CIS:一种基于迭代扩张的微阵列数据聚类算法

王晓明[1] 印 莹[2]
(辽宁科技大学电信学院 鞍山 114044)[1] (东北大学 沈阳 110004)[2]

摘 要 DNA微阵列技术使同时监测成千上万的基因表达水平成为可能。直接把传统聚类算法用于高维基因表达数据分析会受到"维难"的困扰。特征转换和特征选择是两种常用的降维方式,但前者产生的新特征难以用原来的领域知识解释,后者通常会丢失信息。另外,传统的聚类算法通常由用户指定聚类参数,参数设置不同对聚类结果有很大的影响。针对上述问题,本文提出了一种新的基于迭代扩张的微阵列数据聚类算法-CIS。它不采用特征转换和特征选择的方式,并自动确定聚类参数。CIS反复用最新得到的样本聚簇得到新的聚类基因,然后以新的基因聚簇为特征重新聚类样本,逐步求精,最终的结果容易解释且避免了信息的丢失。该方法降低了由于用户缺少领域知识引起的实验误差。CIS算法被应用于两个真实的微阵列数据集,实验结果证实了算法的有效性。

关键词 微阵列,聚类,降维

CIS: An Iterative Spread-based Algorithm for Clustering Micro-array Data

WANG Xiao-Ming[1] YIN Ying[2]
(Liaoning University of Science and Technology, Anshan 114044)[1] (Northeastern University, Shenyang 110004)[2]

Abstract DNA Micro-array technique makes it possible to simultaneously monitor the expression levels of tens of thousands of genes. The traditional clustering methods will suffer from the curse of dimensionality when directly applied to Micro-array data. The two common dimensionality reduction methods, i. e. feature transformation and feature selection, are unsuitable for the analysis of Micro-array data, since the former generates the new features difficult to interpret and the latter misses some information. Besides, most traditional clustering algorithms need the user-specific parameters, which may result in quite different results. In this paper, we present an iterative spread-based algorithm, namely CIS, for clustering Micro-array data, which selects threshold automatically. Instead of feature selection and feature transformation, in a progressively refining manner, CIS repeatedly partitions the genes with the new-generated sample clusters as features, and then partitions the samples with the new-generated gene clusters as features. The algorithm is applied to two real gene Micro-array data sets. Experiment results confirm its effectiveness and efficiency.

Keywords Micro-array, Clustering, Dimensionality reduction

1 前言

DNA微阵列(Micro-array)技术是近年来实验分子生物学上的一项重大突破。这项技术的广泛应用使人们能够同时观察某一生命现象中成千上万个基因的动态表达水平,一次简单的微阵列实验就会产生数以百万计的基因表达数据。正确的理解和解释隐藏在结果数据中的信息,有助于基因功能组、基因调控网络构建、肿瘤分型、肿瘤分类、药物靶位识别[1~3]等许多方面的研究。如何从大量的基因和复杂的生物网中发现令人感兴趣的模式是功能基因组时代面临的一项巨大挑战,迎接这项挑战的首要一步是聚类技术的使用。

聚类的目的是在大量数据中发现隐含的自然结构和兴趣模式。随着微阵列技术的迅猛发展,许多传统的聚类技术,如K-means[4]、SOM[5]等,可以被直接应用于基因表达数据的研究分析。但是,这种做法存在着一些潜在的问题。

首先,高维空间中,一个点与距离它最近点的距离接近于距离它最远点的距离[6]。基于样本聚类的时候,每个样本被看作 n 维(大约 $10^3 \sim 10^4$ 维)空间中的一个点,整个聚类过程可以看作在几千维的空间中对不到一百个点进行划分。数据点的极度稀疏,使得不加任何改动,直接把适合低维空间聚类的传统算法用于基因表达数据分析难以得到理想的结果。对此,一个通用的解决办法是降维,主要有特征转换和特征选择两种降维方式。特征转换的方式以PCA[7]为典型代表,它把原空间特征进行线性组合,得到新的特征,降低了原始空间的维数。这种方式存在一定的局限性。一方面,新特征是原特征的线性组合,保留了数据点之间的相对距离,当数据中存在大量不相关属性的时候,隐藏其中的聚簇仍然不能被有效地发现。基因表达数据中存在大量的无关数据[8],因而特征转换的降维方式对基因表达数据集不适用。另一方面,即使是经过最简单的线性组合得到的特征,也很难用原来的领域知识进行解释,这又在一定程度上限制了特征转换方式在基因表达数据分析中的应用。

另一种常用的降维方法是特征选择,基于特征选择方式的算法有很多,比如文[9,10]等。这种方法的思想是评估每个特征与其当前对应聚簇的相关程度(常被称为相关指标),反复迭代,每迭代一次,过滤掉一部分相关指标值较小的特征。筛选出一小部分与聚簇最相关的特征,作为最终的度量特征进行样本聚类,得到最后结果。这种方法存在的问题是迭代开始的时候,相关指标的计算比较粗,这时削减掉一部分特征(通常是全部特征的2/3),很可能会造成重要信息的丢失,忽略重要功能的基因的发现。例如,$g_1, g_2, \cdots, g_{1000}$ 代表1000条不同的基因,$r_{i1}, r_{i2}, \cdots, r_{i1000}$ 分别代表对应基因的相

关指标值，并假定按降序排列，$s_1, s_2, \cdots, s_{100}$ 代表 100 个样本。经过一次迭代后，得到两个样本聚簇 $C_1=\{s_1, s_3, \cdots, s_{99}\}$，$C_2=\{s_2, s_4, \cdots, s_{100}\}$。根据这个结果赋予每个基因（特征）一个相关指标，按相关指标值从大到小排列，削减掉后面 2/3 的基因，$g_{331}, g_{332}, \cdots, g_{1000}$。然而，此时 $g_{331}, g_{332}, \cdots, g_{1000}$ 的相关指标值较低，很可能是因为第一次聚类的结果不准确，很多样本被误分到那些本不应属于的聚簇中造成的。如 s_1 本应属于 C_2 却被分到了 C_1 中，s_2 本应属于 C_2 却被分到了 C_1 中，等等。随着迭代的进行，误分率逐渐降低，一些开始较低的相关指标值会逐渐变大。如果这些基因在它们的相关指标值变大之前就被过滤掉，那么隐含在其中的信息就会丢失。针对这个问题，我们提出用“reclustering”的方法改进基于过滤的方法。这种方法不必在每次迭代中削减基因的总数，以免丢失信息，只是将它们反复地进行重新组合，使每次组合的结果都尽可能比前一次的精确。然后用每个越来越精确的基因组作为特征，重新进行样本的划分。这样，既降低了丢失信息的危险，又能达到较高的聚类准确性。我们把这种方法应用于 colon 和 leukemia 数据集，实验结果验证了该算法的有效性。

多数聚类算法存在的另一个主要问题是要求用户预先指定结果聚簇的个数 k 或密度阈值 δ，设定不同的 k 或 δ 得到的结果差异很大[11]。在领域知识未知的前提下，给出正确的 k 或 δ 是非常困难的，在基因表达数据分析中，这种情况尤为突出。好的聚类算法应该尽量减小用户指定的参数对聚类结果的影响或者由算法本身来逐步调整所需相关参数的大小。本文中，我们在对原数据集随机重组的基础上，把原数据集打乱成一个由原有数据构成的噪音数据集，通过这个噪音数据集的聚类结果，来决定算法的参数。用 P 代表聚类算法所需的一组参数 $p_1, p_2, \cdots, p_n$，在最初选定一组参数 P_1 后，噪音数据集中可能还会存在若干聚簇，很显然，它们是不具有统计意义的。这说明参数组 P_1 不能完全消除噪音聚类，继续变化参数组 P_1 为 P_2，$P_3, \cdots, P_m$。取定参数组 P_m 时，噪音数据集中不再出现任何聚簇，这时的参数组 P_m 可作为聚类原始数据集所需的优化参数，它消除了噪音数据的影响。

本文第 2 节介绍了相关知识，第 3 节详细讨论了 CIS 算法，第 4 节给出了 CIS 在结肠癌样本数据集和白血病数据集上的实验结果，最后对全文进行了总结。

2 相关知识

本节主要介绍与基因表达数据聚类分析相关的一些基本概念，包括基因表达矩阵，数据预处理，数据标准化和相似性度量。

2.1 基因表达矩阵

来自于微阵列实验的基因表达矩阵通常被描述成一个 $m \times n$ 的矩阵，如图 1 所示。图中每一行代表一个基因或 EST，通常统一用符号 $g_i (i \in \{1,2,\cdots,m\})$ 表示。每一列代表一个时间点，组织切片或实验条件，在不引起混淆的情况下，统一用符号 $s_j (j \in \{1,2,\cdots,n\})$ 表示。矩阵中每个具体的值 e_{ij} 代表样本 s_j 的第 i 个基因的表达水平。在进行样本聚类的时候，每个样本被看成 m 维空间上的一个点 $s_j(e_{1j}, e_{2j}, \cdots, e_{mj})$。为了提高聚类的准确性，通常要削减不相关或冗余的属性，也就是用一个子集 $A \subset \{e_{1j}, e_{2j}, \cdots, e_{mj}\}$ 作为属性去进行样本划分。很容易看出，选择不同的 A 会得到不同的聚类结果，A 越接近真实聚簇，划分的结果越准确。

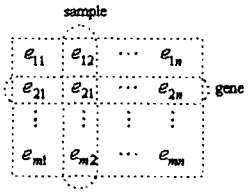


图 1 基因表达矩阵

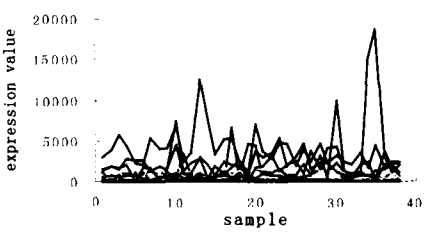
2.2 数据预处理

某个特定的生命过程并非同时涉及整个基因组，其中一小部分基因集合在起着决定性的作用[1]。图 2(A) 和图 2(B) 分别对应 colon 数据集和 leukemia 数据集中部分基因的表达强度分布示意图。图中 X 轴代表数据集中的样本，Y 轴代表基因的表达强度。每条曲线代表从数据集中随机抽取的一个基因在所有样本上的表达水平变化情况。从图中可以清楚地看到，有一些基因在所有样本上的表达水平都差不多。显然，这部分基因对样本划分是不起多大作用的，可以用最小标准差(std)法来移除这些基因。对每个基因按照公式(1)计算它的标准差。

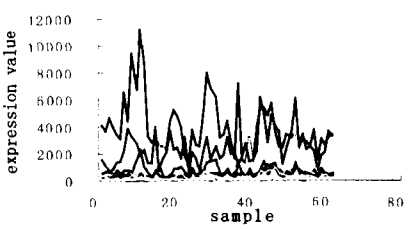
$$std(g_i)=\sqrt{\frac{1}{n}\sum_{j=1}^{n}(e_{ij}-\overline{e_i})^2}, \tag{1}$$

其中 $\overline{e_i}=\frac{1}{n}\sum_{j=1}^{n}e_{ij}$

如果一个基因的 std 很小，甚至接近 0，说明它在每个样本上的表达值变化不大，有理由将其删除。



(A) 结肠癌数据集中部分基因的表达强度值分布示意图



(B) 白血病数据集中部分基因的表达强度值分布示意图

图 2

2.3 数据标准化

微阵列实验中存在大量的度量差。聚类前，需要对表达矩阵标准化。我们对公式(1)处理后的数据集进行了如下标准化。首先，表达矩阵 E 的每一列除以本列的平均值：

$$e_{ij}=e_{ij}/\overline{e_j}, \overline{e_j}=(1/m)\sum_{i=1}^{m}e_{ij} \tag{2}$$

然后，再按公式(3)标准化每一行，使每一行的平均值为 0，方差为 1。

$$e''_{ij}=\frac{e'_{ij}-\overline{e'_i}}{\|e'_i\|}, \overline{e'_i}=(1/n)\sum_{j=1}^{n}e'_{ij}, \|e'_i\|^2=\sum_{j=1}^{n}(e'_{ij}-\overline{e'_i})^2 \tag{3}$$

2.4 相似性度量

聚类分析常用欧几里得距离刻画两个对象的相似程度。高维空间中，任意两点的欧几里得距离接近相等[6]，用欧几里得距离代表基因表达矩阵中两个样本的相似性不太适合。本文用皮尔逊相关系数代替常用的欧几里得距离，从模式相似性的角度更好地刻画了两个对象间的相似程度。具体公式如下：

$$\rho_{x,y}=\frac{k(\sum_{i=1}^{k}x_i\times y_i)-(\sum_{i=1}^{k}x_i)\times(\sum_{i=1}^{k}y_i)}{\sqrt{[k\sum_{i=1}^{k}x_i^2-(\sum_{i=1}^{k}x_i)^2][k\sum_{i=1}^{k}y_i^2-(\sum_{i=1}^{k}y_i)^2]}} \tag{4}$$

其中，$X=(x_1,x_2,\cdots,x_k)$，$Y=(y_1,y_2,\cdots,y_k)$，k 是向量 X 和 Y 的长度。

3 CIS 聚类算法

本节将详细地介绍 CIS，(Clustering via Iterative Spread Clustering)聚类算法。该算法受“逐步求精”思想的启发，在无需用户指定聚类参数的情况下，不断地用新得到的样本聚簇指导基因的重新划分(不削减基因总数)，反过来，再用更精确的基因聚簇指导样本的重新划分，逐步得到正确的样本聚簇。CIS 不仅能发现近似的样本划分，而且还具有发现子划分的能力。

3.1 定义

首先，介绍算法中用到的几个定义。

定义 1(父类和子类) 一个聚簇 C，在某个条件下分解为几个子聚簇 $C_1,C_2,\cdots,C_n$，称 C 为 C_i 的父类，C_i 为 C 的子类，$i\in\{1,2,\cdots,n\}$。

CIS 算法中，父类和子类不是绝对的。子类可以作为新的父类继续分解，当前的父类是由以前的子类分解而来。

定义 2(父类分解) 如果有一个聚簇 C，它在某个条件下分裂为几个子聚簇 $C_1,C_2,\cdots,C_n$，而且 $\forall i,||C|-|C_i||\geqslant\delta,i\in\{1,2,\cdots,n\}$，我们就说父类 C 已经分解。

δ 是一个预定的阈值，表示父类至少要失去几个成员才被认为分解。根据表达数据的实际情况，我们把这个值设定为聚簇本身所含成员个数的 1/10。聚类是一个复杂的过程，允许有合理的误差，因此当一个聚簇失去了很少($\leqslant\delta$)的几个成员时，完全有理由认为这是由“偶然”因素造成的，特别是当聚簇本身所含成员比较多的时候。

定义 3(子类生成) 如果一个聚簇 C_i 是从聚簇 C 中分裂出来的，那么当它的规模超过一个给定的阈值 γ 时，称它形成了一个新的聚簇。

聚类过程中，往往会因为一些“偶然”的原因形成一些没有意义的随机聚簇，一个最粗略的评估办法就是簇的大小，如果一个簇的规模小于指定的阈值，它很可能没有什么实际含义。按照微阵列实验的惯例，我们把这个值定为 7。

需要指出的是，并不是一个父类的分解必然会导致新的子类生成。例如，经过若干次分解后，产生了一个聚簇 C_k，$|C_k|=12>7$。C_k 进一步分解为 C_{k1} 和 C_{k2} 两个子类，$|C_{k1}|=|C_{k2}|=6$。此时，我们认为 $|C_{k1}|$，$|C_{k2}|$ 的规模太小($\leqslant 7$)，没有新的子类生成，但是父类 C_k 已经分解。

3.2 初始聚类

根据公式(4)建立样本的相似矩阵 M_s 和基因的相似矩阵 M_g 后，我们开始聚类。假设有一个相似矩阵(图 3(A)所示)，其必有如下性质：

$\forall e_{ij}\in M,e_{ij}=e_{ij}$ 即 M 为对称矩阵。M 简记为 M'(图 3(B)所示)，其中所有元素构成的集合记为 R。建立一个有序列表 $L=\{r_i|r_i(R$ 且 $r_i>r_j$ if $i>j\}$。对 M_g 来说，列表 L 会比较长，我们可以用基于 entropy 的方法来缩减它的长度。与其它离散化的方法不同，基于 entropy 的方法更有可能将区间边界定义在准确位置。对 L 进行离散化后，得到若干区间，不妨记为 $[r_1,r_2],[r_3,r_4],\cdots,[r_{t-1},r_t]$。对每一个 $[r_{i-1},r_i]$，按公式(5)计算 d 值

$$d=\sqrt{|(r_{i-1},r_i)|} \tag{5}$$

其中，$|(r_{i-1},r_i)|$ 代表区间 (r_{i-1},r_i) 内的点数。

$$M=\begin{array}{c}\begin{matrix}O_1 & O_2 & \cdots & O_n\end{matrix}\\\begin{bmatrix}e_{11} & e_{12} & \cdots & e_{1n}\\e_{21} & e_{22} & \cdots & e_{2n}\\\vdots & \vdots & \ddots & \vdots\\e_{n1} & e_{n2} & \cdots & e_{nn}\end{bmatrix}\end{array}\begin{matrix}O_1\\O_2\\\vdots\\O_n\end{matrix}\qquad M'=\begin{array}{c}\begin{matrix}O_1 & O_2 & \cdots & O_n\end{matrix}\\\begin{bmatrix}e_{11} & & & \\e_{21} & e_{22} & & \\\vdots & \vdots & \ddots & \\e_{n1} & e_{n2} & \cdots & e_{nn}\end{bmatrix}\end{array}\begin{matrix}O_1\\O_2\\\vdots\\O_n\end{matrix}$$

(A) 初始相似矩阵　　(B) 简化的相似矩阵

将 $[r_{i-1},r_i]d$ 等分，所有的等分点以及 r_{i-1},r_i，作为 $[r_{i-1},r_i]$ 内所有点的代表。从小到大对所有的 $[r_{i-1},r_i](2\leqslant i\leqslant t)$ 处理后，就得到一个缩减的有序列表 L'。

取 $l_i\in L'$，观察所有对象与 O_i 的相似度 $\mathrm{sim}(O_i,O_j),i,j\in[1,n]$。如果 $\mathrm{sim}(O_i,O_j)\geqslant 1_l$，则 O_i 和 O_j 为一类，得到一个初始聚簇 $cluster(O_i)=\{O_j|\mathrm{sim}(O_i,O_j)\geqslant 1_l,i,j\in[1,n]\}$。再从所有与 O_i 同类的 O_j 出发逐渐进行簇的扩张，即如果 $\mathrm{sim}(O_j,O_k)\geqslant 1_l,O_j\in\mathrm{cluster}(O_i)$，$O_k$ 也成为 $\mathrm{cluster}(O_i)$ 的一个成员。这个过程一直进行到 $\mathrm{cluster}(O_i)$ 不能再扩张为止。若此时 $|\mathrm{cluster}(O_i)|<n$，另选一个 $O_m\notin\mathrm{cluster}(O_i)$，重复上述过程，得到 $\mathrm{cluster}(O_m)$，最终 $\bigcup\mathrm{cluster}(O_i)=\{O_1,O_2,\cdots,O_n\}$。

不同的 l_i 会得到不同的结果，按照定义 1，某些聚簇在不同的 l_i 下都存在，这些聚簇是令人感兴趣的。例如，聚簇 C 在 $l_i,l_{i+1},\cdots,l_k$ 下都存在，$\Delta l=l_k-l_i$ 可以作为簇 C 的一种兴趣度度量，簇 C 的 Δl 越大，我们对它越感兴趣。Δl 是 CIS 中聚簇 C 的兴趣度度量参数。

下面讨论参数 Δl_c 的自动确定。把原矩阵打乱，组成另一个 $m\times n$ 的随机矩阵 M''，这时每个 $e_{ij}'\in M'$ 不一定代表样本 s_j 的第 i 个基因的表达水平，整个矩阵成为一堆“噪音”数据。从 l_i 开始，逐渐调高 l_i，当到达 l_t 时，如果所有的原始聚簇都已经分解，那么 $\Delta l_c=l_t-l_i$。Δl_c 消除了噪音聚簇。确定 Δl_c 之后，把所有 $\Delta l\geqslant\Delta l_c$ 的聚簇添加到结果集合 resultSet 中。

3.3 迭代的重聚类

用上述方法得到的样本聚簇 $S_1,S_2,\cdots,S_t$ 是不够准确的，因为所有基因都作为特征参与了样本划分。同样，可以得到基因的初始划分 $\{G_1,G_2,\cdots,G_s\}$。用 G_i，$i\in[1,s]$ 为特征划分 S(S 代表所有样本的集合)的效果要比用 G(G 代表所有基因的集)的效果好，因为 G_i 比 G 更接近一个真实的聚簇，尽管它也不是非常准确。以 $G_i,i\in[1,s]$ 为特征，依次划分 $S,S_1,S_2,\cdots,S_t$，得到新的样本聚簇 $S'_1,S'_2,\cdots,S'_t$。同理，$\{S'_1,S'_2,\cdots,S'_t\}$ 比 S 或 $\{S_1,S_2,\cdots,S_t\}$ 更接近真实划分。我们再以 S'_i，$i\in[1,t']$ 为特征重新划分 G，得到基因的新划分 $\{G'_1,G'_2,\cdots,G'_s\}$，$\{G'_1,G'_2,\cdots,G'_s\}$ 也应该比 G 或 $\{G_1,G_2,\cdots,G_s\}$ 的效果好。整个过程迭代进行，直到没有新的聚簇产生。这个过程没有像 PCA[7] 那样，用维组合的方式削减特征，也没有像文[9,10]那样，用维削减的方式减少基因总数，保留了结果的可解释性，避免了被削减基因中隐含信息的丢失。CIS 反复用最新得到的样本聚簇重新聚类基因，然后以新的基因聚簇作为特征重新聚类样本，逐步求精，得到正确的

样本划分及相关的基因聚簇。

3.2 节和 3.3 节中的算法分别如算法 1 和算法 2 所示。算法 1 描述了每次迭代过程中扩张聚类的过程。选定相似矩阵中的某个元素 M_{ij}，从有序列表 L 中的最小值 l_1 开始，扩张 M_{ij} 所属聚簇，直到从 M_{ij} 所属聚簇中的每个成员出发聚簇都不能再被扩张为止。取遍 L 中的所有值，返回所有 $\Delta l \geqslant \Delta l_c$ 的聚簇。算法 2 是 CIS 的迭代过程。经过预处理和标准化后，从初始的数据集开始，每次把算法 1 返回的样本聚簇添加进新的样本聚簇集合中，用返回的基因聚簇更新原来的基因聚簇集合，作为下一次迭代的对象属性。反复迭代，直到所有的父类都已分解，然而没有新的子类生成。

算法 1 Spread(M)

```
输入:相似矩阵 M
输出:Δl≥Δlc 的样本聚簇或基因聚簇
Begin
1. for l_i=l_1 to lt /*从有序列表中选定递增的 l_i 值*/
2.    for column=1 to n /*开始第一个聚簇的扩张*/
3.       {for row=1 to n
4.          {if M[row][column]≥l_i;/*若当前元素值大于选定的 l_i
            值,把 row 对应的元素添加进 column 对应的聚簇*/
5.             cluster (column)←row;}
6.          for each m∈cluster (column)/*对当前聚簇中的每个元
素扩张*/
7.             for line=1 to n
8.                {if M[line][m]≥l_i and line ∉ cluster (column)
9.                   then
10.                     cluster(column)←line; }
       }
11.   return all clusters with Δ_1≥Δl_c;/*返回所有符合条件的聚簇
*/
End
```

算法 2 CIS

```
输入:基因表达矩阵
输出:样本聚簇及相关的基因聚簇
Begin
1  数据预处理;
2  数据标准化;
3  SampleClusters=S; NewestSampleClusters=ϕ
   NewestGeneClusters=G;
4  Add Spread (M) to SampleClusters;/*添加新产生的样本聚簇到
   集合 SampleClusters 中*/
5  Update NewestGeneClusters with Spread (M_g);/*用新产生的基
   因聚簇更新集合 GeneClusters*/
6  While condition terminated is not satisfied/*如果没有未分解的
   父类和新生成的子类,结束整个过程*/
   {// 开始迭代
7     for each cluster S_i in SampleClusters
8     for each cluster G_j in NewestGeneClusters
9        {partition S_i with G_j;/*用 GeneClusters 中的新成员划分
         SampleClusters 中的元素*/
10          Add new-generated sample clusters to SampleClusters;}
11       Update NewestSampleClusters with all new-generated sam-
         ple clusters;/*更新集合 GeneClusters*/
12       Partition G with each cluster in NewestSampleClusters; /*
         用新生成的样本聚簇划分基因集合*/
13       Update NewestGeneClusters with the result obtained by 12;
      //迭代终止}
End
```

4 实验结果及分析

本节给出了 CIS 算法在两个真实微阵列数据集上的实验结果。第一个数据集包括 62 个患有结肠癌的病人样本。其中，40 个样本来自病人的癌变部位，其余 22 个样本取自病人未发生癌变的部位。从 6500 个基因中取出 2000 个表达水平超出一定阈值的基因作为兴趣基因组。第二个数据集包括 38 个白血病病人的样本。其中，27 个样本取自 ALL(急性淋巴细胞白血病)病人，11 个样本取自 AML(急性骨髓细胞白血病)病人。27 个 ALL 样本中，19 个与 B 细胞有关，8 个与 T 细胞有关。观察每个样本在 5000 个基因上的表达水平。

为了评估实验结果的有效性，我们对聚类结果分别进行了质量分析和可靠性分析。下面对这两种方法加以简要介绍。

4.1 聚类结果的质量分析

如果数据集中聚簇的真实结构可知，可用聚类结果和真实结构的比较评估聚类结果的质量。纯度(purity)和有效性(efficiency)反映了聚类结果与真实结构的符合程度。假定样本的真实划分为 $S=\{S_1,S_2\}$，S' 是结果中的某个聚簇，S' 和 S_1 的符合程度可以用公式(6)表示：

$$\text{purity}(S'|S_1)=\frac{|S'\cap S_1|}{|S'|};\ \text{efficiency}(S'|S_1)=\frac{|S'\cap S_1|}{|S_1|} \tag{6}$$

S' 与 S_2 的符合程度只需把公式(6)中的 S_1 替换为 S_2。

4.2 聚类结果的可靠性分析

虽然纯度和有效性指标对聚类结果进行了某种程度的评估，但是它们并不能说明聚类结果的可靠性。就是说，聚类结果可能是偶然因素造成的。下面，我们用预测强度来度量聚类结果的可靠性。

如果一个聚类算法把 n 个样本划分为 S_1 和 S_2 两组，$S_1=\{s_1,\cdots,s_k\}$，$S_2=\{s_{k+1},\cdots,s_{n-1}\}$，$s_n$ 作为留出的测试样本，G 是与当前划分相关的基因集合。每个基因 $g_i\in G$ 根据表达矩阵中 e_m 的值对留出的测试样本 s_n 进行投票，e_m 接近 μ_1(S_1 中对应基因的平均表达水平)，认为 s_n 应该属于 S_1，e_m 接近 μ_2(S_2 中对应基因的平均表达水平)，则认为 s_n 应该属于 S_2。依次用所有的 $S_i\in S_1$ 或 $S_i\in S_2$，$i=\{1,2,\cdots,n\}$ 作为留出的测试样本。统计最后的投票结果，V_1 和 V_2。S_n 的预测强度 $=\left|\frac{V_1-V_2}{V_1+V_2}\right|$。大多数样本的预测强度高说明聚类结果有较高的可靠性，不是偶然因素造成的。

4.3 数据集聚类结果分析

本节用 4.1 节和 4.2 节中介绍的方法分别对结肠癌样本数据集和白血病样本数据集的聚类结果进行了分析。

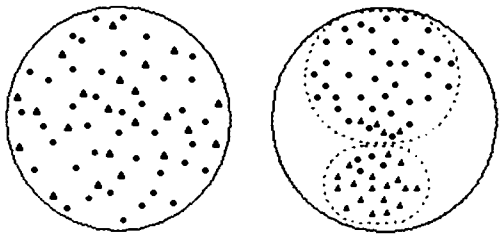
4.3.1 结肠癌数据集聚类结果分析

聚类结果的质量分析：实验中我们发现 G_{17} 具有划分肿瘤样本和正常组织样本的能力。G_{17} 是一个仅包括 8 个基因的基因聚簇。以它为特征，可以识别出 22 个正常组织样本中的 18 个。如果把结果中所有未被识别为正常组织的样本标记为肿瘤，我们就得到了肿瘤样本和正常组织样本的一个近似划分。肿瘤样本聚簇共有 40 个样本，其中 36 个为肿瘤样本，4 个为误分的正常样本，purity=90%，efficiency=90%。正常样本聚类共有 22 个样本，其中 18 个为正常组织样本，4 个为误分的肿瘤样本，purity=82%，efficiency=82%。表 1 列出了这 8 个基因及它们的相关描述。

表 1 G_{17} 中 8 个基因的描述

Hsa. 3152D31885gene1	"Human mRNA (KIAA0069) for ORF (novel proetin), partial cds
Hsa. 3207U27143gene1	"Human protein kinase C inhibitor-I cDNA, complete cds.
Hsa. 3088X72727gene1	H. sapiens tunp mRNA for transformation upregulated nuclear protein
Hsa. 3115X82103gene1	H. sapiens mRNA for beta-COP
Hsa. 2793X68194gene1	H. sapiens h-Sp1 mRNA
Hsa. 3316U20998gene1	"Homo sapiens signal recognition particle subunit 9 (SRP9) mRNA, complete cds
Hsa. 3117R492313' UTR1 38397	MITOCHONDRIAL PHOSPHATE CARRIER PROTEIN PRECURSOR (H (HUMAN)
Hsa. 305L19437gene1	TRANSALDOLASE (HUMAN); contains Alu repetitive element; contains PTR5 retrepetitive element

图 4(A)，图 4(B)分别是用所有基因为特征聚类 62 样本和用表 1 中的 8 个基因为特征聚类 62 样本的示意图。



(A)显示了62个样本的初始分布示意图，每个三角点代表一个正常组织样本，每个圆点代表一个肿瘤样本
(B)显示了以表1中的8个基因为特征，62个样本的聚类结果分布示意图

图4 结肠癌数据集的实验结果

实验还发现了结肠癌样本的子划分。例如，分别以 G_2，G_3，G_{10}，G_{14} 等基因聚簇作为特征，都发现了纯度为100%的结肠癌样本聚类。在这些聚类中，样本集{25,26,28,29,30,33,34,35,37}出现的频率较高，它很可能是结肠癌样本的一个子类。我们将对此作进一步的研究。

聚类结果的可靠性分析：图5显示了结肠癌样本预测强度值的变化情况。从图中可以看出，62个样本中共有59个样本的预测强度值大于0.5，这说明聚类结果的可靠性较高，不可能是偶然因素造成的，有实际的统计意义。

4.3.2 白血病数据集聚类结果分析

表2 G_{24} 中29个基因的描述

M16336_s_at	"CD2 CD2 antigen (p50), sheep red blood cell receptor"
M23323_s_at	T-CELL SURFACE GLYCOPROTEIN CD3 EPSILON CHAIN PRECURSOR
L05148_at	Protein tyrosine kinase related mRNA sequence
X68742_at	"Integrin, alpha subunit"
U23852_s_at	T-lymphocyte specific protein tyrosine kinase p56lck (lck) abberant mRNA
L28821_at	MANA2 Alpha mannosidase II isozyme
X95677_at	ArgBPIB protein
U41654_at	RagA protein
U49835_s_at	CHIT1 Chitinase 1
L13977_at	LYSOSOMAL PRO-X CARBOXYPEPTIDASE PRECURSOR
X03934_at	T-cell antigen receptor gene T3-delta
M13792_at	ADA Adenosine deaminase
X87241_at	HFat protein
S78187_at	M-PHASE INDUCER PHOSPHATASE 2
X76223_s_at	MAL gene exon 4
M12886_at	"TCRB T-cell receptor, beta cluster"
U67171_at	Selenoprotein W (selW) mRNA
U93049_at	SLP-76 associated protein mRNA
Z47055_s_at	"Partial cDNA sequence, farnesyl pyrophosphate synthetase like-4"
U14603_at	Protein tyrosine phosphatase PTPCAAX2 (hPTPCAAX2) mRNA
D11327_s_at	"PTPN7 Protein tyrosine phosphatase, non-receptor type 7"
M37271_s_at	T-CELL ANTIGEN CD7 PRECURSOR
U50743_at	"Na,K-ATPase gamma subunit mRNA"
X04145_at	"CD3G CD3G antigen, gamma polypeptide (TiT3 complex)"
M37815_cds1_at	CD28 gene (glycoprotein CD28) extracted from Human T-cell membrane glycoprotein CD28 mRNA
J04132_at	"CD3Z CD3Z antigen, zeta polypeptide (TiT3 complex)"
X59871_at	TCF7 Transcription factor 7 (T-cell specific)
X14975_at	CD1 R2 gene for MHC-related antigen
D00749_s_at	T-CELL ANTIGEN CD7 PRECURSOR

聚类结果的质量分析：白血病数据集聚类结果中，G_{24} 是一个非常令人感兴趣的基因聚簇，包括29个基因。以它为特征，可以识别出11个AML中的10个(记为 S_1)，只漏了一个。如果把所有未被识别为AML的样本标记为ALL(记为 S_2)，我们就得到了ALL-AML的一个近似完美的划分，purity(S_1 | AML)=100%，efficiency(S_1 | AML)=91%。同理，purity(S_2 | ALL)=96%，efficiency(S_2 | ALL)=100%。表2列出了这29个基因及对它们的描述。

G_7 是另一个有趣的基因聚簇，包括15个基因。以它为聚类特征，得到一个由10个样本组成的ALL聚簇，其中8个样本都是T细胞相关的。CIS能识别出ALL样本中T细胞相关的样本和B细胞相关的样本，说明它具有发现肿瘤子类型的能力。如果某种肿瘤的子类型未知，CIS对预测未知肿瘤的子类型会起到一定的作用。

此外，分别用 G_5，G_{15}，G_{19} 为特征划分样本的时候，得到的样本聚类的纯度和有效性也很高。分别是 $purity_{G5}=75\%$，$fficiency_{G5}=100\%$；$purity_{G15}=80\%$，$efficiency_{G15}=75\%$；$purity_{G19}=75\%$，$efficiency_{G19}=82\%$。

聚类结果的可靠性分析：图6显示了38个白血病样本预测强度值的变化情况。从图中可以看出，38个样本中共有37个样本的预测强度值大于0.5，说明CIS在白血病样本数据集上同样能得到可靠性较高的聚类结果。

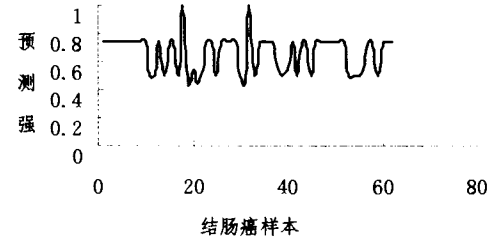


图5

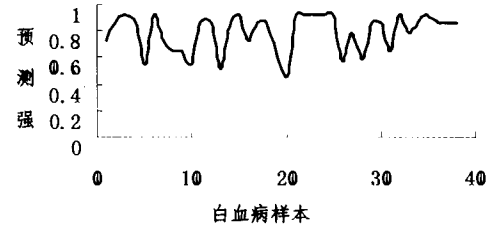


图6

结论 本文提出了一种用于基因表达数据分析的无参数聚类算法—CIS。与常用的基于降维的高维空间聚类算法不同，CIS充分考虑了基因表达数据的应用特点，没有用维削减的方式减少基因总数，避免了基因中隐含信息的丢失，也没有用维组合的方式降维，保留了结果的可解释性。该算法反复用重新组合的基因聚簇为特征，逐步求精，发现隐藏在大量基因表达数据中的样本聚类。而且，CIS由算法自身决定聚类参数，降低了用户输入的参数对结果的影响程度。

算法的有效性在两个真实的数据集(结肠癌数据集和白血病数据集)上得以展示。实验结果证实即使数据集的真实结构事先未知，没有用户指定的聚类参数，CIS仍有可能得到正确的结果。CIS是高维基因表达数据分析中一种非常有用的方法。

参考文献

1 Brazma A, Vilo J. Minireview: Gene expression data analysis. Federation of European Biochemical societies, June 2000, 480: 17～24

2 D'haeseleer P, Liang S, Somogyi R. Genetic network inference: from co-expression clustering to reverse engineering. Bioinformatics, 2000, 16(8):707～726
3 Sharan R, Elkon R, Shamir R. Clustering Analysis and its Application to Gene Expression Data. 2001
4 Han J, Kamber M. Data Mining: Concepts and Techniques. In: The Morgan Kaufmann Series in Data Management Systems, Jim Gray, Series Editor Morgan Kaufmann Publishers, ISBN 1-55860-489-8, August 2000
5 Kohonen T. Self-Organization and Associative Memory. Spring-Verlag, Berlin, 1984
6 Beyer K, Goldstein J, Ramakrishnan R, et al. When is the nearest neighbor meaningful? Lecture Notes in Computer Science, 1999, 1540:217～235
7 Hastie T, Tibshirani R, Eisen M B, et al. Gene shaving' as a method for identifying distinct sets of genes with similar expression patterns. Genome Biology, 2000, 2(1)
8 Efron B, Tibshirani R, Goss V. et al. Microarrays and Their Use in a Comparative Experiment: [Tech. report]. Stanford University, 2000
9 Tang Chun, Zhang Aidong. An Iterative Strategy for Pattern Discovery in Multi-dimensional Data Sets. In: 11 th International Conference on Information and Knowledge Management (CIKM 2002). McLean, VA, November 2002
10 Tang Chun, Zhang Li, Zhang Aidong, Ramanathan M. Interrelated Two-way Clustering: An Unsupervised Approach for Gene Expression Data Analysis. In: Proc. of 2nd IEEE International Symposium on Bioinformatics and Bioengineering. Bethesda, MD. November 2001
11 Jiang Daxin, Tang Chun, Zhang Aidong. Clustering Analysis for Gene Expression Data: A Survey. IEEE Transactions on Knowledge and Data Engineering (TKDE)

(上接第 104 页)

$=\rho'(i)$，这与 $\rho_{r7}(i)$ 是一个信任的主体的假设相矛盾。

当 $l=3$ 时，$a_p=((\theta',\rho',\sigma'),sned_3(i,r,\{|V|\}_{pk(r)}))$，为了从 $(\theta',\rho',\sigma')(i,r,\{|V|\}_{pk(r)})$ 获取 $(\theta_{r7},\rho_{r7},\sigma_{r7})(nr)$，必有 $(rid',\rho',\sigma')(V)=((rid_{r7},\rho_{r7},\sigma_{r7}))(nr)$ 且 $\rho'(r)$ 是一个非信任的主体。根据引理 2，可知存在 $i2$ 使得 $a_{i2}=((\theta',\rho',\sigma'),read_2(r,i,\{|ni,V,r|\}_{pk(i)}))$，又因 $(\theta',\rho',\sigma')(V)\notin M_p$，则据引理 3，存在 $r2$，使得 $a_{r2}=((\theta_{r2},\rho_{r2},\sigma_{r2}),send_2(r,i,\{|W,nr,r|\}_{pk(i)})$，从而 $\rho'=\rho_{r2}(r)$，且 $(\theta_{r2},\rho_{r2},\sigma_{r2})(nr)=(rid',\rho',\sigma')(V)=(rid_{r7},\rho_{r7},\sigma_{r7})(nr)$，根据引理 4，可知 $\theta_{r2}=\theta_{r7},\rho_{r2}=\rho_{r7}$，即 $\rho'(r)=\rho_{r2}(r)=\rho_{r7}(r)$，但因为 $\rho'(r)$ 是一个非信任的主体，$\rho_{r7}(r)$ 而是一个信任的主体，这就得出了矛盾。

上面的证明可以看出，入侵不能获取临时值 nr，故假设不成立，机密声明事件 $claim_7$ 得证。

(2)同步性(一种强认证性)声明事件 $claim_9$ 的证明

证明：设 $a\in Tr(nsl),r9\in N$，且 $(\theta_r,\rho_r,\sigma_{r9})\in Inst,rng(\rho r)\subseteq Agent_T,a_{r9}=((rid_r,\rho_r),claim_9(r,nisynch))$。

因为 $prec(nsl,9)=\{1,2,3\}$，所以为了证明声明事件 $claim_9$ 正确性，必须找一个执行初始角色的运行，该运行中的事件与标记 1,2,3 事件同步。根据引理 2，可知存在 $(0\leqslant r1<r2<r3<r9)$ 和 $\sigma_{r1}\subseteq\sigma_{r2}\subseteq\sigma_{r3}\subseteq\sigma_{r9}$ 使得

$$a_{r1}=((\theta_r,\rho_r,\sigma_{r1}),read_1(i,r,\{|i,W|\}_{pk(r)}))$$
$$a_{r2}=((\theta_r,\rho_r,\sigma_{r2}),sead_2(i,r,\{|W,nr,r|\}_{pk(i)}))$$
$$a_{r3}=((\theta_r,\rho_r,\sigma_{r3}),read_1(i,r,\{|nr|\}_{pk(r)}))$$

从 $claim_7$ 证明过程中可知 nr 满足机密性，因此根据引理 3，存在 i_3 和 $(\theta_i,\rho_i,\sigma_{i3})$，对于 $i_3<r_3$ 和 $a_{i3}=((\theta_i,\rho_i,\sigma_{i3}),sned_3(i,r,\{|V|\})_{pk(r)})\wedge(\theta_r,\rho_r,\sigma_{r3})(nr)=(\theta_i,\rho_i,\sigma_{i3})(V)$，根据引理 2，可知存在 $i1<i2<i3$，使得

$$a_{i1}=((\theta_i,\rho_i,\sigma_{i1}),read_1(i,r,\{|i,ni|\}_{pk(r)}))$$
$$a_{i2}=((\theta_i,\rho_i,\sigma_{i2}),read_2(i,r,\{|ni,V,r|\}_{pk(i)}))$$
$$a_{i3}=((\theta_i,\rho_i,\sigma_{i3}),read_1(i,r,\{|V|\}_{pk(r)}))$$

现在仅需证明运行 θ_i 是运行 θ_r 满足同步性，为此，创建 $r_1<i_2,i_1<r_1$ 并且使它们对应的发送事件和读事件是相互匹配的。

首先，对于 a_{i2}，因为 $(\theta_r,\rho_r,\sigma_{r3})(nr)$ 是机密的，$(\theta_i,\rho_i,\sigma_{i2})(V)$ 也是机密，所以根据引理 3，存在 $r2'<i2$，使得 $a_{r2'}=((\theta_r,\rho_r,\sigma_{r2}))$，$send_2(r,i,\{|W,nr,r|\}_{pk(i)})$ 从而 $(\theta_i,\rho_i,\sigma_{i2})(\{|ni,V,r|\}_{pk(i)})=(\theta_{r'},\rho_{r'},\sigma_{r2'})(\{|W,nr,r|\}_{pk(i)})$，进而可知 $(\theta_r,\rho_r,\sigma_{r3})(nr)=(\theta_i,\rho_i,\sigma_{i3})(V)=(\theta_{r'},\rho_{r'},\sigma_{r2'})(nr)=(\theta_i,\rho_i,\sigma_{i3})(V)=(\theta_{r'},\rho_{r'},\sigma_{r2'})(nr)$，所以，根据引理 4，可知 $\theta_r=\theta_{r'}$，而且 $r2=r2'$，即证明了事件 $ai2$ 和 $ar2$ 是同步的。

其次，对于 $ar1$，因为 $(\theta_r,\rho_r,\sigma_{r1})(W)$ 是秘密的，所以根据引理 3，存在 $i1'<r1$，使得 $a_{i1'}=((\theta_{i'},\rho_{i'},\sigma_{i1'}))$，$send_1(i,r,\{|i,ni|\}_{pk(r)})$ 和 $(\theta_r,\rho_r,\sigma_{r1}))(\{|i,W|\}_{pk(r)})=(\theta_{i'},\rho_{i'},\sigma_{i1'})(\{|i,ni|\}_{pk(r)})$，则两个对应的 a_{i2} 和 a_{r2} 满足 $(\theta_i,\rho_i,\sigma_{i2})(ni)=(\theta_r,\rho_r,\sigma_{r2})(W)=(\theta_{i'},\rho_{i'},\sigma_{i1'})(ni)$。从而根据引理 4 可知 θ_i 和 $\theta_{i'}$ 是相等的，即证明了事件 a_{r1} 和 a_{i1} 是同步的，因此，同步性声明事件 $claim_9$ 得到证明。

结束语 本文在对操作语义模型进行研究的基础上，构建了基于结构化操作语义的安全协议分析框架，该框架不仅可以分析具体的安全协议，更重要的是，它为安全协议的形式化分析提供了一个统一的语义模型。在后继的工作中我们将用此框架进一步规范更多的安全性质并研究基于此框架的自动推理技术。

参 考 文 献

1 Meadows C. Formal methods for cryptographic protocol analysis: Emerging issues and trends. IEEE Journal on Selected Areas in Communications, 2003, 21(1):44～54
2 Thayer F J, Herzog J C, Guttman J D. Strand spaces: Why is a security protocol correct? In: Proc. of the 1998 IEEE Symposium on Security and Privacy Los Alamitos: IEEE Computer Society Press, 1998. 160～171
3 Cremers C J F, Mauw S. Operational Semantics of security protocols. Scenarios: Models, Transformations and Tools. In: International Workshop Models, Dagstuhl Castle, Germany: Springer, 2005. 66～89
4 Cremers C J F, Mauw S, Vink E P. Defining Authentication in A Trace Model. In: Proc. of the First International Workshop on Formal Aspects in Security and Trust, Pisa, Italy, 2003. 131～145
5 Cremers C J F. Compositionality of Security Protocols: A Research Agenda. Vodca 2004 ENTCS, 2006, 142(3):99～110
6 Lowe G. A Hierarchy of Authentication Specifications. In: Proceedings of the 10th IEEE Computer Security Foundations Workshop, IEEE Computer Society Press, 1997. 31～43