

系统发生树构建方法综述*

郭静¹, 王超¹, 张宏彬¹, 陈峻^{2,3}

(1. 扬州工业职业技术学院 电子信息工程系, 江苏 扬州 225127; 2. 扬州大学 信息工程学院, 江苏 扬州 225009; 3. 南京大学 软件新技术国家重点实验室, 南京 210093)

摘要: 介绍了系统发生树构建的传统算法,总结了近年来用于构建系统发生树的各种群智能算法,主要包括蚁群算法、遗传算法和粒子群算法等。最后介绍了目前较新的系统发生网络分析方法,并对未来发展作了展望。

关键词: 系统发生树; 群智能算法; 蚁群算法; 遗传算法; 粒子群算法; 系统发生网络

中图分类号: TP301 **文献标志码:** A **文章编号:** 1001-3695(2013)03-0647-09

doi:10.3969/j.issn.1001-3695.2013.03.002

Review of phylogenetic tree construction

GUO Jing¹, WANG Chao¹, ZHANG Hong-bin¹, CHEN Ling^{2,3}

(1. Dept. of Electronic & Information Engineering, Yangzhou Polytechnic Institute, Yangzhou Jiangsu 225127, China; 2. College of Information Engineering, Yangzhou University, Yangzhou Jiangsu 225009, China; 3. National Key Laboratory of Novel Software Technology, Nanjing University, Nanjing 210093, China)

Abstract: This paper first introduced the traditional algorithm for phylogenetic tree construction, and then reviewed the swarm intelligence algorithms used to construct phylogenetic trees in recent years, such as ant colony algorithm, genetic algorithms and particle swarm optimization. Finally, this paper introduced the application of phylogenetic networks, and the future development prospect.

Key words: phylogenetic tree; swarm intelligence algorithms; ant colony algorithm; genetic algorithms; particle swarm optimization; phylogenetic networks

系统发生分析是指根据现有生物基因或物种多样性来重建生物的进化史,是生物信息学中一个非常重要的问题。一个可靠的系统发生的推断,将揭示出有关生物进化过程的顺序,有助于了解生物进化的历史和进化机制。进化历史可以表示成一棵树,即系统发生树。系统发生树根据物种之间的遗传历史描述了它们的世袭关系^[1]。由于能揭示不同物种和基因之间的关系,系统发生树在分子遗传学、医学及药物设计中具有很重要的意义。一棵表示 n 个物种之间进化关系的系统发生树通常用二叉树来表示,由叶子、分支点(内部节点)和分支长度组成。其中,叶子节点被称做分类单元,通常表示一个对象(蛋白质、物种等)的集合,是一项研究中的基本单位;分支点因为不能被直接观察到通常称为假定分类单元,分支长度则代表两个分支点之间或分枝点与叶子之间的距离。

Bodlaender 等人^[2]已经证明构建系统发生树是一个 NP 完全问题。由任意 n 个物种的集合构建系统发生树可以得到很多不同的拓扑结构。根据是否指出最近公共祖先分类单元所在位置,可以将系统发生树分成有根树和无根树^[3]两种。记有根树和无根树的数目分别为 NT_r 和 NT_u ,式(1)(2)分别给出了计算有根树和无根树数目的方法。

$$NT_u = \frac{(2n-3)!}{2^{n-2}(n-2)!} \quad (1)$$

$$NT_r = \frac{(2n-5)!}{2^{n-3}(n-3)!} \quad (2)$$

从公式中可以看出,随着研究物种数目 n 的增加,可能的有根系统发生树和无根系统发生树的数目迅速增加。例如,当用最大似然法构建 15 个物种的系统发生树时,可以得到 213、458、46、676、875 棵不同的树。由此可见,要对每种可能的系统发生树都进行穷举,求出完全精确的解显然是不可能的。但是只有其中的一棵树代表了待分析基因或生物体之间的真实进化关系,而目的就是要找出这棵反映真实进化关系的系统发生树。所以研究在适当的时间内得到最优近似解的算法就具有非常重要的实际意义。

1 传统的系统发生树构建算法

目前构建系统发生树的方法可以分为两大类:基于距离的方法,如 UPGMA(非加权平均法)^[4]、邻接法 NJ^[5]以及 Fitch-Margoliash (FM) 法^[6];基于特征的方法,如最大简约法 MP^[7]和最大似然法 ML^[8]。

基于距离的构建方法在构建系统发生树之前首先估算序列两两间的进化距离,利用相似性的原则构建距离矩阵,然后根据不同的算法,从进化距离最短的开始依次聚类,利用距离矩阵计算出最优树,或将分支长度之和最小化,获得优化树。

1.1 非加权平均法

Rohlf 于 1963 提出的非加权平均法 UPGMA^[9,10]是一种较

收稿日期: 2012-07-08; **修回日期:** 2012-09-04 **基金项目:** 国家自然科学基金资助项目(60673060,60773103);江苏省自然科学基金资助项目(BK2008206);江苏省教育厅自然科学基金资助项目(08kjh520012)

作者简介: 郭静(1981-),女,助教,硕士研究生,主要研究方向为并行计算、算法优化(enguoqing@163.com);王超(1982-),男,助教,硕士研究生,主要研究方向为机器学习、车间调度;张宏彬(1967-),男,副教授,高级工程师,工学硕士,主要研究方向为人工智能、现代教育技术等;陈峻(1951-),男,教授,博导,主要研究方向为并行计算、数据挖掘、算法优化等。

为常用的聚类方法,最初是用来解决分类问题的。该方法同样可以用来推断系统发生树,但前提条件是假定对所有进化世系,核苷酸和氨基酸替换的速率是相同的。首先计算各分类群两两间的进化距离,将进化距离最小的两个分类群(如 A、B)最先聚合,将 A、B 合并成一个聚合群 C,用 C 代替 A 和 B,计算 C 与剩余分类群两两间的进化距离,再次将距离最小的两个分类群聚合,合并为一个聚合群 D,依此循环计算,直至所有分类群都包含在一棵树中。此方法在不同谱系间的进化速率有较大差异或有同源序列的平行进化时,常得出错误的拓扑结构^[11],而且当系统发生树的状态空间较大时,UPGMA 法的可操作性极差,因而该方法的使用较为有限。

1.2 邻接法

邻接法 NJ^[12]类似于 UPGMA,在每一步中创建一个新的距离矩阵,并基于距离矩阵来构建系统发生树。不同之处在于,NJ 并不构建聚类,而是直接为内部节点计算距离。NJ 的第一步是用每个节点或分类单元之间的海明距离创建矩阵;然后利用最小距离计算两个节点到直接相连节点的距离,从而计算得到一个新的距离矩阵,并用新的节点代替当前合并的原始节点。NJ 的运算速度最快,但该方法每迭代运算一次均只搜索最近邻居配对,对其他可能的配对不加以考虑,最终只生成单一的最优树,可能会遗漏一些拓扑结构更合理的次优树。为弥补缺陷,Pearson 等人提出遍及邻接法(generalized NJ),将一些最近的树也包含在搜索算法中^[13],逐步迭加,期望找到更好的 NJ 树。以下是 NJ 算法的描述:

算法 1 Neighbor joining^[14]

```

initialization
define  $T$  to be the set of leaf nodes, one for each given sequence and put  $L = T$ .
iteration:
pick a pair of  $i, j$  in  $L$  for which distance from  $i$  to  $j$  is minimal.
define a new node  $k$  and set  $d_{km} = (d_{im} + d_{jm} + d_{ij})/2$ , for all  $M$  in  $L$ .
add  $k$  to  $T$  with edges of lengths  $d_{ik} = (d_{ij} + r_i - r_j)/2$ ,  $d_{jk} = d_{ij} - d_{ik}$ 
joining  $k$  to  $i$  and  $j$  respectively.
remove  $i$  and  $j$  from  $L$  and add  $k$ .
termination:

```

1.3 Fitch-Margoliash 法

Fitch-Margoliash (FM) 法^[15]首先估计树中各分支长度,如果分类群为 3,先假定分类群 A 和 B 之间的进化距离最小,然后与 C 聚合;如果分类群个数多于 3,则将其其他分类群合并为 C,AB 距离不变,A(或 B)与 C 的距离 d_{AB} (或 d_{BC})为 A(或 B)与 C 中所有分类群的距离算术平均值。因此,FM 方法可以计算树中每个分支长度。FM 法是从所有可能的进化树中挑选树长最短的作为最优树,是距离法构建系统发生树中相对较好的方法。不过值得注意的是距离法在将原始数据转换成距离矩阵时难免会丢失一些进化信息^[16]。

2 构建系统发生树的群智能算法

2.1 群智能算法

近年来,受自然界启发的一类算法被称之为群智能算法(SI)^[17-20],吸引了全世界生物计算相关问题的研究者的注意。由简单个体组成的群落与环境以及个体之间的互动行为,可称之为群集智能,其实质就是无智能的主体通过合作表现出智能行为。任何一种由昆虫群体或其他动物社会行为机制而激发设计出的算法或分布式解决问题的策略均属于群智能。群集智能具有灵活性、稳健性和自组织等特性与优点,在没有集中控

制并且不提供全局模型的前提下,为寻找复杂分布式问题的解决方案提供了基础^[20]。

SI 范例的五个基本原则为:a)附近原则(proximity principle),群内个体具有能执行简单的时间或空间上的评估和计算的能力;b)质量原则(quality principle),群内个体能对环境(包括群内其他个体)的关键性因素的变化作出响应;c)多样化反应原则(principle of diverse response),群内不同个体对环境中的某一变化所表现出的响应行为具有多样性;d)稳定性原则(stability principle),不是每次环境的变化都会导致整个群体的行为模式的改变;e)适应性原则(adaptability principle),环境所发生的变化中若出现群体值得付出代价的改变机遇,群体必须能够改变其行为模式。

群智能理论研究领域主要有四种算法,即遗传算法(GA)、蚁群算法(ACO)、粒子群优化算法(PSO)和智能鱼群算法(AFSA)。其中前三种算法已经被广泛地应用于系统发生树的构建问题中。

2.2 构建系统发生树的蚁群算法

2.2.1 蚁群算法

蚁群算法(ACO)^[21-23]是在 20 世纪 90 年代初,由意大利学者 Dorigo 和 Maniezzo 等人提出的,是受到蚂蚁在觅食过程中能发现蚁巢到食物的最短路径这种搜索机制的启发而发展起来的一种群体智能算法,在求解复杂优化问题特别是离散优化问题方面具有优越性。

如图 1 所示,当蚂蚁从巢穴走向食物源时,沿途散发信息素作为群体间的通信物质,某条路径上积累的信息素数量与经过该路径的蚂蚁数目成正比。ACO 用到两个原则:路径的选择和信息素更新的概率。ACO 中的人工蚂蚁在当前节点选择合适路径的概率,是基于信息素的量以及在该节点的每个分支的长度。信息素的更新是由信息素的积累或蒸发带来的变换,蒸发是避免算法过快收敛到局部最优区。

蚁群算法特别适合于在离散优化问题的解空间进行多点非确定性搜索,已经先后被应用到 TSP、二次分配问题、工件调度问题、图着色等多个经典组合优化问题,取得了很好的效果,成为求解组合优化等 NP-hard 问题的一种有潜力的演化算法。

2.2.2 构建系统发生树的蚁群算法

系统发生树的构建问题与标准 TSP 非常相似,可以把 TSP 中的各个城市看做是系统发生树中的每个分类单元,分类单元之间的两两距离类似于两个城市之间的距离。因此,近年来出现了许多用蚁群算法解决系统发生树构建问题的方法。

Korostensky 等人^[24]提出了一种构建系统发生树的方法 CircTree,此方法将系统发生树的构建简化为 TSP,利用给定物种序列的最小得分来构建系统发生树。首先将序列之间的两两距离作为 TSP 算法的输入,输出则是一个回路,代表树中分支的最小得分序列;然后用回路的次序和得分来构建最优树的拓扑结构。为构建系统发生树,用到以下简单的思想:正确的树的得分将和用 TSP 得到的得分相同。这样,用另一算法构建系统发生树并比较树的得分和 TSP 计算出的得分。这一搜索方法有些类似于最大简约法,因此当物种数目较大时,构建系统发生树需要较大的功夫作计算。

Ando 等人^[25]提出了一种蚁群算法,应用栈计数来定义树结构的后缀表示。算法依然用叶子节点之间的两两距离作为输入,利用 TSP 算法访问各个分类单元一次以构建一个环路,如图 2 所示。一棵树的总分支长度可以沿着树的边的环路,通

过计算式 $C(S)$ 来获得,式 $C(S)$ 的计算具体方法参考文献 [26]。

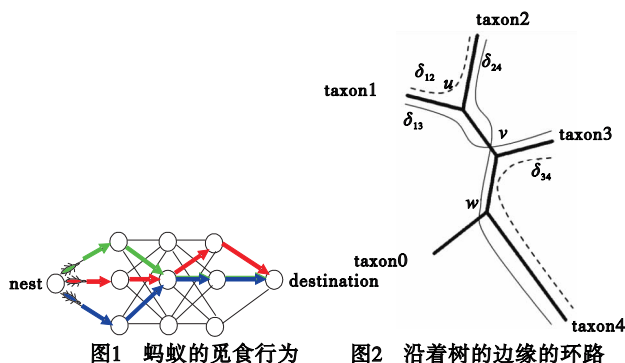


图1 蚂蚁的觅食行为

图2 沿着树的边缘的环路

一棵系统发生树可以用一个后缀表示序列来表示,如图2所示的系统发生树所对应的后缀表示序列为:Tx0 Tx1 Tx2 + Tx3 + + Tx4 +。后缀表示序列必须满足一定的条件才能构成一棵合法的系统发生树。此条件可以用一个栈计数的形式来解释,如图3所示。其中叶子节点用+1来赋值,非叶子节点用-1来赋值,要使后缀序列能合法地表示一棵系统发生树,栈的计数值必须为1。在构建系统发生树的过程中,蚂蚁遵循这一条件访问树中的叶子节点和顶点以构建一个后缀表示序列来表示二叉树,因此算法在模拟实验和用15个物种的蛋白质序列比对中表现出较高的可比较性。

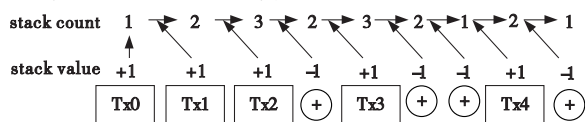


图3 图2所示的树的栈计数法

Kummarkaew 等人^[27]是利用最小分支长度来构建进化树的新算法。在利用 ACO 解决问题前,输入物种的两两距离转换成 Steiner 树问题。算法的第一步是找到两个任意物种的公共祖先,两个物种将合并成为这一新的祖先,这一祖先又被看做是下一步的一个物种。此步骤将物种的数目降低到1,但是也会把分支点的数目增加到 $n-2$ 。此合并步骤一直迭代,直到分支点的数目到达一棵无根树中的内部节点总数 $n-2$ 为止,如图4所示。

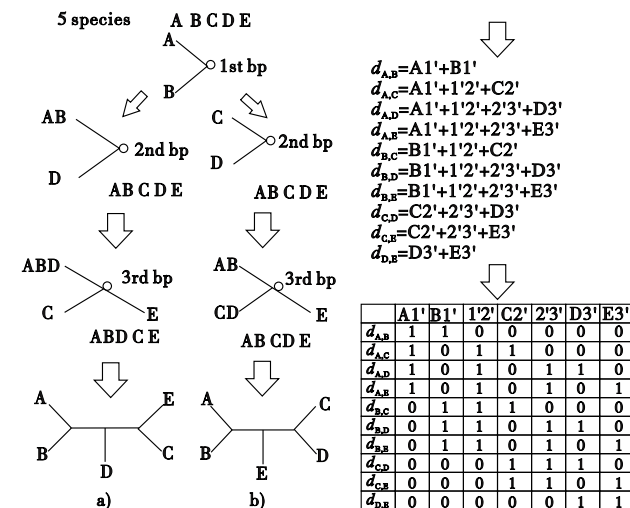
每构建一棵树,所选择的分支点可以表示成大小为 $[1 \times n-2]$ 的向量的形式。例如,一个五个物种的系统发生树(图4(a))可以表示成一个 $[1 \ 2 \ 1]$ 的向量,而 $[1 \ 4 \ 1]$ 向量表示图4(b)中的树,其中向量中的每个元素是在剩余物种中选择的第 k 个两两物种对。例如,向量 $[1 \ 4 \ 1]$ 中的4表示在第二步中选择的物种对 C 和 D 是剩余物种中的第4个物种对。

分支长度可以利用树一分析矩阵(tree-analyzed matrix)和两两距离向量来计算。树一分析矩阵中的每个元素都是一个二进制数,表示该路径上存在的分支,如图5所示。

利用蚁群算法构建进化树时,物种被看做是蚂蚁从初始点到结束区域旅行的节点,在每个区域内蚂蚁被限制智能访问一个节点。区域的数目等于 $n-2$,其中 n 是物种数目。在每个区域中存在 $q(q-1)/2$ 个节点,其中在入口区域中 $q=n$,当前进到结束区域时降低到 $1(q=n, n-1, \dots, 3)$ 。在入口区域中,蚂蚁在移动到下一个区域中的新节点前先随机选择一个节点。然后选择一个新节点的概率取决于信息素以及当前节点和新节点之间的距离(式(3))。当蚁群中的所有蚂蚁都结束它们的旅行后,最短路径或者最短总分支长度树将确定。

$$P_k(i, j, c) = \frac{[\tau(i, j, c)]^\alpha \cdot [1/d(i, j, c)]^\beta}{\sum [\tau(i, u, c)]^\alpha \cdot [1/d(i, u, c)]^\beta} \quad (3)$$

其中: $P_k(i, j, c)$ 是边 (i, j) 在第 c 个区域的概率; τ 是信息素, $d(i, u, c)$ 表示附近区域的两两距离; α 和 β 则是任意常数。

图4 五个物种从相同的开始节点 图5 利用tree-analyzed矩阵的对出发构建出的两棵样树^[29] 分支长度计算

算法中新的两两距离不是常数,而是在每次合并两个物种时重新计算。当两个物种合并为一个新物种时,其他物种到这一新合并出的物种之间的距离将会改变。Kummarkaew 的实验所得到的树相当于利用 NJ 方法构建的树,但是该方法需要较大的处理过程用于定义无用的中间节点。

Perretto 等人^[30]所提出的构建系统发生树的蚁群算法,首先利用物种间的距离矩阵构建一个二维的完全连通图,图中的节点代表物种,边表示物种之间的进化距离,如图6所示。蚂蚁从随机选择的节点出发在图中旅行。蚂蚁从节点 i 出发,根据概率式 $P_k(i, j)$ ^[28] 选择下一个节点 j ,寻找一条使函数值最小的路径,并找到两个物种之间的最短进化距离。

species	A	B	C	D
A	0.00	0.20	0.84	0.83
B	0.20	0.00	0.90	0.89
C	0.84	0.90	0.00	0.01
D	0.84	0.89	0.01	0.00

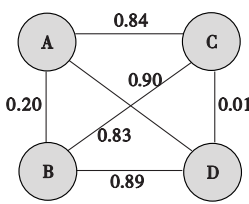


图6 四个物种的距离矩阵和对应的二维图

当此方法中,蚂蚁在先前访问过的两个节点之间创建一个中间节点 u 来表示两个物种的祖先,与 Kummarkaew 等人^[27]的方法不同,此中间节点不在构建树的节点集合中。利用 u 重新计算剩余节点间的距离的方法如式(4),也不同于文献[27]。

$$d_{mu}(i, j) = \begin{cases} d(i, u) + [d(i, u) - d(j, u)]\eta & \text{if } d(j, u) \geq d(i, u) \\ d(j, u) + [d(j, u) - d(i, u)]\eta & \text{if } d(i, u) > d(j, u) \end{cases} \quad (4)$$

迭代以上步骤直到图中所有的节点都加入已访问过的节点集合为止,路径也由此得出。此路径的得分为邻接点的转移概率之和。蚂蚁构建的路径用来更新信息素。在一次循环中所构建的至少一条路径上的所有节点上会有信息素的增长,这一关键点有助于避免陷入局部最大值。根据此方法,利用与解决 TSP 的蚁群算法非常接近的算法,可以很有效地构建出系统发生树。路径建好后,该路径上的所有节点的信息素增加,并受蒸发的影响而消耗。文献[28]给出了信息素更新式。ACO 因此提供了一个最优可能路径的线性列表来构建系统发生树。构建系统发生树的算法如下:

算法2 Tree construction algorithm^[28]

while not(all species grouped)

```

find  $i, j$  pair that have the largest value in the pheromone matrix.
if ( $i$  or  $j$ ) already grouped change index by group index
group  $i, j$  pair into a new species  $k$ 
compute the distance between current species and ancestor
delete the value of  $i, j$  pair
end

```

这些初始结果证明 ACO 可以成功地应用于系统发生树的构建,更重要的是 ACO 可能提供比其他方法更好的结果。实验用到的包括 α 和 β 在内的一组参数,并不是最优的结果。因此,仅是最初的证明不能被看做 ACO 总是精确的证据。

Perretto 等人^[28]也将 ACO 用于系统发生树的构建中,并利用 Seq-Gen 产生序列,然后用 PHYLIP 在线软件包计算序列之间的距离矩阵;再将这些输入数据填入 PHYLIP 中的 Fitch 和 NJ 算法以比较他们提出的 ACO 方法,如图 7 所示。实验中分别用到 8 和 16 个叶子的不同大小的树。对于较少的含 8 个叶子的树,蚁群算法给出最优的全局结果;但对于 16 个叶子,FITCH 给出的结果最好。需要在更大的树上完成更多的实验,但是结果已表明 ACO 可与传统的方法相比拟。

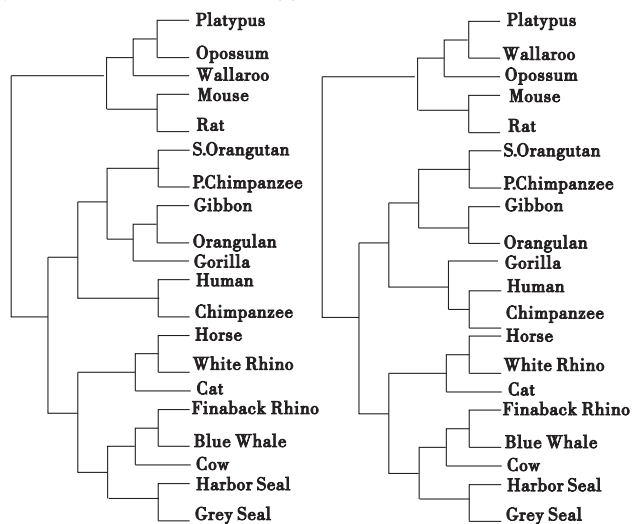


图7 利用ACO和NJ方法产生的系统发生树^[29]

Vittori 等人^[29]提出的 ABPR 算法,是基于蚁群算法的评估系统发生树的距离算法。与前面两种算法类似,以序列间的进化距离为基础,迭代地合并两个分类单元,同时根据树的长度解释所构建系统发生树的质量。计算分支长度的方法是:物种 i 和 j 合并之后,ABPR 算法创建一个新节点 A 作为 i 和 j 的公共祖先。分支 A_i 和 A_j 的长度,即每个物种到公共祖先的分支长度用 NJ 算法中的方法计算。分支 A_i 和 A_j 的长度分别用 C_{A_i} 和 C_{A_j} 表示,可以用式(5)(6)计算。

$$C_{A_i} = d_{ij} \times \frac{R_i}{R_i + R_j} \quad (5)$$

$$C_{A_j} = d_{ij} \times \frac{R_j}{R_i + R_j} \quad (6)$$

其中: d_{ij} 是物种 i 和 j 之间的进化距离; R_i 是物种 i 和其他物种之间的进化距离之和; $R_i = \sum d_{ik}$,且 $0 < k < n$ 。与 NJ 算法中找到的条件相似,如果 $R_i > R_j$, C_{A_i} 将会大于 C_{A_j} 。

此计算方法允许用 ABPR 算法和 NJ 算法得到的评估树的长度之间的直接对比。对于物种水平上的 DNA 序列的系统发生评估,ABPR 算法优于 NJ 算法。由灵长类动物、鸟和哺乳动物恢复的系统发生关系,与用最大简约法、最大似然法和 BA 推断等的优化算法所得到的结果相似。在树的搜索过程中评估多样化的系统发生树方面,ABPR 算法有传统基于距离的算法所没有的优点。

Chen 等人^[30]采用自顶而下的方式构建系统发生树。首先,树的根被定义为对应的基因序列的整个集合,然后集合被分为两个子集,其中每个子集是根的一个孩子;每个子集再进一步分为更小的子集作为他们的孩子;迭代此二分过程,直到每个子集只有一个序列组成为止,这一序列恰好是树的叶子节点。随着二分过程的递归,逐步构建系统发生树,利用蚁群算法完成系统发生树的聚类过程。文献[30]给出了算法框架;此外,还对蚂蚁选择路径的概率、适应度值、信息素的更新作了详细的介绍。此方法构建系统发生树时有两个方面的问题值得注意:

a)如何确定新聚类的中心,从理论上说一个聚类的中心应该是具有到所有其他基因序列的总距离的基因序列。然而,由于所选择的初始中心有可能不是最佳二分聚类的中心,应根据每一次迭代中新分配的基因序列对聚类中心进行调整。

b)算法的终止条件的设定,文中在相邻迭代中得到的聚类趋于稳定时,停止迭代。

ACO-based 算法与传统基于距离的算法相比具有潜在竞争力^[28,31,32]。用较大数据集构建系统发生树时,合并速率较快、精确度较高。当数据集比较大时,目前选择基于距离的方法可以使系统发生树在几分钟或几个小时内得到。

2.3 构建系统发生树的遗传算法

遗传算法应用于系统发生树的构建比较早,许多文献描述了基于 GAs 的最大似然法重建系统发生树,如 Lewis^[33]、Reijmers 等人^[34]、Matsuda^[35]、Poladian 等人^[36]。Shorikhine^[37]做了进一步的工作,又针对最大似然法重建系统发生树,提出了自适应算法。

Reijmers 等人^[34]的算法思想是:首先用一个普通聚类方法决定好树的拓扑结构,然后计算树的分支长度;若得到的结果不是最优树,交换树的分支可以改变树的拓扑结构;整个过程一直迭代直到得到满意的结果为止,此方法的效率很大程度上取决于初始产生的树。图 8 列出了优化系统发生树拓扑结构的步骤。

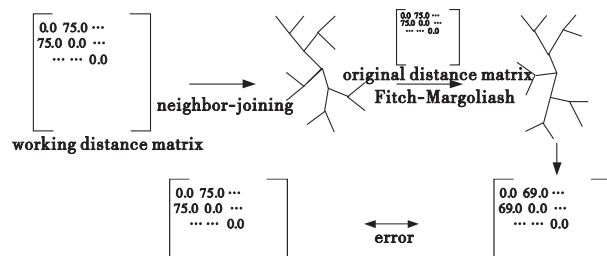


图8 寻找最优树过程的示意图^[34]

初始的拓扑结构由邻接法确定,然后利用 Fitch-Margoliash 计算分支长度,最后得到一个错误。一旦树的拓扑结构和分支长度确定,便可产生一个新的距离矩阵。通过比较此距离矩阵与原始距离矩阵,可以给树分配一个质量标准。重建的质量标准是重建的距离 D_r 和原始距离 D 之间的相对偏差 (relative deviation) 的平方根 rms,计算式如下。

$$\text{rms} = 100 \left[\frac{1}{\frac{n(n-1)}{2} - 1} \sum_i \left(\frac{D'_i - D_i}{D_i} \right)^2 \right]^{\frac{1}{2}} \quad (7)$$

若重建后的距离等于原始距离,则 rms 等于 0;若两个距离相差很大,则 rms 将变得很大。当然,其他标准或是标准的组合也可以用于获得系统发生重建的质量标准。改变后的距离矩阵可以用做图 8 表述的方案为初始点;适应度值是 rms 值的倒数。遗传算法经过若干次选择、重组和变异之后趋向全局

最优解,即表示树的距离矩阵。此方法的缺点是由于它的全局特性,算法在趋于最优树之前很可能需要测试更多不同的拓扑结构。由于在应用邻接法之后已经找到了 rms 值够小的树,在间接或局部搜索过程中不需要测试树的拓扑结构。

利用遗传算法构建系统发生树,首先解决的问题是如何表示树的拓扑结构,文献[16]给出了两种表示方法。其中一种表示方法是距离矩阵表示法,如图9左所示。它不是利用输入的距离矩阵,而是用另一种可以表示并能直接改变的拓扑结构来代替。将距离矩阵映射成一个拓扑结构,因此通过使用遗传算法改变距离矩阵以得到不同的拓扑结构,以上算法中用到的即为距离矩阵表示法^[34]。另一种方法是 Prüfer 序列表示法,如图9右所示,对一棵具有 n 个节点的无根二叉树,可以构建一个 $2(n-2)$ 长度的序列来唯一地表示一棵树。距离矩阵表示法实现简单,但具有较多的缺点,如具有冗余性,评估前须将其转换为拓扑结构,无法直接改变拓扑结构等。Prüfer 序列提供了一个简单的方法,系统地枚举了所有拓扑结构。

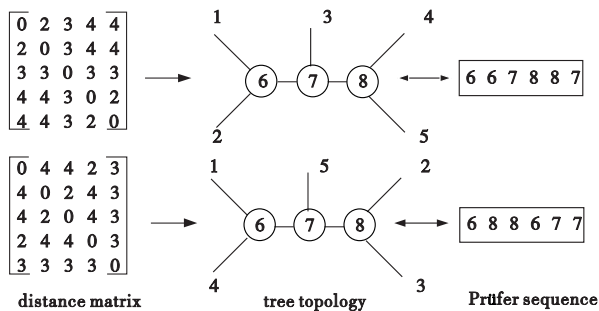


图9 两个拓扑结构及其对应的距离矩阵和Prüfer序列表示法^[16]

其次是初始化问题,大多数此类算法都是从用 NJ 算法得到的结果中选择初始种群,如文献[34,38],Fischer 等人^[38]从结果中选择了 1/10 的种群作为初始种群。图10给出一个随机选择的例子,其中具有最好适应值的个体得到 n 次选择,此处是 6,而最差的得到一次^[38]。

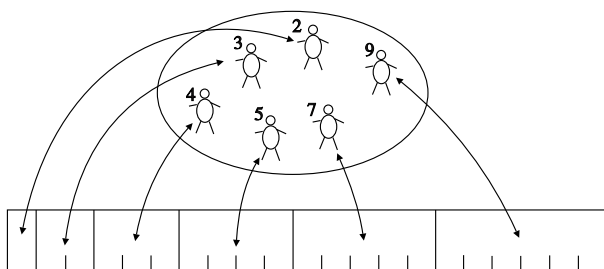


图10 随机选择的例子

初始化之后遗传算法经过若干次选择、重组和变异之后趋向最优解。常用的选择方法有锦标赛选择法、轮盘赌选择法和随机遍历选择法。重组是改变 GA 种群中个体的第一种方式,变异则是改变候选解的第二种方式。变异受到生物变异的启发,并试图保持一些种群中的遗传多样性,变异的其中一个目的是使算法避免陷入局部最小值,或通过防止个体太过相似以完全避免。变异算子有以下三种:交换是一种通过交换两个叶子节点来改变树的变异算子,此方法很难预测适应值的改变;近邻交换(NNI)是一种变异一棵树时广泛使用的方法,如图11所示,通过交换两个相邻子树来实现,也较难预测适应值的改变;子树修剪和嫁接(SPR)是移动树中的子树的变异算子。前两种方法很难预测适应值的改变,此方法存在导致相同适应值的危险,即当子树插入到其移除位置,导致形成相同的拓扑结构时。而结合 NNI,可检查所有可能的结果(尝试所有边作为插入点),然后选择具有最小得分的最佳拓扑结构。

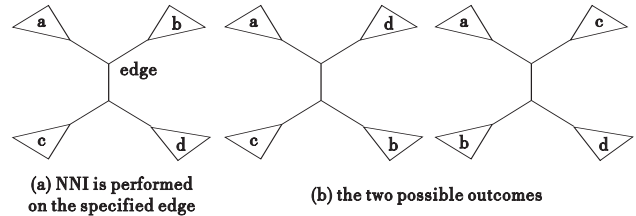


图11 NNI方法图示

Fischer 等人^[38]的算法中选择了随机遍历选择法,重组时借鉴有性繁殖的生物概念,选择了系统发生树中最流行的剪枝—删除—嫁接(prune-delete-graft)重组法,缩写为 PDG^[3]。PDG 重组的示意图^[38]如图12所示。变异时采用的是 NNI 法。并为补偿此类算法大都没有给出 GAS 的较好参数这一缺点,在参数的调整上做了重要工作。

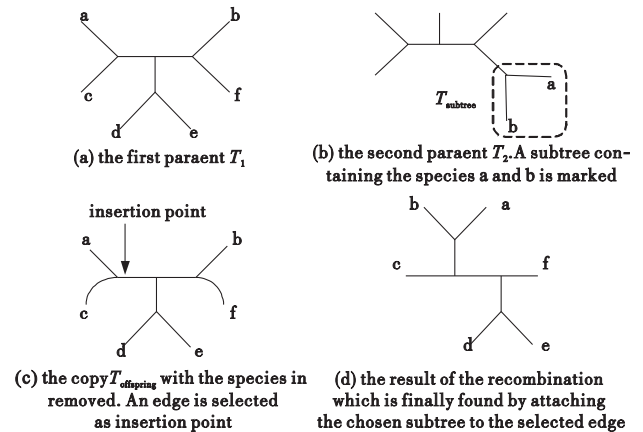


图12 PDG的重组图

遗传算法的优点^[34,39]是除了得到一个最优树之外,还可以得到一组质量类似的树,这种技术在每次循环中只产生一棵树,并且在同代中没有重复的树出现。结果树中含有最优树和很多准最优树,从而大大提高了搜索的效率。缺点是大量参数需要优化,在 25 个对象的情况下有 300 个参数需要优化;通过改变系统发生树的表示可以产生许多参数^[34]。

2.4 粒子群算法

2.4.1 粒子群算法简介

PSO 最初起源于 1995 年,由 Kennedy 和 Eberhart^[40]提出,是属于群智能一类的算法,其思想来源于对鸟类和鱼类的简单社会模型的研究和行为模拟。PSO 是基于群体算法,可以开发个体的群体去探测一个搜索空间的可能范围。不同于 ACO 中利用蚁群间交流的协同机制的观念,鸟类利用的是类似广播的通信策略。每个 agent 是一个粒子,是由以下几个部分组成的结构:优化区域中当前位置的坐标,目前访问过的最好解决方法,被看做邻居的其他 agents 的子集^[41]。

如算法中所示,在每次迭代期间,每个粒子朝着自己目前找到的最好解的方向加速,同时也朝着目前发现的全局最优位置的方向加速^[40]。这意味着当一个粒子发现正确方向的某物时,它会朝着此方向移动,同时所有其他粒子也会跟着它移动。本质上,一只鸟朝着某个方向移动,其余鸟群相对它来移动自己的位置,以保持它们的位置。因此,一个有效的算法应具有利用群体影响平衡个体的任务,并有效地引领群体的整体移动。用来更新粒子群位置和速度的公式由三部分组成^[42]。a) 粒子之前的速度,说明了粒子目前的状态;b) 认知部分(cognition modal),表示粒子本身的思考,使粒子有足够强的全局搜索能力,避免局部极小;c) 社会部分(social modal),体现粒子

间的信息共享。三个部分共同决定了粒子的空间搜索能力。

2.4.2 构建系统发生树的粒子群算法

Lv 等人^[42]将 PSO 应用于系统发生树的构建,并提出了离散粒子群优化算法,实验结果表明,PSO 适合于序列数目少于40的情况。经过分析得出,PSO 与遗传算法相比能够更快地趋向于最优树,但是当序列数目比较大时算法的效率将会降低。

Li 等人^[43]的用 PSO 构建系统发生树的算法名为 PSOML。在算法的迭代过程中,粒子状态的更新是通过某些拓扑结构的重排来实现的。当前拓扑改变方法包括最近邻居交换技术(近邻交换 NNI)、子树剪枝和嫁接技术(SPR)、TBR、 p -ECR。其中前两种技术 NNI 和 SPR 跟遗传算法中的变异算子相同^[38],TBR 技术是先将树 T 中删除一条边,从而创建出子树 t 和 $T-t$,然后在子树 t 和 $T-t$ 的任意两条边的中点之间添加一个新的边,创建一棵新树,如图 13 所示。

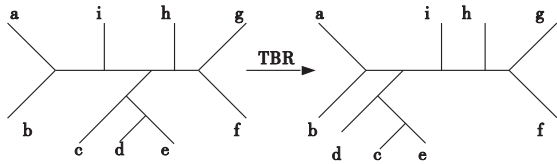


图13 TBR技术

而 p -ECR 技术的思想是:一次 p -ECR 移动意味着一次收缩 p 条边,然后完善细化创建过程中尚未解决的节点以恢复一棵二叉树。一次收缩在树中折叠一条边并确定它的两个端点,而一次细化完善扩展一个未解决的节点成有一条边连接的两个节点。图 14 中的 T_1 和 T_5 是由一个 2-ECR 移动分开的。在 T_1 中收缩两条边 e_1 和 e_2 产生 T_3 ,完善后产生 T_5 。

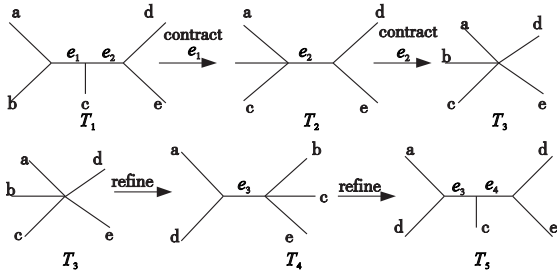


图14 一个2-ECR过程的图解

为提高 p -ECR 的效率, Li 等人^[42]提出了 p -ECRNJ 新算法。 p -ECRNJ 的主要思想是利用 NJ 完善 p -ECR 产生的不确定的节点,随机从进化树 T 收缩 p 条边,随后未解决的节点的细化通过 NJ 方法来实现。 p -ECRNJ 的步骤包括:

a) 收缩阶段。随机选择 p 条边一次收缩,从而产生未解决的树 T^* 。

b) 细化阶段。包括以下两个步骤:(a) 折叠步骤,选择一个未解决的节点进行细化,树 T^* 以未解决的节点为根节点,分别折叠每个满足条件的子树为一个超级节点,条件为该子树的根节点是与根节点相邻的内部节点;(b) NJ 步骤,创建节点或折叠树 T^* 的超级节点的距离矩阵 M 。

算法3 p -ECRNJ^[43]

```
like = -1000000000.0 (a very small real number);
while ( $k_{\text{ecr}} > 0$ )
    randomly select  $p$  edges in  $T$  to contract all at once and produce unresolved tree  $T^*$ ;
    while ( $s_u > 0$ )
        select one unresolved node  $x$  in  $T^*$ ;
        root  $T^*$  at  $x$ ;
        collapse very subtree rooted at the sons of  $x$  into a supernode respectively;
        estimate the initial the distances dijbetween nodes or supernode in  $T^*$ 
```

according to the formula (4);

Initialize r as the remaining nodes in T^* ;

while ($r > 2$)

compute matrix Q by formula (1);

agglomerate i, j with $\min_{i,j} Q_{ij}$;

create the new node C ;

estimate the length of (C, i) and (C, j)

according to formula (2);

replace i and j with $C, r = r - 1$;

estimate d_{ck} for each remaining node k according to (3);

optimize the branches in T^* and compute the likelihood $f(T^*)$ of T^* ;

if ($f(T^*) > \text{like}$) $T_p \leftarrow T^*$; $\text{like} \leftarrow f(T^*)$;

$k_{\text{ecr}} \leftarrow k_{\text{ecr}} - 1$;

return T_p

图 15 给出了一个 2-ECRNJ 的图解。其中黑色节点表示超级节点,白色节点表示叶子节点或内部节点,而实线和虚线代表分支,虚线表示删除的分支,虚线圈内的节点表示要完善的节点。

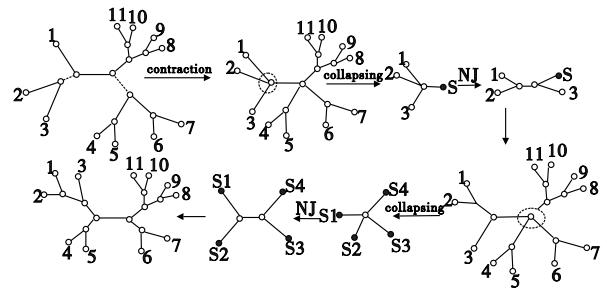


图15 一个2-ECRNJ的图示

在此之后,提出了一种构建系统发生树的启发式方法 PSOML。PSOML 利用 PSO 的基本结构更新由 p -ECRNJ 方法得到的粒子状态。算法中的 PSO 模型与其他方法^[44]相比,可以更快更少的代价得到较好的结果。

以下是文献[43]对 PSOML 算法的描述。可以看出,在 p -ECRNJ 中,参数 p 是通过 v_i^{k+1} 动态调整的,保证 PSOML 可以在一个合理大小空间中搜索。其中 p_i 和 p_g 分别表示粒子已经遇到的最优适应值的位置和群体中最优粒子的位置。速度 v_i 可看做下一代粒子的步长; v_i^{k+1} 表示粒子速度的更新公式,每一代中,每个粒子的速度是利用其遇到的最好位置和任意粒子所遇到的最优位置的信息来更新。

算法4 PSOML^[43]

```
input:  $n$  sequences
output: an evolutionary tree of the  $n$  sequences
1) initialization
    initialize parameters: the number  $M$  of particles in swarm, the maximum velocity  $v_{\text{max}}$  and the maximum generation times  $k_{\text{max}}$ ;
    initialize each particle in swarm;
    evaluate every particle;
    initialize  $p_i$  and  $p_g$ ;
2) if the termination conditions are met, goto 8);
3) calculate the speed  $v_i^{k+1}$  of every particle  $i$ ;
4) calculate  $v_i^{k+1}$  of every particle  $i$ ;
5) if the fitness values of  $v_i^{k+1}$  is larger than that of  $p_i$ , the replace  $p_i$  with  $v_i^{k+1}$ ;
6) if the fitness values of  $p_i$  is larger than that of  $p_g$ , then replace  $p_g$  with  $p_i$ ;
7) goto 3);
8) stop the algorithm and output  $p_g$ .
```

表 1 给出了 PSOML 与 GARLI 算法的比较。从表 1 中的最优结果来看,在 12 个数据集上 PSOML 都优于 GARLI;从平均最好结果来看,两种算法的相对性能跨数据集而变化。在数据集“101_sc”和“500_ZILLA”上,GARLI 优于 PSOML;然而在其他数据集上,PSOML 比 GARLI 要好。在平均运行时间方面,虽然 PSOML 比 GARLI 慢,但它们属于同一数量级。

表1 PSOML 和 GARLI 方法在不同数据集上的结果分析

数据集	GARLI				PSOML			
	best result	mean best	S. dev	mean time	best result	mean best	S. dev	mean time
1	-3918.480	-4412.513	604.677	00:05:14	-3843.046	-3847.477	2.964	0:07:01
2	-1080.739	-1080.740	0.000	00:02:07	-1080.722	-1080.722	0.000	00:02:15
3	-7409.052	-7410.554	2.008	00:08:25	-7390.299	-7392.758	1.738	00:06:46
4	-2833.833	-2833.845	0.012	00:08:25	-2830.424	-2831.607	1.157	00:20:30
5	-7177.114	-7177.170	0.0396	00:23:33	-7027.554	-7027.557	0.000	00:37:36
6	-22434.993	-22473.081	26.933	00:28:32	-21884.302	-22136.810	252.507	00:57:19
7	-65235.077	-65235.401	0.324	01:49:03	-65204.796	-65515.682	473.173	08:22:13
8	-43519.006	-43666.686	138.421	01:15:14	-43313.709	-43520.770	284.947	05:44:31
9	-70626.732	-71621.925	1272.949	02:42:18	-70570.837	-70929.878	202.302	13:45:27
10	-136208.720	-137121.387	1304.878	03:33:30	-135929.584	-135949.933	27.374	08:43:46
11	-117099.012	-117572.576	641.812	02:59:01	-117036.819	-117055.354	18.535	11:03:08
12	-19714.968	-19734.399	13.849	07:25:08	-19586.526	-19835.149	343.674	13:13:15

3 系统发生分析的新方法

3.1 系统发生网络

通常用系统发生树来代表一组分类单元的进化历史,这一模式极大地方便了对假设的讨论和检验,然而,这类模型描述更复杂的进化情况时比较差。与系统发生树相比,网络或许是群体水平(population-level relationships)上关系更好的表示。生命体系实际上是一种由不同的生物化学反应通路模块组成的分子网络系统,生物分子网络作为一种描述生物分子间相互作用关系的方法,在揭示生物体的生长、发育、衰老和疾病等生命系统的基本分子过程和规律中受到越来越多的重视。运用计算生物学的工具,开展生物分子网络分析的理论研究已经成为计算生物学的一个非常重要的研究方向。

在系统发生网络中,节点代表分类群,边表示它们之间的进化关系(在系统发生树中,边简称为分支),系统发生网络可以分为以下三种不同类型:系统发生树构成第一类;第二类是拆分网络,是由系统发生树的组合概括而获得,目的是表示数据集内部或之间的不兼容,如图16(a)所示;第三类是网状网络^[46~48],代表存在于网状事件,如杂交、基因水平转移或重组中的进化历史,如图16(b)所示。

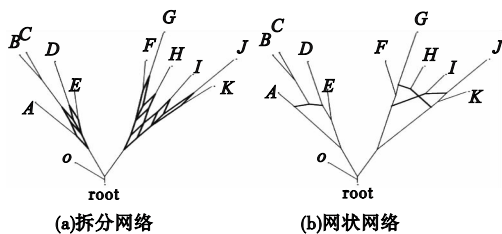


图16 系统发生网络

拆分网络提供了一个隐含进化关系图,平行边集合代表分裂,内部节点不必对应于假设的祖先。相比之下,网状网络提供了一个明确的进化关系图,在这样一个网络中,边代表后裔的谱系或者类似杂交、基因水平转移或重组等的网状事件,所有节点对应于假设祖先。

Huson 等人^[48]描述了拆分网络如何表示树的集合,并为一个网络中相互矛盾的信号是否是树状的引入了一个保守的统计检验。受 SplitsTree3 的启发,介绍了一个名为 SplitsTree4 的程序,即一个根据序列和距离推断不同类型系统发生网络的交互性、综合性的工具。图17是由 SplitsTree4 程序生成的拆分解网络^[48]图。

多个系统发生树的一个网状网络 N 可能有两个或两个以上父母(称为网状节点)的节点。有两种定义 N 的网状数目的

方法:a)将其定义为 N 中的网状节点的数目,在这种情况下,具有最小网状数目的一个网状网络被称为树的一个最佳 I 型网状网络;b)较好的方法是将其定义为 N 中的网状节点的父母的总数减去 N 中网状节点的总数^[47]。这种情况下,具有最小网状数目的一个网状网络被称为树的一个最佳 II 型网状网络。Chen 等人在文献[46]中首先提出了一个构建多系统发生树的一个或所有最佳 I 型网状网络的快速固定参数算法,然后利用此算法与其他思想一起获得一个新的算法来评估输入树的最佳 II 型网状网络的网状数目的下界。这是首个解决此问题的固定参数的算法。他们在 ANSI C 中实现了该算法,并得到 CMPT 和 MaafB 程序。实验表明,CMPT 可以快速地构建最佳 I 型网状网络,而与 Wu^[47]提出的先前的最好程序 PIRN 相比,MaafB 可以在更短的时间内构建最佳 II 型网状网络的更好的下界。另外,Chen 和 Collins 等人^[49,50]也在此方面做了重要的研究工作。

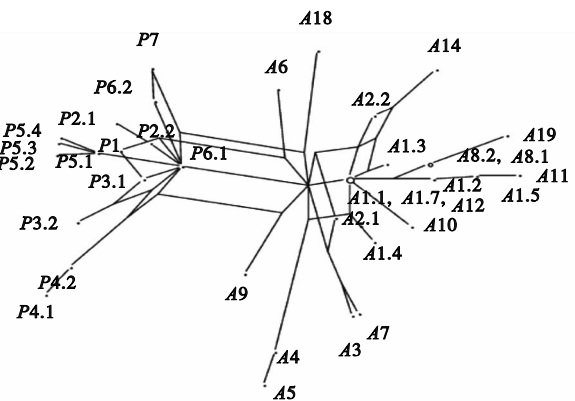


图17 拆分解网络

3.2 基于代谢路径构建系统发生树

蛋白质相互作用网络和代谢路径数据的出现为基于生物体内部的功能模块比对系统发生学研究提供了条件。其中,KEGG 代谢路径是由京都大学所提供的系统分析基因功能、联系基因组信息及功能信息的知识库,提供了一个操作平台,即以基因组信息(genes)和化学物质信息(ligand)为构建模块,通过代谢网络(pathway)将基因组和生物系统联系起来,然后根据功能等级进行归纳分类(brite)。许多学者利用 KEGG 代谢路径进行了研究工作。

目前,已经存在一些利用系统发生分析来处理代谢路径信息的方法。Tohsato 等人^[51]提出一种基于动态规划来计算代谢路径相似性的算法,但其缺陷是只能计算没有分支的代谢路径,不能应用在有分支或者更复杂的代谢路径比对中。Liao 等人^[52]提出将代谢路径中的所有子路径定义为一个集合,特定物种(organism)便可以表示为一个关于代谢路径的二进制向量 Profile,物种间的相似性根据该向量给出,不过实验结果与根据 16srRNA 所建立的进化树存在一定差距。Heymans 等人^[53]将 KEGG 代谢路径重组为不同的酶之间的相互作用图,把代谢路径相似性问题简化为酶图的相似性比对问题。以 Heymans 的思路为基础,Oh 等人^[54]利用图的核 kernel 的概念,提出改进算法来计算代谢路径间的相似性,该算法的局限性在于没有突出“非核”子图部分在计算代谢路径相似性中的作用。

赵建邦等人^[55]将代谢路径的比对问题进行抽象,将其转换为图之间的相似性问题,然后基于“核”概念计算子图间的相似性,最后基于相似性矩阵构建系统发生树,并用 Tree-View^[56]将其表示出来。实验结果证明,通过本方法构建的系

系统发生树更接近于 NCBI 分类法。

4 结束语

系统发生树的构建是生物信息学中一个极其复杂而又重要的问题,随着物种数目的增长,可能的系统发生树个数也呈指数级增长,因此,选择一种合适的构建方法非常有必要。本文首先介绍了传统的系统发生树构建方法。随后介绍了构建系统发生树的群智能算法,主要总结了近年来出现的包括蚁群算法、遗传算法和粒子群算法在内的各种构建系统发生树的方法。目前,蚁群算法的应用相对较多,有的是应用了类似 TSP 的解决方法;也有的方法是先利用物种集合构建一个图,然后利用蚁群遍历图中所有节点一次,最终得到一条可以代表系统发生树的最优路径;还有算法直接利用蚁群从物种集合中遍历,从而得到一个系统发生树的后缀表示。遗传算法构建系统发生树一般是从一组初始的拓扑结构出发,计算分支长度,若结果不是最优树,则通过改变树的拓扑结构来寻找最优树。此类算法的关键在于初始拓扑结构的优劣,以及选择算子和变异算子的选择。而 PSO 构建系统发生树的关键是粒子状态的更新方法,也即当前拓扑结构的改变方法。许多改变拓扑结构的方法既可适用于遗传算法,也可适应于粒子群算法。

本文第 3 章对系统发生分析的新方法——系统发生网络以及基于代谢路径构建系统发生树等问题作了简单介绍。生物分子网络分析的理论研究已经成为计算生物学的一个非常重要的研究方向,应利用相关算法构建基因网络来处理不完整的谱系排序,影响种群、亚种和早期物种之间的进化关系。而基于代谢路径构建系统发生树还有大量具有挑战的研究工作,首先是需要改善树型结构比对算法,因为通过现有方法构建的系统发生树是严格的二叉树,而通过 NCBI 分类法所构建的树是普通树,二者本身就存在较大差异,而且现有的基于 cousin pair 来判断树的相似性只是关注二叉树的局部分支,并没有在全局层次上关注子树的相似性;其次,由实验所得到的系统发生树需要考虑非叶子节点的意义,使系统发生树的构建更多地作为物种的演化提供理论依据。

大规模的系统发生树分析已在计算网络上进行^[57,58], Joshi 等人^[56]的工作利用分子钟假设和细胞自动机研究了 HIV 序列的进化。文献[58]采用细胞自动机预测未来系统发生树中的 DNA 序列,由于从一个祖先到一个后代的转换途径的可能性是巨大的,他们使用计算网络和中间件技术,探索大量用于基因突变的细胞自动机规则。

由于此问题是难以计算的,精确算法仅用于小群体的物种。在最大简约法中,计算时间增长如此之快,当序列数目增加时,实践中只能找到包含约 20 个序列的最佳解决方案。因此,构建系统发生树的并行算法的研究是有必要的,Blazewicz 等人^[59]已经在这方面做了一些工作,提出了并行自适应存储器编程算法,此算法是基于最大简约法和系统发生树构建的一些已知的近邻搜索算法,该算法实现了超线性加速,并找到了具有较高质量的解决方案。

参考文献:

- [1] LUO Wen-jian, CAO Xian-bin, WANG Xu-fa. An immune genetic algorithm based on immune regulation [C]//Proc of Congress on Evolutionary Computation. [S. l.]: IEEE Press, 2002: 801-806.
- [2] BODLAENDER H L, FELLOWS M J, WARNOW T Y. Two strikes against perfect phylogeny [C]//Proc of the 19th International Colloquium on Automata, Languages and Programming. Berlin: Springer-

Verlag, 1992: 273-283.

- [3] CUTELLO V, MORELLI G, NICOSIA G, *et al.* Immune algorithm with aging operators for the siring folding problem and the protein folding problem [C]//Proc of the 5th European Conference on Computation in Combinatorial Optimization. Berlin: Springer-Verlag, 2005: 80-90.
- [4] SOKAL R R, MICHENER C D. A statistical method for evaluating systematic relationships [J]. *University of Kansas Scientific Bulletin*, 1958, 38(22): 1409-1438.
- [5] SAITOU N, NEI M. The neighbor-joining method: a new method for reconstructing phylogenetic trees [J]. *Molecular Biology and Evolution*, 1987, 4(4): 406-425.
- [6] FITCH W M, MARGOLIASH E. Construction of phylogenetic trees [J]. *Science*, 1967, 155(760): 279-284.
- [7] DAY W, JOHNSON D, SANKOFF D. The computational complexity of inferring rooted phylogenies by parsimony [J]. *Mathematical Biosciences*, 1986, 81(1): 33-42.
- [8] ROCH S. A short proof that phylogenetic tree reconstruction by maximum likelihood is hard [J]. *IEEE/ACM Trans on Computational Biology and Bioinformatics*, 2006, 3(1): 92-94.
- [9] SNEATH P H A. Numerical taxonomy [C]//Bergey's Manual of Systematic Bacteriology. 2005: 39-42.
- [10] ROHLF F J. Classification of Aedes by numerical taxonomic methods [J]. *Annals of the Entomological Society America*, 1963, 56(6): 798-804.
- [11] LI Jing-yan. The pdaic method for constructing molecular evolutionary trees from sequences data [J]. *Zoological Research*, 1992, 13(4): 387-396.
- [12] FITCH W M. Toward defining the course of evolution: minimum change for a specified tree topology [J]. *Systematic Biology*, 1971, 20(4): 406-416.
- [13] PEARSON W R, ROBINS G, ZHANG Tong-tong. Generalized neighbor-joining: more reliable phylogenetic tree reconstruction [J]. *Molecular Biology & Evolution*, 1999, 16(6): 806-816.
- [14] DURBIN R, EDDY S R, KROGH A, *et al.* Biological sequence analysis: probabilistic models of proteins and nucleic acids [R]. Cambridge: Cambridge University Press, 1999.
- [15] FELSENSTEIN J. Evolutionary trees from DNA sequences: a maximum likelihood approach [J]. *Journal of Molecular Evolution*, 1981, 17(6): 368-376.
- [16] PENNY D. Towards a basis for classification: the incompleteness of distance measures incompatibility analysis, and phenetic classification [J]. *Journal of Theoretical Biology*, 1982, 96(2): 129-142.
- [17] BONABEAU E, DORIGO M, THERAULAZ G. Swarm intelligence: from natural to artificial systems [J]. *Journal of Artificial Societies and Social Simulation*, 2001, 4(1).
- [18] ENGELBRECHT A P. Fundamentals of computational swarm intelligence [M]. Hoboken: Wiley, 2005.
- [19] BENI G, WANG Jing. Swarm intelligence in cellular robotic systems [C]//Proc of NATO Advanced Workshop on Robots and Biological Systems. Berlin: Springer, 1993: 703-712.
- [20] BONABEAU E, DORIGO M, THERAULAZ G. Swarm intelligence: from natural to artificial systems [M]. Oxford: Oxford University Press, 1999.
- [21] DORIGO M, MANIEZZO V, COLON A. Ant system: optimization by a colony of cooperating agents [J]. *IEEE Trans on Systems, Man, and Cybernetics-Part B*, 1996, 26(1): 29-41.
- [22] COLON A, DORIGO M, MANIEZZO V. Distributed optimization by ant colonies [C]//Proc of the 1st European Conference on Artificial Life. 1991: 134-142.
- [23] DORIGO M, GAMBARDELLA L M. Ant colony system: a cooperative

- learning approach to the traveling salesman problem[J]. *IEEE Trans on Evolutionary Computation*, 1997, 1(1): 53-66.
- [24] KOROSTENSKY C, GONNET G H. Using traveling salesman problem algorithms for evolutionary tree construction[J]. *Bioinformatics*, 2000, 16(7): 619-627.
- [25] ANDO S, IBA H. Ant algorithm for construction of evolutionary tree [C]//Proc of Congress on Evolutionary Computation. San Francisco: Morgan Kaufmann Publishers, 2002: 1552-1557.
- [26] GONNET G, KOROSTENSKY C, BENNER S. Evaluation measures of multiple sequence alignments[J]. *Journal of Computational Biochemistry*, 1999, 7(1-2): 261-276.
- [27] KUMNORKAEW M, KU Hong-ming, RUENGLERTPANYAKUL P. Application of ant colony optimization to evolutionary tree construction [C]//Proc of the 15th Annual Meeting of the Thai Society for Biotechnology. 2004.
- [28] PERRETTO M, LOPES H S. Reconstruction of phylogenetic trees using the ant colony optimization paradigm[J]. *Genetics and Molecular Research*, 2005, 4(3): 581-589.
- [29] VITTORI K, DELBEM A C B, PEREIRA S L. Ant-based phylogenetic reconstruction (ABPR): a new distance algorithm for phylogenetic estimation based on ant colony optimization[J]. *Genetics and Molecular Biology*, 2008, 31(4): 974-981.
- [30] CHEN Ling, QIN Ling, LIU Wei, *et al.* A phylogenetic tree constructing method based on ant colony partitioning[C]//Proc of the 1st International Conference on Information Science and Engineering. 2009: 758-761.
- [31] QIN Ling, CHEN Yi-xin, PAN Yi, *et al.* A novel approach to phylogenetic tree construction using stochastic optimization and clustering [J]. *Bioinformatics*, 2006, 7(4): S24.
- [32] CATANZARO D, PESENTI R, MILINKOVITCH M C. An ant colony optimization algorithm for phylogenetic estimation under the minimum evolution principle[J]. *BMC Evolutionary Biology*, 2007, 7(1): 228.
- [33] LEWIS P O. A genetic algorithm for maximum-likelihood phylogeny inference using nucleotide sequence data [J]. *Molecular Biology Evolution*, 1998, 15(3): 277-283.
- [34] REIJMERS T H, WEHRENS R, DAEYAERT F D, *et al.* Using genetic algorithms for the construction of phylogenetic trees: application to G-protein coupled receptor sequences[J]. *BioSystems*, 1999, 49(1): 31-43.
- [35] MATSUDA H. Construction of phylogenetic trees from amino acid sequences using a genetic algorithm[C]//Proc of Genome Informatics Workshop. 1995: 19-28.
- [36] POLADIAN L, JERMIIN L S. What might evolutionary algorithms (EA) and multi-objective optimisation (MOO) contribute to phylogenetics and the total evidence debate? [C]//Proc of Genetic and Evolutionary Computing Conference. 2004.
- [37] SKOURIKHINE A. Phylogenetic tree reconstruction using self-adaptive genetic algorithm[C]//Proc of the 1st IEEE International Symposium on Bioinformatics and Biomedical Engineering. 2000: 129-134.
- [38] FISCHER S W, LINDBERG M P. Phylogenetic inference using genetic algorithm-based least squares methods[D]. Aarhus: Aarhus University, 2006.
- [39] 李军令, 赵宏伟, 马志强, 等. 基于遗传算法的最大似然法构建系统发生树[J]. *东北师大学报: 自然科学版*, 2008, 40(1): 36-39.
- [40] KENNEDY J, EBERHART R. Particle swarm optimization [C]//Proc of IEEE International Conference on Neural Network. Piscataway: IEEE Service Center, 1995: 1942-1948.
- [41] DORIGO M, Di CARO G. The ant colony optimization meta-heuristic [M]//New Ideas in Ptimization. London: McGraw-Hill, 1999: 11-32.
- [42] LV Hui-ying, ZHOU Wen-gang, ZHOU Chun-guang. A discrete particle swarm optimization algorithm for phylogenetic tree reconstruction [C]//Proc of International Conference on Machine Learning and Cybernetics. 2004: 2650-2654.
- [43] LI Jian-fu, GUO Mao-zu. A new approach to evolutionary tree reconstruction combining particle swarm optimization with *p*-ECR[J]. *International Journal of Computational Intelligence Research*, 2008, 4(2): 187-195.
- [44] SHI Y, EBERHART R C. Empirical study of particle swarm optimization [C]//Proc of Congress on Evolutionary Computation. 1999: 1945-1950.
- [45] ZWICKL D J. Genetic algorithm approaches for the phylogenetic analysis of large biological sequence datasets under the maximum likelihood criterion[D]. Austin: University of Texas, 2006.
- [46] CHEN Zhi-zhong, WANG Lu-sheng. Algorithms for reticulate networks of multiple phylogenetic trees[J]. *IEEE/ACM Trans on Computational Biology and Bioinformatics*, 2012, 9(2): 372-384.
- [47] WU Yu-feng. Close lower and upper bounds for the minimum reticulate network of multiple phylogenetic trees [J]. *Bioinformatics*, 2010, 26(12): 140-148.
- [48] HUSON D H, BRYANT D. Application of phylogenetic networks in evolutionary studies [J]. *Molecular Biology Evolution*, 2006, 23(2): 254-267.
- [49] CHEN Zhi-zhong, WANG Lu-sheng. HybridNET: a tool for constructing hybridization networks[J]. *Bioinformatics*, 2010, 26(22): 2912-2913.
- [50] COLLINS L, LINZ S, SEMPLE C. Quantifying hybridization in realistic time [J]. *Journal of Computational Biology*, 2011, 18(10): 1305-1318.
- [51] TOHSATO Y, MATSUDA H, HASHIMOTO A. A multiple alignment algorithm for metabolic pathway analysis using enzyme hierarchy [C]//Proc of International Conference on Intelligent Systems for Molecular Biology. Menlo Park: AAAI Press, 2000: 376-383.
- [52] LIAO Li, KIM S, TOMB J F. Genome comparisons based on profiles of metabolic pathways[C]//Proc of the 6th International Conference on Knowledge-based Intelligent Information and Engineering Systems. 2002: 469-472.
- [53] HEYMANS M, SINGH A K. Deriving phylogenetic trees from the similarity analysis of metabolic pathways [J]. *Bioinformatics*, 2003, 19(1): 138-146.
- [54] OH S J, JOUNG J G, CHANG J H, *et al.* Construction of phylogenetic trees by kernel-based comparative analysis of metabolic networks [J]. *BMC Bioinformffics*, 2006, 7: 284-295.
- [55] 赵建邦, 高琳, 宋佳. 一种基于代谢路径构建系统发生树的有效方法[J]. *电子学报*, 2009, 37(8): 1633-1638.
- [56] PAGE R D M. TreeView: an application to display phylogeneic trees on personal computers [J]. *Computer Applications in the Biosciences*, 1996, 12(4): 357-358.
- [57] JOSHI Y, VADHIYAR S. Analysis of DNA sequence transformations on grids [J]. *Journal of Parallel and Distributed Computing*, 2009, 69(1): 80-90.
- [58] KATARIYA P R, VADHIYAR S S. Phylogenetic predictions on grids [C]//Proc of the 5th IEEE International Conference on e-Science. Washington DC: IEEE Computer Society, 2009.
- [59] BLAZEWCZ J, FORMANOWICZ P, KEDZIORA P, *et al.* Adaptive memory programming: local search parallel algorithms for phylogenetic tree construction [J]. *Annals Operations Research*, 2011, 183(1): 75-94.
- [60] MILLONAS M M. Swarms, phase transitions, and collective intelligence[M]//Artificial Life III. Reading: Addison Wesley, 1994.