计算机研究与发展
Journal of Computer Research and Development
ISSN 1000-1239/CN 11-1777/TP
45(Suppl.): 175-179, 2008

一种新的DNA序列重复片段的查找算法

郭 顺[1] 管河山[2] 姜青山[1]
[1](厦门大学软件学院 福建厦门 361005)
[2](厦门大学计算机科学系 福建厦门 361005)
(gsgowell@163.com)

A Novel Algorithm for Finding Approximate Tandem Repeats in DNA Sequences

Guo Shun[1], Guan Heshan[2], and Jiang Qingshan[1]
[1](*Software School, Xiamen University, Xiamen, Fujian* 361005)
[2](*Department of Computer Science, Xiamen University, Xiamen, Fujian* 361005)

Abstract In gene analysis, finding approximate repetitions in DNA sequence is an important problem. Proposed here is a new efficient algorithm (MSATR) for detecting approximate tandem repeats in genomic sequences. The theoretical analysis and experimental results show that both space and time complexity of the algorithm is $O(n)$. This algorithm also allows a wide range of definitions of similarity of approximate tandem repeats. The experiment results show that this algorithm is superior to other methods in finding results and it is also time saving.

Key words DNA sequence mining; approximate repetitions; segment-similarity; MSATR

摘 要 寻找DNA序列中的重复片段是DNA序列挖掘中的一项重要的研究内容,它是基因分析的一个重要问题.通常的方法采用特定的索引结构如后缀树、后继数组等,算法效率有待提高.提出一种新的索引结构,并在此基础上提出了MSATR算法.MSATR算法可以适用于各种不同相似度定义的DNA重复片段的查找.分析和实验表明,MSATR算法时间和空间复杂度为$O(n)$.实验结果表明,MSATR算法有较好的查找效率,并且MSATR算法能得到较好的查找结果.

关键词 DNA序列挖掘;相似性重复片段;片段相似度;MSATR

中图法分类号 TP301.6

生物信息学(bioinformatics)是生命科学、计算机科学、信息科学和数学等学科交汇融合所形成的一门交叉学科[1],DNA序列数据是生物信息学的主要研究对象之一.通过分析DNA序列,科学家不仅能够解已有的序列,而且能够更好地研究新的序列及其功能,解读序列在生物体中充当的角色,进而理解生命本质[2].而在DNA序列中经常包含一些连续的重复模式称为重复片段(tandem repeats, TRs).这些重复的片段同它们的生物功能一样不能被完全理解,但是,它们被确定在基因组织和进化上起着重要的作用[3-4].目前,重复片段作为一个重要的遗传标记,已经广泛运用于精密遗传连锁作图、肿瘤生化研究、法医学个体识别、亲子鉴定和群体遗传学分析等领域[5-6].因此,采用数据挖掘技术,查找和分析这种重复片段显得十分必要.

现有的查找算法[7-12]包括完全重复片段查找的算法和相似性重复片段查找的算法.完全重复片段查找算法[7-9]查找的是DNA序列中多个完全相同的模式串联而成的片段,这些算法的复杂度为$O(n\log n)$.但是,由于基因的突变、迁移、反转,重复片段往往不完全相同[11].于是,相似性重复片段(approximate tandem repeat, ATR)被提出来,并

收稿日期:2008-06-16
基金项目:国家自然科学基金项目(10771176);厦门大学校科研基金项目(0630-X01117)

且出现了大量的查找 ATRs[10-12]算法.

本文提出了一种新的算法 MSATR(motif-divide based search approximate tandem repeats),该算法先在第 1 阶段根据模式长度将 DNA 序列划分成多个串联的模式放入数组,再在第 2 阶段将已经建立的数组中的模式进行连接.实验和分析表明,同 SUA_SATR[12]算法相比,本文提出的算法在算法效率和查找结果方面都有进一步提高.

1 相关工作

较早的 DNA 序列重复片段查找算法[10]定义的相似度标准一般都是基于编辑距离(edit distance),查找过程需要子序列间两两对比,算法的效率较低,只限制于找到长度较短的重复序列,而且输入序列的大小受到限制.而后,Kurtz 等人[11]提出了基于后缀树的 REPuter 算法,同以往的算法相比,该算法在查找前建立了后缀树这样的索引结构,算法的效率得到了提高.但是,该算法的查找过程仍然需要将子序列两两对比,难以找到 DNA 序列中出现次数较高的重复序列.2007 年,Wang[12]等人在此过程的基础上提出了新的索引结构——后继数组,以及基于"海明距离"的相似度标准,并且基于此提出了 SUA_SATR[12]算法.同 Kurtz 等人[11]的 REPuter 算法相比,SUA_SATR[12]算法建立的索引结构所需要的空间复杂度更低,建立索引结构所需要的时间更短,相似度的定义更加合理,算法的效率有进一步的提高.另外,算法克服了 REPuter[11]算法难以找到 DNA 序列中出现次数较高的重复序列的缺点.

然而,SUA_SATR[12]算法定义的索引结构模式单元的首字母必须相同,这使得查找到的重复片段模式间的首字母必须相同,限制了查找到的重复片段的数量.再者,算法建立的后继数组中相邻的两个模式单元在 DNA 序列中的位置并不相邻,形成相似段后还需要根据后继信息再进行连接,形成重复片段.另外,在形成相似片段的过程中,对于不相邻的模式也进行了比对,这在一定程度上影响了算法的效率.

所以,如果能够建立一种新的索引结构,结构中相邻的模式在 DNA 序列中同样相邻,而且结构中模式的长度都相同,那么,只需要依次将相邻的模式进行比对、连接就能得到要查找的重复片段,这将进一步提高算法的效率.而且,划分的模式没有首字母必须相同的限制,这将提高算法能查找到重复片段的数量.根据这种思想,本文提出了 MSATR(motif-divide based search approximate tandem repeats)算法.

2 MSATR 算法

MSATR 算法采用与 SUA_SATR[12]算法相同的相似度定义.结合其定义,对本文算法要找的重复片段做出如下定义:

定义 1. MATR(motif-similarity based approximate tandem repeats):如果重复片段 $T=T_1T_2\cdots T_r(r\geqslant 2)$满足片段相似度不低于阈值(任意两个模式间相似度不低于阈值).例如,相似度阈值为 0.75,DNA 序列为 ACCT|AGCT|AACT|ATCT,模式长度为 4,因为任意两个模式的相似度都为$(4-1)/4=0.75$,故该重复片段就是一个满足条件的 MATR.

定义 2. MATR 的表示方法:用一个四元组(s,i,d,p)表示一个 MATR,其中 s 为重复片段,i 表示该片段在序列中的起始位置(序列起始位置为 1),d 表示重复片段模式长度,p 表示周期,即模式重复次数.

2.1 MSATR 算法原理

算法先对 DNA 序列进行划分建立索引结构,而后,将索引结构中的模式根据相似度进行连接得到要找的重复片段.

建立索引结构的过程如下:对长度为 n 的 DNA 序列 seq,根据模式长度 k 进行 k 次划分,每次划分从 seq 的第 $i(0\leqslant i\leqslant k-1)$个位置开始,将 seq 分成 $\lfloor (n-i)/k \rfloor$ 个连续的模式,并依次放入一个数组中.划分完毕后总共得到 k 个数组,很显然,这 k 个数组的总元素个数小于 n.

得到这 k 个数组后我们可以发现,这 k 个数组中相邻的模式在 seq 中同样相邻.所以,只要依次将这 k 个数组扫描一遍,将模式根据相似度进行连接,就能找到模式长度为 k 的重复片段.

2.2 MSATR 算法过程

算法主要分为分段和连接两个阶段,分段阶段就是第 2.1 节所说的建立索引结构的过程.

例如,DNA 序列片段 seq 为 AGTTCTAACAGGAAGACGT.按照模式长度 4 进行分段,总共要进行 4 次划分,建立 4 个数组.第 i 次划分从 seq 的第 i 个位置开始,每 4 个字符作为一个长度为 4 的

模式,依次放入第 i 个数组中. 对于划分到 *seq* 末尾长度不到 4 的片段,舍去不加入数组中. 划分完后我们得到如表 1 中的 4 个数组:

表 1 序列 *seq* 按照模式长度 4 划分得到的数组

数组	模式			
Array1	AGTT	CTAA	CAGG	AAGA
Array2	GTTC	TAAC	AGGA	AGAC
Array3	TTCT	AACA	GGAA	GACG
Array4	TCTA	ACAG	GAAG	ACGT

第 2 个阶段是连接,同样以上述例子来说明这个过程. 得到表 1 中的 4 个数组后,我们要做的只是依次将 4 个数组扫描一遍,根据相似度函数 S 来判断模式是否连接,并将符合条件的片段加入 MATR 集合中. 以下是具体步骤.

开始扫描时,先将数组中第 1 个元素放入缓冲区中,初始化缓冲区以及其位置、最小周期、模式长度等信息. 如开始扫描 Array1 时,将第 1 个元素"AGTT"加入缓冲区,最小周期 p 设置为 1,模式长度为 4,起始位置为 1.

然后从数组的第 2 个元素开始,将数组中的元素同缓冲区片段根据相似度函数进行比较判断.

如果数组中元素同缓冲区中片段相似,则将缓冲区中片段同该元素连接,形成新的片段,并将片段的周期加 1,再跳到下一个元素同新的缓冲区片段进行比较. 否则,判断缓冲区中片段的周期.

如果缓冲区中片段周期小于给定参数(最小周期),则将缓冲区替换为新的数组元素,并替换片段的位置信息等. 否则,将缓冲区中片段以及相关信息加入 MATR 集合中,再替换缓冲区以及片段相关信息.

当一个数组扫描完时,需要再次判断缓冲区中片段周期,来决定是否将数组中最后片段是否加入 MATR 集合中.

对表 1 中的每个数组按上述步骤进行扫描. 当所有数组扫描完毕时,就得到了 *seq* 中模式长度为 4 的所有重复片段,并且存放在 MATR 集合中.

2.3 MSATR 算法的主要参数

模式长度范围(a,b):要查找的 MATR 最小模式长度为 a,最大模式长度为 b,对于每一个模式长度,都按照第 2.2 节的算法过程查找该长度模式的重复片段.

最小周期 p:找到的 MATR 模式的周期都大于等于 p.

相似度函数 S:根据相似度定义来判断已经存在的相似片段同新的模式是否相似. 该函数可以根据需要,对不同的相似度定义进行相应的设置. S 函数主要包括 3 个参数,分别是已经存在的相似片段 *segment*,同该片段相邻的模式 *motif*,以及相似度阈值 r. 伪代码如算法 1 所示.

算法 1. MSATR 算法伪代码.

输入:DNA 序列 *seq*、模式长度范围(a,b)、最小周期 p、相似度函数 S、相似度阈值 r.

输出:模式长度范围(a,b)里的重复片段集合.

```
for i := a to b do
  for j := 1 to i do
    Divide (seq,j,arry[j]);
  end for
  /* 根据模式长度 i 把 seq 进行 i 次划分,并把每次划分的模式放入第 j 个数组 arry[j] 中 */
  for k := 1 to i do /* 对划分的 i 个数组进行扫描 */
    Buffer=arry[k][0];
    Startposition=k;
    Period=1; /* 初始化缓冲区从第 2 个位置起,对数组中的元素进行扫描 */
    for d := 1 to arry[k].length do
      if (S(buffer, arry[k][d],r)) then
        Buffer+=arry[k][d];
        Period++;
      end if
      else
        if (Period>=p) then
  MATR.add(Buffer,Startposition,I,Period);
        end if
        Buffer=arry[k][d];
        Startposition+=d×i;
        Period=1; /* 替换缓冲区 */
      end else
    end for
  end for
end for
```

2.4 MSATR 算法复杂度分析

对于模式长度 k,总共要对序列进行 k 次划分. 假设序列的长度为 n,则每 i 次划分只是将序列分成 $\lfloor (n-i)/k \rfloor$ 段,并依次放入数组中. 总共运行所需

要的时间为 $k\times(\lfloor(n-i)/k\rfloor)<n$. 而连接步骤做的只是把划分好的 k 个数组中的不大于 n 的所有模式依次扫描一遍，同缓冲区的片段做比较. 所以，连接步骤的时间复杂度依然是 $O(n)$. 假设模式长度范围为 m，则整个算法运行所需要的时间为 $m\times(n+n)=2mn$. 所以，算法的时间复杂度为 $O(n)$.

算法所需要的空间包括存放总数不大于 n 的模式的索引结构以及用来存放查找到的 MATR 集合的空间. 所以，整个算法所需要的空间复杂度为 $O(n)$.

3 实验与结果分析

实验主要从两方面进行评估，一是从算法的运行时间进行评估，再者是从算法找到的重复片段的数量进行评估.

3.1 实验数据

实验数据采用真实的来自 GenBank (http://www.ncbi.nlm.nih.gov/Genbank) 序列数据库的基因序列(人类第 22 号染色体[13]序列(34,107,095 nt)片段).

3.2 实验环境

本实验环境为 Pentium 4 CPU 3.00 GHz, 1.00 GB内存，WinXp 系统，JCreator Pro3.50，算法用 Java 语言实现.

3.3 实验参数设置

相似度函数 S 按照 SUA_SATR[12]算法的相似度定义进行设置. 最小周期数 p 设置为 2. 模式长度范围 (a,b)：在实验中，根据 SUA_SATR[12]算法运行得到的结果，模式长度范围主要集中在 1～80 之间，所以以此范围作为本文算法的模式长度范围参数.

3.4 时间效率比较

如图 1 所示，随着序列长度的不断增加，运行时间也相应增加. 但是，相比之下，MSATR 查找所需要的时间明显要低于 SUA_SATR[12]算法所需要的时间. 同时，在相同相似度的情况下，随着序列长度的增加，MSATR 查找算法的时间明显增长更为缓慢.

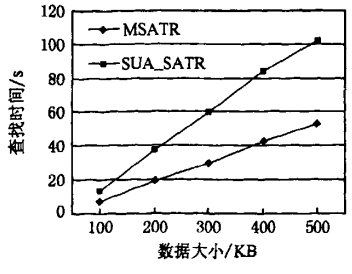

图 1 MSATR 与 SUA_SATR[12]算法查找时间的比较 (相似度阈值为 0.75)

3.5 查找结果比较

由图 2 可以看出，对于同样大小的数据集，在相同模式范围、相同相似度定义、相同相似度阈值0.75的情况下，MSATR 算法找到比 SUA_SATR 更多的 MATR. 这主要是因为 SUA_SATR 算法限制了相似片段中每个模式的首字母必须相同. 而 MSATR 算法并不存在这种限制，能够找出更多符合相似度条件的重复片段.

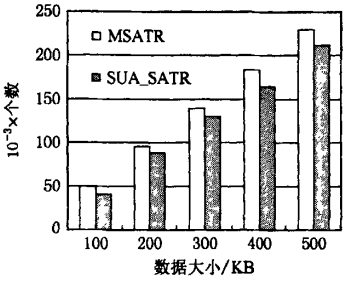

图 2 MSATR 与 SUA_SATR 算法找到的重复片段数量对比(相似度阈值为 0.75)

4 结 论

本文提出了一种新的 DNA 序列的相似性重复片段的查找算法 MSATR，该算法具有线性的时间和空间复杂度. 通过实验分析结果表明，在相同相似度定义、相同模式长度范围的情况下，该算法的查找时间和查找结果优于其他同类的算法. 另外，MSATR 算法可以根据参数来调整查找的结果，使得该算法的查找更加具有针对性和目的性. 新的更加具有生物学意义的相似度度量是今后工作的一项重要挑战.

参 考 文 献

[1] Luscombe N M, Greenbaum D, Gerste M. In what is bioinformatics? A proposed definition and overview of the field. Methods Information in Medicine, 2001, 40(4): 346-358

[2] 朱杨勇，熊赟. DNA 序列数据挖掘技术. 软件学报，2007, 18(11): 2766-2781

[3] Li Y, Korol A, Fahima T, *et al*. Microsatellites: genomic distribution, putative functions and mutational mechanisms. Molecular Ecology, 2002, 11(12): 2453-2465

[4] Kashi Y, King D, Soller M. Simple sequence repeats as a source of quantitative genetic variation. Trends in Genetics, 1997, 13(2): 74-78

[5] Beleza S, Alves C, Gonzalez-Neira A. Extending STR markers in Y chromosome haplotypes. International Journal of Legal Medicine, 2003, 117(1): 27-33

[6] Gilmore S, Peakall R, Robertson J. Short tandem repeat (STR) DNA markers are hypervariable and informative in Cannabis sativa: implications for forensic investigations. Forensic Science International, 2003, 131(1): 65-74

[7] Apostolico A, Prefarata F. Optimal off-line detection of repetitions in a string. Theoretical Computer Science, 1983, 22(3): 297-315

[8] Crochemore M. An optimal algorithm for computing the repetitions in a word. Information Processing Letters, 1981, 12(5): 244-250

[9] Main M, Lorentz R. An O (nlogn) algorithm for finding all repetitions in a string. Journal of Algorithms, 1984, 5(3): 422-432

[10] Benson G. Tandem repeats finder: A program to analyze DNA sequences. Nucleic Acid Research, 1999, 27(2): 573-580

[11] Kurtz S, Choudhuri J V, Ohlebusch E, *et al*. REPuter: The manifold applications of repeat analysis on a genomic scale. Nucleic Acids Research, 2001, 29: 4633-4642

[12] 王锡,赵毅,陈白尘,等. DNA序列中基于后继数组索引的SATR查找算法. 东北大学学报(自然科学版), 2007, 28(2): 209-212

[13] NCBI. [2008-04-16]. http://www.ncbi.nlm.nih.gov/mapview/maps.cgi?taxid=9606&chr=22

郭 顺 男,1982年生,硕士研究生,主要研究方向为数据挖掘.

管河山 男,1981年生,博士研究生,主要研究方向为时间序列、数学模型、数据挖掘.

姜青山 男,1962年生,教授,博士生导师,主要研究方向为数据挖掘、图像处理、数据库系统、模糊集理论与应用.

一种新的DNA序列重复片段的查找算法

作者： 郭顺， 管河山， 姜青山
作者单位： 郭顺,姜青山(厦门大学软件学院 福建厦门 361005)， 管河山(厦门大学计算机科学系 福建厦门 361005)