

文章编号:1673-1689(2008)05-0086-05

代谢网络自动绘制的快速网格布局算法

何胜^{1,2,3}, 梅娟^{1,2}, 石贵阳^{1,2}, 王正祥^{1,2}, 李炜疆^{*1,2}

(1. 江南大学生物工程学院, 江苏无锡 214122; 2. 江南大学工业生物技术教育部重点实验室, 江苏无锡 214122; 3. 江苏技术师范学院计算机科学与工程学院, 江苏常州 213001)

摘要: 描述众多代谢物之间的拓扑关系的代谢网络, 是抽象的高维数据。为了帮助人们分析这些复杂的数据, 需要开发高效的可视化算法。近年来日益引起关注的网格布局算法在代谢网络自动绘图中显示了很好的特性, 其面临的一个主要问题是如何有效降低计算量以满足快速、准实时网络绘图的需求。作者提出了一种快速网格布局算法, 采用邻域试探和扰动再优化的全局搜索策略, 能够数秒内产生高质量的典型代谢网络布局, 适用于更广泛的代谢网络可视化分析应用。

关键词: 代谢网络; 网格布局算法; 绘图; 可视化

中图分类号: Q 811.4

文献标识码: A

A Fast Grid Layout Algorithm for Automatic Drawing Complex Metabolic Network

HE Sheng^{1,2,3}, MEI Juan^{1,2}, SHI Gui-yang^{1,2}, WANG Zheng-xiang^{1,2}, LI Wei-jiang^{*1,2}

(1. School of Biotechnology, Jiangnan University, Wuxi 214122, China; 2. Key Laboratory of Industrial Biotechnology, Ministry of Education, Wuxi 214123, China; 3. School of Computer Science, Jiangsu Teachers University of Technology, Changzhou 213001, China)

Abstract: Describing topological relationships between large amounts of metabolites, the metabolic networks are abstract, high-dimensional data. Efficient visualization algorithm is necessary to aid human analyses of such complex data. Developed in recent years, grid layout exhibits good features in automatic drawing of metabolic networks and thus attracts increasing interests. A main challenge of grid layout algorithms is to decrease the high computational cost to satisfy the requirement of fast, nearly real-time network drawing. A fast grid layout algorithm using neighborhood-test search and re-optimization-after-perturbation strategies were proposed. The new algorithm could produce high-quality layouts in a few seconds for typical large-scale metabolic networks and was suitable for more versatile applications for visual analysis of metabolic networks.

Key words: metabolic network; grid layout algorithm; graph drawing; visualization

收稿日期: 2008-07-08.

基金项目: 国家 863 计划项目(2006AA020204); 江苏技术师范学院青年科研基金项目(KYY06081).

作者简介: 何胜(1971-), 男, 安徽枞阳人, 发酵工程博士研究生.

* 通讯作者: 李炜疆(1964-), 男, 陕西榆林人, 理学博士, 教授, 博士生导师, 主要从事生物信息学研究. Email: wjlee01@gmail.com.

代谢网络的计算机辅助建模是系统生物学领域的重要研究课题之一。在后基因组时代,基因组学研究和高通量实验技术推动着网络数据的迅速增长,极大地促进了整体代谢网络的研究,从全局观点分析代谢途径之间的相互关系,从而更准确地了解细胞的代谢过程。整体代谢网络由众多代谢途径整合交织而成,涉及大量的节点(代谢物)和它们之间的关系(反应,用边表示),从信息学角度看,是复杂的高维数据结构。用二维图形表示网络结构,将这种复杂抽象的数据可视化,在计算机辅助建模和网络数据查询系统中是不可或缺的。

1 现存的代谢网络自动绘图算法

网络可视化问题的核心是自动布局算法,即将节点和边按照一定规则集有序地布置到平面上。作为整体代谢网络的子网络的代谢途径,因为其结构简单,较易绘制,所以可以直接采用静态图形和手工描绘,如KEGG^[1-2]。但正如Brandenburg于1998年指出,手工绘图难以应对数据库的不断更新和满足用户自定义代谢途径的特定需求。所以自动布局及相关算法日益受到重视^[3]。

代谢过程明确的流向是代谢途径的本质特征,Sugiyama^[4]提出的阶层式布局(hierarchical layout)^[4-5]算法能有效地显示代谢途径整体流向^[3,6]。首先在尽量保证边的箭头方向一致的前提下分出不同的层次并确定每个节点所属的层次,然后调整同一层次上的节点位置,使得边的总长度不断降低的同时边交叉尽可能减少,形成清晰美观的总体布局。

代谢物循环过程所对应拓扑上的圈(cycle)结构也是代谢途径显著的特点。Karp等^[7]在可视化EcoCyc代谢数据库时率先应用环布局(circular layout)算法探查代谢途径中的圈结构,Wegner等^[8]、Schreiber^[6]和Becker等^[8]则在此基础上用阶层算法或力导向算法(force directed)^[9]计算出其余节点的位置并和圈结构进行整合,从而构建代谢途径的完整布局。此类复合算法可以较好地解决简单代谢途径的自动布局问题。当由大量代谢途径组合成一个庞大的整体代谢网络时,原有的明确流向被完全打乱,各节点间的联系由于紧密耦合变得错综复杂,难以从整体布局中寻觅到清楚的网络流向,这时力图将功能联系紧密的节点布局在一起以突出功能模块就更有意义;而且,研究表明,代谢网络本身具有明确的模块结构^[10]。因此对复杂的代谢网络问题,难以体现面向模块的阶层和环布局算

法就无法适应。

连续力导向是一种不依赖拓扑结构简化并一定程度倾向于呈现功能模块的算法,因而被用来对复杂的代谢网络进行布局。其基本思想是将所有节点当作相互排斥的粒子,而边则意味着相关节点间存在吸引力。当力场达到平衡时,有边连接的节点则倾向于聚集在一起,而没有边连接的节点则将互相远离^[9]。然而,该方法的缺陷也是明显的。由于力场模型中节点间普遍存在的排斥作用,其结果倾向于产生开放延伸型布局,不利于清晰模块的形成;而直接相连的节点之间的吸引作用导致节点过分靠近,以致于难以清楚分辨某些相邻节点,无法保证布局的整齐、美观。

鉴于连续力导向布局算法的种种缺陷,Li等^[11]提出一个离散布局方法,称为网格布局(grid layout)算法,将节点放置在网格格点上避免节点重叠同时使节点间距适中,通过设计适当的目标函数并优化之,将布局问题转化为最优化问题。对酵母细胞周期调控网络等实例的布局计算表明,网格布局不仅能生成紧凑整齐的布局,而且节点在布局中的几何位置分布与生物功能模块有很好的对应关系,显示了网格布局在复杂网络可视化中的优势。因此,网格布局引起了较多关注。如东京大学的Miyano研究小组,为提高网格算法的效率,从引入生物学属性^[12-13]和改进算法优化流程^[14]等角度做了持续有效的工作。

网格布局算法的一个主要缺点是计算量大,对大型网络仍然需要耗费较多的时间才能生成满意的布局。为了动态整合迅速增长的网络数据和清晰展示复杂网络结构,需要设计更加高效的布局算法。因此,笔者提出一种快速网格布局算法,采取邻域试探和扰动再优化的全局搜索策略以提高算法运行速度,对大规模代谢网络的运算结果表明,完全能够满足大规模代谢网络可视化需求。

2 快速网格布局算法

所有的节点和边依据其几何关系,按照一定的规则布置到平面或空间,就构成一个网络布局。通常采用的布局描述中,节点被视作一个不考虑大小的几何点;节点间连接的直线段表示边。本文即采取这种简化方案,所以只要给定节点的几何位置就可以完全描述一个布局。

作为一种特殊平面网络布局的网格布局,将所有的节点放在方格格点上。设节点*i*的坐标为 $r_i = (x_i, y_i)$, $i = 1, 2, \dots, n$, n 为网络中的节点数, x_i

和 y_i 应为整数。所有节点的坐标构成的矢量 $\mathbf{R} = (r_1, r_2, \dots, r_n)$ 描述了一个网格布局。自动布局算法就是根据网络的拓扑结构,依照一定的规则计算出 \mathbf{R} 。笔者首先定义评价布局质量的目标函数 $f(\mathbf{R})$,然后设计优化算法,通过极小化目标函数得到具体布局。

2.1 目标函数

目标函数是网格布局算法要解决的首要问题,需要抽象出适于优化的数学对象。目标函数应该表现为所有成对粒子相互作用的总和,即

$$f(\mathbf{R}) = \sum_{i < j} \varphi(w_{ij}, d_{ij}),$$

$\varphi(w_{ij}, d_{ij})$ 是节点对 i 和 j 之间的值并且

$$\varphi(w_{ij}, d_{ij}) = w_{ij} d_{ij},$$

w_{ij} 是节点 i 和 j 之间依据拓扑结构定义的权重; d_{ij} 是节点 i 和 j 之间的距离,不一定是欧几里德距离,可以定义为不同方式。从简化计算量角度考虑,本文定义为:

$$d_{ij} = |x_i - x_j| + |y_i - y_j|$$

2.2 权重矩阵

权重矩阵是网格布局算法的关键,其中的矩阵元素是 w_{ij} ,反映了节点 i 和 j 之间关系的密切程度。如果布局中希望两个节点离得远点,相应的权重可以取负值,模拟成排斥关系;反之权重取正值,模拟成吸引关系。权重矩阵元素 w_{ij} 一般根据网络拓扑来定义,其原则是:紧密相关的节点赋以正值,而疏远的节点赋以负值。一种简便的方法是 w_{ij} 按照节点 i 和 j 之间路径距离赋值。

假设:

$$\mathbf{M}^{(k)} = (\mathbf{A} + \mathbf{I})^k \quad k = 1, 2, \dots,$$

其中:

$$A_{ij} = \begin{cases} 1, & \text{if nodes } i \text{ and } j \text{ are connected;} \\ 0, & \text{otherwise.} \end{cases}$$

其中 \mathbf{I} 为单位矩阵, \mathbf{A} 为网络的邻接矩阵,本文忽略边的方向将网络视作无向图,因此 \mathbf{A} 是对称矩阵。 $\mathbf{M}_y^{(k)}$ 反应了节点 i 和 j 之间的路径距离。一个基于 $\mathbf{M}_y^{(k)}$ 的权重矩阵例子如下:

$$w_{ij} = \begin{cases} 3, & \text{if } M_{ij}^{(1)} > 0; \\ 1, & \text{if } M_{ij}^{(1)} = 0 \text{ and } M_{ij}^{(2)} > 0; \\ 0, & \text{if } M_{ij}^{(2)} = 0 \text{ and } M_{ij}^{(3)} > 0; \\ -1, & \text{if } M_{ij}^{(3)} = 0 \text{ and } M_{ij}^{(4)} > 0; \\ -2, & \text{otherwise} \end{cases}$$

可以看出,如果节点 i 和 j 之间路径距离越短,节点间吸引越强;如果路径距离 > 4 , 节点相互排斥。为

了防止节点被排斥 ($w_{ij} < 0$) 到无限远,笔者引入抱和排斥项 $\varphi_{\min} < 0$, 所以目标函数 $\varphi(w_{ij}, d_{ij})$ 项修正为,

$$\varphi(w_{ij}, d_{ij}) = \max(w_{ij} d_{ij}, \varphi_{\min}).$$

2.3 快速网格布局主算法

主算法 FastGridLayout 描述了快速网格算法的整体流程。首先,产生随机的初始化布局 \mathbf{R} , 并初始化 f_{\min} , 由子函数 Perturb(\mathbf{R}, ρ) 随机扰动形成扰动后的布局 \mathbf{R}' , 再由子函数 LocalMin(\mathbf{R}) 优化 \mathbf{R}' 而得到候选布局,不断重复以上过程 n 次从中选择最优结果。算法结束时 \mathbf{R}_{\min} 是搜索到的最优解, f_{\min} 是相应的目标函数。

Algorithm. FastGridLayout(\mathbf{R}, n, ρ)

1. $\mathbf{R} \leftarrow$ a random layout
2. $\mathbf{R}_{\min} \leftarrow \mathbf{R}, f_{\min} \leftarrow$ LocalMin(\mathbf{R})
3. repeat n times
4. $\mathbf{R}' \leftarrow$ Perturb(\mathbf{R}, ρ)
5. $f_{\text{trial}} \leftarrow$ LocalMin(\mathbf{R}')
6. if $f_{\text{trial}} < f_{\min}$ {
7. $f_{\min} \leftarrow f_{\text{trial}}, \mathbf{R}_{\min} \leftarrow \mathbf{R}', \mathbf{R} \leftarrow \mathbf{R}'$
8. }
9. else $\mathbf{R} \leftarrow \mathbf{R}_{\min}$
10. end repeat

2.4 邻域试探策略和 LocalMin(\mathbf{R}) 算法

邻域试探是指在每个节点周围邻域空间内,通过不断改变节点的位置试探,并比较每次试探产生的目标函数以达到优化的目的。该策略变全局搜索为局部搜索,有效地压缩了搜索空间,可以极大地提高效率。

局部优化算法 LocalMin(\mathbf{R}) 采用邻域试探策略优化扰动后的布局,形成候选布局。引入最小变化操作子 $T_{\alpha p}$, 表示节点 α 移动到一个空格点 p , $T_{\alpha p} \mathbf{R}$ 即表示在布局 \mathbf{R} 中,节点 α 已经移动到空格点 p 。采用贪心策略,依次针对每个节点的最小变化操作 $T_{\alpha p}$, 通过目标函数的计算和比较,检查所有可能的最小变化,寻找到一个最好位置点并记录。由于依次搜索每个节点的最好值,所以从全局看来,这种贪心策略找到的布局是一种局部最优布局。

Algorithm. LocalMin(\mathbf{R})

// \mathbf{R} is the input layout; when exiting, \mathbf{R} is the optimized layout.

1. $f_{\min} \leftarrow f(\mathbf{R})$
2. repeat {
3. for each node α {
4. for each vacant neighboring point p of

```

α
5.   if  $f(T_{op}R) < f_{min}$  {
6.      $f_{min} \leftarrow f(T_{op}R), R \leftarrow T_{op}R$ 
7.   }
8.   }
9.   }
10.  until  $f_{min}$  no longer changes
11.  return  $f_{min}$ 
    
```

2.5 扰动再优化策略和 Perturb(R, p)算法

通过邻域试探搜索后优化的布局是一个候选布局,为了产生更好的结果,需要从不同的候选布局中“好中选优”。如何有效产生用于优化的新布局,既能提高算法效率,又能得到较好布局结果是关键问题。笔者采用一种扰动再优化策略。算法如下,

Algorithm. Perturb(R, p)

```

1.  for each node  $\alpha$  in  $R$  {
2.     $\xi \leftarrow a(0,1)$ -random number
3.    if  $\xi < p$  {
4.      move  $\alpha$  to a randomly chosen neighboring
      vacant point
5.    }
6.  }
7.  return  $R$ 
    
```

给定扰动概率 p ,依次让布局中的每个节点以该概率决定是否改变几何位置,如果随机数 $\xi \rightarrow p$,节点移到其邻域的某一随机选择的空节点,由此得到一个经过扰动后的布局。

扰动再优化策略的重要意义在于两点。首先,引入了扰动概率 p 。由于受扰动的布局来源于上一次经过优化的候选布局(上一次的最佳布局),扰动概率 p 表征在每次扰动后,作为新搜索出发点的受扰动布局与原来的候选布局的相似程度。 p 越小,节点位置更新的可能性越小,表示扰动后的布局与上一次的候选布局差别越小,对原来的布局表现出越好的记忆性,然而算法也越容易陷入布局目标函数的局部极小,使得新搜索产生更好布局的可能性降低。 p 越大,节点位置更新的可能越大,扰动后的布局与已找到的最佳布局差别越大,记忆性也越弱;极限情况 $p = 1$ 则意味着扰动过程中,所有节点位置参与更新,等价于全局范围内的随机搜索更新。 p 值太小容易陷入局部极小,而太大会失去对上次候选布局中某些较好结果(比如模块)的记忆,都不合适。笔者的计算和实验表明, p 取值 0.3 ~ 0.7 能在提高算法效率的同时得到较好的布局结果。其

次,引入了邻域搜索。每个节点在更新范围是其邻域而非全局范围,通过有效压缩节点更新范围,大大提高扰动速度。

3 实验结果

本次实验采用的代谢网络数据由 KEGG 上多个代谢途径的整合而成,来自于 <http://www.genome.jp/kegg/pathway/map/map00400.html>,如表 1 所示:共 257 个节点,289 条边。

表 1 由代谢途径整合而成的代谢网络

Tab.1 The metabolic network composing of metabolic pathways

代谢途径名称	节点数	布局图中基本形状
PHENYLALANINE TYROSINE TRYPTOPHAN BIOSYNTHESIS	24	
TYROSINE METABOLISM	88	
ALKALOID BIOSYNTHESIS-I	50	
ALKALOID BIOSYNYHESIS-II	45	
PHENYLALANINE METABOLISM	39	
PUROMYCIN BIOSYNTHESIS	11	

3.1 算法参数设置

在主算法中的参数设置如下: $n = 20$;布局全局范围是 $64 * 64$ 的矩形区域;随机扰动的邻域范围是以节点为中心边长为 7 的矩形区域;邻域搜索范围是以节点为中心、边长为 13 的矩形区域;节点被扰动的几率 $p = 0.7$ 。

3.2 权重矩阵方案

权重矩阵的设置会影响到布局结果中模块间和模块内部节点分布的风格(如紧凑或松散),实验取值是,当节点直接相连,即路径距离为 1 时, $w_{ij} = 6$;如果路径距离 ≥ 3 , $w_{ij} = -1$ 。如此设置的权重矩阵的取值既能保证模块内部紧凑,模块间又能清晰区隔。

$$w_{ij} = \begin{cases} 6, & \text{if } M_{ij}^{(1)} > 0; \\ 2, & \text{if } M_{ij}^{(1)} = 0 \text{ and } M_{ij}^{(2)} > 0; \\ -1, & \text{if } M_{ij}^{(2)} = 0 \text{ and } M_{ij}^{(3)} > 0; \\ -1, & \text{if } M_{ij}^{(3)} = 0 \text{ and } M_{ij}^{(4)} > 0; \\ -1, & \text{otherwise} \end{cases}$$

3.3 运行结果

在上述参数情形下,快速网格算法平均运行时间是 5~10 s(windows XP, 内存 1 GB, CPU 1.66 GHz Intel duel core T2050)。

已有的最快的自动绘图算法是 Kojima 等提出

的 CBS-grid^[14],在对 250 个节点进行布局的平均运行时间为 40 s(内存 4 GB,CPU 3.6 GHz)^[14]。运行结果整体布局如图 1 所示,可以看出,布局结果的模块结构划分非常清晰。

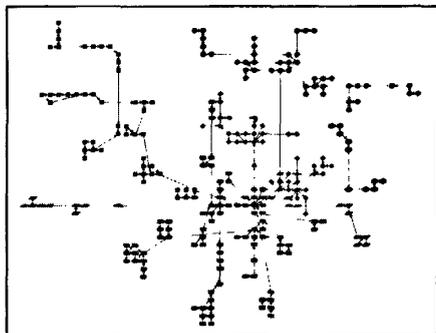


图 1 快速网格算法布局结果

Fig. 1 The layout result of fast grid algorithm

4 结 语

快速网格布局法着重从寻找复杂代谢网络的功能模块的目标出发,将网络节点的拓扑结构信息准确有效地关联到权重矩阵,并在此基础上构造出简洁的目标函数,采取邻域试探和扰动再优化策略优化目标函数,通过引入扰动概率 p ,限制搜索范围和扰动范围等关键技术,在保证较好的布局质量的同时,极大地提高算法效率,能满足大规模代谢网络绘图需求和快速处理代谢网络数据库动态的数据。

由于网格算法的全部节点是放置在格点上,当出现多个节点共线时,偶尔会出现不同的边相互覆盖的情形,可能会导致实际并不存在连接的节点看上去存在连接的假象,即所谓的“边贯穿节点”的问题。如何避免这种情形,进一步完善算法,这是快速网格算法有待解决的问题。

参考文献(References):

- [1] Kanehisa M, Araki M, Goto S, et al. KEGG for linking genomes to life and the environment[J]. *Nucleic Acids Research*, 2008, 36(1):480-484.
- [2] Kanehisa M, Goto S. KEGG: Kyoto encyclopedia of genes and genomes[J]. *Nucleic Acids Research*, 2000, 28(1):27-30.
- [3] Wegner K, Kummer U. A new dynamical layout algorithm for complex biochemical reaction networks[J]. *BMC Bioinformatics*, 2005(6):212-224.
- [4] Sugiyama K, Tagawa S, Toda M. Methods for visual understanding of hierarchical system structures[J]. *IEEE Trans Syst Man Cybern*, 1981(11):109-125.
- [5] Dwyer T, Koren Y, Marriott K. Drawing directed graphs using quadratic programming[J]. *IEEE Trans Vis Comput Graph*, 2006, 12(4):536-548.
- [6] Schreiber F. High quality visualization of biochemical pathways in BioPath[J]. *In Silico Biol*, 2002, 2(2):59-73.
- [7] Karp P D, Paley S. Automated drawing of metabolic pathways[C]. Tallahassee: Proceedings of the 3rd International Conference on Bioinformatics and Genome Research, 1994.
- [8] Becker M Y, Rojas I. A graph layout algorithm for drawing metabolic pathways[J]. *Bioinformatics*, 2001, 17(5):461-467.
- [9] Battista G D, Eades P, Tamassia R, et al. Algorithms for drawing graphs: an Annotated Bibliography[J]. *Comput Geom- Theor Appl*, 1994(4):235-282.
- [10] Barabasi A L, Oltvai Z N. Network biology: understanding the cell's functional organization[J]. *Nat Rev Genet*, 2004, 5(2):101-113.
- [11] Li W, Kurata H. A grid layout algorithm for automatic drawing of biochemical networks[J]. *Bioinformatics*, 2005, 21(9):2036-2042.
- [12] Kato M, Nagasaki M, Doi A, et al. Automatic drawing of biological networks using cross cost and subcomponent data [J]. *Genome Inform*, 2005, 16(2):22-31.
- [13] Kojima K, Nagasaki M, Jeong E, et al. An efficient grid layout algorithm for biological networks utilizing various biological attributes[J]. *BMC Bioinformatics*, 2007(8):76-92.
- [14] Kojima K, Nagasaki M, Miyano S. Fast grid layout algorithm for biological networks with sweep calculation[J]. *Bioinformatics*, 2008:196.

(责任编辑:秦和平)