一种基于主集分割的基因芯片聚类算法*

滕 莉[1+], 付旭平[2], 李宏宇[1], 李 瑶[2], 陈文斌[3], 李荣宇[4], 沈一帆[1]

[1](复旦大学 计算机科学与工程系,上海 200433)

[2](复旦大学 生命科学学院 遗传研究所,上海 200433)

[3](复旦大学 数学系,上海 200433)

[4](上海博星基因芯片有限责任公司,上海 200092)

A Microarray Cluster Algorithm Based on Dominant Set Segmentation

TENG Li[1+], FU Xu-Ping[2], LI Hong-Yu[1], LI Yao[2], CHEN Wen-Bin[3], LI Rong-Yu[4], SHEN Yi-Fan[1]

[1](Department of Computer Science and Engineering, Fudan University, Shanghai 200433, China)

[2](Institute of Genetics, School of Life Science, Fudan University, Shanghai 200433, China)

[3](Department of Mathematics, Fudan University, Shanghai 200433, China)

[4](Shanghai BioStar Genechip Inc., Shanghai 200092, China)

+ Corresponding author: Phn: +86-852-60801741, E-mail: tengli.hust@263.net, http://www.cse.cuhk.edu.hk/~lteng/

Received 2004-05-31; Accepted 2005-02-04

Teng L, Fu XP, Li HY, Li Y, Chen WB, Li RY, Shen YF. A microarray cluster algorithm based on dominant set segmentation. *Journal of Software*, 2005,16(9):1591−1598. DOI: 10.1360/jos161591

Abstract: Clustering algorithms are wildly used in the research of microarray data to extract groups of genes or samples that are tightly coexpressed. In most of them, some parameters should be predefined artificially, however, it is very difficult to determine them manually without prior domain knowledge. To handle this problem, an iterative clustering algorithm is proposed. Firstly, by sorting the original data by dominant set, similar genes would be aligned together. It's hard to specify the cluster boundary. A criterion is presented to partition a cluster from the sorted data according to the property that the distances between the inside elements are smaller than that of outside elements. The idea is to remove the cluster form the current data set, repeat the process, and stop the algorithm when the stop criterions are satisfied. The new clustering algorithm is analyzed on several aspects and tested on the published yeast cell-cycle microarray data. The results of the application confirm that the method is very applicable, efficient and has good ability to resist noise.

Key words: microarray; dominant set; clustering; coexpressed; sorting

摘 要: 聚类算法广泛应用于生物芯片数据分析中,用于寻找表达相似的基因或样本.大多数已有算法都需要

* Supported by the National Natural Science Foundation of China under Grant No.60473104 (国家自然科学基金)

作者简介: 滕莉(1980－),女,山东莒南人,硕士,主要研究领域为生物信息,聚类算法;付旭平(1975－),男,博士,讲师,主要研究领域为分子生物学;李宏宇(1980－),男,博士生,主要研究领域为图像分割,数据聚类;李瑶(1965－),女,博士,教授,博士生导师,主要研究领域为基因芯片技术及其功能基因组;陈文斌(1970－),男,博士,副教授,主要研究领域为并行计算,快速算法;李荣宇(1973－),男,硕士,工程师,主要研究领域为基因芯片生物学;沈一帆(1965－),男,博士,教授,博士生导师,主要研究领域为科学计算,可视化.

人为地给出一些参数,然而在没有先验知识的情况下,人为地确定这些参数是十分困难的.为了解决这一难题,提出了一种迭代的聚类算法.首先用主集方法对原有基因进行重新排序,使高度相似的基因排列在特定区域.类的分割界线通常难于确定.提出一种标准,根据类内元素间的距离远小于类外元素间的距离的性质,从排序后的数据集中划分出一个类.将找到的类从当前数据集中排除以后,对剩下的数据重复以上处理,直到满足所提出的循环停止条件为止.从多方面分析了该算法的性能,并将该算法应用于酵母细胞周期的芯片表达谱数据聚类.理论分析和应用结果都表明,该算法是实用、有效的,并且有很好的抗噪性能.

关键词: 基因芯片;主集;聚类;相关表达;排序
中图法分类号: TP181 文献标识码: A

基因芯片是近年发展起来的一种分子生物学高新技术,在生命科学研究中有着广泛的应用前景.基因芯片技术应用过程中产生大量的数据,如何处理和分析这些数据并从中提取出有价值的生物学信息,是一个极为重要的问题.聚类是一个最为常用的工具,多种方法,例如 Self-Organizing Maps[1],hierarchical clustering[2,3], Self-Organizing Tree Algorithm[4],*K*-means[5],simulated annealing[6],Cluster Affinity Search Technique[7], Quality-based 聚类算法[8]等,已成功地应用于高维基因芯片数据的分析[9−11]中.大多数聚类算法起源于非生物相关的研究领域,在实际应用中存在着一些不足之处[12].例如,*K*-means 算法、Self-Organizing Maps 需要预先定义类的个数(算法的参数),而在对基因表达谱进行聚类时,类的个数通常是未知的,改变这一参数往往会极大地影响聚类结果;层级聚类结果通常用树状图表示,虽然能够表示基因之间相似的层次关系,但不能直接得到相似基因组成的类.并且在绝大多数方法中,所有基因都被划分到某个类中,即便是某条基因与类中其他基因的相似性并不高,但由于与其他类相似性更低,因此只能被归并到这个类中,造成了该类的"噪声".一个类的总体表达效果将受噪声影响,这样的结果也将不利于进一步的分析(例如,motif 的寻找或基因功能预测).

本文给出一种新的聚类算法,减少了用户给定参数,并具有良好的性能,我们称其为主集分割法(Ds_Clust).从本质上讲,下述算法是一个启发式的两步方法,依次确定各个类(类的个数预先不知道,所以这不是算法的参数).第 1 步借用主集的概念对原始数据进行重排序,定位一个类;第 2 步从数据性质出发确定这个类.结果以各个类的形式单独存在,同一类中每个元素都非常相似,类内噪声很小,而类间相似程度不高.该方法不需要预先定义类的数目,抗噪能力强.

本文第 1 节简要介绍有关基因芯片数据聚类的有关知识.第 2 节详细论述本文提出的聚类算法,然后从多个方面比较该算法和一些常用方法,分析算法性能,并用实验验证其可行性.第 3 节将本文提出的算法应用于公用酵母细胞周期数据聚类,以考核其性能.最后是结论以及今后的工作.

1 基因芯片数据聚类

聚类算法广泛应用于基因芯片数据分析中.聚类的目的是按照某种相似性判断标准,将那些具有相同性质的基因表达谱向量聚集起来,同时将性质不同的基因表达谱向量分开.基因数据聚类的结果是找到具有高相似度(称为相关表达)的表达谱基因的类.可以把每个基因的表达谱看作实数向量,其中各元素为某一特定基因在不同实验条件下的表达水平.基因芯片数据可以用一个实数矩阵 G 表示.矩阵 G 每行代表一种基因,每列代表一种实验条件.矩阵中元素 $G(i,k)$ 表示基因 i 在条件 k 下的表达值.基因之间的相互关系可以用一个关系矩阵 $A=(a_{ij})$表示,a_{ij} 表示基因 i 和基因 j 的相似性,相似性可以通过欧氏距离或者皮尔斯相关系数等来衡量.

2 算法介绍与分析

2.1 主集法简介

通常,类是指一个元素集合,它应该满足两个基本条件:类内元素的相似性高和类间元素相似性低.当集合内的元素表示带权图的顶点时,类的概念就等价于这样一个集合:连接该集合内部顶点的边权值较大,而连接不同集合间的顶点的边权值较小.在这里,主集的概念和类的定义是相似的.下面我们首先介绍一下主集的概念.

2.1.1 主集的定义(dominant set)

主集的使用由来已久,但是一直没有正式的定义.2003 年,Pavan 和 Pelillo 将此方法用于图像分割,并在文献[13]中给出了主集的确切定义.文中指出,对于一个带权无向图 $G=(V,E)$,按照边上定义的权值在定点集合 V 上寻找主集 S.首先 S 应该是 V 的一个子集,同时,它必须满足以下两个条件: S 内部的元素具有相同性质, S 与 S 外的元素之间性质不同.从这个意义上来讲,主集的概念也就等同于类的概念.直接按定义去求取主集非常繁琐,Pavan 和 Pelillo 提出了一种更为有效的方法——二次型法,将寻找主集的问题转化为用迭代的方法求解单纯形的极大值问题,这里选用的迭代法是理论生物学和进化论中称为繁殖方程(replicator equation)的方法.

2.1.2 通过求解二次型极值来求取主集

对于一个带权无向图 $G=(V,E)$, V 是所有顶点的集合, E 是所有边的集合,它的权矩阵可以用 W 来表示.下面考虑一个带约束条件的极值问题,最大化:

$$f(x)=x'Wx,\quad x\in\Delta \tag{1}$$

这里, $\Delta=\{x\in R^n: x_i\geq 0 \text{ for all } i\in V \text{ and } e'x=1\}$,这个约束条件称作 R^n 上的标准单纯形, e 是全 1 列向量.

向量 x 的支持集可以定义为向量 x 中非零元素的下标集合,也就是说,

$$\sigma(x)=\{i\in V: x_i>0\} \tag{2}$$

从文献[13]中,我们可以得到下面的定理.它在主集和二次型(1)的局部解之间建立了联系,因此可以通过求解二次型的局部最大值来求解主集.

定理 **1**. 如果 S 是顶点集合的一个主子集,那么它的加权特征向量就是二次型(1)的严格局部极值解;相反,如果 x 是二次型(1)的严格局部极值,那么向量 x 的支持集 $\sigma(x)$ 所对应的顶点就构成一个主集.

其具体推导过程可参见文献[13].在标准单纯形上求二次型极大值的最直接的方法就是繁殖方程法.它源于进化游戏论(evolutionary game theory).这种方法的优点是,它能够用简单的几行高级编程语言实现.繁殖方程法有两种常用形式:离散和连续时间的动态系统.本文使用离散动态方程:

$$x_i(t+1)=x_i(t)\frac{(WX(t))_i}{X^T(t)WX(t)} \tag{3}$$

其中 $x_i(t)$ 是向量 $X(t)$ 的分量, t 是离散化后的时间.我们把 t 作为迭代步数,在应用过程中,令 $W=A$ (见后).我们注意到,二次型(1)的局部最优解 x' 实际上给我们提供了类的隶属度信息, x 的分量大小表示分量对应的点属于当前类的可能性.Pavan 和 Pelillo 使用向量 x 的支持集作为分割标准,然而,以此标准进行分类是不准确的.可以利用向量 x 中所包含的类的隶属度信息来设计新的聚类方法,具体步骤将在第 2.2 节中详细加以介绍.

2.1.3 关系矩阵 A 的计算

上述过程的推导和计算都是基于关系矩阵 A (affinity matrix)的. A 是一个对称矩阵,本文中我们采用以下的指数形式定义计算关系矩阵,其元素 a_{ij} 表示对应两个点 i 和 j 的相关性,其值越大,表示两点之间的相关性越大;相反,点的相关性就越小:

$$a_{ij}=\exp\left(-\frac{\|g_i-g_j\|}{\delta}\right) \tag{4}$$

其中 $\delta>0$ 是缩放因子,用来起调节作用,以控制聚类敏感度. g_i 和 g_j 分别表示第 i 和 j 个基因数据向量,实际上就是基因矩阵 G 的第 i 和 j 行.

2.2 主集法的改进与提高

2.2.1 基因数据排序

如上所述,向量 $X(t)$ 的分量可以表示类的隶属度信息,每一分量 $x_i(t)$ 对应于原数据集 G 中的一条基因向量,数值越大的分量间相似性越高.按照 $x_i(t)$ 的值从大到小对原始基因进行排序,对应的基因表达向量依次记为 $g'_1,g'_2,...,g'_n$.经排序后,相似性高的数据排列在序列前段,该区域数据密度高,以此作为聚类的中心,是合理的推测.这样一来,我们在下一步计算中就有了明确的方向,极大地简化了计算.

我们从细胞周期数据集[14](介绍见后)中随机抽取了 72 条基因表达谱数据.图 1(b)为经主集法排序后的基因表达图谱,图谱最下端(从 g_1 开始,对应 $X(t)$ 的最大分量)的基因表达谱向量具有高度相似性.图 1(c)为层级聚类方法处理后的结果,具有相似基因相邻的特点,但各个类混杂在一起难于确定.为了证实处理后产生的聚类效果确实以及数据本身的实际信息,我们将原始各行数据的各列顺序随机打乱.这一过程破坏了原本存在于表达向量之间的相关性,如图 1(d)所示,随机处理后的数据经主集法排序后不产生相似的效果,可见图 1(b)中的结果确实与数据的实际生物信息有关.

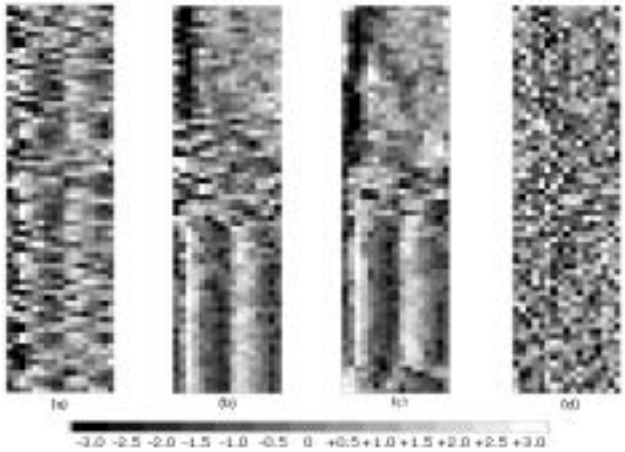


Fig.1 Comparison of several results

图 1 多处理结果比较

如图 1 所示为选取的 72 条基因在 17 个时间点上的表达值图谱,每一行代表一个基因(表达谱向量),每一列代表一次实验或是一次采样.灰度值的不同代表表达值的数值高低,灰度值越高代表表达值越低,其值由黑色(代表−3.0)到白色(代表+3.0)渐变.其中,(a)是原始数据,(b)是经主集法排序后的图谱,(c)是由 hierarchical clustering 聚类后的表达图谱,(d)是随机数据经主集算法排序后的图谱.

2.2.2 从数据序列中分割出一个类

将排序后的基因依次记为 $g_1, g_2, ..., g_n$,属于同一个主集的基因由于对应的 $x_i(t)$ 较大,会沉淀在序列底部(如图 1(b)所示).以欧式距离作为相似性的判断,先判断 g_1 和 g_2 之间的距离.假如 g_1 和 g_2 之间的距离大于一个初始设定的距离值 d_{ini},则停止分割,否则将 g_1 和 g_2 作为初始类.接着,判断 g_3 是否属于这个类,若不属于,则停止分割;若属于,则 g_3 加入类中,并用相同的方法接着依次判断,直至得到分割的类为止.通过图 2 可以对这一过程有所理解.

假如 $\{g_1, g_2, ..., g_k\}$ 属于同一个类,如何判断 g_{k+1} 是否属于这个类呢?令 $D = \{g_1, g_2, ..., g_k\}$ 为类内元素的集合,$U = \{g_{k+2}, g_{k+3}, ..., g_n\}$ 为类外元素的集合,那么

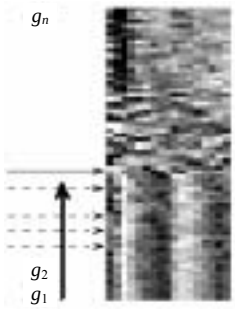


Fig.2 Cut one cluster from the list

图 2 从序列中分割出一个类

$$d_D = \frac{1}{k}\sum_{g_i \in D} d_{i,k+1} \tag{5}$$

$$d_U = \frac{1}{n-k-1}\sum_{g_i \in U} d_{i,k+1} \tag{6}$$

其中,$d_{i,k+1}$ 表示 g_i 和 g_{k+1} 之间的距离,本文采用欧氏空间距离.

根据类的性质,若

$$d_U > d_D \tag{7}$$

即 g_{k+1} 到类 $\{g_1, g_2, ..., g_k\}$ 的距离平均值小于 g_{k+1} 到其他非主集基因的距离平均值 g_{k+1},那么认为 g_{k+1} 属于这个类.此判断对于类的性质的规定,将影响最终的聚类结果.由于在对基因表达谱进行聚类时,目的是寻找表达谱数据高度相似的基因,而条件(7)过于宽松,本文采用的判断条

件是 $d_U \geq 3d_D$,这样规定将有助于减少结果类中的噪音.若放松此约束条件,则结果中类的半径增大,个数减小.

2.3 全局算法

全局算法是一个迭代过程,用以上方法每次找到一个类,然后将该类从原数据集 G 中去除掉,再对剩下的数据重复以上过程.为了进一步分析的需要,我们认为那些包含的基因数目大于或等于某个常数的类是有意义的类.整体算法所需参数包括一些设定为固定值的内部参数(如合法类中应包含基因的最小数 MIN _ *SIZE*)以及原始数据集 G 本身.符合迭代停止条件时终止循环(见后).以下的伪码用以说明全局算法的过程以及输入、输出:

```
INPUT   G = {g_i', i = 1,..., N} , K = 1, 某些固定参数(如 MIN _ SIZE ).
WHILE   停止迭代条件不满足
        C_k = locate _ center(G);          /* 第 1 步,定位一个类的中心 */
        S_k = find _ cluster(G, C_k);      /* 第 2 步,在剩余的数据中确定一个类 */
        G = G − S_k;                       /* 每次从数据集 G 中去除属于当前类 S_k 的元素 */
        IF   #S_k >= MIN _ SIZE            /* 其中 #S_k 表示类 S_k 中数据点的个数 */
             OUTPUT   S_k;                 /* 若 S_k 是合法类则输出,否则不输出 */
             K++;
        ENDIF
ENDWHILE
```

2.3.1 迭代停止条件

在全局算法中,符合迭代停止条件时终止循环.也就是当以下任意一种情况成立的时候:

1. 所有的数据已被处理完毕,也就是说 G 为空时;
2. 在连续的数次循环中,找到的类中的基因的个数小于合法类中应包含基因的最小数.

该算法在 MATLAB 里执行,实现很简单.下面我们将讨论算法性能以及实验结果.

2.4 算法性能分析与比较

该算法需要多次迭代,且结果种类的个数由数据性质决定,很难给出该算法的时间复杂度的精确度量.然而,可以根据主要参数的相关运算来估计整体算法时间复杂度.用 N 表示基因个数, d 为基因向量维数, *VS* 表示找到的类的个数,则在确定一个类的过程中,本算法的时间复杂度为 $\sim O(N*d)$.本算法实现步骤与 QT_Clust[8] 相似,由表 1 可见,本算法在时间复杂度等方面优于 QT_Clust.另外,Ds_Clust 算法基于数据集本身的数据性质,找到的是数据点中的紧密集,在聚类的过程中不会将相似性不高的点强制地分入某一类中,因此,该算法具有很好的抗噪能力.该算法对于原始数据点的输入顺序不敏感;几乎不需要人为指定参数,使所需的领域知识最少化;聚类结果具有可伸缩性,其结果更符合数据特性;可以处理高维数据,数据维数的变化只改变计算量的大小,而不影响算法性能.表 2 从多方面比较了 Ds_clust 方法与 3 种最常用的聚类算法:*K*-means,hierarchical clustering 和 SOM 方法.

Table 1 Comparison between Ds_Clust and QT_Clust

表 1 Ds_Clust 与 QT_Clust 的比较

	Ds_clust	QT_Clust
User input	1. Data set G 2. Initial distance d_{ini}	1. Data set G 2. Radius of the cluster R 3. The minimum number of the left genes (stop criterion)
Time complexity	$\sim O(N*d*VS)$	$\sim O(N^2*d*VS)$
The scale of the cluster is flexible	Yes	No
Need predetermination of the cluster number	No	No
Result	Deterministic	Deterministic

Table 2 Comparison of Ds_Clust, *K*-means, hierarchical clustering and Self-Organizing Maps

表 2 Ds_Clust 与 *K*-means,hierarchical clustering,Self-Organizing Maps 方法的比较

	Ds_clust	*K*-means	Hierarchical clustering	SOM
Result format	Set of clusters	*K* clusters	Tree structure (difficult to interpret for large data set)	Set of predefined number of clusters
Principal user-defined parameters	Initial distance d_{ini}	Number of clusters *K*	–	Number of clusters/ node topology
Number of clusters	Produced by the algorithm	Predefined	–	Predefined
Whether include all genes in clusters	No	Yes	Yes	Yes
Ability of noise resistance	Strong	Not strong	Not strong	Strong
Data sequence sensitivity	No	Yes	No	Yes
Computational complexity of one run	Linear in *N*	Linear in *N*	Quadratic in *N*	Linear in *N*

经过以上分析,可以说 Ds_Clust 是一种实用的、高效的芯片数据聚类算法,适用于基于数值特征的聚类应用,但将不适用于类间嵌套的应用.此类问题还有待于进一步研究.

3 实验结果与分析

我们将 Ds_Clust 用于酵母细胞周期的芯片表达谱数据聚类.这批数据对 1 567 条基因在等间隔的 17 个时间点进行了采样,这段时间包含 2 个细胞周期,顺序依次为 G_1→S→G_2→M→G_1→S→G_2→M.原始数据可在网站(http://genomics.stanford.edu)中获得.细胞由一次分裂中期到下一次分裂中期的历程称为细胞周期.细胞周期的运转是十分有序的,是基因有序表达的结果.我们从原始数据中筛选出 1 000 条周期性最强的基因进行聚类.

当在 1.8G 奔腾处理器、256M 内存的 PC 机上处理时,共耗时 65.17 秒.图 3 列出了所得结果的前 8 个类的平均表达水平和误差, *NG* 的值表示该类中包含的基因个数.第 1 类基因在 G_1 期的中期高表达,这类基因的表达具有极强的周期性.这类基因主要的作用是 DNA 合成、复制、重组和修复.第 3 类基因在 M 期表达值高,这类基因主要与有丝分裂相关,例如酵母芽殖和极性化.第 6 类基因在 S 期表达值高,这类基因包括如组蛋白基因等主要与 DNA 的合成相关的基因.第 5 类基因在 G_2 期高表达,此类基因主要是一些转录调控因子,这些基因的高表达正是为细胞进入有丝分裂作好准备.第 2 类和第 4 类基因可能与线粒体能量代谢相关,但具体的生物意义还有待研究.这些类的意义和一些文献[5,15]报道的类相符.另外,我们计算每个类中的基因与这个类中心的相关系数值,所有的相关系数值都在 0.8 以上,这也说明本文所述聚类算法的抗噪性能强.

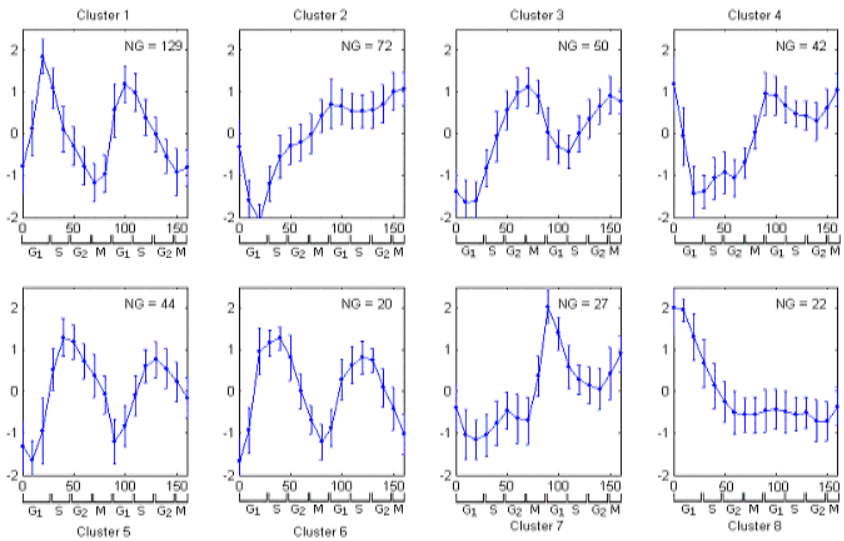
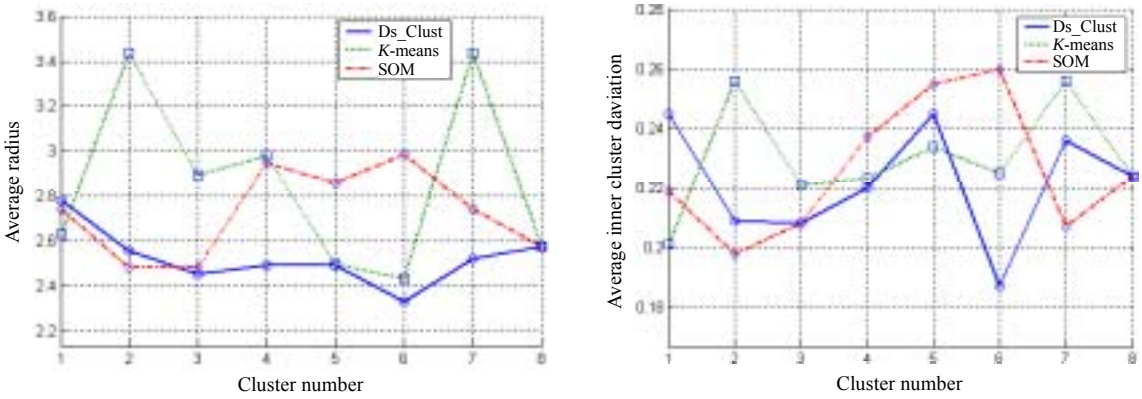


Fig.3 Average expression level and error of the first eight clusters

图 3 前 8 个类的平均表达水平及误差

然后,我们用 *K*-means(指定 *K*=8)以及 SOM 方法(初始化为 4×2 个神经元)方法对上述 8 个类中的 406 条基因进行聚类.图 4 比较了由 3 种方法得到每个类的性质.图 4(a)比较了类的半径大小,图 4(b)比较了类内方差大小.可见,由 Ds_Clust 得到的类具有较小的半径以及较小的组内方差,所以,用 Ds_Clust 聚类,类内噪音更小,效果更好.



(a) Radius of each cluster (b) Inner cluster deviation of each cluster

(a) 各类半径比较 (b) 各类组内方差比较

Fig.4 Comparison of the results of Ds_Clust, *K*-means and SOM

图 4 Ds_Clust,*K*-means,SOM 的聚类结果比较

4 结束语

将基因表达谱数据看作高维空间中的一组向量或点,可用聚类算法对其进行分析.大多数已有算法都需要人为地给出一些参数,然而在没有先验知识的情况下,人为地确定这些参数是十分困难的.如 *K*-means 方法和 SOM 方法都需要人为地预先给定结果类的个数.且 *K*-means 方法在处理较大数据量时具有较理想的扩展性和效率,但可能陷入局部最优,且结果受噪声和异常值的影响较大.SOM 方法可以处理部分数据及含缺失值的数据,算法稳健,结果易于可视化,但作为神经网络算法,要确定的参数太多,而且这些参数具有一定的经验性.层级聚类可以得到基因之间层级性的关系,结果用树状图表示,易于可视化,但结果不稳定,受初始变量和所选参数的影响较大.为了解决这些难题,本文提出了一种迭代的聚类算法,几乎无须人为给定输入参数,结果类的大小和结构由数据自身的特点决定.该算法对数据输入顺序不敏感,具有较优的时空效率以及很好的抗噪性能,所得结果性能好,对进一步的分析提供有效的保证.选择更合适的相似性判定标准以及改善分割的标准是有待改进的地方.

References:

[1] Tamayo P, Slonim D, Mesirov J, Zhu Q, Kitareewan S, Dmitrovsky E, Lander ES, Gloub TR. Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. Proc. of the National Academy of Sciences, USA, 1999,96:2907−2912.

[2] Carr DB, Somogyi R, Michaels G. Templates for looking at gene expression clustering. Statistical Computing & Statistical Graphics Newsletter, 1997,8:20−29.

[3] Eisen MB, Spellman PT, Brown PO, Bottstein D. Cluster analysis and display of genome-wide expression patterns. Proc. of the National Academy of Sciences, USA, 1998,95:14863−14868.

[4] Herrero J, Valencia A, Dopazo J. A hierarchical unsupervised growing neural network for clustering gene expression patters. Bioinformatics, 2001,17:126−136.

[5] Tavazoie S, Hughes JD, Campbell MJ, Cho RJ, Church GM. Systematic determination of genetic network architecture. Nature Genetics, 1999,22:281−285.

[6] Lukashin AV, Fuchs R. Analysis of temporal gene expression profiles: Clustering by simulated annealing and determining the optimal number of clusters. Bioinformatics, 2001,17(5):405-414.

[7] Ben-Dor A, Yakhini Z. Clustering gene expression patterns. Journal of Computational Biology, 1999,6:281−297.

[8] Heyer LJ, Kruglyak S, Yooseph S. Exploring expression data: identification and analysis of coexpressed genes. Genome Research, 1999,9(11):1106−1115.

[9] de Risi JL, Iyer VR, Brown PO. Exploring the metabolic and genetic control of gene expression on a genomic scale. Science, 1997, 278:680−686.

[10] Lander ES. Array of hope. Nature Genetics, 1999,21:3−4.

[11] Schena M, Shalon D, Davis R, Brown P. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. Science, 1995,270:467−470.

[12] Sherlock G. Analysis of large-scale gene expression data. Brief Bioinformatics, 2001,2(4):350−362.

[13] Pavan M, Pelillo M. A new graph-theoretic approach to clustering and segmentation. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. Computer Society, 2003. 98−104. http://www.dsi.unive.it/~pelillo/papers/cvpr03.pdf

[14] Cho RJ, Campbell MJ, Winzeler EA, Steinmetz L, Wodicka L, Wolfsberg TG, Gabrielian AE, Landsman D, Lockhart DJ, Davis, RW. A genome wide transcriptional analysis of the mitotic cell cycle. Molecular Cell, 1998,2(1):65−73.

[15] Getz G, Levin E, Domany E, Zhang MQ. Super-Paramagnetic clustering of yeast gene expression profiles. Physics A, 2000,279: 457−464.

****

《中国计算机学会通讯》杂志创刊

近日，由中国科学院院士张效祥题写刊名的《中国计算机学会通讯》正式出版。这本杂志由中国计算机学会主办，高等教育出版社出版，面向计算机专业人士及信息领域的相关人士。杂志利用中国计算机学会的学术优势，组织信息技术各个领域最有影响的专家撰稿，全面介绍计算机科学技术发展的最新趋势，并预测未来一段的技术发展趋势，具有权威性和指导性，可以帮助读者更加开阔视野，了解 IT 最前沿的动态，把握 IT 发展方向，适合与计算机相关的科研、教学，以及产业和管理等各方面的人士阅读。

《中国计算机学会通讯》2005 年为季刊，2006 年为双月刊，2007 年为月刊。杂志每期 100 页，开本 200×273，全彩色铜版精美印刷。

杂志主编是中国计算机学会理事长、中国科学院计算技术研究所所长、中国工程院院士李国杰研究员；执行总编是中国计算机学会理事冀复生先生，有多名国内知名专家是该杂志的编委。目前，杂志的主要栏目有专题报道（封面故事）、展望、观点、会员园地等。其中专题报道就计算机科学的某一领域进行全面、深入的综合报道，使读者对于该领域有一个权威和整体的了解；展望报道计算机业界的发展趋势的综述文章；观点就计算机技术和业界发展发表业界专家的独到见解；会员园地包括学会动态、学会和有关会议等活动的预报、职场信息、新书架等。杂志还将翻译《Communications of ACM》、《IEEE Computer》和《IEEE Spectrum》等外刊的优秀文章。

正如李国杰主编所指出的，我国目前已经有很多计算机类的学术刊物，但是关心这些刊物的主要是文章的作者，而《中国计算机学会通讯》作为学术与技术应用相结合的学术性会刊，其目的主要不是为研究室和科研人员提供一个新的发表文章的载体，而是让计算机科技工作者和关心计算机发展的各类人士更全面、更深刻地了解相关技术的发展趋势。因此，杂志将努力通过介绍和探讨 IT 领域的最新技术、观点和理论，传播 IT 技术的新知识和新进展，加强 IT 界的学术、技术与应用交流，促进 IT 领域科研、生产、教学和市场之间的密切结合，发掘和培养 IT 技术人才，促进国内 IT 技术水平的提高和 IT 产业的发展，促进 IT 和其他领域的交流。

《中国计算机学会通讯》杂志免费赠送学会会员（学生会员除外）（不零售），这是中国计算机学会本届理事会为加强会员服务的重大举措。

杂志编辑部欢迎专家、读者、会员踊跃投稿。相信本杂志将成为信息技术领域的一份重要学术刊物。