计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2015.20150180
52(8):1784-1793, 2015

一种面向蛋白质复合体检测的图聚类方法

王 杰 梁吉业 郑文萍
(山西大学计算机与信息技术学院 太原 030006)
(计算智能与中文信息处理教育部重点实验室(山西大学) 太原 030006)
(xhcwj@sina.com)

A Graph Clustering Method for Detecting Protein Complexes

Wang Jie, Liang Jiye, and Zheng Wenping
(*School of Computer & Information Technology, Shanxi University, Taiyuan* 030006)
(*Key Laboratory of Computation Intelligence & Chinese Information Processing (Shanxi University), Ministry of Education, Taiyuan* 030006)

Abstract Protein-protein interaction (PPI) networks are widely present in complex biological networks. The topological features of PPI networks play an important role in analyzing the functional modules in networks. Some graph clustering methods have been successfully used to complex networks to detect protein complexes in PPI networks. Traditional graph clustering algorithms in PPI analyzing methods primarily focus on hard clustering for a network, while, nowadays soft clustering algorithms to find overlapped clusters have become one of the hotspots of current research. Existing soft clustering algorithms pay less attention on small-scale non-dense clusters, while some small-scale non-dense clusters often have important biological meaning in PPI networks. A measuring method of the association strength of edges is developed based on node neighborhoods in networks, and then a soft clustering algorithm named flow-simulation graph clustering (F-GCL) on the basis of flow simulation is presented to detect complexes in a PPI network. Experiments show that the proposed soft clustering algorithm F-GCL can simultaneously find out overlapping clusters and small-scale non-dense clusters without improving the running time. Compared with MCODE (molecular complex detection), MCL (Markov clustering), RNSC (restricted neighborhood search clustering) and CPM (clique percolation method) algorithms on six Saccharomyces cerevisiae PPI networks, the algorithm F-GCL shows considerable or better performance on three evaluating indicators: *F-measure*, *Accuracy* and *Separation*.

Key words flow simulation; graph clustering; soft clustering; protein-protein interaction network; protein complex

摘 要 蛋白质互作用(protein-protein interaction, PPI)网络是广泛存在的一类复杂生物网络,其网络拓扑特征与功能模块分析密切相关.图聚类是对复杂网络进行分析和处理的一种重要计算方法.传统的PPI网络中蛋白质复合体检测算法通常对网络图中的对象进行硬划分,而寻找网络中的重叠簇的软聚类算法已成为当前研究热点之一.现有的软聚类算法较少关注寻找网络中具有重要生物意义的小规模非稠密簇.对此,基于网络中结点邻域给出了边关联强度的度量方法,并在此基础上提出了一种基于流

收稿日期:2015-02-28;**修回日期**:2015-05-22
基金项目:国家自然科学基金项目(61432011,61272004);山西省自然科学基金项目(2011011016-1);教育部高等学校博士学科点专项科研基金项目(200801081017)
通信作者:郑文萍(wpzheng@sxu.edu.cn)

模拟的 PPI 网络中复合体检测的图聚类(flow-simulation graph clustering, F-GCL)算法,该算法可以在快速发现 PPI 网络中的重叠簇的同时找到小规模非稠密簇;同时,与 MCODE(molecular complex detection),MCL(Markov clustering),RNSC(restricted neighborhood search clustering)和 CPM(clique percolation method)算法在 6 个酿酒酵母 PPI 网络上进行比较,该算法在 *F-measure*,*Accuracy*,*Separation* 方面表现了较好的性能.

关键词　流模拟;图聚类;软聚类;蛋白质互作用网络;蛋白质复合体

中图法分类号　TP181

蛋白质是构成生命体的关键成分并参与细胞中大多数的生物过程.细胞中的蛋白质通常不是独立地发挥作用,而是以蛋白质复合体等形式来实现特定的生物功能[1-2].近年来,通过酵母双杂交、微阵列等高通量技术获得了大规模的蛋白质互作用数据[3],利用这些蛋白质互作用数据可以直接构造蛋白质互作用(protein-protein interaction, PPI)网络,其中每个结点表示一个蛋白质,每条边表示蛋白质间的一个相互作用.蛋白质复合体通常可以表示为 PPI 网络中的紧密连通子网络,检测 PPI 网络中蛋白质复合体的问题就可以抽象为寻找网络图中的紧密连通子网络[4-5].

相较于基于向量模型的传统聚类方法,图聚类既保留了对象间原始存在的关联信息等局部属性,又能够利用图模型中的拓扑特征如结点度、边权重、路径距离等全局属性.在处理和分析复杂网络图方面,图聚类方法有天然的优势,能更好地发现复杂网络中紧密连通的簇结构,利用网络图的拓扑结构对图中结点进行划分,使得簇内边相对稠密而簇间边稀疏[6],每一个簇对应 PPI 网络中的一个蛋白质复合体.

2000 年,Dongen[7]给出了基于流模拟的无监督的 Markov 聚类(Markov clustring, MCL)算法,通过扩展与膨胀 2 种操作模拟流动过程,不断增强连接紧密顶点对的关联强度,不断减弱连接松散顶点对之间的关联强度. MCL 算法能够发现网络中星形结构的簇,但找不到重叠簇. 2012 年 Shih 和 Parthasarathy[8]提出了基于 MCL 算法的软聚类(soft regularized Markov clustering, SR-MCL)算法以获得重叠簇,但需要额外的去冗余簇过程.

2003 年,Bader 和 Hogue[9]基于稠密子图检测给出了分子复合体识别(molecular complex detection, MCODE)算法.从单体簇开始,依次选择具有最大权重的结点作为种子,迭代地将得分高于用户所设定阈值的邻居结点连接到对应簇中. 2005 年,Palla 等人[10]给出派系过滤算法(clique percolation method, CPM),从 k-团出发,合并具有 $k-1$ 个公共邻居结点的 k-团簇,CPM 可发现重叠簇,但不关注于网络图中大量存在的小规模非稠密复合体.

2004 年,King 等人[11]提出了基于划分的约束近邻搜索聚类(restricted neighborhood search clustering, RNSC)算法,根据结点连通性定义了聚类成本函数,通过最小化成本函数将网络划分为不同的簇,并通过簇的大小、簇的密度和功能同源性来过滤得到最终结果. RNSC 算法有 7～8 个运行参数,不能将结点划分入不同的簇,且其中的随机过程对计算结果稳定性有一定影响.

此外,2010 年 Qin 等人[12]给出了基于谱图理论的谱聚类(spectral clustering, SP)算法. 2011 年 Lee 等人[13]提出了基于多元信息的挖掘稠密重叠子图(mining dense overlapping subgraphs, MDOS)算法,2014 年 Xu 等人[14]结合蛋白质功能相似性给出了复合体预测(complex predictor, CPredictor)算法. MDOS 和 CPredictor 算法效果依赖于先验知识.

一个 PPI 网络中通常存在大量的重叠簇以及一些小规模非稠密簇,而上文所述经典图聚类方法无法同时发现 PPI 网络中的重叠簇以及小规模非稠密簇.基于此,本文提出了一种基于流模拟的 PPI 网络中复合体检测的软聚类(flow-simulation graph clustering, F-GCL)算法,可以快速实现对 PPI 网络中的蛋白质进行软聚类,并同时发现网络中的小规模复合体.该算法基于网络中结点邻域给出的边关联强度度量方法可以较准确地刻画结点间的连接倾向性;算法综合考虑图中边属性、结点属性和路径距离来进行种子结点选取和簇扩展,能找到 PPI 网络中的重叠簇及小规模非稠密簇,并且降低了重叠簇的冗余度、节省了去冗余过程.与 MCODE,MCL,RNSC,CPM 算法在 6 个酿酒酵母 PPI 网络上进行比较,该算法在 *F-measure*,*Accuracy*,*Separation* 方面显示出良好的性能.

1 基本概念

一个蛋白质互作用网络可以表示为一个包含自环的无向图 $G=(V,E)$,其中 V 表示结点集合,E 表示边集合. 每个结点表示一个蛋白质,每条边表示蛋白质间的相互作用. 结点自环表示在互作用网络中结点与其本身存在互作用. 图 G 中结点 v_i 和 v_j 之间的边表示为 v_iv_j 或 e_{ij}. 结点 v_i 和 v_j 间的一条路径是指顶点不重复的点边交替序列 $v_ie_{i0}v_0e_{01}v_1\cdots e_{mj}v_j$,其中所含边的数目称为该路径的长度. 从 v_i 到 v_j 的具有最小长度的路径称为 v_i 到 v_j 的最短路径,其长度称为 v_i 到 v_j 的距离.

设 $\boldsymbol{A}_{n\times n}$ 为网络图 G 的邻接矩阵,即:

$$A_{i,j}=\begin{cases}1, & v_iv_j\in E,\\ 0, & \text{否则},\end{cases} \tag{1}$$

其中,$n=|V|$.

结点 v 在图 G 中的开邻域 $N_G(v)$ 定义为 $N_G(v)=\{u|uv\in E(G)\}$,在不发生混淆时直接记作 $N(v)$,其中 $u\in N_G(v)$ 称为 v 的相邻顶点.

定义 1. 结点 v 的 r-邻接结点集合 $N_r(v)$ 表示为

$$N_r(v)=\begin{cases}N(v), & r=1;\\ N_{r-1}(v)\bigcup\{u\in V\mid f(v,u)=r\}, & r>1;\end{cases} \tag{2}$$

其中,$f(v,u)$ 表示结点 v 到结点 u 的距离,r 称为路径长度阈值.

2 基于流模拟的图聚类算法 F-GCL

流模拟算法的基本思想是基于顶点对的最短路径长度,结合复杂网络呈现的小世界[15]及 6 度分割等理论,以网络图中某些结点为源点,在其 r-邻接结点集合中进行功能流动,使得其他结点获得功能流量,最终高于所设定流量阈值的顶点将预测为与源点具有相同功能[16]. 本文提出的基于流模拟的图聚类算法 F-GCL 对 PPI 网络中的蛋白质复合体进行检测,主要包含 3 个关键步骤:种子选取、聚类流扩展和簇选择.

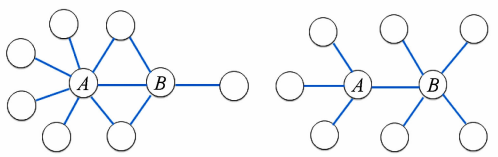
文献[17-19]指出一些蛋白质复合体在 PPI 网络中并不呈现局部稠密的形式,而仅与中心蛋白质连通. 为了更好地刻画这一特征,本文给出 PPI 网络中的边权重定义方法,以刻画网络中边端点间的关联强度;顶点权重定义为与之关联的边权重之和. 与经典相似性度量 Jaccard 相似度相比,该关联强度度量更准确刻画了结点连接结构的差异对结点间连接强度的影响以及叶子结点对高度结点的依赖. 算法过程为:首先根据顶点权重选择种子结点;然后在其 r-邻接结点集合内进行流模拟过程,对集合内顶点打分;最终与种子结点簇内连通且打分高于其权重 k 倍的顶点被认为与种子结点属于同一簇. 其中,k 是算法的唯一参数. 算法结束后,每一个簇对应一个蛋白质复合体.

2.1 计算边关联强度

聚类分析中通常以对象间距离或相似性度量为基础,目标是使类内紧密而类间稀疏. 网络图结点间基于结构等价性的相似性度量应用最为广泛的是 Jaccard 相似度,如式(3)所示:

$$J(v_iv_j)=\frac{|N(v_i)\bigcap N(v_j)|}{|N(v_i)\bigcup N(v_j)|}. \tag{3}$$

Jaccard 的基本假设认为若网络图中 2 个结点间的公共邻接点越多,这 2 个结点在结构上越相似或相等. 当 2 个相邻结点的拓扑结构差异较大时,该度量不能很好地反映结点之间的连接倾向性. 如图 1 所示,网络可以看作由以结点 A 为中心及以结点 B 为中心的 2 个星形子网络所构成.



(a) Example 1 (b) Example 2

Fig. 1 Two network examples.

图 1 2 个网络实例

考虑边 AB 的连接强度,除结点 A 外,图 1(a)中与结点 B 相邻的其他 3 个结点中,仅有 1 个不与结点 A 相邻;而如图 1(b)所示,与结点 B 相邻的其他 4 个结点中,有 2 个不与结点 A 相邻. 可以看出,图 1(a)中结点 B 更倾向与结点 A 连接. 图 1(a)(b)中 $J(AB)$ 均为 2/9,无法体现出两者在这类结构上的差异. 为了更好地考虑相邻结点连接结构的差异对两者连接强度的影响,本文基于结点间的邻域定义了关联强度来描述网络图中结点之间边连接的紧密程度.

对图 G 中的边 $uv\in E$,不失一般性,假设结点度 $d(v)\leqslant d(u)$,则 $|N_G(v)\bigcap N_G(u)|/|N_G(v)|\geqslant|N_G(v)\bigcap N_G(u)|/|N_G(u)|$. 即 v 和 u 的公共相邻点集 $N_G(v)\bigcap N_G(u)$ 在 v 的邻域 $N_G(v)$ 中所占比重相对大. 可以认为结点 v 和 u 位于同一功能模块中

的可能性与$|N_G(v)\cap N_G(u)|/|N_G(v)|$正相关.因此定义边$v_iv_j\in E(G)$的关联强度$w(v_iv_j)$为

$$w(v_iv_j)=\begin{cases}\dfrac{|N(v_i)\cap N(v_j)|}{\min\{|N(v_i)|,|N(v_j)|\}-1}, \\ \quad |N(v_i)|>1 且 |N(v_j)|>1; \\ 1,\ |N(v_i)|=1 或 |N(v_j)|=1.\end{cases} \tag{4}$$

边关联强度w的取值范围为$[0,1]$.当相邻结点v_i和v_j间不存在任何共同邻居且不存在叶子结点时$w(v_iv_j)=0$,当结点v_i的邻居集合$N(v_i)$为结点v_j邻居集合$N(v_j)$的子集或者v_i的唯一邻居为v_j,则$w(v_iv_j)=1$.此时图1(a)中$w(AB)=2/3$,图1(b)中$w(AB)=2/4$,体现出了两者间结构上的差异.

关联强度w描述了:1)高度结点的吸引力或低度结点的倾向性.如图2所示,结点B的邻居全部与结点A直接连通,因此可以认为结点B非常紧密地连通于结点A,$w(AB)=1$;对于与结点A连通的结点H和结点E分别有:$|N(A)|\cap|N(H)|=2$,$|N(A)|\cap|N(E)|=2$,因为$|N(H)|=4<|N(A)|$有$w(AH)=2/3$,$|N(E)|=9<|N(A)|$则$w(AE)=0.25$.所以$w(AH)>w(AE)$意味着结点A与结点H间相同邻接点所占比重更高,结点H相较于结点E更加接近于结点A或者说结点A对结点H吸引力更大;结点D与结点G没有任何共同邻居$w(DG)=0$;由$|N(F)|=1$可以认为结点F与结点E非常紧密有$w(EF)=1$.2)对网络图中极其稀疏部分的划分.如图3是酿酒酵母的1个局部子网络,直观上可分为5组分.由于连通5组分的结点间无任何公共邻居点,关联强度$w=0$一定程度上描述了5个组分间的稀疏边界,从而对网络图中边极其稀疏的边界进行初始划分,最后利用簇判定条件来进一步确定类内成员及类边界.

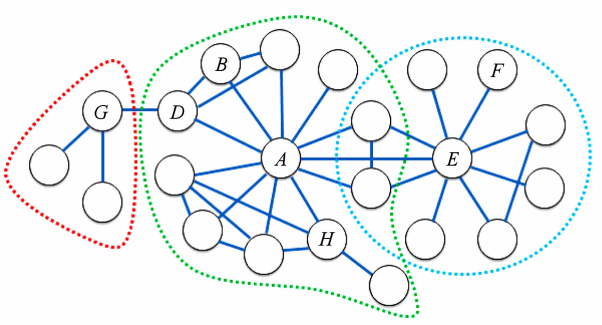


Fig. 2　A network example to illustrate how to calculate the strength of association.

图2　一个网络实例说明如何计算关联强度

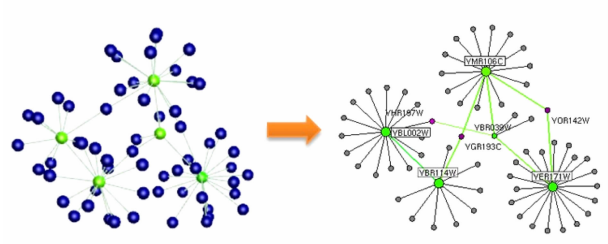


Fig. 3　A local sub network of Saccharomyces cerevisiae.

图3　酿酒酵母的局部子网络

2.2　种子结点选取

真实网络大多是结点度分布符合幂律分布的无标度网络,具有严重的异质性,少数的高度结点在网络中起到了至关重要的主导作用.基于此,将结点度作为种子结点的选取标准是一种简单而常见的方式,但该方式无法对结点的重要性作细致的定量分析,一个结点的重要程度不仅与结点间的度数有关,还取决于结点间的关联强度.1个对象P与其他多个对象有联系,P对于其中一些对象的影响可能比较轻微,而对另一些对象的影响可能相对较大.由此,基于边关联强度的种子结点权重定义既考虑结点度,同时也反映了该结点在局部范围内的重要程度.结点v_i权重定义如下:

$$w(v_i)=\sum_{v_j\in N(v_i)}w(v_iv_j). \tag{5}$$

定义2. 种子结点.给定网络图$G=(V,E)$,一个结点$v\in V$是一个种子结点,当该结点满足如下条件:

1) $w(v)>0$;

2) $w(v)=\max\{w(v_j)\},v_j\notin\bigcup C_i$.

其中,C_i表示已得到的第i个簇.

对网络图中结点依结点权重从大到小排序,依次选取当前未划入任何簇中且具有最大权重的结点作为下一个簇的种子.即已有簇中的所有结点均不可作为下一个簇的种子,但可以成为其他簇中的成员结点,由此既减少了冗余簇的产生且无需额外的过滤方法,同时也允许一个结点被划分到不同的簇中进而发现簇间的重叠部分.如图2所示为本算法在$k=1.8$时得到的3个不同簇,根据种子结点选取规则,该网络图中的中心结点依次为结点A,E,G.首先得到以结点A为中心的簇$C(A)$,此时$C(A)$内的所有结点将不作为种子结点出现,但可以作为其他簇中的成员结点从而达到软划分,如$|C(A)\cap C(E)|=2$.

2.3　簇的发现

本文算法F-GCL基于网络流思想对种子结点进行扩展,进而发现新簇.图论中,一个流网络表示

为 $G=(V,E,w)$,其中 w 表示边的容量即权重,对于每条边$(uv)\in E$ 具有一个非负实数权重 $w(uv)$. 网络中的流从 1 个源点流向 1 个或多个终点,其最大流量受到边权重的限制. 通过对网络流理论中流守恒特性的弱化,即可发展出一个直观的基于流模拟的图聚类算法.

最初,一个种子结点 v_s 被选择为流动的源点,以该源点为中心的簇记为 $C(v_s)$. 从源点 v_s 开始,在第 l 次迭代中距离源点的路径距离为 l ($l\in\{0,1,2,3\}$)以内的结点将获得流量,流量是基于关联强度 w 的累加. 迭代结束后,直接与源点相连的结点将获得 3 次的流量,路径距离为 2 的结点将获得 2 次的流量,以此类推,公式化表示为

$$R_l(v_i)=R_{l-1}(v_i)+\sum_{v_x\in N(v_i),R_{l-1}(v_x)>0}w(v_xv_i). \tag{6}$$

初始时,R 为

$$R_0=\begin{cases}\infty, & v_s \text{ 为种子结点;}\\ 0, & \text{否则.}\end{cases} \tag{7}$$

r 次迭代后,每个结点所得到的总流量 $FlowScore$ 定义为

$$FlowScore(v_i)=\sum_{l=1}^{r}\sum_{\substack{v_x\in N(v_i),\\ R_{l-1}(v_x)>0}}w(v_xv_i). \tag{8}$$

网络图背景下,每个簇直观上应该是连通的,即一个簇中任意一对结点间至少存在一条连通路径,若结点 u 无法到达结点 v,则它们无法分组到同一个簇中. 满足判定条件 F 并且至少存在一条边与当前 $C(v_s)$ 连通的蛋白质被依次加入到簇 $C(v_s)$ 中,即最终所得簇中的所有成员结点均满足该判定条件且连通,F 定义如下:

$$F(v_i)=FlowScore(v_i)-k\times w(v_i)\geqslant 0, \tag{9}$$

其中,k 为判定参数,判定函数 F 具有如下性质:

性质 1. 在连通网络图中,当 $k=0$ 时,距离种子结点 r 内的所有结点属于一个簇,若 r 为该网络图直径,则该网络图中所有结点为一个簇.

性质 2. 在连通网络图中,当 $k=(z,\infty)$ 时,该网络图中每个种子结点为一个簇. 其中 $z=\max\{FlowScore(v)/w(v)\}$.

性质 3. 在连通网络图中,当 k 单调递增时,簇的数量单调非减.

计算方法的研究过程中应该尽可能避免无规则随机过程的引入,但设置数量合适的参数有助于处理不同结构的数据,一定程度上提高方法的泛化能力. 不同的参数设置可能会给出不同的结果供专业人员根据自身经验及需要进行进一步的分析,这完全符合目前科学研究特别是跨领域研究的过程. 在图 2 所示实例中,根据 k 的不同将会得到不同的网络划分,当 $k=1$ 时 $C(A)=C(A)\bigcup C(E)$,即以结点 A 为中心的簇与以结点 E 为中心的簇将合并为一个以结点 A 为种子的簇. 应该采用哪种划分将由面向的实际问题来确定. 算法流程如算法 1 所示:

算法 1. F-GCL 算法流程.

输入:网络图 $G=(V,E)$ 邻接矩阵 $\boldsymbol{A}_G$(其中 $|V|=n$,$V=\{v_1,v_2,\cdots,v_n\}$)、流动半径 r、判定条件参数 k;

输出:簇集合 C.

① $C=\varnothing$;

② 根据式(4)计算图 G 中每条边的关联强度:$\boldsymbol{W}_{n\times n}(G)=Weight(\boldsymbol{A}_G)$;

③ 根据边关联强度矩阵 $\boldsymbol{W}_{n\times n}(G)$ 计算结点权重:$w(v_i)=\sum_{v_j\in N(v_i)}w(v_iv_j)$;

④ 根据权重从大到小对结点进行排序,新结点序列为 $u_1,u_2,\cdots,u_n$,即 $w(u_1)\geqslant w(u_2)\geqslant\cdots\geqslant w(u_n)$;

⑤ 根据种子选取策略查找源点 u_i,通过模拟流动过程,以流动半径 r 内的所有结点为判定对象,产生以 u_i 为核心的簇 C_i;

⑥ $C=C\bigcup\{C_i\}$;

⑦ 重复步骤⑤⑥直到无法找到新的种子结点.

3 实验与结果分析

本文选择多个 PPI 网络数据集和复合体参照数据集对算法进行验证,全面反映算法的稳定性及泛化能力. 从蛋白质互作用数据库 BioGRID:BIOGRID-ORGANISM-Saccharomyces_cerevisiae 中根据文献来源[20-26]分别获取了 6 个不同酿酒酵母互作用网络(见表 1 所示)进行算法实验,并以 MIPS[20] 的蛋白质复合体数据对算法结果进行评价.

Table 1 Six PPI Networks

表 1 6 个蛋白质互作用网络

Items	Uetz	Ito	Ho	Gavin02	Gavin06	Kragan
Proteins	926	2937	1564	1352	1430	2675
Interactions	865	4038	3600	3210	6531	7088

3.1　评价指标

本文选择3个统计量 *F-measure*, *Accuracy*, *Separation* 作为评价指标[12,20,27],以从不同的角度对算法作出客观的性能分析及评价. *F-measure* 由 *precision* 和 *recall* 两部分构成,对算法的聚类结果与复合体参照集从整体上进行评价. *Accuracy* 作为一个经典的评价指标,对簇与复合体参照集的匹配程度进行度量;但其本身存在一定局限性,例如聚类结果中的单体簇可能会促使该评价指标无法合理反映簇的质量,此时其分量 PPV 将取最大值1. 针对 *Accuracy* 指标在单体簇等情况下的评价局限性,文献[20]定义了簇与复合体参照数据集的新统计量 *Separation*.

3.2　结果分析

本文提出的F-GCL算法仅有1个参数 k,当结点 v 的打分 $FlowScore(v)$ 与其自身权重 $w(v)$ 的比值 $FlowScore(v)/w(v)\geqslant k$ 时,认为当前结点 v 与当前簇联系紧密,判断 v 与当前种子结点属于同一簇. 依经验,参数默认取值为 $k=3.0$.

PPI网络中一些蛋白质在不同的复合体中参与功能的表达,MCL,RNSC等硬划分算法无法将一个蛋白质划分到不同的簇中,而本文所提出的F-GCL算法可以很好地找到网络中的重叠簇,表2给出了F-GCL算法得到的部分重叠簇.

PPI网络中广泛存在着小规模非稠密复合体,尤其针对大小低于2的复合体,现有算法通常会丢失或抽出后单独处理. 例如MIPS复合体参照集中大小为2的复合体约为28.6%. MCODE与CPM算法不能给出这类潜在的簇. F-GCL算法在整个流动过程中会对每个结点进行遍历,将网络图中的结点作为不同的角色分配到各自的簇中,其簇的个数与大小依赖于结点所得流量以及与种子结点的连通性,可以很好地发现这些有重要意义的小簇. 如表3列出了该算法所得到的部分小簇,这些小簇都真实地存在于复合体参照集中,表3中第2列为对应复合体名称.

Table 2　Examples of Overlap Clusters Obtained by F-GCL

表2　F-GCL算法获得的部分重叠簇实例

C_A	C_B
YBR253W, **YML007W**, **YGR252W**, YOR174W, YHR041C, **YBR081C**, **YDR448W**, YNR010W, YOL051W, YPR070W, YMR112C, YHR058C, YOL135C, YBR193C, YKR095W, YGL151W, YPR168W, YGL025C, YLR071C, YNL236W, YER022W, YGR104C, YDR308C, YCR081W, YDR443C, YPL042C, **YHR099W**, **YCL010C**, **YOL148C**, **YPL254W**, **YDR392W**, **YDR167W**, **YGL066W**, **YLR055C**, **YDR176W**, **YMR223W**	YGL112C, YGR274C, YER148W, YBR198C, **YDR167W**, YCR042C, YMR227C, **YML007W**, **YGR252W**, YDR044W, **YDR176W**, YML015C, YDR145W, YPL011C, YMR236W, YMR005W, **YCL010C**, **YOL148C**, **YBR081C**, YDR448W, **YPL254W**, **YGL066W**, **YLR055C**, **YDR392W**, **YHR099W**, **YMR223W**, YML114C
YPR175W, **YDR121W**, **YBR278W**	YNL262W, **YPR175W**, **YBR278W**, **YDR121W**,
YOR116C, YOR207C, YPR110C, YPR190C, YNR003C, **YBR154C**, **YOR224C**, **YPR187W**, YKL144C, YKR025W, YDL150W, YDR045C, YNL151C, YJL011C, YDR005C, YNL113W, **YPR010C**, **YJR063W**, **YOR210W**, **YNL248C**, YGL156W, **YOR340C**	YOR341W, **YPR010C**, **YJR063W**, **YOR340C**, **YOR210W**, **YBR154C**, **YOR224C**, **YPR187W**, **YNL248C**
YGR119C, YMR294W, YGL170C, YGL172W, **YJL041W**, YPR083W	YJL061W, YKL061W, **YJL041W**
YBR253W, **YML007W**, **YGR252W**, YOR174W, YHR041C, **YBR081C**, **YDR448W**, YNR010W, YOL051W, YPR070W, YMR112C, YHR058C, YOL135C, YBR193C, YKR095W, YGL151W, YPR168W, YGL025C, YLR071C, YNL236W, YER022W, YGR104C, YDR308C, YCR081W, YDR443C, YPL042C, **YHR099W**, **YCL010C**, **YOL148C**, **YPL254W**, **YDR392W**, **YDR167W**, **YGL066W**, **YLR055C**, **YDR176W**, **YMR223W**	YGL112C, YGR274C, YER148W, YBR198C, **YDR167W**, YCR042C, YML114C, YMR227C, **YML007W**, **YGR252W**, YDR044W, **YDR176W**, YML015C, YDR145W, YPL011C, YMR236W, YMR005W, **YCL010C**, **YOL148C**, **YBR081C**, **YDR448W**, **YPL254W**, **YGL066W**, **YLR055C**, **YDR392W**, **YHR099W**, **YMR223W**
YMR308C, YER178W, **YOR362C**, **YMR314W**, **YOL038W**, **YML092C**, YGR113W, YOL108C, YOR373W, YLL021W, YPL020C, YAL061W, YML059C, **YFR050C**, **YGR135W**, **YER094C**, **YFL007W**, **YIL009C-A**	YDL188C, YML109W, **YOR362C**, YPR103W, **YFR050C**, **YMR314W**, **YOL038W**, **YML092C**, YGR135W, YOR157C, YGR253C, **YER094C**, YGL011C, YJR009C, YOR014W, **YFL007W**, **YIL009C-A**, YJL001W, YDL134C

Table 3 Examples of Small Clusters Obtained by F-GCL

表 3 F-GCL 算法获得的部分小簇实例

Cluster	Complex Label
YER090W, YKL211C	Anthranilate-synthase
YMR106C, YMR284W	Ku-complex
YAL009W, YHR004C	Nem1p-Spo7p-complex
YLR182W, YDL056W	MBF-complex
YNL126W, YHR172W, YLR212C	Gamma-tubulin-complex
YJR005W, YJR058C, YOL062C, YBL037W	AP-2-complex

由表 2 和表 3 可看出本文提出的 F-GCL 算法可以找到重叠簇,同时可以发现 PPI 中存在的小规模蛋白质复合体. 下面在 *F-measure*, *Accuracy*, *Separation* 三个统计指标上评估各算法的性能.

以 MIPS 复合体为参照集,将算法 F-GCL 与 MCL, MCODE, RNSC, CPM 进行比较,结果如表 4～10 所示. 表 4～9 分别给出了算法在 6 个数据集上实验结果的 *F-measure*, *Accuracy*, *Separation* 值,以及各算法对应评价指标的排名. 表 10 给出了算法在 *F-measure*, *Accuracy*, *Separation* 评价指标上的平均排名. 结果表明本文算法 F-GCL 取得了较好的性能表现.

Table 4 Results on Uetz Data Set

表 4 Uetz 数据集实验结果

Algorithm	*F-measure*	*F-measure* Order	*Accuracy*	*Accuracy* Order	*Separation*	*Separation* Order
F-GCL	0.1119	2	0.2532	1	0.2496	1
RNSC	0.1133	1	0.2509	2	0.2329	2
MCL	0.0990	3	0.2230	3	0.2117	3
MCODE	0.0328	4	0.0715	5	0.1101	4
CPM	0.0260	5	0.0838	4	0.0934	5

Table 5 Results on Ito Data Set

表 5 Ito 数据集实验结果

Algorithm	*F-measure*	*F-measure* Order	*Accuracy*	*Accuracy* Order	*Separation*	*Separation* Order
F-GCL	0.0884	3	0.2788	2	0.2196	1
RNSC	0.0985	2	0.3226	1	0.1992	2
MCL	0.1084	1	0.2391	3	0.1586	3
MCODE	0.0087	5	0.0669	5	0.0853	5
CPM	0.0370	4	0.1498	4	0.1404	4

Table 6 Results on Ho Data Set

表 6 Ho 数据集实验结果

Algorithm	*F-measure*	*F-measure* Order	*Accuracy*	*Accuracy* Order	*Separation*	*Separation* Order
F-GCL	0.1406	2	0.2964	2	0.2610	1
RNSC	0.1539	1	0.3031	1	0.2310	2
MCL	0.1016	4	0.2641	3	0.2069	3
MCODE	0.0174	5	0.0911	5	0.1066	5
CPM	0.1142	3	0.2140	4	0.1312	4

Table 7 Results on Gavin02 Data Set

表 7 Gavin02 数据集实验结果

Algorithm	*F-measure*	*F-measure* Order	*Accuracy*	*Accuracy* Order	*Separation*	*Separation* Order
F-GCL	0.3902	1	0.4535	1	0.3571	1
RNSC	0.2460	3	0.4088	3	0.2529	3
MCL	0.3379	2	0.4208	2	0.3000	2
MCODE	0.0896	5	0.1624	5	0.2391	4
CPM	0.2259	4	0.3069	4	0.1179	5

Table 8　Results on Gavin06 Data Set

表 8　Gavin06 数据集实验结果

Algorithm	*F-measure*	*F-measure* Order	*Accuracy*	*Accuracy* Order	*Separation*	*Separation* Order
F-GCL	0.357 5	1	0.447 4	3	0.314 2	3
RNSC	0.257 7	4	0.462 4	2	0.275 1	4
MCL	0.297 5	2	0.471 1	1	0.319 5	1
MCODE	0.274 9	3	0.296 9	5	0.316 5	2
CPM	0.185 4	5	0.412 7	4	0.099 7	5

Table 9　Results on Kragan Data Set

表 9　Kragan 数据集实验结果

Algorithm	*F-measure*	*F-measure* Order	*Accuracy*	*Accuracy* Order	*Separation*	*Separation* Order
F-GCL	0.229 7	2	0.441 9	2	0.265 5	1
RNSC	0.180 8	4	0.481 9	1	0.247 0	4
MCL	0.212 0	3	0.438 2	3	0.254 6	2
MCODE	0.172 5	5	0.279 0	5	0.249 4	3
CPM	0.299 1	1	0.387 2	4	0.125 7	5

Table 10　Average Order of Algorithms on *F-measure*, *Accuracy*, *Separation*

表 10　算法在 *F-measure*, *Accuracy*, *Separation* 上的平均排名

Algorithm	Average Order		
	F-measure	*Accuracy*	*Separation*
F-GCL	1	2	1
RNSC	2	1	3
MCL	2	3	2
MCODE	5	5	4
CPM	4	4	5

综上所述,本文提出的 F-GCL 算法既可以将一个对象划分到不同的类从而发现重叠簇,同时也能找出 PPI 网络中存在的小规模非稠密复合体. 算法运行参数少且在发现簇的过程中无需去冗余操作,因此可以快速发现 PPI 网络中的蛋白质复合体. 此外,在 *F-measure*, *Accuracy*, *Separation* 三个统计指标上的评估表明该算法具有较好的性能.

4　结束语

已有的基于硬划分思想蛋白质复合体检测图聚类算法(如 MCL,RNSC 等),不能找到 PPI 网络中存在的重叠簇;而基于密度的复合体检测算法(如 MCODE,CPM 等)侧重于寻找网络图中的稠密子图,而不关注小规模非稠密簇. 到目前为止,很少有算法能够同时找到重叠簇以及小规模非稠密簇. 本文提出的基于流模拟的 PPI 网络中复合体检测软聚类算法 F-GCL,在不破坏原始网络结构的情况下可以快速发现 PPI 网络中的重叠簇,同时弱化了方法对密度等拓扑属性的依赖性,使网络稠密部分获得较高流量得分,稀疏部分也可以获得合适的流量得分,从而能够识别网络中存在的小规模非稠密复合体. 以 MIPS 为复合体参照集,在 3 个统计指标 *F-measure*, *Accuracy*, *Separation* 上将该算法与 RNSC,MCL,MCODE,CPM 算法进行比较表明,F-GCL 在 6 个不同 PPI 数据集上表现出了较好的性能.

PPI 网络中的蛋白质复合体发现有助于细胞中功能的预测及进一步分析[28],是生物信息学中一个重要的研究课题,图聚类在计算方法方面为复合体检测提供了有力的支持. 随着复杂网络的不断产生与丰富,如何使图聚类算法能够适应大规模或动态网络是一个新的挑战;另外,针对复杂网络聚类中存在的低密度高模块性和高密度低模块性[29]以及重叠簇,仍然需要改进或发展出更好的算法来推动复杂网络的进一步分析,同时发展适合于重叠簇的评价方法也将对图聚类算法具有重要的引导作用.

参　考　文　献

[1] Huang Chien-Hung, Chou Szu-Yu, Ng Ka-Lok. Robustness of protein complex networks [C] //Proc of the 2nd Int Conf on Computer Science and Service System (CSSS). Los Alamitos, CA: IEEE Computer Society, 2012: 841-844

[2] Guo Maozu, Dai Qiguo, Xu Liqiu, et al. On protein complexes identifying algorithm based on the novel modularity function [J]. Journal of Computer Research and Development, 2014, 51(10): 2178-2186 (in Chinese)
(郭茂祖, 代启国, 徐立秋, 等. 一种蛋白质复合体模块度函数及其识别算法[J]. 计算机研究与发展, 2014, 51(10): 2178-2186)

[3] Yu Liang, Gao Lin, Sun Penggang. Research on algorithms for complexes and functional modules prediction in protein-protein interaction networks [J]. Chinese Journal of Computers, 2011, 34(7): 1239-1251 (in Chinese)
(鱼亮, 高琳, 孙鹏岗. 蛋白质网络中复合物和功能模块预测算法研究[J]. 计算机学报, 2011, 34(7): 1239-1251)

[4] Ji Junzhong, Liu Zhijun, Liu Hongxin, et al. An overview of research on functional module detection for protein-protein interaction networks [J]. Acta Automatica Sinica, 2014, 40(4): 577-593 (in Chinese)
(冀俊忠, 刘志军, 刘红欣, 等. 蛋白质相互作用网络功能模块检测的研究综述[J]. 自动化学报, 2014, 40(4): 577-593)

[5] Zhao Bihai, Wang Jianxin, Li Min, et al. Detecting protein compleses based on uncertain graph model [J]. IEEE/ACM Trans on Computational Biology and Bioinformatics, 2014, 11(3): 486-497

[6] Schaeffer S E. Graph clustering [J]. Computer Science Review, 2007, 1(1): 27-64

[7] Dongen S. Graph clustering by flow simulation [D]. Utrecht, Netherlands: University of Utrecht, 2000

[8] Shih Y K, Parthasarathy S. Identifying functional modules in interaction networks through overlapping Markov clustering [J]. Bioinformatics, 2012, 28(18): 473-479

[9] Bader G D, Hogue C W. An automated method for finding molecular complexes in large protein interaction networks [J]. BMC Bioinformatics, 2003, 4(1): 2

[10] Palla G, Derényi I, Farkas I, et al. Uncovering the overlapping community structure of complex networks in nature and society [J]. Nature, 2005, 435(7043): 814-818

[11] King A D, Przulj N, Jurisica I. Protein complex prediction via cost-based clustering [J]. Bioinformatics, 2004, 20(17): 3013-3020

[12] Qin Guimin, Gao Lin. Spectral clustering for detecting protein complexes in protein-protein interaction (PPI) networks [J]. Mathematical and Computer Modelling, 2010, 52(11/12): 2066-2074

[13] Lee A J, Lin M C, Hsu C M. Mining dense overlapping subgraphs in weighted protein-protein interaction networks [J]. BioSystems, 2011, 103(3): 392-399

[14] Xu Bin, Guan Jihong. From function to interaction: A new paradigm for accurately predicting protein complexes based on protein-to-protein interaction networks [J]. IEEE/ACM Trans on Computational Biology and Bioinformatics, 2014, 11(4): 616-627

[15] Watts D J, Strogatz S H. Collective dynamics of 'small-world' networks [J]. Nature, 1998, 393 (6684): 440-442

[16] Nabieva E, Jim K, Agarwal A, et al. Whole-proteome prediction of protein function via graph-theoretic analysis of interaction maps [J]. Bioinformatics, 2005, 21 (Suppl1): 302-310

[17] Spirin V, Mirny L A. Protein complexes and functional modules in molecular networks [J]. Proc of the National Academy of Sciences of the United States of America, 2003, 100(21): 12123-12128

[18] Ren Jun, Wang Jianxin, Li Min. Identifying protein complexes based on local fitness Method [C] //Proc of the 6th IEEE Int Conf on Bioinformatics and Biomedicine (BIBM). Piscataway, NJ: IEEE, 2012: 205-210

[19] Chen Bolin, Wu Fangxiang. Identifying protein complexes based on multiple topological structures in PPI networks [J]. IEEE Trans on NanoBioscience, 2013, 12(3): 165-172

[20] Brohée S, Helden J V. Evaluation of clustering algorithms for protein-protein interaction networks [J]. BMC Bioinformatics, 2006, 7(1): 488

[21] Uetz P, Giot L, Cagney G, et al. A comprehensive analysis of protein-protein interactions in Saccharomyces cerevisiae [J]. Nature, 2000, 403(6770): 623-627

[22] Ito T, Tashiro K, Muta S, et al. Toward a protein-protein interaction map of the budding yeast: A comprehensive system to examine two-hybrid interactions in all possible combinations between the yeast proteins [J]. Proceedings of the National Academy of Sciences of the United States of America, 2000, 97(3): 1143-1147

[23] Ho Y, Gruhler A, Heilbut A, et al. Systematic identification of protein complexes in Saccharomyces cerevisiae by mass spectrometry [J]. Nature, 2002, 415(6868): 180-183

[24] Gavin A C, Bosche M, Krause R, et al. Functional organization of the yeast proteome by systematic analysis of protein complexes [J]. Nature, 2002, 415(6868): 141-147

[25] Gavin A C, Aloy P, Grandi P, et al. Proteome survey reveals modularity of the yeast cell machinery [J]. Nature, 2006, 440(7084): 631-636

[26] Krogan N J, Cagney G, Yu H, et al. Global landscape of protein complexes in the yeast Saccharomyces cerevisiae [J]. Nature, 2006, 440(7084): 637-643

[27] Li Xiaoli, Wu Min, Kwoh Chee-Keong, et al. Computational approaches for detecting protein complexes from protein interaction networks: A survey [J]. BMC Genomics, 2010, 11 (Suppl1): S3

[28] Liu Qian, Chen Yi-Ping Phoebe, Li Jinyan. *k*-Partite cliques of protein interactions: A novel subgraph topology for functional coherence analysis on PPI networks [J]. Journal of Theoretical Biology, 2014, 340: 146-154

[29] Ren Jun, Wang Jianxin, Li Min, et al. Identifying protein complexes based on density and modularity in protein-protein interaction network [J]. BMC Systems Biology, 2013, 7 (Suppl4): S12

Wang Jie, born in 1988. PhD candidate at Shanxi University. Student member of China Computer Federation. His research interests include data mining, bioinformatics.

Liang Jiye, born in 1962. Professor and PhD supervisor at Key Laboratory of Computation Intelligence & Chinese Information Processing (Shanxi University), Ministry of Education. Distinguished member of China Computer Federation. His research interests include granular computing, data mining, machine learning, etc(ljy@sxu.edu.cn).

Zheng Wenping, born in 1979. PhD and associate professor at Shanxi University. Member of China Computer Federation. Her research interests include graph theory algorithms, bioinformatics, etc.

2015 年起《计算机研究与发展》双月将固定领域专题

致广大读者和作者:

本刊从 2015 年起将双数期约 1/2 版面固定为某个领域,每年将策划该领域的一个热点主题进行集中报道.具体的征文通知将在专题发表前 6 个月发布,请关注期刊网站!

此外,本刊依然欢迎自由来稿.谢谢!

具体领域分布及执行领域编委如下:

刊期	领域	领域编委	投稿方式	截稿日期
2 期	软件技术(含数据库)	孟小峰 xfmeng@ruc.edu.cn 中国人民大学	期刊网站投稿, 备注填写“年+专题名称”	上一年 10 月 1 日左右 (以具体征文通知为准)
4 期	网络技术	林闯 chlin@tsinghua.edu.cn 清华大学	期刊网站投稿, 备注填写“年+专题名称”	上一年 12 月 1 日左右 (以具体征文通知为准)
6 期	体系结构	刘志勇 zyliu@ict.ac.cn 中国科学院计算技术研究所	期刊网站投稿, 备注填写“年+专题名称”	当年 2 月 1 日左右 (以具体征文通知为准)
8 期	人工智能	周志华 zhouzh@nju.edu.cn 南京大学	期刊网站投稿, 备注填写“年+专题名称”	当年 4 月 1 日左右 (以具体征文通知为准)
10 期	信息安全	曹珍富 zfcao@sjtu.edu.cn 上海交通大学	期刊网站投稿, 备注填写“年+专题名称”	当年 6 月 1 日左右 (以具体征文通知为准)
12 期	应用技术	郑庆华 qhzheng@mail.xjtu.edu.cn 西安交通大学	期刊网站投稿, 备注填写“年+专题名称”	当年 8 月 1 日左右 (以具体征文通知为准)