用动态扣分策略消除序列局部联配中的马赛克问题*)

佘 堃[1,2] 黄均才[1,2] 周明天[1,2]

(电子科技大学计算机学院 成都610054)[1] (电子科技大学-卫士通联合实验室 成都610054)[2]

Fixing Mosaic Effect on Sequence Local Alignment with Dynamic Penalty Strategy

SHE Kun[1,2] HUANG Jun-Cai[1,2] ZHOU Ming-Tian[1,2]

(College of Computer Sci&Engineering , Univ. of Electronic Sci&Tech of China, Chengdu 610054)[1]

(Information Security United Lab. of UESTC-Westone, Chengdu 610054)[2]

Abstract The local alignment problem for two sequences requires determining similar regions, one from each sequence, and aligning those regions. The Smith-Waterman algorithm for local sequence alignment is one of the most well-known algorithm in computational molecular biology. This ingenious dynamic programming approach is designed to reveal the highly conserved fragments by discarding poorly conserved initial and terminal segments. However, the local alignment sometimes produces a mosaic of well conserved fragments artificially connected by poorly conserved or even unrelated fragments. This may lead to problems in comparison of long genomic sequences and comparative gene prediction. In this paper we propose a new strategy of dynamic penalty strategy to fix this problem. In the process of computing similarity matrix, if similarity value is larger than the pre-specified threshold X then starting our strategy, when related character mismatches, then penalizing more than others until similarity value is 0 or the process ends. Test results show that this algorithm has better performance by comparison to the standard Smith-Waterman while dose not increase signally the computational complexity both in time and space.

Keywords Sequence alignment, Smith-Waterman algorithm, Mosaic effect, Dynamic penalty strategy

1 引言

生物信息学中,对各种生物大分子序列进行分析是一件非常基本的工作,Paul A. Rota[1]通过测定SARS(又名非典型性肺炎)病毒的基因组序列,找到了含有制造蛋白质指令的部分基因,其中包括公认的制造四种蛋白质的基因,证实了非典病毒是一种全新的冠状病毒,这一成果将有助于加快诊断、治疗和预防非典的步伐。Asao Fujiyama[2]在比较黑猩猩和人类的基因组时,发现了几个包括人类21号染色体上的两个簇(clusters)的可能的位置。Richard D. Wood[3]通过将人类基因组中编码的蛋白质和那些果蝇和线虫编码的相比较,发现在人类基因组中至少有18个蛋白质属于核苷三磷酸酶NACHT家族,且与神经元凋亡阻碍蛋白NAIM有关。在遗传物质长期的演化过程中,原本相同的DNA序列由于其中一条序列缺失了几个片断,或增加了几个片断,或某段子序列发生了位置的变化等,从而导致它们产生了不同的序列,这两条序列不一定能进行精确的匹配,但是它们有一定的相似度。我们应该如何判定序列之间的这种相似性?在寻找序列最优相似比较的算法中有两种算法使用最为广泛:Blast算法和Smith-Waterman[4]算法,Blast算法的运行速度要比Smith-Waterman算法快,但Smith-Waterman算法更为精确。不过,把Smith-Waterman算法用于长序列的局部联配时经常会出现马赛克现象[5],Huang[6]、Zhang[7]试图在后处理阶段解决这一问题,但后处理方法有可能会漏掉一些相似度较高的局部联配[8]。A. N. Arslan[8]提出用正常化局部联配算法(NLA消除马赛克现象)在计算得分矩阵时考虑到了联配序列长度对局部联配质量的影响,并引入长度修正系数L对得分加以修正。NLA虽然消除了马赛克现象,但却也带来了新的问题,即L的确定与数据相关且没有可遵循的固定模式[8]。

本文在分析马赛克问题成因的基础上,提出利用动态扣分策略(对特定情况采用动态扣分以切断异常联配)解决马赛克问题。实验结果表明,在没有显著增加算法的时间复杂度和空间复杂度均的前提下,顺利解决了序列局部联配中的马赛克问题。

2 Smith-Waterman算法介绍

2.1 相关知识

为判定序列之间的相似性,生物学家提出了一种用来评定序列相似性的方法,称为记分函数的方法。

定义1 如果S是一个序列,那么|S|表示S中的字符长度,S[i]表示序列的第i个字符。如果序列S和序列T相同,必须满足如下条件:(1)|S|=|T|;(2)S[i]=T[i]($0<i\leqslant|S|$)。

定义2 如果x和y是两个字符,那么$\sigma(x,y)$表示x和y字符在进行比较时所得的分值,σ称为一个记分函数。记分函数还包括当x为空字符或y为空字符的情况,在序列中一个所谓的空字符'—'表示序列中空字符的位置可能缺失一个未知的字符,我们只能使用空字符来表示这种缺失。

定义3 如果S和T是两个序列,那么S和T的一个相似性比较A可以用S'和T'来表示,其中:(1)|S'|=|T'|;(2)将S'和T'中的空字符除去后所得的序列分别和S、T相同。

*)本文的工作得到电子科学基金51415010101DZ02的资助。佘 堃 在职博士生,主要研究方向为网络计算,信息安全,软件工程。黄均才 硕士研究生,主要研究方向为生物信息学,网络计算,信息安全。周明天 博士生导师,主要研究方向为网络计算,信息安全,分布式计算,并行计算,移动计算。

相似性比较 A 就是 S' 和 T' 中字符一一比对，相似性比较 A 的得分 $Score$ 可以用如下公式表示：

$$Score=\sum_{i=1}^{t}\sigma(S'[i],T'[i]); \quad (1)$$

其中 $t=|S'|=|T'|$。

定义4 对于两个序列 S 和 T，它们的最优相似性比较 A 是指在 S 和 T 的所有相似性比较中得分最高的一个。序列相似性比较算法的主要目标就是如何寻找出序列间的最优相似性比较。

2.2 Smith-Waterman 算法

Smith-Waterman 算法先用迭代方法计算出两个序列的所有可能相似性比较的分值，然后通过动态规划的方法回溯寻找最优相似性比较，具体描述如下：

对于两个序列 S 和 T，S [i]和 T [i]，其中0<i≤|S|，0<j≤|T|，都属于某个字符集 Ω，对 Ω 中的任何元素和空符号，它们两两之间都有一个记分值，用记分函数 σ(x,y)表示，V(i,j)表示序列 S 的前缀 S [1]S [2]…S [i−1]S [i]和序列 T 的前缀 T [1]T [2]…T [j−1]S [j]之间的最优相似性比较的得分。那么有如下公式：

$$V(i,0)=0$$
$$V(0,j)=0 \quad (2)$$
$$V(i,j)=\max\begin{cases}0\\ V(i-1,j-1)+\sigma(S[i],T[j])\\ V(i-1,j)+\sigma(S[i],-)\\ V(i,j-1)+\sigma(-,T[j])\end{cases}$$

通过公式(2)我们可以得到如下的一个得分矩阵[V]：

$$\begin{bmatrix} V(0,0) & V(0,1) & \cdots & V(0,j) & \cdots & V(0,n)\\ V(1,0) & V(1,1) & \cdots & V(1,j) & \cdots & V(1,n)\\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots\\ V(i,0) & V(i,1) & \cdots & V(i,j) & \cdots & V(i,n)\\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots\\ V(m,0) & V(m,1) & \cdots & V(m,j) & \cdots & V(m,n)\end{bmatrix}$$

根据得分矩阵进行动态规划回溯即可获得相似性比较。

Smith-Waterman 算法的整体时间复杂度为 $O(mn)$。使用 Smith-Waterman 算法计算两个序列最大相似性比较时绝大部分计算时间将消耗在计算得分矩阵[V]的值上。

3 马赛克问题分析

马赛克问题定义：在序列局部联配的最优排列中间会经常出现相似度很低的保守区域，这称为马赛克问题。Zhang[7] 给出了马赛克问题的理论原因，如图1所示。

图1 序列联配中包含一段相似度很低的保守区域。如果分值为−X 的保守区域夹在两段分值大于 X 的序列段中，不管 X 有多大，Smith-Waternman 算法都将把这三段区域连成单一的联配。

为说明马赛克问题的成因，我们可以看一个简单的例子。

设 DNA 序列 S="taatcctattgac"，T="aatggggttgat"，为说明问题，我们对序列作了许多简化和抽象。

若取记分函数 $\sigma(x,y)$：

$$\sigma(x,x)=2,\sigma(x,y)=\sigma(x,-)=\sigma(-,y)=-1$$

则可根据式(2)求得得分矩阵如图2所示。

i \ j	0	1 a	2 a	3 t	4 g	5 g	6 g	7 g	8 t	9 t	10 g	11 a
0	0	0	0	0	0	0	0	0	0	0	0	0
1 t	0	0	0	2	1	0	0	0	2	2	1	0
2 a	0				1	0	0	0	1	1	1	3
3 a	0				2	1	0	0	0	0	0	3
4 t	0				5	4	3	2	2	2	1	2
5 c	0	0	2	5	5	4	3	2	1	1	1	1
6 c	0	0	1	4	4	4	3	2	1	0	0	0
7 t	0	0	0	3	3	3	3	2	4	3	2	1
8 a	0	2	2	2	2	2	2	2	3	3	2	4
9 t	0	1	1	4	3	2	1	1				
10 t	0	0	0	3	3	2	1	0				
11 g	0	0	0	2	5	5	4	3				
12 a	0	2	2	1	4	4	4	3				

图2 出现马赛克现象

图2中，中间的保守区域长度为4，得分为−4，夹在两段分值为6和8(均大于4)的序列段中，局部联配的结果为：

```
aatggggttga
|||    ||||
aatccta ttga
```

由于序列较短，看不到问题的严重性。如果保持序列 S 和 T 的构成情况不变而将它们的长度增加为原来的100倍，这时中间的保守区域长度为400，分数为−400，夹在两段分值为600和800(均大于400)的序列段中，而 Smith-Waternman 算法将把这三段区域连成单一的联配，长度为1000，分数1000，于是马赛克现象便出现了。

还是同样的序列 S 和 T，我们仅仅改变记分函数 $\sigma(x,y)$：

$$\sigma(x,x)=4,\ \sigma(x,y)=\sigma(x,-)=\sigma(-,y)=-3$$

同样根据式(2)求得得分矩阵如图3所示。

i \ j	0	1 a	2 a	3 t	4 g	5 g	6 g	7 g	8 t	9 t	10 g	11 a
0	0	0	0	0	0	0	0	0	0	0	0	0
1 t	0	0	0	4	1	0	0	0	4	4	1	0
2 a	0				1	0	0	0	1	1	1	5
3 a	0				2	0	0	0	0	0	0	5
4 t	0				9	6	3	0	4	4	1	2
5 c	0	0	2	9	9	6	3	0	1	1	1	0
6 c	0	0	0	6	6	6	3	0	0	0	0	0
7 t	0	0	0	4	3	3	3	0	4	4	1	0
8 a	0	4	4	1	1	0	0	0	1	1	1	5
9 t	0	1	1	8	5	2	0	0				
10 t	0	0	0	5	5	2	0	0				
11 g	0	0	0	2	9	9	6	4				
12 a	0	4	4	1	6	6	6	3				

图3 没有马赛克现象

图3中无保守区域，没有马赛克现象出现。局部联配结果为：

```
ttga
||||
ttga
```

这说明 Smith-Waterman 算法的局部联配输出结果对记分参数的选择是比较敏感的。

4 Smith-Waterman 改进算法

4.1 利用动态扣分策略改进 Smith-Waterman 算法

从前面的分析可以看出，将 Smith-Waterman 算法用于局部联配时，如果记分函数选择不当则会带来以下问题：扣分过少，容易出现马赛克现象，扣分过多，则又有可能漏掉一些具有生物学意义的局部联配片段。

实际上，前面所说的问题是由于原有算法采用静态扣分策略的必然结果，因此，我们可以采用动态扣分策略加以解决。

在计算得分矩阵时，记下在计算 V[i,j]的过程中沿途出现的 V[r,s](r≤i,s≤j)的最大值 hm(i,j)及其位置 max_i，max_j，并称 hm(i,j)为 V[i,j]的沿途最大值。

为便于说明问题，我们在这里将 hm(i−1,j)，hm(i,j−1)，hm(i−1,j−1)简记为 hm，将 hm 及其自沿途最大值相应的位置(max_i,max_j)起至(i,j)的扣分次数（包括本次）t、距离 d 等统一记为 x(具体实现时用一个结构体表示)，它们将分别影响 σ(S[i],'−')、σ('−',T[j])和 σ(S[i],T[j])(如果 S[i]≠T[j])。

在计算得分矩阵时，记下 V[i,j]的沿途最大值 hm(i,j)及其位置 max_i,max_j，

如果 V[i,j]= V[i−1,j]+σ(S[i],'−')，则 hm(i,j) = hm(i−1,j)；

如果 V[i,j]= V[i,j−1]+ σ('−',T[j])，则 hm(i,j) = hm(i,j−1)；

如果 V[i,j]=V[i−1,j−1]+σ(S[i],T[j])且 V[i,j] ≤hm(i−1,j−1)，则 hm(i,j)= hm(i−1,j−1)；

如果 V[i,j]= V[i−1,j−1]+ σ(S[i],T[j])且 V[i,j]>hm(i−1,j−1)，若已采用动态扣分策略，则按预先设定的得分函数重新计算自(max_i,max_j)至(i,j)的 V(i,j)，并令 hm(i,j)= hm(i−1,j−1)，max_i=i,max_j=j。

将预先确定的记分函数记为：

$\sigma(x,x)=c$，$\sigma(x,y)=\sigma(x,-)=\sigma(-,y)=-b$。

动态扣分策略基本思想：如果存在保守区域，我们争取在离开保守区域前让得分为0，从而将保守区域切断。

设自沿途最大值相应的位置(max_i,max_j)起至(i,j)的扣分次数（包括本次）为 t，距离为 d，按预先确定的记分函数，则自(max_i,max_j)起至(i,j)的记分总和为 $Score(t,d)=(d-t)c-bt$。

令 p 为动态扣分策略对应的每次扣分。

在计算得分矩阵的过程中，当 hm(i,j)>X 时，启动动态扣分策略计算后面的矩阵元素。即

i 当 V[i−1,j]的沿途最大值 hm(i−1,j)大于 X 且 Score(t,d)<0时，σ(S[i],'−')=p，不再取事先设定的分数；

ii 当 V[i,j−1]的沿途最大值 hm(i,j−1)大于 X 且 Score(t,d)<0时，σ('−',T[j])=p，不再取事先设定的分数；

iii 当 V[i−1,j−1]的沿途最大值 hm(i−1,j−1)大于 X 且 Score(t,d)<0时，若 S[i]≠T[j]，则 σ(S[i],T[j])=p，不再取事先设定的分数。

当 V[i,j]=0时，动态扣分策略自动停止。

其余计算与原 Smith-Waternman 算法相同。

下面确定与动态扣分策略对应的每次扣分 p。

设自(max_i,max_j)起是总记分为-X(X 为预先确定的参数)的保守区域，根据动态扣分策略，因在 Score(t,d)>0的情况下记分仍采用预先确定的记分函数，故可将将该保守区域等效为连续扣分的片段，每次扣分 b 分，则等效为扣$[\frac{X}{b}]$次，我们现在希望每次扣分改为 p，扣$[\frac{X}{b}]$次后 V[i,j]=0，即

$$hm-\sum_{t=1}^{[X/b]}p=hm-p\cdot[\frac{X}{b}]=0 \quad (3)$$

由式(3)可确定 p，

$$p=\frac{hm}{[X/b]} \quad (4)$$

当 V(i−1,j−1)、V(i−1,j)和 V(i,j−1)已经计算出来后，我们称 V(i,j)是可计算的，如图4所示。

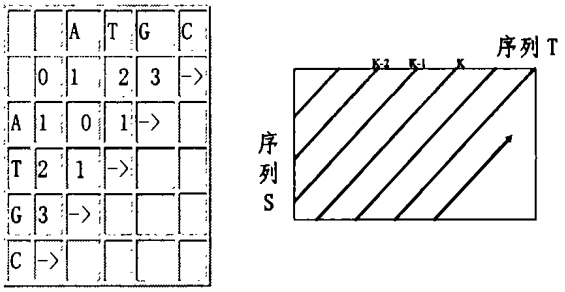

图4 得分矩阵计算过程。"−>"表示该矩阵元素当时可计算

通过上图的分析，在单 CPU 机器上可沿反对角线一行一行地向下计算，启动动态加速扣分策略后我们不必保留所有的 x，x 包括 hm 及其自(max_i,max_j)起至(i,j)的扣分次数(包括本次)t、距离 d 等数据，只保留沿反对角线的前两行就可以了，如在计算沿反对角线的第 k 行时，只保留沿反对角线的第 k−1,k−2行的 x，在多处理机环境下，沿反对角线上的数据可并行计算，这时算法就可以利用这种并行性进行并行优化。

4.2 算法复杂性分析

假设 n=|S|，m=|T|，在开始计算得分矩阵[V]时需要计算矩阵的每一个得分值，计算复杂度为 O(mn)。而回溯算法的计算复杂度为 O(max(m,n))。这样算法的整体时间复杂度为 O(mn)。计算两个序列最大相似性比较时绝大部分计算时间将消耗在计算得分矩阵[V]的值上。而改进算法比原算法需额外使用两个大小为 max(m,n)的一维数组用于存放 x，但总的空间复杂度没变。

5 实验结果

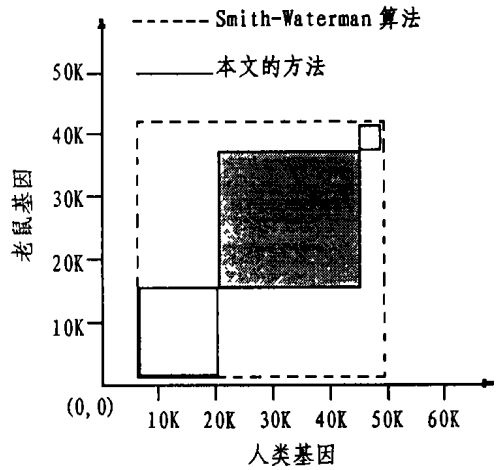

图5 测试结果

图5是人类与老鼠基因序列联配结果(数据取自 GenBank Acc. Nos，人类基因：AF030876，Z47046，Z47066和 Z68193；老

鼠基因:AF121351)。在两个基因序列中,使用 RepeatMasker 的默认设置(http://ftp.genome.washington.edu/RM/RepeatMasker.html)找出重复元素并将其从序列中剔除。虚线矩形框是利用程序 SIM(http://globin.cse.psu.edu)计算所得的结果,SIM 采用的是 Smith-Waternman 的局部联配算法,它所报告的最长的局部联配(位置在人类基因的6817-53081)将非保守区域和高保守区域(图5中的阴影矩形框)连在了一起。实线矩形框是利用本文的改进算法的计算结果,X=100,c=1,b=1。

结论 Smith-Waterman 算法是目前最著名的一种局部联配算法,但将此算法用于同源长序列的局部联配时,经常会出现马赛克问题,该问题也一直困扰着人们。本文提出利用动态扣分策略(对特定情况采用动态扣分以切断异常联配)来解决马赛克问题。实验结果表明,在基本不改变原 Smith-Waterman 算法的前提下,在没有显著增加算法的时间复杂度和空间复杂度的情况下,顺利解决了序列局部联配中的马赛克问题。

参考文献

1 Rota P A, et al. Characterization of a Novel Coronavirus Associated with Severe Acute Respiratory Syndrome. Science, 2003,300(5624)

2 Fujiyama A, et al., Construction and Analysis of a Human-Chimpanzee Comparative Clone Map. Science, 2002, 295(5552): 131~134

3 Wood R D, et al. Apoptotic Molecular Machinery: Vastly Increased Complexity in Vertebrates Revealed by Genome Comparisons. Science, 2001, 291(5507): 1279~1284

4 Smith T F, Waterman, M. S. Identfiction of common molecular Sequence. J. Mol. Biol., 1981, 97: 723~728

5 Vingron M, Waterman M S. Sequence Alignment and penalty choice. J. Mol. Biol., 1984, 235: 1~12

6 Huang X, et al. Parametric recomputing in alignment graph. In: Proc. of the 5th Annual Symposium on Combinatorial Pattern Matching, Asilomar, California 1994. 87~101

7 Zhang Z, et al. Post processing long pairwise alignments. Bioinformatics, 1999, 15: 1012~1019

8 Arslan A N, et al. A New Approach to Sequence Comparison: Normalized Sequence Alignment. Bioinformatics, 2001, 17: 327~337

(上接第37页)

双螺旋问题是公认的神经网络基准测试。它由平面上两条互相缠绕三圈的螺旋线组成,每一圈有32个点。加上一个结束点,每类共有97个点。其平面上的分布如图1。Lang 和 Witbrock 用有直接连接且结构为2-5-5-5-1的 BP 网络来分类双螺旋问题,获得成功[7]。但我们在实验中用同样的网络结构和学习参数,未能获得成功。可见 BP 网络在学习复杂任务时,成功具有很大的偶然性。Falham 和 Lebiere 把级联相关学习网络运用于求解双螺旋问题,最终得到的神经网络的隐节点12到19个之间,平均隐节点数为15.2个[8]。我们运用 Falham 和 Lebiere 发展的级联相关学习网络于双螺旋分类问题,最终生成的网络具有17个隐含节点,共有207个连接权。其泛化结果如图2。从中可以看出,级联相关学习网络的过拟合现象严重,泛化能力差。

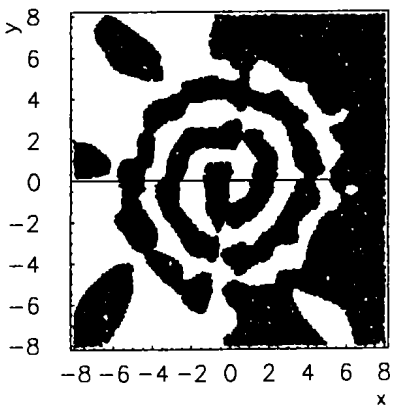

图3

我们设计的方法也能快速地学会所有的训练样本,形成一具有48个隐节点的网络,共有193个权参数。其泛化结果见图3。比较两种方法生成的网络结果,不难看出,两种网络在结构紧致性方面相当,但是,在泛化能力方面,我们设计的网络却优于级联相关网络。

结论 我们已经详细地介绍了结构神经网络的几何策略,并把它运用于双螺旋分类问题的求解。由于在结构网络的过程中没有用到误差极小的方法,所以,形成网络的速度很快。它不仅克服了人为指定的拓扑结构的不适应性的困难,而且,也克服了随机选择初始权重所引起的学习过程缓慢的困难。此外,结构神经网络的几何策略还说明了如何有效地划分特征表示空间是神经网络学习过程中至关重要的问题,也是神经网络面对愈来愈复杂的问题取得成功的关键所在。

结构神经网络的几何策略还存在着值得深入探讨的问题,包括如何更为有效地划分特征表示空间、如何修剪已经形成的网络使网络得以进一步优化地及引入容错功能于特征表示空间划分的过程等。

本文仅给出结构两层前置网络的策略,但并不是它只适用于两层,它完全可以推广到多层网络情况。

参考文献

1 Rumelhart D E, McClelland J L. Parallel Distributed Processing. Cambridge, MA: MIT press, 1987

2 Teng C C, Wab B W. Automated learning for reducing the configuration of a feedforward neural network. IEEE Trans. Neural Networks, 1994, 7(5): 1072~1085

3 LeCun Y, Denker J S, Solla S A. Optimal Brain Damage. in Advances in Neural Information Processing System 2, D. S. Touretzly, Ed. San Mateo, CA: Morgan Kaufmann, 1990. 598~605

4 Hwang J-N, You S-S, Lay S-R. The cascade-correlation learning: a projection pursuit perspective. IEEE Trans. NN, 19996, 7(2): 278~289

5 孙功星,朱科军,戴长江,等. 层次式多子网级联神经网络. 电子学报,1999,27(8):49~51

6 Ji Chuanji, Ma Sheng. Combination of weak classifiers. IEEE Trans. Neural Networks, 1997. 8(1): 32~42

7 孙功星,戴贵亮. 改进 CAS 性能的多网络表决模型. 小型微型计算机系统,2001,22(2):168~170

8 Lang K I, Witbrock M J. Learning To Tell Two Spiral Apart. In: Proc. 1988 Connectionist Models Summer School, 1989. 52~59

9 Fahlman S E, Lebiere C. The Cascade-Correlation Learning Architecture. in Advances in Neural Information Processing Systems 2, D. S. Touretzky, Ed. Los Attos, CA: Morgan Kaufmann, 1990. 524~532