DNA 序列中弱信号基序查找算法比较与分析*)

王建新[1] 杨 德[1] 黄元南[1,2]

(中南大学信息科学与工程学院 长沙 410083)[1] (广东商学院信息学院 广州 510320)[2]

摘 要 在DNA序列中查找基序是生物信息学中一个重要的计算问题,人们针对这一计算问题提出了多种模型和算法。由于DNA序列数据的复杂性,在其中有许多是比强信号基序更难提取的弱信号基序。而目前植入(l,d)基序问题(*PMP*)和扩展植入(l,d)基序问题(EMP)是较适合模拟弱信号基序查找的问题模型。本文归纳分析了基序查找的基本方法、策略和基序模型,指出了各种策略和模型的优势与不足。在此基础上对现有的基于植入基序查找问题模型的主要弱信号基序查找算法进行了分析和实验评估,为选择计算方法查找弱基序信号提供了参考,并讨论了该方向上尚未解决的问题和发展趋势。

关键词 弱信号,基序查找,植入(l,d)问题,算法,一致序列,简图

Comparison and Analysis on Subtle Motifs Discovery Algorithms in DNA Sequence

WANG Jian-xin[1] YANG De[1] HUANHG Yuan-nan[1,2]

(School of Information Science and Engineering, Central South University, Changsha 410083, China)[1]

(School of Information, Guangdong University of Business Studies, Guangzhou, 510320, China)[2]

Abstract Finding motifs is a significant computational problem in bioinformatics, many models and algorithms have been proposed to solve this problem. By reason of the complexity of DNA sequence data, there exist lots of subtle motifs which are much more difficult to be found than strong signals. Up to now, the planted (l,d) motif problem and the extended planted (l,d) motif problem are two suitable models for finding subtle motifs. This paper generalizes and analyzes the methods and strategies of motif discovery and the motif model as well as points out their advantages and disadvantages. Based on the above work, this paper further assays some main present subtle motif discovery algorithms through experiments and make a reference for the users of motif discovery. This paper also provides discussion on some unresolved problems and development trend in this field.

Keywords Subtle signals, Motif discovery, Planted (l,d) problem, Algorithms, Consensus, Profile

1 引言

基序查找是破译基因调控机制的一个关键环节,目前虽然已经产生许多基于实验的基序查找方法,但是这些实验方法却花费很大并且很耗时,因此基序查找领域的趋势是采用基于微阵列的数据计算方法。求解这个问题也有一些通过实验与计算技术相结合的方式,例如一些实验技术如 Chip-chip 和基因表达微阵列能够识别、比较可能是某个给定的转录因子的绑定位点富裕基因区域集合,然后可以通过一些计算方法从生物序列中挖掘出一个相关的绑定基序。

人们最早发现的DNA信号是20世纪70年代发现的限制位点。这种信号相互之间严格匹配,是DNA序列中最简单的信号(属于强信号)。目前人们已经研究出许多模型和算法,能对强信号基序查找给出解决方案[1]。其中的一些概率算法如 GibbsDNA 算法[2]和 MEME 算法[3]先随机产生出一个候选基序作为种子,然后通过迭代优化来查找基序。贪婪算法 Consensus[4]选择一条序列中的子串作为种子,计算其IC值,然后不断添加其它序列来更新其IC值,并输出具有最高IC值的子串作为基序。这些算法虽然能够解决部分强信号基序查找问题,但是难以解决弱信号的查找问题。由于生物序列中大量的信号是非常复杂的,难免出现突变或替换、插入和间隔等情况,并且生物数据中常常会有噪声,使得基序查找问题变得非常困难(如在酵母基因提升子区域中的基序就是很复杂的,基序的用例之间并不严格匹配,而存在替换等情况),导致能发现强信号基序的算法变得不再适用。人们迫切需要适合弱信号基序查找的问题模型和算法。为此文献[1]和文献[5]先后提出了能较好地模拟弱信号基序查找的植入基序查找问题(PMP)和扩展植入基序查找问题(EMP)模型。此后人们根据这些模型提出了许多算法,这些算法采用的策略与用来表示基序的模型有所不同,算法的运算性能也有许多差别。面对众多的模型与算法,生物学家往往难以做出合适的分析和选择。

本文对弱信号基序查找问题模型 PMP 和 EMP,以及查找弱信号基序问题的典型非精确算法和精确算法进行了分析研究,对这些算法的性能进行了实验对比。通过对照分析总结归纳了弱信号基序查找算法研究的策略和进一步研究的方向。

2 弱信号基序查找问题模型

针对弱信号基序查找,人们提出了两种比较合适的模型植入(l,d)基序查找问题(PMP)[1]和扩展植入(l,d)基序查找问题(EMP)[5]。PMP是目前应用和采纳得比较广泛的一种,

*)国家自然科学基金重点项目:生物信息学中的相关组合理论和算法研究(60433020),新世纪优秀人才支持计划 No. NCET-05-0683,长江学者和创新团队发展计划资助 No. IRT0661。王建新 博士,教授,博士生导师,CCF高级会员,主要研究领域为计算机算法、网络优化理论、生物信息学;杨 德 硕士研究生,主要研究领域为生物信息学、计算机理论。

而 EMP 则是对 PMP 一种较完善的改进模型。下面分别介绍这两种模型。

(1)植入(l,d)基序查找问题(PMP)

植入(l,d)-基序查找问题定义[1]:给定 t 条长度均为 n 个核苷酸的样本序列,找在每条序列均出现一个长度为 l 且与一致性基序恰有 d 个失配字符的基序。

在 PMP 模型中,基序与各个用例之间正好有 d 个位置的失配,而任意两个用例之间最多可能出现 $2d$ 个位置的失配。当 d 较大时,会导致许多假的用例符合这一条件,大量的假用例使得信号查找问题变得异常困难。

文献[6]中对这一问题给出了近似的统计分析。假设 p_d 为一个给定的 l 长子串在一条随机 DNA 序列中以最多有 d 个字符发生替换的用例出现在某个给定位置的概率:

$$p_d=\sum_{i=0}^{d}\binom{l}{i}(3/4)^i(1/4)^{l-i} \tag{1}$$

那么这样的用例在 t 条长度为 n 的随机序列的每一条中至少出现一次的近似期望值为

$$E(l,d)=4^l(1-(1-p_d)^{n-l+1})^t \tag{2}$$

由此分析可知(9,2)、(11,3)、(13,4)、(15,5)问题的 $E(l,d)$值均大于 1,也就意味着随机序列中存在与真实基序同样好甚至更好的假基序用例,从而使查找基序变得非常困难,甚至不可解。

PMP 实际上是一种简洁的基序发现问题定义,在真实生物数据中的情况相对要复杂许多。例如 PMP 要求基序在每条序列中均出现一个用例,事实上有的序列中可能没有用例,而有的序列中可能有多个用例;PMP 要求每个用例与一致基序之间允许出现的替换位置正好为 d,而多数情况是最多为 d。又比如受噪声的影响,输入序列中很可能有假阳性序列(也就是不包含任何用例的序列)。文献[7]中指出在生物实验中除了得到一组绑定了转录因子的一个序列集合之外,可能还会得到另一个序列集合 F。这个集合没有被转录因子绑定,也可以认为 F 中不包含任何基序的用例。基于这样一个额外的信息提出了一般化植入(l,d)基序问题[7]:假设存在一个固定但是未知的长度为 l 的串 M(基序)。给定 t 条长度为 n 的序列,每条都含有一个 M 的植入用例;给定 f 条长度为 n 的序列,每条都不包含 M 的 d-变体。目标是要在没有任何关于植入 d-变体的位置的先验信息的条件下找出 M。与 PMP 相比,一般化植入(l,d)基序问题更加贴近生物数据的真实情况。而文献[5]指出了大多数基序发现算法的一个不足之处是要求输入基序的长度信息,然而多数情况下基序的长度信息是未知的。更为可能的状况是了解基序用例之间的进化距离也就是替换比率 d/l,为此文献[5]引入了扩展(l,d)-基序问题。

(2)扩展植入(l,d)基序查找问题(EMP)

EMP 问题定义[5]:给定 t 条长为 n 的输入序列,其中每条序列包含长度为 L 的基序 M 的 0 个或多个植入(l,d)用例,序列集中总共有 k 个植入(l,d)用例。要求在不知道长度 L 和植入(l,d)用例在序列中的位置的情况下,找出 M。其中植入(l,d)用例是与 M 在任意长度为 $l(l\leqslant L)$的窗口中最多有 d 个位置的替换。文中还给出了最大基序的定义。

文献[8]还减少了另一个输入参数 k,将 EMP 问题发展为 FEMP 问题。其依据在于 EMP 中对最大基序的定义不够合理,比如难以判断一个有更多植入(l,d)用例长度较短的基序 M_1 与有较少植入(l,d)用例但长度更长的基序 M_2 哪个才是最大基序。

相对而言,基于 EMP 问题模型的算法需要更少的输入参数,更加符合在新的生物数据集上的实验要求和实际情况。

3 基序模型

虽然基于弱信号基序查找的问题模型有各种各样的计算方案产生,但是各个算法所采用的基序模型基本上是三种:正则表达式模型、矩阵表示模型和串模型。其中用得较多的是矩阵表示模型(又称为简图模型)和串模型(又称为一致序列模型)。接下来分别讨论这两种模型。

(1)矩阵表示模型

矩阵表示模型采用一个 $4*l$ 的实数矩阵(位置权重矩阵 PWMs 或者位置特性计分矩阵 PSSMs)来表示基序,其中第 j 列的数字表示每个字符的概率,因为具有概率性更加具有生物意义。尽管这种表示方法具有更佳的表述性,然而由于存在极大数量的 $4*l$ 实数矩阵构成的解空间,没有算法能够根据某些公式来保证找到最优解。采用这个模型的算法要么会得出子最优解,要么当基序长度大于 10bp 时需要太长的时间来运行[9],也就是说这种表示方法存在计算复杂性障碍需要克服。

(2)串模型

串表示模型是采用长度为 l 的字符如 A,C,T,G 来表示一个基序,在这种模型中不同串的个数是 4^l。在 l 相对较短的情况下,基于这种模型的算法能够找到"最佳"基序。然而不幸的是,在真实的生物数据中有很多基序是不能充分地以这种方式表示的,比如说在某个位置可以由多个字符来代表。这种生物基序在转录因子绑定位点中很常见。基于串模型的算法要找到这种在多个位置变异的基序 M 是困难的。尽管如此,因为串表示法简捷高效,大量的算法都针对这种模型给出解决方案。

4 弱信号基序查找算法分析

在求解强信号基序查找问题上人们已经给出了一些著名的算法,比如文献[2-4]等提出的方案,但是这些计算方法在弱信号数据集上未必可行。

文献[1]提出弱信号基序查找的 PMP 问题之后,人们针对弱信号基序查找问题给出了各种各样的计算方案。这些算法大致可以分为精确的和非精确的,EOMM[9],Voting[10],PMSP[11],YMF[12],MITRA[13],PMS1/PMS2[14]等算法属于精确算法,其它的如 WINNOWER/SP-STAR[1],Random Projection[6],Weeder[15],Pattern Branching/Profile Branching[16],MULTIPROFILER[17],MotifCut[18],UPNT[19]等则归属于非精确的范畴。精确算法能够保证输出正确的结果,但是一般来说比较耗时,在大数据集上或者查找 l 较大的基序信号时往往会变得不可行。非精确算法采用各种优化的策略降低计算复杂性,但是常常容易陷入局部最优,需要设法避免陷入局部的最优解。

虽然有各种各样的精确或非精确算法在查找基序的过程中被采用,但是基本的算法思想可以归结为从检查起始位置到模式空间再到样本空间的发展历程。下面分别讨论这些基本策略。

4.1 基序查找基本算法策略

(1)搜索起始位置的策略

查找基序,人们最初想到的一种简单办法是遍历序列中的基序起始位置(如贪婪算法 CONSENSUS[2],随机算法 GibbsDNA[3]),不过此种方案在大数量的数据集上却不可行。遍历序列中的基序起始位置首先需要定义基序起始位置为 $S\{S_1,S_2,\cdots,S_t\}$,从每个起始位置提取出来的长为 l 的子

串构建出一个矩阵简图，并选出每列分数最高的核苷酸组成一致序列。然后对这个一致序列进行计分 Score(S,DNA)，最后选出得分最高的一致序列作为基序输出。

显然，要检查所有的起始位置需要从{1,1,…,1}遍历到{$n-l+1,n-l+1,\cdots,n-l+1$}共有$(n-l+1)^t$个起始位置组合。对于每个起始位置的组合，计分公式需要进行 l 次操作，因此复杂度为 $O(ln^t)$。这在计算上是几乎不可能的，比如参数 $n=1000,t=10,l=10$ 的数据集，就需要大约 10^{30} 次计算，超过了亿年的计算时间。GibbsDNA 算法虽然采用了吉布斯采样来进行优化，但是由于起始位置的空间太大，在弱信号的情况下，随机选择的位置可能永远也不会接近全局最优的基序。对于大数据集，必须找出更高效的计算方案。

(2)搜索模式空间的策略

求最小总海明距离是一种与选择最高计分一致序列在计分思想本质上相同的方法，不过这种方法只需要遍历所有的 4^l 个长度为 l 的模式。因为这种类型的方法是基于模式的，又称为 PDA(Pattern-Driven Approach)(例如 YMF[12]、MDscan[20])。对于遍历的从 AA…A 到 TT…T 每个长度为 l 的模式，依次检查其在每条序列中出现的一个最佳用例（与其海明距离最小的长度为 l 的子串）并得出与这个用例之间的海明距离，将与所有最佳用例的海明距离累加得到这个模式的最小总海明距离计分，并输出具有最小总海明距离的模式作为最佳基序。

相比于 $(n-l+1)^t$ 个起始位置组合，PDA 方法的 4^l 个模式组合小了许多。不过，尽管目前查找弱信号基序最好的工具可能是 PDA[1]，但是这种方法对于大的 l 值(如 $l \geqslant 10$bp)而言仍然是不现实的，因为 4^l 个模式组合仍然不少(复杂度 $O(nt4^l)$)。

(3)基于样本空间的策略

绕过 PDA 计算复杂度仍然过高的途径是采用样本驱动方法 SDA(Sample-Driven Approach)，将搜索空间限制在样本序列中[1]。采用 SDA，如果偶然地样本中正好有一个与基序精确匹配的用例出现，这种方法就会得到跟 PDA 一样好的结果。如果没有的话，那么就希望在样本中出现的用例能够通过局部优化来揭示出这个模式。

SDA 的一个问题是难以将模式的用例与随机子串区分开来，它们有可能与模式很相似。对于弱基序信号，选出来的样本子串可能与全局最优基序离得较远，从而使结果陷入局部最优。既然 PDA 太耗时，而 SDA 常常错过弱模式，一种很自然的想法就是设计一种混合的方法来扩展 SDA 的搜索能力，同时又避免 PDA 的计算复杂性。扩展的样本驱动方法(ESDA:Extended Sample-driven Approach)[17]便是这样一种混合方法：通过产生样本中每个 l 长子串的所有 d-neighbors 来查找 (l,d)-基序 M(d-neighbors:是指两个模式之间最多有 d 个位置出现替换)。ESDA 中要检查的模式的数目为 $th\sum_{i=0}^{d}\binom{l}{d}3^d$，其中 t 是序列的条数，n 是序列的长度，对比于 PDA 的 4^l 减少了许多。

SDA 的另一个问题是需要保存一个大小为 4^l 的表(保存所有可能的模式)，相比较而言需要更大的内存空间。

尽管 ESDA 相比于 PDA 要快，在实际中还是太慢。文献[17]观察到在 ESDA 中产生的 neighbors 只有少数能够引导到真正的模式 P，而其他的大部分却不会帮助找到真模式。因而问题就是能否通过产生非常少的 neighbors 而不是为每个样本中的 l 子串产生 $\binom{l}{d}3^d$ 个 neighbors。文献[17]引入了一种 ESDA 的变体，采用 Multiprofiles 来限制这种穷举搜索，从而达到不损失准确率的前提下又提高效率的目标。这种方法能够找到弱信号，不过计算消耗依然很大。

4.2 基于 PMP 模型的弱信号基序查找算法

从文献[1]提出了 PMP 问题模型之后，产生了一系列的弱信号基序查找算法。为了比较这些算法的性能，本文根据这些算法所采用的基本技术思路挑选出比较典型的 WINNOWER，YMF，Random Projection，Pattern Branching，MULTIPROFILER，Voting，PMSP 算法及其相应的同类技术思想的算法和改进的算法进行归纳分析。

(1)非精确算法

WINNOWER 算法和 SP-STAR 算法[1]是首先针对弱信号基序查找问题提出的算法。这两个算法采用一种基于图论的计算方案，将 t 条输入样本序列中的每一个 l-mer 子串看作图中的一个点，每两个在不同序列中的子串如果其海明距离在 $2d$ 之内，就在代表它们的两个点之间连一条边，这样就将基序查找问题转化为在 t 分图中查找大小为 t 的团问题。WINNOWER 算法能够解决典型的(15,4)问题，但是在更为复杂的问题(如(15,5)等)上，由于边数目太多，计算复杂性显著增加($O(nt)^{t+2.376}$)。

cWINNOWER 算法[21]是对 WINNOWER 算法的改进，通过引入了一个 consensus 限定于同样一个团上，使得能够去除更多的假边，从而能够查找更加弱的信号，比 WINNOWER 算法的敏感度更高。文献[22]提出的 DPCFG 算法相比 WINNOWER 算法改进了图的构造方法，选出一条序列作为参考序列，其他序列中的 l 长子串与参考序列的 l 长子串(因为每条序列中都会有一个基序用例，那么参考序列中必有一个 l 长子串就是基序用例)之间满足海明距离在 $2d$ 之内就连一条边，依次如此构造出一个图集，然后算法采用动态规划技术找出以序列条数为大小的团。这个图集中用来查找基序的团要远大于 WINNOWER 算法中的团，所以算法更为高效。

文献[18]中提出的 MotifCut 算法也是基于图论的方法，但不再以查找最大团为中心思想，而是采用搜索最大密度子图(MDS)的策略。算法在构造图的过程中将样本序列中的所有 l 长子串转化成 l-mer 集合，所有 l-mer 集合都是图中的一个点，每一对点之间构建一条带权重的边。其中权重是根据两个点之间的相似程度(计算其失配的位置个数)来计算，并通过核苷酸背景分布一般化保证越相似的 l-mer 之间其边的权值越大。不太可能在背景分布中出现的两个 l-mer 之间的权值会得到拔升。一个基序可以看作是一些具有比较高的(相对背景分布)比对相似性的 l-mers 的集合。MotifCut 算法从一种新颖的角度出发查找基序，实验运行结果优于 MEME，AlignACE 等传统算法，不过对于弱信号基序查找问题，其准确性有待提高。

文献[6]提出的 Random Projection 算法采用随机投影的方式从 l 长的位置中选出 k 个位置将样本序列中的所有 l-mers 按这 k 个位置映射到哈希桶中，包含的 l-mers 超过给定域值的哈希桶中的所有 l-mers 作为候选基序进行 EM 提炼。如果经过一定次数的迭代，能够保证植入桶被提炼到，算法就能够输出真正的植入基序。Random Projection 算法实质上是为局部优化选择好的起始点的一种技术思想，算法在解决较为困难的植入(l,d)问题时取得了比 WINNOWER 算法好 20 至 30 倍的执行性能系数。

文献[23]采用统一投影替换了 Random Projection 算法的随机投影过程，用更少的投影次数达到了比随机投影更好的效果。文献[24]则改进了选桶策略，其中被选出来进行提炼的是其邻居桶包含 l-mers 足够多的桶，而不是 Random

Projection 算法中所取用的映射入 l-mers 足够多的桶。这一策略显著地减少了 EM 提炼的次数，选出来的桶具有更大的可能性是真正的植入桶。文献[19]发现在投影过程中减少冗余投影和在提炼过程中减少冗余桶并没有直接的关联，通过融合这两种策略构造出 UPNT 算法。UPNT 算法还进一步改进了选桶策略，在对桶计分的过程中不仅统计邻居桶中 l-mers 的数目而且加上本桶的 l-mers 数目。在不降低成功率的情况下同时减少了迭代次数和提炼桶的数目，从而达到了性能的进一步提升。在 20m 时间以内，求解(14,4)，(16,5)问题能够得到接近 1.0 的执行性能系数。

文献[17]提出的 MULTIPROFILER 算法通过产生样本中每个 l 长子串的所有 d-neighbors(d-neighbors 是指与本子串最多有 d 个位置出现替换的模式集合)来查找(l,d)-基序。算法基于一种 ESDA 的变体模型，通过引入 Multiprofiles 来限制搜索范围，不用为每个样本中的 l 子串产生 $\binom{l}{d}3^d$ 个 neighbors。算法在采用与 Random Projection 同样的计分公式时，实现了比 Random Projection 算法更高的成功率。

文献[16]引入了一种从样本子串分支的 Pattern Branching 算法来进行搜索，与 GibbsDNA 有点近似，但是 GibbsDNA 是从随机种子分支的，而这种方法是从基序空间的子集分支的，因此空间要小得多。算法对所有 neighbours 计分，选出最佳 neighbours 并最终从这些最佳 neighbours 找出基序 s。Pattern Branching 算法在几秒钟时间求解(15,4)问题达到 99.7%的成功率，时间性能显著优于 Random Projection 算法。文献[25]对 Pattern Branching 算法进一步优化加速，能够在 100s 内求解(13,4)问题，6min 中内解决(15,5)问题，2h 内解决(17,6)问题。

(2)精确算法

YMF[12]是一种枚举统计方法，穷举搜索所有长度为 l 的模式，保证找到具有最大 z-score 的所有基序。不过，由于这是一种 PDA 算法，在运行时间上消耗比较大。当 l 长度较大时，计算时间过长，算法只适合求解长度较短($l \leqslant 8$)的基序查找问题。

文献[13]提出了一种 MITRA 算法，原始 MITRA 算法即 MITRA-count 算法，采用后缀树的数据结构，解决了 SDA 空间消耗过大的问题。尽管付出了不断更新数据结构的代价，MITRA-count 算法能够得到比 SDA 更快的速度。另一个 MITRA 算法的版本是基于图的 MITRA-graph，主要思想就是一个模式的用例之间两两相似性信息能够极大地帮助加速 SDA。其借用了最大团的思路，通过构造一个图来建模两两相似性。MITRA-graph 在失配树的每个节点保存一个图中边的列表，在遍历树的同时有效地更新列表。MITRA-graph 比 WINNOWER 移除无用边更有效的途径是在找团的同时关注模式的前缀信息。算法能够在 5min 内解决(15,4)问题(内存 100M)，10min 内解决(14,4)问题(内存 210M)，20min 内解决(16,5)问题(内存 400M)。

Voting 算法[10]所基于的一个很简单的思想就是如果一个子串是一个基序 M 的变体，那么 M 也一定是这个子串的变体。因此，如果在每条序列中的一个 l 长子串都给 l 长序列 s 投且只投一票，那么最后得票数为 t 的序列 s 就有可能是植入的基序。不过由于这种基本思想是基于 PDA 的，l 长度较大时并不实用。改进的方法是将 4^l 长度为 l 的所有序列划分到几个群中，依据它们长度为 l' 的后缀来划分。对于每一个群，其输入序列的每一个子串将会得到处理，投票将会赋予其带特定后缀的变体。由于基本 Voting 算法复杂度随 l 和 d 的大小指数级增长，因此不适合解决大 l 和 d 的问题。对于 $l>15$ 的问题 Voting 引入随机投影思想进行了改进。基本 Voting 算法在求解(13,4)、(15,5)问题时使用的时间分别为 108s 和 22min，是目前能求解这两个问题最快的精确算法。

文献[11]中提出的 PMSP 算法求解(15,5)PMP 问题所需要的时间为 35min，但是 PMSP 还在 12h 的时间内解决了(17,6)PMP 问题，是目前唯一报告解决(17,6)问题的精确算法。PMSP 算法基于算法 PMS1，思想比较简单。首先产生一条序列中 l-mers 的邻域的集合 N，然后检查这个 N 中每一个 l 长子串 x，看在其他 $t-1$ 个序列中是否均有一个 l-mer 与其海明距离为 d。如果通过检查，那么 x 就添加到基序集合中。虽然基本思想较为简单，但是 PMSP 算法的空间和时间消耗都要远远优于 PMS1 算法，在(17,6)问题上还表现出比 Voting 算法更强的性能。

根据上述这些算法采用的基本策略和基序模型以及查找技术的汇总分类参见表 1。

表 1　基于 PMP 问题模型的弱信号基序查找算法分类表

算法	基本策略	基序模型	查找技术
WINNOWER	SDA	串	图，最大团
SP-STAR	SDA	串	图，最大团
cWINNOWER	SDA	串	图，最大团
MotifCut	SDA	矩阵	图，密度子图
Random Projection	SDA	串	随机投影，EM
UPNT	SDA	串	统一投影，EM
MultiProfiler	ESDA	矩阵	多重简图
Pattern Branching	ESDA	串	分支
Profile Branching	ESDA	矩阵	分支
YMF	PDA	串	枚举
MITRA	SDA	矩阵	后缀树
Voting	ESDA	串	枚举投票
PMSP	ESDA	串	枚举

4.3　基于 EMP 模型的弱信号基序查找算法

基于 PMP 模型的算法往往要求输入基序长度和用例与基序一致序列的失配位置数目 d，然而实际的生物实验中常常并不知道这两个参数。针对 PMP 模型的不足，文献[5]提出了扩展植入基序查找问题即 EMP 问题模型，并首先提出了基于这一模型的 Gemode 算法。

Gemode 是一种精确的枚举算法[5,26]，采用矩阵模型表示基序，算法主要有比较、聚簇和卷积三个阶段。Gmode 算法在初始阶段先找出所有在输入序列中至少有 k 个(l,d)变体的 l 长基序，然后通过比较筛选出能合并成 $l+1$ 长基序的 l 长基序，筛选条件是合并成的 $l+1$ 长基序至少应具有 k 个$(l+1)(l,d)$变体，这样地比较和合并就会不断得到更长的基序，直到找出最大基序。Gmode 算法由于初始阶段采用了与文献[1]类似的查找最大团的方式，算法的时间复杂度很大。当 d 较大时，算法过于耗时，比如查找一个长度为 14 的基序需要三个月的时间。

文献[27]提出的算法也通过三步求解问题。算法首先随机选择一条序列进行聚簇的处理，然后在第二步构建出比较接近一致序列的候选基序一致序列集合，第三步则在这个集合的基础上再利用样本序列进行评估选出最佳基序。解决 EMP 问题时算法相应地要增加迭代次数以保证正确性。算法时间性能比 Gemode 有所提高，但是准确度却减低了。

exVote 算法[8]是在 Voting 算法的基础上针对 EMP 问题模型给出的解决方案，是一种精确算法。文献[8]的实验表

明，算法只需要 197.5s 就能解决为(14,4)的基序查找问题(t=20,n=600)，在时间性能上优于 Gemode 算法。算法的基本思想是给输入序列中长度为 L 的子串的每个(l,d)变体都投一票，能够得到至少 k 张票的 L 子串将作为候选基序。目前求决 EMP 问题 exVote 算法的性能已经不错。

5 实验与分析研究

针对弱信号基序查找问题人们提出了许多算法。除了上面这些算法之外，还有许多采用其他技巧的计算方法。面对层出不穷、非常复杂的海量生物数据和各种计算方法，生物学家难以做出合理的选择。为了帮助生物学家更好地了解和分析各种基序查找算法的性能，人们对一些经典的算法进行了基准测试[28]。本文参照文献[28]的实验选出近期提出的部分解决弱信号基序查找问题的算法 MotifCut，Random Projection，UPNT，MULTIPROFILER，Pattern Branching，Profile Branching 和 Voting(本文选择算法的出发点：算法具有一定的代表性；算法的源代码能够获得；由于 PMP 问题模型目前的广泛性，选择的算法均为基于 PMP 问题模型的)进行了实验测试，并对算法性能进行评测和分析。

为了全面地评测算法性能，实验使用与文献[1,6,10,16,19,23,24,29]中相类似的方法来产生合成植入(l,d)数据集，构建了一批强信号数据集和两批弱信号数据集。强信号数据集中，l 和 d 均设置得较小(l=8,d=0)。弱信号数据集其中一批是选择文献[6]中认为较为困难的、较大的 l 和 d 的组合(也即植入用例与基序之间变异的位置较多)，l 和 d 的选择分别为(10,2)、(12,3)、(14,4)、(15,4)、(16,5)、(17,6)。对于每个(l,d)组合均构建 20 组，每组中均有 20 条 600bp 长的随机 DNA 序列(每组也构建一个相应的 l 长基序一致序列)，在每条序列中随机选择一个位置插入与其一致序列正好有 d 个位置发生变异的一个用例来替换原来在此位置的 l 长子串。另一批弱信号数据集则是选择序列长度分别为 1000bp 和 1600bp、序列条数各为 100 的 20 组(15,4)植入 DNA 序列数据。由于序列长度较长并且条数较多，数据中假信号出现的概率增加，也带来了问题复杂性的增强。

评测参数选择了大多数文献中采用的成功率、运行时间、执行性能系数 nPC、位点相关系数 nCC 以及内存消耗。其中成功率的计算标准与文献[6,16,19,23,24,28,29]中的定义一致，在 20 组数据中成功地找出了与一致序列相同的子串和对应的用例认为成功，比如 20 组中全部找出成功率就为 20/20。对执行性能系数的定义，文献[28]与文献[1]中其实本质上是一致的。文献[28]中的定义公式为

$$nPC=\frac{ntp}{ntp+nfn+nfp} \tag{3}$$

其中 ntp(true positives)是指预测用例与已知用例中相符的碱基位置数；nfn(false negatives)是指已知用例中没有被预测出的碱基位置数；nfp(false positives)是指预测用例与已知用例不相符的碱基位置数；ntn(true negatives)是指不属于预测用例和已知用例的碱基位置数。而文献[1]定义为

$$nPC=\frac{|K\cap P|}{|K\cup P|} \tag{4}$$

其中 K 是序列集合中植入用例所占的位点集合。P 是预测出的用例所占的位点集合，本文采用后面这一 nPC 的定义方式进行计算。运行时间和内存消耗均取各组数据的实际消耗的平均值，时间单位为 s，内存消耗单位为 MB。

5.1 植入强信号数据实验

在强信号植入数据集上的实验表明，大多数算法都能很好地解决用例与一致序列无变异或变异位置较少的数据中的基序查找问题。如表 2 所示，所有算法都找到了 20 组植入(8,0)数据中的全部基序；从运算时间来看，MotifCut 的平均运算时间最长(构图和查找最大密度子图时间复杂度太高)，而内存消耗方面 Pattern Branching 较高。Voting 算法在运行时间和成功率方面表现最优，但是其执行性能系数和位点的相关系数则不如其他算法，这与 Voting 算法运行的结果中假阳性信号过多有直接关系。

表 2 在强信号植入(8,0)数据集上实验结果对照(成功项表格中数字表示 20 组中成功找出基序的组数)

算法	MC	PrB	PaB	RP	UPNT	Vt
成功	20	20	20	20	20	20
时间	703.	3.2	1.4	9.3	0.7	0.01
内存	147	157	253	145	139	148
nPC	0.97	0.99	1	0.99	0.99	0.77
nCC	0.98	0.99	1	0.99	0.99	0.87

为简洁表示各算法名称，本文表中分别采用 MC 代表 MotifCut，PrB 代表 Profile Branching，PaB 代表 Pattern Branching，RP 代表 Random Projection，Vt 代表 Voting 算法。

5.2 植入(l,d)组合弱信号的数据实验

根据文献[6]中的统计模型，可以估算出更为困难的(10,2)、(12,3)、(14,4)、(15,4)、(16,5)和(17,6)这六个数据集中每条序列中出现一个一致序列的假信号的期望值分别为 $E(10,2)=6.1e-008$，$E(12,3)=3.19e-007$，$E(14,4)=4.2e-007$，$E(15,4)=2.17e-015$，$E(16,5)=2.33e-007$，$E(17,6)=0.88$。期望值大于 1 的话，基序就将被假信号完全覆盖，从众多假信号中找出基序也就变得异常困难。基于这些估计值期望值的均未超过 1，可以认为是能够从中找出基序信号的(文献[6]认为(17,6)属于几乎不可解的问题)。

表 3 在植入弱信号(l,d)合成数据集上的实验平均运行时间对比

l,d	MC	PrB	PaB	RP	UPNT	Vt
10,2	1019	3.8	1.4	56.9	1.7	0.6
12,3	357.6	16.5	5.8	218.8	12.4	12.3
14,4	4644	18.4	5.5	466.9	112.9	167
15,4	1163	26.4	7.1	127.1	7.9	314.5
16,5	1116	16.4	4.9	1008	953.4	8.1
17,6	5275	27.5	6.5	35658	6280	28.1

表 3 和表 4 分别为在这六个数据集上的实验各算法的运行时间和成功率对照。从表中可以明显看出，由于 l 和 d 的增加，数据中出现假信号的概率相应地增大，问题求解难度在不断增大。各算法在求解准确度和资源消耗上明显不如强信号基序查找的情况。实验还表明，基于简图的算法(Profile Branching 和 MotifCut)在运行时间和成功率方面表现不如基于一致序列的算法。

表 4 在植入弱信号(l,d)合成数据集上的实验成功率对比(表格中数字表示 20 组中成功找出基序的组数)

l,d	MC	PrB	PaB	RP	UPNT	Vt
10,2	1	5	19	4	20	20
12,3	0	0	20	0	20	20
14,4	0	0	3	0	19	20
15,4	0	13	20	11	20	20
16,5	0	0	0	0	20	0
17,6	0	0	0	0	5	0

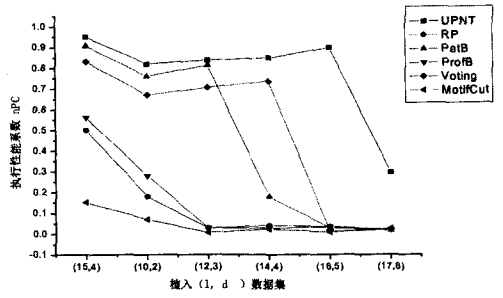

图1 在植入弱信号(l,d)合成数据集上的实验各算法执行性能系数 nPC

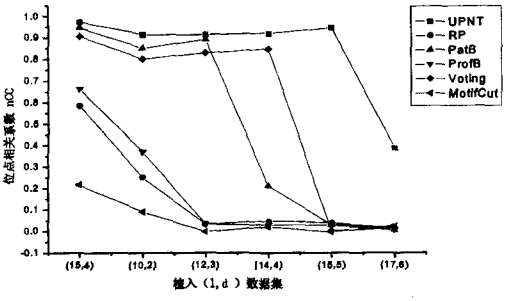

图2 在植入弱信号(l,d)合成数据集上的实验各算法的位点相关系数 nCC

从图1和图2各算法的执行性能系数和位点相关系数的表现中可以看出，随着问题难度的加大，各算法的性能明显下降，特别是在(17,6)数据上都只能得到约为零的结果。在这种更弱的数据上还需要研究性能更强的算法。

5.3 弱信号序列长度超过1000bp的实验

在超过1000bp的DNA弱信号查找实验中，本文只选择出前面实验中的三个已报告能求解超过1000bp长的序列查找问题的算法和另一个MultiProfiler算法进行了实验。选择MultiProfiler算法是根据文献[17]中其在长序列基序查找中的突出性能。从表5和表6中(其中MP代表MultiProfiler)可以看出，各算法基本上都能找出(15,4)的植入弱信号基序，而且 nPC 与 nCC 值都较高。在内存的消耗上变化不大，而时间明显增加。

表5 在植入(15,4)弱信号1000bp长序列数据集上的实验对照

算法	PrB	PaB	MP	Vt
nPC	0.81	0.88	0.88	0.79
nCC	0.89	0.94	0.94	0.83
成功	20/20	20/20	20/20	20/20
时间	519.13	194.41	1432.94	1860.35
内存	157.31	254.09	147.83	145.55

表6 在植入(15,4)弱信号1600bp长序列数据集上的实验对照

算法	PrB	PaB	MP	Voting
nPC	0.63	0.83	0.83	0.75
nCC	0.77	0.91	0.91	0.86
时间	1329.62	503.72	3745.03	2837.85
成功	17/20	20/20	20/20	20/20
内存	151.72	254.58	149.94	147.19

总的来说，基于简图基序模型(PWM或PSSM模型)的算法具有更强的统计表达性(生物学家更倾向于采用)，但是这些算法在解决 l 和 d 较大的基序查找问题上效率却较低，相对更为耗时。文献[9]提出的EOMM算法能够在可接受的时间内解决 l 在8以内的弱信号基序查找问题，但是对于 l 较大的问题仍然需要采用局部搜索策略。研究更为高效的基于简图的算法具有非常实际的意义。

基于一致序列表示基序的算法简单高效，目前求解PMP问题运行速率已经比较快，但是基于EMP模型以及结构化基序的查找问题算法还有待改进，而EMP模型以及结构化基序模型已经被证实更为贴近生物意义，因此基于这两种模型的算法将是今后研究的重点。

结束语 在DNA序列中查找基序信号在最近20年一直是生物信息学领域非常关注的问题。解析这一问题，有助于理解基因共同合作完成某些功能的机理。计算机学家倾向于使用更少的时间和空间来求解这一问题，而生物学家往往更多地倾向于关注统计特性和结果的准确度。

虽然针对某些强基序信号人们已经给出了一些性能不错的计算方法，并得以应用，在DNA序列中却仍有很多更为复杂的弱信号没有合适的计算策略。2000年以来，人们针对弱信号基序查找问题提出了一些模型和算法，这些算法采用的策略各不相同，在运算结果的计分和统计分析上也存在差异。

本文从算法所采用的基序模型和问题模型以及基本策略的角度进行了系统的分析和归纳总结。出于生物学家更多地关注结果的准确性和统计表达性，本文重点研究了近期提出的这些方面性能较为突出的基序查找算法。本文的实验表明，在弱信号基序查找问题上，现有的模型和算法还远没有达到能完美解决问题的程度。而且，由于DNA序列的复杂性和巨量性，层出不穷的新数据需要分析和解释，各种样本数据集适用的计算方案也会有差异。目前在DNA序列中弱信基序查找方面有几个方向值得关注：

(1)基于PWM或PSSM的算法在Cis-regulatory绑定信号的查找中优于其它算法，但是目前只能在可接受的时间内查找较短的基序信号。由于生物学家倾向于更具统计表达性的PWM或PSSM的基序模型，基于这种模型的基序查找算法将会继续成为研究的热点。

(2)植入基序问题(PMP)是一种针对弱信号基序查找的简易模型，虽然能够基本概括在DNA序列中的弱信号基序查找问题，但是真实生物序列的复杂性要求人们给出更强壮的实验模型。扩展植入基序问题(EMP)模型部分解决了PMP模型的不足，并越来越受到关注，但是模型仍然存在需要改进的方面。

(3)在转录因子绑定位点中更为典型的基序结构是一种由两个基序组成的中间带间隔的二分体基序，两个子基序中可能某个子基序只具有弱统计显著性，但是其组合结构却具有很强的统计显著性。如何解决这种强弱搭配的组合基序查找问题是一个重要的研究方向。

致谢 非常感谢M. Tompa, Benjamin Raphael, James King, H. C. M. Leung, A. Price, U. Keich, Eugene Fratkin等慷慨地提供了算法测试所需要的源代码。特别感谢提出植入(l,d)Motif查找问题(PMP)的作者Sing-Hoi Sze给予的诚恳建议和意见。

参考文献

[1] Pevzner P A, Sze S H. Combinatorial approaches to finding subtle signals in DNA sequences [A]// Proceedings of the Eighth International Conference on Intel-ligent Systems for Molecular Biology (ISMB-2000). San Diego, California: AAAIPress [C], 2000:269-278

[2] Lawrence C, Altschul S, Boguski M, et al. Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment [J]. Science, 1993, 262(5131):208-214

[3] Bailey T L, Elkan C. Unsupervised learning of multiple Motifs in

biopolymers using expectation maximization [J]. Machine Learning, 1995, 21(1/2): 51-80
[4] Hertz G, Stormo G. Identifying DNA and protein pat-terns with statistically significant alignments of mul-tiple sequences [J]. Bioinformatics, 1999, 15: 563-577
[5] Styczynski M P, Jensen K L, Rigoutsos I, et al. An ex-tension and novel solution to the (l,d)-motif challenge problem [J]. Genome Informatics, 2004, 15: 63-71
[6] Buhler J, Tompa M. Finding Motifs using random pro-jections [J]. J Comput Biol, 2002, 9(2): 225-242
[7] Leung H C M, Chin F Y L. Generalized Planted (l, d)-Motif Problem with Negative Set [A]// WABI [C]. 2005: 264-275
[8] Leung H C M, Chin F Y L. An Efficient Algorithm for the Extended (l,d)-Motif Problem with Unknown Number of Binding Sites [A]// Proceedings of the IEEE 5th Symposium on Bioinformatics and Bioengi-neering [C]. Min-neapolis, Minnesota, USA: IEEE Press, 2005: 11-18
[9] Leung H C M, Chin F Y L. Finding Exact Optimal Motif in Matrix Representation by Partitioning [J]. Bio-informatics, 2005, 21(2): 86-92
[10] Chin F Y L, Leung H C M. Voting algorithm for discovering long Motifs [A]// Proc. of Asia-Pacific Bio-informatics Conference[C]. 2005: 261-272
[11] Davila J, Balla S, Rajasekaran S. Space and Time Efficient Algorithms for Planted Motif Search [A] // Second International Workshop on Bioinformatics Research and Applications (IWBRA) [C]. Accepted
[12] Sinha S, Tompa M. Discovery of novel transcription factor binding sites by statistical overrepresentation [J]. Nucleic Acids Res, 2002, 30(24): 5549-5560
[13] Eskin E, Pevzner P A. Finding composite regulatory patterns in DNA sequences [J]. Bioinformatics, 2002, 18(1 Suppl): 354-363
[14] Rajasekaran S, Balla S, Huang C H. Exact algorithms for planted motif challenge problem [A]// APBC[C]. 2005: 249-259
[15] Pavesi G, Mauri G, Pesole G. An algorithm for finding signals of unknown length in DNA sequences [J]. Bioinformatics, 2001, 17 (1. Suppl): 207-214
[16] Price A, Ramabhadran S, Pevzner P A. Finding subtle Motifs by branching from sample strings [J]. Bio-informatics, 2003, 19(2. Suppl): 149-155
[17] Keich U, Pevzner P A. Finding motifs in the twilight zone [J]. Bio-informatics, 2002, 18(10): 1374-1381
[18] Fratkin E, Naughton B T, Brutlag D L, et al. MotifCut: regulatory motifs finding with maximum density sub-graphs [J]. Bio-informatics, 2006, 22: 150-157
[19] Wang J X, Yang D. UPNT: Uniform Projection and Neighbourhood Thresholding Method for Motif Discovery [J]. International Journal of Bioinformatics Research and Applications (IJBRA). Accepted
[20] Liu X S, Brutlag D L, Liu J S. An algorithm for finding protein-DNA binding sites with applications to chroma-tin-immunoprecipitation microarray experiments [J]. Nat. Biotechnol, 2002, 20 (8): 835-839
[21] Liang S, Samanta M P, Biegel B A. cWINNOWER algorithm for finding fuzzy dna motifs [J]. J. Bioinfo. Comput. Biol, 2004, 2 (1): 47-60
[22] Yang X, Rajapakse J C. Graphical Approach to Weak Motif Recognition [J]. Genome Informatics, 2004, 15(2): 52-62
[23] Raphael B, Liu L T, Varghese V. A uniform projection method for Motif discovery in DNA sequences [J]. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2004, 1 (2): 91-94
[24] King J, Cheung W, Hoos H H. Neighbourhood Thresh-olding for Projection-based Motif Discovery [J]. Bioinformatics, Accepted
[25] Davila J, Rajasekaran S. Extending Pattern Branching to Handle Challenging Instances [A] // Proceedings of the Sixth IEEE Symposium on Bio-Informatics and BioEn-gineering (BIBE'06) [C]. 2006: 65-69
[26] Jensen K L, Styczynski M P, Rigoutsos I, et al. A generic motif discovery algorithm for sequential data [J]. Bioinformatics, 2006, 22(1): 21-28
[27] Yangb X, Rajapakse J C. Robust Algorithm for Finding Weak Motifs [A] // Proceedings of the 2005 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology[C]. 2005: 1-6
[28] Tompa M, Li N, Bailey T L, et al. Assessing computational tools for the discovery of transcription factor binding sites [J]. Nat Biotechnol, 2005, 23(1): 137-144
[29] 王建新，黄元南，陈建二. 一种基于彩色编码的基序发现算法 [J]. 软件学报，2007，18(6)：1298-1307

(上接第 187 页)

[4] Debreceny R S, Gray G L, Ng J, et al. Embedded audit modules in enterprise resource planning systems: implementation and functionality [J]. Journal of Information Systems, 2005, 19(2): 7-27
[5] Vasarhelyi M A, Halper F B. The continuous audit of online systems[J]. Auditing: A Journal of Practice and Theory, 1991, 10 (1): 110-125
[6] Woodroof J, Searcy D. Continuous audit - Model development and implementation within a debt covenant compliance domain [J]. International Journal of Accounting Information Systems, 2001, 2(3): 169-191
[7] Rezaee Z, Sharbatoghlie A, Elam R, et al. Continuous auditing: building automated auditing capability[J]. Auditing: A Journal of Practice and Theory, 2002, 21(1): 147-163
[8] Murthy U S, Groomer S M. A continuous auditing web services model for XML-based accounting systems [J]. International Journal of Accounting Information Systems, 2004, 5(2): 139-163
[9] Chen Wei, Wang Hao, Zhu Wenming. Study on data-oriented IT audit used in China[A]// Zhu Qingsheng, ed. Proceedings of the 11th Joint International Computer Conference[C]. Singapore: World Scientific Publishing, 2005: 666-669
[10] 国家 863 计划审计署课题组. 计算机审计数据采集与处理技术研究报告[M]. 北京：清华大学出版社，2006
[11] 武海平，余宏亮，郑纬民，等. 联网审计系统中海量数据的存储与管理策略[J]. 计算机学报，2006，29(4)：618-624
[12] 陈伟，刘思峰，Qiu Robin. 审计数据处理方法研究综述[J]. 统计与决策，2007(3)：129-131
[13] Verykios V S, Elmagarmid A K, Houstis E N. Automating the approximate record matching process[J]. Journal of Information Sciences, 2000, 126(1/4): 83-98
[14] 邱越峰，田增平，季文赟，等. 一种高效的检测相似重复记录的方法[J]. 计算机学报，2001，24(1)：69-77
[15] Monge A E. Matching algorithms within a duplicate detection system[J]. IEEE Data Engineer Bulletin, 2000, 23(4): 14-20
[16] 俞荣华，田增平，周傲英，等. 一种检测多语言文本相似重复记录的综合方法[J]. 计算机科学，2002，29(1)：118-121
[17] 陈伟，刘思峰，邱广华. 计算机审计中数据处理新方法探讨[J]. 审计与经济研究，2006，21(1)：37-39，48
[18] 张进，易仁萍，陈伟. 计算机审计中电子数据的清理研究[J]. 审计研究，2004(6)：21-25
[19] Li W S, Clifton C. Semantic integration in heterogeneous databases using neural networks [A]// Bocca J B, Jarke M, Zaniolo C, eds. Proceedings of the 20th International Conference on Very Large Data Bases[C]. Santiago: Morgan Kaufmann, 1994: 1-12
[20] 陈伟，丁秋林，谢强. 交互式数据迁移系统及其相似检测效率优化[J]. 华南理工大学学报(自然科学版)，2004，32(2)：58-61
[21] Navarro G. A guided tour to approximate string matching [J]. ACM Computing Surveys, 2001, 33(1): 31-88
[22] Vidal E, Marzal A, Aibar P. Fast computation of normalized edit distances [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1995, 17(9): 899-902
[23] 陈伟，丁秋林. 数据清理中编辑距离的应用及 Java 编程实现 [J]. 电脑与信息技术，2003，11(6)：33-35，60
[24] 陈伟，张金城，Qiu Robin. 审计数据处理实验中模拟数据生成系统的研究[J]. 计算机工程，2007，33(19)：54-56