最优搜索机制下寻找最优插入–删除种子

陈 科[1]，朱清新[1]，杨 曦[2]

(1. 电子科技大学计算机科学与工程学院 成都 610054; 2. 成都华西中学 成都 610051)

【摘要】空位种子极大地提高了生物分子序列比对的灵敏度，但不适合大量存在插入和删除字符的序列。在空位种子的基础上，提出了带插入–删除的生物序列比对种子，进一步提高了生物序列比对的效率。实验表明，采用最优搜索算法可以有效地在给定约束条件下寻找到最优的插入-删除种子，并且插入-删除种子比同长度的最优空位种子具有更高的生物序列比对敏感度。

关 键 词 插入–删除种子; 生物分子序列比对; 最优系统; 空位种子

中图分类号 TP301 **文献标识码** A doi:10.3969/j.issn.1001-0548.2011.02.027

Finding Optimal Indel Seeds under Optimal Search Mechanism

CHEN Ke[1], ZHU Qing-xin[1], and YANG Xi[2]

(1. School of Computer Science and Engineering, University of Electronic Science and Technology of China Chengdu 610054;
2. Chengdu Huaxi Middle School Chengdu 610051)

Abstract Spaced seeds greatly improve the sensitivity of bio-sequences alignment, but they are not applicable for the sequences containing many insertions and deletions. Based on spaced seeds, we propose seeds with insertions and deletions (Indel seeds), which improves the sequences alignment sensitivity more. Experiments show that adopting optimal search algorithm can find optimal indel seeds under given constraints, further more, indel seeds have higher bio-sequences alignment sensitivity than that of spaced seeds.

Key words indel seed; molecular alignment; optimal systems; spased seed

生物分子序列比对即将两种不同生物种类的生物分子序列(DNA序列或蛋白质序列)进行相似性比对，以研究两种生物的同源性。生物序列比对算法研究是生物信息学最重要的课题之一。在众多生物序列比对算法中，最为著名的是采用启发式算法的BLAST[1-2]系列及其衍生算法。这些算法首先在局部寻找较短的匹配(通常是全配)，然后依据相似度指标进行进一步延伸，从而在给定的约束下得到较长的匹配序列。启发式算法极大地提高了生物序列的匹配效率，在生物序列比对研究领域产生了极为深远的影响。

从BLAST算法过程可以看出，生物序列比对过程中一个极为重要的因素就是寻找局部较短匹配，较短匹配称为序列比对的“种子”。种子是影响序列比对速度和灵敏度的调节阀：较短的种子具有更高的匹配概率，即更高的灵敏度，但由于较多的局部全配的出现需要延伸，直接导致匹配速度的下降；反之，较长的种子灵敏度较低，但具有较高的匹配速度。因此，生物序列比对中匹配速度和灵敏度是一对矛盾，寻找好的种子是生物信息学中的一个重要问题。

默认的BLAST种子可表示为：11111111111，其中1表示匹配，即BLAST要求局部有11个连续匹配的字母，该类种子被称为“连续种子”。由于不允许出现空位，其匹配速度与灵敏度的矛盾最为突出。一个突破来自于PatternHunter[3-4]算法。在PatternHunter算法中，首次提出了“带空位种子”(spaced seeds)的概念，即种子内部在给定位置上允许出现失配。PatternHunter 的默认种子可表示为111*1**1*1**11*111，其中1表示该位置上必须匹配，而*表示该位置上允许失配，即空位。

空位种子的出现在一定程度上缓解了比对过程中速度与灵敏度之间的矛盾，文献[5-7]讨论了良好的或最优的空位种子的设计方法，然而这些方法所

收稿日期：2009 – 06 – 25; 修回日期：2009 – 11 – 02
基金项目：国家自然科学基金(60671033)
作者简介：陈 科(1981 –)，男，博士生，主要从事计算机应用方面的研究.

针对的空位种子均假设种子内部只能出现匹配或失配两种情况，而没有生物序列中常常出现的插入或删除情况，不符合大多数(>95%)生物序列的匹配规律，如大量存在的启动子或非编码区序列。为此，文献[8]提出了带插入−删除的种子，即允许在种子内部引入插入和删除，从而进一步提高序列比对的速度和灵敏度。然而文献[8]并没有解决如何找到最优插入−删除种子这一重要问题。

由于在给定种子长度参数情况下，候选种子个数和灵敏度计算过程均为指数复杂度，因此逐一计算的方法是不可取的。为了解决该问题，采用最优搜索[9-10]机制，从而将问题转化为：在给定资源(通常是搜索时间)限定的条件下，试图以最大的概率寻找问题的最优解。文献[11]将最优搜索机制应用于寻找分组种子(grouped seeds)，取得了一定的进展，将最优搜索应用插入−删除种子。实验证明，插入−删除种子有效地提高了生物序列比对的灵敏度，且最优搜索算法可以有效地找到最优的插入−删除种子。

1　插入−删除种子模型

生物序列比对过程中可能产生失配、匹配、插入或删除4种情况，分别用符号0、1、2和3表示，下面DNA序列比对的例子包含了这4种情况：

查询串：A - C T A T G C - A G T

目标串：A G C - T C - C A G G A

匹配结果表示：1 2 1 3 0 0 3 1 2 0 1 0

值得注意是，匹配过程中不允许出现“32”或“23”，原因在于序列比对的评分体系中，对紧接的插入−删除的罚分要比单独一次失配要严重得多。为了找到得分最高的最佳匹配结果，引入“插入−删除种子”，定义如下。

定义 1　(插入−删除种子) 以1开头，1结尾，中间包含1或*的生物序列，其中1表示匹配，*表示无字符或字符匹配、失配、插入或删除。插入−删除种子 S 可表示为：

$$S = 1^{k_1} *^{i_1} 1^{k_2} *^{i_2} \cdots 1^{k_{m-1}} *^{i_n} 1^{k_m}$$

式中，$1 \in \{1\}$ 且 $* \in \{0,1,2,3\}$；$k_{j_1} > 0, 1 \leqslant j_1 \leqslant m$ 且 $i_{j_2} > 0, 1 \leqslant j_2 \leqslant n$，$1^{k_l}$ 或 $*^{i_l}$ 表示连续 k_l 个1或 i_l 个*，m 和n分别表示连续的1块和*块的个数。S 的长度称为种子长度(Length)，1的个数称为种子权重(Weight)。从定义可以推出：

$$\text{Length}(S) = |S| = |1| + |*| = \sum_{l=1}^{m} k_l + \sum_{l=1}^{n} i_l$$

$$\text{Weight}(S) = |1| = \sum_{l=1}^{m} k_l$$

如种子1*1可表示5种比对结果，即11、1$\underline{0}$1、1$\underline{1}$1、1$\underline{2}$1和1$\underline{3}$1；$\text{Length}(S) = 3$；$\text{Weight}(S) = 2$。

一个插入−删除种子可以产生一系列序列匹配模式(Pattern)，如对于给定的长度为 n 的查询串 Q，在位置 $i \in [4, n]$ 上，由于*可表示1个或0个字符，种子1**1将产生以下匹配模式：

$$\boxed{Q[i-3]} - \boxed{Q[i]},\ \boxed{Q[i-2]} - \boxed{Q[i]},\ \boxed{Q[i-1]} - \boxed{Q[i]}$$

其中，−表示0个、1个或2个任意字符。如对于 $Q = \text{GCATGA}\underline{\text{C}}\cdots$，在下划线位置 $i = 7$ 处，将产生以下模式：

$$\boxed{\text{T}} - \boxed{\text{C}},\quad \boxed{\text{G}} - \boxed{\text{C}},\quad \boxed{\text{A}} - \boxed{\text{C}}$$

种子的命中即该种子的任何一个模式在查询串和目标串的位置 n 上或之前中匹配成功。种子的灵敏度可简单定义为种子的命中概率。文献[8]从数学上定义了插入−删除种子的灵敏度：

$$\text{Sensitivity}(S:n) = \sum_{j=1}^{n} \sum_{a \in T} P(\text{pattern}_a : j)$$

式中，T 为种子生成的模式集合；$P(\text{pattern}_a : j)$ 为模式 a 在位置 j 匹配成功的概率。

可以看出，插入−删除种子比空位种子和连续种子具有更多的模式，因此插入−删除种子比空位种子和连续种子具有更高的命中概率，即具有更高的灵敏度。

然而，在给定种子权值条件下，如何以最大概率找到最优的插入−删除种子，即具有最高灵敏度的种子，是一个非常棘手的问题。为了有效地解决该问题，下面将采用最优搜索机制，在有效资源约束条件下，寻找最优插入−删除种子。

2　最优搜索方法

最优搜索理论[10]用于解决在 N 个未知盒子中以最大概率寻找到目标的问题。对于给定长度和权值的种子，将所有候选种子视为盒子，其中具有最大灵敏度的种子视为目标，采用最优搜索方法寻找最优的插入−删除种子。

根据插入−删除种子的定义，可知候选种子的个数为：

$$N = \frac{(L-2)!}{(w-2)!(L-w)!}$$

式中，L 为种子长度；w 为种子权值。令第 i 个种子为最优种子的概率为 $p(i)$，显然有：

$$p(i) \geqslant 0,\ \sum p(i) = 1$$

定义探测函数[9]：

$$b(i,z)=1-\mathrm{e}^{-z} \qquad i=1,2,\cdots,N$$

运用拉格朗日乘数方法有：

$$l(i,\lambda,z)=p(i)b(i,z)-\lambda z=p(i)(1-\mathrm{e}^{-z})-\lambda z$$

令：

$$\frac{\mathrm{d}l(i,\lambda,z)}{\mathrm{d}z}=p(i)\mathrm{e}^{-z}-\lambda=0$$

得到最优搜索方案为：

$$z_i=f_\lambda^*(i)=\ln\frac{p(i)}{\lambda}$$

令搜索总成本开销限定为 K，则有：

$$\sum_{i=1}^{N}z_i=\sum_{i=1}^{N}\ln\frac{p(i)}{\lambda}\leqslant K$$

因此：

$$\lambda\geqslant\left[\prod_{i=1}^{N}p(i)\right]^{\frac{1}{N}}\mathrm{e}^{-\frac{K}{N}}$$

可得最优探测概率：

$$p(f^*)=\sum_{i=1}^{N}p(i)\,\mathrm{b}(i,f_\lambda^*(i))=\sum_{i=1}^{N}p(i)(1-\mathrm{e}^{-z_i})=$$

$$\sum_{i=1}^{N}p(i)(1-\mathrm{e}^{-\ln\frac{p(i)}{\lambda}})=1-N\lambda$$

当种子长度和权值确定后，可得到候选种子数 N，如当 $L=5$，$w=3$ 时，候选种子有3个，分别为1*1*1、1**11和11**1。对这 N 个候选种子标记下标为1–N，最优探测概率揭示了如何在有限资源限制下产生候选种子下标，这些下标对应的种子中最大的概率包含最优种子。计算最优探测概率的一个重要参数是确定 $p(i)$，即第 i 个种子为最优种子的概率，为了简单起见，本文取 $p(i)=\dfrac{1}{N}$。以 $L=5$，$w=3$ 的种子为例，设资源限制 $K=2N$，取 $\lambda=\min(\lambda)=\dfrac{1}{N}\mathrm{e}^{-\frac{K}{N}}$，得：

$$p(f^*)=1-N\lambda=\mathrm{e}^{-2}$$

以概率 $p(f^*)$ 产生 $K=2N$ 个序列下标。本文中的实验结果为1、3、2、1、2、1。对于多次重复出现的下标，可认为该种子最优的可能性较大，应优先搜索。

通过最优搜索方法确定搜索种子下标并优先搜索重复出现次数较多的候选种子，在有限资源限制下避免了对所有候选种子的穷尽搜索，提高了搜索效率。

3 最优搜索方法寻找插入–删除种子

下面采用最优搜索方法分别考查种子长度从14～20，权值从9～15的插入–删除种子。表1～3分别显示了在不同搜索资源限制条件下的插入–删除种子的搜索结果。

表1 *K*=2*N*条件下插入–删除种子的搜索

种子权值	搜索结果	花费时间/s
9	1111*1**11**11	2.47
10	111*1*111*1*1*1	10.89
11	1111*1*1*1111**1	90.13
12	1111**11*1*111*11	342.12
13	111**111*111*111*1	961.22
14	111*11**1111111*1*1	1 892.93
15	1111*1*1*1*1111111*1	2 034.76

表2 *K*=5*N*条件下插入–删除种子的搜索

种子长度	搜索结果	花费时间/s
9	11*111*1*1**11	3.95
10	11*111*11*1**11	18.43
11	11*11*11*1111**1	136.63
12	111*1*11*1*111*11	527.16
13	111*11*1*1111*11*1	1 463.82
14	11*11*1*1111111*1*1	2 801.37
15	1*1111*1*1*1111*1111	3 216.28

表3 *K*=10*N*条件下插入–删除种子的搜索

种子长度	搜索结果	花费时间/s
9	1*1111*1*1*1*1	8.15
10	1*1111*1*11*1*1	31.34
11	1*1111*1*111*1*1	260.13
12	1*1111*1*1111*1*1	1 013.91
13	1*1111*1*11111*1*1	2 832.44
14	1*1111*1*111111*1*1	5 670.36
15	1*1111*1*1111111*1*1	6 172.56

作为对比，取表3的搜索结果分别在不同相似度指标下与同权值最优空位种子的灵敏度指标进行对比，其中最优空位种子数据来自于文献[5]。表4～6显示了对比结果。

表4 相似度为65%情况下种子灵敏度比较

种子权值	最优空位种子	空位种子灵敏度	插入–删除种子灵敏度
9	11*11*1* 1***111	0.522 2	0.689 4
10	11 *11*** 11*1* 111	0.380 9	0.597 8
11	111*1**1*1**11*111	0.267 2	0.523 7
12	111 *1* 11*1**11*111	0.183 8	0.469 5
13	111*11*11**1*1*1111	0.123 3	0.430 1
14	1111*1*1*11**11*1111	0.081 8	0.403 2
15	111*111*1*11*1**11111	0.053 4	0.384 7

表5　相似度为75%情况下种子灵敏度比较

种子权值	最优空位种子	空位种子灵敏度	插入–删除种子灵敏度
9	11*11*1* 1***111	0.889 5	0.928 2
10	11 *11*** 11*1* 111	0.801 1	0.870 7
11	111**1*11**1*1*111	0.695 9	0.802 4
12	111 *1* 11*1**11*111	0.587 1	0.731 6
13	111*11*11**1*1*1111	0.482 1	0.663 4
14	1111*1*1*11**11*1111	0.388 1	0.602 2
15	1111**1*1*1*11*1111	0.305 5	0.548 5

表6　相似度为85%情况下种子灵敏度比较

种子权值	最优空位种子	空位种子灵敏度	插入–删除种子灵敏度
9	11*11*1* 1***111	0.996 8	0.996 9
10	11 *11*** 11*1* 111	0.990 1	0.990 1
11	111**1*11**1*1*111	0.976 0	0.976 1
12	111 *1* 11*1**11*111	0.952 1	0.962 8
13	111*11*11**1*1*1111	0.917 4	0.942 5
14	1111*1*1*11**11*1111	0.872 2	0.903 3
15	1111**1*1*1*11*1111	0.816 1	0.854 1

4　结 束 语

通过实验数据可以看出，在相同权值和相似度指标下，插入-删除种子比空位种子具有更高的生物序列比对的灵敏度。通过最优搜索算法，可以在有限资源限制下，以最大概率寻找到插入–删除种子。从花费的时间数据上看，当权值增大时，花费时间增长很快，因此，如何优化算法，以更小的资源代价寻找到目标种子，是进一步研究的重要内容。

参 考 文 献

[1] ALTSCHUL S F, GISH W, MILLER W, et al. Basic local alignment search tool[J]. Mol Biol, 1990, 215: 403-410.

[2] ALTSCHUL S F, GISH W, MILLER W, et al. Gapped blast and psi-blast: a new generation of protein database search programs[J]. Nucleic Acids Research, 1997, 25: 3389-3402.

[3] MA B, TROMP J, LI M. PatternHunter: faster and more sensitive homology search[J]. Bioinformatics, 2002, 18(3): 440-445.

[4] LI M, MA B, KISMAN D, et al, PatternHunter II: Highly sensitive and fast homology search[J]. Bioinform Compute Biol, 2004, 2(3): 164-175.

[5] CHOI K P, ZENG F F, ZHANG L. Good spaced seeds for homology search[J]. Bioinformatics, 2004, 20(7): 1053-1059.

[6] CHOI K P, ZHANG L. Sensitivity analysis and efficient method for identifying optimal spaced seeds[J]. Journal of Computer and System Sciences, 2004, 68(1): 22-40.

[7] KEICH U, LI M, MA B, et al. On spaced seeds for similarity search[J]. Discrete Applied Mathematics. 2004, 138(3): 253-263.

[8] MAK D, GELFAND Y, BENSON G. Indel seeds for homology search[J]. Bioinformatics, 2006, 22(14): 341-349.

[9] 朱清新. 离散和连续空间中的最优搜索理论[M]. 北京: 科学出版社, 2005.
ZHU Qing-xin. Optimal search theory in discrete and continuous spaces[M]. Beijing: Science Press, 2005.

[10] ZHU Qing-xin, ZHOU Ming-tian, OOMMEN J. Some results on optimal search in discrete and continuous spaces[J]. Journal of Software, 2001,12: 1748-1751

[11] CHEN Ke, ZHU Qing-xin. Similarity search for sequence based on grouped seeds criterion with optimal search mechanism[C]//Bio Comp'08. [S.l.]:[s.n.], 2008: 844-847.

编　辑　将　晓