doi:10.3969/j.issn.1672-5565.2013.04.13

拽线法:一个构建系统发育树的新算法

陈兆斌
(西北农林科技大学 生命学院,陕西 杨凌,712100)

摘 要:这篇文章要讨论的拽线法(DL)是贪婪算法的一种。和Fitch-Margoliash(FM)一样,DL也是基于距离矩阵构建系统发育树,但是和FM算法相比,DL具有低复杂度、较高的容错性和准确度高的优点。当存在误差时,DL算法只是加大了不在同一个父节点下的基因序列的距离,但能够准确的判断序列的亲缘关系,进而得到完美的进化树拓扑结构;相比之下,FM算法让各个基因序列间的距离均摊了这种误差,从而有可能将本应该具有相同父节点的基因序列分到不同的分支。

关键词:系统发育树;基因组进化;序列分析;算法
中图分类号:344 **文献标识码**:A **文章编号**:1672-5565(2013)-04-317-04

DragLine(DL) method:a new algorithm to built phylogenetic tree

CHEN Zhao-bin
(*College of Life Science*, *Northwest A&F University*, *Yangling Shanxi* 712100, *China*)

Abstract: The DL Method we will present in this paper is one kind of Greedy Algorithm . As Fitch-Margoliash (FM), DL creates the phylogenetic tree based on distance matrix, but has lower computational complexity (the complexity of DL is O(nlgn), the complexity of FM is $O(n^2)$). It is much faster, more accurate and better fault-tolerant. When bias exists, DL method increases the distance between two nodes not sharing the same parent node., This results it a perfect method when it comes to the topological structure of phylogenetic tree. FM algorithm, by contrast, uses the average distance of one node to all the other nodes, which may assign two nearest gene sequences which share the same parent node into different branches.

Keywords: Phylogenetic Tree; Genome Evolution; Sequence Analysis; Algorithm

Built and analysis of phylogenetic tree is an important branch of bioinformatics. Study phylogenetic tree can rebuild ancestral sequence and estimate the time differences. There are a lot of papers about the methods of how to build phylogenetic tree, such as the paper we have refered to in References [1-9]. These methods have both their advantages and disadvantages. It is still an open question whether there is a kind of optimal solution.

1 Introduction to DL

DL is based on the fact that if two nucleotide chains are the nearest in a distance matrix, they will have the same parent node in phylogenetic tree.

The steps are as follows:

Step 1:

Find the nearest two points in a distance matrix served as base points to build the phylogenetic tree. Assume that the two points are P_1 and P_2. (Fig.1)

Fig.1 Step1

Step 2:

Loop execute the following process:

Find the nearest distance between points that have been added to phylogenetic tree and points that have not yet been added to phylogenetic tree in the distance

收稿日期:2013-06-03;修回日期:2013-09-02.
作者简介:陈兆斌,男,本科生,研究方向:生物信息学;E-mail:18700808256@163.com.

matrix. Assume that the two points making up the nearest distance are Pi and P_{i+1}. In order to add P_{i+1} to phylogenetic tree, we use P_{i+1} to drag the line which link Pi with the tree. Assume the drag point is O_{i+1}, then we get three new edges: L1, L2, L3 (Fig.2). We use three equations to work out the length of L1, L2, L3. (Fig.3)

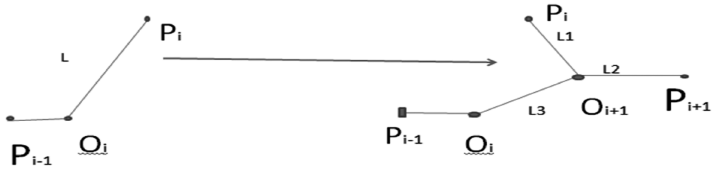


Fig.2 Step2

Loop until all the points are added into phylogenetic tree, then we will get the phylogenetic tree just as we want.

$$\begin{cases} L1+L2=P_iP_{i+1} \\ L1+L3=P_iO_{i+1} \\ L2+L3+P_{i-1}O_i=P_{i-1}P_{i+1} \end{cases}$$

Fig.3 three equations

2 Anexample of how to use DL to build the phylogenetic tree

Follow is a distance matrix(Fig.4):

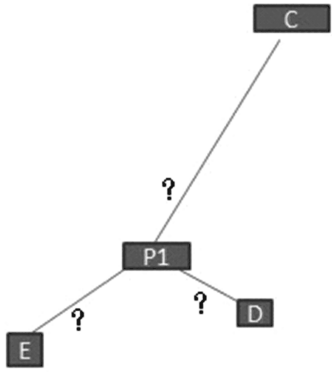
	A	B	C	D	E
A		22	39	39	41
B			41	41	43
C				18	20
D					10
E					

Fig.4 an example distance matrix

Step 1: Find the nearest two points in the distance matrix served as base points to build the phylogenetic tree. Then we get points D, E. DE = 10. (Fig.5)

Fig.5 step1

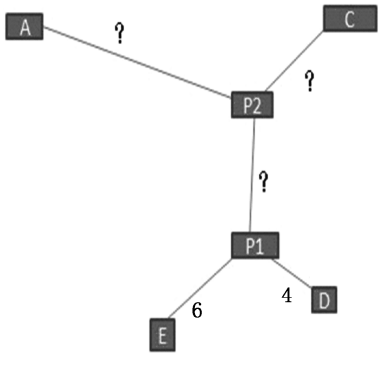
Step 2: Find the nearest distance between points (D, E) that have been added to the phylogenetic tree and points (A, B, C) that have not yet been added to the phylogenetic tree in the distance matrix. The two points are C and D, Use C to drag DE and assume the drag point is p1, then we get three equations. We can work out DP1 = 4 CP1 = 14 EP1 = 6 (Fig.6)



$$\begin{cases} DP1+CP1=DC=18 \\ DP1+EP1=DE \\ EP1+CP1=EC=20 \end{cases}$$

Fig.6 step2

Step 3: Find the nearest distance between points (C, D, E) that have been added to the phylogenetic tree and points (A, B) that have not yet been added to phylogenetic tree in distance matrix. The two points are A and C, Use A to drag CP1, assume the drag point is P2, then we get three equations. We can work out CP2 = 9 P1P2 = 5 AP2 = 30. (Fig.7)

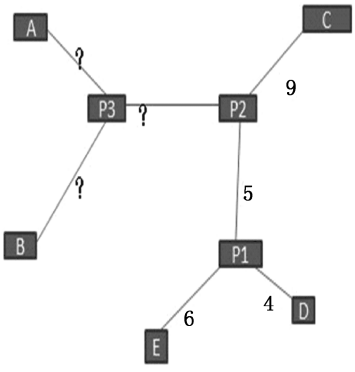


$$\begin{cases} CP2+AP2=CA=39 \\ DP1+P1P2+AP2=AD=39 \\ CP2+P1P2=CP1=14 \end{cases}$$

Fig.7 step3

Step 4: Find the nearest distance between points (A, C, D, E) that have been added to the phylogenetic

tree and points(B) that have not yet been added to the phylogenetic tree in the distance matrix. The two points are B and A. Use B to drag AP2 and assume the drag point is P3, then we get three equations. We can work out AP3 = 10 BP3 = 12 P3P2 = 20. (Fig.8)



$$\begin{cases} AP3+BP3=AB=22 \\ AP3+P3P2+P2C=AC=39 \\ BP3+P3P2+P2C=BC=41 \end{cases}$$

Fig.8　step4

Finally we get a perfect phylogenetic tree(in this case, the result is the same as FM).(Fig.9)

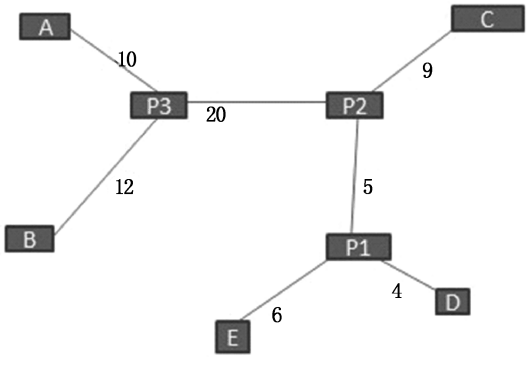


Fig.9　the final result

3　State the inherent defects of algorithm using distance matrix, and then analysis which one is a better algorithm, DL or FM?

There are a lot of papers focusing on the incompleteness of distance measures[10]. Distance matrix is derived from the hamming distance between two DNA stands. Due to that DNA has four types of bases(A, T, G, C) and DNA mutation is random, if we have two chains A, B, We cannot make sure every other chain C promises (AC−BC) is a constant. But in a perfect phylogenetic tree, it must be a constant if they have the same parent node.

For example, in the case below (Fig. 10), we cannot get a perfect phylogenetic tree matter we use FM or DL.

A:CGAGGCATTTCATGAGCTCTAGGCTTAATATCGATCATCGGGATC
B:CGAGACATTCCAGGAGCTCTAGGCTTAATATCGATCATCGGGATC
C:CCAGGCATTTCA TGAACTCTAGACTTAATATTGATAATCGGGATA
D:CCAGGCATTTCAGGAACTCTAGACTTAATATTGTTCATAGGCATG

Fig.10　example sequence

Notes: Among the four chains, A is the original one. On base of A, B has different 3 bases, and C further has other 6 bases not the same with B. On Base of C, D has 6 different bases, but one base change is the same as B.

Table 1 is the Distance matrix for the question. We can see AB = 3. And AB is the nearest. As BC−AC = 8−5 = 3 not equals with BD−AD = 1, then we cannot get a perfect phylogenetic tree. Then which algorithm is more accurate to the reality, FM or DL?

A lot people may think that because FM uses the average distance, it may decrease the error. But the reality is just the opposite.

We can see that when we use FM (Fig. 11), we may think A, B have the same ancestor and C, D have the same ancestor. This is not consistent with the actual situation.

But when we use DL (Fig. 12), we can get the information that C and B are evolved from A in two different directions, and D is evolved based on C thus farther to A and B. this fits perfect to the reality.

Table 1 The distance matrix for the question

	A	B	C	D
A		3	6	12
B			9	13
C				6
D				

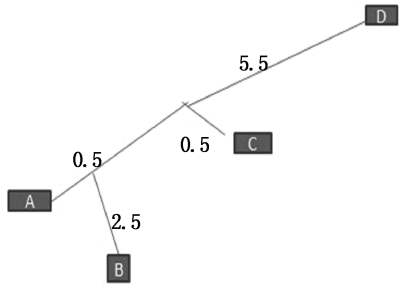


Fig. 11 The answer we use FM

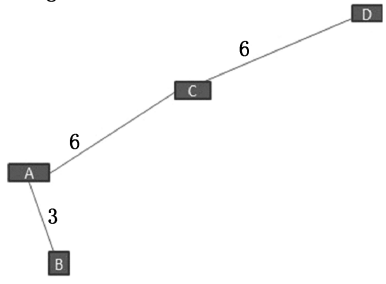


Fig. 12 The answer we use DL

How comes this? After analysis, I find that it is just because DL use the nearest distance, by contrast FM which uses the average distance. When bias exists, DL method only increases the distance between two nodes not sharing the same parent node. But when it comes to the topological structure of the phylogenetic tree, this method is perfect. FM algorithm, by contrast, uses the average distance of one node to all the other nodes, which may assign two nearest gene sequences sharing the same parent node into different branches.

4 Conclusions and Discussions

Besides accuracy, DL has other advantages over FM. DL has less operation steps, less memory consumption and better fault-tolerant. DL is a better way to build the phylogenetic tree. But it still needs more work to be done by biologists and computists lovers to find out if this method really. works well in practice.

参考文献(References)

[1] Yves Pauplin. Direct Calculation of a Tree Length Using a Distance Matrix[J], Journal of Molecular Evolution, 2000, 51(1): 41-47.

[2] 吕宝忠.分子进化树的构建[J].动物学研究, 1993, 14(2): 186-193.

[3] 李建伏, 郭茂祖, 系统发出树构建技术综述[J].电子学报, 2006, 34(11): 2047-2052.

[4] 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, (08): 1633-1638.

[5] 李军令, 赵宏伟, 马志强, 魏利, 冯嘉, 关伟州. 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, (01): 36-38.

[6] 李建伏, 郭茂祖, 刘扬. 一种基于 Quartet Puzzling 和邻接法的进化树构建算法[J]. 计算机研究与发展, 2008 (11): 1965-1973.

[7] 李玉鑑, 徐立业. 不加权算术平均组对方法的改进及应用[J]. 北京工业大学学报, 2007, (12): 1333-1339.

[8] 李刚成, 刘赞波, 曾庆光. 一种基于模糊聚类的构造进化树方法[J]. 计算机应用, 2009, (03): 836-838.

[9] 谭亚芳, 金人超, 一种基于 NJ 的高效构建系统进化树算法[J]. 计算机工程与应用, 2004, (21): 83-86.

[10] 常青, 周开亚.分子进化研究中系统发生树的重建[J].生物多样性, 1998, 6(01): 55-61.