doi:10.3969/j.issn.1672-5565.2013.01.02

DNA 序列频繁近似模式挖掘

姜 华[1*],孟志青[2],周克江[1]
(1. 湖南第一师范学院信息科学与工程系,长沙 410205;2. 浙江工业大学经贸管理学院,杭州 310023)

摘 要:本文在引入近似度等概念的基础上,构造了频繁近似模式,并证明了相关性质,同时提出了相应的频繁近似模式的挖掘算法(SFAP)算法。实验结果表明该算法能有效挖掘 DNA 序列中的频繁近似模式,DNA 序列中频繁近似模式的挖掘为生物学的相关实验提供基础。

关键词:近似序列模式,数据挖掘,DNA 序列
中图分类号:Q518.2 **文献标识码**:A **文章编号**:1672-5565(2013)-01-011-05

Discovery of frequent approximate patterns in Genomic DNA sequences

JIANG Hua[1*], MENG Zhi-qing[2], ZHOU Ke-jiang[1]
(1. *Department of Information Science and Engineering, Hunan First Normal College, Changsha* 410205, *China*;
2. *College of Business and Administration, Zhejiang University of Technology, Hangzhou* 310023, *China*)

Abstract: This paper introduces the concepts and properties of the frequent approximate patterns on the basis of approximation degree format. It proposes an algorithm to search frequent approximate pattern (SFAP) in genomic DNA sequences. Results obtained from experiments demonstrate that the proposed algorithm is efficient. The discovery of the frequent approximate patterns in DNA provides a basis for the further biological research.

Key words: Approximate Sequential Patterns, Data Mining, DNA sequences

1 引言

数据挖掘在生物信息学领域的应用潜力日益受到人们的重视[1],其中 DNA 序列模式挖掘是生物信息学最重要的问题之一。由于 DNA 序列不同于交易序列等序列数据,直接将序列模式挖掘算法应用到生物序列数据时,实验证实在可伸缩性等方面存在问题,如某些数据结构或剪枝策略不能有效适用生物序列数据等。并且,挖掘结果的有效性也不能满足生物学研究需求,比如模式间的模糊匹配含义等。因此,研究者开始了关于专门的 DNA 序列模式挖掘算法的研究。Chaudhuri 等人采用基于统计 DNA 词频的方法聚类 DNA 序列数据[1-2];Wang 等人定义了新的序列间以及序列与簇间的相似度度量,克服了已有的相似度度量局限,并提出了 CLUSEQ 聚类算法[3]; Ester 等人提出的 TOMMSA 算法采用自顶向下搜索策略,能够有效挖掘长模式,在一定程度上提高了挖掘 DNA 长序列模式的效率[4];国内也有人开展了相关研究,文献[5]在新的多支持度框架定义下,提出一种挖掘 DNA 重复序列的有效算法 DnaReSM,其结果为生物学实验提供一定的理论基础。

然而,这些挖掘算法基本上都是基于精确匹配的 DNA 序列模式定义,这种定义不能满足生物学研究中 DNA 序列数据出现内部干扰或基因突变时的挖掘需求。因此有些学者开始研究 DNA 序列中的相似性串联重复片段[6-10],它研究的是 DNA 序列中一些连续的重复模式,重复部分不必完全相同,但它挖掘出来的重复片段必须是连续的,不能有缺失。然而,因为基因突变或迁移等原因,重复片段可能在

收稿日期:2012-08-11;**修回日期**:2012-08-28.
基金项目:湖南省自科科学基金资助项目(09JJ6093);湖南第一师范学院校级课题基金资助项目(XYS10N06)。
*作者简介:姜华,女,湖南人,讲师、硕士,主研方向:数据挖掘,生物信息学等,E-mail:jianghua_clo1@126.com.

位置上发生平移,从而使得重复片段不是连续,而是在某些位置上有缺失。因此,本文研究了 DNA 序列中的频繁近似模式发现(SFAP)算法,主要贡献有两个方面:(1)与相似性串联重复片段不同,本文挖掘出来的频繁近似模式考虑重复片段可能在位置上发生平移,允许重复片段在某些位置上有缺失,挖掘出来的模式不包含重叠,也没有首字符必须相同的限制。(2)与 motif 序列不同,本文不光考虑模式间的海明距离不超过某个给定阈值 k,还考虑模式长度的影响,为此提出近似度的概念。Motif 序列中支持度的定义是模式在 DNA 序列中的重复出现次数,而重复次数在整个序列中所占的比重(百分比)并没有考虑,即假设 DNA 序列长度为 1 000,模式长度为 20 的模式和长度为 2 的模式重复次数都为 50,显然模式长度为 20 的模式比长度为 2 的模式更有意义。因此,考虑重复次数对整个序列的影响,我们提出频繁近似模式支持度的概念。

2 问题定义

首先规定 DNA 序列总是由 A ,C , G, T 四种字符以及作为结束符的 $ 组成的。在此基础上给出模式的相关定义。模式 P 定义如下:$P = <pM1, p_2, \cdots, p_n>$ ($p_i \in \Sigma$)。模式 P 实际上是一个长度为 n 的字符串序列。

定义 1(近似度):在字符表 $\Sigma = \{A, C, G, T\}$ 之上,给定模式 $P = <p_1, p_2, \cdots, p_n>$ 和模式 $P' = <p_1', p_2', \cdots, p_n'>$,我们定义近似度如下:

$$\text{appro\_degree}(P, P') = \frac{|I|}{length(P)}, \text{其中 } I = \{i \mid p_i' = p_i, i = 1, \cdots, n\}$$

其中集合 I 表示两个模式间相同的字符集合,该近似度不光考虑模式间的海明距离,还考虑模式长度的影响。即假设模式间的海明距离相同,模式长度分别为 10 和 100,显然,模式长度为 100 表明两模式更相似。当近似度为 1 时,表示两模式完全匹配。

定义 2 给定模式 $P = <p1, p2, \cdots, p_n>$ 和模式 $P' = <p_1', p_2', \cdots, p_n'>$,符号 *appro_match*($P, P'$) 表示模式 P 和模式 P' 是否满足用户指定的最小近似度,是则为 1,否则为 0。

定义 3(频繁近似模式 *FAP*) 给定序列 S,模式 $P = <p_1, p_2, \cdots, p_n>$ 在序列 S 中是频繁近似模式当且仅当 $\sum_{i=1}^{} \text{appro\_match}(P, P_i) \geq m$

其中

(1) P_i 是 S 的子串;

(2) $|P_i| = |P|$;

(3) m 是用户指定的最小频繁阈值;

(4) 对 $P_i = S[a \cdots b]$, $P_j = S[c \cdots d]$,其中 $i \neq j$,若 *appro_match*(P, Pi) = 1 且 *appro_match*(P, P_j) = 1,则必须 $b < c$ 或 $d < a$(保证不重叠)。

定义 4(频繁近似模式的支持度)给定序列 S,模式 $P = <p_1, p_2, \cdots, p_n>$ 在序列 S 中是频繁近似模式,我们称

$$sup(P) = \frac{\sum_{i=1}^{n} \text{appro\_match}(P, P_i)}{[\text{length}(S)/\text{length}(P)]}$$

是频繁近似模式 P 在序列 S 上的支持度。

支持度越大,表明模式在序列中出现频率越高,当支持度为 1,表明该模式在序列中是一个频繁精确模式。

性质 1 $0 \leqslant appro\_degree(P, P') \leqslant 1$

证明:显然,appro_degree(P, P') ≥0。根据定义 1 可知,模式 $P = <p_1, p_2, \cdots, p_n>$ 和模式 $P' = <p_1', p_2', \cdots, p_n'>$,$I = \{i \mid p_i' = p_i, i = 1, \cdots, n\}$,显然 $|I| \leqslant$ length(P),故 appro_degree(P, P') ≤1。

性质 2 $0 \leqslant \sup(P) \leqslant 1$

证明:显然,sup(P) ≥0。根据频繁近似模式的定义中第(4)条知,对 $P_i = S[a \cdots b]$, $P_j = S[c \cdots d]$,其中 i≠j,若 appro_match(P, P_i) = 1 且 appro_match(P, P_j) = 1,则必须 b < c 或 d < a(保证不重叠),即序列 S 中最多存在个不重叠模式,故 sup(P) ≤1。

性质 3 P 是序列 S 中的频繁近似模式,若序列 S 中总共存在 m 个模式 $P_1, P_2, \cdots, P_m$,其中 $P_1 = S[a_1, a_1']$, $P_2 = S[a_2, a_2']$, $\cdots$, $P_m = S[a_m, a_m']$(其中 $a_1 < a_2 < \cdots < a_2$),均使得 appro_match(P, P_i) = 1,则要使得 S 序列中不重叠模式个数最多,则第一个不重叠模式必然是 P1(即 $S[a_1, a_1']$)。

证明:反证法

根据频繁近似模式的定义(定义 3)可知,$|P_i| = |P|$,设 $|P| = l$,则 $P_1 = S[a_1, a_1 + l - 1]$, $P_2 = S[a_2, a_2 + l - 1]$, $\cdots$, $P_m = S[a_m, a_m + l - 1]$。

若第一个不重叠模式不是 $S[a_1, a_1 + l - 1]$,而是第 i 个模式 $S[a_i, a_i + l - 1]$,其中 $a_i > a_1$。

若 $a_{i+1} - a_i \geq l$,则第二个不重叠模式为 $S[a_i + 1, a_{i+1} + l - 1]$。

若 $a_{i+1} - ai < l$,则比较 a_{i+2} 和 a_i,若 $a_i + 2 - a_i < l$,则比较 a_{i+3} 和 a_i,, $\cdots$,直至 $a_{k+1} - a_i \geq l$,则第二个重叠模式为 $[a_k, a_k + l - 1]$。以此类推,得到第三个不重叠模式,第四个不重叠模式, $\cdots$,第 n 个不重叠模式。

显然,第二个不重叠模式不会与第一个不重叠

模式重叠,因为 $a_i > a_1$,故也不会与模式$[a_1, a_1 + l - 1]$重叠。

若 $a_i - a_1 \geq l$,则$[a_1, a_1 + l - 1]$也是 S 序列中的不重叠模式,这样不重叠模式个数加 1。矛盾

若 $a_i - a_1 <$,则第一个不重叠模式可替换为$[a_1, a_1 + l - 1]$。矛盾

故性质 3 成立。

根据性质 3,可构造不重叠模式个数最多的方法。

根据性质 3,可知第一个不重叠模式为 $S[a_1, a_1 + l - 1]$。

若 $a_2 - a_1 \geq$,则第二个不重叠模式为 $S[a_2, a_2 + l - 1]$,否则比较 a_3 和 a_1,…,直到 $a_k - a_1 \geq l$,则第二个不重叠模式为 $S[a_k, a_k + l - 1]$。以此类推,得到第三个不重叠模式,第四个不重叠模式,…,第 n 个不重叠模式。

该性质可用于对候选模式裁剪,快速找出满足条件的频繁近似模式。

3　频繁近似模式挖掘算法

这一节我们将用频繁近似挖掘(Search Frequent Approximate Pattern,简称 SFAP)算法来挖掘 DNA 序列中的频繁近似模式。SFAP 算法主要的思想是先产生候选模式集,再对候选模式集进行裁剪,验证候选模式的有效性。

3.1　产生候选模式集

扫描长度为 n 的 DNA 序列 *seq*,从第 k 个位置$(0 \leq k < n - i)$开始产生所有长度为 i 的候选模式。例如 DNA 序列片段 *seq* 为 ATTCCACTGGGCGCC,那么长度为 5 的所有候选模式为:ATTCC,TTCCA,TCCAC, CCACT, CACTG, ACTGG, CTGGG, TGGGC, GGGCG,GGCGC,GCGCC。显然,得到的候选模式总数为 $n - i + 1$。

3.2　裁剪和验证

对候选模式进行裁剪和验证是交替进行的。验证候选模式的有效性就是找出那些满足用户预先定义的支持度要求的候选模式。为了求得候选模式的支持度,根据上文的定义 3,关键是要算出 $\sum appro\_match(P, P_i)$,根据频繁近似模式的定义知,即要保证挖掘的模式之间不重叠且重复次数最多。可利用性质 1 对候选模式进行裁剪。将长度为 i 的候选模式集放到集合 C 中,对候选模式 P 的每个起始位置 index,通过哈希函数 H(index) = index 映射到集合 C 上,令 C[H(index)] = C[index] = P,这样模式 P 的起始位置和集合 C 中的下标构成一一对应的关系。模式 P 在和集合 C 中的元素进行比较时,可很快的根据下标找到对应的元素,而不需要与集合 C 中的元素逐个进行比较。

具体算法如下:

(1)从集合 C 中依次取出一个候选模式 P,若模式 P 已经在近似频繁模式集合 S 中,则取下一个候选模式,否则候选模式 P 与集合 C 的其他元素 C[index]进行比较;

(2)若 appro_degree(P, C[index])大于最小近似度,则候选模式 P 的重复次数加 1,index = index + i,候选模式 P 继续与集合 C 的元素 C[index]进行比较;否则 index = index + 1, 候选模式 P 继续与集合 C 的元素 C[index]进行比较;

(3)重复(2),直到 index 大于等于集合 C 中的最大下标。统计候选模式 P 的重复次数和支持度,若候选模 P 的重复次数和支持度均满足指定的最小频繁阈值 m,最小支持度 minsupport,则候选模式 P 是频繁近似模式,加入近似频繁模式集合 S 中。

(4)重复(1),直到集合 C 中每个元素都取出完毕。

4　实验

为验证算法的有效性,我们采用 Homo sapiens human gamma - glutamyl hydrolase, Ampicillin (bla) resistance 和 Homo sapiens growth hormone receptor 三种 DNA 序列作为测试数据。实验任务比较 SFAP 算法和 MSATR[11] 算法找到的模式数量,以及最小支持度和最小近似度的选择对挖掘结果(近似频繁模式个数)的影响。

4.1　SFAP 算法与 MSATR 算法找到的模式数量对比

如图 1 所示,对于同样的测试数据,在同样的频繁阈值(最小频繁阈值为 2),相同的相似性条件(最小近似度为 0.75)下,SFAP 的查找结果要远远多于 MSATR 的查找结果。这是因为 MSATR 算法限制了相似片段中的模式不能有缺失,而 SFAP 不存在这种限制,因此能找出更多的模式。表 1 表明 MSATR 算法难以找到重复次数高的模式,而 SFAP 没有这个限制。

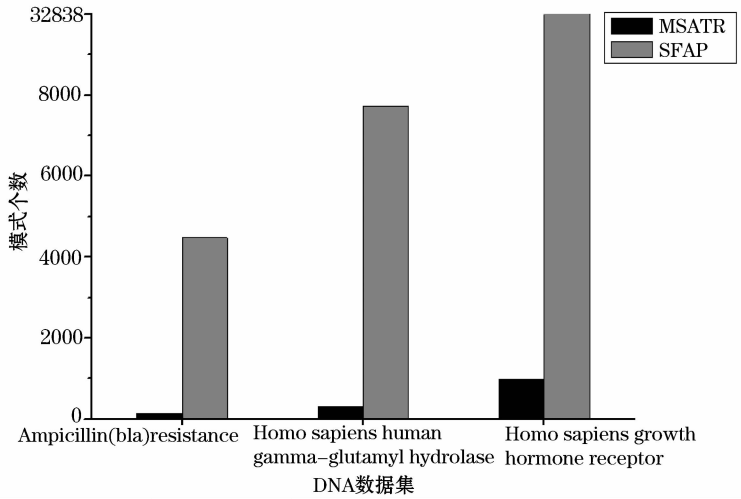

图 1 SFAP 算法与 MSATR 算法找到的模式数量对比(最小近似度为 0.75)

Fig. 1 Number of patterns found by SFAP and MSATR(minimum approximate degree with 0.75)

表 1 SFAP 算法与 MSATR 算法找到的模式数量随最小频繁阈值的变化(测试数据为 Ampicillin (bla) resistance DNA 序列)

Table 1 Number of patterns found by SFAP and MSATR along with the change of minimum frequent threshold (test data: Ampicillin (bla) resistance sequence)

最小频繁阈值	MSATR 找到的模式个数	SFAP 找到的模式个数
2	134	4 473
3	4	3 051
4	0	2 336
10	0	845
50	0	292

4.2 最小近似度和最小支持度的选择对挖掘结果(近似频繁模式个数)的影响

图 2 表明频繁近似模式的数量随着最小近似度要求的降低而迅速增加。当最小近似度为 0.60 时,Homo sapiens human gamma - glutamyl hydrolase DNA 序列中找到的频繁近似模式个数为 4,当最小近似度为 0.45 时,找到的频繁近似模式个数为 5408。Ampicillin (bla) resistance DNA 序列也有类似规律。图 3 表明频繁近似模式的数量随着最小支持度要求的降低而增加。当最小支持度为 70% 时,Homo sapiens human gamma - glutamyl hydrolase DNA 序列中找到的频繁近似模式个数为 1,当最小支持度为 40% 时,找到的频繁近似模式个数为 5312。同样,Ampicillin (bla) resistance DNA 序列也有类似规律。

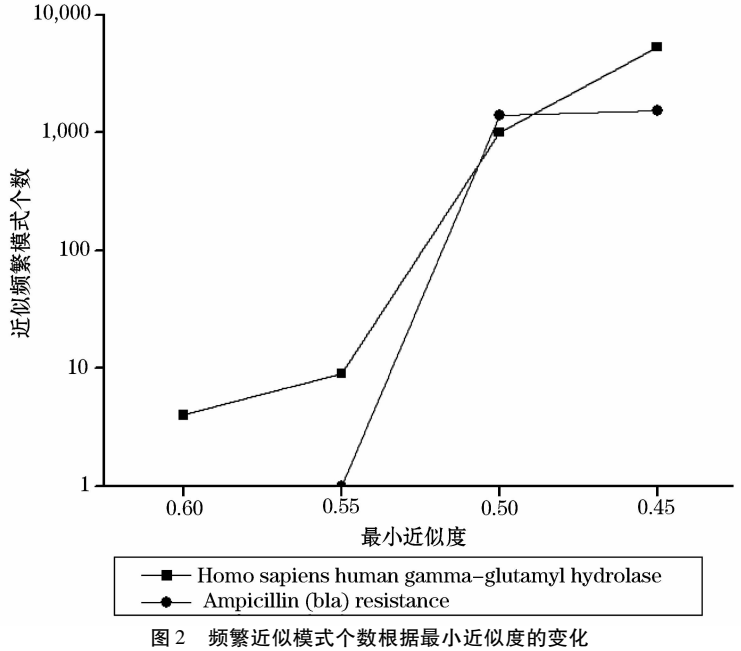

图 2 频繁近似模式个数根据最小近似度的变化

Fig. 2 Number of frequent approximate patterns along with the change of minimum approximate degree

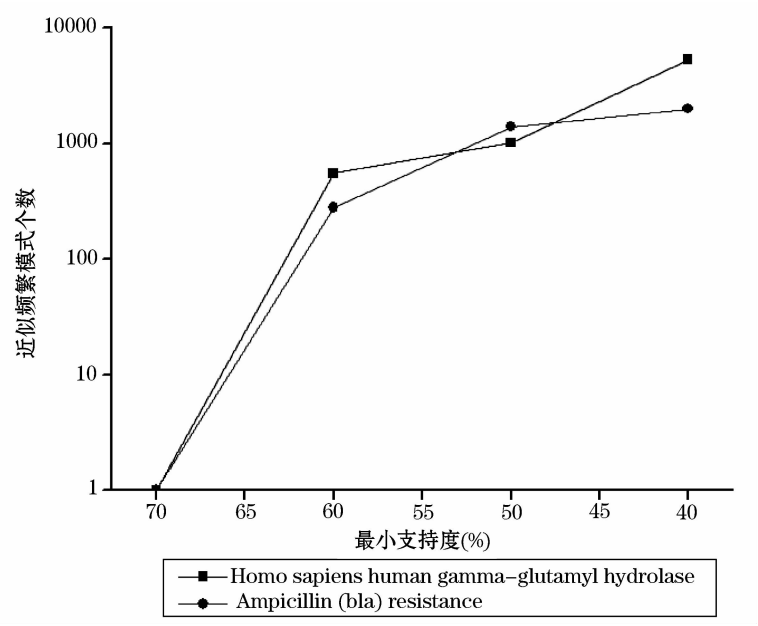

图 3　频繁近似模式个数根据最小支持度的变化

Fig. 3　Number of frequent approximate patterns along with the change of minimum support degree

5　结语

本文提出了一种新的频繁近似模式的定义和相应的挖掘算法 SFAP。通过对一些实际的 DNA 数据进行挖掘,发现了很多频繁近似模式。通过实验表明,在相同近似度定义的情况下,该算法的查找结果远远优于其他同类算法。挖掘得到的频繁近似模式是否具有一定的功能 ,有待于生物家进一步研究鉴定,如何提高算法效率是后续进一步研究的问题。

参考文献(References)

[1] Chaudhuri P, Das S. Statistical analysis of large DNA sequences using distribution of DNA words[J]. Current Science, 2001, 80(9): 1161 – 1166.

[2] Chaudhuri P, Das S. SWORDS: A statistical tool for analyzing large DNA sequences[J]. Journal of Biosciences, 2002, 27(1): 1 – 6.

[3] Yang J, Wang W. CLUSEQ: Efficient and effective sequence clustering[A]. In: Dayal U, Ramamritham K, Vijayaraman TM, eds. Proc. of the 19th Intl Conf. on Data Engineering[C]. Bangalore: IEEE Computer Society, 2003. 101 – 112.

[4] Ester M, Zhang X. A top – down method for mining most specific frequent patterns in biological sequence data[A]. In: Berry MW, Dayal U, Kamath C, Skillicorn DB, eds. Proc. of the 4th SIAM Intl Conf. on Data Mining[C]. 2004. 90 – 101.

[5] 熊赟,陈越,朱扬勇,DnaReSM:一个基于多支持度的 DNA 重复序列挖掘算法[J]. 计算机科学,2007,34(2):211 – 212.

[6] G. M. Landau, J. P. Schmidt, D. Sokol. An Algorithm for Approximate Tandem Repeats[J]. Journal of Computer Biology, 2001, 8(1): 1 – 18.

[7] Kurtz S, Choudhuri JV, Ohlebusch E, Schleiermacher C, Stoye J, Giegerich R. REPuter: the manifold applications of repeat analysis on a genomic scale[J]. Nucleic Acids Research, 2001 Nov, 29(22): 4633 – 42.

[8] Y. Wexler, Z. Yakhini, Y. Kashi, D. Geiger. Finding Approximate. Tandem Repeats in Genomic Sequences[J]. Journal of Computation Biology, 2005, 12(7): 928 – 942.

[9] Yajun Jiang, Zhenlun Yang, Zengrong Zhan. A New Method for Finding Approximate Repetitions in DNA Sequences[A]. 2010 2nd International Conference on Signal Processing Systems[C], 2010, (2): 803 – 809.

[10] 王镝,赵毅,陈白尘,王国仁. DNA 序列中基于后继数组索引的 SATR 查找算法[J]. 东北大学学报(自然科学版),2007, 28(2): 184 – 188.

[11] Qingshan Jiang, Sheng Li, Shun Guo, Dan Wei. A New Model for Finding Approximate Tandem Repeats in DNA Sequences[J]. Journal Of Software, 2011, 6(3): 386 – 394.