DNA 双序列比对问题的算法①

曹 莉, 许玉龙, 邓崇彬

(河南中医学院 信息技术学院, 郑州 450000)

摘 要: 随着生物序列数据库中序列数据的激增, 开发兼有高度生物敏感性和高效率的算法显得极为迫切. 通过对生物序列比对问题中 Needleman-Wunsch 算法和 Smith-Waterman 算法深入分析, 提出了 Smith-Waterman 算法的改进算法, 并通过实验验证该算法, 对改进前后的运行性能进行比较分析. 实验证明, 改进后的算法实现了双序列局部最优解个数的优化, 有效降低了生物序列比对算法时间与空间的复杂性, 提高序列比对的得分率和准确率.

关键词: 生物信息学; 双序列比对; 算法

DNA Pairwise Sequence

CAO Li, XU Yu-Long, DENG Cong-Bin

(School of Information Technoelgy, Henan University of Traditional Chinese Medicine, Zhengzhou 450000, China)

Abstract: With the surge in sequence data of biological sequence database, developing a algorithm which has the high biology sensitivity and efficiency is very urgent. Based on the deep analysis on the Needleman-Wunsch and Smith-Waterman Algorithm of bio-sequence alignment, the author enhances the Smith-Waterman algorithm as well as proves its accuracy through a series of experiments in this paper. Comparing between the Smith-Waterman algorithm and the improved one, the author analyzes the performance of the two algorithms. Experimental results show that the newly improved algorithm can optimize the number of local optimal solutions for pairwise sequence, reduce the complexity in time and space of bio-sequence alignment algorithms, and increase the scores and accuracy of sequence alignments.

Key words: bioinformatics; pairwise sequence alignment; algorithm

序列比较是生物信息学中最基本、最重要的操作, 通过序列比对可以发现生物序列中的功能、结构和进化的信息. 序列比较的根本任务是: 通过比较生物分子序列, 发现它们的相似性, 找出序列之间共同的区域, 同时辨别序列之间的差异. 研究序列相似性的另一个目的是通过序列的相似性, 判别序列之间的同源性, 推测序列之间的进化关系[1].

序列比对尤其是多序列比对问题已经被证明是 NP 完全问题, 即多项式复杂程序的非确定性问题. 所以如何设计出比对速度高同时敏感度好的序列比对算法, 是分子生物学中的一个亟待解决的难题[2,3].

1 序列比对算法的研究现状

按照比对的序列个数, 序列比对分为双序列比对(pair-wise sequence alignment)和多序列比对(multiple sequence alignment). 双序列比对是多序列比对的重要理论基础. 按照比对结果的精确度, 序列比对分为精确比对和近似比对. 按照比对的序列范围, 序列比对分为全局比对(global alignment)、局部比对(local alignment)和半全局比对. 最常见的比对是蛋白质序列

① 基金项目:河南省高等学校重点科研项目(15A520083);河南省 2014 年基础与前沿技术研究计划(142300410428,142300410391);河南中医学院苗圃工程项目(MP2013－74);河南省教育厅 2011 年自然科学研究项目(2011A520027);2014 年度河南省教育厅科学技术研究重点项目(14A520070)
收稿时间:2014-10-08;收到修改稿时间:2014-12-17

之间或核苷酸序列之间的两两比对, 而设计出一个新型的、更快速的且更灵敏的算法是当前 DNA 序列比对研究的热点[4-6].

双序列比对是指对两条生物序列进行插入、删除空格及对字符进行匹配、替换等操作, 计算两条序列的比对得分并最优化比对得分, 比对得分最高或罚分最低的结果为最优比对. 常用于双序列比对的算法有:

1)Needleman 和 Wunsch 提出了 Needleman- Wunsh 算法[7], 该算法最初是为了解决氨基酸序列比对提出的, 后又被应用在核酸序列比对中. 该算法是利用二维矩阵 F(n, m)基于最短距离的原则, 通过某种算法来找出最佳匹配路径. 该算法的原理来源于点阵图, 适用于全局水平和相似性程度较高的双序列之间的比对.

2)T.F.Smith 和 M.S.Waterman 提出了 Smith-Waterman 算法[8], 是一种针对局部相似性区域的动态规划算法, 在对局部序列进行相似性识别时, 具有较高的灵敏度. 但该算法的复杂度最高, 需要在超级计算机上实现. Needleman-Wunsh 算法与 Smith-Waterman 算法一直都是生物信息学中最基本的算法之一.

3)Pearson 和 Lipman 提出了 FASTA 算法, 是一种启发式算法[9], 首先将比对序列中的所有字符转化成 Hash 表, 然后在数据库进行搜索时查询 Hash 表, 检索出可能的匹配. 该算法比 Smith-Waterman 算法更快速但精确性相近.

4)Altschul 提出了 BLAST 算法, 其基本思想是通过产生数量较少但质量较好的匹配片段, 利用片断匹配算法和有效的统计模型来找出目的序列和数据库之间的最佳局部比对效果. BLAST 运算速度最快, 但敏感性最差. BLAST 算法是目前使用最广泛的比对算法[10,11].

2 DNA双序列比对的数学描述

定义 1. 对于给定的两条 DNA 序列 S 和序列 T, |S|表示为 DNA 序列 S 的长度, |T|表示为 DNA 序列 T 的长度. 序列S 和序列T 中的每个字符都来自给定的字母表 Σ . DNA 序列的字母表 Σ ={A, G, C, T}, 由 4 种碱基 A, G, C, T 组成[12].

定义 2. 在进行 DNA 双序列比对时, 需要进行替代(substitute)、删除(delete)与插入(insert)等操作. 空格“-”代表删除或插入操作时所产生的空格. 需要满足以下 3 个条件:

1)DNA 双序列比对是一个 k×l 的矩阵 M (mij) k ×l;

2) " j, $ i, mij≠‘-’, 即在进行空格“-”插入或删除时, 要求不存在全为空格的一列;

3)将比对后的序列中的所有空格删除, 可以得到原来的 DNA 序列.

定义 3. 对于 x, y∈∑∪{-}, 定义 σ(x, y)为计分函数, 表示字符 x, y 比对时的得分, 最简单的

$$\sigma(x,y)=\begin{cases}2(x=y\in\Sigma)\\-1(x\neq y\in\Sigma)\\-2(x="-"\ \ or\ \ y="-"\end{cases} \quad (1)$$

公式 1 表明, 若两个字符相等且都属于字母表, 则得分为2; 若两个字符不相等且都属于字母表, 则得分为-1; 若两个字符存在空格“-”, 则得分为-2. 比对的得分越高, 表示两组序列的相似性越高. 而在进行 DNA 双序列比对时, 得分最高的比对是最优的序列比对.

假设存在两条 DNA 序列, 序列 S=ACGCTAG, 序列 T=CTGACAC, 比对结果为

S=AC-G-CTAG

T=-CTGAC-AC

序列 S 和 T 形成的一个比对 A 时, 序列 S’, T’是插入空格“-”后的序列 S 和 T, 则必须满足|S’|=|T’|,并且序列 S’,T’去掉空格之后就是序列 S 和 T. 序列 S’,T’形成的比对序列 A 的得分是 score(A)=∑i=1s’ σ(S’[i], T’). 根据公式 1 可得[13]:

score(A)=(-2)+2+(-2)+2+(-2)+2+(-2)+2+(-1)=-1,

则序列 S 和序列 T 的比对得分为-1.

3 双序列比对算法

3.1 Needleman-Wunsch 算法

Needleman—Wunsch 算法是典型的全局比对算法, 这种算法适用于全局水平上相似性程度较高的 2 个序列. Needleman 和 Wunsch 于 1970 年在他们的文章中提出了两条序列相似性比对的动态规划算法(dynamic programming). 基本思想是: 基于渐进方法对两条序列对应的字符进行对比, 并将相似性记录在一个矩阵中, 称为计分矩阵; 通过计分矩阵, 从矩阵最右下方

一个单元开始利用回溯法找到最优比对值[14].

对于给定的两条 DNA 序列 S=(s[i], s[2],…, s[m])与序列 T=(t[1], t[2],…, t[n]), M(i,j)表示两条序列的最优比对得分. 在求 M(i,j)时, 必须知道前三项的值, 分别为 $M_{i-1,j}$, $M_{i,j-1}$, $M_{i-1,j-1}$. 最优比对得分的递推公式为:

$$M_{i,j}=\max\begin{cases}M_{i-1,j}+\sigma(s[i],-)\\ M_{i-1,j-1}+\sigma(s[i],t[j])\\ M_{i,j-1}+\sigma(-,t[j])\end{cases} \tag{2}$$

其中, $M_{i,0}=\sum_{k=0}^{i}\sigma(s[i],-)$, $M_{0,j}=\sum_{k=0}^{i}\sigma(-,T[j])$.

对于给定两条 DNA 序列 S、序列 T 与相似矩阵, 回溯法求解最优比对的具体步骤为:

```
Best Align (S,T,M)
For(i=m&&j=n;i>0&&j>0)
if(Mi,j == Mi-1,j+ σ(s[i],-))
i--;
insert("-",T,j);
else if(Mi,j == Mi-1,j-1 + σ(s[i],t))
i--;
j--;
else if(Mi,j == Mi,j-1 + σ(-,T))
j--;
insert("-",S,i);
end;
```

具体步骤为:

1)初始化序列 S 与序列 T 的相似矩阵;

2)根据得分规则, 填写两个碱基对应的得分;

3)从得分矩阵的右下方, 即从最大分值开始进行路线回溯;

4)将回溯路线转化成比对结果.

通过基本步骤得出, Needleman-Wunsch 算法的时间复杂度与空间复杂度都为 O(mn)

假设两条 DNA 序列 S=CGAATTC 与序列 T=CGGATC, 根据递推公式以及 NW 具体步骤, 可以得出以下相似矩阵, 再由公式可以求出回溯路线, 图中箭头表示回溯路线.

则可知最优比对值为 7, 最优比对结果为:

C G A A T T C

C G G A T – C

-----	---	C	G	A	A	T	T	C
----	0	-2	-4	-6	-8	-10	-12	-14
C	-2	2	0	-2	-4	-6	-8	-10
G	-4	0	4	2	0	-2	-4	-6
G	-6	-2	2	3	1	-1	-3	-5
A	-8	-4	0	4	5	3	1	-1
T	-10	-6	-2	2	3	7	5	3
C	-12	-8	-4	0	1	5	6	7

图 1　使用 NW 算法进行序列比对的得分矩阵

3.2 Smith-Waterman 算法

Smith-Waterman 算法是 Smith 和 Waterman 于 20 世纪 80 年代初提出的一种针对局部相似性区域的动态规划算法[3]. Smith-Waterman 算法对 NW 算法进行了改进,对于局部最优化有较大优势. NW 算法与 Smith-Waterman 算法一直都是生物信息学中最基本的算法之一.

在 Smith-Waterman 算法中, 不需比对整个序列. 两个零长字符串即为得分为 0 的局部比对, 这一事实表明在构建局部比对时, 不需要使用负分. 这样会造成进一步比对所得到的分数低于通过 “重设” 两个零长字符串所能得到的分数. 而且, 局部比对不需要到达任何一个序列的末端, 所以也不需要从右下角开始回溯: 可以从得分最高的单元格开始回溯.

递推公式为:

$$S[i,j]=\max\begin{cases}S[i-1,j-1]+\sigma(a_i,b_j)\\ S[i-1,j]+gap\_penalty\\ S[i,j-1]+gap\_penalty\\ 0\end{cases} \tag{3}$$

其中, Mi,0 = 0,M0,j = 0.

对于给定两条 DNA 序列 S、序列 T 与相似矩阵, Smith-Waterman 算法求解最优比对的具体步骤为:

1)初始化由序列 S 与序列 T 组成的相似矩阵;

2)根据已定义的得分规则, 填写两个碱基对应的得分;

3)从得分矩阵的右下方的最大分值开始进行路线回溯;

4)直到找到得分矩阵中为 0 的元素, 停止寻找;

5)将回溯路线转化成比对结果.

通过基本步骤得出, Smith-Waterman 算法的时间

复杂度与空间复杂度都为 O(mn).

对于两条 DNA 序列 S 与序列 T, Smith-Waterman 算法的基本步骤为:

```
Best Align(S,T,M)
For(i=m&&j=n;i>0&&j>0)
gap_penalty=-2;
if(S[i,j]=S[i-1,j]+gap_penalty)
i--;
insert("-",T,j);
else if(S[i,j]=S[i-1,j-1]+σ(ai,bj))
i--;
j--;
      else if(S[i,j] = S[i,j-1] + gap_penalty)
j--;
insert(“-”,S,i);
end;
```

假设两条 DNA 序列 S=AGATCCG 与序列 T=AGCATG, 根据递推公式以及 Smith-Waterman 算法的具体步骤, 可以得出以下相似矩阵, 再由公式可以求出回溯路线.

图 2 中箭头表示回溯路线, 其中空位罚分 gap_penalty=-2.

则可知最优比对值为 7, 最优比对结果为:

A G – A T C C G

A G C A T – – G

	---	A	G	A	T	C	C	G
---	0	0	0	0	0	0	0	0
A	0	2	0	2	0	0	0	0
G	0	0	4	2	0	0	0	1
C	0	0	2	2	0	2	4	2
A	0	2	0	4	2	0	2	2
T	0	0	0	2	6	4	2	0
G	0	0	2	0	4	4	2	4

图 2 SW 算法进行序列比对的得分矩阵

3.3 Smith-Waterman 算法改进

上述两种算法在回溯路径的计算方面是相同的, 二者的区别主要体现在得分矩阵:

1)在初始化阶段, SW 算法中的得分矩阵第一行和第一列全填充为 0(而且第一行和第一列的指针均为空); NW 算法中使用位于顶部的第一个序列中的字符, 并使用空格, 给以前的单元格加了-2 分, 因此第一行得到了 0, -2, -4, -6, ... 这样的序列, 用相似的方法得到第一列的得分.

2)在填充表格时, SW 算法使用 0 代替负分, 只对得分为正的单元格添加返回指针; NW 算法允许负分出现, 每个单元格的值来自以下 3 个中的最大值: ①上方的值-2 ②左边的值-2 ③如果该单元格所在的行与所在的列对应的字符相等, 则为左上值加 1, 否则为左上值-1.

改进算法主要从回溯路径的求解方面进行改进. 上述两种算法在求回溯路径时都是从得分矩阵的右下方找出最大分值进行回溯, 改进算法则在填充得分矩阵时同时记录得分的来源方向, 回溯时直接读取这组数据即可. 算法改进的基本思路:

①在遍历的过程中, 路径只有 3 个方向: 左上、上、左. 对这三个方向分别编号为 0、1、2. 这样, 可以将遍历的过程记录在一个数组中.

②在为二维数组赋值和计算的过程中同时记录数据的来源方向, 将来源方向记录到相应的二维数组中.

③将最优回溯路径记录到一个一维数组中, 输出一维数组中对应的字符即可.

```
i = |S|,j = |T|;k = 0;
begin
if(V[i,j] = V[i-1,j-1] + f(S[i],))
then(i--,j--,k = 0)
endif;
if(V[i,j] = V[i-1,j-1] + f(S[i],-))
   then(i--,insert(‘-’,T,j),k = 1)
endif;
if i = 0 and j = 0     //回溯路径终止处
   endif;
      end;
```

4 算法的实验与比较

以下实验均是在操作系统 Windows XP SP3 中运行, 使用的开发工具是 Dev-C++ , Version 4.9.9.2, Microsoft Visual C++6.0. 实验的硬件环境是: Intel 赛扬双核处理器, T1600 1.66GHz, 内存 1GB.

实验数据来自 NCBI 网站上相关的 DNA 序列, 数据存放到 10 个文本文件中, 文件名依次用 s1,

s2……s10 命名, 每个文件都包含一个 DNA 序列, 且序列长度从 50 递增到 500, 递增长度为 50, 因此有序列 s1 长度＝50, 序列 s2 长度＝100, 序列 s3 长度＝150, ……, 序列 s10 长度＝500. 序列 t 同理. 通过文本自动读入数据进行测试, 程序每次读入长度相等的序列 s 和序列 t 进行比对, 输出比对后的序列, 以及比对所消耗的时间.

4.1 Needleman-Wunsch 算法实验

通过实验验证 Needleman-Wunsch 算法, 并对算法的运行性能进行分析, 由实验得出的数据(如表 1)可以看出, 随着比对序列的长度增加, 比对所消耗的时间也逐渐增加, 同时比对后的序列长度普遍大于比对前的长度. 表 1 是用 10 组 DNA 序列测试 Needleman-Wunsch 算法的运行性能所得出的数据.

表 1　Needleman-Wunsch 算法的运行性能统计

序列 s	序列 t	比对前序列长度	比对后序列长度	程序运行时间(s)
s1	t1	50	57	0.532000
s2	t2	100	110	0.985000
s3	t3	150	163	2.187000
s4	t4	200	222	4.657000
s5	t5	250	279	6.937000
s6	t6	300	332	8.609000
s7	t7	350	387	13.000000
s8	t8	400	443	15.343000
s9	t9	450	498	20.719000
s10	t10	500	552	24.891000

4.2 Smith-Waterman 算法实验

通过实验验证 Smith-Waterman 算法, 并对算法的运行性能进行分析, 由实验得出的数据(如表 2)可以看出, Smith-Waterman 算法在比对消耗的时间方面也是随序列长度的增加而增加, 但增加的速度略小于 Needle-Wunsch 算法, 比对后序列长度也是随着序列长度增加而增加, 但在序列长度较小时, 比对后序列是小于比对前序列的, 在序列长度较大时, 比对后序列才大于比对前序列; 但 Smith-Waterman 算法的比对后序列整体上要小于 Needleman-Wunsch 算法的比对后序列(如图 3). 表 2 是用 10 组 DNA 序列测试 Smith-Waterman 算法的运行性能所得出的数据.

表 2　Smith-Waterman 算法的运行性能统计

序列 s	序列 t	比对前序列长度	比对后序列长度	程序运行时间(s)
s1	t1	50	23	0.594000
s2	t2	100	86	1.047000
s3	t3	150	140	2.672000
s4	t4	200	202	4.703000
s5	t5	250	259	6.953000
s6	t6	300	315	9.875000
s7	t7	350	368	12.921000
s8	t8	400	427	16.485000
s9	t9	450	481	21.047000
s10	t10	500	532	25.047000

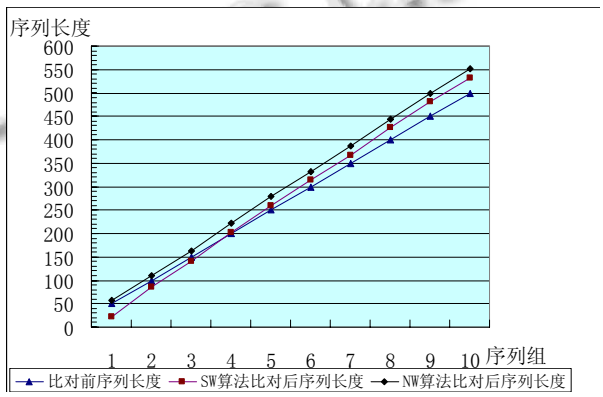

图 3　NW 算法和 SW 算法比对后的序列长度比较

4.3 Smith-Waterman 算法改进实验

通过实验验证 Smith-Waterman 改进算法, 并分析算法性能, 由实验得出的数据(如表 3)可以看出, 改进后的 Smith-Waterman 算法能够尽可能快的找到最优局部比对序列, 所得到的最优局部序列也更短, 在算法运行时间方面也比改进前有明显提升(如图 4). 但同时也存在这样的问题, 需要进行更多的序列测试, 得到更为精确的结果.

表 3　Smith-Waterman 算法改进后的性能参数

序列 s	序列 t	比对前序列长度	比对后序列长度	改进后程序运行时间(s)
s1	t1	50	22	0.546000
s2	t2	100	82	0.594000
s3	t3	150	136	1.547000
s4	t4	200	193	2.922000
s5	t5	250	251	4.719000
s6	t6	300	307	6.078000
s7	t7	350	359	7.765000
s8	t8	400	416	9.812000
s9	t9	450	472	11.609000
s10	t10	500	521	14.907000

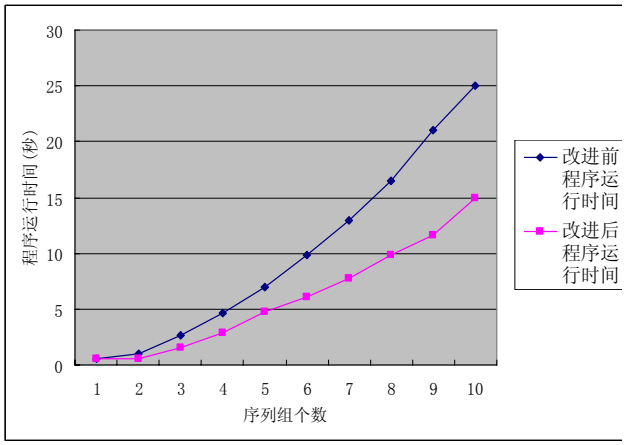

图 4 Smith-Waterman 算法改进前后的运行时间对比

5 结语

以DNA和蛋白质序列为基点, 从数学的抽象思维对序列比对进行描述分析, 进而对生物序列比对中的Needleman-Wunsch 算法和 Smith-Waterman 算法进行深入研究和分析, 并对算法进行改进, 实现了双序列局部最优解个数的优化, 有效降低了生物序列比对算法时间与空间的复杂性, 提高序列比对的得分率和准确率.

参考文献

1 孙勇.基于关键字树和滑动窗口的大规模生物遗传序列的算法研究[硕士学位论文].重庆:西南大学,2013.

2 曹雪卉.DNA 词序列比对及应用[学位论文].哈尔滨:哈尔滨工业大学,2013.

3 Zhu XY, Li KL, Salah A. A data parallel strategy for aligning multiple biological sequences on multi-core computers. Computers in Biology and Medicine, 2013.

4 丁茂华.生物序列数据库相似性搜索算法研究[硕士学位论文].扬州:扬州大学,2013.

5 Orobitg M, Cores F, Guirado F, Roig C, Notredame C. Improving multiple sequence alignment biological accuracy through genetic algorithms. The Journal of Supercomputing, 2013, 653.

6 Flouri T, Frousios K, Iliopoulos CS, Park K, Pissis SP, Tischler G. GapMis: a tool for pairwise sequence alignment with a single gap. Recent Patents on DNA & Gene Sequences, 2013, 72.

7 向旭宇.基因序列与结构的信息分析及应用算法研究[学位论文].长沙:湖南大学,2010.

8 詹青,王亚东.基于平均交互信息量的 DNA 序列相似性分析.智能计算机与应用,2011,4:31–34,52.

9 Othman MTB, Abdel-Azim G. Genetic algorithms with permutation coding for multiple sequence alignment. Recent Patents on DNA & Gene Sequences, 2013, 72.

10 赵裕众.生物序列分析算法的研究及其应用[学位论文].合肥:中国科学技术大学,2010.

11 Jose L, Daniel R, Anuj S, Eric K, Zhang JF. RNA global alignment in the joint sequence-structure space using elastic shape analysis. Nucleic Acids Research, 2013, 4111.

12 马海晨,韦刚,吴百峰.基于 GPGPU 的生物序列快速比对.计算机工程,2012,4:241–244.

13 吕品一,郑珩,劳兴珍.蛋白质共进化分析研究进展.生物信息学,2010,01:34–37.

14 刘维.生物序列模式挖掘与识别算法的研究[学位论文].南京:南京航空航天大学,2010.