动态蛋白质相互作用网络中基于 GPU 矩阵乘的马尔可夫聚类研究

张锦雄　李陶深

广西大学计算机与电子信息学院，南宁，530004
广西高校并行与分布式技术重点实验室，南宁，530004

摘　要　本文首先基于 GPU 实现了矩阵乘并行算法 DNS，然后分别针对蛋白质复合物和功能模块研究了动态蛋白质相互作用网络的构造，随后将基于 GPU 矩阵乘的 MCL 算法用于动态蛋白质相互作用网络，分别设计了挖掘蛋白质复合物的算法 MCL-IPC-STDPINs 和挖掘蛋白质功能模块的算法 SBBMCL-IFM-TDPINC，最后对所设计的两个算法进行了时空复杂度和并行性分析，说明了算法的可行性和有效性。

关键字　蛋白质复合物，蛋白质-蛋白质相互作用，时空动态蛋白质相互作用网络，时序动态蛋白质相互作用网络链，马尔可夫聚类，基于 GPU 矩阵乘

Study on A GPU Matrix Multiplication-based Markov Clustering in Dynamic Protein Interaction Networks

Zhang Jin Xiong

School of Computer, Electronics & Information
Guangxi University,
Nanning 530004;
Guangxi Colleges and Universities Key Laboratory of Parallel and Distributed Computing
Nanning 530004;
zhangjx@gxu.edu.cn

Li Tao Shen

School of Computer, Electronics & Information
Guangxi University,
Nanning 530004;
Guangxi Colleges and Universities Key Laboratory of Parallel and Distributed Computing
Nanning 530004;
tshli@gxu.edu.cn

Abstract—In this paper, a GPU-based matrix multiplication is implemented according to parallel matrix multiplication algorithm DNS, two types of dynamic protein interaction networks are constructed for mining protein complexes and functional modules respectively, the main idea of GPU matrix multiplication-based Markov Clustering is utilized to design two methods MCL-IPC-STDPINs and SBBMCL-IFM-TDPINC respectively, and two designed methods are analyzed in terms of time and space complexity and parallelism.

Key words—protein, complex, protein-protein interaction, spatiotemporal dynamic protein interaction networks, temporal dynamic protein interaction network chain, Markov Clustering, GPU-based matrix multiplication

1 引　言

蛋白质是生物功能的执行者，大多数蛋白质不单独发挥生物功能，蛋白质-蛋白质相互作用(Protein-Protein Interaction, PPI)是蛋白质实施生物功能的主要形式。经过多年的研究和大量实验，细胞器水平的 PPI 数据与日俱增。随着 PPI 数据的大量积累，研究者们试图从中发现生物过程的功能单位及其作用机理。蛋白质复合物和功能模块是相互作用的蛋白质参与生物过程时形成的常见功能子结构[1]，正确全面认知蛋白质复合物和功能模块对于理解生命有机体细胞器功能机理和结构组织具有重要意义。于是，以 PPI 数据为基础挖掘蛋白质复合物和功能模块的研究成为蛋白质组学研究的热点。以蛋白质为结点，以 PPI 为边，基于 PPI 数据建模为网络，形成蛋白质相互作用网络。随着酵母双杂交(Y2H)[2]技术、串联亲和纯化-质谱(TAP-MS)[3]技术和蛋白质芯片(Protein Chip)[4]技术等高通量实验技术的飞速发展，PPI 数据呈指数级增长，基于蛋白质相互作用网络系统地挖掘蛋白质复合物和功能模块的计算方法层出不穷。

自从 Van Dongen 提出马尔可夫聚类算法 MCL[5]以来，MCL 以其简单、适应性强、鲁棒性强、可扩展、快速有效以及较少的参数而被广泛使用，尤其在生物信息学领域中被用于蛋白质相互作用网络[6-9]中挖掘两种功能子结构：蛋白质复合物[10-13]和功能模块[14-18]。这些研究都是将 MCL 算法用于单幅静态蛋白质相互作用网络，以挖掘功能子结构。

细胞周期或细胞响应环境刺激都会引发不同的生物过程，在此过程中蛋白质会根据功能的需要参与蛋白质复合物的装配和解配[19]。当前开放数据库的蛋白质相互作用数据是在不同的时间地点条件下产生的，这些蛋白质相互作用数据仅说明蛋白质之间存在相互作用，但却没有说明这些相互作用在何时何地发生。

基因表达数据是一组基因在若干均匀间隔时间点上转录的 mRNA 的丰度采样值，它可以反映一组基因在整个采样过程的动态表达模式。蛋白质亚细胞定位数据记录了一组蛋白质在细胞周期中出现在不同亚细胞区室的情况，它反映了一个细胞周期中蛋白质为发挥生物功能而曾经出现的亚细胞场所。易见，蛋白质亚细胞定位数据提供了空间信息，而基因表达数据提供了时间信息。显然，基因表达和蛋白质亚细胞定位具有具有时空动态性，很自然蛋白质相互作用也随之具有时空动态性。简言之，蛋白质之间的相互作用是随时空环境变化而呈动态性[20]。

为了近真地反映细胞系统中蛋白质及其相互作用的动态性，人们研究动态蛋白质相互作用网络的构建以建模细胞水平生物过程。文献[21]运用 3-sigma 动态阈值方法[22]确定蛋白质活性时刻后，构建了时空活跃蛋白质相互作用网络。本着如下猜想：基于动态蛋白质相互作用网络挖掘蛋白质复合物和功能模块将比基于静态蛋白质相互作用网络更具优势。文献[23-26]分别在构建各自的动态蛋白质相互作用网络后，运用 MCL 算法检测蛋白质复合物；文献[27]在构建动态蛋白质相互作用网络后，将萤火虫算法 FA 分别与算法 MCL 及其变型 R-MCL 和 SR-MCL 融合，提出算法 F-MCL、FR-MCL 和 FSR-MCL 以检测动态蛋白质相互作用网络中的蛋白质功能模块。

马尔可夫聚类以模拟网络流的随机游走方式，对网络转移概率矩阵交替地执行扩展(Expansion)和膨胀(Inflation)操作，以强化稠密连接区域的网络流，弱化稀疏连接区域的网络流，从而实现网络流随机游走概率的再分配与分化，最终根据不同的概率完成网络的划分并达到聚类的目的。因此，马尔可夫聚类算法需要对以矩阵形式表示的网络迭代地执行扩展和膨胀两个操作，以实现网络模块挖掘。由于涉及大量的矩阵运算，MCL 算法需要消耗大量时间，因此文献[28]基于 MPI 并行编程模型，提出了马尔可夫聚类的并行化算法以解决大规模生物网络聚类问题，然而该方法仅针对静态蛋白质网络讨论 MCL 算法中扩展和膨胀操作的并行化。MCL 算法并行化的核心是矩阵乘法并行化，文献[29]对几种矩阵乘的并行算法进行对比分析，分析结果表明，DNS 算法[30]具有最好的时间复杂度。

本研究首先按照 DNS 算法实现了基于 GPU 的矩阵乘，从而实现 MCL 算法中扩展操作的 GPU 加速；然后针对静态蛋白质网络中蛋白质复合物和功能模块这两种功能子结构，讨论了两种动态蛋白质相互作用网络的构建；随后将基于 GPU 矩阵乘的 MCL 算法分别用于设计挖掘蛋白质复合物和功能模块的算法；最后分析了算法的时空复杂度和并行性。

2 相关概念

2.1 R可靠蛋白质相互作用网络

给定一个带可靠性得分的蛋白质相互作用数据集 $PPIS=(PNS, PPS, Score)$，$PNS=\{1, ..., M\}$为蛋白质结点集，$PPS=\{(i, j)|i, j\in PNS\}$为蛋白质相互作用集，相互作用$(i, j)$的可靠性得分为 $Score(i, j)\in\{1, ..., 999\}$[31]。另有一个蛋白质相互作用数据集 $PPI=(PN, PP)$，$PN=\{1, ..., N\}$为蛋白质结点集，$PP=\{(i, j)|i, j\in PN\}$为蛋白质相互作用集。于是用 $PPIS$ 的可靠性得分按式(1)给 PPI 打分得到 $PPIs=(PN, PP, s)$。

$$s(i,j)=\begin{cases}0 & , \ if \quad (i,j)\notin PP\\ 1 & , \ if \quad (i,j)\in PP, \ (i,j)\notin PPS\\ Score(i,j) & , \ if \quad (i,j)\in PP, \ (i,j)\in PPS\end{cases} \tag{1}$$

进一步，本研究选取 R 为可靠性得分阈值，并构造 R 可靠蛋白质相互作用网络 $PINs|R=(PN, PP|R, s)$，其中 $PP|R=\{(i, j)|s(i, j)\geq R, i, j\in PN\}$。设 AR 为 R 可靠蛋白质相互作用网络 $PINs|R$ 的邻接矩阵，则邻接矩阵 AR 的元素 $ar_{i,j}$ 按式(2)计算，其中 $R\in\{1, ..., 999\}$[31]。

$$ar_{i,j}=\begin{cases}1 & , \ if \quad s(i,j)\geq R\\ 0 & , \ otherwise\end{cases} \tag{2}$$

2.2 蛋白质定位数据

假定有 N 个蛋白质在 L 个亚细胞区室的定位情况，则蛋白质定位数据可用 PCL：$\{0, 1\}^{N\times L}$ 表示，记 $PCL=\{pcl_{i,c}\}$，$i=1, ..., N$，$c=1, ..., L$。在整个细胞周期中，若蛋白质 i 曾在亚细胞区室 c 出现过，则 $pcl_{i,c}=1$，否则 $pcl_{i,c}=0$。于是，$pcl_{i,c}$ 按式(3)定义。

$$pcl_{i,c}=\begin{cases}1 & , \ \begin{array}{l}Protein\ i\ is\ localiæd\\ in\ compartment\ c.\end{array}\\ 0 & , \ otherwise\end{cases} \tag{3}$$

2.3 基因表达数据

假设有 N 个基因在 T 个时刻经归一化的表达值，则基因表达数据可用矩阵 GEV: $R^{N\times T}$ 表示，记 $GEV=\{gev_{i,t}\}$，$i=1, ..., N$，$t=1, ..., T$。

（1） 基因编码蛋白质活跃

假设基因 i 及其对应的基因表达数据为 $gev_{i,t}$，$t=1, ..., T$，令 $\overline{gev_i}=\frac{1}{T}\sum_{t=1}^{T}gev_{i,t}$ 表示基因 i 的基因表达

平均值，于是用 $ap_{i,t}$ 表示基因 i 编码的蛋白质在时刻 t 的活跃情况。当时刻 t 基因 i 的表达值大于等于平均值时，$ap_{i,t}$=1 表示基因 i 编码的蛋白质在时刻 t 活跃，否则 $ap_{i,t}$=0 表示基因 i 编码的蛋白质在时刻 t 不活跃，于是，$ap_{i,t}$ 按式(4)定义。

$$ap_{i,t}=\begin{cases}1 & , \quad if \quad gev_{i,t}\geq\overline{gev_i}\\ 0 & , \quad otherwise\end{cases} \tag{4}$$

（2） 基因表达相关性

假定有基因 i 和 j，它们对应的基因表达数据为 $gev_{i,t}$ 和 $gev_{j,t}$，t=1, ..., T。基因 i 和 j 的表达相关性可用皮尔森相关系数(*Pearson correlation coefficient*，*pcc*)度量，于是基因 i 和 j 的皮尔森相关系数可按式(5)计算[32]，其中 $\overline{gev_i}$ 和 $\overline{gev_j}$ 分别表示基因 i 和 j 的基因表达平均值。

$$pcc(i,j)=\frac{\sum_{t=1}^{T}(gev(i,t)-\overline{gev_i})(gev(j,t)-\overline{gev_j})}{\sqrt{\sum_{t=1}^{T}(gev(i,t)-\overline{gev_i})^2}\sqrt{\sum_{t=1}^{T}(gev(j,t)-\overline{gev_j})^2}} \tag{5}$$

式(5)量化了两个基因在整个采样周期之间的表达相关性，而基因编码的蛋白质相互作用则必然是相应基因在相同时刻短期内共同表达，因此需要按式(6)量化以某时刻 t 为中心短期内（前后共 5 个时间点）两个基因的表达相关性。

$$pcc(i,j)^t=\frac{\sum_{k=t-2}^{t+2}(gev(i,k)-\overline{gev_i^t})(gev(j,k)-\overline{gev_j^t})}{\sqrt{\sum_{k=t-2}^{t+2}(gev(i,k)-\overline{gev_i^t})^2}\sqrt{\sum_{k=t-2}^{t+2}(gev(j,k)-\overline{gev_j^t})^2}} \tag{6}$$

对于连续时间点活跃的蛋白质，其编码基因 i 应该在连续时间点上均有显著表达，因此可按式(7)计算该编码基因 i 在连续时间点上短期基因表达自相关性，其中，$\overline{gev_i^t}=\frac{1}{5}\sum_{k=t-2}^{t+2}gev_{i,k}$，$\overline{gev_j^t}=\frac{1}{5}\sum_{k=t-2}^{t+2}gev_{j,k}$，$\overline{gev_i^{t+1}}=\frac{1}{5}\sum_{k=t-1}^{t+3}gev_{i,k}$。

$$pcc(i,i)^t=\frac{\sum_{k=t-2}^{t+2}(gev(i,k)-\overline{gev_i^t})(gev(i,k+1)-\overline{gev_i^{t+1}})}{\sqrt{\sum_{k=t-2}^{t+2}(gev(i,k)-\overline{gev_i^t})^2}\sqrt{\sum_{k=t-1}^{t+3}(gev(i,k)-\overline{gev_i^{t+1}})^2}} \tag{7}$$

3 关键技术

3.1 马尔可夫聚类 MCL

马尔可夫聚类过程是对转移概率矩阵迭代地执行扩展和膨胀操作，直至矩阵收敛，可见扩展操作和膨胀操作构成了它的基本操作。

扩展操作的实质是执行矩阵幂乘运算。设有方阵 P，则方阵 P 的 n 次幂运算为：$P^n=P^{n-1}\cdot P$。本研究通过循环调用矩阵乘实现扩展操作，并将整个扩展操作简记为：$\mathscr{E}$:$P\rightarrow P'$。

膨胀操作实质是对矩阵每个元素执行 r 次幂后在列方向上进行归一化。设矩阵 P'：$R^{N\times N}$ 和非负实数 r，经膨胀操作后的矩阵为 P''，P_{ij}' 和 P_{ij}'' 分别表示矩阵 P' 和 P''的元素，于是矩阵元素 P_{ij}'' 按式(8)计算。本研究将整个膨胀操作简记为：$\mathscr{F}$:$P'\rightarrow P''$。

$$P_{ij}''=\frac{(P_{ij}')^r}{\sum_{k=1}^{N}(P_{kj}')^r} \tag{8}$$

3.2 基于 GPU 矩阵乘的 MCL

马尔可夫聚类过程中扩展操作的核心是矩阵幂乘运算，是最为耗时的运算，利用 GPU 实现矩阵乘对于加速 MCL 过程效果显著。本研究 GPU 加速 MCL 的实质是按照 DNS 算法编写 GPU 核函数实现矩阵乘法，为此本研究设计模块 mmgpu 实现基于 GPU 的矩阵乘。考虑到 GPU 存储容量不足的情况，本研究还设计了模块 SBBMM，模块 SBBMM 按分块矩阵乘的方式自适应地选择循环调用模块 mmgpu 实现矩阵乘法。图 1 为模块 SBBMM 和模块 mmgpu 的程序流程图。

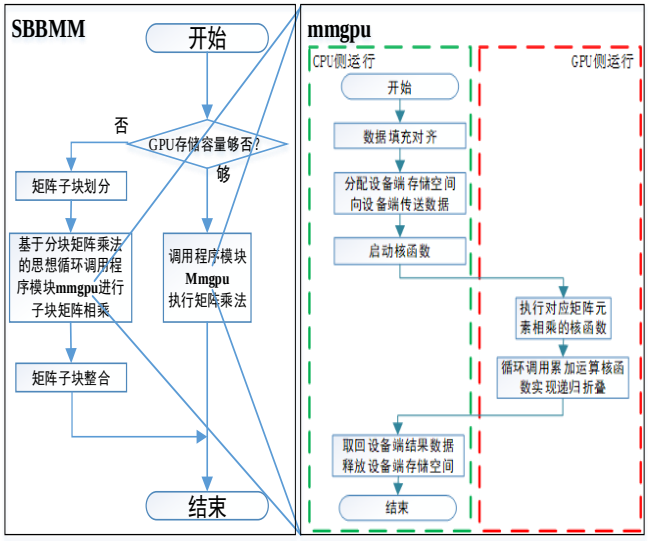

图 1 模块 SBBMM 和模块 mmgpu 的程序流程图

3.3 动态蛋白质相互作用网络

（1） 时空动态蛋白质相互作用网络

蛋白质复合物是特定时间在特定细胞器中形成的，也就是说，蛋白质复合物是在特定时空中形成的。为了挖掘蛋白质复合物，本研究将基因表达数据、蛋

白质亚细胞定位数据和 PPI 数据融合以构造时空动态蛋白质相互作用网络 STDPINs（见图 2）。

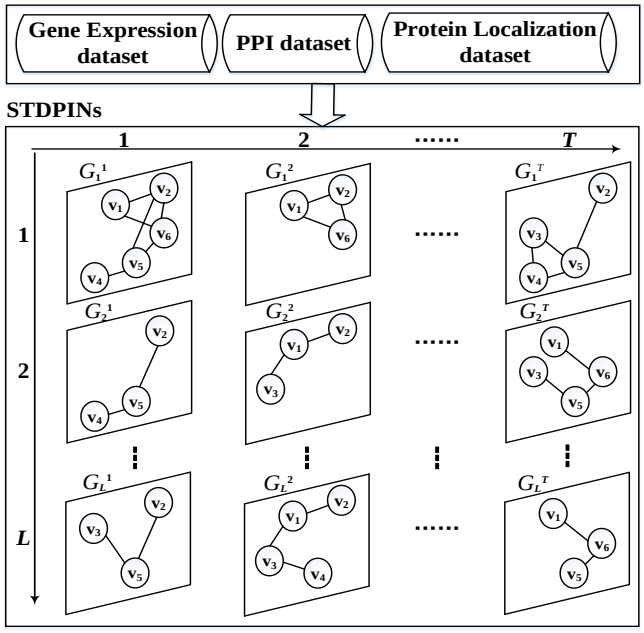

图 2 时空动态蛋白质相互作用网络 STDPINs

给定 $G=(V,E)=\{G_1^1,G_2^1,...,G_L^1,G_1^2,G_2^2,...,G_L^2,...,G_1^t,G_2^t,...,G_L^t,...,G_1^T,G_2^T,...,G_L^T\}$，其中 $t(=1,...,T)$表示采样时刻，$c(=1,...,L)$表示亚细胞区室，$G_c^t=(V_c^t,E_c^t)$，V_c^t 为时刻 t 亚细胞区室 c 内的活跃蛋白质集，E_c^t 为时刻 t 亚细胞区室 c 内的活跃蛋白质相互作用集，$V=\bigcup_{t=1}^{T}\bigcup_{c=1}^{L}V_c^t$，$E=\bigcup_{t=1}^{T}\bigcup_{c=1}^{L}E_c^t$，$T$ 为采样总数，L 为亚细胞区室总数。G_c^t表示时刻 t 亚细胞区室 c 内的活跃蛋白质相互作用构成的网络，可见时空动态蛋白质相互作用网络可以用总数为 $T\times L$ 的一组网络表示。

假定 A_c^t 为 G_c^t 的邻接矩阵，于是邻接矩阵元素 $a(i,j)_c^t$ 可按式(9)计算时刻 t 亚细胞区室 c 中的活跃蛋白质 i 和 j 之间的短期基因表达的皮尔森相关系数来获得，其中 $i,j=1,...,N$，$t=1,...,T$，$c=1,...,L$。

$$a(i,j)_c^t=ar_{i,j}\times ap_{i,t}\times pcl_{i,c}\times pcc(i,j)^t\times pcl_{j,c}\times ap_{j,t} \quad (9)$$

若用单一矩阵 A 来表示整个时空动态蛋白质相互作用网络的邻接矩阵，则 A 可表示为式(10)。

$$A=\begin{pmatrix} A_1^1 & A_1^2 & \cdots & \cdots & A_1^T \\ A_2^1 & A_2^2 & \cdots & \cdots & A_2^T \\ \vdots & \vdots & \ddots & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_L^1 & A_L^2 & \cdots & \cdots & A_L^T \end{pmatrix} \quad (10)$$

由于形成于特定时空的复合物的相互作用蛋白质的表达相关性信息仅局限于特定的 A_c^t 中，因此在挖掘时空动态蛋白质相互作用网络中的复合物时，只需分别对 A_c^t 调用 MCL 算法进行聚类即可。

（3） 时序动态蛋白质相互作用网络链

生物过程中的蛋白质相互作用是时序地发生的，也就是说，生物过程中时序的蛋白质相互作用是跨多个时间点发生的。因此一个特定生物过程对应的功能模块必然存在于跨时间的动态蛋白质相互作用网络中。为了挖掘蛋白质功能模块，本研究将基因表达数据和 PPI 数据融合以构造时序动态蛋白质相互作用网络链 TDPINC（见图 3）。

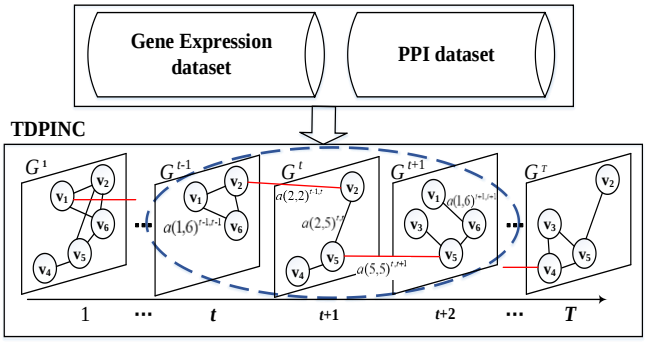

图 3 时序动态蛋白质相互作用网络链 TDPINC

给定时序动态蛋白质相互作用网络链 $G=(V,E)=\{G^1,G^2,...,G^t,...,G^T\}$，其中子图 $G^t=(V^t,E^t)$，$V^t=\{i|ap_{i,t}=1\}$ 表示时刻 t 活跃蛋白质结点集，$V=\bigcup_{t=1}^{T}V^t$ 表示任意时刻活跃蛋白质结点集，$E^t=\{(i,j)|s(i,j)\geq R,\ ap_{i,t}\times ap_{j,t}=1\}$ 表示时刻 t 活跃蛋白质 R 可靠相互作用集合，E^t 中的元素是无向边，也称为网内边，$(<v^t,v^{t+1}>|ap_{v,t}\times ap_{v,t+1}=1)$ 表示由时刻 t 活跃且时刻 $t+1$ 也活跃的结点 v 构成的跨时间点的有向自边，也称为跨网边，于是 $E=\left(\bigcup_{t=1}^{T}E^t\right)\cup\left(\bigcup_{t=1}^{T-1}\{(<v^t,v^{t+1}>|ap_{v,t}\times ap_{v,t+1}=1)\}\right)$，$i,j,v=1,...,N$，$t=1,...,T$。

实质上，E 是由所有的网内边和跨网边构成的集合。由于存在跨网边，这样的一组有序图集构成了一个逻辑上连接的网络链（如图 3 所示），图中红色边即为跨网边。如果任意两个相邻子图间均有跨网边，则所有子图便连成一个逻辑整体。E 中所有边的权值可按式(11)计算，其中 $i,j=1,...,N$，$t,t_1,t_2=1,...,T$。

$$w(i,j)^{t_1,t_2}=\begin{cases} ar_{i,j}\times ap_{i,t}\times pcc(i,j)^t\times ap_{j,t} & ,\ i\neq j,t=t_1=t_2 \\ ap_{i,t}\times pcc(i,i)^t\times ap_{i,t+1} & ,\ i=j,t=t_1=t_2-1 \end{cases} \quad (11)$$

假设 $B^{t,t}$ 为 G^t 的邻接矩阵，其中 $t=1,...,T$，于是邻接矩阵 $B^{t,t}$ 的元素 $b(i,j)^{t,t}$ 可按式(11)取 $w(i,j)^{t,t}$ 的值。类似地，用跨网矩阵 $B^{t,t+1}$，$t=1,...,T$-1 反映跨网边，

于是跨网矩阵 $B^{t,t+1}$ 的元素 $b(i,j)^{t,t+1}$ 可按式(11)取 $w(i,j)^{t,t+1}$ 的值。

既然时序动态蛋白质相互作用网络链 G 是一个逻辑整体，因此也可用单一矩阵 B 表示，于是 B 可按式(12)表示。

$$B=\begin{pmatrix} B^{1,1} & B^{1,2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & B^{2,2} & B^{2,3} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \ddots & \ddots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & B^{t,t} & B^{t,t+1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & B^{t+1,t+1} & B^{t+1,t+2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \ddots & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & B^{T-1,T-1} & B^{T-1,T} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & B^{T,T} \end{pmatrix} \tag{12}$$

易见，B 是一个上三角分块矩阵，而且非零子块分布在正对角线和次对角线子块位置，其中正对角线上的子块矩阵均是对称矩阵，而次对角线上的子块矩阵是非对称的，它反映了时序有向性。

涉及特定生物过程的蛋白质功能模块镶嵌在细胞周期的时序动态蛋白质相互作用网络链中，图 3 中椭圆圈内跨网连接的结点集有可能对应一个蛋白质功能模块。因此，构建单一矩阵 B 能保持蛋白质功能模块在时序动态蛋白质网络链 G 中的逻辑完整性。特别指出，在挖掘时序动态蛋白质相互作用网络链中的功能模块时，需要对 B 调用 MCL 算法进行聚类。

4 算法设计

4.1 简介

本研究针对蛋白质复合物和功能模块的挖掘，利用基于 GPU 矩阵乘分别设计用于时空动态蛋白质相互作用网络 STDPINs 的算法 MCL-IPC-STDPINs 和用于时序动态动态蛋白质相互作用网络链 TDPINC 的算法 SBBMCL-IFM-TDPINC。两个算法都需要进行动态蛋白质网络的构造，结束前都需要进行除重处理。由于所构造动态蛋白质相互作用网络的不同，两个算法的主体部分有明显的差别，下面对两个算法分别加以描述。

4.2 算法描述

（1） 算法 MCL-IPC-STDPINs

图 4 为算法 MCL-IPC-STDPINs 的程序流程图。

算法 MCL-IPC-STDPINs 的伪代码见算法 1。在算法 1 中，第 4-5 行完成一个特定时空动态蛋白质相互作用网络的构造并从中挖掘蛋白质复合物，第 2-7 行实现 $T\times L$ 个时空动态蛋白质相互作用网络的构造并从中挖掘蛋白质复合物，第 8 行进行蛋白质簇的除重处理，第 9 行输出的结果即为蛋白质复合物。易见，算法 1 每构造一个 G_c^t 就调用基于 GPU 矩阵乘 MCL 对矩阵 A_c^t 进行蛋白质复合物挖掘。

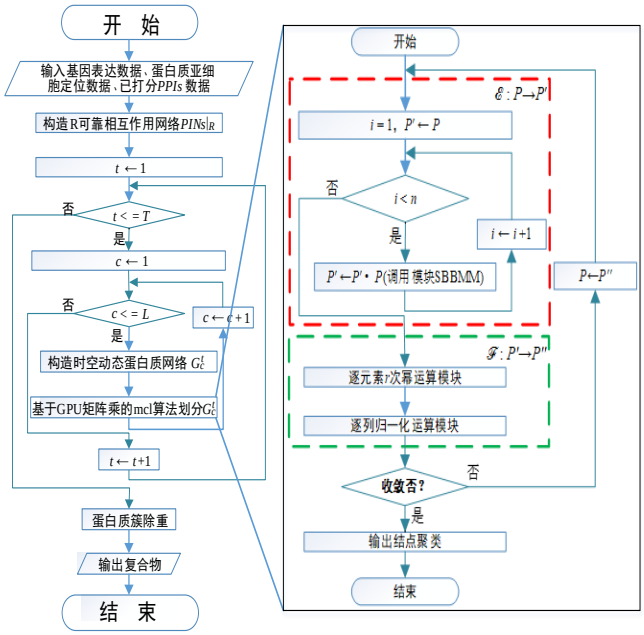

图 4 算法 MCL-IPC-STDPINs 的程序流程图

基于GPU矩阵乘的马尔可夫聚类模块 mcl 的伪代码见算法 2。在算法 2 中，第 1 行执行 P 的 1 次幂运算，第 2-4 行执行矩阵累乘($P'\leftarrow P'\cdot P$)以完成 n 次幂运算，因此整个 1-4 行实现了扩展操作。算法 2 第 5-6 行的两个步骤是实现膨胀操作。

算法 1： 算法 MCL-IPC-STDPINs

输入： 基因表达数据，蛋白质定位数据，已打分 *PPIs* 数据。

输出： 蛋白质复合物集

Begin

1. 构造 R 可靠相互作用网络 $PINs|_R$
2. **For** t=1 to T
3. **For** c=1 to L
4. 构造时空动态蛋白质相互作用网络 G_c^t
5. 调用基于 GPU 矩阵乘 mcl 模块
6. **Endfor**
7. **Endfor**
8. 蛋白质簇除重
9. 输出蛋白质复合物

End

（2） 算法 SBBMCL-IFM-TDPINC

在算法 SBBMCL-IFM-TDPINC 中 MCL 操作的矩阵是形如式(12)的上三角分块矩阵 B，为此本研究利用上三角分块矩阵的特点，按如下方式优化设计 MCL 的流程。

MCL 的扩展操作：$B^n = B^{n-1} \cdot B$，可表示成递推式 $\begin{cases} C^1 = B \\ C^{k+1} = C^k \cdot B, \quad k = 1,...,n-1 \end{cases}$，此处 k 表示迭代的轮次。

由于 B 是分块矩阵，因此 C^{k+1} 的计算可以逐个子块进行，于是 C^{k+1} 的第 I 行第 J 列子块 C_{IJ}^{k+1} 可按式(13)计算。

算法 2：	算法 mcl
输入：	矩阵 A_c^t
输出：	结点聚类

Begin

1. $P' \leftarrow P$
2. **For** i=1 to n-1
3. 调用 SBBMM 模块
4. **Endfor**
5. 逐元素计算 r 次方，并在列方向累加。
6. 逐元素归一化
7. 若 P''未收敛，则 $P \leftarrow P''$，然后转 1。
8. 输出结点聚类。

End

$$C_{IJ}^{k+1} = \sum_{K=I}^{J} C_{IK}^{k} B^{K,J}, \quad 1 \le I \le J \le T, \quad k = 1,...,n-1 \qquad (13)$$

为了加快收敛速度，矩阵 C 的子块的计算过程可以按递推式(14)进行。

$$C_{IJ}^{k} = \begin{cases} C_{II}^{k} B^{I,I}, & 1 \le I = J \le T \\ \sum_{K=I}^{J} C_{IK}^{k} B^{K,J}, & 1 \le I < J \le T \end{cases}, k = 1,...,n-1 \qquad (14)$$

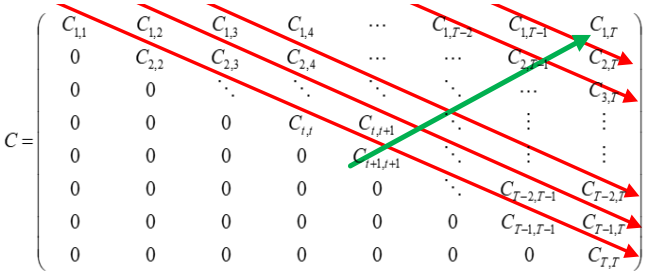

图 5 分块矩阵乘的一种子块矩阵计算顺序

图 5 给出 C_{IJ}^{k} 的一种计算过程，它可沿对角线方向，从主对角线向右上角逐个子块计算。红色箭头反映在同一斜线上子块矩阵的一种计算顺序，绿色箭头反映斜线子块矩阵计算的推进顺序。

图 6 为算法 SBBMCL-IFM-TDPINC 程序流程图。

算法 SBBMCL-IFM-TDPINC 的伪代码见算法 3。

在算法 3 中，第 2-4 行构造正对角线子块，第 5-7 行构造次对角线子块，整个 B 矩阵构造完成后即可调用基于分块矩阵的 MCL 算法进行蛋白质功能模块挖掘。基于分块矩阵的 MCL 算法伪代码见算法 4。

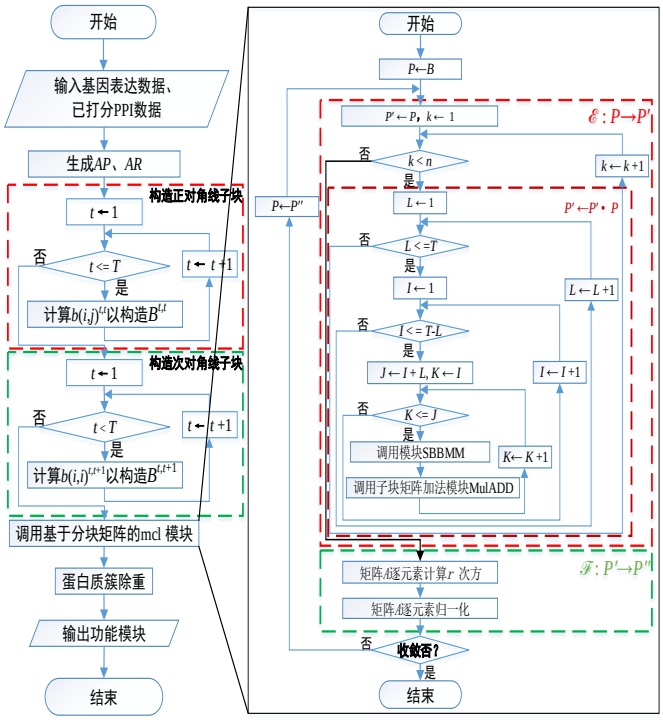

图 6 算法 SBBMCL-IFM-TDPINC 的程序流程图

在算法 4 中，第 2 行完成矩阵 B 的 1 次幂运算，第 7-9 行完成 1 个子块矩阵的计算，第 5-10 行则完成 1 条斜线子块矩阵的计算，第 4-11 行完成矩阵 B 的 1 次累乘运算，第 3-12 行则完成矩阵 B 的 n 次幂运算，因此第 2-12 行完成 1 次扩展操作；第 13-14 行对应完成膨胀操作。

算法 3：	算法 SBBMCL-IFM-TDPINC
输入：	基因表达数据，已打分 *PPI* 数据。
输出：	蛋白质功能模块集

Begin

1. 生成活跃矩阵 *AP* 和 *R* 可靠相互作用矩阵 *AR*
2. **For** t=1 to T
3. 计算 $b(i,j)^{t,t}$ 以构造 $B^{t,t}$
4. **Endfor**
5. **For** t=1 to T-1
6. 计算 $b(i,i)^{t,t+1}$ 以构造 $B^{t,t+1}$
7. **Endfor**
8. 调用基于分块矩阵的 MCL 算法
9. 蛋白质簇除重
10. 输出蛋白质功能模块

End

算法 4：基于分块矩阵 MCL 算法

输入：矩阵 B

输出：结点聚类

Begin

1. 将 B 转换为转移概率矩阵 P
2. $P' \leftarrow P$
3. **For** k=1 to n-1
4. **For** L=1 to T
5. **For** I=1 to T-L+1
6. $J \leftarrow I+L-1$
7. **For** K=I to J
8. 调用模块 SBBMM 和 MulADD。
9. **Endfor**
10. **Endfor**
11. **Endfor**
12. **Endfor**
13. 逐元素计算 r 次方，并在列方向累加。
14. 逐元素归一化
15. 若 P''未收敛，则 $P \leftarrow P''$，然后转 2。
16. 输出结点聚类

End

5 分析讨论

本文根据矩阵并行算法 DNS，设计实现了基于 GPU 的矩阵乘法，为马尔可夫聚类 MCL 算法的扩展操作提供显著加速的可行方案，从而加速 MCL 算法。

本研究将基于 GPU 矩阵乘的 MCL 算法用于动态蛋白质相互作用网络以挖掘蛋白质复合物和功能模块。针对蛋白质复合物的挖掘，本研究构造了时空动态蛋白质相互作用网络并提出了算法 MCL-IPC-STDPINs；而针对蛋白质功能模块的挖掘，本研究构造了时序动态蛋白质相互作用网络链并提出了算法 SBBMCL-IFM-TDPINC。本研究以 GPU 矩阵乘为基础，因此在本节的讨论是以 GPU 矩阵乘为单位，对于 GPU 矩阵乘内部的时空复杂度和并行性本文不做分析。

5.1 时空复杂度分析

下面主要对本研究提出的两个算法在主机侧时空开销方面进行复杂性分析。

由算法 1 易见，$T \times L$ 个时空动态蛋白质相互作用网络通过双重循环被串行地构造并实施马尔可夫聚类，而存储空间开销取决于一个蛋白质相互作用网络的规模，所以对于算法 MCL-IPC-STDPINs 而言，构造动态蛋白质相互作用网络的时间复杂度为 $O(T \times L \times N^2)$，其马尔可夫聚类过程的时间复杂度为 $O(T \times L)$，而空间复杂度则为 $O(N^2)$。

通过分析算法 3，算法 SBBMCL-IFM-TDPINC 构造动态蛋白质相互作用网络的时间复杂度为 $O(T \times N^2)$；通过分析算法 4，算法 SBBMCL-IFM-TDPINC 的马尔可夫聚类过程的时间复杂度为 $O(T^2)$。本文用式(12)的矩阵 B 来表示时序动态蛋白质相互作用网络链，所以算法 SBBMCL-IFM-TDPINC 的空间复杂度则为 $O(T \times N^2+(T-1) \times N^2+...+2 \times N^2+N^2)=O((T \times N)^2)$。

5.2 并行性分析

由算法 1 可见，算法 MCL-IPC-STDPINs 循环地对不同时空的蛋白质相互作用网络进行马尔可夫聚类，由于不同时空的蛋白质相互作用网络逻辑上相互独立，因此若有足够的 GPU 资源，不同时空蛋白质相互作用网络的马尔可夫聚类过程均可并行进行。

通过分析算法 4，算法 SBBMCL-IFM-TDPINC 的扩展操作的一次累乘过程存在大量的并行性。在图 5 中，红色箭头反映了在同一斜线上的子块矩阵的一种计算顺序。事实上，对式(14)分析可知，在同一斜线上的子块矩阵计算彼此不存在相关，是完全可以并行执行的，而且同一斜线上子块矩阵计算涉及的矩阵乘法也都是互不相关的，因此在有足够的 GPU 资源和内存空间的情况下，涉及的所有矩阵乘法均可并行执行。而图 5 中绿色箭头展示的方向是不同斜线子块矩阵的计算顺序。

此外，MCL 算法的膨胀操作也具有明显的并行性，也可采用 GPU 进行加速。总之，为充分利用已有 GPU 资源，可以采用分组方式均衡分配马尔可夫聚类过程，最大限度地利用系统资源实现动态蛋白质相互作用网络功能子结构的快速挖掘。

6 结束语

本文研究首先实现基于 GPU 的矩阵乘并行算法 DNS，然后研究并提出基于 GPU 矩阵乘的 MCL 算法从动态蛋白质相互作用网络中挖掘功能子结构的方法，最后分析所提出方法的时空复杂度和并行性，为有效挖掘动态蛋白质相互作用网络中功能子结构提供了解决方案。

参 考 文 献

[1] SPIRIN V, MIRNY L. Protein complexes and functional modules in molecular networks[J]. Proceedings of the National Academy of Sciences of the United States of America, 2003, 100(21):12123-12128

[2] 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

[3] HO Y, GRUHLER A, HEIBUT A, et al. Systematic identification of protein complexes in Saccharomyces

cerevisiae by mass spectrometry[J]. Nature, 2002, 415(6868): 180-183

[4] ZHU H，BILGIN M，BANGHA m R, et a1．Global analysis of protein activities using proteome chips[J]. Science, 2001, 293(5537): 2101-2105

[5] VAN DONGEN S M. Graph clustering by flow simulation[D]. Utrecht: University of Utrecht, 2000

[6] BROHEE S, VAN HELDEN J. Evaluation of clustering algorithms for protein-protein interaction networks[J]. BMC Bioinformatics, 2006, 7:488

[7] VLASBLOM J, WODAK S J. Markov clustering versus affinity propagation for the partitioning of protein interaction graphs[J]. BMC Bioinformatics, 2009, 10:99

[8] SATULURI V, PARTHASARATHY S, Scalable graph clustering using stochastic flows: Applications to community discovery[C]. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD'09, New York, USA, 2009, 737-746

[9] SATULURI V, PARTHASARATHY S, UCAR D. Markov clustering of protein interaction networks with improved balance and scalability[C]. In: Proceedings of the First ACM International Conference on Bioinformatics and Computational Biology-BCB, Niagara Falls, NY, USA, 2010, 247-256

[10] SHIH Y K, PARTHASARATHY S, Identifying functional modules in interaction networks through overlapping Markov clustering[J]. Bioinformatics, 2012, 28: 473-479

[11] Lei X J, Wang F, Wu F X, et al. Protein complex identification through Markov clustering with firefly algorithm on dynamic protein–protein interaction networks[J]. Information Sciences, 2016, 329: 303-316

[12] SRIHARI S, NING K, LEONG H W. Refining Markov Clustering for complex detection by incorporating core-attachment structure[J]. Geome Information, 2009, 23(1):159-168

[13] SRIHARI S, NING K, LEONG H W. MCL-CAw: a refinement of MCL for detecting yeast complexes from weighted PPI networks by incorporating core-attachment structure[J]. BMC Bioinformatics, 2010,11: 504

[14] ENRIGHT A J, VAN DONGEN S, OUZOUNIS C A. An Efficient Algorithm for Large-Scale Detection of Protein Families[J]. Nucleic Acid Research, 2002, 30(7): 1575-1584

[15] GU L, HAN Y, WANG C, et al. Module overlapping structure detection in PPI using an improved link similarity-based Markov clustering algorithm[J]. Neural Computing and Applications, 2018, 31(5):1481-1490

[16] HWANG W, CHO Y R, ZHANG A, et al. A Novel Functional Module Detection Algorithm for Protein-Protein Interaction Networks[J]. Algorithms for Molecular Biology, 2006, 1:24

[17] HWANG W, CHO Y R, ZHANG A, et al. CASCADE: A Novel Quasi All Paths-Based Network Analysis Algorithm for Clustering Biological Interactions[J]. BMC Bioinformatics, 2008, 9: 64

[18] INOUE K, LI W, KURATA H. Diffusion Model Based Spectral Clustering for Protein-Protein Interaction Networks[J]. PLos ONE, 2010, 5(9): e12623

[19] LEVY E D, PEREIRA J B. Evolution and Dynamics of Protein Interactions and Networks[J]. Current Opinion in Structure Biology, 2008, 18(3): 349-357

[20] PRZYTYCKA T M, SINGH M, SLONIM D K. Toward the Dynamic Interactome: It's About Time[J]. Briefings in Bioinformatics, 2010, 11: 15-29

[21] 李 敏, 孟祥茂. 动态蛋白质网络的构建、分析及应用研究进展[J]. 计算机研究与发展, 2017, 54(6): 1281-1299

[22] WANG J X, PENG X Q, LI M, et al. Construction and application of dynamic protein interaction network based on time course gene expression data[J]. Proteomic, 2013, 13(2):301-312

[23] TANG X, WANG J, LIU B, et al. A Comparison of the Functional Modules Identified from Time Course and Static PPI Network Data[J]. BMC Bioinformatics, 2011, 12: 339

[24] WANG J X, PENG X Q, LI M, et al. Construction and application of dynamic protein interaction network based on time course gene expression data[J]. Proteomic, 2013, 13(2):301-312

[25] SHEN X J, YI L, JIANG X P, et al. Mining Temporal Protein Complex Based on the Dynamic PIN Weighted with Connected Affinity and Gene Co-Expression[J]. PLoS ONE, 2016, 11(4): e0153967

[26] XIAO Q H, WANG J X, PENG X Q, et al. Detecting protein complexes from active protein interaction networks constructed with dynamic gene expression profiles[J]. Proteome Science, 2013, 11(Suppl 1):S20

[27] LEI X J, WANG F, WU F X, et al. Detecting functional modules in dynamic protein-protein interaction networks using Markov Clustering and Firefly Algorithm[C]. In: Proc. of 2014 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2014, 75-81

[28] 孙佳敏，朱嘉富，杨伏长等. 大规模生物网络马尔可夫聚类的并行化算法[J]. 计算机应用，2019，39(1):66-71

[29] 陈鹏，樊小超. 几种矩阵乘并行算法的对比分析[J]. 新疆师范大学学报(自然科学版)，2012，31(3)，5-10

[30] Dekel Eliezer, Nassimi David, Sahni Sartaj. Parallel matrix and graph algorithms[J]. SIAMJ. Comput. 1981, 4(10):657-675

[31] STRING http://string-db.org/cgi/download_page.pl [DB]

[32] Goh KI, Cusick ME, Valle D, et al. The human disease network[J]. Proceedings of the National Academy of Sciences of the United States of America, 2007, 104 (21): 8685-8690