

项目一：MSA

人工智能 CS410 2021年秋季

姓名：李子龙 学号：518070910095 日期：2021 年 10 月 12 日

目录

1	题目	2
1.1	Topic	2
1.2	Requirements	2
1.3	Rules	2
2	动态规划算法	2

1 题目

1.1 Topic

Implement three algorithms to solve multiple sequence alignment (MSA) problems.

1.2 Requirements

- (1) Implement dynamic programming (DP) algorithm to find the optimal solution.
- (2) Implement A-star (A*) algorithm to find the optimal solution.
- (3) Implement genetic algorithm to find the optimal/suboptimal solution.

1.3 Rules

表 1: Cost Matrix			
	Match $\alpha(p, p)$	Mismatch $\alpha(p, q)$	Gap δ
Cost	0	3	2

The table above shows the pairwise cost matrix. For multiple sequence alignment, the cost should be calculated in a cycle pairwise manner. Note that GAP-GAP is a match and should be considered as 0 cost. For every query, find the best alignment(s) in the database with the lowest cost.

2 动态规划算法

在算法与复杂性课程^[1]里, 已经提到了双序列比对的动态规划算法, 如图 1 所示, 双序列比对对于一个状态只需要考虑三个临近状态的转移, 分别是对齐 α , 间隔 δ_x 、 δ_y , 转换行动如表 2 所示。对于每一个状态, 都需要考虑经过哪一条路径消耗最小, 于是就有了如算法 1 的动态规划状态转移方程。

Algorithm 1: 双序列比对动态规划 MSA

Input: $x_1x_2 \cdots x_m, y_1y_2 \cdots y_n, \alpha, \delta$

Output:

```

1 for  $i \leftarrow 0$  to  $m$  do  $M[i, 0] = i\delta$ ;
2 for  $j \leftarrow 0$  to  $n$  do  $M[0, j] = j\delta$ ;
3 for  $i \leftarrow 1$  to  $m$  do
4   for  $j \leftarrow 1$  to  $n$  do
5      $M[i, j] = \min(\alpha[x_i, y_j] + M[i-1, j-1], \delta + M[i-1, j], \delta + M[i, j-1]);$ 
6 return  $M[m, n]$ ;
```

表 2: 双序列行动坐标变换表

	i	j
α	+1	+1
δ_x	0	+1
δ_y	+1	0

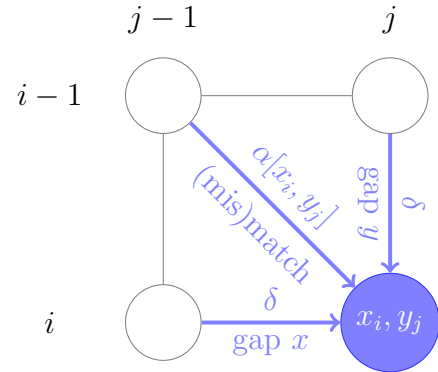


图 1: 双序列比对

对于三序列比对, 情况就复杂地多, 需要同时考虑七条路径。

表 3: 三序列行动坐标变换表

	i	j	k
$\delta_x \delta_y \alpha_z$	0	0	1
$\delta_x \alpha_y \delta_z$	0	1	0
$\delta_x \alpha_y \alpha_z$	0	1	1
$\alpha_x \delta_y \delta_z$	1	0	0
$\alpha_x \delta_y \alpha_z$	1	0	1
$\alpha_x \alpha_y \delta_z$	1	1	0
$\alpha_x \alpha_y \alpha_z$	1	1	1

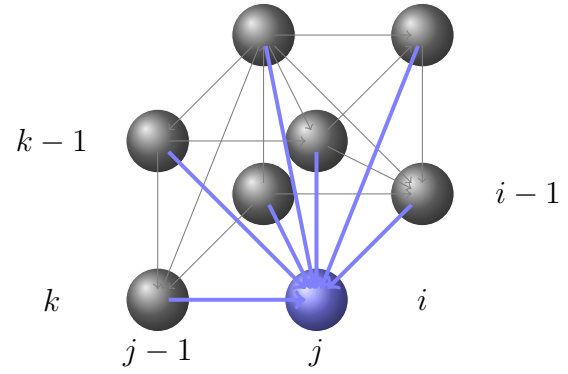


图 2: 三序列比对

可以统一化为多序列比对问题。对于 l 条序列比对, 其行动转换方法可以被表示为二进制从 $(\underbrace{0 \cdots 01}_{l \text{ digits}})_2$ 到 $(\underbrace{1 \cdots 11}_{l \text{ digits}})_2$ 内所有的数, 计算损耗使用上三角成对比较, 规则统一为

$$\text{compare} = \begin{cases} 0, & (-, -) \parallel (p, p) \\ 2, & (p, -) \parallel (-, q) \\ 3, & (p, q) \end{cases}$$

并在确定每一次行动后记录路径, 最后回溯路径到原点。

参考文献

- [1] XIAOFENG G. Algorithm & complexity class lab 06[EB/OL]. 2021. <https://github.com/LogCreative/AlgAndComplexity/blob/master/Lab06/Code-SequenceAlignment.cpp>.