基因表达数据中局部模式的查询

姜 涛 李战怀 尚学群 陈伯林 李卫榜
(西北工业大学计算机学院 西安 710072)

摘 要 基因表达数据分析一般是通过挖掘局部模式来实现的。保序子矩阵是局部模式挖掘中一种经典的模型,可以获取到在若干条件下表现出一致趋势的一组基因。高通量基因微阵列技术的进步,促进了海量基因表达数据的产生,使得对高性能基因表达数据分析算法的需求极为迫切。现有方法大多数是通过批量挖掘的方法来分析数据,即使有通过查询方式来获取精确结果的方法,其全面性与性能也有待提高。为了提高数据分析的效率与准确性,首先提出一种基于前缀树的基因表达数据索引 gIndex,然后给出了一种基于列关键词查询的保序子矩阵分析方法 GEQ$_e$。其不经过批量挖掘,只需要建立索引并通过关键词来完成正相关/负相关/时滞等模式的查询。实验结果表明,与现有方法相比,所提算法具有良好的数据分析效率与可扩展性。

关键词 基因表达数据,局部模式,保序子矩阵,关键词查询
中图法分类号 TP311.13 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.9.035

Local Pattern Query from Gene Expression Data

JIANG Tao LI Zhan-huai SHANG Xue-qun CHEN Bo-lin LI Wei-bang
(School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China)

Abstract Local pattern mining plays an important role in gene expression data analysis. One classical model in local pattern mining is order-preserving subMatrix (OPSM), which captures the general tendency of subset of genes in subset of conditions. With the development of high-throughput gene microarray techniques, it produces massive of gene expression datasets. In this situation, it is urgent to design high performance algorithms. Most of the existing methods are batch mining technique, even though it can be addressed by query method, the comprehensiveness and behaviors still should be improved. To make data analysis efficient and accurate, we first proposed a prefix-tree based indexing method for gene expression data, then gave a column keyword based OPSM query methods. It uses index and search method instead of batch mining to query positive, negative and time-delayed OPSMs. We conducted extensive experiments and compared our method with existing methods. The experimental results demonstrate that the proposed method is efficient and scalable.

Keywords Gene expression data, Local pattern, Order-preserving submatrix, Keyword-based query

1 引言

高通量基因微阵列技术的发展,促进了大量基因表达数据的产生。基因表达数据实际上反映的是测量到的基因转录产物中的 mRNA 的丰度[1]。相比其他生物,由于人体环境变化多样且基因表达随着时空的变化而改变,因此基因表达数据更为复杂、数据量更大、增长速度更快。基因微阵列之上的基因表达数据可以看作 $n \times m$ 的矩阵,其中 n 为基因数目(行数),m 为实验条件个数(列数),矩阵中的每个属性值代表某个基因在某个实验条件下的表达水平。包括基因表达数据在内的生物信息数据催生了数据挖掘(尤其是双聚类)技术的发展。现有大多数聚类方法通常要寻找的是在所有实验条件下共表达的基因,但是实验表明基因并不一定在所有实验条件下表达同样的功能。同样,共表达的基因表达水平也是不尽相同的。在这种情况下,发现若干基因在若干实验条件下表现出一致上升或者下降趋势的保序子矩阵(Order-Preserving SubMatrix, OPSM)模型应运而生。保序子矩阵的挖掘为基因功能预测、发现基因协同表达网络、研制药物、预防疾病等提供了技术支持。尽管批量挖掘方法在大多数应用场景下很有效,但是由于批量结果过于庞大,使得生物学家对数据的分

到稿日期:2016-04-18 返修日期:2016-06-07 本文受国家"九七三"重点基础研究发展规划(2012CB316203),国家"八六三"高技术研究发展计划(2015AA015307),国家自然科学基金重点项目(61033007,61332014),国家自然科学基金面上项目(61272121,61572367),中央高校基础研究经费项目(3102015JSJ0011)资助。

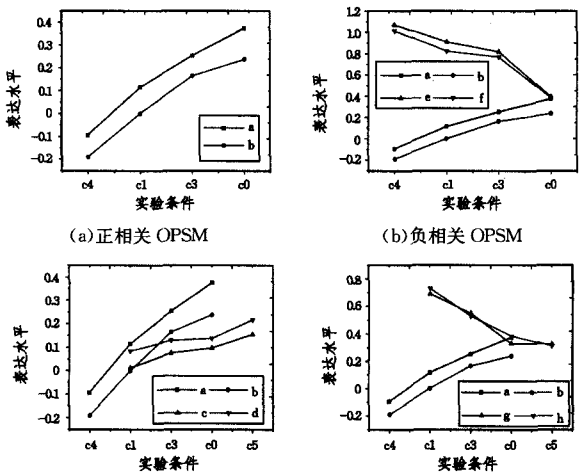
姜 涛(1983—),男,博士生,CCF 学生会员,主要研究方向为生物信息挖掘、数据管理、大数据分析,E-mail:jiangtao@mail.nwpu.edu.cn;**李战怀**(1961—),男,博士,教授,CCF 高级会员,主要研究方向为数据管理、数据挖掘;**尚学群** 女,博士,教授,CCF 高级会员,主要研究方向为生物信息学、数据挖掘、数据管理;**陈伯林** 男,博士,副教授,CCF 会员,主要研究方向为生物信息学、数据挖掘、数据管理;**李卫榜**(1979—),男,博士生,主要研究方向为数据管理、数据挖掘、大数据分析。

析效率较低。本研究试图从索引与查询的角度给出新的数据检索方法。

自 Hartigan 等人[2]发表开创性成果之后，即将矩阵分为若干个含有近似值的子矩阵，双聚类方法得到巨大的发展。Cheng 和 Church[3]首先将双聚类方法应用于基因表达数据的分析中。Ben-Dor 等人[4]设计了一种称为保序子矩阵的模型(OPSM)，用来发现特定类型的双聚类。随后，研究者提出了基于定量[3,4]和定性[5]标准的 OPSM 挖掘方法、约束型聚类[6]、基于查询的双聚类[7]、基于关键词的 OPSM 查询[8]等方法。虽然上述方法性能很优越，但是每种双聚类方法都将算法的应用领域限制在了发现特定类型的双聚类中，没有很好的通用性。因此，随着基因表达数据获取能力的增强，亟待寻找新的、通用的且性能更好的数据挖掘与管理方法。

在设计通用性强且性能较好的新方法的过程中，存在诸多的挑战：1)在不进行批量数据挖掘的前提下，如何保证索引的高效性与索引数据的最小化目标；2)如何保证输入极少的关键词且能返回相关性较高的结果；3)如何提供多种类型的 OPSM 的查询也是用户比较关注的一个问题。

所关注的多类型 OPSM 主要包括正相关 OPSM、负相关 OPSM、时滞正相关 OPSM、时滞负相关 OPSM，如图 1 所示。正相关的 OPSM 指若干基因在若干实验条件下表达趋势一致，如图 1(a)所示；负相关 OPSM 则指在若干相同的实验条件下，部分基因正向表达，部分基因反向表达，如图 1(b)所示；时滞正相关 OPSM 与正相关 OPSM 基本相同，不同点是其中部分基因的表达延迟若干个时间点，如图 1(c)所示；时滞负相关 OPSM 与负相关 OPSM 基本相同，不同点是其中部分基因的表达延迟若干个时间点，如图 1(d)所示。



(a)正相关 OPSM　(b)负相关 OPSM

(c)时滞正相关 OPSM　(d)时滞负相关 OPSM

图 1　多种类型 OPSM 举例

由于 OPSM 检索的研究工作处于萌芽阶段，因此相关研究成果还比较少。与本研究最相关的研究工作为文献[8,9]。文献[8]提出基于前缀树的索引 pIndex，同时给出了基于行/列关键词的精确/模糊查询方法。该方法虽然性能优越，但是只能提供正相关类型的 OPSM 的查询，无法满足本研究的需求。文献[9]为了保证查询结果具有较高的相关性，提出了基于用户自定义约束(必须连接、不能连接、间隔约束、数目约束等)与多维联合索引的 OPSM 查询方法。同样，其也只能保证正相关类型的 OPSM 的查询。而本研究的目的是提供正相关、负相关、时滞等多种类型 OPSM 的查询工作。

为了解决上述问题，首先给出一种基于前缀树的索引方法 gIndex，该方法为基于关键词的后续快速遍历提供了条件。接着，提出一种基于关键词的多种类型 OPSM 查询方法 GEQ_c，其有效地丰富了用户的使用选择。最后，在真实数据上进行了广泛的实验，实验结果证明了所提方法的有效性与高效性。

本文的创新点如下：

(1)弥补了现有工作[8]的不足，增加了负相关/时滞相关模式的检索；

(2)提出了一种基于列关键词的多类型 OPSM 查询方法；

(3)进行了广泛的实验，实验结果证明了所提方法的有效性与高效性。

本文第 2 节介绍了与本研究相关的工作；第 3 节给出了基础概念和多类型 OPSM 查询问题描述；第 4 节描述了索引的创建方法，以及正相关、负相关、时滞等类型 OPSM 的查询方法；第 5 节给出了实验结果；最后总结全文。

2　相关工作

双聚类的概念最初由 Hartigan 等人[2]提出，其是对矩阵中的行与列同时聚类的一种方法，被命名为 Direct 聚类。Cheng 和 Church[3]将“并不是所有列都属于同一聚类”的理念应用在了基因表达数据分析中。Ben-Dor 等人[4]介绍了一种特殊的双聚类模型 OPSM，并证明了其是 NP 难问题。随后，研究者们提出了基于定量测度和定性测度的 OPSM 挖掘方法。数值测度包括均方残差 MSR[4]、平均相关值 ACV[10]、平均斯皮尔曼秩相关系数 ASR[10]、平均一致性相关指数 ACSI[10]等。定性测度包括上升、下降和无变化[5]。除上述两类方法外，还有约束型双聚类[6]和基于查询的双聚类[7]。

基于定量测度的双聚类：Cheng 等人[3]介绍了一种删除节点的方法，从基因表达数据中挖掘那些均方残差 MSR 比较小的子矩阵。其克服了传统方法的一些缺点，使得可以自动发现部分属性下具有相似表达的若干基因。基于 Cheng 等人[3]提出的 δ-bicluster 模型，Yang 等人[11]为减小数据缺失值的影响，给出一种 δ-cluster 模型。Cho 等人[12]给出两种与 MSR 类似的平方和残留度量标准，同时提出两种快速的基于 K 中值的双聚类算法来有效地挖掘双向聚类。Chen 等人[13]利用最小均方错误 MMSE 测度来鉴别多种类型的线性模式。Denitto 等人[14]利用 Max-Sum 测度来提升双聚类的质量。

基于定性测度的双聚类：Zhao 等人[15]提出子空间聚类方法 g-Cluster 来发现具有正负相关的共调控基因。Chen 等人[16]给出一种基于上下位模式的方法来发现正负相关的基因聚类。Zhao 等人[17]介绍了一种发现最大子空间共调控基因聚类方法。Wang 等人[18,19]设计了发现时滞表达模式与局部保守聚类的方法。印莹等人[20]提出一种从时序微阵列数据中挖掘同步和异步共调控基因聚类的方法。Painsky 等人[21]为了发现“每行只属于一类而每列可以属于多个聚类”的双聚类，介绍了基于最优集合覆盖的方法。Henriques 等人[22]提出名为 BicSPAM 的方法，其是第一个试图解决 OPSM 允许对称并且能够容忍不同级别的噪声的方法。Trapp 等人[23]介绍了两种精确的动态规划公式来发现具有相同或相反趋势的 OPSM。Jiang 等人[24]提出一种基于蝶形

网络的并行分割与挖掘方法来扩展并改善 OPSM 的挖掘性能。

对于正相关、负相关、时滞等多种类型 OPSM 的查询问题，研究工作还比较少。与本问题最相关的工作为文献[8,9,25]。Jiang 等人[8]在带有列表头的前缀树索引的基础上，提出了基于行/列关键词的精确/模糊查询方法，与本研究不同，其支持的查询类型较少，只提供正相关 OPSM 的搜索。随后，文献[25]设计并实现了 OPSM 的挖掘、索引与查询工具 OMEGA。为了从 OPSM 数据中找出更相关的聚类，Jiang 等人[9]给出一种 OPSM 的约束型查询方法。其中包括基于枚举序列索引的查询方法和多维联合查询方法。前者只提升了查询效率，后者既提升了查询效率又减少了索引的数据量。

3 问题描述

本节主要介绍相关概念与解决正相关、负相关和时滞的 OPSM 查询的问题描述，用到的相关符号如表 1 所列。如果后文中没有特别说明，术语“行”与“基因”、“列”与“实验条件”将交替使用，因为其在本文中具有相同的含义。

表 1 本文用到的符号

符号	描述	符号	描述
G	基因集合	C	实验条件集合
g	部分基因	c	部分实验条件
g_i	一个基因	c_i	一个实验条件
D(G,C)	源数据集	e_{ij}	D 中的一条属性
δ	行阈值	d	滞后时间点个数

定义 1(正相关保序子矩阵，POPSM) 给定数据 $D(G,C)$($n\times m$ 的矩阵)，$M_i(g,c)$ 是 D 中的一个子矩阵，且 $g\subseteq G$，$c\subseteq C$。若 M_i 是一个正相关保序子矩阵，则有 g 中的每一行数据 e 关于列标签子集 c 的排列严格单调递增或递减，即 $e_{i1}\leqslant e_{i2}\leqslant\cdots\leqslant e_{ij}\leqslant\cdots\leqslant e_{ik}$，或 $e_{i1}\geqslant e_{i2}\geqslant\cdots\geqslant e_{ij}\geqslant\cdots\geqslant e_{ik}$，其中$(i1,\cdots,ij,\cdots,ik)$是列标签 c 的一个排列。

例 1 如图 1(a)所示，基因 a,b 在实验条件 c_4,c_1,c_3,c_0 上的表达值严格单调递增，所以基因 a,b 在实验条件 c_4,c_1,c_3,c_0 上是正相关 OPSM。

定义 2(负相关保序子矩阵，NOPSM) 给定数据 $D(G,C)$($n\times m$ 的矩阵)，$M_i(g,c)$ 是 D 中的一个子矩阵，且 $g\subseteq G$，$t\subseteq C$。若 M_i 是一个负相关保序子矩阵，则有 $g_{up}\subseteq g$ 中的每一行数据关于列标签子集 c 的排列严格单调递增，即 $e_{i1}\leqslant e_{i2}\leqslant\cdots\leqslant e_{ij}\leqslant\cdots\leqslant e_{ik}$，$g_{down}\subseteq g$ 中的每一行数据关于列标签子集 c 的排列严格单调递减，即或 $e_{i1}\geqslant e_{i2}\geqslant\cdots\geqslant e_{ij}\geqslant\cdots\geqslant e_{ik}$，其中$(i1,\cdots,ij,\cdots,ik)$是列标签 c 的一个排列，$g_{up}\cup g_{down}=g$。

例 2 如图 1(b)所示，基因 a,b 在实验条件 c_4,c_1,c_3,c_0 上的表达值严格单调递增，而基因 e,f 在实验条件 c_4,c_1,c_3,c_0 上的表达值严格单调递减，所以基因(a,b)与(e,f)在实验条件 c_4,c_1,c_3,c_0 上是负相关 OPSM。

定义 3(时滞保序子矩阵，LOPSM) 给定数据 $D(G,C)$($n\times m$ 的矩阵)，$M_i(g,c)$ 是 D 中的一个子矩阵，且 $g\subseteq G$，$t\subseteq C$。若 M_i 是一个时滞保序子矩阵，则有 $g_{forward}\subseteq g$ 中的每一行数据关于列标签子集 $c_{forward}(c_{i1},\cdots,c_{ij},\cdots,c_{ik})$的排列严格单调递增，即 $e_{i1}\leqslant e_{i2}\leqslant\cdots\leqslant e_{ij}\leqslant\cdots\leqslant e_{ik}$；$g_{lag}\subseteq g$ 中的每一行数据关于列标签子集 $c_{lag}(c_{i'1},\cdots,c_{i'j},\cdots,c_{i'k})$的排列严格单调递增，即 $e_{i'1}\leqslant\cdots\leqslant e_{i'j}\leqslant\cdots\leqslant e_{i'k}$，其中 $c_{i1}-c_{i'1}=\cdots=c_{ij}-c_{i'j}=\cdots=c_{ik}-c_{i'k}=d$，$d$ 是延迟时间点的个数。当每一行数据关于列标签子集的排列严格单调递减时，定义同样成立。

例 3 如图 1(c)所示，基因 a,b 在实验条件 c_4,c_1,c_3,c_0 上的表达值严格单调递增，且基因 c,d 在实验条件 c_1,c_3,c_0,c_5 上的表达值严格单调递增，后者比前者延迟了一个时间点，所以基因(c,d)在实验条件 c_1,c_3,c_0,c_5 上是基因(a,b)在实验条件 c_4,c_1,c_3,c_0 上的时滞正相关 OPSM。

例 4 如图 1(d)所示，基因 a,b 在实验条件 c_4,c_1,c_3,c_0 上的表达值严格单调递增，而基因 g,h 在实验条件 c_1,c_3,c_0,c_5 上的表达值严格单调递减，后者比前者延迟了一个时间点，所以基因(g,h)在实验条件 c_1,c_3,c_0,c_5 上是基因(a,b)在实验条件 c_4,c_1,c_3,c_0 上的时滞负相关 OPSM。

问题描述：给定数据集 $D(G,C)$($n\times m$ 的矩阵)、时滞时间点数 d 和最小基因个数 δ，查询出所有满足定义 1—定义 3 类型的 OPSM。

4 基于前缀树的多类型 OPSM 查询

本节提出一种基于前缀树索引的支持多类型 OPSM 查询的方法。其有效地将数值型搜索问题转换成了特殊序列模式查询问题，主要包括创建索引与 OPSM 查询两部分。

4.1 创建索引

(1)预处理

首先，将原始基因表达数据的每一行数据按照升序排列，具体排序方法可以利用经典的快速排序算法。接着，将每一行数据中的每个元素替换成排序前所在列的标签。这样，原始数据就变成了序列数据。

例 5 将表 2 中基因 a 在所有实验条件下的表达值排序之后，得到数组向量〈−0.201，−0.181，−0.094，0.115，0.254，0.375〉。接着将各个数字替换为排序前所在的列标签，得到序列向量〈c_2,c_5,c_4,c_1,c_3,c_0〉，如表 3 首行所列。表 2 中其它基因表达数据的排序与替换结果如表 3 所列。

表 2 基因表达数据样例(行为基因，列为实验条件)

	c_0	c_1	c_2	c_3	c_4	c_5
a	0.375	0.115	−0.201	0.254	−0.094	−0.181
b	0.238	0	0.150	0.165	−0.191	0.132
c	0.097	0.013	0.284	0.076	—	0.155
d	0.138	0.084	−0.159	0.129	—	0.217
e	0.394	0.909	0.443	0.818	1.070	0.227
f	0.385	0.822	0.426	0.768	1.013	0.226
g	0.329	0.690	0.244	0.550	—	0.327
h	0.384	0.730	0.066	0.529	—	0.313

表 3 序列矩阵

	1	2	3	4	5	6
a	c_2	c_5	c_4	c_1	c_3	c_0
b	c_4	c_1	c_5	c_2	c_3	c_0
c	c_1	c_3	c_0	c_5	c_2	
d	c_2	c_1	c_3	c_0	c_5	
e	c_5	c_0	c_2	c_3	c_1	c_4
f	c_5	c_0	c_2	c_3	c_1	c_4
g	c_2	c_5	c_0	c_3	c_1	
h	c_2	c_5	c_0	c_3	c_1	

定理 1(排序的合理性与有效性) 给定 n 行 m 列的基因表达数据，将每行表达数据排序并替换为列标签的作法是合理并且有效的。

证明：本质上，OPSM 是在若干实验条件下表现出相同或

者相反表达趋势的一种模式。这种表达趋势就是前一值相对于后一值的上升或者下降。如果在挖掘、索引或查询过程中再去比较每两列之间的大小，则势必需要较长的响应时间。如果将这个两两比较的过程提前离线处理，则会大大提高后续在线处理的性能。所以对基因表达数据排序并替换为列标签的做法是合理并且有效的。

定理 2(预处理时间复杂度) 给定 n 行 m 列的基因表达数据，则在理想情况下，预处理步骤的时间复杂度为 $O(nm\log m)$；最坏情况下，预处理步骤的时间复杂度则为 $O(nm^2)$。

证明：快速排序算法在理想与最坏情况下的时间复杂度分别为 $O(m\log m)$ 与 $O(m^2)$。因为预处理的数据为 n 行，所以预处理步骤的时间复杂度在理想与最坏情况下的时间复杂度分别为 $O(nm\log m)$ 与 $O(nm^2)$。

(2)创建 gIndex 索引

本文借鉴前期的研究成果中的 pIndex 索引[8]。基本方法分为两部分：1)创建前缀树；2)创建列表头。在创建前缀树的过程中，每一个列标签放置在一个节点中，随后将共享该列标签序列的基因名放在该列分支所在的叶子节点中。同时，每创建一个树节点，遍历一次列表头，检查是否存在该列标签。如果不存在，则将该列标签放置在列表头中的主键中，并将值所在位置与该树节点所在地址用指针链接；如果存在，则同样将该列标签所指向的最后一个节点与本节点用指针链接。最后，将基因名放置在该分支结束的列标签所在的节点中。

例 6 假如表 2 中序列的输入顺序为 c,d,a,g,h,b,e,f。首先创建索引的第一个分支 $c_1c_3c_0c_5c_2$，其中每一个列标签存放在一个节点中，同时按照列标签出现的先后顺序建立列表头，在遍历到分支的最后一个节点时，将基因名放置在其中。接着，创建索引的第二个分支 $c_2c_1c_3c_0c_5$，由于 c_2 已经存在于列表头中，只需要将上个分支中 c_2 所在节点的指针指向该节点。其它节点与分支的创建与上述过程类似。最终创建好的 gIndex 索引如图 2 所示。

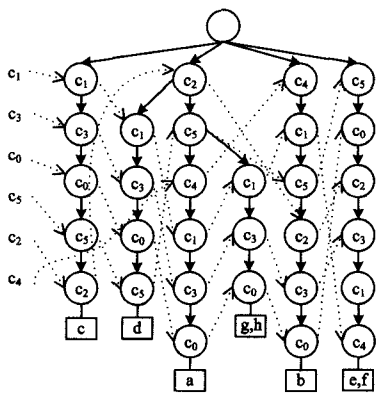


图 2 gIndex 索引举例

定理 3(gIndex 的时间复杂度) 给定数据 $D(G,C)$($n\times m$ 的矩阵)，则创建 gIndex 的时间复杂度为 $O(nm^2)$。

4.2 查询方法

多类型 OPSM 查询方法的核心是基于列的精确查询 EQ_c。首先通过列表头来确定列关键词在索引中的位置；接着从定位到的关键词所在分支节点开始，以自底向上的方式来遍历该分支，检验该分支是否包含所有列关键词，且检查这些关键词在分支中的顺序是否一致；如果一致，则返回那些所含基因个数大于阈值 σ 的分支。具体细节参考算法 1。

定理 4(列表头与自底向上遍历相结合的有效性) 在查询过程中，列表头与自底向上遍历相结合的搜索方法是有效的。

证明：给定列关键词后，首先搜索表头，找到最后一个关键词所在链接的首节点。在这一点上减少了该节点以下分支的搜索。接着以自底向上的方式遍历每一个候选分支。如果该分支节点中保存的列关键词顺序和数量都符合条件，则立即终止该分支的遍历；反之，则终止的时间可能更早。而传统的基于自顶向下的搜索方式首先要从根节点开始遍历，一旦某个分支不符合条件，还要回溯到根节点并选择新的分支进行搜索。相比于本文提出的方法，其多了许多不必要的回溯。所以列表头与自底向上遍历相结合的搜索方法是有效且高效的。

算法 1 基于列的精确查询(EQ_c)

输入：实验条件关键词 c，基因个数阈值 σ

输出：存放有〈c,g〉键值对的哈希表 result

```
1. key←c 中最后一个关键词;
2. keyNode←列表头中 key 所在节点;
3. if (keyNode 为空) then return 空;
4. while (keyNode 不为空) do
5.    itNode ← keyNode 的父节点;count ← |c|−2;
6.    while (count ≥ 0) do
7.       if (c 中第 count 个关键词= itNode 中的列标签) then
         count − −;
8.       if (itNode 为根节点) then break;
9.       else itNode← itNode 的父节点;
10. if (count < 0) then 将 keyNode 放入 nodes;
11. keyNode ← keyNode 所在的下一个分支节点;
12. for (nodes 中的每一个节点 node) do
13.      nameSet ← 从节点 node 所在分支中取出所有基因名;
14. if (nameSet. size()≥σ) then 将键值对〈c,nameSet〉放入 result
    中;
```

定理 5(基于列的精确查询算法的时间复杂度) 给定 n 行序列数据创建的 gIndex 索引与列关键词 c，则基于列的精确查询算法的时间复杂度为 $O(n|c|)$。

证明：首先通过列表头进行最后一个关键词的定位，由于基本上 n 行数据中都含有该关键词，因此定位分支的数量为 n。接着要搜索并验证一下每个分支中是否有 $|c|$ 个关键词，所以基于列的精确查询算法的时间复杂度为 $O(n|c|)$。

在基于列的精确查询算法 EQ_c 的基础上，提出了多类型 OPSM 查询算法 GEQ_c。当查询正相关 OPSM 时，直接调用 EQ_c 算法(算法 2 第 1 行)。对于负相关 OPSM 查询，首先反转实验条件关键词，接着调用 EQ_c 算法(算法 2 第 2,3 行)。对于时滞不超过 d 的正相关 OPSM 查询，首先获取关键词序列 c 中从第 i 个元素开始的序列，其中 $0\leqslant i\leqslant d$，接着调用 EQ_c 算法(算法 2 第 4－7 行)。对于时滞不超过 d 的负相关 OPSM 查询，首先获取关键词序列 c 中从第 i 个元素开始的序列，接着调用 EQ_c 算法(算法 2 第 8－11 行)。多类型 OPSM 查询算法如算法 2 所示。

算法 2 多类型 OPSM 查询 (GEQ_c)

输入：实验条件关键词 c，时滞时间点数 d，基因个数阈值 σ

输出：存放有〈c,g〉键值对的哈希表 result

```
1. EQc(c,σ);// 正相关 OPSM 查询
```

2. $c_{reverse}$←列关键词 c 的反向序列；
3. $EQ_c(c_{reverse},\sigma)$；//负相关 OPSM 查询
4. for (i←1 to 最大时滞 d) do
5. col←c 中从第 i 个元素开始的序列；
6. EQ_c(col,σ)；// 时滞正相关 OPSM 查询
7. 验证并剔除假阳性搜索结果；
8. for (i←1 to 最大时滞 d) do
9. col←$c_{reverse}$中从第 i 个元素开始的序列；
10. EQ_c(col,σ)；//时滞负相关 OPSM 查询
11. 验证并剔除假阳性搜索结果；

例 7　给定根据表 3 中的数据创建的如图 2 所示的索引、列关键词 c_4,c_1,c_3,c_0 和基因个数阈值 2,查询符合条件的正相关 OPSM。

首先从列表头中查询包含关键词 c_0 的分支,再以自底向上的方式遍历各个分支,同时检测其中是否依次包含 c_3,c_1,c_4,然后得到基因 a,b 所在的分支符合条件,最后得到〈c_4、c_1、c_3、c_0:a、b〉为正相关 OPSM(见图 1(a))。

例 8　给定根据表 3 中的数据创建的如图 2 所示的索引,列关键词 c_4,c_1,c_3、c_0 和基因个数阈值 2,查询符合条件的负相关 OPSM。

与例 7 的唯一不同是反转关键词。查询过程同上,最后得到〈c_4、c_1、c_3、c_0:e、f〉为〈c_4、c_1、c_3、c_0:a、b〉的负相关 OPSM(见图 1(b))。

例 9　给定根据表 3 中的数据创建的如图 2 所示的索引、列关键词 c_4,c_1,c_3,c_0、时滞时间点数 1 和基因个数阈值 2,查询符合条件的时滞正相关 OPSM。

首先搜索包含 c_1,c_3,c_0 的分支,接着一一验证是否真正符合条件,然后找出基因 c,d 所在的分支符合条件,最后得到〈c_1、c_3、c_0、c_5:c、d〉为〈c_4、c_1、c_3、c_0:a、b〉的时滞正相关 OPSM(见图 1(c))。

例 10　给定根据表 3 中的数据创建的如图 2 所示的索引、列关键词 c_4,c_1,c_3,c_0、时滞时间点数 1 和基因个数阈值 2,查询符合条件的时滞负相关 OPSM。

与例 9 的唯一不同是反转关键词。查询过程同上,最后得到〈c_1、c_3、c_0、c_5:g、h〉为〈c_4、c_1、c_3、c_0:a、b〉的负相关 OPSM(见图 1(d))。

4.3　优化方法

通过例子可以发现,在搜索过程中利用关键词的个数以及关键词的顺序来及时地剪枝(见规则 1 与规则 2),及早地剪掉不符合条件的分支,可以有效地提升算法的搜索效率。

规则 1(基于列关键词个数的剪枝)　在对 gIndex 索引遍历的过程中,一旦发现索引节点中存储的列关键词个数少于所输入的关键词个数,可以立即停止该分支的遍历。

证明:因为用户指定了关键词,显然其应该包含在查询结果之中。

规则 2(基于列关键词顺序的剪枝)　在对 gIndex 索引遍历的过程中,一旦出现索引节点中存储的顺序与所输入的关键词顺序不一致,可以立即停止该分支的遍历。

证明:OPSM 顾名思义,即若干行在若干列下的表达趋势一致,所以其对列的顺序相对敏感。因此查询结果一旦不符合用户输入的关键词顺序,理应及早停止对该分支的遍历。

经过分析发现,规则 2 的剪枝效率要高于规则 1,所以一般情况下使用规则 2。在顺序一致且下一个节点就是根节点的情况下,才使用规则 1。

5　实验评估

本节主要评估 pfTree[8],pIndex[8] 与 gIndex 这 3 种索引方法,以及基于 3 种索引方法的多种类型 OPSM 的查询算法的有效性与高效性。实验中用到了两种数据:真实数据和生成数据。大多数实验是在真实数据上进行的,因为其是真实需求的来源。实验用到的设备是 1.87GHz 频率、16GB 内存且运行着 Ubuntu 14.04 的浪潮服务器。所有方法均用 Java 语言编写,由 Eclipse 4.3 编译运行。

数据生成方法:先从网站[26]上下载如表 4 所列的 6 个数据集;接着利用快速排序法对每一行的表达值排序;最后将每一个表达值替换成对应的列标签,使其变成序列数据[24]。

表 4　实验中用到的基因表达数据集

数据集	文件名	行数	列数
D_1	adenoma	12488	6
D_2	a549	22283	11
D_3	5q_GCT_FILE	22278	24
D_4	krasla	12422	50
D_5	bostonlungstatus	12625	94
D_6	bostonlungsubclasses	12625	202

实验主要验证的内容为:

(1)gIndex,pIndex 和 pfTree 3 种索引的大小基本相等,当数据集中实验条件个数较少时,3 种方法索引的数据量只是原始数据的 2%。

(2)虽然 pfTree 在索引的创建等方面稍稍优于 gIndex 与 pIndex,但是基于 gIndex 与 pIndex 索引的 OPSM 查询方法要优于基于 pfTree 索引的查询方法 1 到 2 个数量级。

(3)另外,基于 gIndex 索引的 OPSM 查询方法 GEQ_c 等支持正相关、负相关、时滞等多种类型的 OPSM 的查询。

本文所提方法的查询准确率都为 100%,而其他方法大多数是批量挖掘方法,不能输入具体的关键词,没有可比性,所以本文就不再比对具体的查询准确率。

5.1　索引性能

首先评估所提方法在如表 4 所列的数据集上创建索引的大小。由于 pfTree,pIndex 与 gIndex 3 种索引方法都是前缀树的变种,索引大小基本一致,因此这里统一给出索引与原始数据的比值,如图 3 所示。当数据集中的实验条件个数为 6 时(D_1),随着行数的增长,索引的大小占原始数据量的比重由 8%左右降低到 2%。当数据集中的实验条件个数为 11 时(D_2),随着行数的增长,索引的大小占原始数据量的比重由 73%左右降低到 60%。同样,当实验条件个数增多时,比如 24,50,94,202 时,索引占原始数据的比值变化分别为 90%到 87%、97%到 95%、99%到 98%、99.5%到 99.2%。本实验表明在实验条件维度相对较小的情况下,基于前缀树的索引具有相对较好的压缩性能。

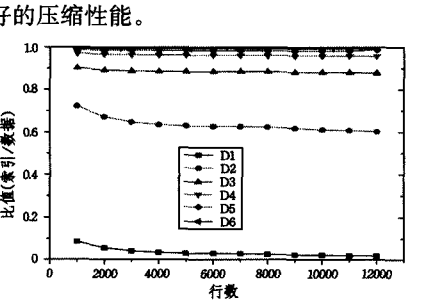


图 3　在 6 个数据集上,索引大小随着行数增长的变化情况

接下来评估 pfTree,pIndex 与 gIndex 3 种索引方法在如表 4 所列的数据集上创建索引的时间,如图 4 所示。本实验用到的数据集为 D_6。当数据集中的行数由 1000 增长到 12000 时,pfTree 方法的索引时间由 0.6s 增长到 3.4s,pIndex 方法的索引时间由 8.8s 增长到 1585.5s,gIndex 方法的索引时间由 8.6s 增长到 1392.2s。虽然 pfTree 的索引创建时间远远短于 pIndex 与 gIndex 索引方法的,但是其查询性能却远逊于后两种方法,而用户关注较多的也正是查询的响应时间。另外,由于 gIndex 索引比 pIndex 方法少创建一个辅助表头,因此其索引的耗时要少于后者,这一现象在行数较多时表现得尤为明显。

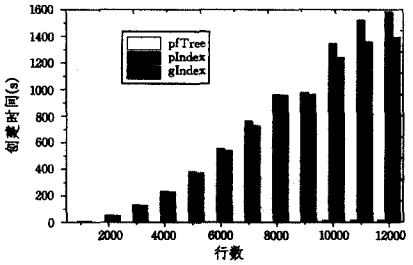


图 4 行数变、列数不变的情况下,索引算法在单机上的运行时间

5.2 查询性能

本节评估 5 种查询方法的性能。5 种方法分别为:1)基于 pfTree 索引与列关键词的精确查询方法 EQ_c-pfTree;2)基于 pIndex 索引与列关键词的精确查询方法 EQ_c-pIndex;3)基于 gIndex 索引与列关键词的负相关 OPSM 精确查询方法 GEQ_c-nega;4)基于 gIndex 索引与列关键词的时滞正相关 OPSM 精确查询方法 GEQ_c-posi_delay;5)基于 gIndex 引与列关键词的时滞负相关 OPSM 精确查询方法 GEQ_c-nega_delay。由于基于 gIndex 索引与列关键词的正相关 OPSM 精确查询方法 GEQ_c-posi 与 EQ_c-pIndex 方法基本相同,因此予以省略。在查询时滞 OPSM 时,其时滞步长 d 为 1。

首先测试随着列关键词数目变化时各种查询方法的性能。索引所用到的数据为 10000 行的数据集 D_6。具体的查询响应时间如图 5 所示。在列关键词个数由 3 增长到 7 的过程中,EQ_c-pfTree 查询方法的响应时间由约 830ms 逐步减少到约 800ms,EQ_c-pIndex 方法的查询响应时间由约 480ms 减少到约 45ms,GEQ_c-nega 方法的查询响应时间由约 210ms 减少到约 45ms,GEQ_c-posi_delay 方法的查询响应时间(在除去列数 3 的情况下,其为 1900ms)由约 270ms 减少到约 50ms,GEQ_c-nega_delay 方法的查询响应时间(在除去列数 3 的情况下,其为 1800ms)由约 210ms 减少到约 50ms。由于时滞 OPSM 的关键词少一个,其候选结果较多,因此会相对比较耗时。随着列关键词的增加,这种情况会越来越不明显。本实验证明所提方法不仅能保证多种 OPSM 的查询,而且也能保证每种方法的耗时都比较少。

最后在 6 种数据集上评估所提方法在同一查询下(4 个列关键词,KiWi[27] 方法除外)的响应时间。索引所用到的数据为 10000 行的数据集。具体的查询响应时间如图 6 所示。随着数据维度的增长,EQ_c-pfTree 查询方法的响应时间呈指数级飞速增长,而其他 4 种方法基本保持在一定的时间范围内。具体情况如下:EQ_c-pfTree 查询方法的响应时间由约 25ms 急速增长到约 820ms,EQ_c-pIndex 与 GEQ_c-nega 方法的查询响应时间基本相同(由约 10ms 增长到约 70ms。在数据集 D_6 上,前者响应时间略高,约为 260ms),GEQ_c-posi_delay 与 GEQ_c-nega_delay 方法的查询响应时间基本相同,且略高于 EQ_c-pIndex 与 GEQ_c-nega 方法的查询响应时间,其由约 20ms 逐步增长到 220ms。Kiwi 方法由于是一种从基因表达数据中批量挖掘 OPSM 的方法,没有列关键词选项,因此采用其默认设置。KiWi 在 D_2 数据集上的运行时间最长(109ms),在 D_3 和 D_1 数据集上的运行时间稍短(分别为 47ms 和 31ms),在 D_4,D_5 和 D_6 数据集上的运行时间最短(基本为 16ms)。本实验证明所提方法具有查询的有效性与良好的可扩展性。

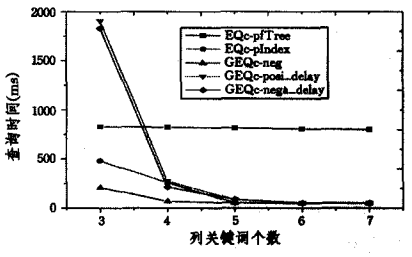


图 5 在数据集 D_6 上,列关键词变化时查询算法在单机上的运行时间

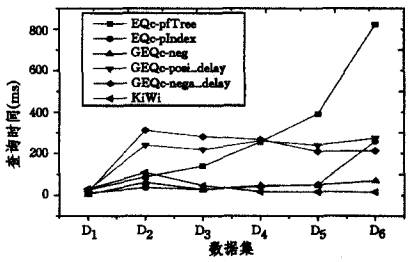


图 6 在 6 个数据集下,查询 4 个列关键词时算法在单机上的运行时间

结束语 本文给出一种支持直接从基因表达数据中查询多类型 OPSM 的方法。首先给出一种带有列表头的基于前缀树的索引 gIndex,接着设计了支持正相关、负相关以及时滞等类型 OPSM 查询的算法 GEQ_c。其避免了通过批量挖掘的方式来搜索 OPSM 所消耗的代价,大大提高了基因表达数据的索引以及多类型 OPSM 的查询效率。

尽管本研究给出了多种类型的 OPSM 的查询工作,但是还有诸如 shifting、scaling、shifting-scaling 等多种类型的 OPSM 的查询有待研究。本质上,其也是可以转化为本文的研究工作,但也有自身的特点,即使使用现有的方法可以解决,在性能上还是有待提高的。在未来的研究中,试图寻找新的索引与查询方法来改善多类型 OPSM 的搜索效率。

参 考 文 献

[1] Wang Zhen-jia, Li Guo-jun, Robinson Robert W, et al. UniBic: Sequential Row-based Biclustering Algorithm for Analysis of Gene Expression Data[J]. Scientific Reports, 2016, 6

[2] Hartigan J A. Direct Clustering of a Data Matrix[J]. Journal of the American Statistical Association, 1972, 67(337): 123-129

[3] Cheng Yi-zong, Church George M. Biclustering of Expression Data [C]// Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology (ISMB). AAAI, 2000: 93-103

[4] Amir B D, Benny C, Richard K, et al. Discovering Local Structure in Gene Expression Data: the Order-Preserving Submatrix Problem [J]. Journal of Computational Biology, 2003, 10(3/4): 373-384

(下转第 223 页)

for small files in HDFS[C]// IEEE International Conference on Cluster Computing and Workshops, 2009 (CLUSTER'09). IEEE, 2009: 1-4

[20] Chambers D W. Key performance indicators[J]. Journal of the American Dental Association (JADA), 2013, 144(3): 242-244

[21] Simundic A. Quality indicators[J]. Biochemia Medica, 2008, 18 (3): 311-319

[22] Page L, Brin S, Motwani R, et al. The PageRank Citation Ranking: Bringing Order to the Web[J]. Stanford InfoLab, 1998, 9 (1): 1-14

[23] Kang F, Jin R, Sukthankar R. Correlated Label Propagation with Application to Multi-label Learning[C]// IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2006: 1719-1726

[24] Rendle S. Factorization Machines[C]// 2010 IEEE 10th International Conference on Data Mining (ICDM). 2010: 995-1000

[25] Fan R, Chang K, Hsieh C, et al. LIBLINEAR: A Library for Large Linear Classification[J]. Journal of Machine Learning Research, 2008, 9(12): 1871-1874

[26] Mh Z. Receiver-operating characteristic (ROC) plots: a fundamental evaluation tool in clinical medicine[J]. Clinical Chemistry, 1993, 39(4): 561-577

(上接第 196 页)

[5] Wang Hai-xun, Wang Wei, Yang Jiong, et al. Clustering by Pattern Similarity in Large Data Sets [C]// Proceedings of the 28th ACM SIGMOD International Conference on Management of Data. 2002: 394-405

[6] Pensa Ruggero G, Boulicaut Jean-Francois. Constrained Co-clustering of Gene Expression Data[C]// Proceedings of the 8th SIAM International Conference on Data Mining (SDM). 2008: 25-36

[7] Faris A, Bader Joel S, Rajul A, et al. Query-based Biclustering using Formal Concept Analysis [C]// Proceedings of the 12th SIAM International Conference on Data Mining (SDM). 2012: 648-659

[8] Jiang Tao, Li Zhan-huai, Chen Qun, et al. Towards OrderPreserving SubMatrix Search and Indexing [M]// Database Systems for Advanced Applications: Proceedings of the 20th International Conference on Database Systems for Advanced Applications (DASFAA) Part II, 2015: 309-326

[9] Jiang Tao, Li Zhan-huai, Shang Xue-qun, et al. Constrained Query of Order-Preserving SubMatrix in Gene Expression Data [J]. Frontiers of Computer Science, 2016, 10(5): 1-5

[10] Wassim A, Mourad E, Hao Jin-kao. BicFinder: a Biclustering Algorithm for Microarray Data Analysis [J]. Knowledge and Information Systems, 2012, 30(2): 341-358

[11] Yang Jiong, Wang Wei, Wang Hai-xun, et al. δ-Clusters: Capturing Subspace Correlation in a Large Data Set [C]// Proceedings of the 18th International Conference on Data Engineering (ICDE). IEEE press, 2002: 517-528

[12] Cho H, Dhillon Inderjit S, Guan Yu-qiang, et al. Minimum Sum-Squared Residue Co-clustering of Gene Expression Data [C]// Proceedings of the 4th SIAM International Conference on Data Mining (SDM). SIAM Press, 2004: 114-125

[13] Chen Shu-hua, Liu Juan, Zeng Tao. MMSE: A Generalized Coherence Measure for Identifying Linear Patterns[C]// Proceedings of IEEE International Conference on Bioinformatics and Biomedicine (BIBM). IEEE press, 2014: 489-492

[14] Matteo D, Alessandro F, Manuele B. Biclustering Gene Expressions using Factor Graphs and the Max-sum Algorithm[C]// Proceedings of the 24th International Conference on Artificial Intelligence (IJCAI). AAAI Press, 2015: 925-931

[15] Zhao Yu-hai, Wang Guo-ren, Yin Ying, et al. A Novel Approach to Revealing Positive and Negative Co-Regulated Genes [J]. Journal of Computer Science and Technology, 2007, 22(2): 261-272

[16] Chen Jiun-rung, Chang Ye-in. An of Up-Down Bit Pattern Approach to Coregulated and Negative-Coregulated Gene Clutering of Microarray Data[J]. Journal of Computational Biology, 2011, 18(12): 1777-1791

[17] Zhao Yu-hai, Yu Xu, Wang Guo-ren, et al. Maximal Subspace Coregulated Gene Clustering [J]. IEEE Transactions on Knowledge and Data Engineering, 2008, 20(1): 83-98

[18] Wang Guo-ren, Yin Lin-jun, Zhao Yu-hai, et al. Efficiently Mining Time-Delayed Gene Expression Patterns [J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B, 2010, 40(2): 400-411

[19] Wang Guo-ren, Zhao Yu-hai, Zhao Xiang-guo, et al. Efficiently Mining Local Conserved Clusters from Gene Expression Data [J]. Neurocomputing, 2010, 73(7-9): 1425-1437

[20] Yin Ying, Zhao Yu-hai, Zhang Bin, et al. Mining Synchronous and Asynchronous Co-Regulated Gene Clusters from Time Series Microarray Data [J]. Chinese Journal of Computers, 2007, 30(8): 1302-1314(in Chinese)
印莹, 赵宇海, 张斌, 等. 时序微阵列数据中的同步和异步共调控基因聚类[J]. 计算机学报, 2007, 30(8): 1302-1314

[21] Amichai P, Saharon R. Optimal Set Cover Formulation for Exclusive Row Biclustering of Gene Expression [J]. Journal of Computer Science and Technology, 2014, 29(3): 423-435

[22] Rui H, Maderia Sara C. BicSPAM: Flexible Biclustering using Sequential Patterns [J]. BMC Bioinformatics, 2014, 15(1): 1-20

[23] Trapp Andrew C, Li Chao, Patrick F. Recovering All Generalized Order-Preserving SubMatrices: New Exact Formulations and Algorithms [J/OL]. Annals of Operations Research, 2016. http://link. springer. com/article/10. 1007% 2Fs10479-016-2173-9

[24] Jiang Tao, Li Zhan-huai, Chen Qun, et al. Parallel Partitioning and Mining Gene Expression Data with Butterfly Network [M]// Database and Expert Systems Applications: Proceedings of the 24th International Conference on Database and Expert Systems Applications (DEXA), Part I. 2013: 129-144

[25] Jiang Tao, Li Zhan-huai, Chen Qun, et al. OMEGA: An Order-Preserving SubMatrix Mining, Indexing and Search Tool [M]// Machine Learning and Knowledge Discovery in Database: Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/ PKDD), Part III. 2015: 303-307

[26] Broad Institute. Datasets. rar and 5q_gct[DB/OL]. http://www. broadinstitute. org/cgi-bin/cancer/datasets. cgi

[27] Gao B J, Griffith O L, Ester M, et al. On the Deep Order-preserving Submatrix Problem: a Best Effort Approach [J]. IEEE Transactions on Knowledge and Data Engineering, 2012, 24(2): 309-325