母基因的功能已知, 且其基因表达数据和功能都可在公共数据库中得到, 是广为使用的测试数据。测试结果显示 DENGENE 相比以往的算法性能有很大的提高。

1　DENGENE 算法

进行聚类处理的基因表达数据是一个 $m \times n$ 维矩阵 X, 它表示 m 个基因在 n 种不同条件下测得基因表达值, 即每个基因可以认为是 n 维空间内的一个点, m 个点组成集合 D, 点 p 和 q 的距离表示为 $\mathrm{dist}(p, q)$。

1.1　基本思想

基于密度的聚类算法将聚类看成是数据空间中被低密度区域分割开的高密度区域。下面介绍 DENGENE 的主要思想。

(1) DENGENE 用密度来勾勒数据分布的疏密程度。有如下定义:

定义 1　(eps 邻域) 点 p 的 eps 邻域定义为 $N_{\mathrm{eps}}(p) = \{q \in D \mid \mathrm{dist}(p, q) \le \mathrm{eps}\}$

定义 2　(点的密度) 对 D 中任意点 p 和距离 eps, p 的 eps 邻域内的点数称为点 p 基于距离 eps 的密度, 记为 $\mathrm{density}(p, \mathrm{eps})$, 即 $\mathrm{density}(p, \mathrm{eps}) = |N_{\mathrm{eps}}(p)|$。在规定了 eps 时, 可简写为 $\mathrm{density}(p)$。

(2) 密度大于某一阈值 t 的点称为核心点(Core Point)。使用核心点来代表聚类, 从而能够得到任意形状的聚类。事实上, 核心点是聚类的骨架。聚类时主要集中在核心点的处理上。对于非核心点, 计算它到离它最近核心点的距离。如果大于邻域半径则视为噪声; 否则将它合并到离它最近核心点所对应的聚类中。这种利用骨架进行聚类扩展的原理使得基于密度的方法能够得到任意形状的聚类, 即

定义 3　(核心点) 对于给定的邻域半径 eps, 密度大于等于 MinPts 的点称为核心点。

(3) DENGENE 通过使用一致性检测有效提高聚类结果的精确度。如前所述, 在处理基因表达数据时, 基于密度的算法的最大优点就是可以有效滤除噪声。但是传统的基于密度的算法(如 DBSCAN) 并不能满足高精度的要求, 在引言中也提到了 DBSCAN 处理典型的基因表达数据集的问题。这里用一个二维数据分布的简单例子来说明这个问题, 如图 1 所示。图 1 中有三个聚类 C_1、C_2、C_3。其中 C_1、C_2 的密度较大, C_3 的密度较小。当使用较大的全局密度阈值时, 只有聚类 C_1 和 C_2 可以被检测出来, C_3 则被视为噪声; 如果用较小的全局密度阈值使得 C_3 能检测出来, C_1 和 C_2 则合并为一个聚类。这就是使用 DBSCAN 处理基因表达数据的常见问题。

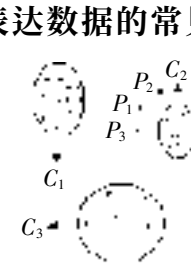

图 1　二维空间的一个例子

DENGENE 使用一致性检测克服了上述缺陷, 能够精确地同时检测出 C_1、C_2 和 C_3。当使用较小的密度阈值检测出 C_3 时, 点 P_1 也成为核心点。但是可以清楚看到, P_1 与相邻聚类 C_1、C_2 的性质明显不一样。事实上可用一些指标来度量 C_1 及 C_2 的性质。由于 $P_1 \sim P_3$ 并不具有与 C_1、C_2 一致的性质, 不应将 $P_1 \sim P_3$ 加入 C_1、C_2 中。这样 C_1 和 C_2 就不会合并。这里将检查新的核心点是否具有与要加入的聚类同一性质的措施, 称为一致性检测。在具体实施一致性检测时, 首先要提取出能度量聚类一致性的具体指标。同时, 为了更好地得到聚类性质, 规定生成聚类时必须从密度较大的地方开始, 再向密度较低的地方扩展, 且每一步扩展均必须沿当前密度最大的方向。这样能够更加准确地获取聚类的性质。事实上, 在 DENGENE 框架下可有多种方式来定义聚类的一致性。下面列出两种具体措施, 并假设当前聚类为 C, 进行一致性测试的核心点为 P。

措施 1　通过定义聚类的当前平均密度来度量一致性

①计算 C 的当前平均密度:

$$\mathrm{avDensity}(C) = \sum_{p \in C} \mathrm{density}(p) / |C|$$

②如果 $\mathrm{density}(P) \ge \alpha \times \mathrm{avDensity}(C)$, 则将 P 加入 C 中; 否则不加入。这里 α 是系数, 由用户设定。

由于 DENGENE 是基于密度的算法, 计算聚类 C 的平均密度简单易行、计算复杂度低。α 的值越小, 则聚类扩张的区域越大, 反之亦然。一般地, 为了能够扩张, 要求 $\alpha < 1$。事实上 $\alpha = 0$ 就是 DBSCAN 算法。在图 1 的例子中, 虽然 P_1 在较小的

密度阈值下成为核心点,但是它的密度值与 C_1、C_2 的平均密度值都相差甚远,即 P_1 不能通过一致性检测,不应将 P_1 合并到 C_1 或 C_2 中。这样 DENGENE 得到了三个聚类: C_1、C_2 和 C_3。

措施 2 通过考查核心点 P 的邻域中已合并入 C 的点数来进行一致性检测

①计算点 P 的 eps 邻域中已合并入 C 的点数 N。

②如果 $N/\mathrm{density}(P) \geq \beta$ 则将点 P 合并到 C 中; 否则不予合并。这里 β是系数,由用户设定。

措施 2 的基本原理如下: 根据密度的定义, density(P) 实际上是点 P 的 eps 邻域内的所有点的数目。由于聚类首先都是从最密的地方开始生成,如果 P 与 C 联系较紧密,则 P 的邻域中已有相当数量的点被合并入聚类 C 中。这里用 β作为阈值 (要求 $\beta<1$) , β的值越小,则聚类扩张的区域越大,反之亦然。事实上 $\beta=0$ 就是 DBSCAN 算法。

注意, DENGENE 只在聚类扩展的过程中对新的核心点做处理时才进行一致性检测。在具体使用 DENGENE 时,可采用措施 1 或 2, 或两者并用。

(4) DENGENE 定义了峰点,以适应新的搜索方式。峰点是密度比给定邻域内的点都大的核心点。DENGENE 从峰点开始进行聚类扩展,将相互距离小于 eps 且通过一致性检测的核心点合并进该聚类中,完成骨架的聚类。如果没有相应的核心点可再进行扩展,则结束该聚类的处理,转入下一个聚类的处理。这样有如下定义:

定义 4 (峰点) 满足以下两个条件的点 p 称为峰点:

①p 是核心点;

②不存在核心点 q 使得 $\mathrm{density}(q) > \mathrm{density}(p)$ 且 $q \in N_k(p)$。$N_k(p)$ 是 p 的邻域,用来确定峰点比较范围。

峰点就是 $N_k(p)$ 内密度最大的核心点。

1.2 算法实现

这里用伪代码来描述算法的具体实现:

```
specify parameters: MinPts, eps, Nk( p) and α;
compute and store the eps-neighborhood for each point;
clusterId[ all] =0;
clusterNo = 0;
for i = 1: D. size
   p = D[ i] ;
   if p is a peak point && clusterId( p) = = 0
   coreList. setEmpty( ) ;
   clusterNo ++ ;
   coreList. insert( p) ;
   while( coreList! = empty)
      c = maxDensity( coreList) ;
      coreList. remove( c) ;
      find the eps-neighborhood NC of c;
      compute the un-processed points Nu within NC;
      compute the un-processed core points Nuc within NC;
      t = homogeneity_test( c) ;
      if t = = true
         clusterId[ Nu] = clusterNo;
         coreList. insert( Nuc) ;
         endif
   endwhile
   endif
endfor
```

算法首先设置好相关参数。其中参数 MinPts 和 eps 用来计算密度及确认核心点; $N_k(p)$ 用来确定峰点时的比较范围 (参见定义 4) ; α为进行一致性检测的控制参数。D 表示所有点组成的集合。D. size 用来表示该集合中点的总数。算法为每个点计算了相应的 eps 邻域,但必须先计算每对点之间的距离。其复杂度为 $O(m^2)$。由于数据规模不大,且每个点的 eps 邻域所包含的点不多(一般只有几个) , 为每个点保存了其 eps 邻域。clusterId 用来存储聚类结果: clusterId[k] = 0 表示基因 k 是噪声, clusterId[k] = j ($j > 0$) 表示基因 k 属于第 j 个聚类。最初通过 clusterId[all] = 0 将所有基因都初始化为 0。clusterNo 用来表示当前聚类的编号, coreList 用来表示聚类扩张时核心点的集合。c = maxDensity(coreList) 表示从 coreList 中找出密度最大的核心点 c。homogeneity_test(c) 表示对点 c 进行一致性检测,结果为 True 表示通过, False 表示没有通过。

每次聚类都是从峰点出发,如果其已经被聚类则不予处理; 否则将其加入 coreList 中。每步从 coreList 中取出密度最大的核心点作为当前处理的核心点,这样保证了聚类的扩展始终沿着密度最大的方向进行。程序中将 c 视为当前处理的核心点, N_C 是 c 的 eps 邻域, N_u 表示 N_C 中未处理的点, N_{uc} 表示 N_C 中未处理的核心点。程序将 N_{uc} 加入到 coreList 中,以便将来用于扩展聚类,并将当前核心点邻域内所有未处理的点合并入当前聚类中。当没有相应的可以扩张的核心点时则 While 循环结束,即该聚类完成,进入下一个聚类的处理中。

设 D 中核心点的数目为 N, 由于 While 循环每次执行从 coreList 中删除一个核心点, While 的执行次数最多为 N。且由于记录了每个点的 eps 邻域,其计算复杂度很低。前面计算距离和各点的 eps 邻域的复杂度是 $O(m^2)$, 则算法总的复杂度是 $O(m^2)$。这里 m 是待处理的基因总数,这在处理基因表达数据时是可接受的。

2 测试数据描述及算法性能评价

为了评价算法性能,采用了两组广为使用的啤酒酵母基因表达数据集对算法进行测试,并对 DENGENE、K-均值算法[5]、基于模型的五种算法[6]和 CAST[7]算法进行了性能比较。第一组数据含有大量噪声,通过比较 DENGENE 和 K-均值算法对该组数据的处理,着重说明 DENGENE 对于噪声的鲁棒性; 第二组数据所含噪声较少但分类明确,通过比较 DENGENE 和基于模型的五种算法、CAST 算法对该组数据的聚类结果,着重说明 DENGENE 所得的聚类结果的高精确度。在对基因表达数据的处理中, DENGENE 的一致性检测只采用了措施 1, 其控制参数为 α。

2.1 测试数据描述

第一组数据是文献[2] 中使用的数据集,该数据集是2 467 个基因在 79 种实验条件下测得的表达数据。其中的基因都有相应的功能注解。该数据集可在 http://genome-www.stanford.edu/clustering/下载。该数据集的特点是含有大量噪声; 其生物学特征是该数据集中存在大量基因; 其功能与其他基因截然不同。换言之,将这些基因放到任何一个聚类中都没有生物学上的显著意义,所以可将其视为噪声。

第二组数据集是文献[4] 中使用的数据集。本文使用的是文献[6] 中提取的一个注释完全的子集。该组数据的特点是噪声少且分类明确: 它是 384 个在不同时间点达到峰值的基因, 在 17 个时间点的基因表达数据。该数据集对应细胞周期

的五个不同阶段,因而可根据基因达到峰点所在细胞周期的不同将该子集划分为五个聚类。由于该组数据的第 10、11 时间点的数据测量不可靠,通常的处理是将这些不可靠的数据删去[5],以提高聚类效果。本文在预处理时也删除这两列。在 http://www.cs.washington.edu/homes/kayee/model 可下载这两个子集。

为简化起见,将第一个数据集合称为 Eisen2000,将第二个数据集称为 Cho384。

2.2　实验结果比较和分析

2.2.1　Eisen2000 数据

删去 Eisen2000 数据集中表达值为空的基因,得到进一步处理的对象: 605 基因在 79 个时间点的表达值。

在使用 DENGENE 对 Eisen2000 进行处理时,设 eps = 15, MinPts = 4,查找峰点的区域 N_k 半径是 N_{eps} 半径(即 eps)的 0.6 倍, α = 0.7,得到 10 个聚类。在使用 K-均值算法对 Eisen2000 进行处理时,将聚类数目 K 设置为 30。

本文着重分析两种算法处理结果对应的生物学意义。在 DENGENE 生成的 10 个聚类中,聚类 1 基本上是蛋白质合成功能;聚类 4 全是染色质结构(Chromatin Structure)功能;聚类 6 基本上是糖酵解功能;聚类 8 是 ATP 合成(ATP Synthesis)及氧化磷酸化(Oxidative Phosphorylatio)功能;聚类 9 全是蛋白质降解(Protein Degradation)功能;聚类 10 全是蛋白质合成功能。其成功地找出了文献[2]中用手工方法找出的聚类。而在 K-均值算法生成的 30 个聚类中,只有 2 个聚类中基因的功能大体相同。事实上,K-均值算法得到的聚类中大部分被噪声污染了,而 DENGENE 由于滤除了噪声,得到的聚类基本上具有对应的生物学功能。

2.2.2　Cho384 和 Cho237 数据

由于已知 Cho384 和 Cho237 的真正分类结果,则可用修正的 Rand 指数(Adjusted Rand Index)[8] 来衡量算法的精确度。修正的 Rand 指数值越高,则聚类结果越接近真正的分类结果,算法的精确度越高。

图 2 是基于模型的五种算法(包括 EI、VI、VVV、Diagonal、EEE)[6] 对 Cho384 的聚类结果。文献[9]通过比较层次聚类算法、K-均值算法和 CAST 算法,认为 CAST 算法的性能最优。图 2 中也列出了 CAST 算法。图 2 的横坐标是聚类的数目,纵坐标是修正的 Rand 指数值,每条曲线表示相应的算法性能。对于具体算法,使得其修正的 Rand 指数最大的聚类数目就是对该算法最合适的聚类数目。在图 2 列出的各算法中,CAST 算法和基于模型的算法 EI 表现最好,并在取 6 个聚类时得到最好的聚类结果,对应的修正 Rand 指数稍大于 0.5。

由于在使用 DENGENE 时无须设置聚类数目,图 2 中没有列出 DENGENE 的性能曲线,而在图 3 中给出了 DENGENE 处理 Cho384 的结果。图 3 显示了参数设置为 eps = 0.1 ~0.2, MinPts = 4 ~7,将 $N_k(p)$ 设置为 $N_{eps}(p)$, α = 0.7 时的处理结果。图 3 中的横坐标是 eps 值,纵坐标是对应的修正 Rand 指数之值,四条不同的曲线对应于 MinPts = 4 ~7 时的处理结果。图 3 中 DENGENE 所得的聚类结果对应的修正 Rand 指数基本上都在 0.45 以上。最好的聚类结果是当 eps = 0.1, MinPts = 5 时,将 Cho384 分为 5 个聚类,其修正的 Rand 指数为 0.666。

通过图 2 和 3 的比较,DENGENE 相比基于模型的五种算法和 CAST 算法,聚类结果对应的修正 Rand 指数大为提高,充分显示了 DENGENE 算法的高精确度。

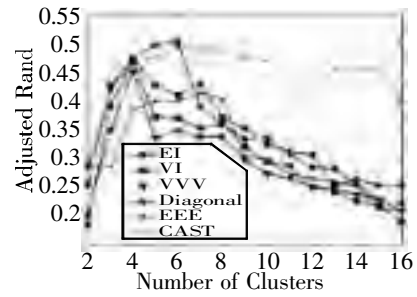

图 2　其他算法对Cho384的聚类结果

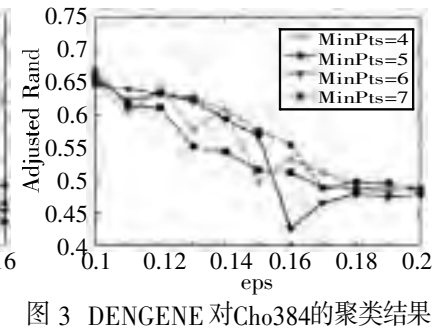

图 3　DENGENE 对Cho384的聚类结果

2.3　参数设置

一般的,eps 的取值可以参照 DBSCAN 算法[5]。结合基因表达数据处理的实际情况,MinPts = 4 ~6 均可满足实际需要。在上述实验中,一致性检测的方式都是采取第一种措施,参数 α = 0.7 都可以取得较好的效果。事实上由于聚类扩张时是从最密的地方开始,为了能够顺利扩张聚类,α 的取值一般小于 1。对于峰点半径,一般取稍微小于 eps 即可。

3　结束语

本文根据基因表达数据分析的特点,提出了一种高精度的基于密度的聚类算法 DENGENE。算法从峰点出发进行聚类的扩展,同时引入同一性检测以判定是否将当前核心点加入到当前聚类中。与已有的适用于基因表达数据的聚类算法相比,DENGENE 能有效地从高噪声的基因表达数据中提取出生物学信息,同时具有较高的精确度。在两组经典的啤酒酵母基因表达数据集上的测试也证实了上述结论。

参考文献:

[1] IANG Daxin, TANG Chun, ZHANG Aidong. Cluster analysis for gene expression: a survey[J]. **IEEE Transactions on Knowledge and Data Engineering,** 2004, **16**(11): 1370-1386.

[2] EISEN M B, SPELLMAN P T, BROWN P O, *et al*. Cluster analysis and display of genome-wide expression patterns[J]. **Proc. of National Academy Science,** 1998, **95**(25): 14863-14868.

[3] ESTER M, KRIEGEL H, SANDER J, *et al*. A density-based algorithm for discovering clusters in large spatial databases with noise [C]. Portland: AAAI Press, 1996: 226-231.

[4] CHO R J, CAMPBELL M J, WINZELER E A, *et al*. A genome-wide transcriptional analysis of the mitotic cell cycle[J]. **Mol. Cell,** 1998, **2**(1): 65-73.

[5] TAVAZOIE, HUGHES J D, CAMPBELL M J, *et al*. Systematic determination of genetic network architecture[J]. **Natural Genetics,** 1999, **22**(3): 281-285.

[6] YEUNG K Y, FRALEY C, MURUA A, *et al*. Model-based clustering and data transformations for gene expression data[J]. **Bioinformatics,** 2001, **17**(10): 977-987.

[7] BENDOR A, SHAMIR R, YAKHINI Z. Clustering gene expression patterns[J]. **Journal of Computational Biology,** 1999, **6**(3/4): 281-297.

[8] MEWES H W, HEUMANN K, KAPS A, *et al*. MIPS: a database for protein sequences and complete genomes[J]. **Nucleic Acids Research,** 2002, **30**(1): 31-34.

[9] HUBERT L, ARABIE P. Comparing partitions[J]. **Journal of Classification,** 1985, **2**(1): 193-218.

[10] YEUNG K Y, HAYNOR D R, RUZZO W L. Validating clustering for gene expression data[J]. **Bioinformatics,** 2001, **17**(4): 309-318.