

文章编号: 1673-1689(2010)04-0634-07

# 基于剖面隐马尔可夫模型的多序列比对

李成渊<sup>1</sup>, 龙海侠<sup>2</sup>, 孙俊<sup>1</sup>, 须文波<sup>\*1</sup>

(1. 江南大学 信息工程学院, 江苏 无锡 214122; 2. 江南大学 教育学院, 江苏 无锡 214122)

**摘要:** 多序列比对被称为 NP 完全问题, 是生物信息中最基本的问题之一。目前, 广泛使用剖面隐马尔可夫模型解决多序列比对问题。作者在粒子群优化算法的基础上, 提出了将量子粒子群优化算法用于剖面隐马尔可夫模型的训练过程, 进而构建了一种基于剖面隐马尔可夫模型和量子粒子群优化算法的多序列比对算法。从核酸序列和 BaliBASE 比对数据库中选取了一些比对例子进行了模拟实验, 并与其他算法进行了比较, 结果表明, 所提出的算法能在有限的时间内不仅能找到理想的隐马尔可夫模型, 而且能得到最优的比对结果。

**关键词:** 多序列比对; 剖面隐马尔可夫模型; 量子粒子群优化算法

中图分类号: Q 811.4

文献标识码: A

## Multiple Sequence Alignment Based On the Profile Hidden Markov Model

LI Cheng-yuan<sup>1</sup>, LONG Hai-xia<sup>2</sup>, SUN Jun<sup>1</sup>, XU Wen-bo<sup>\*1</sup>

(1. School of Information Technology, Jiangnan University, Wuxi 214122, China; 2. School of Education, Jiangnan University, Wuxi 214122, China)

**Abstract:** Multiple sequence alignment (MSA), known as NP-complete problem, is one of the basic problems in computational biology. At present Profile Hidden Markov Model (HMM) was widely used in multiple sequence alignment. This manuscript presented the quantum-behaved particle swarm optimization (QPSO) which was based on particle swarm optimization. The proposed algorithm was used to optimize the profile HMM. Furthermore, an integration algorithm based on the profile HMM and QPSO for the MSA was constructed. Then the approach was evaluated by a set of standard instances which are chosen from nucleotides sequences and the benchmark alignment database, name as BaliBASE. Finally our results are compared with other algorithms. The result shown that the proposed algorithm not only finds out the perfect profile HMM, but also obtains the optimal alignment of multiple sequence.

**Key words:** multiple sequence alignment, profile hidden markov model, quantum-behaved particle swarm optimization

---

核苷酸或氨基酸的多序列比对或联配是生物信息学中最重要、最具有挑战性的任务之一。多序

收稿日期: 2009-08-07

作者简介: 李成渊(1980-), 男, 江苏无锡人, 生物信息学博士研究生。Email: lichengyuaning@gmail.com

\* 通信作者: 须文波(1946-), 男, 江苏无锡人, 教授, 博士生导师, 主要从事生物信息学方面的研究。

Email: xwb\_sytu@hotmail.com

列比对问题是一个将不等长的多个序列通过插入空位变成等长的过程, 这些位置上的空位代表着相比对的序列从共同的祖先通过插入/删除操作的进化过程。利用多序列比对算法得到的最优比对, 可用于找出蛋白质家族的模体 (motifs) 或保守区域 (conserved domains), 可用于预测蛋白质的结构和功能, 也可用于进行系统发育的分析<sup>[1-2]</sup>。

目前主要有下列 3 种策略用于多序列比对。第一种策略是“渐进比对”策略<sup>[3-4]</sup>, 其基本思想是: 迭代地利用两序列动态规划算法, 先由两条序列的比对开始, 逐渐添加新序列, 直到所有序列都加入为止。第二种策略是使用随机优化算法, 如模拟退火算法(SA)<sup>[5]</sup>, 遗传算法(GA)<sup>[6]</sup>; 第三种策略基于概率模型的隐马尔可夫模型<sup>[7-8]</sup>。

本研究中使用第三种策略。在多序列比对的过程中, 隐马尔可夫模型主要解决 3 个问题: 一是得分问题, 二是联配问题, 三是训练问题。将得分问题用来评估模型的性能, 联配问题用来实现多序列的比对, 训练问题用来优化模型的参数。最常用的训练隐马尔可夫模型模型的方法是基于统计和重估的方法, 比如 Baum-Welch 算法。但是 Baum-Welch 算法是一个局部最优算法, 使用此算法得到的最终比对结果通常远离全局最优。最近还出现了粒子群算法(PSO), 此算法也是一个局部最优算法。

为了克服 Baum-Welch 算法和 PSO 算法的缺点, 我们使用量子粒子群优化算法(QPSO)<sup>[9-10]</sup>来训练隐马尔可夫模型, 不仅参数个数少, 随机性强, 并且能覆盖所有解空间, 保证算法的全局收敛。

### 1 剖面隐马尔可夫模型的拓扑结构

使用的是一种标准的剖面隐马尔可夫模型的拓扑结构用于多序列的比对, 最初由 Krogh et al. (1994)<sup>[7]</sup>提出的, 见图 1。该模型包含了一系列的状态( $S_1, S_2, \dots, S_n$ ) 这些状态被分成 3 组, 分别是匹配状态(M)、插入状态(I)和缺失状态(D)。该模型是包含这 3 种状态重复集的简单的从左至右的结构。为了研究的方便, 再引入了两个额外的状态: 开始状态(begin)和结束状态(end)。状态之间通过转移概率  $a_{ij}$  相联系, 转移概率具有下面的性质:

$a_{ij} \geq 0, 1 \leq i, j \leq n$  并且  $\sum_{j=1}^n a_{ij} = 1, 1 \leq i \leq n_0$ 。  
 一个匹配状态或一个插入状态按照一定的符号发出概率  $b_j(k)$  发出一个可见符号  $v_k$ , 符号发出概率具有下面的性质:  $b_j(k) \geq 0, 1 \leq j \leq n, 1 \leq k \leq M$  并且  $\sum_{k=1}^M b_j(k) = 1, 1 \leq j \leq n$ , 这里  $M$  为每个符号对应的观察符号数目。缺失状态、开始状态、结束状态都不发出任何符号。

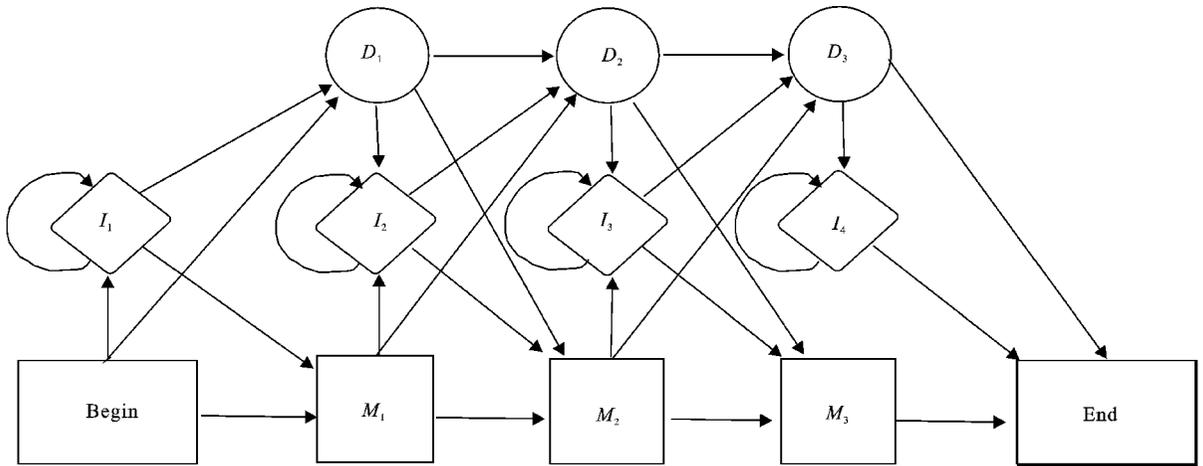


图 1 用于多序列比对的隐马尔可夫模型  
 Fig. 1 An example of a simple HMM of length 3 for MSA

当使用图 1 所示的隐马尔可夫模型进行多序列比对时, 每条序列从开始到结束通过这些状态穿越模型, 在这些待比对序列中进行空位字符‘-’的插入和删除操作, 得到一个多序列比对结果的矩阵 A

$= (a_{ij})_{m \times n}$ , 其中  $a_{ij} \in alph\_set \cup \{-\}$ 。矩阵 A 中的每一列为一个位点上的比对, 矩阵 A 的第 i 行对应参与比对的第 i 个序列, 序列中非空字符的先后顺序在比对中保持不变。

## 2 基于剖面隐马尔可夫模型和QPSO的多序列比对

用量子粒子群优化算法训练剖面隐马尔可夫模型时,每一个粒子代表一个隐马尔可夫模型,通过不断的更新粒子的位置来优化隐马尔可夫模型。在训练中保持模型的长度不变,仅仅优化模型的参数:转移概率和符号发出概率。对图1所示的隐马尔可夫模型的拓扑结构,我们取待比对序列的平均值  $m$  为模型的长度,不考虑初始状态和结束状态,则模型的状态数为  $3m+1$ , 状态转移概率参数为  $3(3m+1)$  个; 设字符集大小为  $|A|$ , 共有  $(2m+1)|A|$  个符号发出概率。所以每个粒子是维数为  $9m+3+(2m+1)|A|$  的一个实数编码串。所以DNA模型的参数个数是  $17m+7$ , 蛋白质模型的参数个数是  $49m+23$ 。根据转移概率和符号发出概率的性质,在对粒子对应的隐马尔可夫模型模型进行评价前,需要先对隐马尔可夫模型中的状态转移概率和符号发出概率进行归一化,以满足  $3m+1$  个转移概率归一化约束方程和  $2m+1$  个符号发出概率归一化方程。

根据粒子群算法训练的结果,得到全局最优的粒子对应的隐马尔可夫模型,接下来用此模型使用Viterbi算法进行序列的比对,得到最优的比对结果,并用基于SP(sum of pairs)打分系统的目标函数评估比对的结果。

在整个算法的过程中,根据剖面隐马尔可夫模型的拓扑结构,所使用的转移概率的形式见表1所示。

表1 转移概率

Tab.1 The transition probability

$M$ 的转移	$I$ 的转移	$D$ 的转移
$M_i \rightarrow M_{i+1}$	$I_i \rightarrow M_i$	$D_i \rightarrow M_{i+1}$
$M_i \rightarrow I_{i+1}$	$I_i \rightarrow I_i$	$D_i \rightarrow I_{i+1}$
$M_i \rightarrow D_{i+1}$	$I_i \rightarrow D_i$	$D_i \rightarrow D_{i+1}$

### 2.1 模型的训练问题

**2.1.1 量子粒子群优化算法(QPSO)** PSO<sup>[11]</sup> 是基于种群的进化搜索技术,但是所有基本和改进的PSO算法不能保证算法的全局收敛<sup>[12]</sup>。因为PSO的进化方程式使所有粒子在一个有限的样本空间中搜索。根据粒子群的基本收敛性质,受量子物理基本理论的启发,Sun等人提出了QPSO(Quantum-behaved Particle Swarm Optimization<sup>[9-10]</sup>算法是对整个PSO算法进化搜索策略的改变,进化方

程中不需要速度向量,形式更简单,参数更少且更容易控制。

在QPSO算法中,粒子按照下面3个公式进行更新:

$$p = (\varphi_1 \times P_{id} + \varphi_2 \times P_{gd}) / (\varphi_1 + \varphi_2) \quad (1)$$

$$\text{其中 } \varphi_1 = \text{rand}(0, 1); \varphi_2 = \text{rand}(0, 1)。$$

$$\text{mbest} = \frac{1}{M} \sum_{i=1}^M P_i = \left[ \frac{1}{M} P_{i1}, \dots, \frac{1}{M} P_{id} \right] \quad (2)$$

其中  $M$  为粒子的个数。

$$x_{id} = P_{id} + \alpha^* | \text{mbest}_d - x_{id} | * \ln(1/u) \quad (3)$$

$$u = \text{rand}$$

**2.1.2 评估训练算法的质量** 在粒子群优化算法中,需要评估每个粒子所代表的模型的质量,使用的评估函数为:

$$\text{Log\_odd}(O, \lambda) = - \frac{1}{N} \sum_{i=1}^N \frac{\log_2 P(O_s | \lambda)}{l_i} \quad (4)$$

这里  $O = \{O_1, O_2, \dots, O_M\}$  是给定的待比对序列的集合,序列的个数为  $M$  个。 $l_s$  是序列  $O_s$  的长度。Log\_odd的值越大,说明使用量子粒子群优化算法训练得到的隐马尔可夫模型,模型的参数是最优的,模型的稳定性和可靠性都较好。

### 2.2 模型的联配问题

**2.2.1 序列联配的过程** 给定3条待比对序列: O(1): AGGCT; O(2): GAACTGTA; O(3): AGCCTTA。按照下面的两个步骤可以得到序列比对的结果:

1) 根据Viterbi算法和图1所示的隐马尔可夫模型拓扑结构找出每条序列所对应的状态序列,如图2所示,每条序列的氨基酸碱基对应一个匹配状态或一个插入状态。

O\*(1): M M I M I  
O(1): A G G C T  
O\*(2): M I M I I M M I  
O(2): G A A C T G T A  
O\*(3): M M I I M I  
O(3): A G C C T T

图2 每条序列所对应的状态序列

Fig.2 Each sequence corresponding to the state sequence

由图3生成的状态序列,我们可以进行空位字符“ $\rightarrow$ ”的插入操作,见图3。所有的序列中与匹配状态  $M$  相对应的氨基酸碱基是比对的,这些氨基酸碱基位于同一列;与插入状态  $I_k$  相对应的氨基酸碱基位于  $M_{k-1}$  或  $D_{k-1}$  之后,  $M_k$  或  $D_k$  之前。

```
O*(1): M D M I D M I D
O(1):  A _ G G _ C T _
O*(2): M I M I I M M I
O(2):  G A A C T G T A
O*(3): M D M I I M I D
O(3):  A _ G C C T T _
```

图 3 空位字符 '-' 的插入

Fig.3 Insert the gap characters '-'

由步骤(1)和步骤(2), 得到联配后的序列为图 4 所示。

```
O(1): A _ G G _ C T _
O(2): G A A C T G T A
O(3): A _ G C C T T _
```

图 4 联配后的序列

Fig 4 Aligned sequences

2.2.2 评估比对序列的质量在利用 Viterrbi 算法获得的路径进行比对后, 需要通过基于 SP( sum of pairs) 打分系统的目标函数<sup>[13]</sup> 对对比结果进行评估。我们使用下面标准的 sum-of-pairs 打分函数:

$$SOP = \sum_{i=1}^{n-1} \sum_{j=i+1}^n D(l_i, l_j) \quad (5)$$

这里,  $l_i$  是已比对的序列,  $D$  为距离矩阵。

为了避免在比对过程中空位的积聚, 我们从 SOP 分数中推演出仿射几何学的空位代价, 对于比对结果中一条序列的空位代价按照下面的公式进行计算:

$$\text{Gap cost} = \text{GOP} + n \times \text{GEP} \quad (6)$$

这里 GOP 表示第一个开口空位的固定罚分, GEP 表示对于扩展的空位的罚分,  $n$  为一条序列中空位的个数。对于已比对的每条序列的空位都要计算相应的空位代价。多序列比对结果的 SOP 的分值减去空位代价的总和, 即为 SOP 的分值。

### 3 实验结果

#### 3.1 实验数据

3.1.1 数据集 1(模拟的核苷酸序列) 使用软件 Rose<sup>[8]</sup> 软件产生如下的核酸序列: “low-short” “low-long”, “high-short”, 和 “high-long”。

3.1.2 数据集 2(Pfam 数据库中的 3 个蛋白质家族) 3 个蛋白质家族分别为 G5, CagY\_M and Interferon, 它们来自 Pfam 数据库<sup>[14]</sup>, 见表 2。为了避免隐马尔可夫模型模型中的过拟合, 分别把 3 个蛋白质家族分成训练集和测试集。

表 2 蛋白质家族  $N$ : 序列的个数, LSEQ: 数列的平均,  $T$ : 训练集的大小

Tab.2 Protein families  $N$ : the number of sequences LSEQ: the average of series  $T$ : The size of training sets

蛋白质家族	序列数量	数列平均	训练集大小规模
G5	277	79	150
CagY_M	549	31	150
IstB	738	147	150

#### 3.2 实验设置

实验中, 我们使用均匀分布初始化所有的种群; 分别使用了 Baum-Welch(BW)、Particle Swarm Optimization (PSO)、Quantum-behaved Particle Swarm Optimization(QPSO) 3 种算法训练隐马尔可夫模型, 用 Log-odd 目标函数来评估 4 种训练算法的性能, 并且使用 SOP 打分系统的目标函数来评估对比效果。每种算法分别迭代 1 000 次, 共进行了 20 次模拟实验。

对于 PSO 算法, 使用如下的参数: 种群数= 20; 惯性权重 ( $\omega$ ) 从 1.0 到 0.5 线性减少;  $c_1 = c_2 = 2.0$  最大速度 ( $v_{max} = 1.0$ )

对于 QPSO 算法使用如下的参数: 种群大小= 20; 收缩扩张系数, 从 1.0 到 0.5 线性减少; 并且在 SOP 打分函数中, 核酸和蛋白质分别使用下面的打分矩阵、空位开放和空位扩充。

1) 对于核酸序列, 使用 ClustalW 1.81 的 “sw gap” 置换分数表, 空位开放和空位扩充分别为 15、7。

2) 对于蛋白质序列, 使用 BLOSUM 62 打分矩阵, 空位开放和空位扩充分别为 11、2<sup>[9]</sup>。

#### 3.3 实验结果

表 3 给出了核酸序列的结果, 这里是以 Log-odd score 作为目标函数, 用来评估训练隐马尔可夫模型的质量, 从表中可以看出, QPSO 取得了最好的平均值, PSO 算法的结果和 BW 算法的结果相当, 在 Low-long 序列和 High-long 序列上的结果不如 BW 得出的结果好, 可以说明, PSO 算法随着序列个数的增加效果不如 BW 的效果好。

表 4 也给出了核酸序列的结果, 这里是以 SOP score 作为目标函数, 用来评估核酸序列的比对的效果。从表 4 可以看出, QPSO 优于其他所有的算法, 其次为 CW 工具得出的比对结果优于 BW 和 PSO 算法得出的结果, PSO 算法也是一种局部最优算法, 对 BW 算法没有多大的改进。

表3 隐马尔可夫模型得到的核酸序列的 Log-odd scores 平均值和方差

Tab. 3 HMM log-odds scores  $\pm$  standard error of Nucleotide sequences

核酸序列	BW 算法	PSO 算法	QPSO 算法
Low - short	394. 6	418. 4 $\pm$ 7. 35	448. 8 $\pm$ 9. 68
Low - long	226. 5	215. 8 $\pm$ 8. 26	310. 7 $\pm$ 4. 68
High - short	186. 3	195. 6 $\pm$ 2. 31	253. 8 $\pm$ 3. 45
High - long	280. 8	256. 4 $\pm$ 8. 52	345. 8 $\pm$ 9. 73

表4 核酸序列比对结果的 SOP 平均值和方差

Tab. 4 SOP scores for the final alignments of Nucleotide sequences

核酸序列	CW 算法	BW 算法	PSO 算法	QPSO 算法
Low - short	2 514. 9	2 024. 7	2 154. 3 $\pm$ 25. 36	2 638. 4 $\pm$ 20. 364
Low - long	8 760. 7	8 623. 9	8 430. 8 $\pm$ 15. 015 4	9 873. 3 $\pm$ 18. 462
High - short	4 316. 5	4 270. 2	4 294. 7 $\pm$ 13. 564 3	4 351. 9 $\pm$ 13. 375
High - long	7 781. 7	7 418. 2	7 538. 5 $\pm$ 22. 864 5	7 848. 8 $\pm$ 29. 688 1

表5 隐马尔可夫模型得到的蛋白质序列训练集的 Log-odd scores 平均值和方差

Tab. 5 HMM log-odds scores  $\pm$  standard error for the training sets of three protein families

蛋白质家族	BW 算法	PSO 算法	QPSO 算法
G5	103. 146	101. 450 0 $\pm$ 1. 89	154. 94 $\pm$ 1. 06
CagY_M	11. 178	20. 090 $\pm$ 1. 14	28. 255 $\pm$ 0. 92
Interferon	158. 314	141. 736 $\pm$ 2. 41	179. 549 $\pm$ 1. 43

表6 蛋白质序列训练集的最优的 SOP scores 的平均值和方差

Tab. 6 SOP scores for the final alignments of the training sets of three protein families

蛋白质家族	CW 算法	BW 算法	PSO 算法	QPSO 算法
G5	189	192	176 $\pm$ 113. 1	229 $\pm$ 63. 7
CagY_M	- 142	- 138	- 120 $\pm$ 131. 5	- 106 $\pm$ 103. 37
Interferon	3 226	3 294	3 772 $\pm$ 101. 4	4 136 $\pm$ 98. 5

表7 隐马尔可夫模型得到的蛋白质序列测试集的 Log-odd scores 平均值和方差

Tab. 7 HMM log-odds scores  $\pm$  standard error for the validation sets of three protein families

蛋白质家族	BW 算法	PSO 算法	QPSO 算法
G5	78. 357	141. 450 0 $\pm$ 2. 42	154. 94 $\pm$ 1. 12
CagY_M	32. 832	20. 090 $\pm$ 2. 13	20. 255 $\pm$ 1. 08
Interferon	102. 652	95. 736 $\pm$ 1. 54	179. 549 $\pm$ 1. 28

表8 蛋白质序列测试集的最优的 SOP scores 的平均值和方差

Tab. 8 SOP scores for the final alignments of the validation sets of three protein families

蛋白质家族	CW 算法	BW 算法	PSO 算法	QPSO 算法
G5	- 836	- 803	- 796 $\pm$ 241. 2	- 914 $\pm$ 85. 6
CagY_M	- 795	- 658	- 725 $\pm$ 198. 4	- 804 $\pm$ 105. 7
Interferon	2 986	2 352	2 176 $\pm$ 341. 8	3 215 $\pm$ 127. 1

表5~8概述了3个蛋白质家族的隐马尔可夫模型训练和序列比对的结果。表5和表6分别给出了训练集的最优的 Log-odd scores 和 SOP scores 的平均值和方差,表7和表8分别给出了评估集的最优的 Log-odd scores 和 SOP scores 的平均值和方差。从4个表中,可以看出,对于蛋白质家族,无论是在训练集还是评估集上,QPSO 算法在模型的训练上和比对的效果上都优于 BW 和 PSO 算法。

图 5、6、7 刻画了以 Log-odd score 作为目标函数, 算法分别运行 20 次得到的核酸序列和蛋白质序列的平均值的收敛过程。

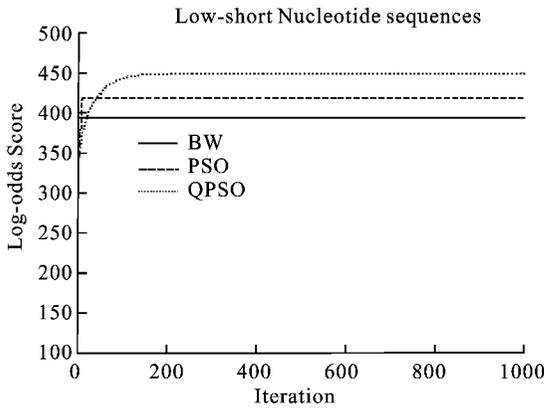


图 5 Low-short 核酸序列的 Log-odd 平均分  
Fig. 5 Mean Log-odds scores for Low-short Nucleotide

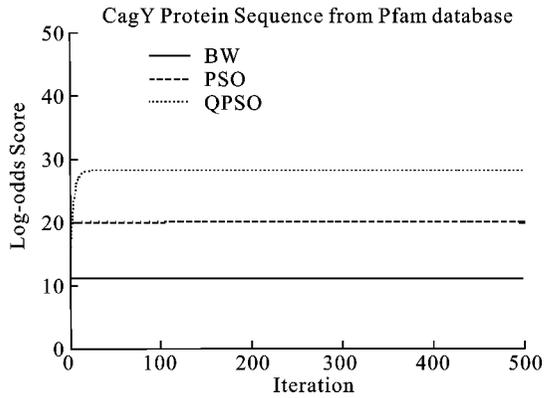


图 6 Cag\_Y 蛋白质序列的 Log-odd 平均分  
Fig. 6 Mean Log-odds scores for training sets of CagY\_M protein family

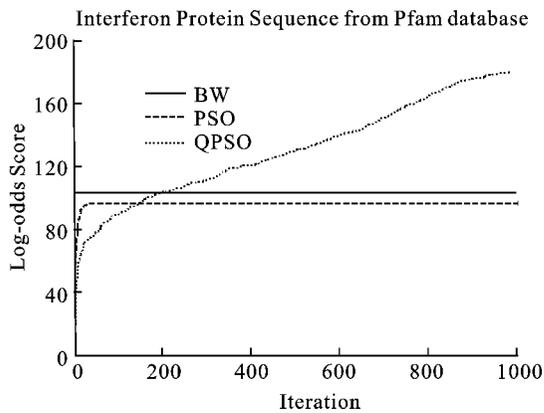


图 7 Interferon 蛋白质序列的 Log-odd 平均分  
Fig. 7 Mean Log-odds scores for validation sets of Interferon protein family

从图中看出, QPSO 性能最好, 在迭代的最后得到的 Log-odd score 的分值也最好, 除了图 9, PSO 算法的性能优于 BW 算法。从收敛速度上看,

BW 和 PSO 的收敛速度最快, 其次为 QPSO 算法, 但是 QPSO 算法在迭代进行到一半的时候, Log-odd score 的值就已经超过 PSO 算法和 BW 算法, 并且在整个运行过程中, QPSO 都在不断的提高自身的性能, 而 PSO 算法和 BW 算法已经早早地收敛了。

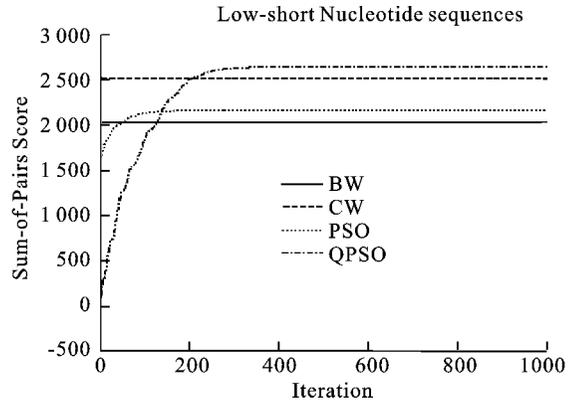


图 8 Low-short 核酸序列的 SOP 平均分  
Fig. 8 Mean SOP scores for Low-short Nucleotide

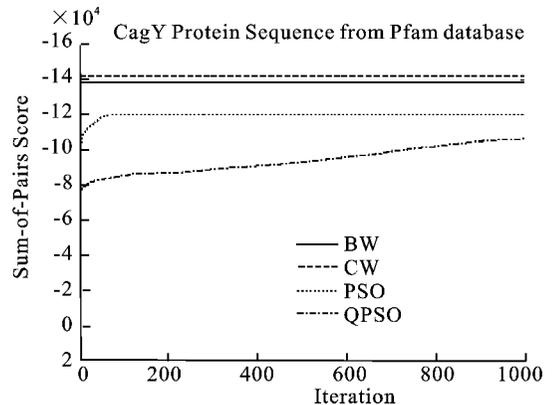


图 9 Cag\_Y 蛋白质序列的 SOP 平均分  
Fig. 9 Mean SOP scores for training sets of CagY\_M protein family

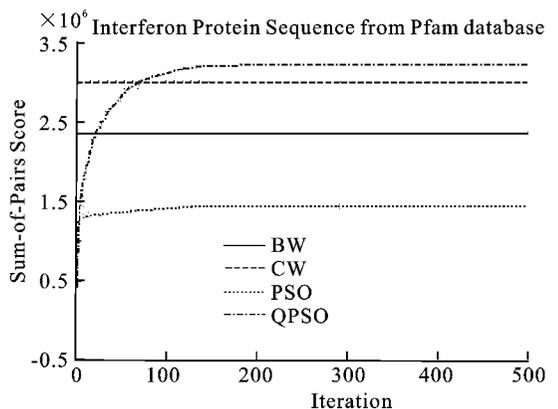


图 10 Interferon 蛋白质序列的 SOP 平均分  
Fig. 10 Mean SOP scores for validation sets of Interferon protein family

图 8、9、10 表明了以 SOP score 作为目标函数, 算法分别运行 20 次得到的核酸序列和蛋白质序列的平均值的收敛过程。收敛情况与上述情况相同。

## 4 结 语

提出了使用 QPSO 算法来训练剖面隐马尔可夫模型的参数, 进行多序列的比对。从实验结果可以看出, 以 Log-odd score 作为目标函数, QPSO 算法相比 BW 算法和 PSO 算法, 是一种非常有效的训练 HMM 模型模型的方法。并且从实验结果还可以得出, 以 SOP 作为目标函数, QPSO 比其他所有

的方法而言, 能产生较好的比对结果。这是因为 QPSO 算法是一种全局收敛算法, 并且需要调整的参数也比较少。

在计算时间上, QPSO 消耗的时间和 PSO 消耗的时间相当, 远远大于 BW 算法的运行时间。平均来说, QPSO 和 PSO 的运行时间为 6 小时, BW 的运行时间为 5 小时。并且, 随着序列个数和序列长度的增加, QPSO 消耗的时间也随着增大。

在进一步的研究工作中, 我们将进一步改善 QPSO 算法, 来提高 HMM 的性能, 减少序列比对的时间。

## 参考文献(References):

- [ 1 ] GUAN Wei-hong, X Z-Y, ZHU Ping. Nonlinear prediction analysis of properties in protein sequences(I) [J]. **Journal of Food Science and Biotechnology**, 2008, 27(1): 71- 75.
- [ 2 ] GUAN Wei-hong, X. Z.-Y., ZHU Ping. Nonlinear prediction analysis of properties in protein sequences(II) [J]. **Journal of Food Science and Biotechnology**, 2008, 27(2): 103- 105.
- [ 3 ] Thompson J D, Higgins D G, Gibson T J. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice [J]. **Nucl Acids Res**, 1994, 22(22): 4673- 4680.
- [ 4 ] Feng D F, Doolittle R. Progressive sequence alignment as a prerequisite to correct phylogenetic trees [J]. **Journal of Molecular Evolution**, 1987, 25(4): 351- 360.
- [ 5 ] Kim J, Pramanik S, Chung M J. Multiple sequence alignment using simulated annealing [J]. **Comput Appl Biosci**, 1994, 10(4): 419- 426.
- [ 6 ] E Jung Lee, S F S, Chen-Chia Chuang, et al. Genetic algorithm with ant colony optimization (GA-ACO) for multiple sequence alignment [J]. **Applied Soft Computing**, 2008, 8(1): 15- 17.
- [ 7 ] Mamitsuka H. Finding the biologically optimal alignment of multiple sequences [J]. **Artif Intell Med**, 2005, 35(1-2): 9- 18.
- [ 8 ] Loytynoja A, Milinkovitch M C. A hidden Markov model for progressive multiple alignment [J]. **Bioinformatics**, 2003, 19(12): 1505- 1513.
- [ 9 ] Jun S, Wenbo X, Bin F. In A global search strategy of quantum-behaved particle swarm optimization [J]. **Cybernetics and Intelligent Systems**, 2004, 321: 325- 331.
- [ 10 ] Jun S, Bin F, Wenbo X. In Particle swarm optimization with particles having quantum behavior [J]. **Evolutionary Computation**, 2004, 321: 325- 331.
- [ 11 ] Kennedy J, Eberhart R. In Particle swarm optimization [J]. **Neural Networks**, 1995, 1994: 1942- 1948.
- [ 12 ] Clerc M, Kennedy J. The particle swarm- explosion, stability, and convergence in a multidimensional complex space [J]. **Evolutionary Computation**, 2002, 6(1): 58- 73.
- [ 13 ] Thompson J D, Plewniak F, Poch O. A comprehensive comparison of multiple sequence alignment programs [J]. **Nucleic Acids Res**, 1999, 27(13): 2682- 2690.
- [ 14 ] Sonnhammer E L, Eddy S R, Durbin R. Pfam: a comprehensive database of protein domain families based on seed alignments [J]. **Proteins**, 1997, 28(3): 405- 420.

(责任编辑: 李春丽)