基于关键字树的DNA多序列星比对算法

邹 权,郭茂祖,王晓凯,张涛涛
(哈尔滨工业大学计算机科学与技术学院,黑龙江哈尔滨 150001)

摘 要: 在构建进化树、比较单体型序列等生物信息学研究中,需要比对多个相似程度很高的DNA序列.对于数量多、序列长的多序列比对问题,通常使用时间复杂度较低的星比对算法.然而在处理大规模数据时,星比对的平方时间复杂度依然不能满足需要.因此,在星比对思想的基础上,本文结合关键字树理论,先找出完全匹配的区域,然后比对剩余区域,以达到降低期望时间复杂度的目的.两组实验证明了本文算法的有效性,在取得相同比对效果的情况下,本文算法运行时间小于其他方法.

关键词: 多序列比对; 星比对; 关键字树; Aho-Corasick算法; 生物信息学
中图分类号: TP301 **文献标识码:** A **文章编号:** 0372-2112 (2009) 08-1746-05

An Algorithm for DNA Multiple Sequence Alignment Based on Center Star Method and Keyword Tree

ZOU Quan, GUO Mao-zu, WANG Xiao-kai, ZHANG Tao-tao
(*School of Computer Science and Technology, Harbin Institute of Technology, Harbin, Heilongjiang* 150001, *China*)

Abstract: Multiple sequence alignment is necessary and important for reconstructing evolutionary trees and comparing haplotype sequences. Center star method is always used to deal with lots of long sequences. However, square time complexity is a bottleneck for large data. In this paper, we propose a novel keyword tree based algorithm for improving the center star method. Aho-Corasick algorithm is employed to match a set of substrings and the rest regions are aligned by dynamic programming. Experiments show that the improved method runs faster than the initial center star method and clustalx.

Key words: multiple sequence alignment; center star method; keyword tree; Aho-Corasick algorithm; bioinformatics

1 引言

在生物信息学中,最基本的信息处理方法是对序列数据进行比对.它对于发现生物序列中的功能、结构和进化信息具有非常重要的意义.从片断组装、进化分析、单体型比较到RNA结构预测,都是以序列比对为基础的,特别是多序列比对(Multiple Sequence Alignment, MSA)的效率,直接影响到这些问题的解决效果.因此设计一个合理高效的序列比对算法已成为生物信息学领域一个非常重要的研究课题.

序列比对是指给定两个或多个字符串,通过对每个字符串分别添加、删除字母或加入空格,使得给定的字符串具有最大的相似性.求解多序列比对一般以双序列比对为基础.目前双序列比对问题可以用动态规划的方法求得其最优解,但多序列比对问题至今仍是生物信息学中棘手的问题之一.已经证明多序列比对的最优解问题是NP完全问题[1],所以目前大多是应用启发式算法或组合优化算法求得其近似最优解.

启发式算法一般依靠迭代策略,即基于一种产生比对的方法,通过迭代的方式优化多序列比对的结果.Davie M提出用粒子群优化算法解决MSA问题,Ikeda T等提出了基于A*的启发式算法,许多研究者利用遗传算法处理MSA,Bimey E率先提出了用隐马尔可夫模型解决MSA问题[2,3].这些算法一般都具有鲁棒性并且对序列的个数不敏感,但也存在不足,比如粒子群算法的结果不稳定,其优劣程度取决于随机数的选取,A*算法运行时间过长,遗传算法容易陷入局部极优.

组合优化算法的思想是将多序列比对转化为双序列比对来解决,按照其转化策略可以分为树比对算法和星比对算法.树比对算法是对给定一个多序列的序列集合 $S=\{s_1,\cdots,s_q\}$,找到一棵有 q 个叶结点的进化树,并与序列 $s_1,\cdots,s_q$ 建立起一一对应的关系,通过对进化

收稿日期:2007-07-09;修回日期:2009-03-31
基金项目:国家自然科学基金(No.60671011,No.60741001,No.60871092);黑龙江省杰出青年科学基金(No.JC200611);黑龙江省自然科学重点项目基金(No.ZJG0705)

树的内部节点指派序列,计算每条边的计分值.进化树所有边的计分和是树的计分.树比对的目标就是要找到一个能使该计分值最大的序列指派,并通过进化树和其节点上的序列指派得到最终的比对结果.星比对可以看作是树比对的一种特例,即进化树只有一个内部节点和 q 个叶结点.指派给内部节点的序列被称作中心序列[4].

树比对问题在限定计分模式和字母表大小时是 NP 难题[1],存在一个发现优化解的算法,但它与序列的个数呈指数关系,因此在对大量的序列进行比对时,如果想要快速得到近似最优解,通常使用星比对算法.但无论多个序列的相似程度如何,星比对的时间开销均与序列的个数的平方和序列平均长度的平方成正比.如果序列的相似程度很高,那么各个序列之间在局部会存在子字符串完全匹配的情况.由于动态规划的时间开销与序列长度呈平方关系,而字符串匹配与序列长度呈线性关系,如果完全匹配的子字符串在比对序列中占有很大的比重,那么在将多序列比对转化为双序列比对之后,只需要利用动态规划比对剩余未匹配部分即可,从而大大提高了星比对算法的运行效率.本文将利用这种"先匹配,再比对"的思想改进星比对算法,将期望时间复杂度降为线性.

2　关键字树理论和 Aho-Corasick 搜索算法

关键字树理论和 Aho-Corasick 算法是为了在线性时间内解决如下问题:给定一个长字符串 T 和一个短字符串集合 $P=\{P_1,P_2,\cdots,P_z\}(z\in N,z>1)$,在 T 中找出所有 $P_i(i=1,2,\cdots,z)$的出现.该问题的解决是通过对集合 P 构造关键字树[1],记为 K,然后在 T 中利用 K 进行搜索.其中搜索的算法即 Aho-Corasick 算法.利用该方法在 T 中搜索出所有 P_i 的出现所花费的时间是 $O(m+n+k)$,其中 $m=|T|,n=\sum|P_i|$,k 是 T 中所有的 P_i $(i=1,2,\cdots,z)$出现次数之和.

2.1　关键字树理论介绍

关键字树的定义如下:

定义 1　集合 $P=\{P_1,P_2,\cdots,P_z\}(z\in N,z>1)$的关键字树是一棵有根树,它的根记为 K,满足

(1)每一条边都明确的标定一个字母.

(2)从同一结点分开的任意两条边对应着不同的字母.

(3)每一个模式(*pattern*)$P_i(i=1,2,\cdots,z)$都对应着一个结点 v,从树根 K 出发到达 v 的路径可以恰好正确地拼出字符串 P_i,并且树 K 的每一个叶结点都对应 P 的某一个模式.

例　$P=\{potato,tattoo,theater,other\}$,关键字树如图 1 所示.

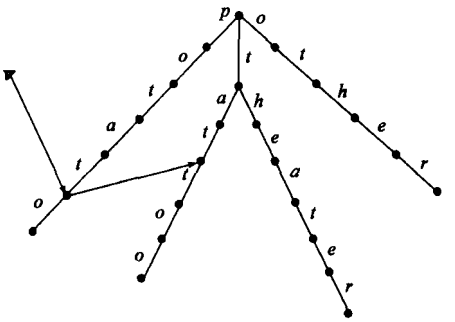


图1　集合P对应的关键字树和一例失效链接

令 $L(v)$表示从根结点到结点 v 的字符连接起来得到的字符串.

定义 2　$lp(v)$表示 $L(v)$中最长后缀的长度,该后缀是 P 中某个模式的前缀.在关键字树中从根节点开始寻找该前缀,结束时的结点记为 n_v.当 $lp(v)=0$ 时,$n_v=K$.有序对(n,n_v)称为一个失效链接(failure link)[6].

显然上例中 $L(v)=potat$,$lp(v)=|tat|=3$,v 结点的失效链接如图 1 所示.失效链接的建立是 Aho-Corasick 算法加速搜索的关键,它将原本多项式时间的搜索开销降低到线性.在线性时间内得到一棵关键字树的所有失效链接是关键字树理论中的核心.可以采用归纳的思想:若 v 是根 r,或 v 是 r 的儿子,则 $n_v=r$;假设距根结点的距离小于等于 k 的所有结点 v 的 n_v 都已经得到,我们现在求距 r 的距离为 $k+1$ 的结点 v 的 n_v.

假设所求的结点为 v(距 r 为 $k+1$ 的结点),他的父亲是 v',v'到 v 这条边所标定的字母是 x.①若 $n_{v'}$的下一个字符有 x,设该边的另一个结点为 w',即有 $n_v=w'$;②若 $n_{v'}$到它所有儿子的边所标定的字母均不是 x,则字符串 $L(n_v)$是 $L(n_{v'})$的一个后缀跟上 x,又因为该后缀与根结点开始的字符串匹配(类似前缀),所以可以检测 $n_{n_{v'}}$后面是否有 x,如果没有则继续下去,直到后面出现 x 或找到根节点结束.

2.2　Aho-Corasick 搜索算法

在对关键字树建立完所有的失效链接后,可以利用 Aho-Corasick 算法在线性时间内搜索出所有 $P_i(i=1,2,\cdots,z)$的出现.

Aho-Corasick 搜索算法

输入:P 的关键字树以及所有的失效链接

输出:所有 P_i 在 T 中出现的位置

定理 1　利用关键字树和 Aho-Corasick 算法在 T 中搜索集合 P 中所有元素的时间开销是 $O(m+n+k)$.其中预处理阶段的时间复杂度是 $O(n)$,搜索阶段的时间复杂度是 $O(m+k)$[4].

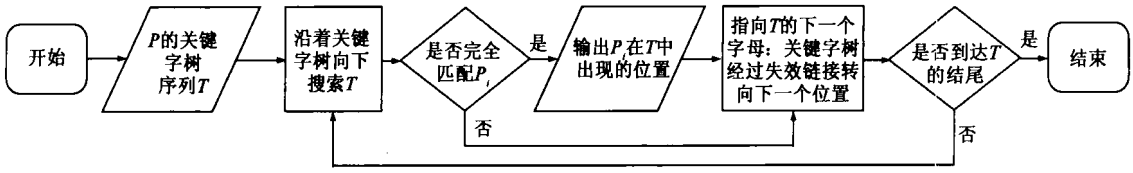


图2 Aho-Corasick算法流程图

3 改进的星比对算法

3.1 改进算法的主要思想

星比对算法的思想是利用中心序列将多序列比对问题转化为双序列比对问题.原始星比对算法的过程是:先选取中心序列,再将中心序列与其他序列依次比对,然后按照"一旦为空格,始终为空格"的思想将所有比对得到的空格汇入中心序列,同时将新添入的空格加入其他比对后的序列,进而完成多序列比对[9].中心序列选取和双序列比对过程中用的都是动态规划方法,其平方级的时间复杂度影响了整个算法的执行效率.然而对于相似程度很高的两个序列,有较长的子字符串是完全匹配的,那么只需要用动态规划方法比对剩余部分即可完成双序列比对.利用关键字树理论搜索完全匹配子字符串的时间是线性的,从而节省了期望时间开销.

基于上述想法,为了降低原始星比对算法的期望运行时间,本文对原始星比对算法进行改进.分割某个序列得到一个子字符串的集合,为该集合构建关键字树,依次在其他的序列中利用 Aho-Corasick 算法搜索该集合中子字符串的出现,记录所有出现的次数总和.选取次数总和最大的被分割序列作为中心序列,然后依次比对中心序列和其他序列之间未匹配部分,按照"一旦为空格,始终为空格"的思想汇总,完成多序列比对.改进之处在于:(1)利用分割的思想和 Aho-Corasick 算法来寻找中心序列,降低了时间复杂度;(2)在比对中心序列和其他序列的过程中,将避开 Aho-Corasick 算法搜索出的完全匹配的子字符串,只比对剩余部分,提高了效率.

3.2 改进算法的详细步骤

首先要寻找中心序列.设需要进行比对的多个序列分别是 $P_1,P_2,\cdots,P_n$.对于每个序列 P_i,分别进行如下操作:

(1)将 P_i 分割成等长的 k 段(如果最后一段长度不够,可忽略),令每段的长度 $r=\lfloor n/k \rfloor$.将这 k 个子字符串分别记为 $P_{i1},P_{i2},\cdots,P_{ik}$,对它们组成的集合建立一棵关键字树 T_i.这里 k 一般取值 $O(n/\log n)$时,效果最好[7].

(2)在所有的 $j\neq i$ 的 P_j 中利用 Aho-Corasick 算法搜索 T_i,记出现的子字符串的个数是 n_{ij},那么 P_i 与其他序列的相似程度可以用 $N_i=\sum_{j=1,j\neq i}^{n} n_{ij}$ 来衡量.

对于 $i=1,2,\cdots,n$,比较所有的 N_i,最大 N_i 对应的序列 P_c 就是中心序列.值得注意的是,本文将 DNA 序列视为随机序列,并没有考虑子串的相对顺序问题,因此本文算法适合应用于构建进化树、比较单体型序列等假设列之间相互独立的问题,而不适合于基因组重排等考虑字符串前后关联的问题.

(3)得到中心序列后,将 P_c 依次与 $P_i(i=1,2,\cdots,n$ 且 $i\neq c)$进行比对.先考察上一步用 Aho-Corasick 算法在 P_i 中搜索出的 P_c 的子字符串编号及在 P_i 中出现的位置,将这些完全匹配的子字符串分别在 P_c 和 P_i 中标记.这些标记将 P_c 和 P_i 分割成若干段,然后用动态规划方法依次对分割出来未被标记的段进行比对.分别记录需要在 P_c 和 P_i 中插入空格的位置,记为 S_{ci} 和 S_i.

(4)依次比对完 P_c 与 $P_i(i=1,2,\cdots,n$ 且 $i\neq c)$后,对所有需要插入空格的位置 $S_{ci}(i\neq c)$汇总,汇总后的位置记为 S_c,分别比较 S_c 和 $S_{ci}(i=1,2,\cdots,n$ 且 $i\neq c)$,将新汇入的空格位置加入到 S_i 中.依次按照 $S_i(i=1,2,\cdots,n)$中记录的位置将空格插入到 P_i 并输出,得到多序列比对结果.

3.3 算法复杂性分析

假设序列 $P_1,P_2,\cdots,P_n$ 的长度均为 m,建立关键字树的时间复杂度是 $O(m)$,在其他的字符串中搜索并计算 N_i 所花费的时间是 $O(nm)$,那么,对于 $i=1,2,\cdots,n$ 计算出所有的 N_i 并找出中心序列所花费的时间就是 $O(n^2m)$.因 m 较大,该运行时间远远小于原始星比对算法寻找中心序列的时间 $O(n^2m^2)$.

选出中心序列之后,还要作 n-1 次双序列比对,花费的时间是 $O(nm^2)$.由于在寻找中心序列时利用 Aho-Corasick 算法已经搜索出部分匹配的子字符串,因此只需要比对剩余部分.如果两个序列相似度高,则需要比对的子字符串长度远小于 m,花费的时间会大大降低.当多个序列的相似程度非常高时,相对于寻找中心序列,该 $n-1$ 次双序列比对的时间开销可以忽略不计.

将中心序列与其他序列比对完之后,还要对插入空格的位置汇总,输出比对结果.汇总空格插入位置的时间开销不大,每次只需要比较 m 个位置即可,因此汇总空格位置的时间花费是 $O(nm)$.根据空格位置输出比对结果的时间花费也是 $O(nm)$.

因此,本算法的最坏时间复杂度是 $O(n^2m)+O(nm^2)$.由于 n 远远小于 m,可以认为最坏时间复杂度为 $O(nm^2)$,好于原始星比对算法的 $O(n^2m^2)$.在多个序列的相似程度很高的情况下,期望运行时间可以控制在 $O(n^2m)$内.由于 nm 表示多个序列的长度总和,在序列个数远小于序列长度且相似度高的情况下,可以认为该期望运行时间与多个序列的长度和呈线性关系.

4 实验结果对比和分析

多序列比对问题在生物信息学应用领域多以子问题出现,因此本文采用的两组实验数据分别来自两个不同的实验项目——预测 RNAse P 的二级结构和构建不同地域 dengo 病毒株的进化树.这两个实验项目都需要对多个序列进行比对,本文用 4 种不同的算法比对了实验序列,并且将根据比对结果分析可能的生物规律.本次实验旨在对比 4 种算法的期望时间复杂度和比对效率,这两项可以由实验结果中的"程序平均运行时间"和"SP 值"(sum-of-pairs)[8]来反映.

4.1 预测 RNA 二级结构中的实验结果

在研究不同种微生物体的 RNAse P RNA 的二级结构时,需要对它们的一级序列进行比较.本文分别用新算法、原始的星比对算法、树比对算法和 clustalx 软件比对了 16 种不同微生物体的 P RNA 的核苷酸序列(长度在 300p 到 400p 之间).这些微生物体包括根瘤土壤杆菌(Agrobacterium tumefaciens)、深红螺菌(Rhodospirillum rubrum)、普氏立克次体(Rickettsia prowazekii)以及池塘浮渣(Pond Scum)等.数据来自于北卡罗莱纳州立大学分子生物学实验室[9].具体实验结果如表 1 所示.在本次实验中,所有算法所对应的程序均运行在主频为 Pentium4 3.00GHz,内存为 1.00G 的 PC 机上.

表 1　16 条 RNAse P 序列的比对结果情况对比

	改进后的星比对算法	原始的星比对算法	clustalx(软件)	树比对算法
程序平均运行时间	172 毫秒	875 毫秒	3.3 秒左右	无法运行
SP 值	17015	16790	17616	无

4.2 构建进化树的实验结果

在研究不同地域的 dengo 病毒株的进化关系时,Yong-Hyuk Kim 等人选择了 17 种不同地区(巴西、印尼、墨西哥、纽卡斯尔、普利茅茨、斯里兰卡、塔希提岛和泰国等地)的 dengo 病毒株序列进行了比对[10].这 17 个序列的长度均为 1485p,彼此之间十分相似.本文也利用上述 4 种方法对该 17 个序列作了比对实验,运行环境同上,结果如表 2 所示.

表 2　17 条 dengo 病毒株序列的比对结果情况对比

	改进后的星比对算法	原始的星比对算法	clustalx(软件)	树比对算法
程序平均运行时间	390 毫秒	13250 毫秒	52 秒左右	无法运行
SP 值	8488	8488	8472	无

4.3 实验结果分析

在第一组实验中,改进后的星比对算法的运行时间是原始星比对算法的 19.7%,而 SP 值仅仅增加了 1.3%.在第二组实验中,改进后算法的运行时间是原算法的 2.9%,而 SP 值没有变化.由于星比对算法得到的结果本来就是近似最优解,因此改进后算法得到结果的 SP 值略高于原始星比对算法不会影响生物实验的结论.但改进后的算法在执行时间上远远优于其他方法.

本算法的优越性在实验二中体现得尤为明显.这是因为实验二中的实验序列相似程度非常高.不同地区人所携带的 dengo 病毒株,只是在个别位点发生置换(可以从比对结果中观察到),而且发生变异的位点占总位点数的 1%以下.对于相似度高的序列,本算法在建立好关键字树后,可以在其它序列中使用 Aho-Corasick 算法依次搜索到多个子串,因此在逐个与中心序列比对的过程中,避免了重复比对这些相同的子串.由于与中心序列比对时使用的是动态规划方法,运行时间是与比对长度呈平方关系.如果序列之间的相似度越高,则 Aho-Corasick 算法搜索到的子串个数越接近 k,那么在依次与中心序列比对阶段需要比对的长度就越小,因此与原始星比对算法相比,本算法就越节省时间.

在生物实验中比对多个序列也会有新的发现,例如:通过对实验二中的 dengo 病毒株比对时发现经过变异的各种 dengo 病毒株的长度仍然相同,在观察插入空格的位点和不匹配的位点之后,可以很容易看出 dengo 病毒株在进化时所发生的变异都是相邻的一个或几个位点前后置换.由此可以看出,一种快速高效的多序列比对算法在生物信息研究中显得十分重要.

5 结论和展望

利用 Aho-Corasick 算法可以在线性时间内从一个长字符串中搜索到一个短字符串集合元素所有出现的思想,本文改进了原始的星比对算法.该算法比对结果的 SP 值虽然不是最优解,但是实验结果表明它近似等于原始星比对算法得到的 SP 值,且该算法的期望时间复杂性优于原始星比对算法.两组实验结果很好地说明了本算法的执行效率.

本文算法的适用范围是比对相似程度高的 DNA 序列,如分析基因进化关系、比较单体型序列等.由于算法依靠完全匹配节省时间,因此无法将其扩展至基于罚分矩阵的蛋白质序列比对.

本文的两例实验是在研究生物信息学中两个重要分支——分子结构预测和构建进化树时,对多序列比对的应用.多序列比对的效果直接影响到它们的结论,由此可见基于快速高效的多序列比对算法的程序软件是生物信息学研究的重要工具.根据本文提出的高效的多序列比对算法,去开发相应的结构预测算法、构建进化树算法等是未来的后续工作.另外,改进后的星比对算法还未能解决空隙罚分、节省空间开销等序列比对中的重要问题,这些都有待作更深入研究.

参考文献:

[1] L Wang, T Jiang. On the complexity of multiple sequence alignment[J]. Journal of Computational Biology, 1994, 1(4): 337 - 34.

[2] Robert C Edgar, Serafim Batzoglou. Muitiple sequence alignment[J]. Current opinion in structural biology. 2006, 16(3): 368 - 373.

[3] Notredame C, Higgins D. G. SAGA: sequence alignment by genetic algorithm[J]. Nucleic Acids Research. 1996, 24(8): 1515 - 1524.

[4] Serafim Batzoglou. The many faces of sequence alignment[J]. Briefings in Bioinformatics. 2005, 6(1): 6 - 22.

[5] Aho A V, Corasick M J. Efficient string matching: an aid to bibliographic search[J]. Communications of ACM, 1975, 18(6): 333 - 340.

[6] D Gusfield. Algorithms on strings, trees and sequences: computer science and computational biology[M]. Cambridge: Cambridge University Press. 1997.

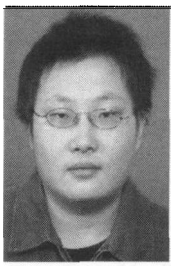
[7] Quan Zou, Maozu Guo, Yang Liu, Taotao Zhang. An Improved Algorithm for Aligning DNA/RNA Sequences Quickly[A]. International Conference on Computational Intelligence and Security[C]. Harbin: IEEE Press, 2007. 825 - 828.

[8] D Gusfield. Efficient methods for multiple sequence alignment with guaranteed error bounds[J]. Bulletin of Mathematical Biology. 1993, 55(1): 141 - 154.

[9] Brown J W. The Ribonuclease P database[J]. Nucleic Acids Research, 1999, 27(1): 314.

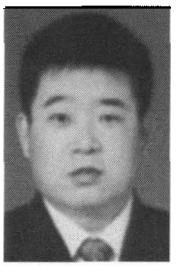
[10] A Rambaut. Estimating the rate of molecular evolution: incorporating noncontemporaneous sequences into maximum likelihood phylogenies[J]. Bioinformatics, 2000, 16(4): 395 - 399.

作者简介:



邹 权 男,1982 年生于黑龙江佳木斯,博士研究生,主要研究方向:RNA 结构预测、多序列比对、microRNA 识别等生物信息学问题与算法.

E-mail: guoer713108@gmail.com



郭茂祖 男,1966 年生于山东夏津,博士后,教授、博士生导师,中国人工智能学会理事兼机器学习专业委员会常委.主要研究方向:生物信息学与计算生物学、机器学习与数据挖掘、新型计算模型.

E-mail: maozuguo@hit.edu.cn

* 本文中提出的算法、软件和源代码的下载地址为:
http://dbgroup.cs.tsinghua.edu.cn/zouquan/improvedcenterstar/