高技术通讯 2009 年 第 19 卷 第 2 期:188 ~ 193

doi: 10.3772/j.issn.1002-0470.2009.02.014

一种高效挖掘生物网络闭合频繁子图的算法①

彭佳扬② 杨路明 王建新 刘 振 李 敏

(中南大学信息科学与工程学院 长沙 410083)

摘 要 针对生物网络中频繁子图的挖掘问题,提出了一种基于 FP-树结构的 MaxFP 算法。此算法以代谢路径作为研究对象,在适合于生物网络图简化模型的基础上,采用一种不产生候选集的改进 FP-growth 算法挖掘生物网络中的闭合频繁子图。此算法考虑了基于频繁项目集的算法应用于网络的缺陷,根据生物网络的特点对 FP-growth 算法进行了改进。实验证明,提出的 MaxFP 算法比基于 Apriori 的频繁模式挖掘算法运行速度快,不仅能挖掘出最大的频繁子图,且能找到更多具有生物意义的频繁子图。

关键词 生物网络, 图挖掘, 闭合频繁子图, FP-树, FP-growth 算法

0 引 言

随着生物实验水平和技术的提高,大量的生物序列数据被提取出来。计算机学者开发了 BLAST 和 CLUSTAL W 等程序包来发现序列中频繁出现的模式,为序列和生物过程的研究做出了很大的贡献[1,2],对了解功能、结构和进化信息具有很重要的意义。分子生物学从序列发展到生物网络,产生了新一代的体现生物分子间关系与相互作用的实验数据[3,4],这些数据被抽象成图,例如蛋白质相互作用网络、基因调控网络、信号传递路径和代谢路径。挖掘这些生物网络中的频繁模式对理解基因组的发展具有很重要的作用,已经逐渐成为研究的焦点。而在序列中发现高频模式的算法和工具不能满足于挖掘新一代的实验数据,为了挖掘图中的频繁模式,需要研究更适合子图挖掘的算法和工具。

经典的生物网络频繁模式挖掘算法有:2000 年 Cook 和 Holder 提出的通过重复枚举递归调用来解决问题的 Subdue 算法[5],这种贪婪算法很耗时;2001 年 Inokuchi 等人提出的采用邻接矩阵模型和基于点扩展进行图挖掘的 AGM 算法[6],它挖掘大的稀疏图的代价很昂贵;2001 年 Kuramochi 和 Karypis 提出的采用稀疏邻接列表数据结构和基于边扩展的方式进行图挖掘的 FSG 算法[7],该算法适合于稀疏图的挖掘。AGM 算法和 FSG 算法都是基于 Apriori 算法[8],采用候选-检查技术来挖掘频繁子图,在挖掘时需要重复地扫描数据库。由于生物网络图中的节点不可避免地存在相同的标签,因此在生成候选图时通常会产生同构候选图,需要花大量时间排除同构子图。后来的算法通过更有效的模型来降低候选项的规模,通过更有效的优化技术来减少对空间的搜索。较为典型的算法有:2002 年 Yan 和 Han 提出的 gSpan 算法[9],该算法采用 DFS(深度优先搜索)词典排序法和最小 DFS 代码这两种技术避免同构子图;2004 年 Koyutürk 等人提出的 MULE 算法[10],该算法将生物网络中的顶点进行合并,对顶点进行唯一标识,采用基于反馈的深度优先枚举算法;2005 年 Hu 和 Yan 提出的 Codense 算法[11],该算法把频繁子图挖掘问题转换为在二次图中挖掘全连通子图问题。

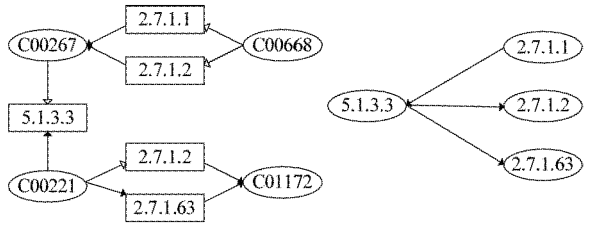
实际上,针对生物网络的特性挖掘闭合频繁模式就能满足研究的需要[12],不需要挖掘所有的频繁模式。本文利用适合生物网络的图简化模型,对闭合频繁子图的挖掘提出了新的算法——MaxFP。该算法将 FP-树结构应用于图论中,对 FP-growth 算法[13]进行改进来挖掘生物网络中的闭合频繁子图。实验结果证明基于 FP-树结构的挖掘闭合频繁子图的 MaxFP 算法要比基于 Apriori 的挖掘最大频繁子图的 MULE 算法效率高很多,并且能找到更多的具有生物意义的频繁模式。

1 模型及相关定义

本文以代谢路径作为研究对象。代谢路径表示

① 国家自然科学基金(60433020)和新世纪优秀人才支持计划(No. NCET-05-0683)资助项目。
② 女,1980 年生,博士生;研究方向:生物计算、数据挖掘、参数计算;联系人,E-mail: pjycsu@mail.csu.edu.cn
(收稿日期:2007-11-30)

执行某种特殊代谢功能的化学作用过程,由代谢物-代谢产物关系的化合物相互链接、相互作用而成。代谢路径用有向图来建模,图中的节点代表化合物,每条边对应一种反应或者一个酶[14],一条边的方向表示化合物是代谢物还是代谢反应后的产物。如图 1(a)所示,方框代表酶,椭圆代表代谢物,代谢物与代谢产物之间由酶作用,用边的方向表示它们之间的作用关系。



(a) 酵解的部分代谢路径示例　　(b) 简化后的有向图模型

图 1　代谢路径的有向图模型

挖掘代谢路径的目标是发现频繁出现的酶相互作用模式,研究者只对酶之间的关系感兴趣,不考虑代谢物,则可以将模型转换成一种更简单的图。在这种模型中,图中的节点代表酶,从一个酶到另一个酶之间的有向边代表前一个酶的产物是后一个酶的代谢物,如图 1(b)所示。在代谢路径中一个酶在同一路径中可以出现多次,表示这个酶在整个代谢过程中的不同时间点都参与作用。这样的现象在图模型中表现为图中不同节点的标签(酶)相同。研究者对代谢路径进行频繁子图的挖掘,对结构瞬时关系不感兴趣,只关心酶之间的相互作用关系,因此可以将图中相同标签的两个节点合并到一起,对模型进行简化。如图 1(a)中的酶 2.7.1.2 参与了两次作用,在简化图 1(b)中就合并成了一个节点。值得一提的是,只要保留好合并点的属性,就可以从合并了点的图简化模型中恢复瞬时模型。

在生物网络的应用中用唯一点标签模型不会产生子图同构,从而大大简化了图挖掘问题[10]。节点标签的唯一性也表示了边的唯一标识。把边作为基础数据集来解决频繁子图挖掘的问题,不考虑多数据类型也可以降低问题的难度。

接下来给出代谢路径、模型及问题的定义。

定义 1　代谢路径 $P(M,Z,R)$,代谢物 M,酶 Z,反应 R。每个反应 $r\in R$,与反应相关的酶 $Z(r)\subseteq Z$,反应物 $S(r)\subseteq M$,产物 $T(r)\subseteq M$。

定义 2　给定代谢路径 $P(M,Z,R)$,P 相对应的有向图 $G(V,E)$。每个酶 $zi\in Z$ 都对应一个节点 $vi\in V$,从 vi 到 vj 有一条边,即 $(vi,vj)\in E$,当且仅当存在 $r1,r2\in R$ 时,$zi\in Z(r1)$,$zj\in Z(r2)$,且 $T(r1)\cap S(r2)\neq\Phi$。

定义 3　给定一组有向图 $G1,G2,\cdots,Gn$ 和一个频繁度阈值 ε,若某个子图出现在 n' 个图中,$n'\leqslant n$,n' 为这个子图的频繁度 sup,$n'\geqslant\varepsilon$ 时就说它是频繁的。

定义 4　如果某个频繁子图 X 没有被其他相同频繁度的频繁子图 Y 所包含,即当 $sup(X)=sup(Y)$ 时,不存在 $X\subseteq Y$,我们就说这个频繁子图 X 是闭合的[15]。

闭合频繁子图包含了最大频繁子图中的所有子图。挖掘生物网络中的频繁子图仅考虑子图的最大性是为了避免冗余[10],闭合性也可以达到这个目的,并且符合生物网络挖掘的特性。

在代谢路径中挖掘闭合频繁子图的问题可以描述为:在一组有向图中挖掘频繁度大于频繁阈值的所有连通的闭合频繁子图。

2　算　法

2.1　相关工作

MULE 算法采用基于反馈的深度优先枚举算法,把频繁项目集挖掘的基本思路加上约束条件应用到频繁子图挖掘中,通过添加与已得的子图相连接的边来维持连通性;通过保存已经访问过的边来避免冗余。虽然 MULE 算法利用连通性改进的 Apriori 算法在原候选集的基础上进一步压缩了候选项集的大小,但仍然无法解决 Apriori 算法本身存在的性能瓶颈问题:可能产生庞大的候选集,计算空间开销很大;需要重复地扫描事务数据库,通过模式匹配检查很大的候选集合,需要很大的 I/O 负载,计算时间开销很大;运算空间时间的需求都很大,无法满足大规模数据的运算。将 FP-growth 算法与 Apriori 算法做比较后不难发现 FP-growth 算法很好地解决了前者的性能瓶颈问题。FP-growth 算法不使用候选集,减少了计算空间开销;只进行两次数据库扫描,极大地减轻了 I/O 负载,减少了计算时间开销。

尽管 FP-growth 算法有其优点,但要将其运用到挖掘闭合频繁子图问题仍有一些问题需要解决:FP-growth 算法挖掘出的频繁项目集包含所有的频繁项,而不是闭合的频繁项;FP-growth 算法要对单个路径中的所有组合进行频繁判断,对于挖掘长模

式会付出很大的时空代价;FP-growth 算法通过一棵 FP-树挖掘出频繁项目集,无法保证结果图的连通性。

2.2 MaxFP 算法

针对上述问题,本文提出了 MaxFP 算法,该算法利用 FP-树结构压缩图数据库,并对 FP-growth 算法进行了有效的改进。

首先从代谢路径数据库中提取一组代谢路径图,转换成简化图;接下来对这组图建 FP-树[16],将对图数据库频繁模式的挖掘问题转换成挖掘 FP-树问题;接着由长度为 1 的频繁模式(初始后缀模式)开始,构造它的条件模式基[16];继而构造它的(条件)FP-树,并递归地在该树上进行挖掘。模式增长通过后缀模式与由条件 FP-树产生的频繁模式连接实现。

FP-growth 算法在对 FP-树进行挖掘时,需要考虑单个路径产生频繁模式的所有组合。由于在生物网络中挖掘出的频繁子图必须具有生物意义,都是连通的,所以大都为较长的模式。本文的 MaxFP 算法对此进行了改进,引入了一个新的概念高变频组合 β:设所有组合形成的集合为 A,所有 β 形成的集合为 B,$B \subset A$;对于某个组合 $a \in A$,且在集合 A 中不存在一种组合 b,使得 $a \subseteq b$ 且 a 和 b 的频繁度相同,则 $a \in B$。例如:AS:19; SS:17; SA:17; AG:15 为一条路径 P,频繁度阈值为 15 的高变频组合为 {AS,SS,SA,AG:15}∪{AS,SS,SA:17}∪{AS:19},即所有频繁度大于频繁度阈值且发生了频繁度变化的闭合频繁模式集。这样对于较长的单条路径进行频繁判断也很高效且不会遗漏具有生物意义的组合。

MaxFP 算法描述如下。

算法:MaxFP。利用 FP-树结构和模式增长技术挖掘频繁子图。

输入:一组代谢路径图 D;最小频繁度阈值 ε。

输出:频繁子图(边集)。

(1) 构造 FP-树。

1) 扫描图数据库 D 一次。收集频繁边的集合 F 和它们的频繁度 *support*。对 F 按频繁度降序排序,结果为频繁边表 L。

2) 创建 FP-树的根结点,以"null"标记它。对于 D 中每个图 G,执行:

选择 G 中的频繁边,并按 L 中次序排列。设排序后的频繁边表为 $[p \mid P]$,其中 p 是第一个元素,而 P 是剩余元素的表。调用 **insert _ tree** ($[p \mid P]$, T)。该过程执行情况如下:如果 T 有子女 N 使得 $N.label = p.label$,则 N 的频繁度计数 n 增加 1;否则创建一个新节点 N,将其 n 设置为 1,链接到它的父节点 T,并且通过节点链结构将其链接到具有相同 *label* 的节点。如果 P 非空,递归地调用 **insert _ tree** (P, N)。

(2) FP-树的挖掘通过调用 **MaxFP** (***TP _ tree***, ***null***)实现。

procedure **MaxFP** (***Tree***, α):

1) **if** *Tree* 含单个路径 P **then**

2) **for** 路径 P 中节点的每个高变频组合 β

3) 产生模式 $\beta \cup \alpha$,其 *support* = β 中节点的最小频繁度;

4) **else for each** αi 在 *Tree* 的头部{

5) 产生一个模式 $\beta = \alpha i \cup \alpha$,其 *support* = αi.*support*;

6) 构造 β 的条件模式基,然后构造 β 的条件 FP-树 $Tree_\beta$;

7) **if** $Tree_\beta \neq \varnothing$ **then**

8) 调用 **MaxFP** ($Tree_\beta$, β);}

2.3 后处理

为了保证 MaxFP 算法挖掘出来的频繁边集是连通的子图,并且结果更具有生物意义,需要对算法输出的结果进行进一步的处理,主要工作包括:将挖掘算法产生的结果集(频繁子图集)转换成连通频繁子图集;删除连通频繁子图集中一些无意义或重复的结果项;插入一些有意义的中间结果项到最终结果集中。

通过以下的方法将频繁子图集转换成连通频繁子图集。

ExchangeFreitem ()。将频繁子图集转换成连通频繁子图集。

(1) **for each** 频繁子图 γi,$\gamma i = \{e1, e2, e3, \cdots, en\}$,$\gamma i \in$ 频繁子图集 A {

(2) 对 γi 调用函数 **Exchange** (γi) }

Exchange ()。将频繁子图 γi 转换成连通频繁子图。

(1) $B = \{ei\}$,$C = \gamma i - B$,$ei \in \gamma i$;

(2) **for each** ej,$ej \in C$ {

(3) **if** 存在 ej 与 ei 连通 **then**

(4) $B = B + \{ej\}$,$C = \gamma i - B$;

(5) **else** 输出 B}

(6) **if** C! = ∅ **then Exchange** (C);

将频繁子图集转换成连通频繁子图集的过程中,可能会产生一些重复的连通边集,例如:设频繁

子图 A 和 B,若 A 的转换结果中包含连通频繁子图 C,B 的转换结果中包含连通频繁子图 D,且 $D \subseteq C$,则结果项 D 是无意义的或重复的,需要将其删除。

模式增长的技术使得一些有意义的 1-频繁边可能不在结果集中[16]。由于本文的 MaxFP 算法采用模式增长的方法挖掘频繁边,因此需要将这些 1-频繁边插入到结果集中。例如某个 1-频繁边 e 的频繁度为 n,而包含边 e 的子图频繁度均小于 n,则此时的 1-频繁边 e 是有意义的,需要将连通边集 $\{e\}$ 插入到结果集中。

本文提出的 MaxFP 算法对 FP-growth 算法进行改进之后,不但具有 FP-growth 算法的优点,并且适合挖掘生物网络频繁模式的需求。本研究通过用真实的生物网络数据进行的实验证明了该算法的优势。

3 实验与分析

本文从 KEGG 代谢路径数据库(http://www.kegg.com/)[14]中提取了不同生物的代谢路径对 MaxFP 算法进行测试。KEGG 是现在比较流行的公开的全面的代谢路径的数据库,到 2007 年为止,KEGG 包含了 500 多种生物的碳水化合物(carbohydrate)、能量(energy)、油脂(lipid)、核苷酸(nucleotide)和氨基酸(amino acid)等的代谢路径图。本文选取了文献[10]中测试 MULE 算法的两种代谢路径,丙胺酸-天门冬氨酸盐(alanine-aspartate)和谷氨酸盐(glutamate),分别选用不同规模的代谢路径进行了多次测试,并且将实验结果与 MULE 算法进行了比较。实验结果如表 1 所示。

从实验结果比较表中可以看出 MaxFP 算法运算时间很短,比 MULE 算法高效。不使用候选集使得 MaxFP 算法的运算时间低于 MULE 算法;利用 FP-树压缩数据使得 MaxFP 算法在整个运算过程中只对数据库进行两次扫描,大大减少了 I/O 开销,因此在运算时间开销上明显高于 MULE 算法。从表中不难发现,MaxFP 算法的时间开销随实验集规模的增长或频繁阈值的降低而缓慢增长,与其他参数无关;而 MULE 算法的时间开销不仅与这两个参数相关,还与最大结果图规模(结果图中拥有最多边的图的边数)紧密相关,这是因为结果图的规模越大,MULE 算法扫描数据库的次数就越多,I/O 开销就越大。因此对于规模越大的数据库,MaxFP 算法比 MULE 算法越高效。

表 1 选用不同的频繁阈值挖掘不同规模的代谢路径结果比较

代谢路径	实验集规模(点,边)	频繁度阈值	最大结果图规模	1-频繁项数	频繁子图数		时间(s)	
					MULE 算法	MaxFP 算法	MULE 算法	MaxFP 算法
丙胺酸-天门冬氨酸盐(alanine-aspartate)	40 (692,2123)	6	16	27	14	55	5.818	1.125
		13	4	10	6	10	1.405	0.328
		15	3	6	3	5	1.063	0.297
	55 (952,2978)	8	16	32	22	87	7.624	2.197
		17	7	12	7	13	3.136	0.453
		20	3	6	3	6	1.36	0.411
	70 (1227,3845)	10	17	35	15	100	11.214	3.901
		24	7	11	6	10	3.897	0.578
		30	3	6	4	5	1.689	0.52
谷氨酸盐(glutamate)	40 (669,2657)	10	6	12	3	10	2.064	0.328
		12	3	7	3	6	1.063	0.302
	55 (948,3891)	16	6	9	2	9	2.691	0.437
		18	3	5	3	5	1.359	0.422
	70 (1033,4155)	18	3	11	5	10	1.5	0.489
		20	3	5	3	5	1.489	0.469

从表中还可以看出,MULE 算法的输出结果比 MaxFP 算法的输出结果少。因为 MULE 算法只求出了最大频繁子图集,而 MaxFP 算法不仅求出了最大频繁子图集,同时求出了同样有意义的闭合频繁子图集。以 alanine-aspartate 规模为 40 的实验集的结果为例,当频繁度阈值为 13 时,MULE 算法得到的结果为(SS, SG, GH, GS),(AS, SA, SS),(AB, AG),(GH, AG),(DG),(AB2)6 个频繁子图(见图

2)；MaxFP 算法得到的结果为(13：SS，SG，GS，GH)，(13：DG)，(13：AB2)，(14：AB，AG)，(15：AS，SA，SS)，(15：GH，AG)，(16：GH)，(17：AB)，(17：AG)，(19：SS)(见图 3)；当频繁度阈值为 15 时，MULE 算法得到的结果为(AS，SA，SS)，(GH，AG)，(AB)(见图 4)；MaxFP 算法得到的结果为(15：AS，SA，SS)，(15：GH，AG)，(16：GH)，(17：AB)，(17：AG)，(19：SS)(见图 5，其中点 aspS，argG，purA，……表示酶；SS，SG，GA，……表示两点之间的连线)。从图 2、图 3、图 4、图 5 中不难看出，MULE 算法只能找到符合给定阈值的最大频繁子图，有一些频繁度高

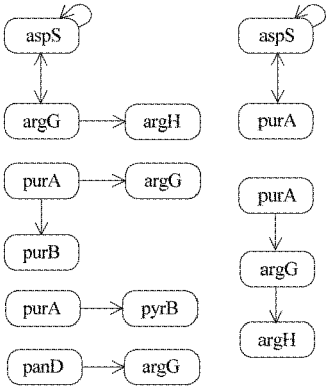


图 2 MULE 算法的结果集(阈值 13)

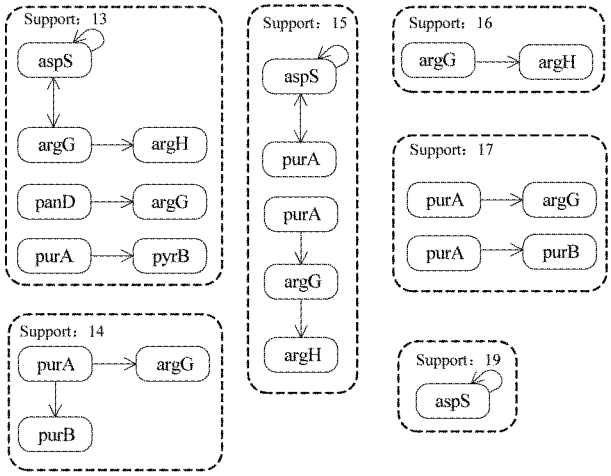


图 3 MaxFP 算法的结果集(阈值 13)

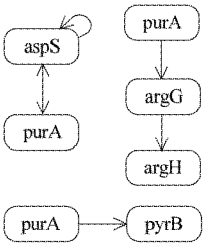


图 4 MULE 算法算法的结果集(阈值 15)

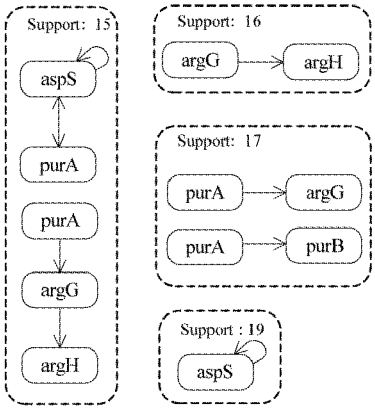


图 5 MaxFP 算法的结果集(阈值 15)

于最终输出的频繁子图，且属于其子图的频繁子图就被屏蔽掉了，得到这些子图在生物意义上是有必要的；MaxFP 算法不仅得到给定阈值的最大频繁子图，而且求出了更多的大于给定阈值的频繁子图，并且标明了输出的闭合频繁子图的阈值大小。MaxFP 算法得到的结果是 MULE 算法结果的超集。

比较图 3 和图 5 不难发现，图 5 的结果在图 3 中都存在，MaxFP 算法在阈值为 15 时得到的结果就是阈值为 13 时得到的结果中所有频繁度大于等于 15 的频繁子图；而比较图 2 和图 4 可以发现 MULE 算法的结果不存在这种特点。对于实际的大规模生物网络数据的挖掘，一次挖掘可能需要较长的时间，如果一个运算结果只对应于一个阈值，一旦阈值选择不合理，可能需要多次运算才能得到需要的结果。MaxFP 算法的一次运算结果对应于多个阈值，有效地减少了阈值选择和运算的次数。

实验证明，本文基于 FP-growth 的 MaxFP 算法比基于 Apriori 的 MULE 算法快；不仅能挖掘出最大连通频繁子图，且能找到更多的具有生物意义的闭合频繁子图，一次运算可以挖掘出基于 Apriori 的频繁模式挖掘算法多次运算的结果。

4 结 论

本文利用适合生物网络的图简化模型，对闭合连通频繁子图的挖掘提出了新的算法 MaxFP，该算法利用 FP-树结构压缩图数据库，有效地改进了 FP-growth 算法。MaxFP 算法比 MULE 算法更大地提高了算法的运行效率，实验数据分析表明 MaxFP 算法比 MULE 算法获得了更多的具有生物意义的实验结果。模型和算法还可以很好地被应用到其它的生物网络或者类似的复杂网络中。今后的工作将主要放

在如何从非唯一标号的图中挖掘闭合频繁模式。

参考文献

[1] Altschul S F, Madden T L, Scheffer A A, et al. Gapped BLAST and PSIBLAST: a new generation of protein database search programs. *Nucleic Acids Res*, 1997,25(17): 3389-3402

[2] Thompson J D, Higgins D G, Gibson T J. CLUSTALW: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res*, 1994,22(22), 4673-4680

[3] Hartwell L H, Hopfield J J, Leibler S, et al. From molecular to modular cell biology. *Nature*, 1999,402(6761): C47-C51

[4] Oltvai Z N, Barabási A L. Life's complexity pyramid. *Science*, 2002,298 (5594): 763-764

[5] Cook D J, Holder L B. Graph-based data mining. *IEEE Intell Syst*, 2000,15(2): 32-41

[6] Inokuchi A, Washio T, Okada T, et al. Applying the Apriori-based graph mining method to mutagenesis data analysis. *Comput Aided Chem*, 2001,2: 87-92

[7] Kuramochi M, Karypis G. Frequent subgraph discovery. In: Proceedings of the 2001 IEEE International Conference on Data Mining, California, USA, 2001. 313-320

[8] Agrawal R, Srikant R. Fast algorithms for mining association rules. In: Proceedings of the 20th International Conference on Very Large Data Bases (VLDB'94), Santiaogo di Chile, Chile, 1994. 487-499

[9] Yan X, Han J. gSpan: Graph-based substructure pattern mining. In: Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM'02), Maebashi City, Japan, 2002. 721-724

[10] Koyutürk M, Grama A, Szpankowski W. An efficient algorithm for detecting frequent subgraphs in biological networks. *Bioinformatics*, 2004,20(1): i200-i207

[11] Hu H Y, Yan X F, Huang Y, et al. Mining coherent dense subgraphs across massive biological networks for functional discovery. *Bioinformatics*, 2005,21(1): i213-i221

[12] Olken F. Biopathways and protein interaction databases. In: A lecture in Bioinformatics Tools for Comparative Genomics: A short course, Berkeley, CA, 2003

[13] Han J W, Pei J, Yin Y W. Mining frequent patterns without candidate generation. In: Proceedings of the 2000 ACMSIGMOD International Conference on Management of Data, Dallas, TX, USA, 2000. New York: ACM Press, 2000. 1-12

[14] Krishnamurthy L, Nadeau J, Özsoyoˇglu G, et al. Pathways database system: an integrated system for biological pathways. *Bioinformatics*, 2003,19(8): 930-937

[15] 胡孔法, 唐小丽, 达庆利等. 一种高效挖掘高维数据的频繁闭合模式算法. 东南大学学报(自然科学版). 2007,37(4):569-573

[16] Han J W, Kamber M. Data Mining Concepts and Techniques. 2nd Edition. Singapore: Elsevier (Singapore) Pte Ltd, 2007. 233-249

An efficient algorithm for detecting closed frequent subgraphs in biological networks

Peng Jiayang, Yang Luming, Wang Jianxin, Liu Zheng, Li Min

(School of Information Science and Engineering, Central South University, Changsha 410083)

Abstract

In this paper, the MaxFP, an algorithm for detecting frequent subgraphs in biological networks based on the FP-tree structure is proposed. The algorithm takes the metabolic pathway as the object of study, and uses an improved FP-growth algorithm without generating candidates for detecting closed frequent subgraphs in biological networks based on the simplification model appropriate to biological networks. Accordingly the MaxFP Considers the defects of the algorithms based on item-set mining which are applied to networks, and improves the FP-growth algorithm according to the features of biological networks. The experimental results show that the MaxFP runs faster than the algorithms based on Apriori, and the MaxFP not only detects maximal frequent subgraphs, but also finds more frequent subgraphs having biological meaning.

Key words: biological networks, graph mining, closed frequent subgraph, FP-tree, FP-growth algorithm