基于生物网络的频繁 Hamilton 子图挖掘算法

董安国[1,2],高 琳[2],邱在秦[3],赵建帮[2],周晓峰[2]
(1 长安大学 理学院,陕西 西安 710064;2 西安电子科技大学 计算机学院,陕西 西安 710071;
3 西安石油大学 理学院,陕西 西安 710065)

[摘 要] 【目的】在生物网络的功能模体发现问题中涉及到频繁子图的挖掘,而功能模体通常是一个非树型结构的子图,甚至具有 Hamilton 回路。为了减少挖掘出子图的结果集,提高频繁子图挖掘的效率,分析了在生物网络中挖掘频繁 Hamilton 子图的算法。【方法】对网络连接矩阵构造了一种运算,得到网络路径信息,通过对路径的合并,搜索出网络中所有的 Hamilton 子图。【结果】在理论分析和证明的基础上,给出了 2-路径和 3-路径的搜索算法,进而构造了 Hamilton 子图的搜索算法,并对算法的复杂度进行了分析,最后将算法应用于真实生物网络,找出了频繁 Hamilton 子图。【结论】与现有子图搜索算法相比,由于搜索的只是 Hamilton 子图,减少了搜索结果集,同时引入了代数运算并构造了矩阵的快速迭代算法,提高了挖掘效率,试验结果也验证了算法的高效性。

[关键词] 生物网络;Hamilton 回路;Hamilton 子图;频繁 Hamilton 子图
[中图分类号] TP311.12 [文献标识码] A [文章编号] 1671-9387(2008)05-0185-06

An algorithm for finding Hamilton subgraph in biological network

DONG An-guo[1,2],GAO Lin[2],QIU Zai-qin[3],ZHAO Jian-bang[2],ZHOU Xiao-feng[2]
(1 *School of Science, ChangAn University, Xi'an, Shaanxi* 710064, *China*;
2 *School of Computer Science and Technology, Xidian University, Xi'an, Shaanxi* 710071, *China*;
3 *School of Science, Xi'an Shiyou University, Xi'an, Shaanxi* 710065, *China*)

Abstract: 【Objective】 When referred to functional motif discovery in biological networks, the most important step is to find subgraphs with an enhanced number of internal links with feedback, such as nontree-like and Hamiltonian circle structure. To improve the efficiency of frequent subgraph mining, the algorithm for mining subgraph with the nature of Hamiltonian circle was provided. 【Method】 A matrix theory-based approach was developed for such subgraphs and the properties of Hamiltonian subgraphs waer studied. 【Result】 Then the algorithms were provided for searching the 2-path and 3-path based on the properties to find subgraph with Hamiltonian cycle. Also, the complexity of the algorithm were analyzed and the algorithms to real biological network were applied. 【Conclusion】 The simulation results show that our algorithms have high efficiency compared with the existing algorithm since the strategy was adopted to reduce the number of subgraph under the constraint of Hamiltonian circle.

Key words: biological network; Hamilton circle; Hamilton subgraph; frequenct Hamilton subgraph

近十几年来,基于图数据挖掘算法的研究已成为计算机科学研究的重要内容,在生物信息学、药物研制、社会网络、集成电路的布局布线、Web 数据挖掘、软件测试、网络工程等众多领域积累了大量的

* [收稿日期] 2007-12-28
[基金项目] 国家自然科学基金项目(60574039);长安大学科技发展基金项目(07J04)
[作者简介] 董安国(1964—),男,浙江象山人,副教授,硕士,主要从事计算机算法设计及生物信息学研究。
E-mail:donganguo@zwu.edu.cn

关于图的数据,这些数据中包含了大量的重要信息,为此需要对这些图进行分析[1-2]。例如在基因调控网络和蛋白质相互作用的网络中,存在某种特殊结构的子网络结构,其代表了一种具有特定生物功能的结构,在社会网络中需要查找具有一定功能的社团结构等问题。

频繁子图挖掘的研究工作始于 1994 年,Cook 等[3]提出了著名子图挖掘算法 SUBDUE,该算法采用最小描述长度原则压缩原始图,并利用启发式的搜索策略挖掘频繁子图,以牺牲结果集的完整性为代价,使挖掘效率得以提高,算法的主要目标是针对生物应用领域的特殊问题。后来 SUBDUE 还扩展成为图分类算法,称为 SubdueCL[4],在 SubdueCL 中不再采用最小描述长度,而是采用基于子图置信度的启发式方法。2005 年,Deshpande 等[5]同样提出了化学化合物分类中挖掘频繁子图的算法。Yoshida 等[6]提出了一个子图挖掘算法 GBI,该算法类似 SUBDUE 算法,但采用了不同的启发式搜索策略。Inokuchi 等[7]于 2001 年提出了一个基于 Apriori 思想的频繁子图模式挖掘算法 AGM,随后又提出了各种特殊类型的频繁子图挖掘算法[8-9]。此后,又有各种不同的频繁子图挖掘算法被提出来,比较有影响的有 Kuramochi 等[1]提出的 FSG 算法及 Wernicke[2]提出的 ESU 算法。在一般意义下,由于没有利用任何先验信息,频繁子图的搜索工作量很大,所以除了上述这些子图模式挖掘的通用算法外,研究人员还提出了大量运用于实际问题的、带有约束的子图模式挖掘算法[10-12]。

在生物网络中,具有树型子图的结构没有回馈、前馈环、聚集等功能,只能反应各个节点间不完整的关系,这种结构不具有生物功能,而树型子图发生频率相当高,即使高频率的树型子图也不会成为功能模块。所以,在针对生物网络的应用中,Berg 等[13]在 2004 年提出了非树型子图的挖掘算法,其目的是为了减少搜索结果集,提高挖掘效率,但由于非树型子图的范围还比较大,当网络规模很大时,搜索效率还不尽如人意;作为一类特殊的非树型结构,Hamilton 子图的结构包含更多的回馈功能,与非树型子图相比,由于搜索结果集更少,有利于提高搜索效率,所以挖掘频繁 Hamilton 子图在生物网络的功能结构分析中,具有重要的意义。

基于生物网络等具体的应用背景,本研究提出了一种频繁 Hamilton 子图搜索算法,即首先利用代数方法,通过构造连接矩阵的运算,得到网络中的路径信息,通过对路径的合并搜索出网络中所有的 Hamilton 子图;然后利用文献[14]的同构算法,统计出各子图频率,通过对频率的比较挖掘出频繁 Hamilton 子图;最后利用真实的生物数据对算法进行验证。现将研究结果报道如下,以期为频繁 Hamilton 子图挖掘算法在生物网络等领域的应用提供参考。

1 定义和符号

关于 Hamilton 子图搜索算法的描述中,涉及到一些名词和符号,为此,先给出一些定义并对符号加以说明。下文所指的子图是导出子图[1],路径均为简单路径,图的阶数是指该图的节点数。

1.1 定 义

定义 1 在无向图 G 中,如果存在一条回路 Γ,使得 Γ 穿程于 G 的每一结点 1 次且仅 1 次,则称 G 为 Hamilton 图,对应的回路 Γ 为 Hamilton 回路。

定义 2 G_1 是无向图 G 的 1 个子图,如果 G_1 是 Hamilton 图,则称 G_1 为 G 的 Hamilton 子图。

定义 3 1 条从节点 i 到 j 的路径所经历的边的个数称为该路径的长度,长度为 k 的路径称为 k-路径。

定义 4 设 $G(V,E)$ 为 1 个有向图,如果将 G 中的所有边的方向去掉,并去掉重边,得到 1 个无向图 G_1,如果 G_2 为 G_1 的 Hamilton 子图,且 G_2 的节点对应于有向图 G 的子图为 G_3,则称 G_3 为 G 的拟 Hamilton 子图。

1.2 符号和变量

$V=\{v_1,v_2,\cdots v_n\}$ 为图 $G(V,E)$ 的节点,本文令 $v_i=i(i=1,2,\cdots n)$;

$\boldsymbol{A}=(a_{ij})_{n\times n}$ 表示 G 的连接矩阵,即:

$$a_{ij}=\begin{cases}1 & \text{节点 } i \text{ 和 } j \text{ 有边相连}\\ 0 & \text{否则}\end{cases};$$

S_{ij} 表示 G 中同时与节点 i 和 j 相连的节点的全体(图 1),即:

$$S_{ij}=\{k\mid a_{ik}=a_{kj}=1(i,j,k\in V)\};$$

$\boldsymbol{M}^{(k)}=(m_{ij}^{(k)})_{n\times n}$ 为 k-路径连接矩阵,其中 $m_{ij}^{(k)}$ 表示从节点 i 到 j 的 k-路径条数;R^i 表示 G 中与节点 i 至少有 2 条 2-路径的节点的集合,即 $R^i=\{j\mid m_{ij}^{(2)}\geqslant 2\}$;$T_{ij}$ 表示从节点 i 到 j 的 3-路径中 j 的前一个节点的集合(如图 2 所示);Q^i 表示 G 中与节点 i 至少有 2 条 3-路径的节点的集合,即 $Q^i=\{j\mid m_{ij}^{(3)}\geqslant 2\}$。

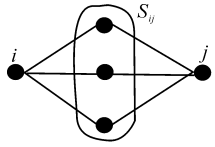


图 1　i 到 j 的 2-路径结构图

Fig. 1　2-path structure from i to j

2　k-路径连接矩阵及 k-路径的计算

2 条起点和终点相同的路径,如果内部没有相同的节点,那么它们所经过的节点就可以构成 1 个 Hamilton 子图。例如,1 个 6 阶的 Hamilton 子图可以通过 2 条起点和终点相同的 3-路径得到。所以要搜索出图 G 中所有固定大小的 Hamilton 子图,首先需确定 k-路径。以下主要就 $k=2$ 及 $k=3$ 的情形进行讨论。

定理 1　$\boldsymbol{A}$ 为图 G 的连接矩阵,设 $\boldsymbol{B}=(b_{ij})_{n\times n}=\boldsymbol{A}^2$,则节点 i 的度为 b_{ii},$|S_{ij}|=b_{ij}$。

证明:设 $k\in S_{ij}$,则由 S_{ij} 的定义得 $a_{ik}=a_{kj}=1$,由于 $\boldsymbol{A}$ 是布尔(0～1)矩阵,$\boldsymbol{B}=\boldsymbol{A}^2$。所以 $b_{ij}=\sum_{k=1}^{n}a_{ik}a_{kj}=\sum_{k\in S_{ij}}a_{ik}a_{kj}=|S_{ij}|$。$b_{ii}=\sum_{k=1}^{n}a_{ik}a_{ki}$ 为 $\boldsymbol{A}$ 的第 i 行 1 的个数,也就是节点 i 的度。

例 1:如图 3 所示的网络,其连接矩阵 $\boldsymbol{A}$ 及 $\boldsymbol{B}$ 如下:

$$\boldsymbol{A}=\begin{bmatrix}0&1&1&0&0&0&1&1\\1&0&1&0&0&0&0&0\\1&1&0&1&0&1&0&1\\0&0&1&0&1&0&0&0\\0&0&0&1&0&1&0&0\\0&0&1&0&1&0&1&0\\1&0&0&0&0&1&0&0\\1&0&1&0&0&0&0&0\end{bmatrix};$$

$$\boldsymbol{B}=\boldsymbol{A}^2=\begin{bmatrix}4&0&2&1&0&2&0&0\\0&2&0&1&0&1&1&2\\2&0&5&0&2&0&2&1\\1&1&0&2&0&2&0&1\\0&0&2&0&2&0&1&0\\2&1&0&2&0&3&0&1\\0&1&2&0&1&0&2&1\\0&2&1&1&0&1&1&2\end{bmatrix}。$$

由图 3 可见,$b_{46}=2$,所以从 4 到 6 长度为 2 的路径有 2 条:$a_{43}=a_{36}=1$,$a_{45}=a_{56}=1$。所以 $S_{46}=\{3,5\}$,$R^1=\{3,6\}$。由定理 1 可知,$\boldsymbol{M}^{(2)}=\boldsymbol{B}$,$R^i=\{j\,|\,b_{ij}\geqslant 2\}$,如果 $j\in R^i$,则 $1-p-j$ 构成 1 条从节点 1 到 j 的 2-路径,其中 $p\in S_{1j}$。

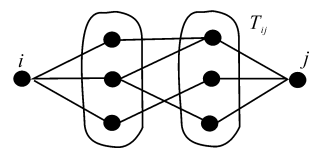


图 2　j 到 i 的 3-路径结构图

Fig. 2　3-path structure from i to j

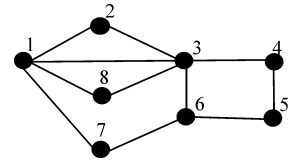


图 3　具有 8 个节点的网络

Fig. 3　A 8-nodes graph

定理 2　设 $\boldsymbol{B}=\boldsymbol{A}^2$,记 $d_i=b_{ii}$,并更新 b_{ii}($i=1,2,\cdots,n$)为 0,则 $\boldsymbol{M}^{(3)}=\boldsymbol{BA}-\boldsymbol{A}[diag(d_1,d_2,\cdots,d_n)-\boldsymbol{E}]$,其中 $diag(d_1,d_2,\cdots,d_n)$ 为以($d_1,d_2,\cdots,d_n$)为对角元素的对角矩阵,$\boldsymbol{E}$ 为 n 阶单位矩阵。

证明:由定理 1,d_i($i=1,2,\cdots,n$)为节点 i 的度;设 $\boldsymbol{W}=(w_{ij})_{n\times n}=\boldsymbol{BA}$,根据矩阵乘法的定义,$w_{ij}$ 表示从节点 i 到节点 j 长度为 3 的所有路径(包括有重复节点的路径)的条数。在图 4 中,w_{ij} 为 4,对应的路径为:$i-j-a-j$,$i-j-b-j$,$i-j-c-j$,$i-a-b-j$,但前 3 条是无效的路径。只要 i 和 j 有边,那么每 1 个与 j 有边相连的节点,均能生成 1 个无效的路径,共有 d_j-1 条,如果 j 和 i 没有边,则不会生成无效的路径。所以节点 i 到节点 j 的无效路径条数为 $a_{ij}(d_j-1)$,故 i 到 j 的 3-路径条数为 $w_{ij}-a_{ij}(d_j-1)$,即 $\boldsymbol{M}^{(3)}=\boldsymbol{BA}-\boldsymbol{A}[diag(d_1,d_2,\cdots,d_n)-\boldsymbol{E}]$。

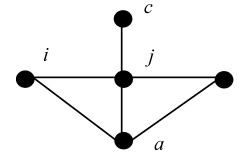


图 4　i 到 j 局部连接关系图

Fig. 4　Paths from i to j

下一个定理给出了 T_{1j} 的算法,通过它可以确定出所有从节点 1 到 j 的 3-路径。

定理 3　$T_{1j}=\{k\,|\,b_{1k}=1,a_{kj}=1,j\notin S_{1k}\}\cup\{k\,|\,b_{1k}>1,a_{kj}=1\}=\{k\,|\,b_{1k}\neq 0,a_{kj}\neq 0,S_{1k}\backslash(j)\neq\phi\}$。

证明:由 T_{1j} 的定义知,如果 $k\in T_{1j}$,则 k 和 j 一定相连,即 $a_{kj}=1$;k 和 1 一定存在 2-路径,且有两种可能的情形:

①1 和 k 只有 1 条 2-路径,且该路径不经过 j,否则,$1-j-k-j$ 就不是一个简单路径,与 $k\in T_{1j}$ 矛盾;所以 $k\in\{k|b_{1k}=1,a_{kj}=1,j\notin S_{1k}\}=\{k|b_{1k}=1,a_{kj}=1,S_{1k}\setminus\{j\}\neq\phi\}$。

②1 和 k 有多条 2-路径,这样 $k=\{k|b_{1k}>1,a_{kj}=1\}=\{k|b_{1k}>1,a_{kj}=1,S_{1k}\setminus\{j\}\neq\phi\}$。

从而有:$T_{1j}=\{k|b_{1k}=1,a_{kj}=1,j\notin S_{1k}\}\cup\{k|b_{1k}>1,a_{kj}=1\}=\{k|b_{1k}\neq0,a_{kj}\neq0,S_{1k}\setminus\{j\}\neq\phi\}$。

例 2:如图 5 所示,因为 $b_{16}=2,a_{63}=1$,所以 $6\in T_{13}$;$b_{12}=2,a_{23}=1$,所以 $2\in T_{13}$;$b_{15}=1,a_{53}=1$,但 $S_{15}=\{3\}$,即 $3\in S_{15}$,所以 $5\notin T_{13}$;从而 $T_{13}=\{6,2\}$。由定理 3 可以得到 T_{1j},如果 $j\in R^1$,则 $1-a-p-j$ 构成 1 条从节点 1 到 j 的 3-路径,其中 $a\in T_{1j}$,$p\in S_{1a}$。

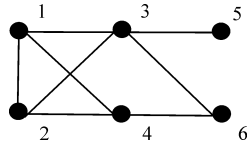


图 5 具有 6 个节点的网络

Fig. 5 A 6-nodes graph

3 搜索算法

根据定理 1~3 的结论设计了 Hamilton 子图的搜索算法,4 阶和 6 阶 Hamilton 子图的搜索描述如下。

3.1 4 阶 Hamilton 子图的搜索算法

$\boldsymbol{A}$ 是 G 的连接矩阵,n 表示 $\boldsymbol{A}$ 的阶数,具体算法如下:

Step 1:while n≥4 do

Step 2:$\boldsymbol{B}=\boldsymbol{A}^2$

Step 3:$R^1=\{j|b_{1j}\geqslant2,j\neq1\}$

Step 4:for all $j\in R^1$ do

Step 5:$S_{1j}=\{k|a_{1k}=a_{kj}=1\}$

Step 6:$subgraph=\{(1,a,b,j)|a,b\in S_{1j}\}$

Step 7:end for

Step 8:remove first row and column from A

Step 9:end while

3.2 6 阶 Hamilton 子图的搜索算法

6 阶 Hamilton 子图是通过 2 条具有相同起点和终点的 3-路径合成得到的,如图 6 所示。当 $|T_{1j}|=1$ 时,不能产生 6 阶 Hamilton 子图。如图 7 所示,当 $|T_{1j}|\geqslant2$ 时,经过 $a,b\in T_{1j}$ 的 2 条 3-路径可以构成 1 个 6 阶 Hamilton 子图。

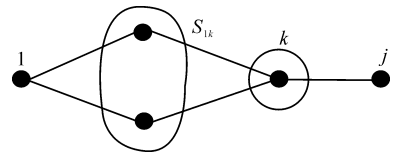


图 6 j 仅一个入口,即 $|T_{1j}|=1$

Fig. 6 Case $|T_{1j}|=1$

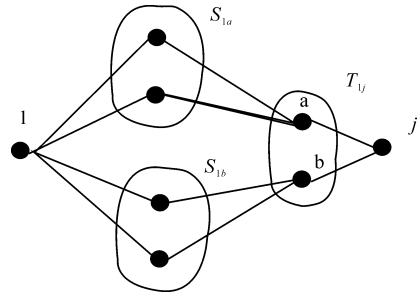


图 7 j 有多个入口,即 $|T_{1j}|>1$

Fig. 7 Case $|T_{1j}|>1$

$\boldsymbol{A}$ 是 G 的连接矩阵,n 表示 $\boldsymbol{A}$ 的阶数,$\boldsymbol{E}$ 为单位矩阵,$\boldsymbol{D}$ 是以 $\boldsymbol{B}$ 对角元素构成的对角矩阵。具体算法如下:

Step 1:while $n\geqslant6$ do

Step 2:$\boldsymbol{B}=\boldsymbol{A}^2$, $d_i\leftarrow b_{ii}$, $b_{ii}\leftarrow0$, $M^{(3)}=\boldsymbol{BA}-\boldsymbol{A}(D-E)$

Step 3:$Q^1=j|m_{1j}{}^{(3)}\geqslant2$

Step 4:for all $j\in Q^1$ do

Step 5:$T_{1j}=\{k|b_{1k}\neq0,a_{kj}\neq0,S_{1k}\setminus(j)\neq0\}$

Step 6:if $|T_{1k}|\geqslant2$ do

Step 7:for all $a,b\in T_{1k}$

Step 8:$S_{1a}=\{k|a_{1k}=a_{ka}=1\}$, $S_{1b}=\{k|a_{1k}=a_{kb}=1\}$

Step 9:for all $x\in S_{1a}$ and $y\in S_{1b}$

Step 10:if x,a,y,b are all different

Step 11:$subgraph=\{1,x,a,y,b,j\}$

Step 12:endfor

Step 13:remove the first line and column from A

Step 14:endwhile

在 3.1 和 3.2 中,分别描述了 4 阶和 6 阶 Hamilton 子图的搜索算法;在搜索 5 阶 Hamilton 子图时,可以通过上述描述的算法找出节点 1 到 j 的所有 2-路径(记为 L_2)和 3-路径(记为 L_3),从 L_2 和 L_3 中分别任取一条路径并将它们合并在一起(起点对起点,终点对终点),就构成一个回路,如果这个回路中的 5 个节点互不相同,则原图中对应的这 5 个

节点就导出 1 个 5 阶 Hamilton 子图,遍历完所有的节点 $j(j\neq 1)$,就找出了包含节点 1 的所有 5 阶 Hamilton 子图,去掉节点 1 以及与它相连的边,并从 1 开始对节点重新编号,重复上述过程就可以找出 G 中所有的 5 阶 Hamilton 子图。要搜索更大规模的子图,可归结为如何确定更长的路径,但在生物网络模体发现问题中,子图规模一般不超过 7 个节点[15]。

3.3 有向图的拟 Hamilton 子图的搜索

在有向图的模体发现问题中,往往需要搜索弱连通的子图,由于有向图的 Hamilton 子图结构非常特殊,所以对有向图只讨论拟 Hamilton 子图的搜索问题。

根据拟 Hamilton 子图的定义,只要将有向图 G 的连接矩阵 $\boldsymbol{M}$ 转化为对应的无向图 G_1 的连接矩阵 $\boldsymbol{A}$,就可以利用上文无向图的方法搜索到有向图 G 的所有拟 Hamilton 子图。有向图 G 的连接矩阵 $\boldsymbol{M}$ 和其对应的无向图的连接矩阵 $\boldsymbol{A}$ 之间有如下的关系:$\boldsymbol{A}=\boldsymbol{M}\vee\boldsymbol{M}^{\boldsymbol{T}}$,$\vee$ 表示对元素进行逻辑求或运算。

4 计算复杂度分析

定理 4　设 $\boldsymbol{A}$ 为一个 n 阶矩阵,记 $\boldsymbol{A}=\begin{bmatrix}a_{11} & \alpha^T\\ \beta & A_1\end{bmatrix}$,则 $\overline{\boldsymbol{A}}^2$ 表示 $\boldsymbol{A}^2$ 划去第 1 行第 1 列元素后所得的矩阵,有 $\boldsymbol{A}_1^2=\overline{\boldsymbol{A}}^2-\boldsymbol{\beta}\alpha^T$。

证明:因为 $\boldsymbol{A}^2=\begin{bmatrix}a_{11} & \alpha^T\\ \beta & A_1\end{bmatrix}\begin{bmatrix}a_{11} & \alpha^T\\ \beta & A_1\end{bmatrix}=\begin{bmatrix}a_{11}^2+\alpha^T\beta & a_{11}\alpha^T+\alpha^TA_1\\ a_{11}\beta+A_1\beta & \beta\alpha^T+A_1^2\end{bmatrix}$,

所以 $\boldsymbol{\beta\alpha^T}+\boldsymbol{A}_1^2=\overline{\boldsymbol{A}}^2$,从而 $\boldsymbol{A}_1^2=\overline{\boldsymbol{A}}^2-\boldsymbol{\beta\alpha^T}$。

利用定理 4 给出的算法,在整个搜索迭代过程中,除第 1 步外,每一步计算的 $\boldsymbol{A}^2$ 复杂度为 $O(n^2)$。

对所有的迭代步骤和不同的 j,假设 $|S_{1j}|$ 的最大值为 M_1,$|T_{1j}|$ 的最大值为 M_2。

4.1 4 节点情形

迭代算法第 k 步:

(1)计算 $\boldsymbol{A}^2$ 的第 1 行,$O(n^2)$;

(2)确定 $S_{1j}(j=2,3,\cdots,n)$,$O(n^2)$;

(3)搜索到所有 4 阶 Hamilton 子图,$O(M_1^2n)$,理论上 M_1 与 n 是有关的,但当边比较稀疏时,$M_1\ll n$,故搜索量约为 $O(n)$。在生物信息背景下,图的连边是比较稀疏的,所以完成全部过程的总计算量约为 $O(n^3)$。

4.2 6 节点情形

迭代算法第 k 步:

(1)计算 $\boldsymbol{B}=\boldsymbol{A}^2$,$O(n^2)$;

(2)确定 $S_{1j}(j=2,3,\cdots,n)$,$O(n^2)$;

(3)计算 $\boldsymbol{C}=\boldsymbol{BA}$ 的第 1 行,$O(n^2)$;

(4)确定 $T_{1j}(j=2,3,\cdots,n)$,$O(n^2)$;

(5)搜索到所有 6 阶 Hamilton 子图,$O(M_1^2M_2^2n)$,理论上 M_1、M_2 与 n 是有关的,但当边比较稀疏时,$M_1\ll n$ 且 $M_2\ll n$,这样搜索量约为 $O(n)$。在生物信息背景下,图的连边是比较稀疏的,所以完成全部过程的总计算量约为 $O(n^3)$。

搜索 6 阶以上的子图,将要确定更长的路径,如果不考虑设计矩阵乘法的快速算法,总运算量为 $O(n^4)$。

5 仿真试验

利用 Matlab7.1 对本研究的算法进行了仿真试验研究,试验使用的电脑是 Intel Celeron(R)2.26 GHz,内存为 256 M。试验数据来源于文献[16],分别是基因调控网络 *E. coli*(大肠杆菌)、Yeast(酵母)和 SeaUrchin(海胆)。

利用本研究介绍的算法,对这 3 个真实生物网络进行仿真试验,由于 3 个网络都是有向图,试验搜索出来的子图均是拟 Hamilton 子图,统计出的数量及运算时间如表 1 所示。

表 1　拟 Hamilton 子图数量及子图运算时间统计

Table 1　Numbers of quasi Hamilton subgraph and running time

网络 Network	边数 Edge	节点数 Node	4 阶子图 Size-4		6 阶子图 Size-6	
			子图数量 Number	运行时间/s Time	子图数量 Number	运行时间/s Time
大肠杆菌 *E. coli*	519	423	202	2.5	1 467	18.7
酵母 Yeast	1 079	688	1 633	10.1	4 320	97.5
海胆 SeaUrchin	81	45	73	0	746	0.4

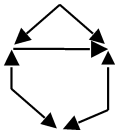
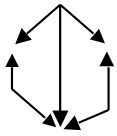
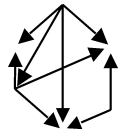
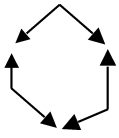
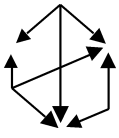
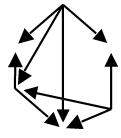
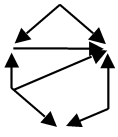
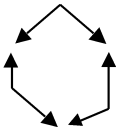
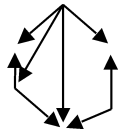
由表 1 可以看出,对于 3 个真实网络,由于其边比较稀疏(Edge/Node <2),所以拟 Hamilton 子图的数量很少(*E. coli* 6 阶子图的总数是 22 532 584 个[10]),从而搜索效率高,同时也减少了子图同构分类的时间。

对搜索到的结果集,利用文献[10]的同构算法

进行分类,统计出各网络的频繁拟 Hamilton 子图如 表 2 所示。

表 2 频率最大的 3 个 6 阶拟 Hamilton 频繁子图

Table 2 6-subgraph quasi Hamilton with maximum frequency

大肠杆菌 *E. coli*		酵母 Yeast		海胆 SeaUrchin	
子图 Subgraph	频率 Frequency	子图 Subgraph	频率 Frequency	子图 Subgraph	频率 Frequency
	0.290 0		0.303 7		0.032 2
	0.218 8		0.251 2		0.025 5
	0.127 4		0.192 6		0.020 1

6 总结与展望

在生物网络模体发现问题中,总要涉及到在 1 个大图中搜索小子图的问题,考虑到该问题的复杂性以及 Hamilton 子图结构的重要性,本研究提出了一种搜索 Hamilton 子图的算法,分析了算法的复杂度,并利用真实生物数据进行了仿真试验。目前,生物网络的模体规模基本上不超过 7 个节点[15],考虑到这一具体的应用背景,本研究只给出了节点数小于 7 的子图搜索算法。对于节点数更多、类型更加一般的子图搜索问题,有待于进一步研究。

[参考文献]

[1] Kuramochi M, Karypis G. An efficient algorithm for discovering frequent subgraphs [J]. IEEE Transactions on Knowledge and Data Engineering, 2004, 16(9): 1038-1051.

[2] Wernicke S. Efficient detection of network motifs [J]. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2006, 3(4): 347-359.

[3] Cook D J, Holder L B. Substructure discovery using minimum description length and background knowledge [J]. In Journal of Artificial Intelligence Research, 1994, 1: 231-255.

[4] Jonyer I, Cook D J, Holder L B. Discovery and evaluation of graph-based hierarchical conceptual clusters [J]. Journal of Machine Learning Research, 2001, 2: 19-43.

[5] Deshpande M, Kuramochi M, Karypis G. Frequent substructure based approaches for classifying chemical compounds [J]. IEEE Transactions on Knowledge and Data Engineering, 2005, 17(8): 1036-1050.

[6] Yoshida K, Motoda H, Indurkhya N. Graph-based induction as a unified learning framework [J]. Journal of Applied Intelligence, 1994, 4: 297-328.

[7] Inokuchi A, Washio T, Okada T, et al. Applying the apriori-based graph mining method to mutagenesis data analysis [J]. Journal of Computer Aided Chemistry, 2001, 2: 87-92.

[8] Inokuchi A, Washio T, Motoda H. Generalization for frequent subgraph mining [J]. Transaction of the Japanese Society for Artificial Intelligence, 2004, 19: 368-378.

[9] Gudes E, Shimony S E, Vanetik N. Discovering frequent graph patterns using disjoint paths [J]. IEEE Transactions on Knowledge and Data Engineering, 2006, 18(11): 1441-1456.

[10] Kashtan N, Itzkovitz S, Milo R, et al. Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs [J]. Bioinformatics, 2004, 20(11): 1746-1758.

[11] Hu H, Yan X, Huang Y, et al. Mining coherent dense subgraphs across massive biological networks for functional discovery [J]. Bioinformatics, 2005, 21(1): 213-221.

[12] Ross D K, Srinivasan A, Dehaspe L. WARMR: A data mining tool for chemical data [J]. Journal of Computer Aided Molecular Design, 2001, 15: 173-181.

[13] Berg J, Lässig M. Local graph alignment and motif search in biological networks [J]. PNAS, 2004, 101: 14689-14694.

[14] Toran J. On the hardness of graph isomorphism [J]. SIAM Journal on Computing, 2004, 33(5): 1093-1108.

[15] Mason O, Verwoerd M. Graph theory and networks in biology [EB/OL]. [2007-08-17]. http://www.hamilton.ie/systemsbiology/ files/2006/ graph_theory_and_networks_in_biology.pdf.

[16] Uri Alon. Collection of complex networks[DB/OL]. [2007-08-20]. http://www.weizmann.ac.il/mcb/UriAlon/.