Computer Science and Application 计算机科学与应用, 2015, 5(8), 271-277
Published Online August 2015 in Hans. http://www.hanspub.org/journal/csa
http://dx.doi.org/10.12677/csa.2015.58035

Hans汉斯

An Efficient Algorithm for Mining DNA Sequences Based on the Association Matrix

Guojun Mao, Jingxin Yang

School of Information, Central University of Finance and Economics, Beijing
Email: maximmao@hotmail.com

Received: Sep. 1st, 2015; accepted: Sep. 20th, 2015; published: Sep. 24th, 2015

Abstract

The DNA analysis is the core of bioinformatics research, and as an important technology to support bioinformatics, the data mining has been widely applied to the analysis of DNA sequences. Compared to the transaction sequences in traditional business areas, DNA sequences have the characteristics that are item-less but length-longer, so the classic sequence mining algorithms are not perfectly suitable for the DNA sequence pattern mining. Based on the analysis of DNA sequence mining demands, we propose an efficient data structure, called Association Matrix. Such a structure can compress a long DNA sequence into a matrix form which can be effectively analyzed. Therefore, by making use of the space compactness of this structure, we can deal with DNA sequences with a super-long length in a limited memory. Based on the Association Matrix, we design an efficient mining algorithm to find the key segments from DNA Sequence. Experiments show that the proposed algorithm performs well in DNA sequence mining.

Keywords

DNA Sequence, Data Mining, Association Matrix, Key Sequence Mining

一种基于关联矩阵的高效DNA序列挖掘算法

毛国君，杨静欣

中央财经大学信息学院，北京
Email: maximmao@hotmail.com

收稿日期：2015年9月1日；录用日期：2015年9月20日；发布日期：2015年9月24日

文章引用：毛国君，杨静欣. 一种基于关联矩阵的高效 DNA 序列挖掘算法[J]. 计算机科学与应用, 2015, 5(8): 271-277.
http://dx.doi.org/10.12677/csa.2015.58035

摘　要

DNA分析是生物信息学研究中基础而核心的工作，而数据挖掘作为支撑生物信息学的重要技术，已经被广泛应用到DNA序列的分析中。与传统的商业领域的事务序列相比，DNA序列具有项目符号少但序列长度长的特点，因此经典的序列挖掘算法很难适应DNA序列的模式挖掘需要。本文在分析DNA序列的挖掘需求基础上，提出了一种称为关联矩阵的数据结构。关联矩阵能够将序列数据压缩成可分析的矩阵形式，所以它的空间紧凑性能够使得超长的DNA序列能够在有限的内存中加以处理。基于关联矩阵结构，设计了高效的DNA序列的关键序列挖掘算法。实验说明了本文算法在DNA序列分析中的高效性。

关键词

DNA序列，数据挖掘，关联矩阵，关键序列挖掘

1. 引言

随着分子生物学技术的发展，DNA 序列等生物数据已经被信息化和网络化，为相关的研究提供了大量的实验数据。事实上，为了从生物数据中挖掘出相关知识，信息生物学研究已经和数据挖掘等信息处理技术有机结合，并且形成了一个新的交叉研究领域。DNA 序列是最重要的生物数据之一，通过挖掘 DNA 序列，研究者可以发现序列数据背后隐藏的有价值的规律，解读相应生物体中的生物特征[1]。

一般地说，一个 DNA 序列可以抽象成由固定大小的项目集(字母集)生成的字符串。这样的字符串有 2 个主要特点：(1)被观察的事务集只有四个字母，即腺嘌呤(A)、胸腺嘧啶(T)、鸟嘌呤(G)和胞嘧啶(C)。这与广泛研究的其它商务数据有很大的差别。例如，在一个购物篮的数据库中，一般涉及成千上万的商品编号，因此被观察的项目全集很大。(2)DNA 序列通常都是很长的序列。例如，人类的基因组有大约 300 万核糖核酸组成。与之相比，通常的购物篮的数据库一般只有 10 到 20 的长度。这些特点使得 DNA 序列的挖掘与普通的事务数据库挖掘有很大的差别。现有的典型的序列挖掘算法，如 AprioriAll、GSP 等，很难适应 DNA 序列挖掘的要求。不论是从精度还是效率上都存在问题，因此 DNA 序列的挖掘需要在相应的挖掘理论及其有效算法方面进一步开展工作。

本文针对 DNA 序列的特点，设计了一种称为关联矩阵(Association Matrix)的数据结构。由于这种结构能够将超长的 DNA 序列压缩成空间可控的内存中，因此可以利用它形成新的高效的 DNA 序列挖掘算法。

2. 相关工作

在生物信息学领域，从 DNA 序列中寻找相似性片段(子序列)已经被广泛研究。1995 年，Hirosawa 给出了这一研究领域的基本问题和解决策略[2]。之后许多学者对 DNA 序列的相似性进行了相关的研究[3] [4]。例如，2012 年，Papapetrou 等认为有三种有效方法可以解决 DNA 序列的相似性问题：(1) 基于熵的递归分割技术；(2) 基于重要的统计参数指标来归纳；(3) 基于多数投票的策略[4]。这些方法有一定的使用价值，但是随着研究的深入它们的局限性也逐步暴露出来。由于相似性往往建立在两个序列的长期观察基础上，所以当确认两个生物体的 DNA 序列是否存在关系时，这些方法的有效性已经得到验证[3] [4]。事实上，在一个超长的 DNA 序列中，总是存在特殊的片段(如编码区、poly-区)，它们对于了解和解释 DNA 序列对应的生物学特征是关键的。因此，在一个或者多个超长的 DNA 序列中发现关键的子序列是一个重要的研究问题。

从数据挖掘的观点来看，从一个 DNA 序列中寻找关键子序列的问题可以依靠一些序列挖掘技术进行，其中最重要的技术就是频繁(子)序列挖掘。1995 年，Agrawal 第一次提出并讨论了频繁序列模式挖掘的概念，并且提出的频繁序列模式挖掘算法 AprioriAll 和 AprioriSome，成为这类算法研究的基础[5]。1996 年，Srikant 提出了序列模式挖掘的 GSP 算法，这是一种广度优先的自下而上的算法，而且是目前引用率较高的算法之一[6]。2000 年，Han 等提出了另外一种高效的序列模式挖掘算法 Free-Span [7]。之后 Han 及其相关学者在候选序列集选择等方面进一步改进，形成 PrefixSpan 等方法[8] [9]。目前为止，已经出现了很多序列模式挖掘算法[10]-[12]。Mao 等对序列挖掘的基本思想和典型算法给出了比较详尽的介绍[13]。然而，这些研究都是以商业应用的事务数据库为背景的，不是专门针对 DNA 序列的。

和本文在方法上高度相关的另一个领域是长序列中挖掘关键子序列的方法研究。Mannila 和他的同事已经在这一领域做出了突出贡献。他们的研究是以因特网上的点击事件序列为背景进行的，包括利用贝叶斯挖掘方法实现事件序列分析[14]；长序列中频繁场景片段(episode)发现[15]；长序列中强事件序列的发现[16]。他们的研究为本文 DNA 序列挖掘提供了可以借鉴的思想。

对本文方法给予启示作用的其它方法还有：长时间序列在线分割方法[17]；使用路径树查找数据规模较大的序列模式算法[18]；发现周期性通配符差距的频繁序列发现算法[19]；Wang 等专门为生物信息数据设计的序列挖掘算法[20]。

3. 关联矩阵设计和 DNA 序列中的关键序列挖掘算法

众所周知，细胞是用 DNA 来存储遗传信息的，而 DNA 则是由两条盘绕在双螺旋结构上的线性链组成。每条链可以看作是由腺嘌呤(A)，胸腺嘧啶(T)，胞嘧啶(C)和鸟嘌呤(G)形成的线性序列。两条线性链严格遵守碱基配对规则，即 A 与 T、C 与 G 配对出现。例如，一条 DNA 序列<ATGTC…>，与之配对的序列一定是<TACAG…>。对于现代生物计算而言，被符号化的 DNA 序列一旦被存储在数据库中，特别是成为网络上的公共数据集，那么它们就可以供科研人员研究和使用。

3.1. 关联矩阵

定义 1 (DNA 序列)：给定字符集合 E = {A,G,C,T}，一个 DNA 序列被表示为 S = <e_1,e_2,⋯,e_L>。即对任意的 i ∈ {1,2,⋯,L}，e_i∈E。

定义 2 (关联矩阵)：给定一个 DNA 序列 s = <e_1,e_2,⋯,e_L>，它的关联矩阵定义为$(P_{ij})_{M\times4}$，它列总是大小为 4，对应{A,G,C,T}的 4 个字符；它的行的大小是动态变化的，每行都与 s 的一个固定长度的子串对应；它的矩阵元素 P_{ij} 是行对应的 s 的子串在列对应的字符前出现的次数。特别地，当行对应的固定子串长度为 K 时，关联矩阵(P_{ij})也被称为 K 阶关联矩阵。

例 1：考虑一个 DNA 序列 s = <ATGTCGTGATTGCATTACTACT>，它的 1 阶关联矩阵图 1 所示。

3.2. 关键子序列

定义 3(关键子序列)：给定一个 DNA 序列 s 和它的一个关联矩阵$(P_{ij})_{M\times4}$，假如把关联矩阵的第 j 列对应的字符拼接在第 i 行对应的字符串之后就可以获得 s 的一个新的子序列，记为 $i\infty j$。设置一个最小关联阈值 Min-Ass，当关联矩阵中的某个 $P_{ij} \geq$ Min-Ass 时，那么利用 $i\infty j$ 操作得到的序列被称为 s 的一个关键子序列。特别地，假如 $i\infty j$ 的长度是 K，那么该序列被称为 s 的一个长度为 K 的关键子序列。

例 2：对于图 1 中的 DNA 序列和一阶关联矩阵，如果 Min − Ass = 2，则我们可以找到长度为 2 的关键子序列：<AT>，<AC>，<TA>，<TT>，<TG>，<CT>和<GT>。

定义 4(最大关键模式)：给定一个 DNA 序列 s，它的一个关键子序列被称为一个最大关键子序列，

$$\begin{array}{c} \\ A \\ T \\ C \\ G \end{array}\begin{array}{c} \begin{array}{cccc} A & T & C & G \end{array} \\ \begin{bmatrix} 0 & 3 & 2 & 0 \\ 2 & 2 & 1 & 3 \\ 1 & 2 & 0 & 1 \\ 1 & 2 & 1 & 0 \end{bmatrix} \end{array}$$

Figure 1. The first level association matrix of the DNA sequence s in Example 1
图 1. 例 1 中 DNA 序列 s 的 1 阶关联矩阵

当且仅当它是关键子序列但不是其他关键子序列的真子串。

3.3. DNA 序列中的关键子序列挖掘算法

事实上，在 DNA 序列中发现关键(子)序列模式是 DNA 分析的一个重要目标。利用关联矩阵结构，我们可以从较短的关键子序列中迭代生成较长的关键子序列。

例 3：在例 2 的基础上，我们可以逐步得到 2 阶、3 阶和 4 阶关联矩阵，如图 2(a)、图 2(b)和图 2(c)所示。

综合例 1 到例 3，对于例 1 给出的原始 DNA 序列，其挖掘关键子序列的挖掘过程归纳为：通过 1 阶关联矩阵，得到 s 的长度为 2 关键子序列集合是{<AT>, <AC>, <TA>, <TT>, <TG>, <CT>, <GT>}；由长度为 2 的关键子序列集合生成 2 阶的关联矩阵，进而得到长度为 3 的关键子序列集合是{<ATT>, <ACT>, <TAC>}；由长度为 3 的关键子序列集合生成 3 阶的关联矩阵，进而得到长度为 4 的关键子序列集合是{<TACT>}；显然，没有发现长度为 5 的关键子序列。此外，对应的最大关键子序列集合也可以被发现，即{<TG>, <GT>, <ATT>, <TACT>}。

基于例 1 到例 3 的解决问题的思路，利用长度递增的逐步迭代思想，算法 AM 给出了从一个 DNA 序列中发现关键子序列集合的过程描述。

算法：AM，DNA 序列中关键子序列挖掘算法

输入：DNA 序列 *s*；最小关联度 *Min-Ass*
输出：*s* 的关键子序列集合 *KS*.

```
1. k←1; M←4; row-set←{A,T,G,C};
2. WHEN row-set is not null DO
3.         generate the s' Association Matrix with size k: (p_ij)_M*4;
4.         row-set←{}
5.         FOR i = 1 TO M
6.                 FOR j = 1 TO 4
7.                         IF p_ij>= Min-Ass THEN insert i∞j into row-set;
8.         add all elements of row-set into KS;
9.         updating M with the size of row-set; k++;
10. ENDDO;
11. Return KS.
```

4. 实验与分析

为了评估本文算法的有效性，我们从美国国家生物技术信息中心的网站(http://www.ncbi.nim.nih.gov)下载了实验数据集。该数据集来自于白细胞介素-6 的 DNA 序列(简称 IL-6)，该数据集长度是 12139。

实验是在一台 4GB 内存、使用英特尔酷睿 i3、1.40 GHz 处理器的计算机上进行的。采用的对比算法是目前引用率较高的序列挖掘算法 GSP [6]。实验的目标主要是检测本文算法的效率(包括时间效率和空

间效率)。

实验 1：(不同的最小关联度下的执行时间)。利用 IL-6 数据集，分别设置最小关联度(对应原始 GSP 算法的最小支持度)为 2%~10%，测试本文提出的 AM 算法和 GSP 算法的运行效率。图 3 给出了对应的实验结果。

图 3 表明：随着最小关联度(最小支持度)的增加，AM 和 GSP 算法的执行时间都在下降。这是因为更大的最小关联度(最小支持度)意味着将生成更少的关键子序列。然而，AM 算法在处理 DNA 序列时效率明显高于 GSP 算法。

实验 2 (不同的最小关联度下的内存空间)。利用 IL-6 数据集，分别设置最小关联度(对应原始 GSP 算法的最小支持度)为 2%~10%，测试本文提出的 AM 算法和 GSP 算法的内存使用情况。图 4 给出了对应的实验结果。

图 4 表明：随着最小关联度(最小支持度)的增加，AM 和 GSP 算法的内存空间使用都在下降。这是因为更大最小关联度(最小支持度)意味着将生成更少的关键子序列。然而，AM 算法在处理 DNA 序列时的空间占用明显优于 GSP 算法，尤其是在产生较多关键模式时。

实验 3：(不同的 DNA 序列长度下的执行时间)。利用 IP-6 数据集，我们截取不同容量的 DNA 序列。考察在固定的最小关联度为 5%时，AM 与 GSP 算法的执行时间的攀升情况。图 5 给出了对应的实验结果。

$$\begin{array}{c} \\ AT \\ AC \\ TA \\ TT \\ TG \\ CT \\ GT \end{array}\!\!\begin{array}{c} \begin{array}{cccc} A & T & C & G \end{array} \\ \begin{bmatrix} 0 & 2 & 0 & 1 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \end{array} \qquad \begin{array}{c} \\ ATT \\ ACT \\ TAC \end{array}\!\!\begin{array}{c} \begin{array}{cccc} A & T & C & G \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix} \end{array} \qquad \begin{array}{c} \\ TACT \end{array}\!\!\begin{array}{c} \begin{array}{cccc} A & T & C & G \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

(a) (b) (c)

Figure 2. The 2nd, 3rd and 4th level association matrixes of the DNA sequence in Example 1
图 2. 例 1 中 DNA 序列 s 的 2 阶、3 阶和 4 阶关联矩阵

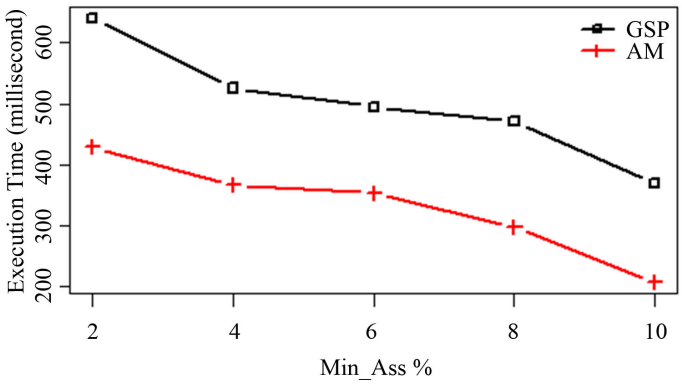

Figure 3. Curve: executing time changes of AM and GSP with increasing minimum association degrees
图 3. 最小关联度增加时 AM 与 GSP 执行时间的比较

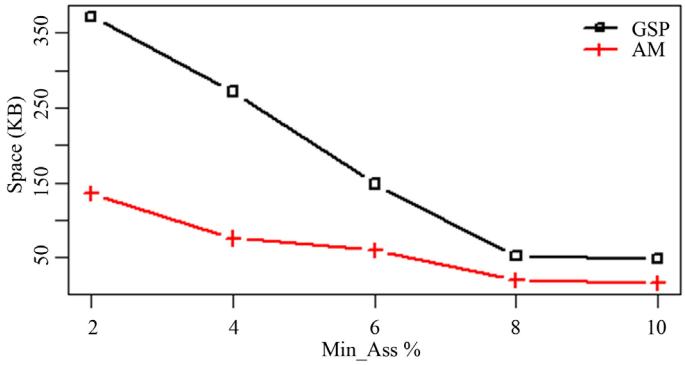

Figure 4. Curve: space changes of AM and GSP with increasing minimum association degrees
图 4. 最小关联度增加时 AM 与 GSP 内存空间的比较

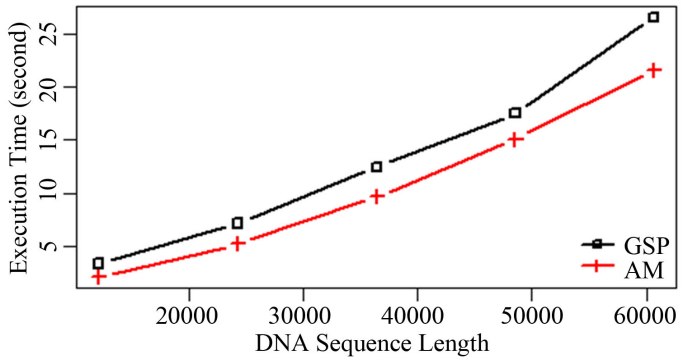

Figure 5. Curve: executing time changes of AM and GSP with increasing sizes of DNA sequences
图 5. 随着 DNA 序列长度增加时 AM 与 GSP 执行时间的比较

图 5 表明：在固定的最小关联度(支持度)下，随着 DNA 序列长度的增加，AM 和 GSP 算法的执行时间都会有所增加，但是它们都具有很好地可攀升性。当然，AM 算法整体是优于 GSP 算法的。

5. 结论

本文针对 DNA 序列挖掘的特殊需求，设计了一个关联矩阵的数据结构。它的空间紧凑型使得它更适合用于挖掘“小项目集但大容量”的序列。毫无疑问，DNA 序列属于这样类型的序列。然而，在现实世界中，这样的序列还有很多，如：生物信息学的蛋白质序列也仅有 20 个字母可以表示；随着时间变化的股票涨跌序列(可以只关心涨、平、跌 3 种状态)等。因此“小项目集但大容量的序列”是序列挖掘的一个具有很好研究和应用价值的分支。本文研究的问题和设计的方法只是其中的一部分，将来还有许多工作需要进一步探索。

基金项目

国家自然科学基金“基于数据分布评估和支持向量机方法的分布式数据流挖掘模型和算法研究”(No. 62173293)资助。

参考文献 (References)

[1] 朱扬勇，熊赟 (2007) DNA 序列数据挖掘技术. *软件学报*, **11**, 2767-2781.

[2] Hirosawa, M., Totoki, Y., Hoshida, M. and Ishikawa, M. (1995) Comprehensive study on iterative algorithms of mul-

tiple sequence alignment. *Computer Applications in the Biosciences*, **11**, 13-18. http://dx.doi.org/10.1093/bioinformatics/11.1.13

[3] Bell, D.A. and Guan, J.W. (2003) Data mining for motifs in DNA sequences. *Rough Sets, Fuzzy Sets. Data Mining and Granular Computing*: *Lecture Notes in Computer Science*, **2639**, 507-514. http://dx.doi.org/10.1007/3-540-39205-X_85

[4] Papapetrou, P. and Benson, G. and Kollios, G. (2012) Mining poly-regions in DNA. *International Journal of Data Mining and Bioinformatics*, **6**, 406. http://dx.doi.org/10.1504/IJDMB.2012.049278

[5] Agrawal, R. and Srikant, R. (1995) Mining sequential patterns. *Proceeding of the 6th International Conference on Data Engineering*, Taipei, 6-10 March 1995, 3-14. http://dx.doi.org/10.1109/icde.1995.380415

[6] Srikant, R. and Agrawal, R. (1996) Mining sequential patterns: Generalizations and performance improvements. *Proceedings of the 5th International conference on Extending Database Technology*: *Advances in Database Technology*, Avognon, 25-29 March 1996, 3-17. http://dx.doi.org/10.1007/bfb0014140

[7] Han, J. and Pei, J. (2000) Free-span: Frequent pattern-projected sequential pattern mining. *Proceedings of 6th International Conference of Knowledge Discovery and Data Mining*, Boston, 20-23 August 2000, 355-359.

[8] Pei, J., Han, J., Asl, B., Chen, Q., Dayal, U. and Hsu, M. (2001) PrefixSpan: Mining sequential patterns efficiently by prefix-projected patterns growth. *In Proceeding of the 17th International Data Engineering*, Heidelberg, 2-6 April 2001, 215-224.

[9] Pei, J., Han, J., Asl, B., Wang, J., Pinto, H., Chen, Q., Dayal, U. and Hsu, M. (2004) Mining sequential patterns by pattern-growth: The Prefix-Span approach. *IEEE Transactions on Knowledge and Data Engineering*, **16**, 1424-1440. http://dx.doi.org/10.1109/TKDE.2004.77

[10] Mohammed, J. (2001) SPADE: An efficient algorithm for mining frequent sequences. *Machine Learning*, **42**, 31-60. http://dx.doi.org/10.1023/A:1007652502315

[11] Liao, Z.X., Peng, W.C. and Hu, X.Y. (2006) Mining multi-domain sequential patterns. *Workshop on Software Engineering, Databases, and Knowledge Discovery*, Taipei, 12-15 December 2006, 334-339.

[12] Liu, Y.C., Chen, L.C., Liu, C.W. and Tseng, V.S. (2014) Effective peak alignment for mass spectrometry data analysis using two-phase clustering approach. *International Journal of Data Mining and Bioinformatics*, **9**, 52-66. http://dx.doi.org/10.1504/IJDMB.2014.057780

[13] 毛国君，段丽娟，王实，石云 (2007) 数据挖掘原理与算法(第二版). 清华大学出版社，北京, 217-218.

[14] Arjas, E., Mannila, H., Salmenkivi, M., Suramo, R. and Toivonen, H. (1996) BASS: Bayesian analyzer of event sequences. *Proceedings of the 12th COMPSTAT*, Barcelona, 28-31 July 1996, 199-204. http://dx.doi.org/10.1007/978-3-642-46992-3_20

[15] Mannila, H., Toivonen, H. and Verkamo, I. (1997) Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*, **1**, 259-289. http://dx.doi.org/10.1023/A:1009748302351

[16] Mannilaa, H. and Salmenkivi, M. (2001) Finding simple intensity descriptions from event sequence data. *Proceedings of the 7th ACMSIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, 26 August 2001, 341-346. http://dx.doi.org/10.1145/502512.502562

[17] Keogh, E., Chu, S., Hart, D. and Pazzani, M. (2001) An online algorithm for segmenting time series. *Proceedings of IEEE International Conference on Data Mining*, San Jose, 29 November-2 December 2001, 289-296. http://dx.doi.org/10.1109/icdm.2001.989531

[18] Lee, G.L., Chen, Y.C. and Hung, K.C. (2012) Path tree: Mining sequential patterns efficiently in multiple data streams environment. *Proceedings of the International Computer Symposium ICS*, Hualien, 12-14 November 2012, 261-268.

[19] Wu, Y.X., Wang, L.L., Ren, J.D., Ding, W. and Wu, X.D. (2014) Mining sequential patterns with periodic wildcard gaps. *Applied Intelligence*, **41**, 99-116. http://dx.doi.org/10.1007/s10489-013-0499-4

[20] Wang, K., Xu, Y.B. and Yu, J.X. (2004) Scalable sequential pattern mining for biological sequences. *Proceedings of the 13th ACM International Conference on Information and Knowledge Management*, Shanghai, 3-7 November 2014, 178-187. http://dx.doi.org/10.1145/1031171.1031209