基于最小生成树的多序列联配算法*)

胡桂武[1,2] 郑启伦[1] 彭 宏[1] 邓伟林[3]

(华南理工大学计算机科学与工程学院 广州510640)[1] (广东商学院数学系 广州51020)[2]

(广东职业技术学院计算机系 广州510300)[3]

摘 要 多序列联配(MAS)是现代生物信息学中的重要工具之一,MAS问题是NP-难的,因此需要一些启发式方法在合理的时间内联配大的数据集。本文提出了一个基于最小生成树的多序列联配算法,并使用BALiBASE标准数据集合,对我们的算法进行了性能评价,结果表明算法较之ClustalX类的算法其精确度更高。

关键词 多序列联配,最小生成树,联配算法,分子生物学

Multiple Sequence Alignment Based Minimum Spanning Tree

HU Gui-Wu[1,2] ZHENG Qi-Lun[1] PENG Hong[1] DENG Wei-Lin[3]

(College of Computer Science and Engineering, South China University of Technology, Guangzhou 510640)[1]

(Department of Mathematics, Guangdong Business College, Guangzhou 510320)[2]

(Department of Computing, Guangdong Industry Technical College, Guangzhou 510300)[3]

Abstract Multiple Sequence Alignment (MSA) is one of the most important tools in modern biology. The MSA problem is NP-hard; therefore, heuristic approaches are needed to align a large set of data within a reasonable time. In this paper, a new MSA algorithm is proposed. We use a Minimum Spanning Tree (MST) algorithm to construct a guide tree in which the sequences are aligned. Quality assessment of our algorithm with ClustalX was conducted using the BAliBASE benchmarks. It is found that our algorithm can provide alignments which are better than those from ClustalX in most test cases.

Keywords Multiple sequence alignment, Minimum spanning tree, Algorithm, Molecular biology

1 引言

目前DNA和蛋白质序列数据库的规模正呈指数增加,这就迫切地需要开发用于对这些海量数据进行分析的方法和工具。多序列联配作为一种新方法,它把DNA序列和蛋白质序列信息分析作为源头,在获得了蛋白质编码区的信息及与蛋白质有关的信息之后,进行系统发育重建、功能重要区域的说明、蛋白质和RNA高级结构的模拟和预测。因此多重序列的联配问题是生物信息学中的一个基础而又重要的问题,两个序列的联配问题可以用动态规划算法求得其最优解,但多序列联配的最优解问题归结成一个未解决的NP完全问题。因此,近年来,许多工作把多重序列的联配问题化为两两联配问题,以便找到其近似最优解。目前,启发式算法是最受关注的方法之一,许多成功的多序列联配软件如:MULTAL, PILEUP和ClustalX都是基于该类方法,这些联配程序具有以下相同的步骤:

i)用标准的联配算法计算联配序列两两之间的距离。

ii)用一个聚类算法构造一颗进化树。

iii)基于进化树进行序列联配。

它们仅仅在序列联配和序列组联配的次序上不一样。

本文构造并且实现了一种新的启发式联配算法,简称MstMsa。其基本思想是用最小生成树去决定序列以及序列组联配时的次序。算法测试的数据来源于标准数据库BAliBASE[5],实验结果表明该方法在测试的大多数例子中好于著名的联配程序ClustalX的结果,同时也验证了该算法的有效性。

2 多序列联配描述

2.1 多序列联配概念

两条序列$S=s_1s_2\cdots s_n$,$T=t_1t_2\cdots t_m$,间的联配具体操作过程:在序列S和T中可以插入空格字符"-",得到两个一样长的字符序列S'和T',并且S'和T'中的空格除去后所得到的序列分别为S和T;S_i,t_i表示一个核苷酸或氨基酸残基,有效的残基类型集合用Σ表示,对于DNA序列$\Sigma=\{A,G,C,T\}$,对蛋白质序列,Σ包含了20个字符,每个字符代表一种氨基酸,两条序列联配问题一般化推广即:多序列联配[4]。

定义1 对于给定的序列组$S_1,S_2,\cdots,S_k$,一次多序列联配是将它们映射为可能包含空格的$S'_1,S'_2,\cdots,S'_k$,其中满足:

i)$|S'_1|=|S'_2|=\cdots=|S'_k|$;

ii)序列S'_i去除空格后为序列$S_i(1\leqslant i\leqslant k)$。

2.2 多序列联配中的基本问题

多重序列联配在研究生物体的功能、进化、序列和结构之间的内在联系方面正变得日趋重要,已经成为生物信息学中一个很重要且具有挑战性的复杂问题,在此介绍一下其相关的基本问题[6,7]。

记分矩阵(scoring matrix):对每一对氨基酸或核苷酸的替代、插入、删除运算预置数值,由这些预置数值构成的矩阵就叫做记分矩阵。对于DNA序列联配使用的记分矩阵相对

*)基金项目:国家自然科学基金(编号:30230350)资助。胡桂武 博士研究生,主要研究方向为数据挖掘,人工智能,生物信息学。郑启伦 博导,主要研究方向为多值逻辑,智能计算。彭 宏 博导,主要研究方向为数据挖掘技术,神经网络。邓伟林 主要研究方向为数据挖掘,遗传算法。

简单，但对于蛋白质序列的联配，情况要复杂得多，通常一种记分矩阵往往难以满足需要，目前应用最广泛的两种记分矩阵是 Margaret Dayhoff 提出的 PAM 矩阵以及 Henikoff S. 和 Henikoff JG. 提出的多种 BLOSUM 记分矩阵。

空位罚分(gap penalty)：在进行序列联配时，为了更好地反映序列的相似性，就必须考虑在序列联配时插入空位并进行罚分以控制空位插入的合理性。目前有两种罚分方法：一种是一般空隙罚分，即不对空格簇与孤立空格加以区分，一个空隙是通过一个线性函数 $w(k)$ 减罚的，$w(k)=bk$，这里 k 是空格数，b 是一个与空格的计分有关的绝对值；另外一种是仿射空隙罚分函数，由 k 个空格构成的空隙的减罚函数定义为 $w(k)$，$w(k)=h+gk$，且 $w(0)=0$，h 是开放罚分，g 是延展罚分。

打分系统：对于一个联配，为了评价其质量的优劣，必须建立一个适当的打分系统，该打分系统应该考虑到许多生物学因素并且容易计算，以及能反映恰当的间隙罚分，从而使得最优化联配结果对应于所有联配中最好的得分和最小的罚分。目前基于不同的联配目的有不同的打分系统，比如：树状打分系统，循环打分系统，SP 打分系统等。这里简单介绍应用比较广泛的 SP 打分系统。SP(sum of pairs)打分被定义为一个多重联配中所有的双重联配的打分总和，对于 k 条序列联配的分值为 k 条序列中任意两条序列(共有 C_k^2 种可能)的分值 V 之和，则 $SP=\sum_{i=1}^{C_k^2} V_i$。

3 序列组联配算法

在这一部分，我们构造了一种新的序列组联配算法，前提条件在同一序列组中的序列的长度是一样长，在序列组联配过程中该算法遵循“一旦有一个空位，总有一个空位”准则。在序列组联配空位罚分中充分使用仿射空隙罚分。

已知两个序列组：$X=\{X_1,X_2,\cdots,X_m\}$ 和 $Y=\{Y_1,Y_2,\cdots,Y_n\}$，其中 $X_k=X_{k1}X_{k2}\cdots X_{kl_1}$，$Y_k=Y_{k1}Y_{k2}\cdots Y_{kl_2}$，并 L_1,L_2 分别是序列组 X 和 Y 中序列的长度，令

$$X[i]=\begin{pmatrix}X_{1i}\\X_{2i}\\\vdots\\X_{mi}\end{pmatrix}\ i=1,\cdots,L_1,\quad Y[i]=\begin{pmatrix}Y_{1i}\\Y_{2i}\\\vdots\\Y_{ni}\end{pmatrix}\ i=1,\cdots,L_2$$

则 $X=X[1\cdots L_1]$，$Y=Y[1\cdots L_2]$ 用动态程序设计实现两组序列联配，由于仿射空隙罚分中，必需区别空隙中的第一个空格和其它空格以进行不同的减罚分。通常通过使用3个数组来达到的。每一个数组中的入口项具有如下意义：

$A_{i,j}=X[1\cdots i]$ 与 $Y[1\cdots j]$ 之间联配的最大计分，其中 $X[i]$ 与 $Y[j]$ 匹配。

$B_{i,j}=X[1\cdots i]$ 与 $Y[1\cdots j]$ 之间联配的最大计分，其中 X 的一个空格与 $Y[j]$ 匹配。

$C_{i,j}=X[1\cdots i]$ 与 $Y[1\cdots j]$ 之间联配的最大计分，其中 $X[i]$ 与 Y 的一个空格匹配。

$T_{i,j}=X[1\cdots i]$ 与 $Y[1\cdots j]$ 之间联配的最大计分。

则 $A_{i,j},B_{i,j},C_{i,j},T_{i,j}$ 满足：

$A_{0,0}=0\quad A_{i,0}=-\infty\quad A_{0,j}=-\infty$，

$B_{i,0}=-\infty\quad B_{0,j}=-m\sum_{l=1}^{n}\sum_{p=1}^{j}w(-,Y_{lp})$，

$C_{i,0}=-n\sum_{k=1}^{m}\sum_{p=1}^{i}w(X_{kp},-)\quad C_{0,j}=-\infty$，

$T_{0,0}=0\quad T_{i,0}=-n\sum_{k=1}^{m}\sum_{p=1}^{i}w(X_{kp},-)$

$T_{0,j}=-m\sum_{l=1}^{n}\sum_{p=1}^{j}w(-,Y_{lp})$

$$A_{i,j}=\sum_{k=1}^{m}\sum_{l=1}^{n}w(X_{ki},Y_{lj})+\max\begin{cases}A_{i-1,j-1}\\B_{i-1,j-1}\\C_{i-1,j-1}\end{cases}$$

$$B_{i,j}=\max\begin{cases}A_{i,j-1}-m\sum_{l=1}^{n}w(-,Y_{lj})\\B_{i,j-1}-m\sum_{l=1}^{n}w(-,Y_{lj})\\C_{i,j-1}-m\sum_{l=1}^{n}w(-,Y_{lj})\end{cases}$$

$$C_{i,j}=\max\begin{cases}A_{i-1,j}-n\sum_{k=1}^{m}w(X_{ki},-)\\B_{i-1,j}-n\sum_{k=1}^{m}w(X_{ki},-)\\C_{i-1,j}-n\sum_{k=1}^{m}w(X_{ki},-)\end{cases}$$

$$T_{i,j}=\max\begin{cases}A_{i,j}\\B_{i,j}\\C_{i,j}\end{cases}$$

$1\leqslant i\leqslant L_1$，$1\leqslant j\leqslant L_2$ 不妨设仿射空隙罚函数为 $w(k)=h+gk$，由于序列组中的元素可能为“—”(GAP)，令 $w(-,-)=0$，同时 $w(-,Y_{lj})$，$w(X_{ki},-)=g$ 如果在同一行序列中“—”连续出现，否则为 $h+g$。

GroupAlignment Algorithms：

Step0：输入待联配的两个序列组 $X=\{X_1,X_2,\cdots,X_m\}$ 和 $Y=\{Y_1,Y_2,\cdots,Y_n\}$。

Step1：计算 DP 矩阵 $T=(T_{i,j})_{L_1\times L_2}$

Step2：回溯 DP 矩阵 T 找到序列组的最终多序列联配。

Step3：输出多序列联配。

4 基于最小生成树的多序列联配算法

首先构造待联配序列集合的完全图：其中每一个接点表示一条序列，i,j 两点之间边的权值 e_{ij} 即 i,j 两序列之间的距离 $distance_{i,j}$，$distance_{i,j}$ 计算如下：

$$distance_{i,j}=\frac{1}{SP_{i,j}-SP_{\min}}+1 \qquad (1)$$

$SP_{i,j}$ 是序列 i 与序列 j 之间的联配得分(用动态规划算法)，$SP_{\min}$ 表示所有序列之间联配得分的最小值。

在以上的基础上我们可以简要介绍我们的算法，基本的算法包括以下3个主要部分：

1) 首先构造联配序列集合的完全图 G，用动态规划算法计算每两条序列之间的 SP 得分，由式(1)得到任意两个接点之间边的权值，同时得到序列的距离距阵。

2) 用 Kruskal 算法[12]求得一棵最小生成树(MST)。

3) 基于已经得到的 MST 使用本文的的序列组联配算法对序列进行联配，详细的过程如下：

Algorithm：**MstMsa**（Multiple Sequence Alignments Using MST）

输入：联配序列集合：$S=\{S_1,S_2,\cdots,S_n\}$。

输出：序列集合 S 的多序列联配。

Step 0：如果 $|S|\leqslant 1$，则结束。

Step 1：按(1)计算序列组 S 的距离矩阵，同时构造联配序列集合的完全图。

Step 2：求完全图的一棵最小生成树 $G=(V,E)$。

Step 3：把最小生成树 G 边的权值按递增的次序排列

Step 3.1：如果 $|V|\leqslant 2$，则转 Step 5。

Step 3.2：删除权值最大的一条边，最小生成树 G 分成为两棵最小生成树 $G_1=(V_1,E_1)$，$G_2=(V_2,E_2)$。

Step 4：令 $G=G_1$ 递归应用 Step 3。

令 $G=G_2$ 递归应用 Step 3。

Step 5：在 G_1 和 G_2 上应用本文的序列组联配算法。

现在通过一个例子来解释该算法，假设将要联配的序列集合 $S=\{S_1,S_2,S_3,S_4,S_5\}$。

Step 1：假设得到一个距离矩阵(表1)。

表1 距离矩阵

	S_1	S_2	S_3	S_4	S_5
S_1	0.00	0.94	0.91	0.97	0.89
S_2		0.00	0.95	0.90	0.96
S_3			0.00	1.00	0.99
S_4				0.00	0.98
S_5					0.00'

Step 2：用 Kruskal 算法得到最小生成树 MST（每一个节点中的数代表联配序列的序号）。

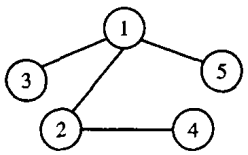

图1 MST G

经过 Step 3 和 Step 4我们可以得到下列的子树：

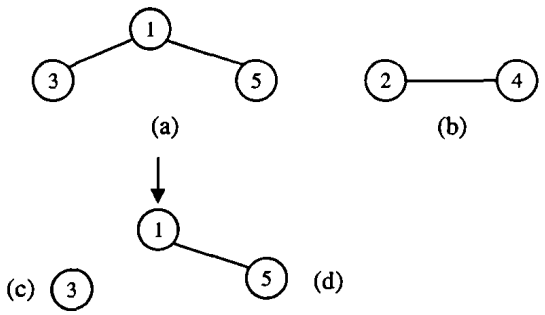

图2 The descendant MST

图1分成图2(a)和(b)，即：最小生成树 $G=(V,E)$分成 $G_a=(V_a,E_a)$，$G_b=(V_b,E_b)$，其中 $V_a=\{S_1,S_3,S_5\}$，$V_b=\{S_2,S_4\}$。由于 V_a 中的序列个数大于2，同理由 Step 3把 G_a 分成为 $G_c=(V_c,E_c)$，和 $G_d=(V_d,E_d)$，其中 $V_d=\{S_1,S_5\}$，$V_c=\{S_3\}$。最后得到3个序列组 $V_d=\{S_1,S_5\}$，$V_c=\{S_3\}$和 $V_b=\{S_2,S_4\}$。

Step 5：使用序列组联配算法得到最终的多序列联配，联配序列次序如下：

1)首先对 V_d 中的 S_2和 S_5进行联配，联配结果记为：Group (1)。

2)把 S_3与 Group 1进行联配，结果记为：Group (2)。

3)对 S_2和 S_4进行联配，结果记为 Group (3)。

4)最后联配 Group (2) 和 Group (3)得到最后结果。

5 实验结果

算法的实现是用 VC^{++}编程，在奔腾1.8GHz，内存256 M 的机器上运行的。在联配过程中对序列不采用权值。并且空隙的罚分使用仿射空隙罚分函数，所有的测试例子来自于标准数据集合 BAliBASE，为了准确地评价算法的性能，采用 SPS 得分标准去评价联配的质量。SPS 描述如下：

对于一个测试例子的已知联配，假设该联配包含 N 个序列和 M 列，令联配的第 i 列为：$A_{i1},A_{i2},\cdots,A_{iN}$。对于每一个残基 A_{ij}和 A_{ik}，定义 P_{ijk}，如果 A_{ij}和 A_{ik}在参考序列中也配对了，则 $P_{ijk}=1$，否则 $P_{ijk}=0$，第 i 列的得分 S_i 定义为：

$$S_i=\sum_{j=1,j\neq k}^{n}\sum_{k=1}^{N}P_{ijk}$$

整个联配的 SPS 得分为：

$$SPS=\sum_{i=1}^{M}S_i/\sum_{i=1}^{M_r}S_{ri}$$

其中 M_r 是参考联配中的列数，S_{ri}是参考联配第 i 列的得分。SPS 最大为1最小为0，越大说明联配的质量越好。

表2给出了本文算法和 ClustalX 的 SPS 得分比较。在该表中，两个算法都采用 ClustalX 的默认参数，开放罚分是10，扩展罚分是0.2，替代矩阵是 PAM-250。从表可知，在大多数测试例子中 MstMsa 联配质量好于 ClusalX 的联配结果。既然 ClustalX 是目前最有名的联配软件包之一，实验的结果再次证明了新算法的优越性和有效性。

表2 The score of three algorithms

Test case	Nse	MinL	MaxL	MstMsa SPS	ClustalX SPS
1csp_ref1	5	66	70	0.952	0.908
2cba_ref1	5	237	259	0.644	0.487
1plc_ref1	5	88	99	0.893	0.876
1aho_ref1	5	61	65	0.855	0.757
1amk_ref1	5	242	254	0.969	0.893
1pysA_ref4	4	234	785	0.652	0.482
kinase1_ref4	7	289	481	0.706	0.582
1pfc_ref4	10	111	362	0.383	0.249
1ivy_ref5	7	406	441	0.684	0.734

Nseq 是测试例子中序列的条数，MinL 和 MaxL 分别是测试例子中最短和最长序列的长度。

结论 综上所述，本文从图论的角度，提出了一种基于最小生成树的启发式联配算法，实验结果表明该算法具有更高的精确度，同时也表明该方法是一种新的并且有前途的方法，我们未来的工作是对算法实现的改进和更加广泛的测试，以及与分而治之算法，遗传算法等相关方法的结合研究。

参 考 文 献

1 JIANG Tao, Kearney P, Li Ming. Some Open Problems in Computational Molecular Biology[J]. J of Algorithms, 2003, 34: 194～201

2 Carrillo H, Lipman D J. The multiple sequence alignment problems in biology. SIAM J. Appl. Math., 1998, 48: 1073～1082

3 Chan S, Wang A, Chu D. A survey of multiple sequence comparison methods. Bull. Math. Bio., 1992, 54: 563～360

4 Notredame C. Recent progress in multiple sequence alignment: a survey. Pharmacogenomics, 2002, 3(1): 131～44

5 Feng D F, Doolittle R F. Progressive sequence alignment as a prerequisite to correct phylogenetic trees. J. Mol. Evol., 1987, 25(4): 351～60

6 Khuller S. Journal of Algorithms, 2000, 34: 194

7 Korostensky C R. Algorithms for Building Multiple Sequence Alignments and Evolutionary Tree: [Dissertation]. Swiss Federal Institute of Technology, 2000

8 Giegerich R, Wheeler D. Pairwise SequenceAlignment. http://www.techfak.uni-bielefeld.de/bed/Curric/PruAli/Prwali.html, 1996

9 Thompson J D, Plewniak F, Poch O. BAliBASE: a benchmark alignment database for the evaluation of multiple alignment programs. Bioinformatics, 1999, 15(1): 87～88

10 Lipman D J, Altschul S F, Kececioglu J D. A tool for multiple sequence alignment. In: Proc. of the National Academy of Sciences of the United States of America, 1989, 86: 4412～4415

11 Altschul S F, Gish W, Miller W, Myers E W, Lipman D J. Basic local alignment search tool. Journal of Molecular Biology, 1990, 215: 403～410

12 Rosen K H. Discrete Mathematics and Its Application. Copyright 1998 by The McGraw-Hill Companies, Inc. Jointly published by China Machine Press/Mc Graw-Hill, 1999

13 Notredame, Thompson J D, Plewniak F, Poch O. A comprehensive comparison of multiple sequence alignment programs. Nucleic Acids Res., 1999, 27: 2682～2690