生物物理学报　第二十一卷　第二期　二〇〇五年四月
ACTA BIOPHYSICA SINICA　Vol.21 No.2　Apr. 2005

DNA序列中模式发现的一种快速算法

李冬冬,　王正志,　杜耀华,　晏　春
（国防科技大学自动控制系，长沙 410073）

摘要：模式发现是生物信息学的一个重要研究方向，但目前的大部分算法还不能保证获得最优的模式。文章推导了针对三个序列片段相似性关系的判据，将其作为剪枝规则，提出并实现了一种深度优先的穷举搜索算法——判据搜索算法（criterion search algorithm, CRISA），理论分析表明，对绝大多数模式发现问题，CRISA 具有多项式的计算时间复杂度和线性的空间复杂度。对仿真的和实际的生物序列数据的测试也表明，CRISA 能够快速而完全地识别出序列中所有的模式，具有优于其它算法的总体评价，能够应用于实际的模式发现问题。

关键词：模式发现；判据；深度优先搜索
中图分类号：Q61

1 引　　言

所谓模式（motif）就是指 DNA 序列中保守的序列片段。通常人们认为 DNA 序列总是处于不断的突变过程中，而其中的某些区域，比如启动子区域，由于对生物体的生存具有至关重要的意义，因而在进化的过程中更保守一些，并表现为 DNA 序列中的模式。另一方面，一个保守的序列片段往往意味着一个重要的功能区域，它们通常对应着 DNA 序列中重要的功能性或结构性元素。从 DNA 序列中发现这些模式的过程就是模式发现（motif finding），它是生物信息学的一个重要研究内容。模式发现有助于决定序列的功能和阐明序列之间的进化关系，并且在识别 DNA 序列的调控信号方面有着重要的应用。

模式发现的问题可以描述为[1]：给定一组（n 个）DNA 序列（称为样本），每个序列的长度为 N，寻找一个长度为 l 的序列片段（模式），它至少在 k 个样本中出现，在每次出现中最多可以有 d 个不匹配的碱基。这样的序列片段称为(l,d)–k 模式，它在样本中的出现称为实例。

对模式发现问题已经提出了若干算法[2-4]，但是，Pevzner 和 Sze[1]的工作表明，这些算法对一个(15,4)的模式法发现问题（所谓的挑战问题，以下称标准测试问题）的执行效果都不理想。因此，他们提出一个称为 WINNOWER 的算法[1]来解决这一问题。随后，一些更加有效的算法也被提了出来，包括 Bulher 和 Tompa[5]的随机投影算法、Keich 和 Pevzner[6,7]的 MULTIPROFILER 算法、Eskin 和 Pevzner[8]的 MITRA 算法以及 Price 等[9]的 Branching 算法等，它们都能够较好地解决标准测试问题。其中，MITRA 算法是穷举搜索算法[8]，它能够确保找到最佳模式（如果存在的话），其他算法有的并非穷举搜索算法[5,9]，有的则是采用了启发式规则的穷举搜索算法[1,6,7]，然而都存在不同程度的丢失最优解的可能。另一方面，对标准测试问题的计算结果表明，MITRA 算法所需的计算时间要远远多于其他的算法，其内存使用量也是相当大的，这是它的不足之处。

最简单的模式发现算法是基于模式的算法[10,11]（pattern driven algorithm, PDA），它依次检验所有的 4^l 个长度为 l 的序列片段，对它们在 DNA 序列中出现的实例给出得分，从而找出其中具有较高得分的模式。由于 PDA 的计算复杂度是 l 的指数函数，因此，对于较大的 l，这样的算法是不现实的。在 PDA 的基础上，人们又提出了基于样本的算法[12-14]（sample driven algorithm, SDA），其基本思想是对样本中的每个长度为 l 的序列片段，计算出它的满足一定条件的相邻序列片段集合，并在这个集合中进行搜索。由于此时的搜索空间（相邻

收稿日期: 2004-06-04
基金项目：国防科大基础研究项目 JC02-03-021
通讯作者：李冬冬，电话：(0731)4574991，
E-mail：Li_dong_dong05@sina.com

片段集合）远远小于基于模式的算法的搜索空间，因而能够获得高得多的计算效率。目前较为有效的算法都利用了这一思想，只不过在减小搜索空间的策略上各不相同，比如随机投影算法[5]就是采用的随机性优化过程，而 PatternBranching 算法[9]采用的是启发式的优化过程，MITRA 则是采用 SDA 进行穷举搜索的一个例子。

本文提出了一种新的 SDA 穷举搜索算法：判据搜索算法（criterion search algorithm，CRISA），它采用本文中提出的针对三个序列片段相似性关系判据作为剪枝规则，能够有效地裁减搜索空间，从而获得较高的执行效率。对算法的理论分析和实验测试（仿真数据和真实的生物序列数据）表明，CRISA 具有良好的执行效率，能够快速而完全地识别出序列中所有的模式，这使得它具有优于其它算法的总体评价，能够应用于实际的模式发现问题。

2 方 法

当我们从一组序列中寻找一个模式的时候，我们实际寻找的是它在每个序列中的实例，之后再从这些实例中恢复出这个模式。例如，对一个(l,d)–k的模式发现问题（样本集合为 n 个长度为 N 的序列），我们首先从 k 个不同的样本中选择 k 个长度为 l 的子序列（这 k 个子序列称为一个子序列组，或者简称为一个组），然后尝试恢复出一个模式，使得每个子序列与该模式之间不超过 d 个不匹配的碱基（也称为两者的距离不超过 d），如果能够恢复出这样的一个模式，那么这个组就是有效的。从样本序列中找出所有的有效组是模式发现问题的关键，一旦某个模式的有效组被找到，就可以采用一些简单的方法（比如位置权重矩阵）来恢复这个模式。为了使讨论简洁，这里我们假设 $k=n$ 成立（即在每个样本中都至少存在一个实例），由此得到的算法与没有这个条件时的算法之间的区别是很小的，我们将在后文讨论这一差别。

很明显，最简单也是最直接的方法是测试所有可能的子序列组，但这一算法的复杂度是样本数目的指数函数，因而是不现实的。实际上，在通常的模式发现问题中，绝大多数的子序列组是无效的，它们不可能被用于恢复出一个模式，这就需要一种能够及早判断出无效子序列组的方法，它可以在完全列出一个子序列组之前就判断出它是否有效，从而提高算法的效率。

一种简单的判断方法建立在这样一个事实的基础之上：一个模式的任何两个实例之间的距离不超过 $2d$。由此可以得到如下的必要条件：

$$d(\mathrm{P}_{i_1 j_1},\mathrm{P}_{i_2 j_2})\leqslant 2d,\ i_1\neq i_2,\ 0\leqslant i_1,\ i_2\leqslant n-1 \qquad (1)$$

即：一个有效的子序列组，其中的任意两个子序列之间的距离不超过 $2d$。公式（1）中，$d(X,Y)$ 为序列 X 和 Y 之间的距离，i_k 是子序列组中的第 k 个子序列。我们称这样的一个必要条件为一个判据，这样，公式（1）就是一个判据，下文称之为判据 1。

使用判据 1，我们能够设计一种从样本中寻找所有有效组的算法。首先，我们从第一个样本中选择一个子序列，并把它加入到候选组中，在此之前这个候选组是一个空集。然后，对下一个样本进行扫描，寻找一个子序列，它对候选集中的每个元素都满足判据 1。如果找到了这样的一个子序列，将它加入到候选集中，并对下一个样本继续执行这一操作，直到所有的样本都完成了扫描或者在某一个样本中无法找到一个满足条件的子序列为止。在前一种情况下，候选组中所包含的子序列就是一个有效组，我们就可以从中恢复出一个模式。而在后一种情况下，就需要向前回溯一步，进行下一次扫描。当第一个样本中所有的子序列都被扫描过以后，算法结束。

虽然这个算法很简单，它却具有一个很好的特性：如果样本中存在一个(l,d)–k 模式，此算法一定能够把它找出来。然而，另一方面，它也有着致命的缺陷：由于判据 1 仅仅检查两个子序列之间的关系，判据太松，许多不能恢复出模式的随机组也能够满足这个判据，因此，真正的有效组将会被这些漏网的无效的随机组淹没。

显然，一个判据能够检查越多的子序列之间的关系，就能够排除掉越多的无效组，从而能够更容易地发现真实的模式。按照这一思想，我们提出了如下针对三个子序列的判据，并且实现了判据搜索算法 CRISA。

对任意的三个子序列 O、Q 和 R，设 $d(O,Q)=x$。因此，在 O 和 Q 之间存在 x 个不同的碱基，其余的 l–x 个碱基是相同的，不妨假设子序列的前 x 个碱基是不同的。这样，这三个子序列可以分别写成 $O=O_1+O_2$、$Q=Q_1+Q_2$ 和 $R=R_1+R_2$，它们满足：

$$|O_1|=|Q_1|=|R_1|=x,\ |O_2|=|Q_2|=|R_2|=l-x \tag{2}$$

$$d(O_1,Q_1)=x,\ d(O_2,Q_2)=0 \tag{3}$$

其中$|Q|$表示序列片段 Q 的长度。定义两个序列 A 和 B 的复合序列 C 为它们的各个位置上相应碱基的“或”操作，即 C 中的每个碱基表示 A 和 B 相应位置上的两个碱基的并集，记为 $C=A\cup B$。定义复合序列 C 与普通序列 A 之间的距离为：

$$d(C,A)=\sum_{i=1}^{l} t_i \tag{4}$$

$$t_i=\begin{cases}0, & a_i\in c_i\\ l, & \text{其他}\end{cases} \tag{5}$$

设 $d(O_2,R_2)=d(Q_2,R_2)=z$，$d(O_1\cup Q_1,R_1)=y$，如果这三个节点是来自于一个(l,d)模式，那么下面的判据不等式组成立：

$$x\leqslant 2d, y\leqslant 2d, z\leqslant 2d \tag{6-1}$$

$$z\leqslant 2d-\left\lfloor\frac{x}{2}\right\rfloor \tag{6-2}$$

$$y\leqslant 3d-x \tag{6-3}$$

$$y+z\leqslant 4d-x-\left\lfloor\frac{x}{2}\right\rfloor \tag{6-4}$$

这里，$\lfloor x\rfloor$是不小于 x 的最小整数。

证明：

不等式（6-1）显然是成立的。

对（6-2），如果我们把模式 M 也写成 $M=M_1+M_2$ 的形式，那么，由于 $d(O_2,Q_2)=0$，易知 $d(O_2,M_2)=d(Q_2,M_2)$。

由于

$$d(O,M)=d(O_1,M_1)+d(O_2,M_2)\leqslant d$$

$$d(Q,M)=d(Q_1,M_1)+d(Q_2,M_2)\leqslant d$$

并且：

$$x=d(O_1,Q_1)\leqslant d(O_1,M_1)+d(Q_1,M_1)$$

故有：

$$d(O_1,M_1)+d(Q_1,M_1)+d(O_2,M_2)+d(Q_2,M_2)\leqslant 2d$$

$$\Rightarrow\quad x+2d(Q_2,M_2)\leqslant 2d$$

$$\Rightarrow\quad d(Q_2,M_2)\leqslant d-\left\lfloor\frac{x}{2}\right\rfloor \tag{7}$$

在公式（7）的最后一步推导中，我们用到了 $d(Q_2,M_2)$是整数这个性质。从假设中我们知道，$d(R_2,M_2)\leqslant d(R,M)\leqslant d$，因此：

$$\begin{aligned}z=d(Q_2,R_2)&\leqslant d(Q_2,M_2)+d(R_2,M_2)\\&\leqslant d-\left\lfloor\frac{x}{2}\right\rfloor+d=2d-\left\lfloor\frac{x}{2}\right\rfloor\end{aligned} \tag{8}$$

从而（6-2）式成立。

为证明（6-3），我们将下标为 1 的部分序列再进行分解，由 $d(O_1\cup Q_1,R_1)=y$ 可知，三个片段（O_1、Q_1和R_1）在 y 个位置上的碱基各不相同。因此，这些片段可以写成$O_1=O_{11}+O_{12}$，等，它们满足：

$$d(O_{11},Q_{11})=d(O_{11},R_{11})=d(Q_{11},R_{11})=|O_{11}|=y \tag{9}$$

使用条件 $d(O_1\cup Q_1,R_1)=y$ 和公式（9），得 $d(O_{11}\cup Q_{11},R_{11})=y$。

由于

$$d(O_1,M_1)=d(O_{11},M_{11})+d(O_{12},M_{12})\leqslant d$$

$$d(Q_1,M_1)=d(Q_{11},M_{11})+d(Q_{12},M_{12})\leqslant d$$

故有：

$$d(O_{11},M_{11})+d(Q_{11},M_{11})+d(O_{12},M_{12})+d(Q_{12},M_{12})\leqslant 2d \tag{10}$$

注意到 $d(O_1,Q_1)=x$，$d(O_{11},Q_{11})=y$，于是：

$$d(O_{12},M_{12})+d(Q_{12},M_{12})\geqslant d(O_{12},Q_{12})=x-y \tag{11}$$

设 $d(O_{11}\cup Q_{11},M_{11})=m$，这就是说，在 m 个位置上，子片段O_{11}和Q_{11}中的碱基都与M_{11}中的不同，而另外 $y-m$ 个位置上，则有一个子片段的碱基与M_{11}的不同（因为O_{11}和Q_{11}中的每个碱基都不相同）。因此：

$$d(O_{11},M_{11})+d(Q_{11},M_{11})\geqslant 2m+y-m=y+m \tag{12}$$

将（11）和（12）代入（10），得：

$$m+x\leqslant 2d \tag{13-1}$$

即

$$d(O_{11}\cup Q_{11},M_{11})\leqslant 2d-x \tag{13-2}$$

由于 $d(O_{11}\cup Q_{11},R_{11})=y$ 以及 $d(M_{11},R_{11})\leqslant d$，从而有：

$$y=d(O_{11}\cup Q_{11},R_{11})\leqslant d(O_{11}\cup Q_{11},R_{11})+d(M_{11},R_{11})$$

$$\Rightarrow\quad y\leqslant 2d-x+d=3d-x \tag{14}$$

因此（6-3）成立。

从 $d(O_2,R_2)=z$，$d(O_{11}\cup Q_{11},R_{11})=y$，以及公式（7）和公式（13），可知：

$$\begin{aligned}y+z&=d(O_{11}\cup Q_{11},R_{11})+d(O_2,R_2)\\&\leqslant d(O_{11}\cup Q_{11},M_{11})+d(R_{11},M_{11})+d(O_2,M_2)+d(R_2,M_2)\\&\leqslant d(O_{11}\cup Q_{11},M_{11})+d(O_2,M_2)+d(R,M)\\&\leqslant (2d-x)+\left(d-\left\lfloor\frac{x}{2}\right\rfloor\right)+d\\&=4d-x-\left\lfloor\frac{x}{2}\right\rfloor\end{aligned} \tag{15}$$

因此，不等式（6-4）成立。

证毕。

不等式组（6）就是针对三个子序列的判据（判据 2）。它是一个必要条件而非充分条件，因此，某个模式(l,d)的有效组中的任意三个子序列都将满足这一判据。但是，如果一个组中的任意三个子序列都满足判据 2，它们却不一定属于同一个(l,d)模式，这里有一个假阳性错误概率。寻找针对四个或者更多子序列的判据是可能的，它们将具有

更小的误判概率，然而，寻找这样的判据是困难的，而且其相应的算法也会更加复杂。另一方面，我们对 CRISA 的分析和测试表明，利用判据 2 实现的算法是足够有效的。

3 算法实现

我们的目标是要从 n 个长度为 N 的样本中寻找所有的(l,d)–n 模式。采用下面的方法，我们设计了 CRISA 算法，它是一个深度优先的搜索算法，采用判据 2 作为其剪枝规则。

首先，我们把样本中所有的子序列列成 n 行，来自于同一个样本的子序列列在同一行上。然后，为相邻两层中的任意两个子序列建立一个连接，从而构成一个层次图，其中，子序列是图的节点，这些子序列之间的连接就是图的边。在这样的层次图中，一个子序列组（有效的或无效的）就表现为一条从第一层到最后一层的一条通路，我们的任务是寻找所有的有效组（它们也被称为有效路径）。一个简单的示意图如图 1 所示。

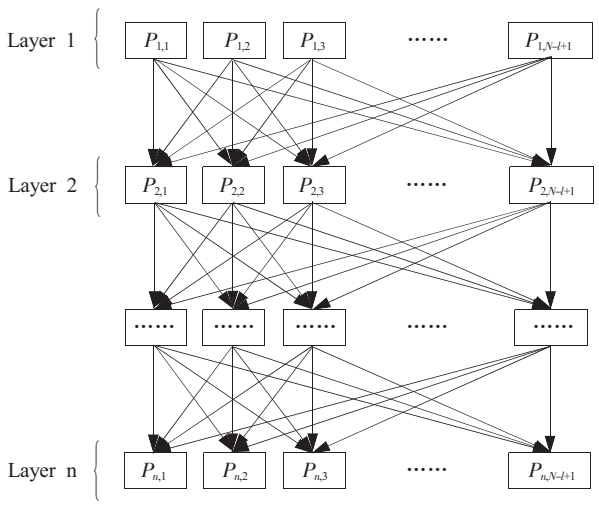

Fig.1 The layer graph for motif finding

图中的方框表示子序列（节点），P_{ij} 表示第 i 个样本中的第 j 个子序列，同一层中的子序列来自于同一个样本。从第一层到第 n 层的路径表示一个子序列组（有效的或无效的），路径上的每一个节点对应于相应样本中的子序列。

在这个层次图中的每一层上都有$(N–l+1)^n$ 个节点，因而图中共有$(N–l+1)^n$ 条路径。 如果简单地对所有的路径进行检验以找出其中的有效路径，其计算复杂度将是 $O(N^n)$，这是不能实际应用的。因此，我们提出了一种深度优先的搜索算法——判据搜索算法 CRISA，它使用判据 2 作为剪枝规则，对通常的模式发现问题，该算法的复杂度为 N 和 n 的多项式函数（见后文）。完整的 CRISA 算法如下：

1）算法开始

2）外循环：对第一个样本中的每个子序列 P_{1j}（其长度为 l），执行：

（1）、初始化部分路径为 $L=(P_{1j})$，并置 $k_i=0$，$1\leqslant i\leqslant n$

（2）、置 $i=2$

（3）、内循环：执行下面的操作，直到 $i=1$ 为止：

① 如果 $k_i>N–l+1$，向上回溯一层

② 否则：

a、对第 i 层中的子序列 P_{ik_i}和 L 中每两个子序列，都使用判据 6 进行一次检查

b、如果 P_{ik_i}通过了所有的检查，把它加入到部分路径 L 中，并检查下一层中的节点（如果 $i<n$）或者输出路径 L 并回溯一层（如果 $i=n$）；否则：置 $k_i=k_i+1$

（4）、内循环结束

3）外循环结束

4）算法结束

在进行算法测试的时候，我们发现得到的输出路径往往是一组路径而不是一条最优路径，而且这些路径之间只有个别的节点存在差别。出现这一现象的原因就是我们前面提到的算法中存在的误判概率，即某些节点偶然地通过了判据验证，从而使得一条非最优的路径得到了扩展的机会。我们通过一个简单的后处理来解决这一问题，即首先记录所有的输出路径 L，当内层循环结束以后，为这些路径上的所有节点（序列片段）建立位置权重矩阵，由此得到其中的保守序列。

在前面的算法讨论中，我们假定了条件 $k=n$。当此假设不成立的时候，算法需要在两个地方进行修改。首先，此时我们不能确定参考序列中一定含有一个实例，因此，我们应该至少进行 $n–k+1$ 次运算，每次选择一个不同的样本序列作为参考序列，由此可以保证至少在一次运算中找到正确的模式。其次，在这种情况下，我们所找到的路径中可以只包含 k 个节点，也就是说，允许路径中在 $n–k$ 个层上不包含合法的节点。因此，我们在算法中引入了空节点，即在每一层节点中，额外添加一个特

殊的节点，对包含空节点的部分路径，如果它满足：

1）它的由其它节点（正常节点）构成的部分路径是有效的；

2）该部分路径中的空节点数目不超过 $n-k$。

那么此包含空节点的部分路径就是有效的。这样，在算法的第 12 行，我们就能够采用这个标准来判断部分路径是否有效，从而使得算法能够找到这种包含空节点的有效路径。

4　复杂度分析

为了计算算法的复杂度，我们定义通过概率 p 为任意三个子序列满足判据 2 的概率。如果这三个子序列是独立的，并且如果序列中四种碱基的出现是等概率的，那么，通过概率可以计算如下：

$$p=\sum_{x=0}^{2d} C_l^x\left(\frac{3}{4}\right)^x\left(\frac{1}{4}\right)^{l-x}\cdot\sum_{y,z} C_x^y\left(\frac{2}{4}\right)^x\cdot C_{l-x}^z\left(\frac{3}{4}\right)^z\left(\frac{1}{4}\right)^{l-x-z} \tag{16}$$

这里 x、y 和 z 的定义与判据 2 相同，并且满足判据 2。如前所述，判据 2 应该被使用三次，每一次选择一条不同的子序列作为其中的第三条子序列，这三次使用的效果可以综合成一个新的判据 2'，由此能够计算出相应的新的通过概率 p'。判据 2' 具有与判据 6-2 相似的形式和推导方法，这里略去了它们的表达形式，而只在表 1 中列出了 p' 的常用数值。

Table 1　Pass probabilities of some normal motifs

l	d	p	p'	p'/p_{2d}
11	2	1.1e-5	6.0e-6	8.0e-4
12	3	7.2e-4	3.7e-4	6.8e-3
13	3	1.1e-4	5.6e-5	2.3e-3
14	4	3.3e-3	1.6e-3	0.015
15	4	6.7e-4	3.1e-4	5.4e-3
16	4	1.2e-4	5.2e-5	1.9e-3
16	5	0.01	5.1e-3	0.027
18	5	5.7e-4	2.4e-4	4.2e-3
18	6	0.024	0.012	0.043
28	8	6.7e-5	2.0e-5	6.8e-4
30	9	1.3e-4	4.4e-5	9.3e-4

在 CRISA 中，任意的子序列 $P_{ij}(i\geqslant 3)$都会被检查 C_{i-1}^2 次，每一次检查针对部分路径中的两个节点（记为 X 和 Y）和 P_{ij} 进行。由于部分路径中的节点是不独立的（除了 i=3 的情况），这里的通过概率应该是一个条件概率，其条件是 X 和 Y 在同一个部分路径中（这个概率近似为 p_{2d}）。因此，当 $i>3$ 时，CRISA 中的通过概率可以近似为 p'/p_{2d}。这里：

$$p_d=\sum_{i=0}^{d} C_l^i\left(\frac{3}{4}\right)^i\left(\frac{1}{4}\right)^{l-i} \tag{17}$$

它是两个子序列之间的距离不超过 d 的概率。

表中，l 是子序列的长度，d 是最大不匹配数目，p 和 p' 分别是判据 2 和判据 2' 的通过概率，p_{2d} 是两个子序列之间的距离不大于 $2d$ 的概率。

我们用 W_{ij}=1 表示从第一层中的节点 P_{1j} 出发到达第 i 层的部分路径的数目，显然有：

$$W_{1j}=1 \tag{18-1}$$

$$W_{2j}=N-l+1 \tag{18-2}$$

对后面的层次，如果没有剪枝规则，那么将会有 $W_{i-1,j}\cdot(N-l+1)$条部分路径到达层次 i。由于在 CRISA 中，剪枝规则是判据 2'，因此，在 W_{ij} 的表达式中将会增加一个概率因子$(p'/p_{2d})^{\alpha_i}$。这里，我们使用参数 α_i 作为一个近似的指数，如果对 C_{i-1}^2 次独立的检查，这个指数就是 C_{i-1}^2，而对于完全不独立的检查（所有的检查是完全一样的），这个指数将是 1（因为至少有一次检查），故 $1\leqslant\alpha_i\leqslant C_{i-1}^2$。由此，有：

$$W_{ij}\approx W_{i-1,j}\cdot(N-l+1)\cdot(p'/p_{2d})^{\alpha_i}$$

$$\approx(N-l+1)^{i-1}\cdot(p'/p_{2d})^{\sum_{k=3}^{i}\alpha_k},\quad 3\leqslant i\leqslant n \tag{18-3}$$

这样，CRISA 的算法复杂度可以计算为：

$$comp=\sum_{j=1}^{N-l+1}\sum_{i=3}^{n} W_{ij}\cdot C_{i-1}^2\cdot T \tag{19}$$

其中，T 是采用判据 2（判据 2'）进行一次检查所需要的时间，对固定的参数 l 和 d，它是一个常数。将（18）代入（19），有：

$$comp\approx(N-l+1)T\cdot\sum_{i=2}^{n}(N-l+1)^{i-1}(p'/p_{2d})^{\sum\alpha_i}C_{i-1}^2 \tag{20}$$

这是一个非常复杂的表达式，为了得到一个简

洁的计算复杂度上界，我们把所有的参数 α_i 都设为 1（因为 α_i 是一个概率的指数），忽略掉 $N-l+1$ 中的 $l-1$ 项，并使用 i^2 代替 C_{i-1}^2，得到：

$$comp \approx N^2T\cdot\sum_{i=2}^{n}(Np'/p_{2d})^{i-2}i^2$$

$$<N^2n^2T\cdot\sum_{i=2}^{n}(Np'/p_{2d})^{i-2} \quad (21)$$

因此，如果 $Np'/p_{2d}<1$，（21）式中的求和项将是一个常数，此时的算法复杂度将是 $O(N^2n^2T)$，这是 n 的多项式函数，而如果 $Np'/p_{2d}>1$，那么算法复杂度的上限将是 $O((Np'/p_{2d})^{n-2}N^2n^2T)$，这是 n 的指数函数。

上述结论是在把所有的参数 α_i 都设为 1 以后得到的，但是实际上，由于 α_i 是 C_{i-1}^2 次检查的通过概率的指数，而且这些检查并不是完全不独立的，因此，α_i 的实际值要大于 1，并且随着 i 的增大而增大。这样，这个计算复杂度还可以降低。以极端情况为例，即如果所有的检查都是独立的，那么 $\alpha_i=C_{i-1}^2$，因此，（20）可以写成：

$$comp \approx N^2T\cdot\sum_{i=2}^{n}\left(N(p'/p_{2d})^{\frac{i(i-1)}{6}}\right)^{i-2}i^2 \quad (22)$$

显然，只要 $p'/p_{2d}<1$，就会存在某个 i_0，使得对于所有的 $i>i_0$ 有：

$$N(p'/p_{2d})^{\frac{i(i-1)}{6}}<1 \quad (23)$$

此时，（22）式中的求和项将收敛到一个常数，算法复杂度是 $O(N^2n^2T)$，这是一个 n 的多项式函数。但是如果 $p'/p_{2d}>1$，那么算法复杂度的上限将是 $O\left((p'/p_{2d})^{\frac{(n-1)(n-2)}{2}}N^{n+1}n^2T\right)$，这是 n 的指数函数。

判据 $Np'/p_{2d}<1$ 比判据 $p'/p_{2d}<1$ 的要求严格得多。在所有情况下，只要 $Np'/p_{2d}<1$，我们就可以断言，算法的复杂度是序列数 n 的多项式函数。从表 1 中可以看到，许多通常的模式都是满足条件 $Np'/p_{2d}<1$ 的。而且我们刚才的分析表明，对表 1 中那些不满足条件 $Np'/p_{2d}<1$ 的模式，由于参数 α_i 的存在，算法的复杂度仍然可能是序列数 n 的多项式函数（在所有检查都是在独立的情况下，只要 $p'/p_{2d}<1$，就可以使得算法的复杂度是序列数 n 的多项式函数）。这就是为什么 CRISA 适用于绝大多数模式发现问题的原因。

至于算法的空间复杂度，由于我们采用的是深度优先的搜索算法，所需要的内存相当少：在构造层次图的时候，需要记录每个节点在输入序列中的位置；在搜索的过程中，需要记录当前的局部路径以及与之相关的判据，并为每个层次保留一个搜索指针；当搜索到一条合适的路径时，为它构造一个位置权重矩阵，并记录其一致序列作为输出。这些内存开销都很小，它们不超过输入序列的线性函数。实际上，CRISA 算法的主要内存开销在于保存输入的样本序列：我们对算法的测试表明，对标准测试问题，算法的内存约为 20 K，其中输入数据就占了约 12 K。

5 算法测试

我们采用模拟数据和真实的生物序列数据分别对算法进行了测试。模拟数据的产生方法如下：首先随机产生一组（n 个）样本序列，每个样本序列长度为 N，并产生一个长度为 l 的随机序列片段，以之作为待发现的模式。之后，随机选择 k 个样本序列，在每个样本序列中，随机选择一个位置 x，$1\leqslant x\leqslant N-l$, 用模式的一个实例替换从 x 开始的 l 个碱基。这个实例也是随机产生的，其产生方式是从模式中随机选择 $y(y\leqslant d)$个碱基，并用不同的碱基加以替代。在所有的测试中，我们使用的参数为 n=20，N=600，这也是标准测试问题的参数。表 2 中给出了对 CRISA 的测试结果，作为对比，表中还列出了 MITRA 算法对同样难度的问题的测试结果[8]。

对模式识别问题，算法的计算时间不仅与算法本身相关，还与识别的模式相关。Keich 和 Pevzner[7]以及 Buhler[5]都对此作出了分析，其分析基本相同。这里我们采用 Buhler 给出的近似公式，为：

$$E(l,d)=4^l(1-(1-p_d)^{N-l+1})^n \quad (24)$$

$$p_d=\sum_{i=0}^{d}C_l^i\left(\frac{3}{4}\right)^i\left(\frac{1}{4}\right)^{l-i} \quad (25)$$

其中 $E(l,d)$是在随机选择的 k 个长度为 n 的序列中，存在的(l,d)模式的期望数目，此模式在每个序列中至少出现一次。显然，$E(l,d)$越大，表示随机模式越多，因而真实的模式就越弱。如果这个数字接近或者超过 1，模式将不可识别。因此，它可以作为对模式的一个近似衡量指标，对表 2 中的各个测试模式，我们给出了它们的 $E(l,d)$值。另一方

面，如前所述，通过概率影响着 CRISA 的计算复杂度，我们把它也列在了表中。

需要说明的一点是，由于我们无法对 MITRA 和 CRISA 采用相同的数据进行测试，因而表中的测试结果只是对具有相同参数的随机样本序列集合的测试结果，其中，MITRA 的结果来自于原始论文[8]。因此，表中的数值只应该看作是一个近似的比较。

Table 2　The performance of MITRA and CRISA

No.	(l,d)–k	$E(l,d)$	TIME/MEM			Pass Prob.
			MITRA-C	MITRA-G	CRISA	
1	(11,2)-20	5.4e-17	60/5	60/5	2/0.02	8.0e-4
2	(12,3)-20	3.2e-7	60/5	240/100	6/0.02	6.8e-3
3	(13,3)-20	8.1e-16	120/5	120/40	2/0.02	2.3e-3
4	(14,4)-20	4.2e-7	240/5	600/210	100/0.02	1.5e-2
5	(15,4)-20	2.2e-15	300/5	300/100	5/0.02	5.4e-3
6	(16,5)-20	2.3e-7	1500/5	1200/400	1750/0.02	2.7e-2
7	(18,6)-20	7.1e-8	15000/5	2400/650	14600/0.02	4.3e-2
8	(28,8)-20	1.6e-58	-/-	240/50	3/0.02	6.8e-4
9	(30,9)-20	1.6e-58	-/-	300/90	4/0.02	9.3e-4

Note: for all tests listed in the table, n = 20 and N = 600. E(l,d) is the expected number of random motifs appearing at least once in each sequence. TIME/MEM is the compute time (in seconds) and maximum memory occupied (in megabytes). MITRA-C and MITRA-G are the two realization of MITRA with a little difference. Pass Prob. is the pass probability. Blank entries '-/-' means that the algorithm failed to solve the problem in reasonable time period. MITRA is performed on machine with a Pentium Ⅲ 750 MHz processor and 1 GB of RAM, and CRISA is performed on machine with a Pentium Ⅲ 733 MHz processor and 256 M of RAM, whose compute time is averaged by 10 runs.

使用 $E(l,d)$为标准，我们能够把表 2 中的模式分为 3 类：2、4、6 和 7 是非常微弱的模式，1、3、5 是较弱的模式，而 8 和 9 是较强的模式。但是这种分类与两种算法的结果都不匹配。按照通过概率，我们能够得到一个更加合理的分类：4、6 和 7 是非常微弱的模式，2、3 和 5 是较弱的模式，而 1、8 和 9 则是较强的模式。对这两种算法，模式越强，算法所需的计算时间越少。

表 2 中的结果表明，对后两类模式，CRISA 远远快于 MITRA 算法，对第一类模式，CRISA 在一个模式上快于 MITRA，而在另两个模式上慢于 MITRA。另一方面，对表中所有的模式，CRISA 使用的内存数仅仅为几十千字节，这一数值要远远小于 MITRA 算法的几兆甚至几十兆字节。这表明，CRISA 能够在绝大多数的模式发现问题中工作得很好。

对标准测试问题，我们比较了几种最新算法的执行效果，结果在表 3 中。从表中可以看到，CRISA 与 MITRA 是基于穷举法的完全搜索，它们能够保证找到样本中存在的模式，而其它三种算法都会出现丢失模式的可能。但是 MITRA 的计算时间最长，而 CRISA 的速度仅次于 PatternBranching，快于另外三种算法。这里我们必须指出，由于我们不能采用相同的测试数据对这些算法进行测试，表中的结果都是从原始论文中获得的，因而这种比较的结果只能大致地反应算法的执行效果之间的关系。

Table 3　The performance of CRISA and some other algorithms

Algorithm	Exhaust search	Running time (s)	CPU (MHz)	Reference
PROJECTION	No	120	667	[5]
MULTIPROFILE	No	60	1000	[9]
MITRA	Yes	300	750	[8]
PatternBranching	No	3	1000	[9]
CRISA	Yes	5	733	—

最后，我们选择了一些包含已知模式的实际生物序列对 CRISA 进行了测试，这些样本集合包括：

一组包含经过试验验证的 *E. coli* 的 CRP 绑定点信号的序列，一共 18 条，每条的长度为 105 个碱基[15]。

一组包含酵母的转录调控子 PDR3 的序列，一共有 7 条，每条序列的长度为 500 碱基[16]。

一组包含酵母 DNA 绑定位点 USR1 信号的序列，共 11 条，每条长 550 个碱基[17]。

其中的后两组数据来自于 SCPD 数据库[18]。这里，由于 USR1 信号可以在 DNA 序列的互补链上出现，因此，我们把原始序列与它的逆向互补链相连，再进行计算，以此来寻找这些互补链上的模式。

Table 4 Result of CRISA on *E.Coli* CRP sample

Sample	*N*,*n*	Reference motif	(*l*,*d*)−*k*	CRISA motif
CRP	105, 18	TGTGANNNNGNTCACA	(20,7)-16	TTTGTGANNNAGTTCACATT TTGTGANNNAGTTCACATTT TGTGANNNAGTTCACATTTT
PDR3	500, 7	TCCGYGGA	(10,1)-7	TTCCGCGGAA
USR1	550, 11	T(A/G/T)GCCGCCTA	(13,2)-11	TNGGCGGCTAAAT

表中，CRISA motif 中的 N 表示在该位置上没有一种碱基能够在超过一半的实例中出现，加粗的字符表示与参考模式相符的字符。从表中可以看到，我们的 CRISA 算法在 CRP 样本组中发现了三个模式，它们实际上是同一个模式的不同位移，并且都包含了参考模式；对 PDR3 样本组，CRISA 发现的模式也与参考模式符合得相当好；只是对 USR1 样本组，CRISA 发现的模式与参考模式之间存在明显的偏差。我们检查了算法的结果，发现原始文献[17]所给出的参考模式并没有考虑逆向链上的模式，而从 SCPD 中获得的原始数据则表明，一些信号被明确地加上了 c 标志（complement），因此我们认为这些信号应该是能够出现在逆向链上的，而且，在测试中，CRISA 算法所找到的实例的位置与 SCPD 中所标出的信号位置完全一致，因此，我们认为 CRISA 算法所发现的模式是可信的。

6 讨 论

本文中，我们提出了一种描述三个序列片段相似性关系的判据，使用这一判据作为剪枝规则，我们实现了一种新的深度优先搜索算法——判据搜索算法（criterion search algorithm, CRISA）来寻找一组 DNA 序列中的模式。我们对 CRISA 进行了深入的分析，并采用仿真数据对算法进行了测试，测试的结果表明，对绝大多数模式发现问题，CRISA 算法都能够迅速而准确地找出样本集合中所有的模式，这一测试结果表明了算法的有效性。之后，我们采用实际生物数据对算法也进行了测试，结果表明算法能够准确地找到序列中存在的模式，这说明算法也是实用的。

CRISA 算法存在的一个问题是，由于我们对判据 2 的推导是基于 Hamming 距离进行的，因而不能用于寻找具有插入 / 删除错误的模式。这一问题也或多或少地存在于许多其它的算法中。由于大部分的生物模式并不含有插入 / 删除错误，因此这一问题并不会严重影响算法的应用。另一方面，我们同样可以针对具有插入 / 删除错误的模式发现问题，推导出与判据 2 相类似的判据。这是一个值得研究的问题。

算法的另一个问题是 CRISA 的扩展问题，即将它推广到组合模式的发现问题。一个组合模式是指两个（或者更多）个普通的模式在样本中同步出现，彼此之间相隔一个不确定的长度，这样的模式发现问题需要进行更多的计算。由于 CRISA 算法能够迅速而准确地求解绝大部分的模式发现问题，因而它很适合于进行这样的推广。这将是我们下一步的工作。

参考文献:

[1] Pevzner PA, Sze S. Combinatorial approaches to finding subtle signals in DNA sequences. In: Philip EB, Michacl G, Russ BA, Nancy J, Debra H, Thomas L, Julie CM, Eric DS, Chris S, Shawn S, Helge W. Proceeding of the 8th International Conference on Intelligent Systems for Molecular

Biology. San Diego: AAAI Press, 2000. 269~278

[2] Hertz G, Stormo G. Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics*, 1999,15(7):563~577

[3] Bailey T, Elkan C. Unsupervised learning of multiple motifs in biopolymers using expectation maximization. *Machine Learning*, 1995,21(1-2):51~80

[4] Lawrence C, Altschul S, Boguski M, Liu J, Neuwald A, Wootton J. Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science*, 1993,262 (5131):208~214

[5] Buhler J, Tompa M. Finding motifs using random projections. *J Comput Biol*, 2002,9(2):225~242

[6] Keich U, Pevzner PA. Finding motifs in the twilight zone. *Bioinformatics*, 2002,18(10):1374~1381

[7] Keich U, Pevzner PA. Subtle motifs: defining the limits of motif finding algoritms. *Bioinformatics*, 2002,18 (10): 1382~1390

[8] Eskin E, Pevzner PA. Finding composite regulatory patterns in DNA sequences. *Bioinformatics*, 2002,18(1):354~363

[9] Price A, Ramabhadran S, Pevzner PA. Finding subtle motifs by branching from sample strings. *Bioinformatics*, 2003,19 (s2):ii149~ii155

[10] Brazma A, Jonassen I, Eidhammer I, Gilbert D. Approaches to the automatic discovery of patterns in biosequences. *J Comput Biol*, 1998,5(2):279~305

[11] Tompa M. An exact method for finding short motifs in sequences with application to the Ribosome Binding Site problem. In: Thomas L, Reinhard S, Peer B, Douglas LB, Janice IG, Hans WM, Ralf Z. Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology. Heidelberg, Germany: AAAI Press, 1999. 262~271

[12] Waterman MS, Arratia R, Galas DJ. Pattern recognition in several sequences: consensus and alignm ent. *Bull Math Biol*, 1984,46(4): 515~527

[13] Sagot M. Spelling approximate or repeated motifs using a suffix tree. *LNCS*, 1998,1380,111~127

[14] Gelfand M, Koonin E, Mironov A. Prediction of transcription regulatory sites in Archaea by a comparative genomic approach. *Nucleic Acids Res*, 2000,28(3):695~705

[15] Stormo GD, Hartzell III GW. Identifying protein-binding sites from unaligned DNA fragments. *Proc Natl Acad Sci USA*, 1989,86(4):1183~1187

[16] Wolfger H, Mahé Y, McDermott AP, Delahodde A, Kuchler K. The yeast ATP binding cassette (ABC) protein genes PDR10 and PDR15 are novel targets for the Pdr1 and Pdr3 transcriptional regulators. *FEBS Letters*, 1997,418(3):269~274

[17] GuhaThakurta D, Stormo GD. Identifying target sites for cooperatively binding factors. *Bioinformatics*, 2001,17 (7): 608~621

[18] Zhu J, Zhang MQ. SCPD: A promoter database of yeast Saccharomyces cerevisiae. *Bioinformatics*, 1999,15(7):607~611

A FAST MOTIF FINDING ALGORITHM FOR DNA SEQUENCE

LI Dong-dong, WANG Zheng-zhi, DU Yao-hua, YAN Chun

(*Department of Automatic Control, National University of Defense Technology, Changsha 410073, China*)

Abstract: Motif finding is an important research field in bioinformatics. Many algorithms on motif finding have been developed at present, but among these algorithms only few can find the correct motif surely, such as MITRA. In this paper, a new exhaust search algorithm named CRISA (criterion search algorithm) is proved. It can accomplish the exhaust search with less computation resource. This target is achieved based on the criterion describing the relations between three similar segments deduced in this paper. Using this criterion as pruning rule, CRISA can reduce the search space effectively in the deeply first search process. Theoretical analysis on CRISA is done in this paper, and the results show that under some rather loose conditions, the computational complexity of CRISA is a polynomial function of the length and the number of the input sequences. Then, some tests using simulated and biological data have been done and the results show that it is more efficient than other exhaust search algorithms obviously, and its search speed is even faster than that of many non-exhaust search algorithms.

Key Words: Motif finding; Criterion; Depth first search