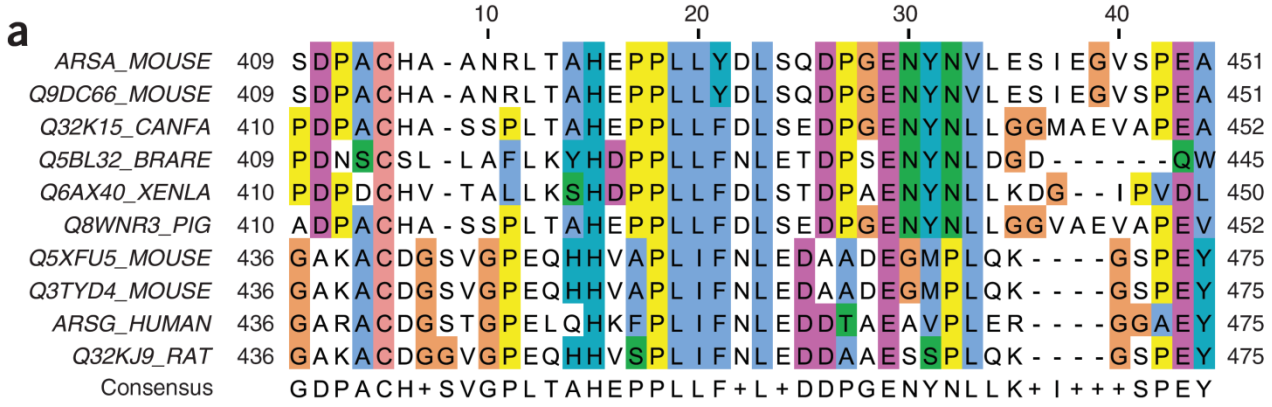
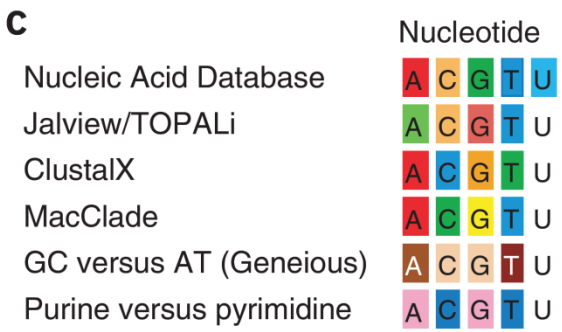
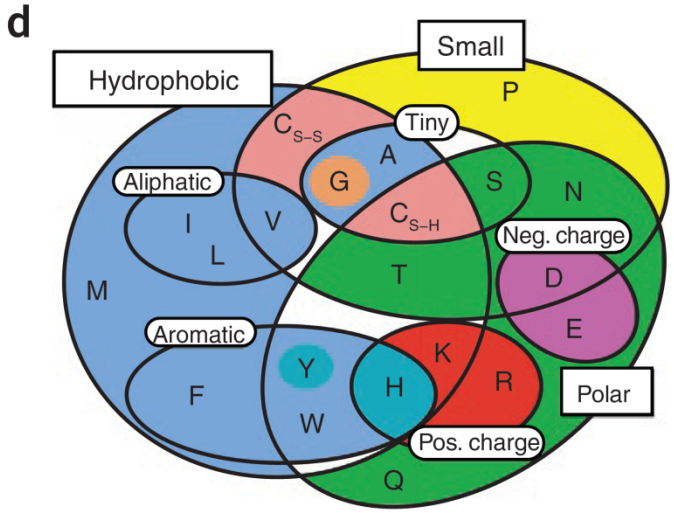
Chapter-03. 生物分子序列比对







本章内容

- 3.1
- 3.2
- 3.3
- 3.4
- 3.5
- 3.6

第1节：

一个有趣的运筹学问题：走遍中国

你我周游全国，从北京出发，要遍游我国34个省级中心，最后回到北京。请设计一条总行程最短的路线。

穷举法

R计算：33! =Factorial(33)

8.68×10^{36} ???

Dynamic programming formule

$$P = \sum_{j=1}^{K} f_j(n_j) \rightarrow \max$$

subject to: $\sum_{j=1}^{K} n_j \le N \qquad n_j \ge 0$

Optimal value function:

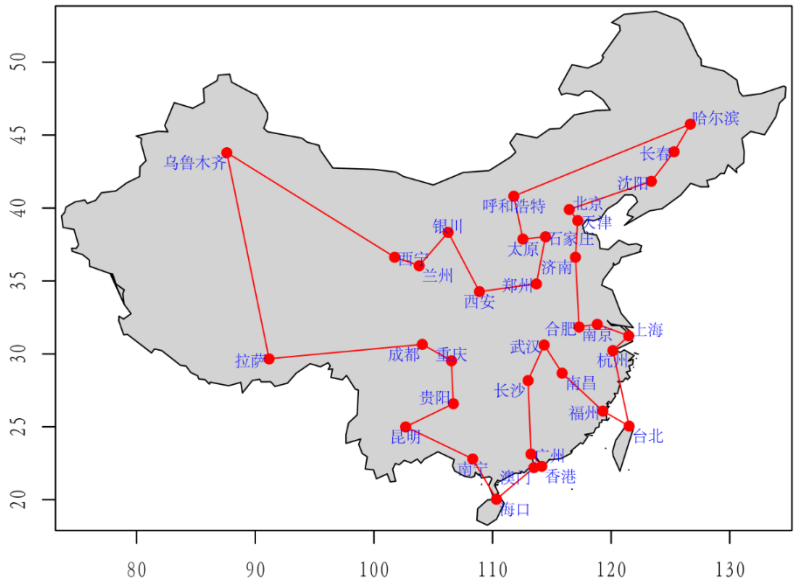
$$G_k(n) = \max_{n_1+\ldots+n_j=n} \left\{ \sum_{j=1}^{k} f_j(n_j) \right\} \qquad F = G_K(N)$$

Recursive equation:

$$G_k(n) = \max_{0 \le m \le n} \{ G_{k-1}(m) + f_k(n-m) \}; \quad 0 \le n \le N.$$

Initialization:

$$G_0(n) = 0; \quad 0 \le n \le N$$



Solution in R Language

```
library(TSP)                    #TSP问题求解包
library(maps)                   #用于绘制地图
library(maptools)               #用于添加文本标签
pr=read.csv("province.csv") #读取数据
f.dis=function(x,y){            #该函数计算地球上两点之间的球面距离
  R=6371                        #地球的平均半径
  x=x*pi/180;y=y*pi/180         #角度化弧度
  a=c(cos(x[2])*cos(x[1]),cos(x[2])*sin(x[1]),sin(x[2])) #x点的直角坐标
  b=c(cos(y[2])*cos(y[1]),cos(y[2])*sin(y[1]),sin(y[2])) #y点的直角坐标
  cosg=sum(a*b)/sqrt(sum(a^2)*sum(b^2))  #计算球面两点过大圆的圆心角的余弦
  dis=R*acos(cosg)                       #得到两点的球面距离
}
k=cbind(pr$jd,pr$wd)                     #获取经度纬度数据
dis.mat=matrix(NA,34,34)                 #预定义矩阵，存放距离数据
for(i in 1:34){
  for(j in 1:34){
    dis.mat[i,j]=f.dis(k[i,],k[j,])
  }
}                                        #计算34个城市两两之间的距离
colnames(dis.mat)=rownames(dis.mat)=pr[,1]          #行列名为城市名
tsp=TSP(dis.mat)                                   #格式转换为TSP
tour=solve_TSP(tsp,method="2-opt")                 #求解TSP问题
path=as.integer(tour)                             #得出数字路线
tour_length(tsp, tour)                            #计算总路线长度
map("world", "China", fill=T, col="lightgray")    #绘制中国地图
map.axes()                                        #加上坐标
xx=pr[path,]
attach(xx)
points(c(jd,jd[1]),c(wd,wd[1]),col=2,pch=19,type="o")   #绘制路线
pointLabel(jd,wd,city,col=4,cex=0.74)                   #标出城市
detach(xx)
```

- 兔子朝着比现在高的地方跳去，他们找到了不远处的最高山峰（局部搜索）
- 兔子喝醉了。它随机地跳了很久，它可能走向高处，也可能踏入平地。但它渐渐清醒了并朝最高方向跳去（模拟退火）
- 兔子失忆被发射到太空，随机落到某地。若不断杀死部分海拔低处的兔子，幸存者就会找到珠峰（遗传算法）
- 兔子们互相转告着哪里的山已找过，且在找过的山都留下了兔子做记号。据此它们制定下一步去哪里寻找的策略（禁忌搜索）

“有志气的兔子”----寻找世界最高的山

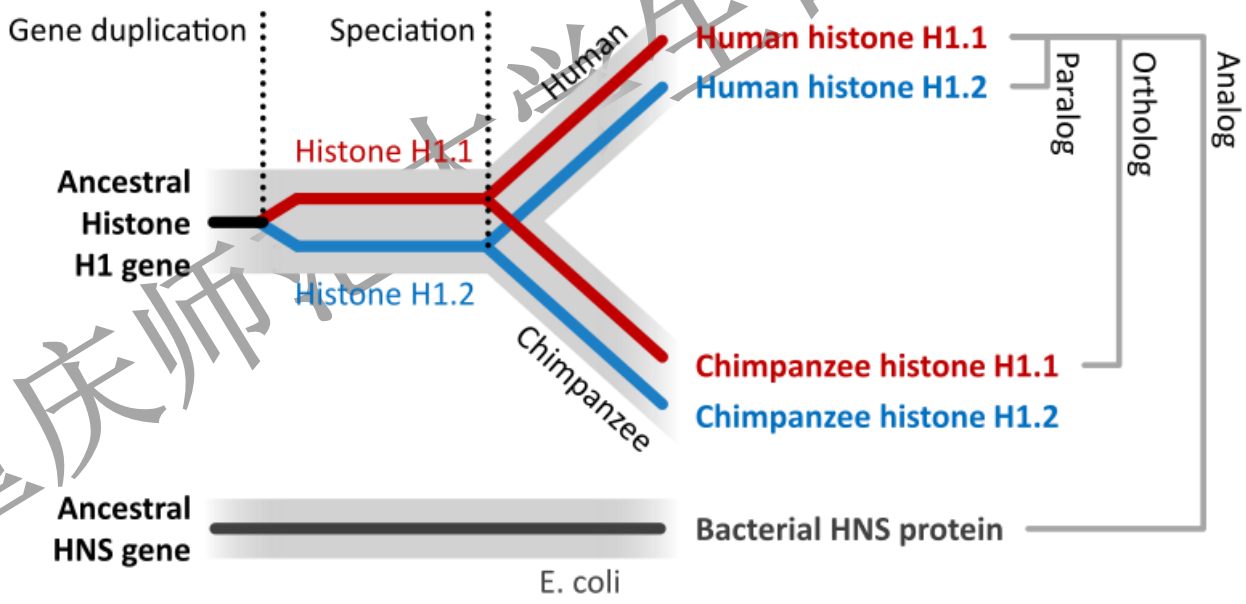
第2节：

重要生物学概念

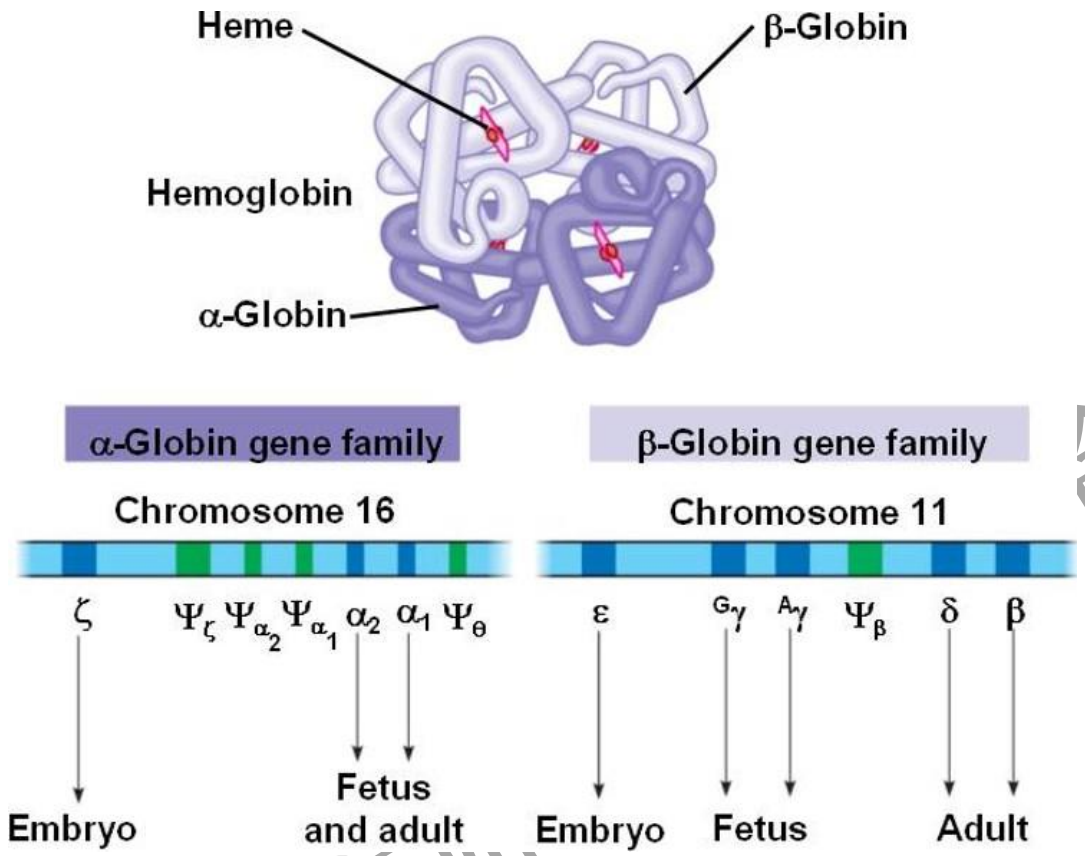
- ✓ 同源性
- ✓ 相似性
- ✓ 空位罚分
- ✓ 序列比对

2.1 什么是同源性？

- **Sequence homology** is the biological homology between DNA, RNA, or protein sequences, defined in terms of shared ancestry in the evolutionary history of life.
- Two segments of DNA can have shared ancestry because of **three phenomena**: either a speciation event (orthologs), or a duplication event (paralogs), or else a horizontal (or lateral) gene transfer event (xenologs).



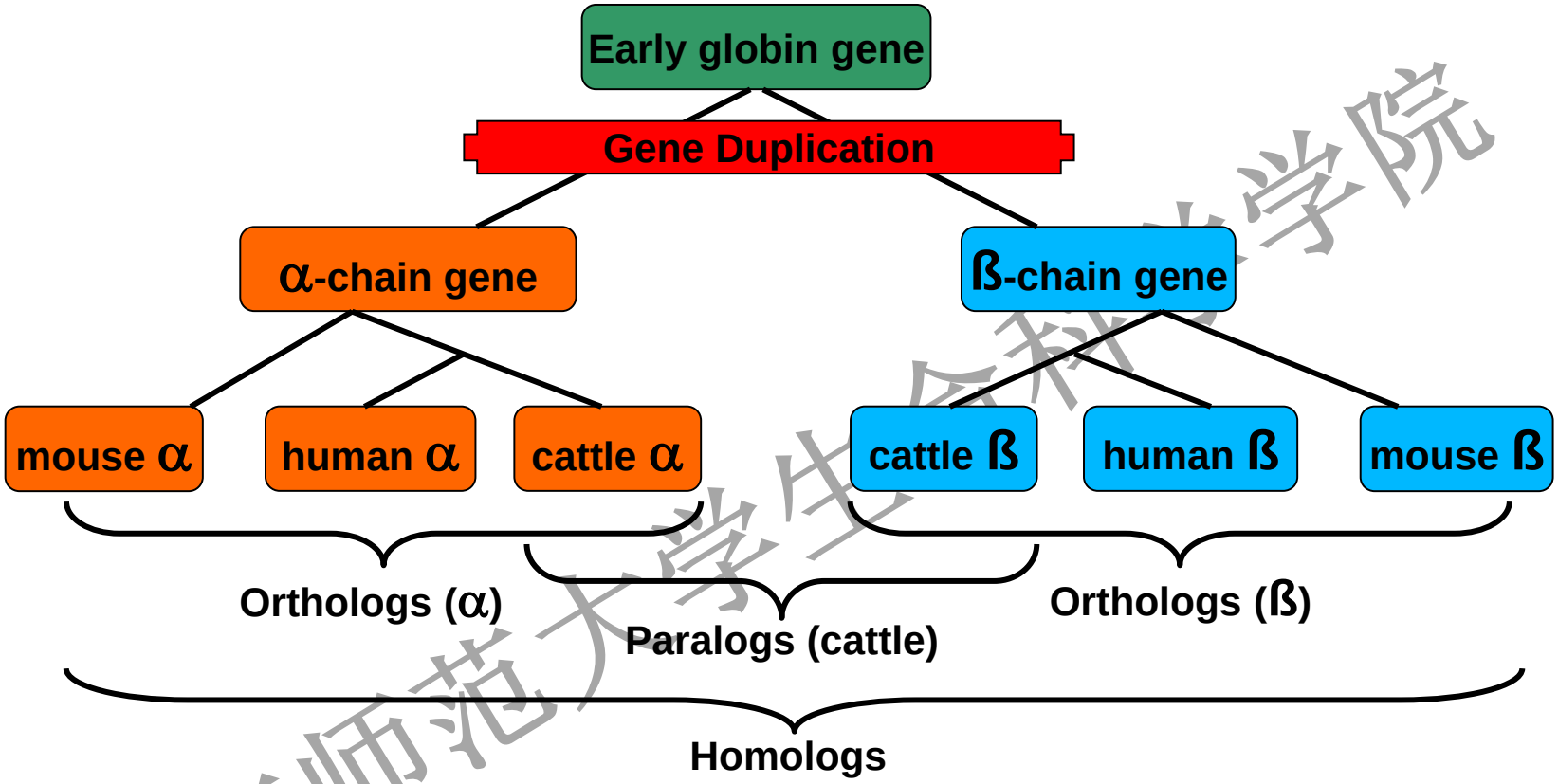
https://en.wikipedia.org/wiki/Sequence_homology



- **Orthologs**（垂直同源）: genes derived from same ancestor and have the same function in different species.
- **Paralogs**（水平同源）: genes within one organism that arose due to gene duplication and usually have evolved different functions.
- **Homology**(同源)—具有共同的祖先，通常序列比较相似。
- **Similarity**(相似)—序列相像，但不一定同源。

➢ Genes which are similar but not identical and occur in multiple copies throughout the genome are called **multigene families**.

Orthologous *v.s.* paralogous homologs



Orthologs – diverged after speciation – *tend to have similar function*

Paralogs – diverged after gene duplication – *some functional divergence occurs*

Therefore, for linking similar genes between species, or performing "annotation transfer", identify orthologs

相似性与同源性

- 相似的序列并不一定同源
- 相似性是可以被量化的“计分表”，它是匹配的数量除以比对的长度，通常以百分比%表示
- 同源性一定是指序列来自共同的祖先
- 同源性是一个定性的概念，不能使用序列间具有百分之多少同源性来定义
- **Inferring homology from similarity**
- **Inferring function from homology**
- **From pairwise to multiple sequence alignment**

Ref: William R. *Curr Protoc Bioinformatics*. 2013, June, 42(1):3-1.

重要概念

- 什么是序列比对
- 空位罚分
- 相似性与同源性
- 双序列比对方法
  - 点阵序列比较(Dot Matrix Sequence Comparison)
  - 动态规划算法(Dynamic Programming Algorithm)
- 记分矩阵

Pairwise Alignment *versus* Multiple Alignment

什么是双序列比对？

定义：双序列比对是比较两条核酸或蛋白质序列相似性的一种方法

我们为什么关注序列比对？

- ✓ 相似的序列可能具有相似的功能与结构
- ✓ 发现一个基因或蛋白哪些区域容易发生突变，哪些位点突变后对功能没有影响
- ✓ 发现生物进化方面的信息

序列比对两种类型

- 全局序列比对
  - 定义：在全局范围内对两条序列进行比对打分的方法
  - 适合于非常相似且长度近似相等的序列
- 局部序列比对
  - 定义：一种寻找匹配子序列的序列比对方法
  - 适合于一些片段相似而另一些片段相异的序列

Local Alignment

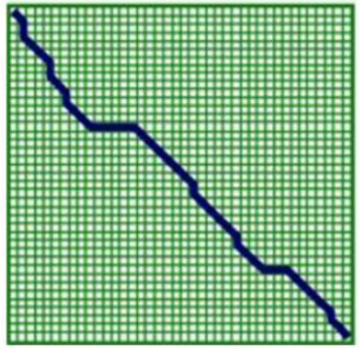
```
5' ACTACTAGATTACTTACGGATCAGGTACTTTAGAGGCTTGCAACCA 3'
            |||| |||||| ||||||||||||||||
         5' TACTCACGGATGAGGTACTTTAGAGGC 3'
```

Global Alignment

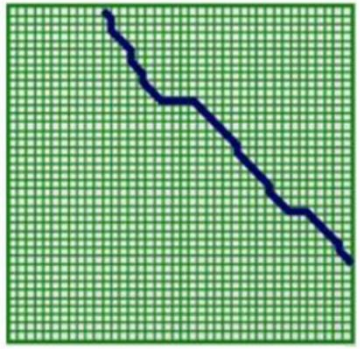
```
5' ACTACTAGATTACTTACGGATCAGGTACTTTAGAGGCTTGCAACCA 3'
   ||||||||||    |||||||  |||||||||||||| |||||||
5' ACTACTAGATT----ACGGATC--GTACTTTAGAGGCTAGCAACCA 3'
```

局部序列比对 vs 全局序列比对

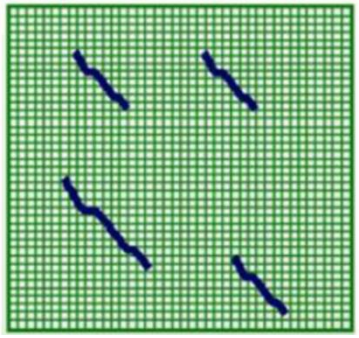
Global: Require an end-to-end alignment of **x**,**y**



Semi-global (glocal): Gaps at the beginning or end of **x** or **y** are free — useful when one string is significantly shorter than the other or for finding overlaps between strings



Local: Find the highest scoring alignment between **x'** a substring of **x** and **y'** a substring of **y** — useful for finding similar regions in strings that may not be globally similar



空位罚分(Gap Penalties)

- 空位为了获得两个序列最佳比对，必须使用空位和空位罚分

- 空位罚分分类：
  - 空位开放罚分(Gap opening penalty)
  - 空位扩展罚分(Gap extension penalty)

- 最优的序列比对通常具有以下特征：
  - 尽可能多的匹配
  - 尽可能少的空位

- 插入任意多的空位会产生较高的分数，但找到的并不一定是真正相似序列

空位罚分(Gap Penalties)

不允许有空位

```
1  GTGATAGACAC
   |||
1  GTGCATAGACAC
```

Score: -21

match = 5
mismatch = -4

允许空位但不罚分

```
1  GTG-ATAGACAC
   |||.|||||||
1  GTGCATAGACAC
```

?

≠

```
1  GTG--ATAGACAC
   |||  |||||||||
1  GTGC-ATAGACAC
```

Score: 55

空位罚分公式

$$Wx=g+r(x-1)$$

Wx: 空位总记分

g: 空位开放罚分

r: 空位扩展罚分

x: 空位长度

参数：

匹配= 1

非匹配= 0

g= 3

r = 0.1

x = 3

Wx= -3 - (3 -1) 0.1 = -3.2

```
  A T G T T A T A C
  | | | |
T A T G T G C G T A T A
```

Score=4

```
  A T G T - - - T A T A C
  | | | |       | | | |
T A T G T G C G T A T A
```

insertion / deletion

score：8 - 3.2 = 4.8

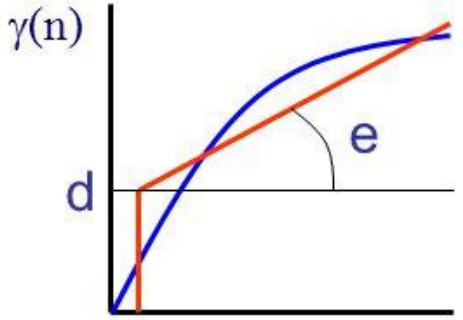
Affine gap penalties (亲和空位罚分)

Compromise: **affine gaps**

$$\gamma(n) = d + (n-1)\times e$$

d: gap open

e: gap extension



Match: 2

Gap open: -5

Gap extension: -1

```
GACGCCGAACG
|||||   |||
GACGC---ACG
```

8x2-5-2 = 9

```
GACGCCGAACG
||||  |  |  ||
GACG-C-A-CG
```

8x2-3x5 = 1

We want to find the optimal alignment with affine gap penalty in

- O(MN) time
- O(MN) or better O(M+N) memory

如何考察两序列而是否相似？

- 基于距离的策略
  - **distance**，如欧氏距离、马氏距离等
- 基于相似性的策略
  - **similarity**

编辑距离（edit distance)

seq1 =	ATC	AGGCT	GCTAGCTA
seq2 =	TAC	ACCTT	CGTGAGCA
Hamming Distance(seq1,seq2)=	2	3	6

相似性得分

			seq1 =	ATC	AGGCT	GCTAGCTA
			seq2 =	TAC	ACCTT	CGTGAGCA
打分规则1	p(a, a)=1 p(a, b)=0 (a≠b)	相似性得分=		1	2	2
打分规则2	p(a, a)=0.8 p(a, b)=0.2 (a≠b)	相似性得分=		1.2	2.2	2.8
打分规则BLAST		相似性得分=		-3	-2	-6

	A	T	C	G
A	5	-4	-4	-4
T	-4	5	-4	-4
C	-4	-4	5	-4
G	-4	-4	-4	5

第3节：

序列的两两比对
（Pairwise Sequence Alignment）

按字符位置重组两个序列，使得两个序列接近一样的长度

序列两两比对基本算法

假设比较300个氨基酸长度的两条序列

$$A_1A_2A_3\cdots\cdots A_{299}A_{300}$$

$$B_1B_2B_3\cdots\cdots B_{299}B_{300}$$

直接方法 —— 生成两个序列所有可能的比对，分别计算代价函数，然后挑选一个代价最小的比对作为最终结果，需要计算 2^{300} 次——天文数字

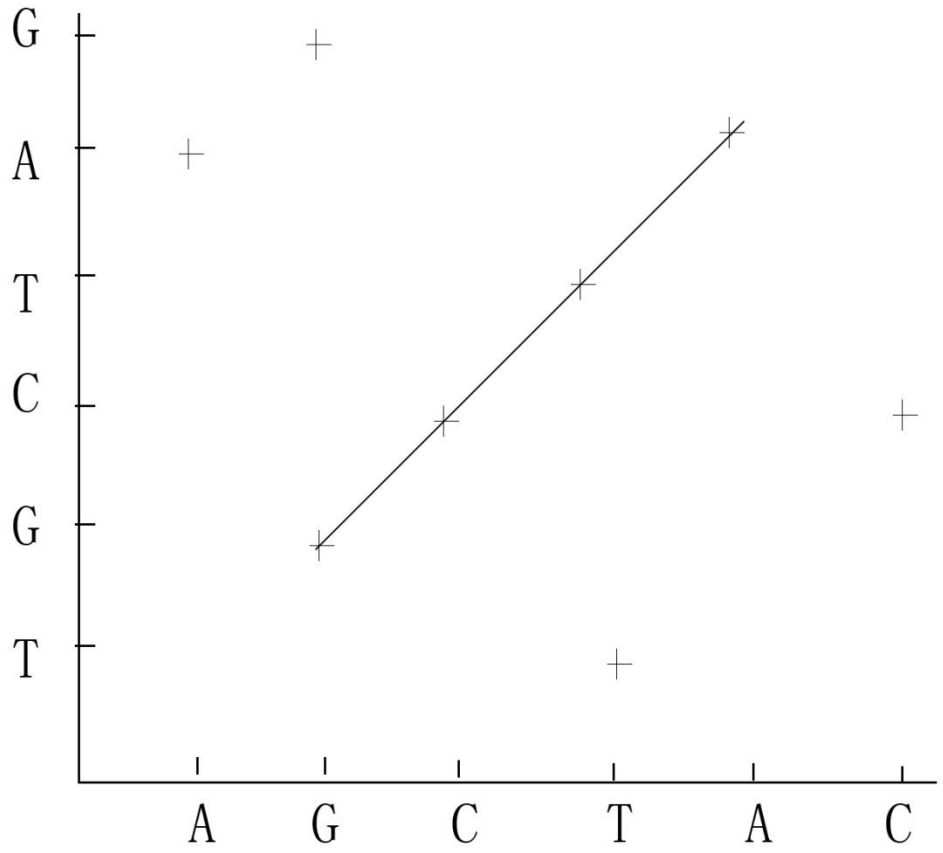
双序列比对方法

- 点阵序列比较 (Dot Matrix Sequence Comparison)
- 动态规划算法 (Dynamic Programming Algorithm)
- 词或K串方法 (Word or *K*-tuple Methods)

点阵序列比较

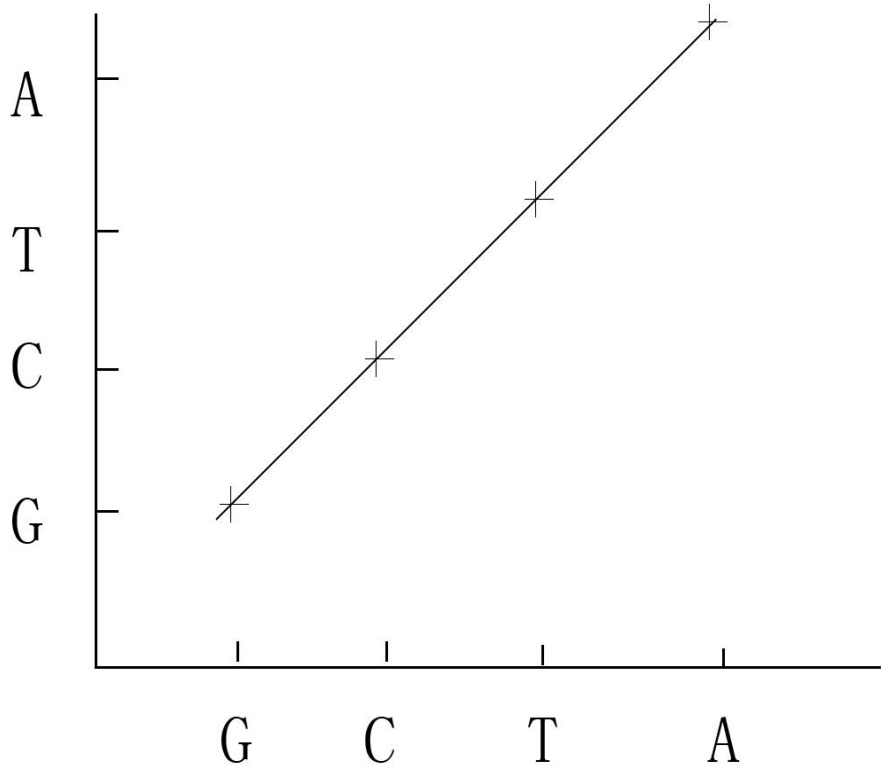
点阵序列比较(Dot Matrix Sequence Comparison) 是将X轴上序列的每一个单元与Y轴上序列的每一个单元进行比较，相同匹配在相应区域用点标记的图形化显示双序列比对方法

从点阵分析我们可以得到什么信息？

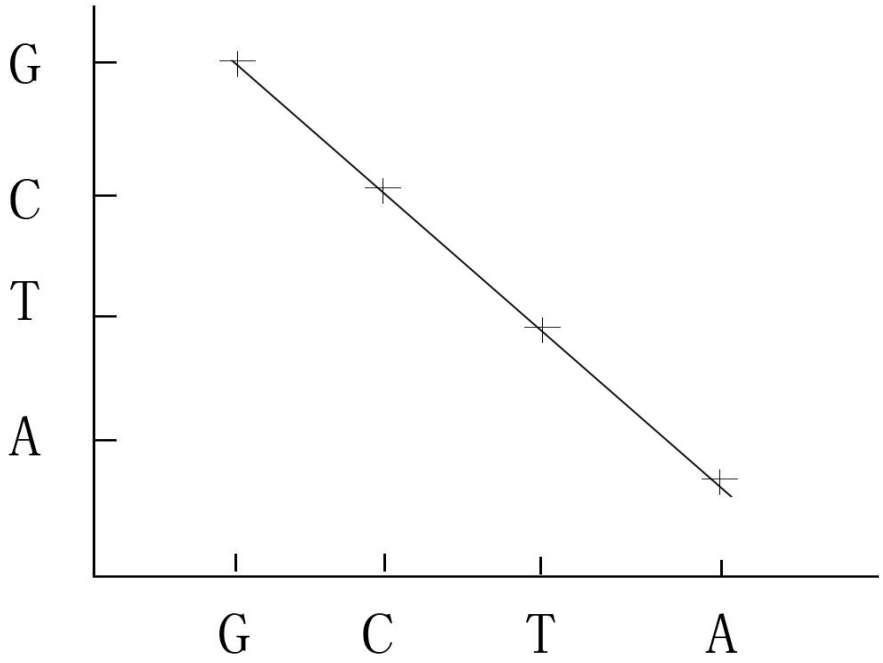


A G C T A C
| | | |
T G C T A G

通过点矩阵进行序列比较

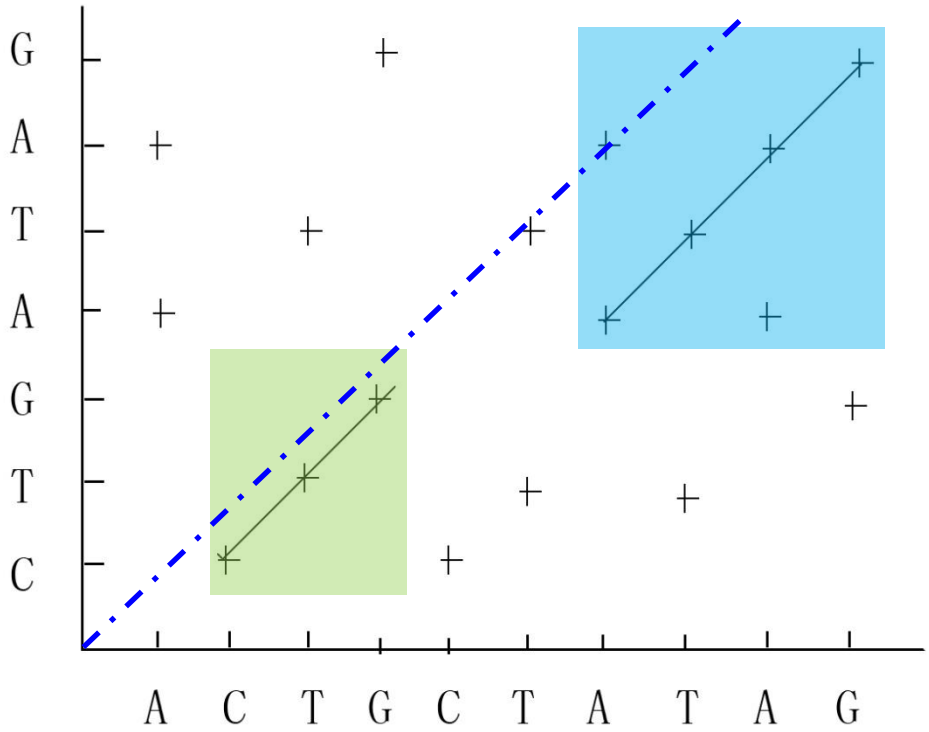


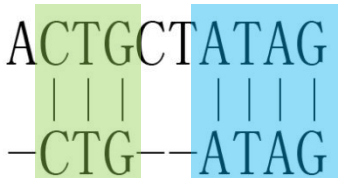
G C T A
| | | |
G C T A



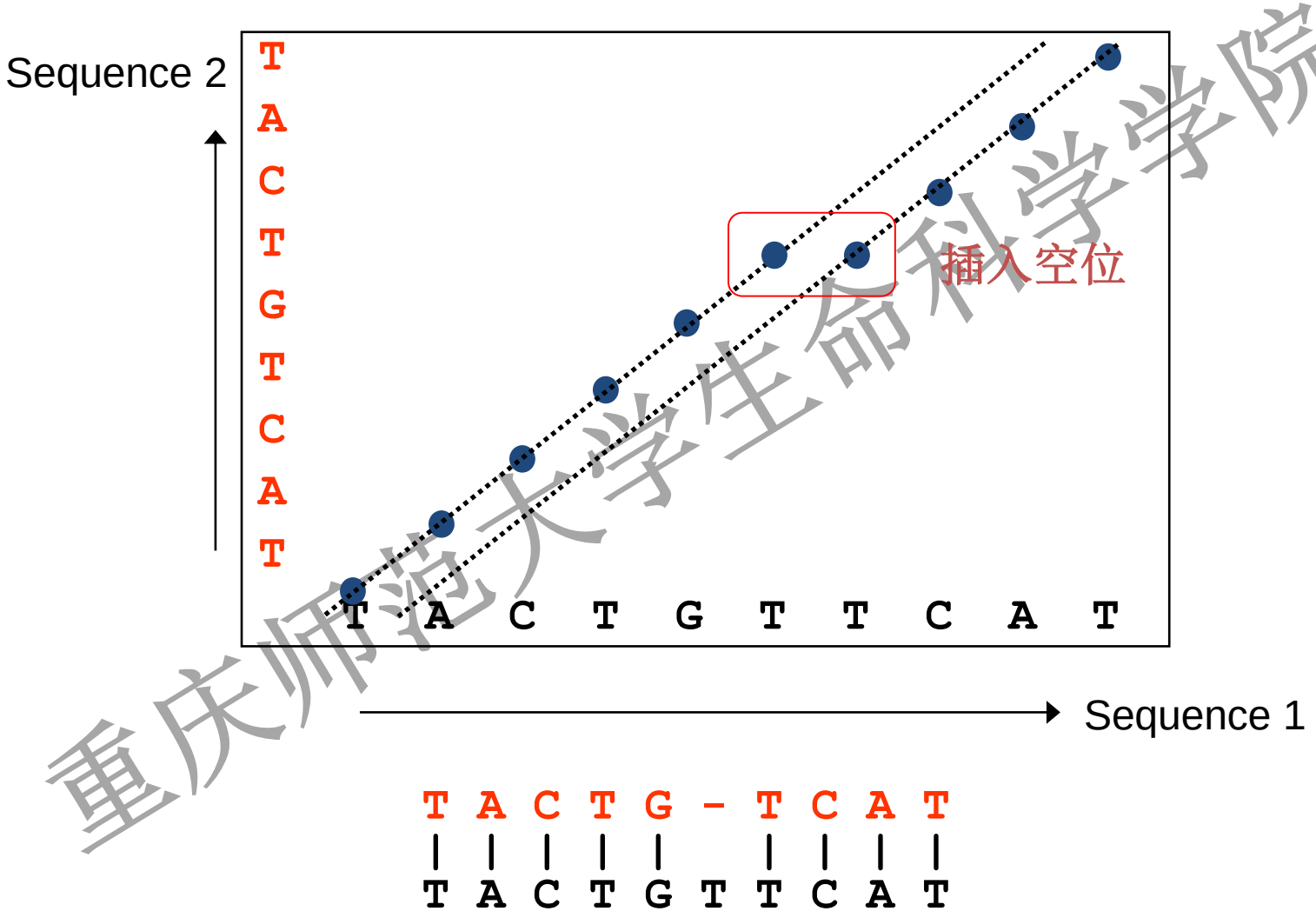
G C T A

A T C G

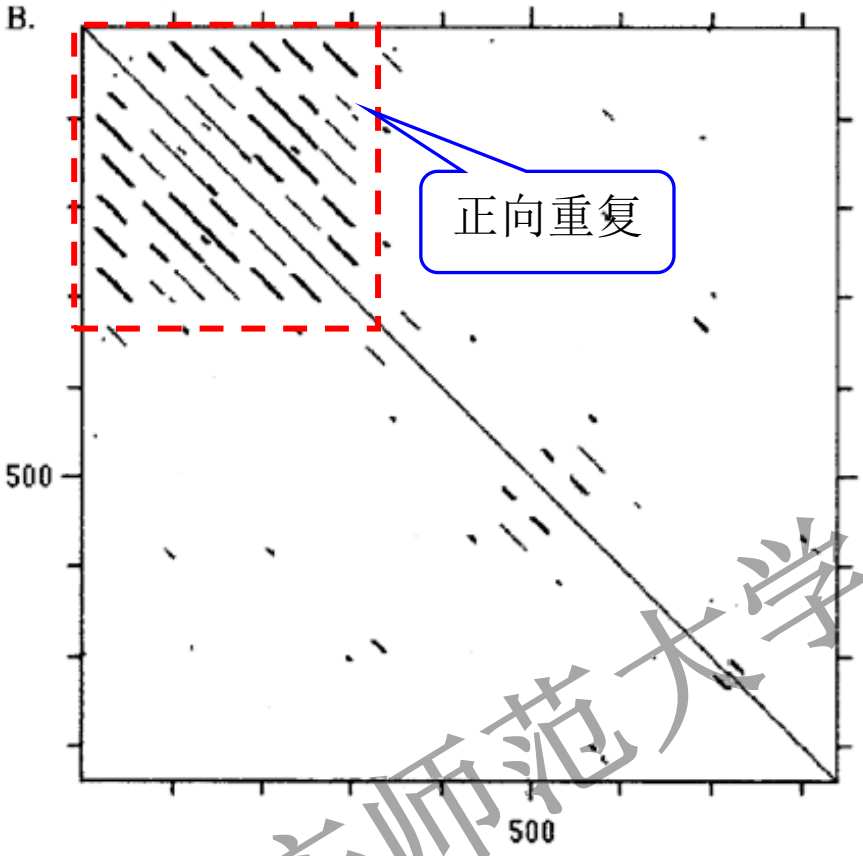

G
A
T
A
G
T
C
A C T G C T A T A G


ACTGCTATAG
CTG--ATAG

点阵分析中的插入或删除

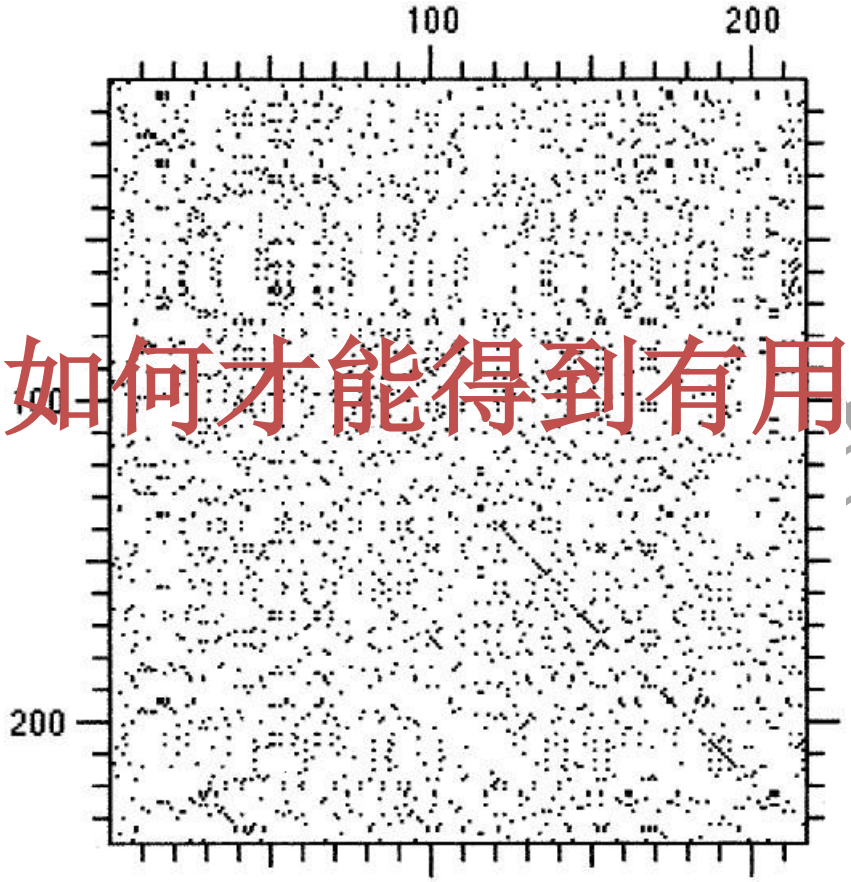



```
T A C T G - T C A T
| | | | |   | | | |
T A C T G T T C A T
```



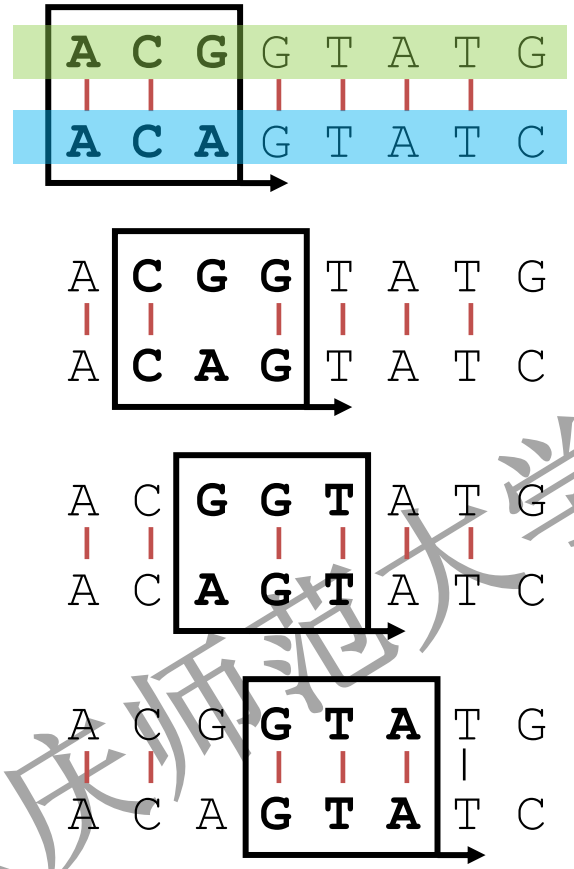
最新研究表明LDLR正向重复序列与动脉粥样硬化密切相关(Wilson *et al*, 2011, *science*, 933-937)

人低脂蛋白受体(human low-density lipoprotein receptor)自身比对

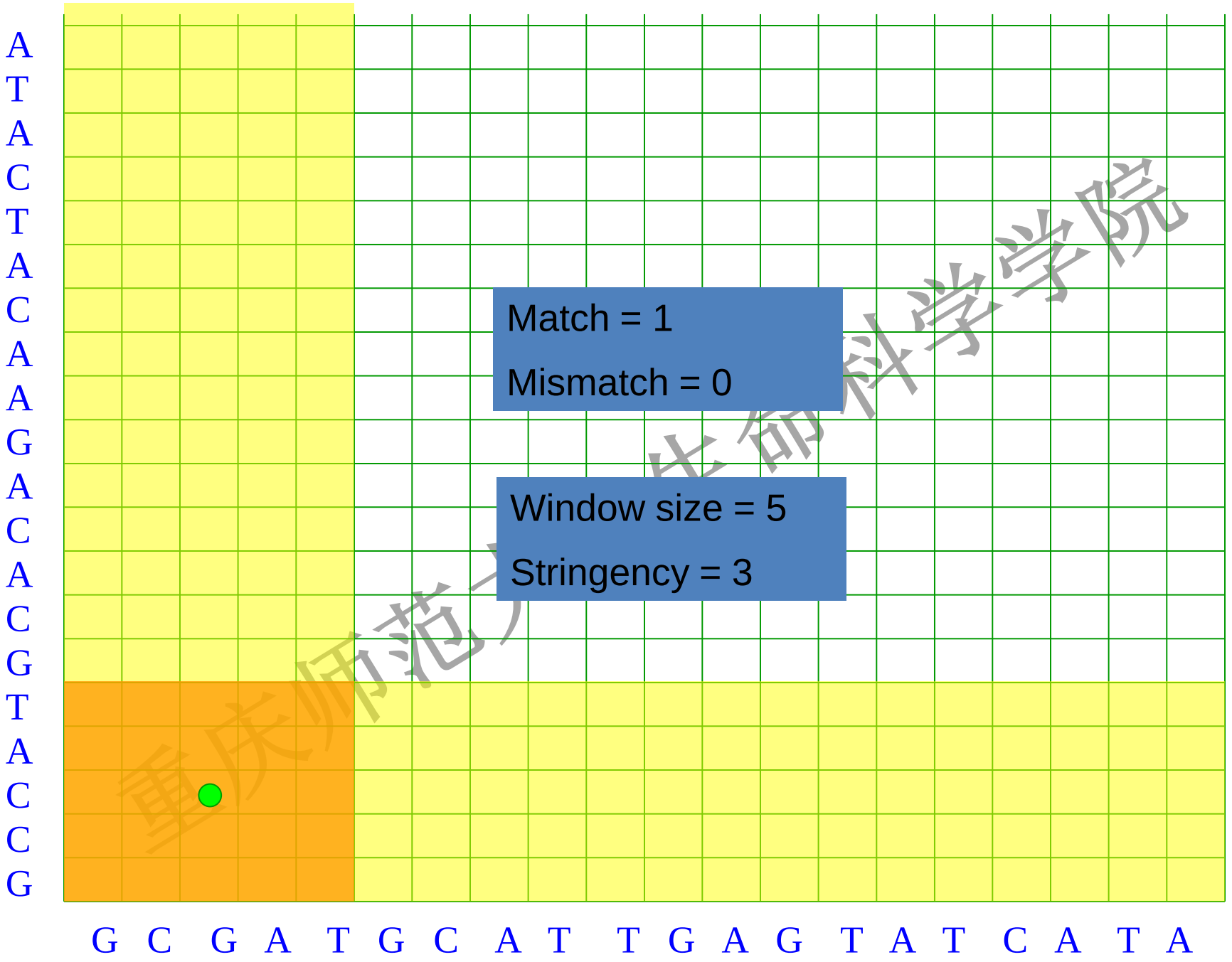


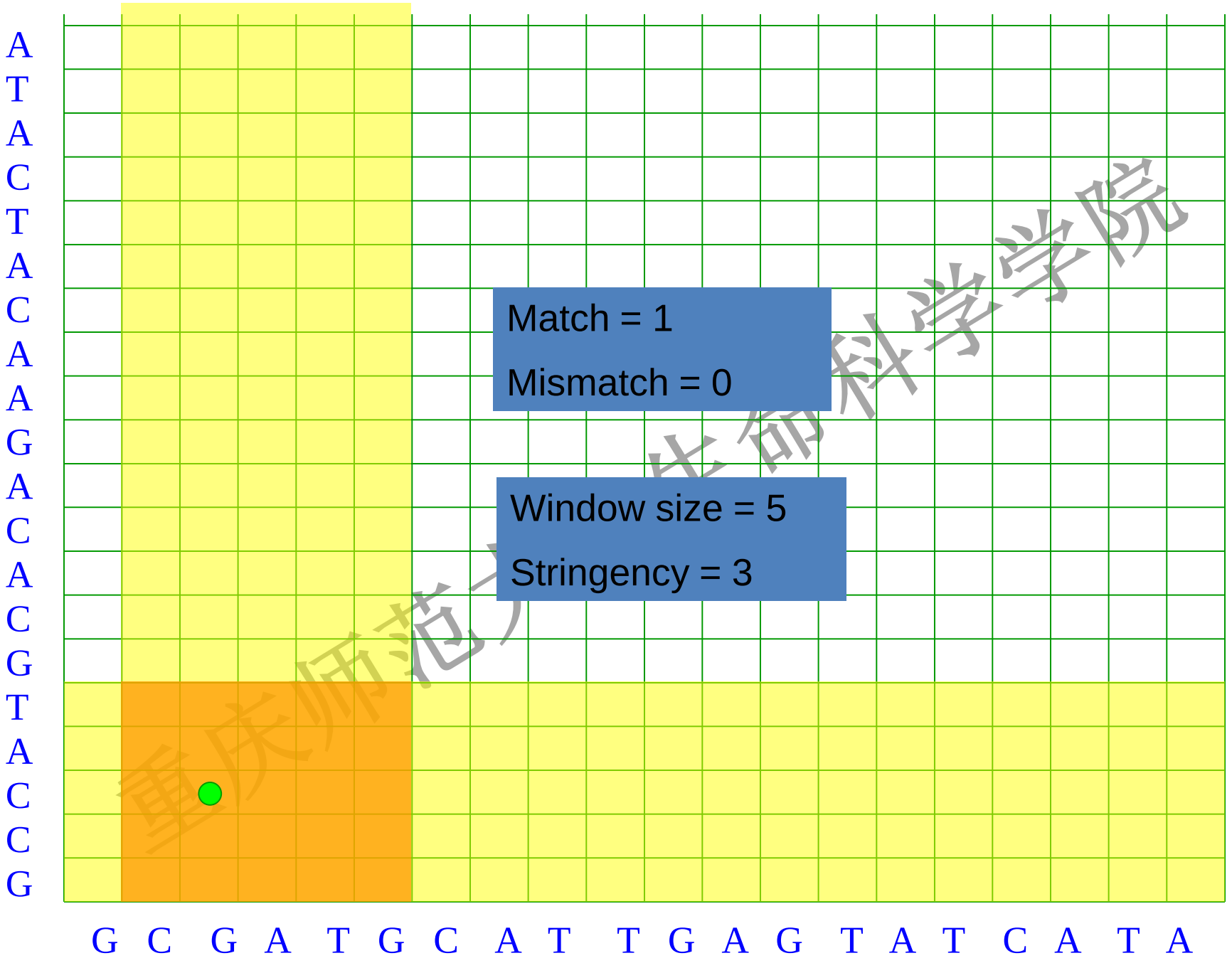
- 编码噬菌体λc1(水平轴)和噬菌体P22 c2(垂直轴)的氨基酸序列间的点阵分析

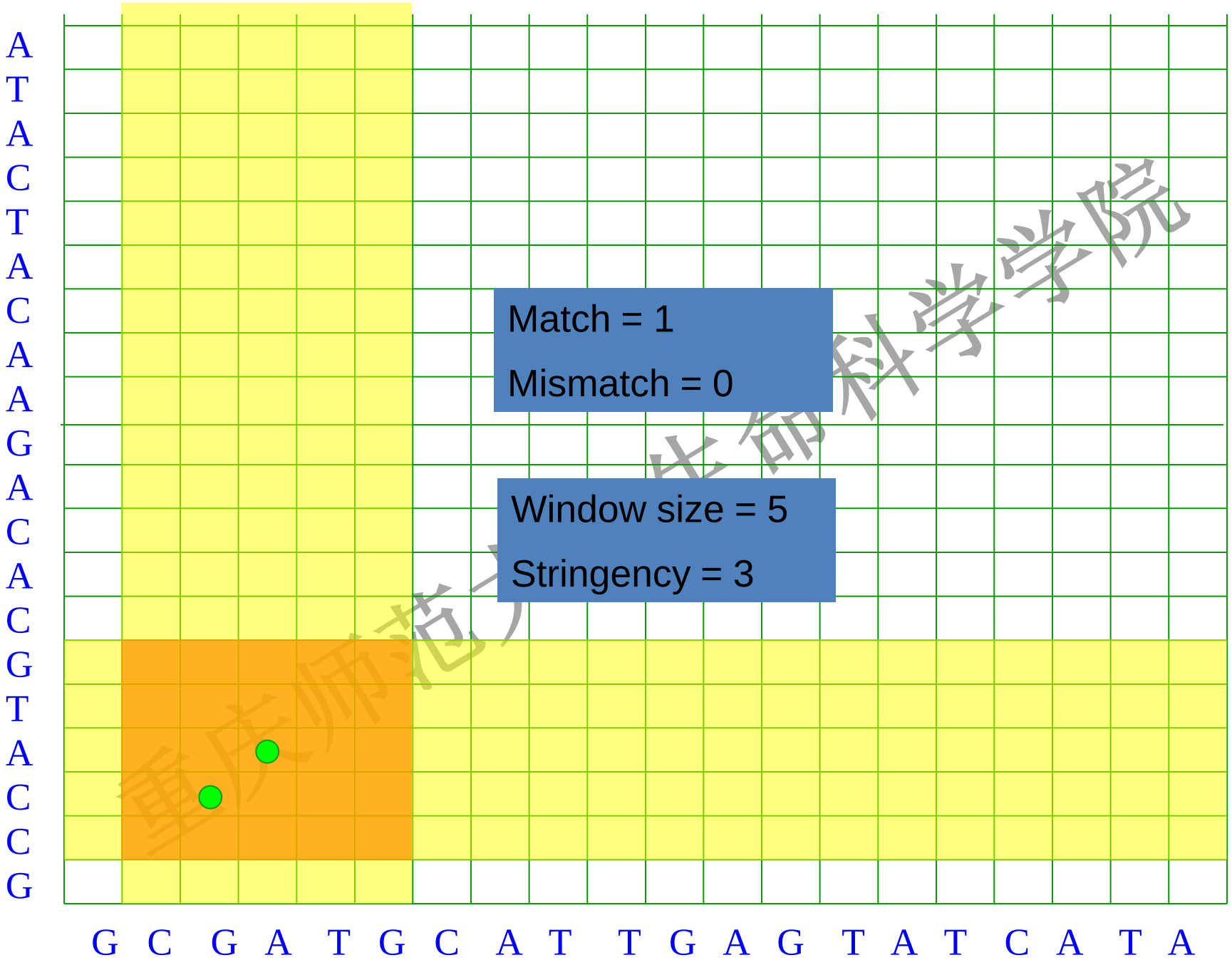
使用滑动窗口技术降低噪声

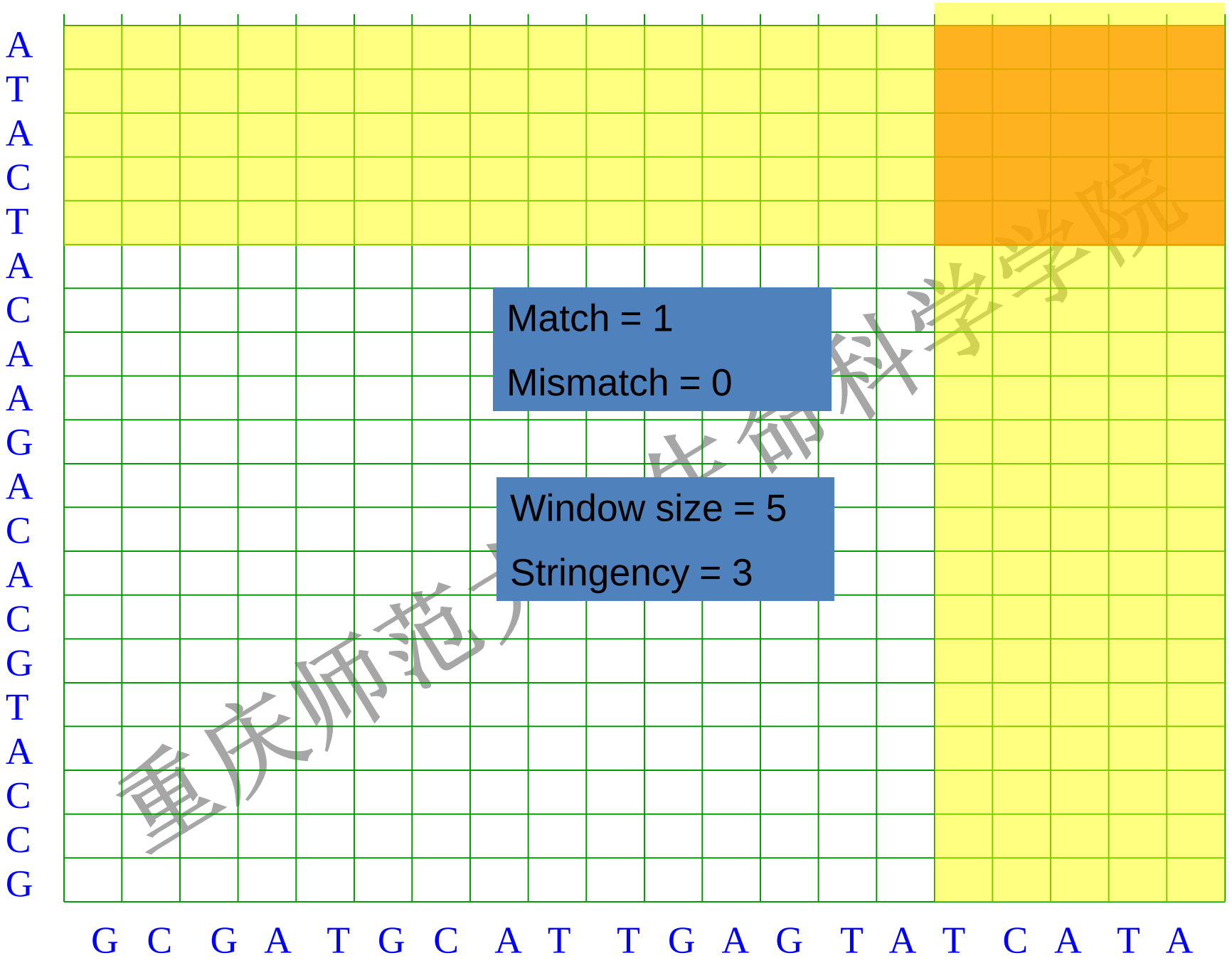


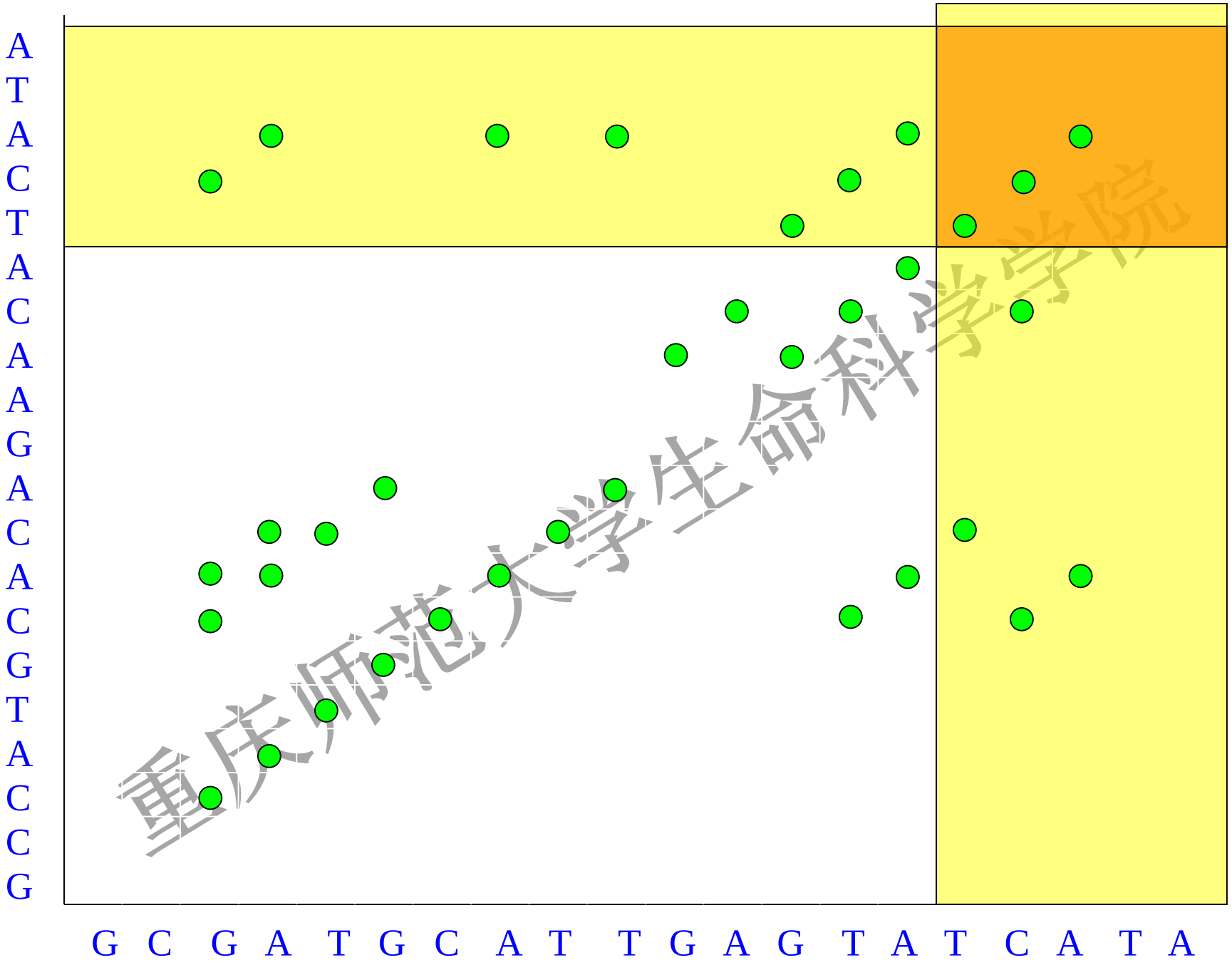
窗口 = 3
阈值 = 3

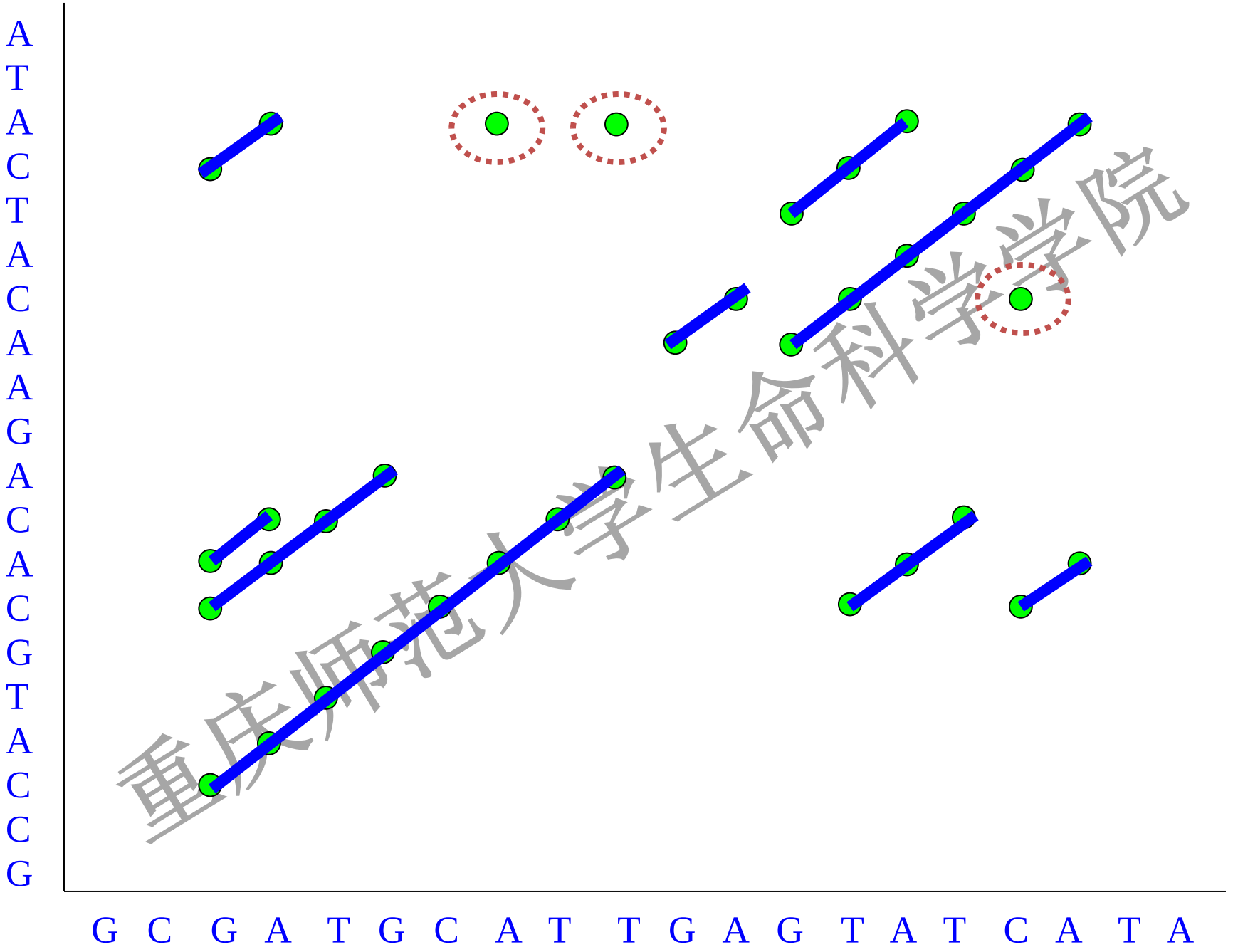

A
T
A
C
T
A
C
A
A
G
A
C
A
C
G
T
A
C
C
G
Match = 1
Mismatch = 0
Window size = 5
Stringency = 3
G C G A T G C A T T G A G T A T C A T A
重庆师范大学生命科学学院


A
T
A
C
T
A
C
A
A
G
A
C
A
C
G
T
A
C
C
G
G C G A T G C A T T G A G T A T C A T A
Match = 1
Mismatch = 0
Window size = 5
Stringency = 3

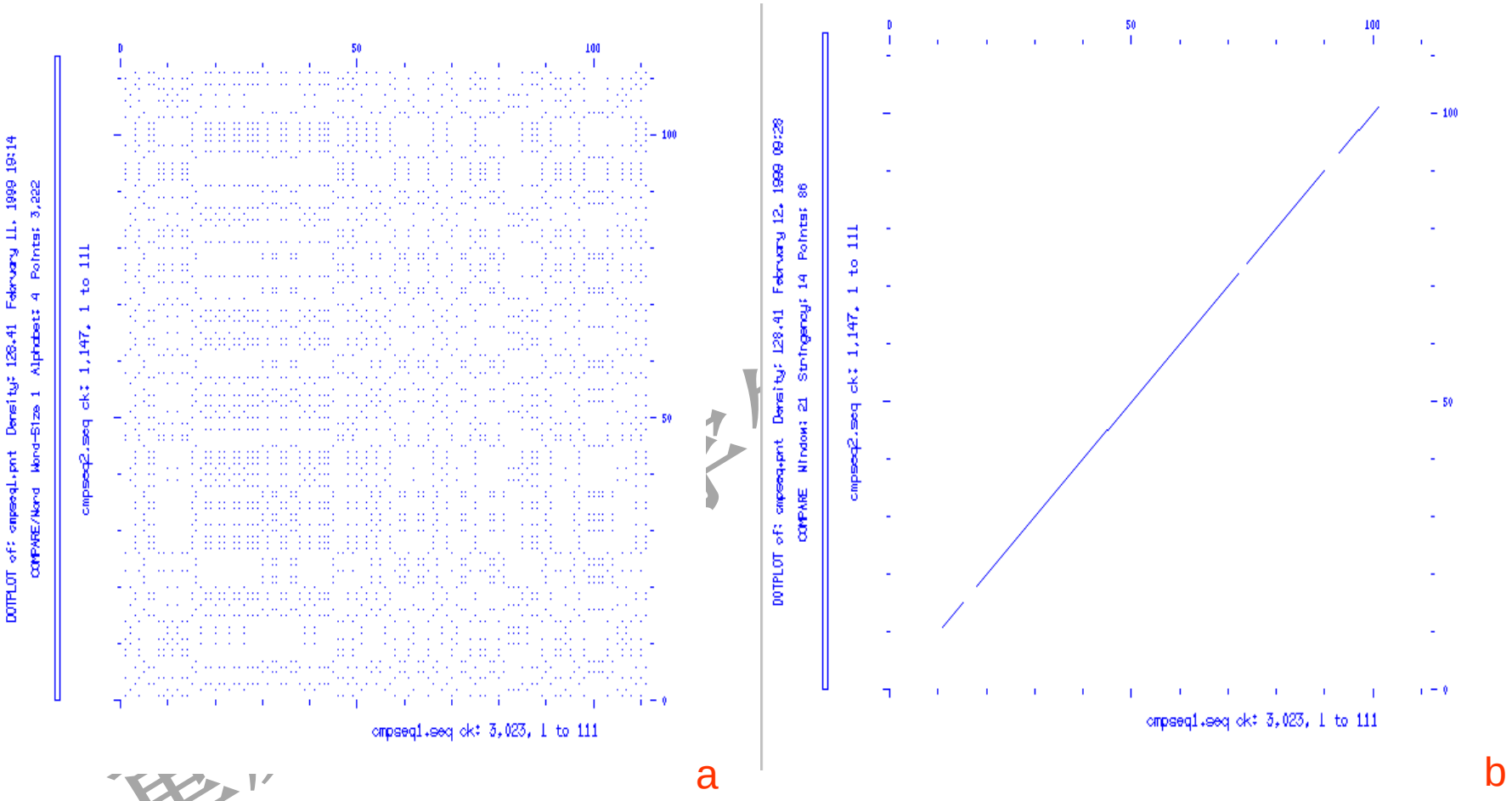

A
T
A
C
T
A
C
A
A
G
A
C
A
C
G
T
A
C
C
G
G C G A T G C A T T G A G T A T C A T A
Match = 1
Mismatch = 0
Window size = 5
Stringency = 3


A
T
A
C
T
A
C
A
A
G
A
C
A
C
G
T
A
C
C
G
Match = 1
Mismatch = 0
Window size = 5
Stringency = 3
G C G A T G C A T T G A G T A T C A T A
重庆师范大学生命科学学院


A
T
A
C
T
A
C
A
A
G
A
C
A
C
G
T
A
C
C
G
G C G A T G C A T T G A G T A T C A T A
重庆师范大学生命科学学院


A
T
A
C
T
A
C
A
A
G
A
C
A
C
G
T
A
C
C
G
G C G A T G C A T T G A G T A T C A T A
重庆师范大学生命科学学院

使用滑动窗口技术降低噪声



（a）对人类（Homo sapiens）与黑猩猩（Pongo pygmaeus）的β球蛋白基因序列进行比较的完整点阵图
（b）利用滑动窗口对以上的两种球蛋白基因序列进行比较的点阵图，其中窗口大小为10个核苷酸，相似度阈值为8，即10个核苷酸中有8个相同时就打一个点

从点阵分析我们可以得到什么信息？

寻找序列中的正向或反向重复序列

相同残基重复出现的低复杂区

发现蛋白质的重复结构域(domain)

点阵分析的优缺点

优点

- ✓ 直观且具备整体性
- ✓ 不依赖任何先决条件，是一种可用于初步分析的理想工具
- ✓ 点阵分析可以用来摸索区分信号和背景标准的严格程度

缺点

- x 滑动窗口和阈值的选择过于经验化
- x 信噪比低
- x 不适合进行高通量的数据分析

点阵分析程序

- DNA Strider (Macintosh)
  - http://www.cellbiol.com/soft.htm
- Dotter (Unix/Linux, X-Windows)
- COMPARE, DOTPLOT (GCG软件)
- PLALIGN (FASTA)
- Dotlet
  - http://www.isrec.isb-sib.ch/java/dotlet/Dotlet.html

动态规划算法

- 动态规划算法(Dynamic Programming Algorithm)是一种计算方法，它的主要思路是把一个问题分成若干个小问题来解决

- 在生物学中应用的两种动态规划算法：Needleman-Wunsch算法（全局比对）和Smith-Waterman算法（局部比对）

动态规划方法 Dynamic Programming

起点 ATTC.....CGAAGA

AGTC.....GAAGGT 终点

(1) ATTC.....CGAAG + A

AGTC.....GAAGG T

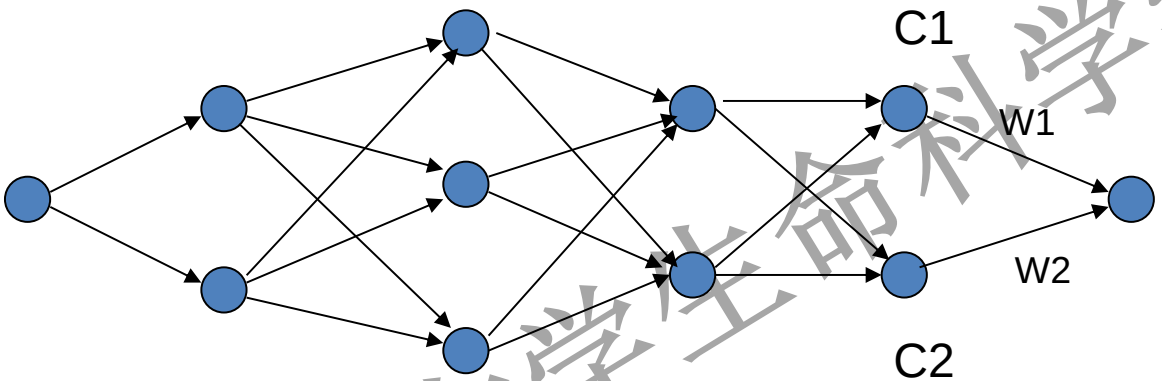
(2) ATTC.....CGAAGA + -

AGTC.....GAAGG T

(3) ATTC.....CGAAG + A

AGTC.....GAAGGT -

最短路经问题



路径1：C1 + w1 ？

路径2：C2 + w2 ？

取最小值！

★ 算法求解：

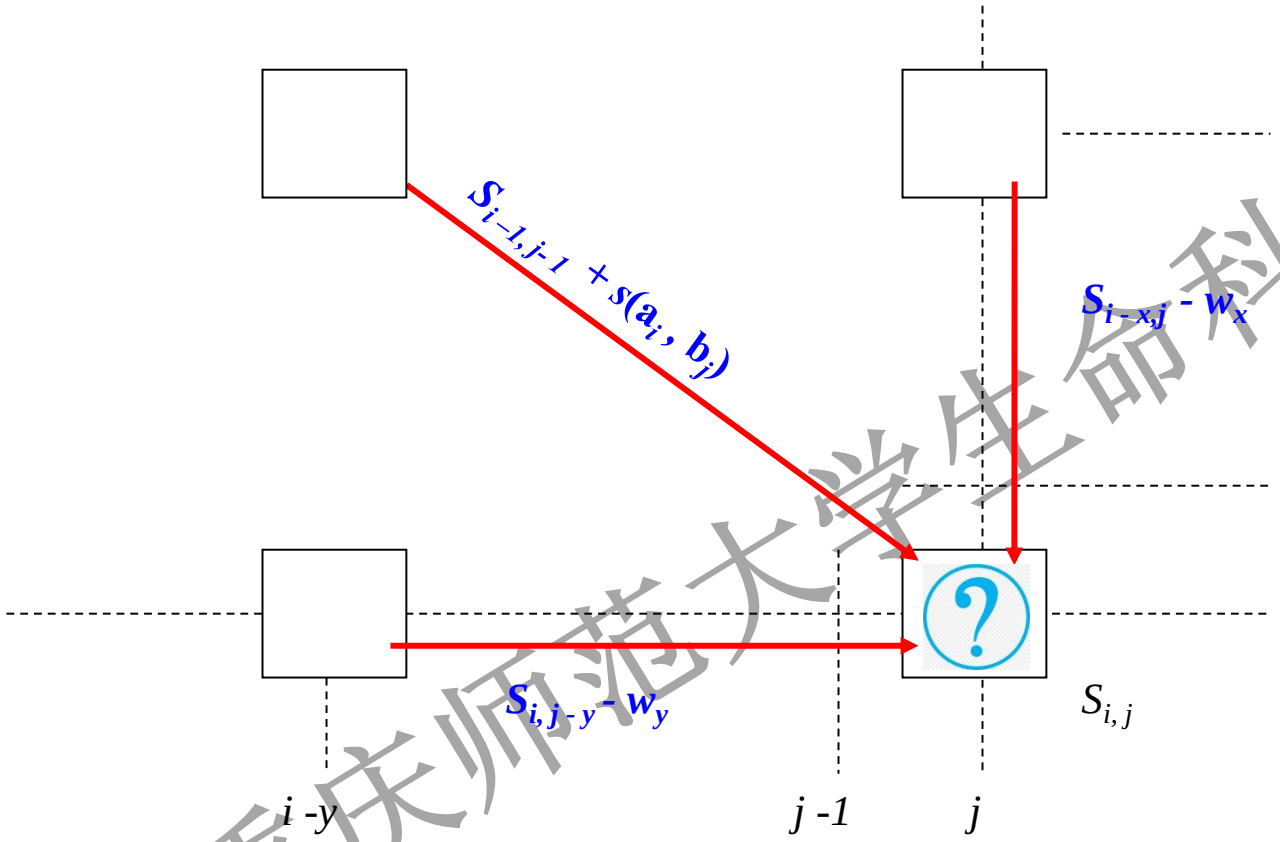
从起点到终点逐层计算

序列比对中某一位点匹配的三种可能性

- *Eg. 匹配=1,非匹配=0, 空位罚分=-1*
  - Sequence1: **CACGA**
  - Sequence2: **CGA**

第一个位点	得分	剩余序列
C C	+1	ACGA GA
- C	-1	CACGA GA
C -	-1	ACGA CGA

动态规划算法的正式表述



$S_{i,j}$这个位置的分数为图中箭头所示三个方向值中最大的一个

说明：Sij是序列a在位置i和序列b在位置j的分值， s(aibj)是位置i 和j上比对分值，wx是在序列a 中长度为x的间隔罚分，wy是序列b中长度为y的间隔罚分

动态规划算法实例

		A	C	T	T	C	G
A							
C							
T							
A							
G							

匹配＝3

错配＝-1

空位＝-2

动态规划算法实例

		A	C	T	T	C	G
	0						
A							
C							
T							
A							
G							

匹配＝3

错配＝-1

空位＝-2

动态规划算法实例

		A	C	T	T	C	G
	0 ←	-2					
A							
C							
T							
A							
G							

匹配＝3

错配＝-1

空位＝-2

动态规划算法实例

		A	C	T	T	C	G
	0 ←	-2 ←	-4 ←	-6 ←	-8 ←	-10 ←	-12 ←
A							
C							
T							
A							
G							

匹配＝3

错配＝-1

空位＝-2

动态规划算法实例

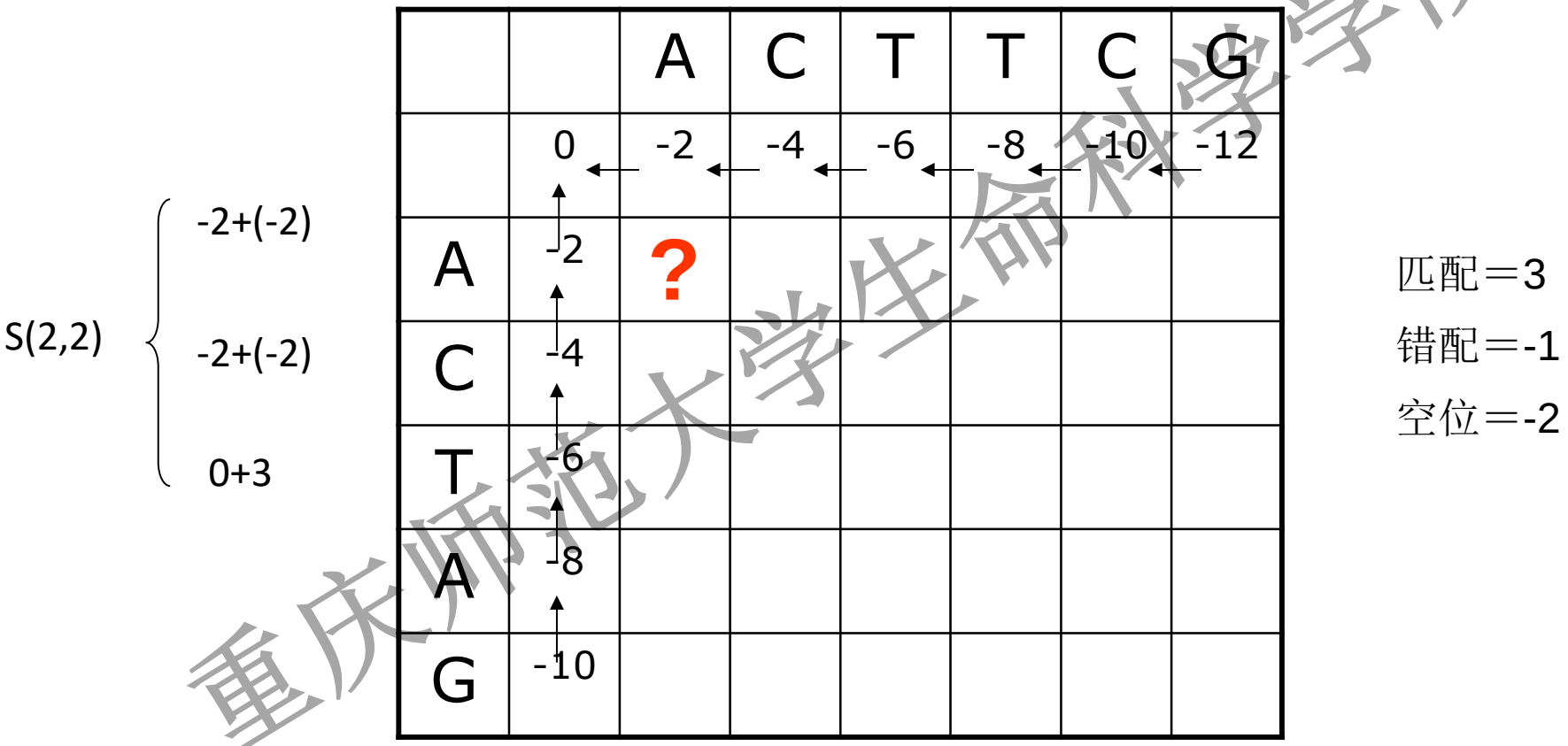
		A	C	T	T	C	G
	0 ←	-2 ←	-4 ←	-6 ←	-8 ←	-10 ←	-12
A	↑ -2						
C							
T							
A							
G							

匹配＝3

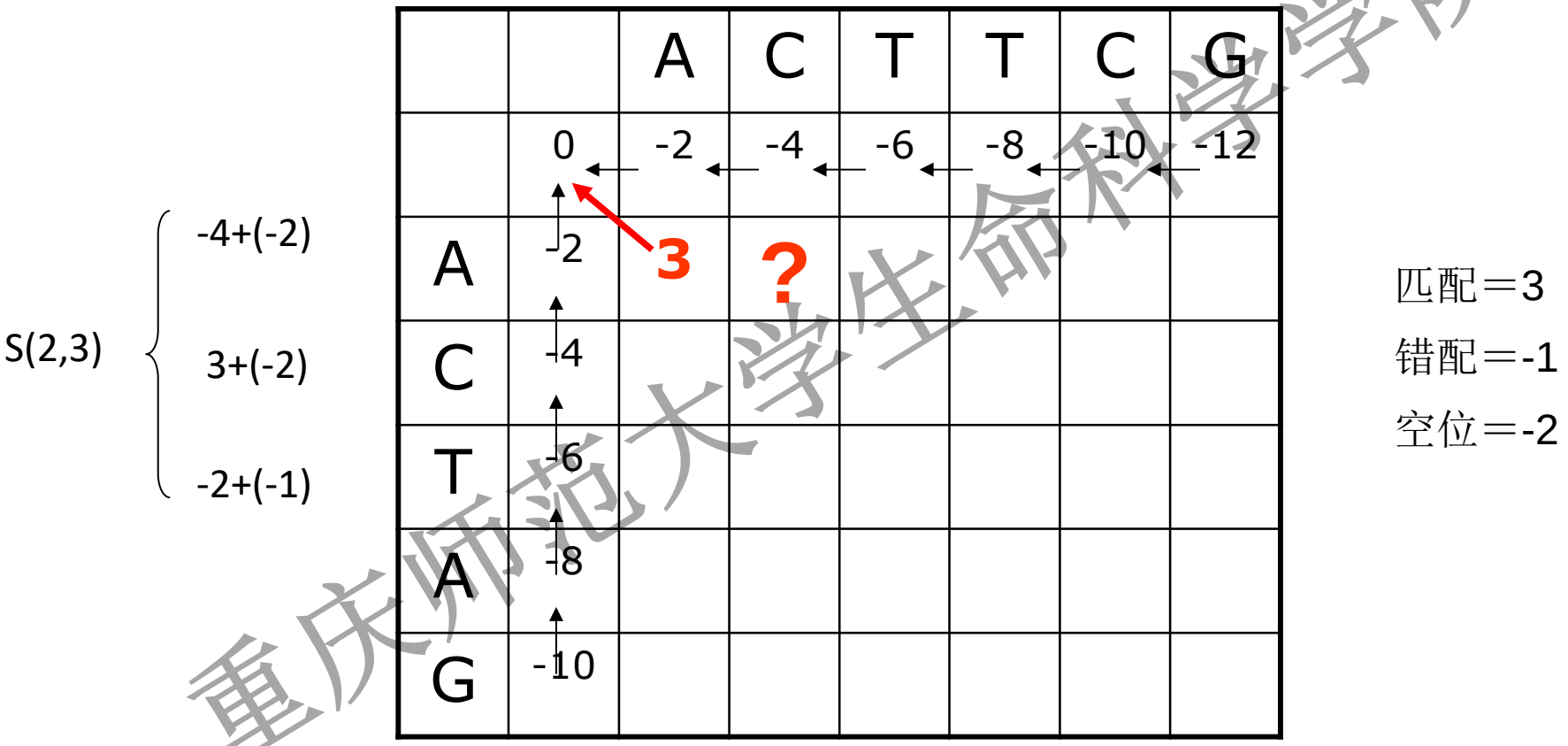
错配＝-1

空位＝-2

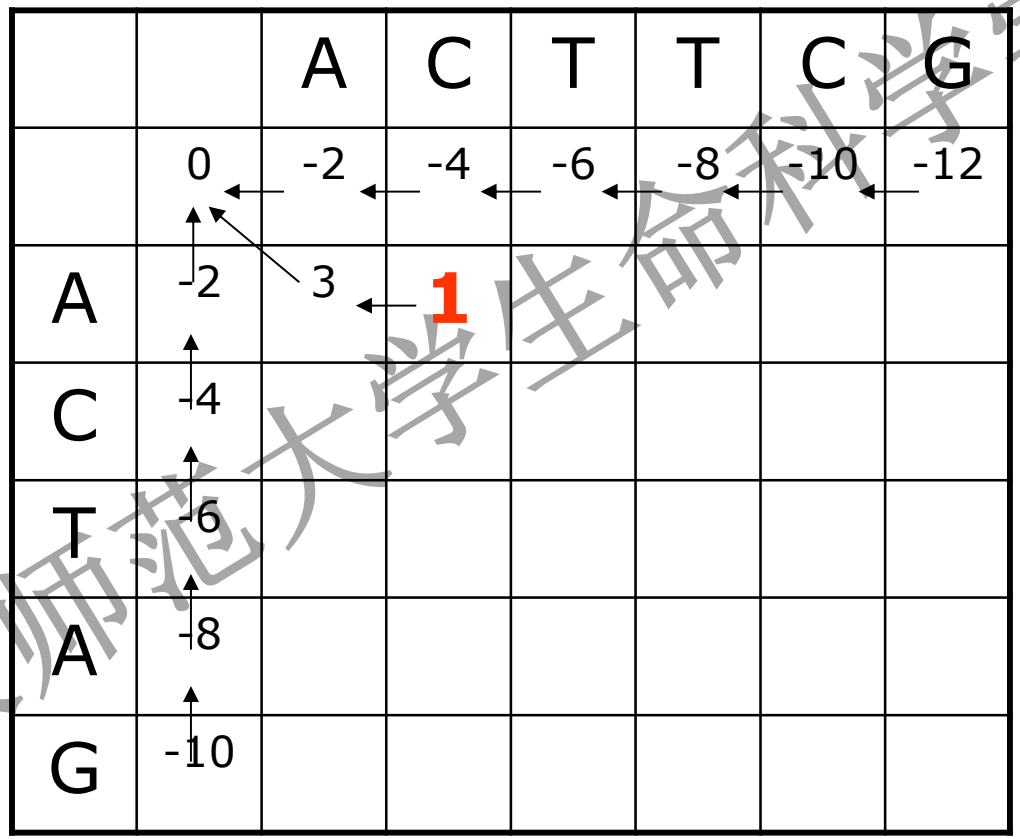
动态规划算法实例



动态规划算法实例



动态规划算法实例

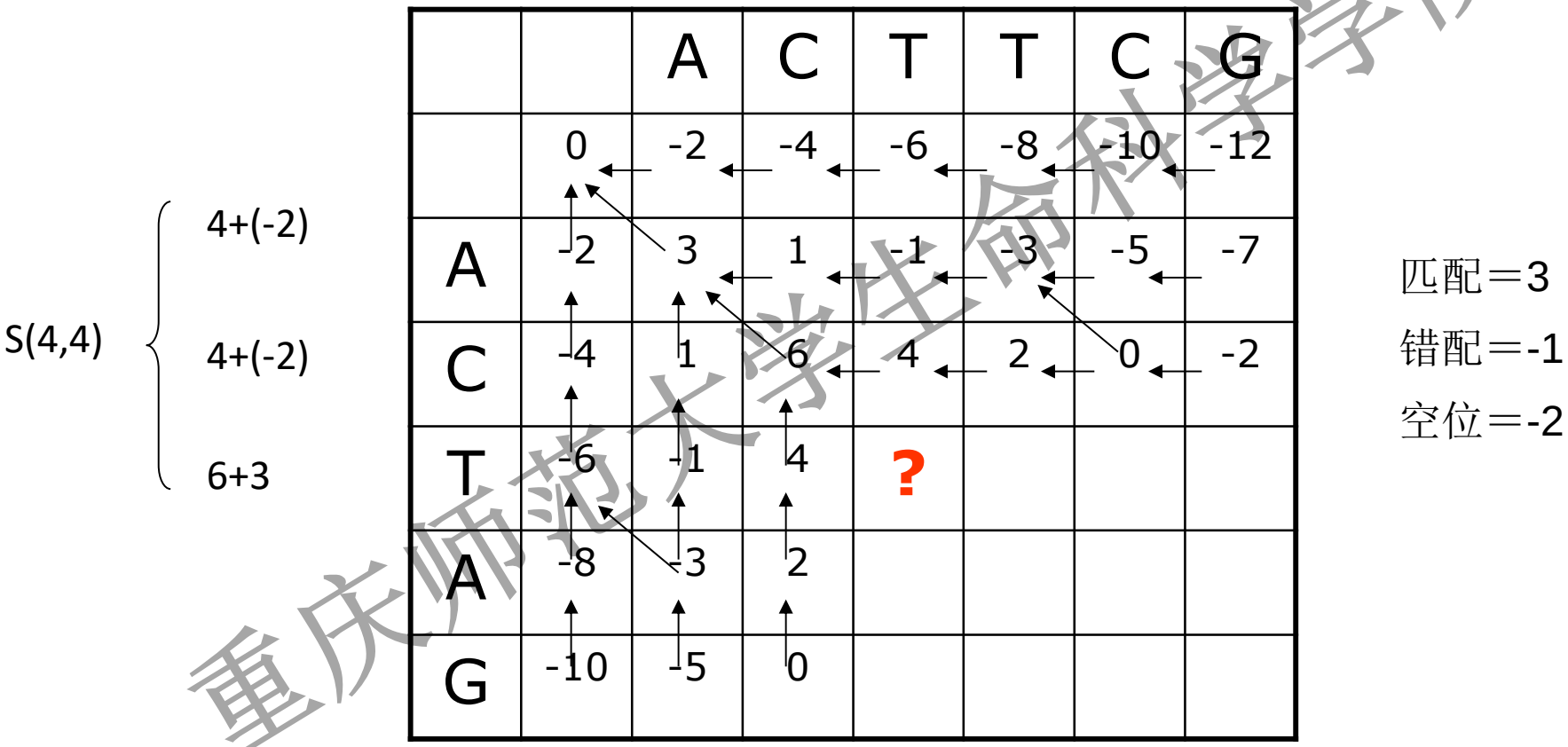


匹配＝3

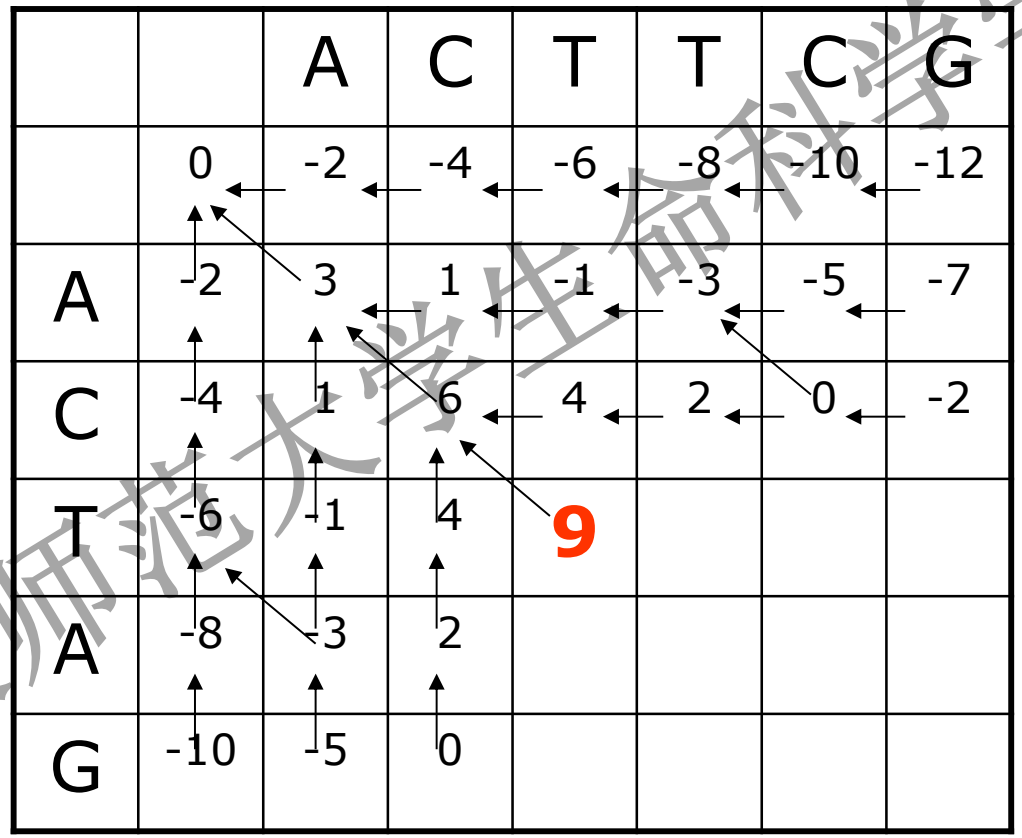
错配＝-1

空位＝-2

动态规划算法实例



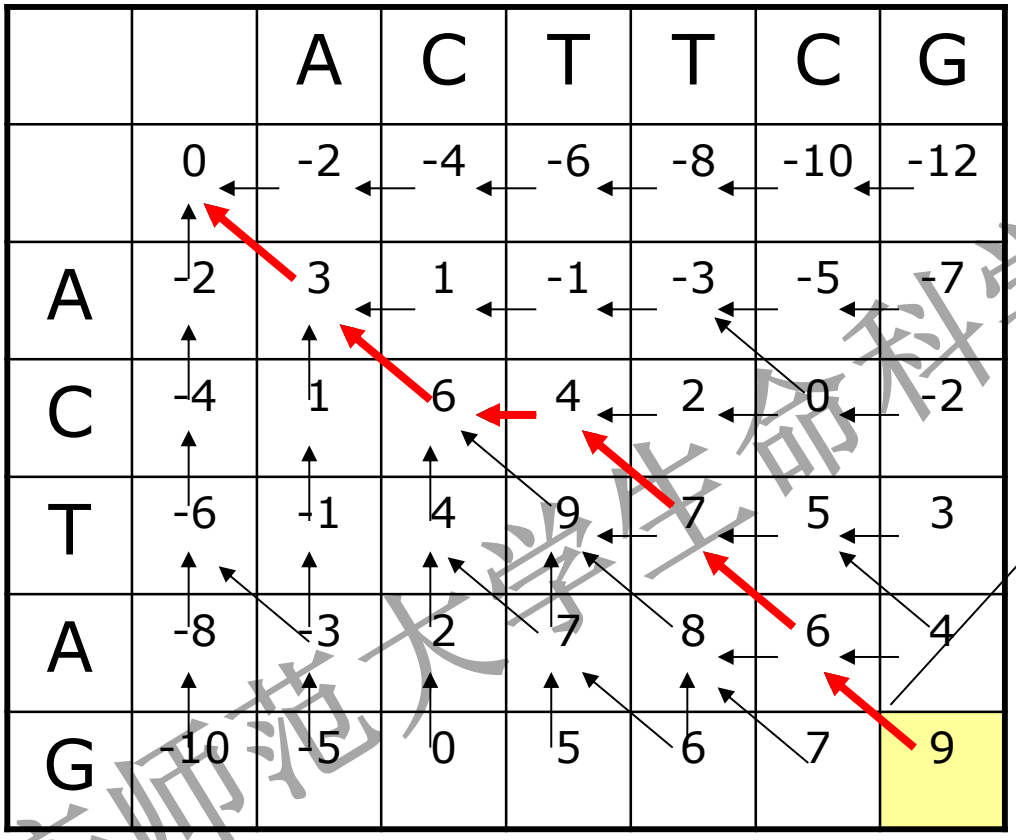
动态规划算法实例



匹配＝3

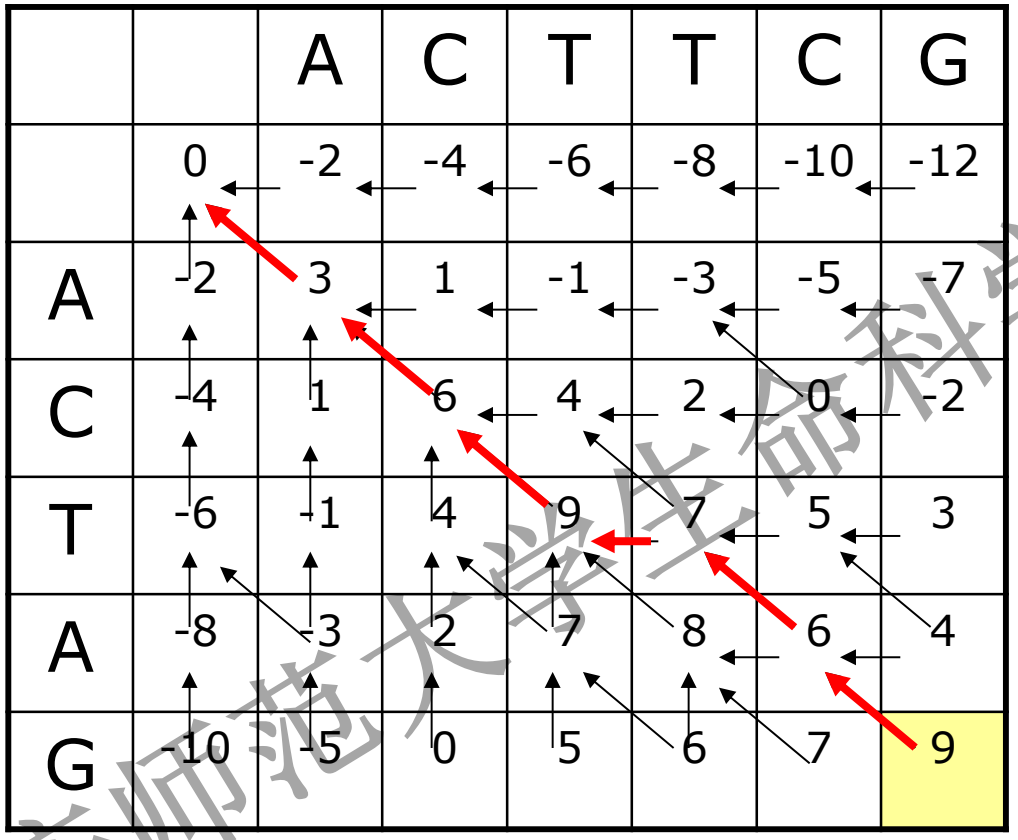
错配＝-1

空位＝-2



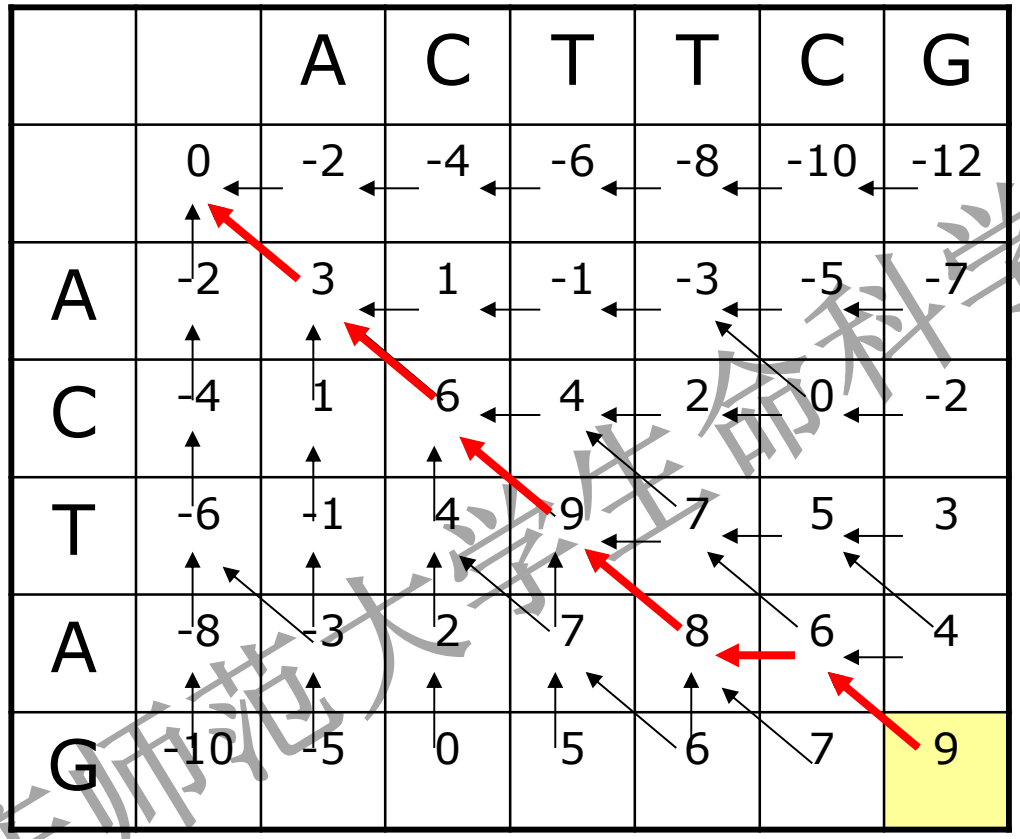
回 溯

A C T T C G
A C - T A G



A C T T C G

A C T - A G



A C T T C G

A C T A - G

比对结果

1. ACTTCG
   AC-TAG

2. ACTTCG
   ACT-AG

3. ACTTCG
   ACTA-G

哪一个是最优比对(optimal alignment)呢?

记分矩阵

计算过程：

	gap	a1	a2	a3	a4
gap	0	1 gap	2 gaps	3 gaps	4 gaps
b1	1 gap	s11			
b2	2 gaps				
b3	3 gaps				
b4	4 gaps				

	gap	a1	a2	a3	a4
gap	0	1 gap	2 gaps	3 gaps	4 gaps
b1	1 gap	s11			
b2	2 gaps	s12			
b3	3 gaps				
b4	4 gaps				

计算过程：

- （1）按行计算

	gap	a1	a2	a3	a4
gap	0	1 gap	2 gaps	3 gaps	4 gaps
b1	1 gap	s11	s21		
b2	2 gaps	s12			
b3	3 gaps				
b4	4 gaps				

- （2）其他方式

	gap	a1	a2	a3	a4
gap	0	1 gap	2 gaps	3 gaps	4 gaps
b1	1 gap	s11	s21		
b2	2 gaps	s12	s22		
b3	3 gaps				
b4	4 gaps				

计算过程：

（3）求最佳路径

	gap	a1	a2	a3	a4
gap	0	1 gap	2 gaps	3 gaps	4 gaps
b1	1 gap	s11	s21	s31	s41
b2	2 gaps	s12	s22	s32	s42
b3	3 gaps	s13	s23	s33	s43
b4	4 gaps	s14	s24	s34	s44

	gap	a1	a2	a3	a4
gap	0	1 gap	2 gaps	3 gaps	4 gaps
b1	1 gap	s11	s21	s31	s41
b2	2 gaps	s12	s22	s32	s42
b3	3 gaps	s13	s23	s33	s43
b4	4 gaps	s14	s24	s34	s44

动态规划算法的数学形式

$$S_{ij}=max\{S_{i-1,j-1},+s(a_ib_j),$$
$$max$$
$$x\geq 1\ (S_{i-x,j}-w_x),$$
$$max$$
$$y\geq 1\ (S_{i,j-y}-w_y)$$
$$\}$$

公式一

$$S_{ij}=max\{S_{i-1,j-1},+s(a_ib_j),$$
$$max$$
$$x\geq 1\ (S_{i-1,j}-w_x),$$
$$max$$
$$y\geq 1\ (S_{i,j-1}-w_y)$$
$$\}$$

公式二

公式一的简化

说明：Sij是序列a在位置i和序列b在位置j的分值， s(aibj)是位置i 和j上比对分值，wx是在序列a 中长度为x的间隔罚分，wy是序列b中长度为y的间隔罚分

Needleman-Wunsch算法

	M	P	R	C	L	C	Q	R	J	N	C	B
P		1										
B												1
R			1					1				
C				1		1					1	
K												
C				1		1					1	
R								1				
N										1		
J									1			
C				1		1					1	
J									1			

Seq1： MPRCLCQRJNCBA
Seq2： PBRCKCRNJCJA

匹配=1，错配=0，空位罚分=0

Needleman-Wunsch算法

	M	P	R	C	L	C	Q	R	J	N	C	B	A
P	0	1	0	0	0	0	0	0	0	0	0	0	0
B	0	0	1	1	1	1	1	1	1	1	1	2	1
R	0	0	2	1	1	1	1	2	1	1	1	1	2
C	0	0	1	3	2	3	2	2	2	2	3	2	2
K	0	0	1	2	3	3	3	3	3	3	3	3	3
C	0	0	1	3	3	4	3	3	3	3	4	3	3
R	0	0	2	2	3	3	4	?					
N													
J									1				
C				1		1					1		
J									1				
A													1

Seq1： MPRCLCQRJNCBA

Seq2： PBRCKCRNJCJA

求出阴影部分所能达到的最大值填入当前位置，并记下到达这一位置的路径

Needleman-Wunsch算法

	M	P	R	C	L	C	Q	R	J	N	C	B	A
P	0	1	0	0	0	0	0	0	0	0	0	0	0
B	0	0	1	1	1	1	1	1	1	1	1	2	1
R	0	0	2	1	1	1	1	2	1	1	1	1	2
C	0	0	1	3	2	3	2	2	2	2	3	2	2
K	0	0	1	2	3	3	3	3	3	3	3	3	3
C	0	0	1	3	3	4	3	3	3	3	4	3	3
R	0	0	2	2	3	3	4	5					
N													
J									1				
C				1		1					1		
J									1				
A													1

Seq1: MPRCLCQRJNCBA

Seq2: PBRCKCRNJCJA

Needleman-Wunsch算法

	M	P	R	C	L	C	Q	R	J	N	C	B	A
P	0	1	0	0	0	0	0	0	0	0	0	0	0
B	0	0	1	1	1	1	1	1	1	1	1	2	1
R	0	0	2	1	1	1	1	2	1	1	1	1	2
C	0	0	1	3	2	3	2	2	2	2	3	2	2
K	0	0	1	2	3	3	3	3	3	3	3	3	3
C	0	0	1	3	3	4	3	3	3	3	4	3	3
R	0	0	2	2	3	3	4	5	4	4	4	4	4
N	0	0	1	2	3	3	4	4	5	6	5	5	5
J	0	0	1	2	3	3	4	4	6	5	6	6	6
C	0	0	1	3	3	4	4	4	5	6	7	6	6
J	0	0	1	2	3	3	4	4	6	6	6	7	7
A	0	0	1	2	3	3	4	4	5	6	6	7	8

Result:

MP-RCLCQR-JNCBA
-PBRCKC-RNJ-CJA

- **Needleman-Wunsch算法**（从第一个到最后一个，一个个比，一条序列比个遍，然后得出最优解）

$$F(i,j)=max\begin{cases}F(i-1,j-1)+s(a_i,b_j),\\F(i,j-1)-wy,\\F(i-1,j)-wx.\end{cases}$$

- **Smith-Waterman算法**（回溯的时候，从得分最高的单元格开始，回溯到得分为 0 的单元格为止）

$$F(i,j)=max\begin{cases}F(i-1,j-1)+s(a_i,b_j),\\F(i,j-1)-wy,\\F(i-1,j)-wx.\\0\end{cases}$$

Smith-Waterman 算法

		A	A	C	C	T	A	T	A	G	C	T
	0	0	0	0	0	0	0	0	0	0	0	0
G	0		0	0	0	0	0	0	0	1	0	0
C	0	0	0		1	0	0	0	0	0	2	1
G	0	0	0	0	0	0	0	0	0	1	0	1
A	0	1	1	0	0	0	1	0	1	0	0	0
T	0	0	0	0	0	1	0	2	1	0	0	1
A	0	1	1	0	0	0	2	0	3	2	1	0
T	0	0	0	0	0	1	1	3	2	2	1	2
A	0	1	1	0	0	0	2	2	4	3	2	1

匹配=1

非匹配=-1

空位=-1

Smith-Waterman 算法

		A	A	C	C	T	A	T	A	G	C	T
	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	0	0	0	1	0	0
C	0	0	0	1	1	0	0	0	0	0	2	1
G	0	0	0	0	0	0	0	0	0	1	0	1
A	0	1	1	0	0	0	1	0	1	0	0	0
T	0	0	0	0	0	1	0	2	1	0	0	1
A	0	1	1	0	0	0	2	0	3	2	1	0
T	0	0	0	0	0	1	1	3	2	2	1	2
A	0	1	1	0	0	0	2	2	4	3	2	1

匹配=1

非匹配=-1

空位=-1

A A C - C T A T A G C T

- G C G A T A T A - - -

Smith-Waterman 算法

- ✓ 首先，在初始化阶段，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）。
- ✓ 第二，在填充表格时，如果某个得分为负，那么就用 0 代替，只对得分为正的单元格添加返回指针。
- ✓ 最后，在回溯的时候，从得分最高的单元格开始，回溯到得分为 0 的单元格为止。除此之外，回溯的方式与 Needleman-Wunsch 算法完全相同

A: GAGCG
B: GGCT

$S(a, b) = 10$ (a = b)
$S(a, b) = -3$ (a ≠ b)
$S(a, b) = -5$ (a = "_" or b = "_")

A Needleman-Wunsch

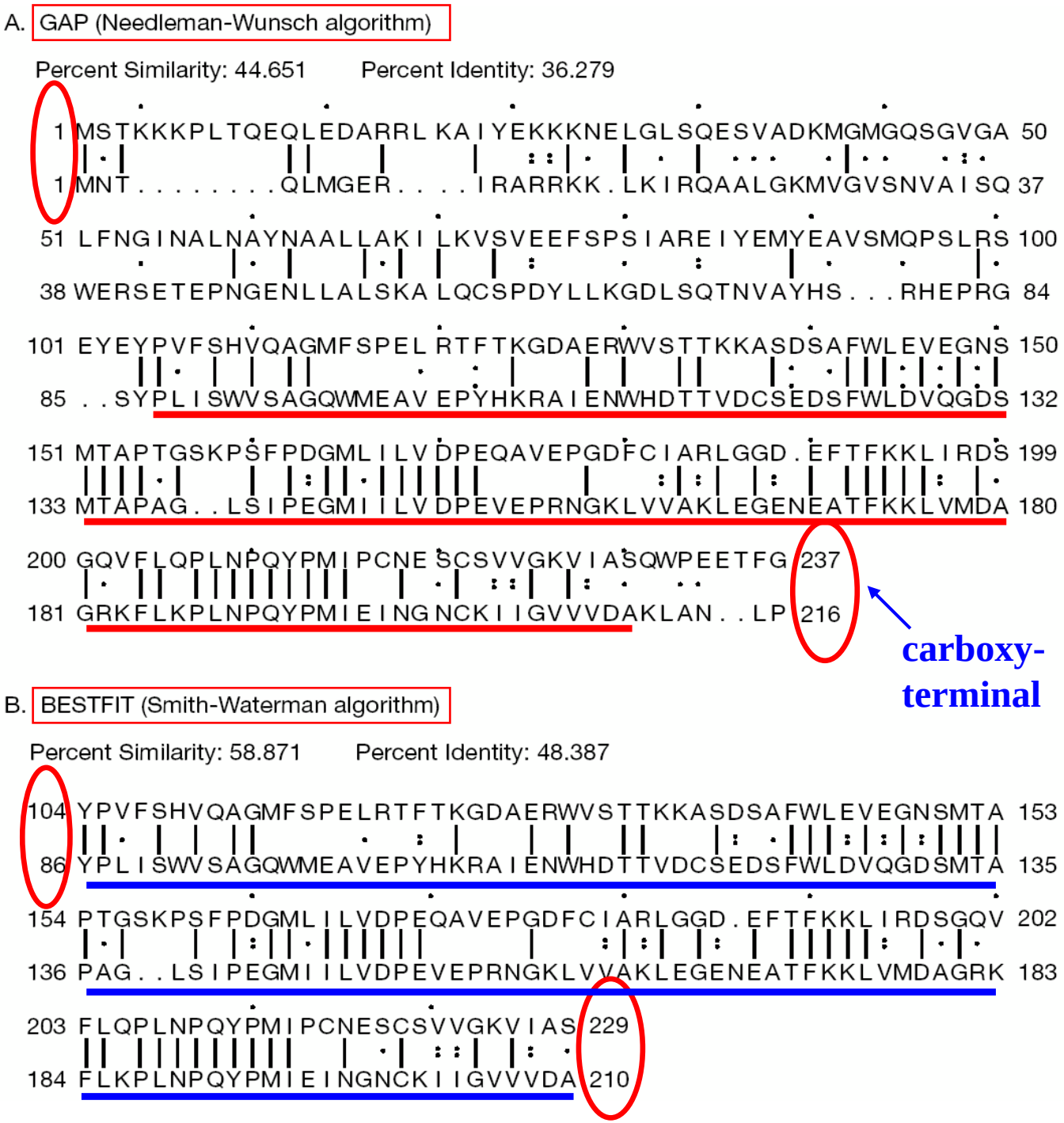
	_	G	A	G	C	G
_	0	-5	-10	-15	-20	-25
G	-5	10	5	0	-5	-10
G	-10	5	7	15	10	5
C	-15	0	2	10	25	20
T	-20	-3	-3	5	20	**22**

G A G C G
G _ G C T

B Smith-Waterman

	_	G	A	G	C	G
_	0	0	0	0	0	0
G	0	10	5	0	0	0
G	0	5	7	15	10	10
C	0	0	2	10	**25**	20
T	0	0	0	5	20	22

G A G C
G _ G C



Genetics Computer Group (GCG) 程序：

GAP (Needleman-Wunsch algorithm) and BESTFIT (Smith-Waterman algorithm)

- 最相似的比对表示为“:”
- 较相似的表示为“.”
- 无相关的表示为空格

算法思想

动态规划算法是运筹学的一个分支，最早用于解决最优化决策问题，其基本思想有点儿类似分治法，将一个大问题拆解为数个子问题，各个子问题顺序求解，最终得到整个大问题的解。这种算法很适合应用到序列比对中，著名的BLAST、CLUSTALW、MFOLD、PHYLIP等都利用了其核心思想，主要包括三个部分：

- **递归**：即将一个大问题转换为多个子问题的过程。对于序列比对来说，长度为N的两条序列大约有$2^{2N}/\sqrt{2\pi N}$种不同的比对方式，这在实际中几乎不可操作，而利用动态规划，将一整条序列变为一个个的碱基进行比对，从计算规模上给予了解决之道。一般性的递归公式为：

$$S(i,j) = max \begin{cases} S(i-1,j-1) + \sigma(x_i, y_j) \\ S(i-1,j) + \gamma \\ S(i,j-1) + \gamma \end{cases}$$

- **动态规划矩阵**：自下而上，记录每个子问题的得分及最佳路径；
- **回溯**：从最后一个位置$S(M,N)$回溯至$(0,0)$，从而得到整条解决路径，但最佳路径可能不止一条。

对于序列比对来讲，动态规划只是从数学上给出了N^2规模的计算方法，然而实际应用中可能面对百万级数量的长度不等的序列，这种时间复杂度就变得不可接受，故BLAST等算法往往是动态规划的一种快速逼近。

第4节：

- 空位罚分
- DNA计分矩阵
- 蛋白质计分矩阵
- 广泛使用的两种矩阵
  - **PAM**
  - **BLOSUM**

记分矩阵(SCORING MATRICES)

- **DNA Scoring Matrices**
- **Amino Acid Substitution Matrices**
  - **PAM (Point Accepted Mutation)**
  - **BLOSUM (Blocks Substitution Matrix)**

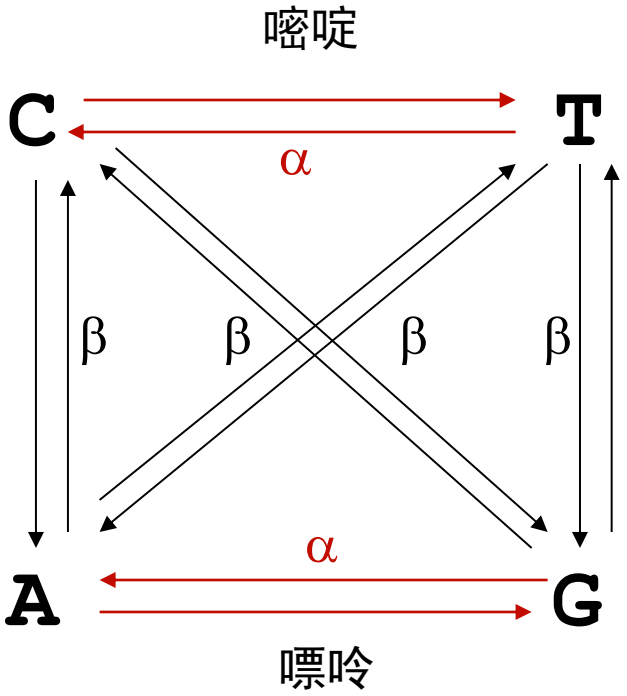
DNA计分矩阵

Sequence 1 actaccagttcatttgatacttctcaaa

Sequence 2 taccattaccgtgttaactgaaaggacttaaagact

	A	G	C	T	
A	**1**	0	0	0	
G	0	**1**	0	0	匹配: 1
C	0	0	**1**	0	错配: 0
T	0	0	0	**1**	分值：5

转换和颠换



- α表示转换(transition)，β表示颠换(transversions)
- 转换比颠换更容易发生

转换和颠换

	A	G	T	C
A	0.99			
G	0.006	0.99		
T	0.002	0.002	0.99	
C	0.002	0.002	0.006	0.99

转换速率是颠换3倍时的模型

蛋白质计分矩阵

```
Sequence 1   PTHPLASKTQILPEDLASEDLTI
             ||||||     |     || ||
Sequence 2   PTHPLAGERAIGLARLAEEDFGM
```

	C	S	T	A	G	P	D	E	Q	N	H	R	K	M	I	L	V	W	Y	F	
C	9																				C
S	-1	4																			S
T	-1	1	5																		T
A	0	1	0	4																	A
G	-3	0	-2	0	6																G
P	-3	-1	-1	-1	-2	7															P
D	-3	0	-1	-2	-1	-1	6														D
E	-4	0	-1	-1	-2	-1	2	5													E
Q	-3	0	-1	-1	-2	-1	0	2	5												Q
N	-3	1	0	-2	0	-2	1	0	0	6											N
H	-3	-1	-2	-2	-2	-2	-1	0	0	1	8										H
R	-3	-1	-1	-1	-2	-2	-2	0	1	0	0	5									R
K	-3	0	-1	-1	-2	-1	-1	1	1	0	-1	2	5								K
M	-1	-1	-1	-1	-3	-2	-3	-2	0	-2	-2	-1	-1	5							M
I	-1	-2	-1	-1	-4	-3	-3	-3	-3	-3	-3	-3	-3	1	4						I
L	-1	-2	-1	-1	-4	-3	-4	-3	-2	-3	-3	-2	-2	2	2	4					L
V	-1	-2	0	0	-3	-2	-3	-2	-2	-3	-3	-3	-2	1	3	1	4				V
W	-2	-3	-2	-3	-2	-4	-4	-3	-2	-4	-2	-3	-3	-1	-3	-2	-3	11			W
Y	-2	-2	-2	-2	-3	-3	-3	-2	-1	-2	2	-2	-2	-1	-1	-1	-1	2	7		Y
F	-2	-2	-2	-2	-3	-4	-3	-3	-3	-3	-1	-3	-3	0	0	0	-1	1	3	6	F
	C	S	T	A	G	P	D	E	Q	N	H	R	K	M	I	L	V	W	Y	F	

蛋白质打分矩阵

保守性替换（conservative substitution）

通常在某些位点上有一些氨基酸被另外一些理化特性相似的氨基酸所代替，这种突变可称为保守性替换。保守性替换一般不会影响蛋白质的结构和功能。

与核苷酸序列比对不同，氨基酸序列比对不仅要考虑残基是否相同，还要考虑残基是否相似以及相似的不同程度。

```
LAAWAAAERDCRVSSFRVKENFDKARFSGTWYAMAKKDPE
   |     |:          |   :  ||||  |     :
ALALTCGAQ.IVT..QTMKGLDIQKV.GTWYSLAMAASD
```

1. 极性中性氨基酸（亲水，含羟基）

2. 碱性氨基酸

氨基酸分类

alanine	丙氨酸	Ala	A	非极性疏水性氨基酸（芳香族氨基酸F、W、Y）
valine	缬氨酸	Val	V	
leucine	亮氨酸	Leu	L	
isoleucine	异亮氨酸	Ile	I	
phenylalanine	苯丙氨酸	Phe	F	
proline	脯氨酸	Pro	P	
methionine	甲硫氨酸	Met	M	
glycine	甘氨酸	Gly	G	极性中性氨基酸（含硫氨基酸C、M）
tryptophan	色氨酸	Trp	W	
serine	丝氨酸	Ser	S	
tyrosine	酪氨酸	Tyr	Y	
cysteine	半胱氨酸	Cys	C	
asparagine	天冬酰氨	Asn	N	
glutarmine	谷氨酰胺	Gln	Q	
threonine	苏氨酸	Thr	T	
aspartic acid	天冬氨酸	Asp	D	酸性氨基酸
glutarmic acid	谷氨酸	Glu	E	
arginine	精氨酸	Arg	R	碱性氨基酸
histidine	组氨酸	His	H	
lysine	赖氨酸	Lys	K	

王镜岩编《生物化学》（第三版）第127页

1. 非极性 R 基氨基酸

这一组共有 8 种氨基酸。4 种带有脂肪烃侧链的氨基酸，即丙氨酸、缬氨酸、亮氨酸和异亮氨酸（图 3-2）；两种含芳香环氨基酸：苯丙氨酸和色氨酸（图 3-6）；一种含硫氨基酸即甲硫氨酸（图 3-3）和一种亚氨基酸，脯氨酸（图 3-5）。这组氨基酸在水中的溶解度比极性 R 基氨基酸小。这组氨基酸中以丙氨酸的 R 基疏水性为最小，它介于非极性 R 基氨基酸和不带电荷的极性 R 基氨基酸之间。

2. 不带电荷的极性 R 基氨基酸

这一组有 7 种氨基酸。这组氨基酸比非极性 R 基氨基酸易溶于水。它们的侧链中含有不解离的极性基，能与水形成氢键。丝氨酸、苏氨酸（图 3-3）和酪氨酸（图 3-6）中侧链的极性是由于它们的羟基造成的；天冬酰胺和谷氨酰胺（图 3-4）的 R 基极性是它们的酰胺基引起的；半胱氨酸（图 3-3）则是由于含有巯基（—SH）的缘故。甘氨酸的侧链介于极性与非极性之间，有时也把它归入非极性类，但是它的 R 基只不过是一个氢原子，对极性强的 α-氨基和 α-羧基影响很小。这一组氨基酸中半胱氨酸和酪氨酸的 R 基极性最强。半胱氨酸的巯基和酪氨酸的酚羟基，虽然在 pH 7 时电离很弱，但与这组中的其他氨基酸侧链相比失去质子的倾向要大得多，例如半胱氨酸的—SH。

3. 带正电荷的 R 基氨基酸

这是一组碱性氨基酸，在 pH 7 时携带正净电荷。属于碱性氨基酸的有赖氨酸、精氨酸和组氨酸（图 3-5）。赖氨酸除 α-氨基外，在侧链的 ε 位置上还有一个$-\overset{+}{N}H_3$；精氨酸含有一个带正电荷的胍基；组氨酸有一个弱碱性的咪唑基。在 pH 6.0 时，组氨酸分子 50% 以上质子化，但在 pH 7.0 时，质子化的分子不到 10%。组氨酸是唯一一个 R 基的 pK_a 值在 7 附近的氨基酸。

4. 带负电荷的 R 基氨基酸

属于这一组的是两种酸性氨基酸：天冬氨酸和谷氨酸（图 3-4）。这两种氨基酸都含有两个羧基，并且第二个羧基在 pH 7 左右也完全解离，因此分子带负电荷。

不同物种3磷酸甘油醛脱氢酶多序列比对

```
果蝇   GAKKVIISAP SAD.APM..F VCGVNLDAYK PDMKVVSNAS CTTNCLAPLA
人类   GAKRVIISAP SAD.APM..F VMGVNHEKYD NSLKIISNAS CTTNCLAPLA
植物   GAKKVIISAP SAD.APM..F VVGVNEHTYQ PNMDIVSNAS CTTNCLAPLA
细菌   GAKKVVMTGP SKDNTPM..F VKGANFDKY. AGQDIVSNAS CTTNCLAPLA
酵母   GAKKVVITAP SS.TAPM..F VMGVNEEKYT SDLKIVSNAS CTTNCLAPLA
古细   GAKKVLISAP PKGDEPVKQL VYGVNHDEYD GE.DVVSNAS CTTNSITPVA

果蝇   KVINDNFEIV EGLMTTVHAT TATQKTVDGP SGKLWRDGRG AAQNIIPAST
人类   KVIHDNFGIV EGLMTTVHAI TATQKTVDGP SGKLWRDGRG ALQNIIPAST
植物   KVVHEEFGIL EGLMTTVHAT TATQKTVDGP SMKDWRGGRG ASQNIIPSST
细菌   KVINDNFGII EGLMTTVHAT TATQKTVDGP SHKDWRGGRG ASQNIIPSST
酵母   KVINDAFGIE EGLMTTVHSL TATQKTVDGP SHKDWRGGRT ASGNIIPSST
古细   KVLDEEFGIN AGQLTTVHAY TGSQNLMDGP NGKP.RRRRA AAENIIPTST

果蝇   GAAKAVGKVI PALNGKLTGM AFRVPTPNVS VVDLTVRLGK GASYDEIKAK
人类   GAAKAVGKVI PELNGKLTGM AFRVPTANVS VVDLTCRLEK PAKYDDIKKV
植物   GAAKAVGKVL PELNGKLTGM AFRVPTSNVS VVDLTCRLEK GASYEDVKAA
细菌   GAAKAVGKVL PELNGKLTGM AFRVPTPNVS VVDLTVRLEK AATYEQIKAA
酵母   GAAKAVGKVL PELQGKLTGM AFRVPTVDVS VVDLTVKLNK ETTYDEIKKV
古细   GAAQAATEVL PELEGKLDGM AIRVPVPNGS ITEFVVDLDD DVTESDVNAA
```

人类lipocalin（脂质运载蛋白）家族多序列比对

```
~~~~~EIQDVSGTWYAMTVDREFPEMNLESVTPMTLTTL.GGNLEAKVTM
LSFTLEEEDITGTWYAMVVDKDFPEDRRRKVSPVKVTALGGGNLEATFTF
TKQDLELPKLAGTWHSMAMATNNISLMATLKAPLRVHITSEDNLEIVLHR
VQENFDVNKYLGRWYEIEKIPTTFENGRCIQANYSLMENGNQELRADGTV
VKENFDKARFSGTWYAMAKDPEGLFLQDNIVAEFSVDETGNWDVCADGTF
LQQNFQDNQFQGKWYVVGLAGNAI.LREDKDPQKMYATIDKSYNVTSVLF
VQPNFQQDKFLGRWFSAGLASNSSWLREKKAALSMCKSVDGGLNLTSTFL
VQENFNISRIYGKWYNLAIGSTCPWMDRMTVSTLVLGEGEAEISMTSTRW
PKANFDAQQFAGTWLLVAVGSACRFLQRAEATTLHVAPQGSTFRKLD...
```

GXW模体

蛋白质打分矩阵

- 我们想要衡量氨基酸配对的相似性程度，这就需要有氨基酸相似性的定量标准。
- 单一打分矩阵满足不了此种需求。
- 相似性打分矩阵，是基于远距离进化过程中观察到的残基替换率，并用不同的分数值表征不同残基之间的相似性程度。恰当选择相似性分数矩阵，可以提高序列比对的敏感度。
- **PAM矩阵**和**BLOSUM矩阵**。

PAM矩阵

- **Margaret Dayhoff** 等研究了**34种蛋白质超家族**（**85%**以上一致性的序列），通过这些**同源蛋白序列**的比对，总结出一个氨基酸被另一个氨基酸替换的概率，从而构建出**PAM矩阵**。

- **PAM（accepted point mutation）**

  可接受点突变

  同源蛋白质在进化过程中会出现一个氨基酸被另一个氨基酸替换的现象，若此种突变通过自然选择被种群接受，并可见于后代的基因组中，便称为可接受点突变。

不同物种3磷酸甘油醛脱氢酶多序列比对

果蝇	GAKKVIISAP	SAD.APM..F	VCGVNLDAYK	PDMKVVSNAS	CTTNCLAPLA
人类	GAKRVIISAP	SAD.APM..F	VMGVNHEKYD	NSLKIISNAS	CTTNCLAPLA
植物	GAKKVIISAP	SAD.APM..F	VVGVNEHTYQ	PNMDIVSNAS	CTTNCLAPLA
细菌	GAKKVVMTGP	SKDNTPM..F	VKGANFDKY.	AGQDIVSNAS	CTTNCLAPLA
酵母	GAKKVVITAP	SS.TAPM..F	VMGVNEEKYT	SDLKIVSNAS	CTTNCLAPLA
古细	GAKKVLISAP	PKGDEPVKQL	VYGVNHDEYD	GE.DVVSNAS	CTTNSITPVA

果蝇	KVINDNFEIV	EGLMTTVHAT	TATQKTVDGP	SGKLWRDGRG	AAQNIIPAST
人类	KVIHDNFGIV	EGLMTTVHAI	TATQKTVDGP	SGKLWRDGRG	ALQNIIPAST
植物	KVVHEEFGIL	EGLMTTVHAT	TATQKTVDGP	SMKDWRGGRG	ASQNIIPSST
细菌	KVINDNFGII	EGLMTTVHAT	TATQKTVDGP	SHKDWRGGRG	ASQNIIPSST
酵母	KVINDAFGIE	EGLMTTVHSL	TATQKTVDGP	SHKDWRGGRT	ASGNIIPSST
古细	KVLDEEFGIN	AGQLTTVHAY	TGSQNLMDGP	NGKP.RRRRA	AAENIIPTST

果蝇	GAAKAVGKVI	PALNGKLTGM	AFRVPTPNVS	VVDLTVRLGK	GASYDEIKAK
人类	GAAKAVGKVI	PELNGKLTGM	AFRVPTANVS	VVDLTCRLEK	PAKYDDIKKV
植物	GAAKAVGKVL	PELNGKLTGM	AFRVPTSNVS	VVDLTCRLEK	GASYEDVKAA
细菌	GAAKAVGKVL	PELNGKLTGM	AFRVPTPNVS	VVDLTVRLEK	AATYEQIKAA
酵母	GAAKAVGKVL	PELQGKLTGM	AFRVPTVDVS	VVDLTVKLNK	ETTYDEIKKV
古细	GAAQAATEVL	PELEGKLDGM	AIRVPVPNGS	ITEFVVDLDD	DVTESDVNAA

1). Dayhoff's 可接受点突变数目（×10）

	A Ala	R Arg	N Asn	D Asp	C Cys	Q Gln	E Glu	G Gly
A								
R	30							
N	109	17						
D	154	0	532					
C	33	10	0	0				
Q	93	120	50	76	0			
E	266	0	94	831	0	422		
G	579	10	156	162	10	30	112	
H	21	103	226	43	10	243	23	10

表示在所研究的同源蛋白中，天冬氨酸被谷氨酸替换，发生了8310次

Dayhoff (1978)

2)、氨基酸出现频率

Gly	8.9%	Arg	4.1%
Ala	8.7%	Asn	4.0%
Leu	8.5%	Phe	4.0%
Lys	8.1%	Gln	3.8%
Ser	7.0%	Ile	3.7%
Val	6.5%	His	3.4%
Thr	5.8%	Cys	3.3%
Pro	5.1%	Tyr	3.0%
Glu	5.0%	Met	1.5%
Asp	4.7%	Trp	1.0%

- blue=6 codons; red=1 codon

3).氨基酸的相对突变几率

每种氨基酸发生突变的次数除以该氨基酸出现的总次数

Asn	134	His	66
Ser	120	Arg	65
Asp	106	Lys	56
Glu	102	Pro	56
Ala	100	Gly	49
Thr	97	Tyr	41
Ile	96	Phe	41
Met	94	Leu	40
Gln	93	Cys	20
Val	74	Trp	18

Note that alanine is normalized to a value of 100.

Trp and Cys are least mutable.

Asn and Ser are most mutable.

- Dayhoff 等根据前述观察到的数据（可接受点突变数目、氨基酸出现频率、氨基酸的相对突变几率）构建出PAM1突变概率矩阵。
- PAM突变概率矩阵是PAM打分矩阵的基础。
- PAM用来表示相对的进化时间。
- PAM1表示一个PAM进化时间，即两个同源蛋白序列有1%氨基酸发生变化的时间
- PAM1突变概率矩阵反映了近缘关系（85%氨基酸一致性）蛋白之间氨基酸替换的规律。

PAM1 突变概率矩阵（万分之一）

Original amino acid

	A Ala	R Arg	N Asn	D Asp	C Cys	Q Gln	E Glu	G Gly	H His	I Ile
A	9867	2	9	10	3	8	17	21	2	6
R	1	9913	1	0	1	10	0	0	10	3
N	4	1	9822	36	0	4	6	6	21	3
D	6	0	42	9859	0	6	53	6	4	1
C	1	1	0	0	9973	0	0	0	1	1
Q	3	9	4	5	0	9876	27	1	23	1
E	10	0	7	56	0	35	9865	4	2	3
G	21	1	12	11	1	3	7	9935	1	0
H	1	8	18	3	1	20	1	0	9912	0
I	2	2	3	1	2	1	2	0	0	9872

表示一个PAM进化时间内同源序列中的丙氨酸有0.21%的可能被替换为甘氨酸

- 利用矩阵的乘法，可将**PAM1矩阵**自乘若干次得到其他的**PAM**矩阵。比如**PAM1矩阵**自乘**250**次便得到**PAM250矩阵**。
- **PAM**后面的数值越大，表示氨基酸的变化越大，进化距离越远。
- **PAM250**表示两个同源蛋白序列中，每100个氨基酸有**250**次变化。
- **PAM250突变概率矩阵**反映了**远缘**关系（**20%**氨基酸一致性）蛋白之间氨基酸替换的规律。

PAM250 突变概率矩阵（%）

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	13	6	9	9	5	8	9	12	6	8	6	7	7	4	11	11	11	2	4	9
R	3	17	4													4	3	7	2	2
N	4	4	6													5	4	2	3	3
D	5	4	8													5	5	1	2	3
C	2	1	1													3	2	1	4	2
Q	3	5	5													3	3	1	2	3
E	5	4	7		1	9	12	5	6	3	2	5	3	1	4	5	5	1	2	3
G	12	5	10	10	4	7	9	27	5	5	4	6	5	3	8	11	9	2	3	7
H	2	5	5	4	2	7	4	2	15	2	2	3	2	2	3	3	2	2	3	2
I	3	2	2	2	2	2	2	2	2	10	6	2	6	5	2	3	4	1	3	9
L	6	4	4	3	2	6	4	3	5	15	34	4	20	13	5	4	6	6	7	13
K	6	18	10	8	2	10	8	5	8	5	4	24	9	2	6	8	8	4	3	5
M	1	1	1	1	0	1	1	1	1	2	3	2	6	2	1	1	1	1	1	2
F	2	1	2	1	1	1	1	1	3	5	6	1	4	32	1	2	2	4	20	3
P	7	5	5	4	3	5	4	5	5	3	3	4	3	2	20	6	5	1	2	4
S	9	6	8	7	7	6	7	9	6	5	4	7	5	3	9	10	9	4	4	6
T	8	5	6	6	4	5	5	6	4	6	4	6	5	3	6	8	11	2	3	6
W	0	2	0	0	0	0	0	0	1	0	1	0	0	1	0	1	0	55	1	0
Y	1	1	2	1	3	1	1	1	3	2	2	1	2	15	1	2	2	3	31	2
V	7	4	4	4	4	4	4	5	4	15	10	4	10	5	5	5	7	2	4	17

同源序列中的丙氨酸有12%的可能被替换为甘氨酸

- 人和黑猩猩同源蛋白的比对，属近缘关系的比较， PAM 1可反映其氨基酸替换的规律。
- 人和细菌同源蛋白的比对，属远缘关系的比较，PAM 250可反映其氨基酸替换的规律。
- PAM后面的数值越大，表示氨基酸的变化越大，进化距离越远。

- 研究PAM矩阵的目的是要在序列比对时，构建一个评价两条序列相关性的打分系统。
- 为了便于打分，Dayhoff 将PAM突变概率矩阵进行对数转换，从而构建出了可以实际应用的PAM打分矩阵。

PAM250打分矩阵
（用于远缘关系比对）

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	2																			
R	-2	6																		
N	0	0	2																	
D	0	-1	2	4																
C	-2	-4	-4	-5	12															
Q	0	1	1	2	-5	4														
E	0	-1	1	3	-5	2	4													
G	1	-3	0	1	-3	-1	0	5												
H	-1	2	2	1	-3	3	1	-2	6											
I	-1	-2	-2	-2	-2	-2	-2	-3	-2	5										
L	-2	-3	-3	-4	-6	-2	-3	-4	-2	2	6									
K	-1	3	1	0	-5	1	0	-2	0	-2	-3	5								
M	-1	0	-2	-3	-5	-1	-2	-3	-2	2	4	0	6							
F	-3	-4	-3	-6	-4	-5	-5	-5	-2	1	2	-5	0	9						
P	1	0	0	-1	-3	0	-1	0	0	-2	-3	-1	-2	-5	6					
S	1	0	1	0	0	-1	0	1	-1	-1	-3	0	-2	-3	1	2				
T	1	-1	0	0	-2	-1	0	0	-1	0	-2	0	-1	-3	0	1	3			
W	-6	2	-4	-7	-8	-5	-7	-7	-3	-5	-2	-3	-4	0	-6	-2	-5	17		
Y	-3	-4	-2	-4	0	-4	-4	-5	0	-1	-1	-4	-2	7	-5	-3	-3	0	10	
V	0	-2	-2	-2	-2	-2	-2	-1	-2	4	2	-2	2	-1	-1	-1	0	-6	-2	4

PAM250 突变概率矩阵（%）

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	13	6	9	9	5	8	9	12	6	8	6	7	7	4	11	11	11	2	4	9
R	3	17	4	3	2	5	3	2	6	3	2	9	4	1	4	4	3	7	2	2
N	4	4	6	7	2	5	6	4	6	3	2	5	3	2	4	5	4	2	3	3
D	5	4	8	11	1	7	10	5	6	3	2	5	3	1	4	5	5	1	2	3
C	2	1	1	1	52	1	1	2	2	2	1	1	1	1	2	3	2	1	4	2
Q	3	5	5	6	1	10	7	3	7	2	3	5	3	1	4	3	3	1	2	3
E	5	4	7	11	1	9	12	5	6	3	2	5	3	1	4	5	5	1	2	3
G	12	5	10	10	4	7	9	27	5	5	4	6	5	3	8	11	9	2	3	7
H	2	5	5	4	2	7	4	2	15	2	2	3	2	2	3	3	2	2	3	2
I	3	2	2	2	2	2	2	2	2	10	6	2	6	5	2	3	4	1	3	9
L	6	4	4	3	2	6	4	3	5	15	34	4	20	13	5	4	6	6	7	13
K	6	18	10	8	2	10	8	5	8	5	4	24	9	2	6	8	8	4	3	5
M	1	1	1	1	0	1	1	1	1	2	3	2	6	2	1	1	1	1	1	2
F	2	1	2	1	1	1	1	1	3	5	6	1	4	32	1	2	2	4	20	3
P	7	5	5	4	3	5	4	5	5	3	3	4	3	2	20	6	5	1	2	4
S	9	6	8	7	7	6	7	9	6	5	4	7	5	3	9	10	9	4	4	6
T	8	5	6	6	4	5	5	6	4	6	4	6	5	3	6	8	11	2	3	6
W	0	2	0	0	0	0	0	0	1	0	1	0	0	1	0	1	0	55	1	0
Y	1	1	2	1	3	1	1	1	3	2	2	1	2	15	1	2	2	3	31	2
V	7	4	4	4	4	4	4	5	4	15	10	4	10	5	5	5	7	2	4	17

PAM250打分矩阵
（用于远缘关系比对）

氨基酸匹配少，氨基酸替换会得到较少的罚分，最终会得到一个较高的分数。

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	2																			
R	-2	6																		
N	0	0	2																	
D	0	-1	2	4																
C	-2	-4	-4	-5	12															
Q	0	1	1	2	-5	4														
E	0	-1	1	3	-5	2	4													
G	1	-3	0	1	-3	-1	0	5												
H	-1	2	2	1	-3	3	1	-2	6											
I	-1	-2	-2	-2	-2	-2	-2	-3	-2	5										
L	-2	-3	-3	-4	-6	-2	-3	-4	-2	-2	6									
K	-1	3	1	0	-5	1	0	-2	0	-2	-3	5								
M	-1	0	-2	-3	-5	-1	-2	-3	-2	2	4	0	6							
F	-3	-4	-3	-6	-4	-5	-5	-5	-2	1	2	-5	0	9						
P	1	0	0	-1	-3	0	-1	0	0	-2	-3	-1	-2	-5	6					
S	1	0	1	0	0	-1	0	1	-1	-1	-3	0	-2	-3	1	2				
T	1	-1	0	0	-2	-1	0	0	-1	0	-2	0	-1	-3	0	1	3			
W	-6	2	-4	-7	-8	-5	-7	-7	-3	-5	-2	-3	-4	0	-6	-2	-5	17		
Y	-3	-4	-2	-4	0	-4	-4	-5	0	-1	-1	-4	-2	7	-5	-3	-3	0	10	
V	0	-2	-2	-2	-2	-2	-2	-1	-2	4	2	-2	2	-1	-1	-1	0	-6	-2	4

PAM(Percent Accepted Mutation)矩阵

- 氨基酸容易被其它生化、物理特性相似的氨基酸替换
- PAM1(1个PAM单位）被定义为每100个残基出现一个被接受的点突变(氨基酸的置换不引起蛋白质功能上的显著变化）
- PAMn是PAM1自乘n次
- PAM250、PAM120、PAM80和PAM60矩阵可用于相似性分别为20%、40%、50%和60%的序列比对

PAM 250

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	B	Z
A	2	-2	0	0	-2	0	0	1	-1	-1	-2	-1	-1	-3	1	1	1	-6	-3	0	2	1
R	-2	6	0	-1	-4	1	-1	-3	2	-2	-3	3	0	-4	0	0	-1	2	-4	-2	1	2
N	0	0	2	2	-4	1	1	0	2	-2	-3	1	-2	-3	0	1	0	-4	-2	-2	4	3
D	0	-1	2	4	-5	2	3	1	1	-2	-4	0	-3	-6	-1	0	0	-7	-4	-2	5	4
C	-2	-4	-4	-5	12	-5	-5	-3	-3	-2	-6	-5	-5	-4	-3	0	-2	-8	0	-2	-3	-4
Q	0	1	1	2	-5	4	2	-1	3	-2	-2	1	-1	-5	0	-1	-1	-5	-4	-2	3	5
E	0	-1	1	3	-5	2	4	0	1	-2	-3	0	-2	-5	-1	0	0	-7	-4	-2	4	5
G	1	-3	0	1	-3	-1	0	5	-2	-3	-4	-2	-3	-5	0	1	0	-7	-5	-1	2	1
H	-1	2	2	1	-3	3	1	-2	6	-2	-2	0	-2	-2	0	-1	-1	-3	0	-2	3	3
I	-1	-2	-2	-2	-2	-2	-2	-3	-2	5	2	-2	2	1	-2	-1	0	-5	-1	4	-1	-1
L	-2	-3	-3	-4	-6	-2	-3	-4	-2	2	6	-3	4	2	-3	-3	-2	-2	-1	2	-2	-1
K	-1	3	1	0	-5	1	0	-2	0	-2	-3	5	0	-5	-1	0	0	-3	-4	-2	2	2
M	-1	0	-2	-3	-5	-1	-2	-3	-2	2	4	0	6	0	-2	-2	-1	-4	-2	2	-1	0
F	-3	-4	-3	-6	-4	-5	-5	-5	-2	1	2	-5	0	9	-5	-3	-3	0	7	-1	-3	-4
P	1	0	0	-1	-3	0	-1	0	0	-2	-3	-1	-2	-5	6	1	0	-6	-5	-1	1	1
S	1	0	1	0	0	-1	0	1	-1	-1	-3	0	-2	-3	1	2	1	-2	-3	-1	2	1
T	1	-1	0	0	-2	-1	0	0	-1	0	-2	0	-1	-3	0	1	3	-5	-3	0	2	1
W	-6	2	-4	-7	-8	-5	-7	-7	-3	-5	-2	-3	-4	0	-6	-2	-5	17	0	-6	-4	-4
Y	-3	-4	-2	-4	0	-4	-4	-5	0	-1	-1	-4	-2	7	-5	-3	-3	0	10	-2	-2	-3
V	0	-2	-2	-2	-2	-2	-2	-1	-2	4	2	-2	2	-1	-1	-1	0	-6	-2	4	0	0
B	2	1	4	5	-3	3	4	2	3	-1	-2	2	-1	-3	1	2	2	-4	-2	0	6	5
Z	1	2	3	4	-4	5	5	1	3	-1	-1	2	0	-4	1	1	1	-4	-3	0	5	6

BLOSUM矩阵（Henikoff夫妇，1992）

- **PAM矩阵**的产生是基于相似性较高（**85%**以上）的序列比对，那些进化距离较远的矩阵（如**PAM250**）是从初始模型中推算出来而不是直接计算得到的，其准确性受到一定限制。
- 而序列分析的关键是检测进化距离较远的序列之间是否具有同源性，因此**PAM矩阵**在实际使用时存在一定的局限。

- **BLOSUM矩阵（blocks substitution matrix）**

  即模块替换矩阵。与PAM矩阵相比，BLOSUM矩阵是根据进化距离较远的蛋白序列模块（保守区域）比对直接计算得到的。

- 因此，BLOSUM矩阵比PAM矩阵总的来说要好，尤其是BLOSUM62被大多数比对搜索工具选作为默认的打分矩阵。

- BLOSUM62来自于≥62％相似度的序列比对。

- BLOSUM80来自于≥80％相似度的序列比对。

BLOSUM62

	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W	
C	9																				C
S	−1	4																			S
T	−1	1	5																		T
P	−3	−1	−1	7																	P
A	0	1	0	−1	4																A
G	−3	0	−2	−2	0	6															G
N	−3	1	0	−2	−2	0	6														N
D	−3	0	−1	−1	−2	−1	1	6													D
E	−4	0	−1	−1	−1	−2	0	2	5												E
Q	−3	0	−1	−1	−1	−2	0	0	2	5											Q
H	−3	−1	−2	−2	−2	−2	1	−1	0	0	8										H
R	−3	−1	−1	−2	−1	−2	0	−2	0	1	0	5									R
K	−3	0	−1	−1	−1	−2	0	−1	1	1	−1	2	5								K
M	−1	−1	−1	−2	−1	−3	−2	−3	−2	0	−2	−1	−1	5							M
I	−1	−2	−1	−3	−1	−4	−3	−3	−3	−3	−3	−3	−3	1	4						I
L	−1	−2	−1	−3	−1	−4	−3	−4	−3	−2	−3	−2	−2	2	2	4					L
V	−1	−2	0	−2	0	−3	−3	−3	−2	−2	−3	−3	−2	1	3	1	4				V
F	−2	−2	−2	−4	−2	−3	−3	−3	−3	−3	−1	−3	−3	0	0	0	−1	6			F
Y	−2	−2	−2	−3	−2	−3	−2	−3	−2	−1	2	−2	−2	−1	−1	−1	−1	3	7		Y
W	−2	−3	−2	−4	−3	−2	−4	−4	−3	−2	−2	−3	−3	−1	−3	−2	−3	1	2	11	W
	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W	

总结

- **（1）核酸打分矩阵设DNA序列所用的字母表为**

  A = { A，C，G，T }

a. 等价矩阵 (unitary matrix)

b. BLAST矩阵

c. 转换-颠换矩阵（transition-transversion matrix）

（嘌呤：腺嘌呤A，鸟嘌呤G；嘧啶：胞嘧啶C，胸腺嘧啶T）

表3.1 等价矩阵表

	A	T	C	G
A	1	0	0	0
T	0	1	0	0
C	0	0	1	0
G	0	0	0	1

表3.2 BLAST矩阵

	A	T	C	G
A	5	-4	-4	-4
T	-4	5	-4	-4
C	-4	-4	5	-4
G	-4	-4	-4	5

表3.3 转移矩阵

	A	T	C	G
A	1	-5	-5	-1
T	-5	1	-1	-5
C	-5	-1	1	-5
G	-1	-5	-5	1

（2）蛋白质打分矩阵

- （i）等价矩阵

$$R_{ij}=\begin{cases}1 & i=j\\ 0 & i\neq j\end{cases}$$

其中R_{ij}代表打分矩阵元素
i、j分别代表字母表第i和第j个字符。

- （ii） 遗传密码矩阵（**genetic code matrix,GCM)**
- （iii）疏水性矩阵 （**hydrophobic matrix)**
- （iv）**PAM**矩阵（**point accepted matrix,PAM**）
- （v） **BLOSUM**矩阵

（**BLOck SUbstitution Matrix，BLOSUM**）

遗传密码矩阵

遗传密码矩阵通过计算一个氨基酸变成另一个氨基酸所需的密码子变化的数目而得到。通常为1 或 2，只有Met到Tyr为 3。

First letter of codon (5′ end)	Second letter of codon: U		C		A		G	
U	UUU	Phe	UCU	Ser	UAU	Tyr	UGU	Cys
	UUC	Phe	UCC	Ser	UAC	Tyr	UGC	Cys
	UUA	Leu	UCA	Ser	UAA	Stop	UGA	Stop
	UUG	Leu	UCG	Ser	UAG	Stop	UGG	Trp
C	CUU	Leu	CCU	Pro	CAU	His	CGU	Arg
	CUC	Leu	CCC	Pro	CAC	His	CGC	Arg
	CUA	Leu	CCA	Pro	CAA	Gln	CGA	Arg
	CUG	Leu	CCG	Pro	CAG	Gln	CGG	Arg
A	AUU	Ile	ACU	Thr	AAU	Asn	AGU	Ser
	AUC	Ile	ACC	Thr	AAC	Asn	AGC	Ser
	AUA	Ile	ACA	Thr	AAA	Lys	AGA	Arg
	AUG	Met	ACG	Thr	AAG	Lys	AGG	Arg
G	GUU	Val	GCU	Ala	GAU	Asp	GGU	Gly
	GUC	Val	GCC	Ala	GAC	Asp	GGC	Gly
	GUA	Val	GCA	Ala	GAA	Glu	GGA	Gly
	GUG	Val	GCG	Ala	GAG	Glu	GGG	Gly

GCM矩阵

	A	S	G	L	K	V	T	P	E	D	N	I	Q	R	F	Y	C	H	M	W	Z	B	X
A	0	1	1	2	2	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2
S	1	0	1	1	2	2	1	1	2	2	1	1	2	1	1	1	1	2	2	1	2	2	2
G	1	1	0	2	2	1	2	2	1	1	2	2	2	1	2	2	1	2	2	1	2	2	2
L	2	1	2	0	2	1	2	1	2	2	2	1	1	1	1	2	2	1	1	1	2	2	2
K	2	2	2	2	0	2	1	2	1	2	1	1	1	1	2	2	2	2	1	2	1	2	2
V	1	2	1	1	2	0	2	2	1	1	2	1	2	2	1	2	2	2	1	2	2	2	2
T	1	1	2	2	1	2	0	1	2	2	1	1	2	1	2	2	2	2	1	2	2	2	2
P	1	1	2	1	2	2	1	0	2	2	2	2	1	1	2	2	2	1	2	2	2	2	2
E	1	2	1	2	1	1	2	2	0	1	2	2	1	2	2	2	2	2	2	2	1	2	2
D	1	2	1	2	2	1	2	2	1	0	1	2	2	2	2	1	2	1	2	2	2	1	2
N	2	1	2	2	1	2	1	2	2	1	0	1	2	2	2	1	2	1	2	2	2	1	2
I	2	1	2	1	1	1	1	2	2	2	1	0	2	1	1	2	2	2	1	2	2	2	2
Q	2	2	2	1	1	2	2	1	1	2	2	2	0	1	2	2	2	1	2	2	1	2	2
R	2	1	1	1	1	2	1	1	2	2	2	1	1	0	2	2	1	1	1	1	2	2	2
F	2	1	2	1	2	1	2	2	2	2	2	1	2	2	0	1	1	2	2	2	2	2	2
Y	2	1	2	2	2	2	2	2	2	1	1	2	2	2	1	0	1	1	3	2	2	1	2
C	2	1	2	2	2	2	2	2	2	2	2	2	2	1	1	1	0	2	2	1	2	2	2
H	2	2	2	1	2	2	2	1	2	1	1	2	1	1	2	1	2	0	2	2	2	1	2
M	2	2	2	1	1	1	1	2	2	2	2	1	2	1	2	3	2	2	0	2	2	2	2
W	2	1	1	1	2	2	2	2	2	2	2	2	2	1	2	2	1	2	2	0	2	2	2
Z	2	2	2	2	1	2	2	2	1	2	2	2	1	2	2	2	2	2	2	2	1	2	2
B	2	2	2	2	2	2	2	2	2	1	1	2	2	2	2	1	2	1	2	2	2	1	2
X	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2

疏水矩阵

	R	K	D	E	B	Z	S	N	Q	G	X	T	H	A	C	M	P	V	L	I	Y	F	W
R	10	10	9	9	8	8	6	6	6	5	5	5	5	5	4	3	3	3	3	3	2	1	0
K	10	10	9	9	8	8	6	6	6	5	5	5	5	5	4	3	3	3	3	3	2	1	0
D	9	9	10	10	8	8	7	6	6	6	5	5	5	5	5	4	4	4	3	3	3	2	1
E	9	9	10	10	8	8	7	6	6	6	5	5	5	5	5	4	4	4	3	3	3	2	1
B	8	8	8	8	10	10	8	8	8	8	7	7	7	7	6	6	6	5	5	5	4	4	3
Z	8	8	8	8	10	10	8	8	8	8	7	7	7	7	6	6	6	5	5	5	4	4	3
S	6	6	7	7	8	8	10	10	10	10	9	9	9	9	8	8	7	7	7	7	6	6	4
N	6	6	6	6	8	8	10	10	10	10	9	9	9	9	8	8	8	7	7	7	6	6	4
Q	6	6	6	6	8	8	10	10	10	10	9	9	9	9	8	8	8	7	7	7	6	6	4
G	5	5	6	6	8	8	10	10	10	10	9	9	9	9	8	8	8	8	7	7	6	6	5
X	5	5	5	5	7	7	9	9	9	9	10	10	10	10	9	9	8	8	8	8	7	7	5
T	5	5	5	5	7	7	9	9	9	9	10	10	10	10	9	9	8	8	8	8	7	7	5
H	5	5	5	5	7	7	9	9	9	9	10	10	10	10	9	9	9	8	8	8	7	7	5
A	5	5	5	5	7	7	9	9	9	9	10	10	10	10	9	9	9	8	8	8	7	7	5
C	4	4	5	5	6	6	8	8	8	8	9	9	9	9	10	10	9	9	9	9	8	8	5
M	3	3	4	4	6	6	8	8	8	8	9	9	9	9	10	10	10	10	9	9	8	8	7
P	3	3	4	4	6	6	7	8	8	8	8	8	9	9	9	10	10	10	9	9	9	8	7
V	3	3	4	4	5	5	7	7	7	8	8	8	8	8	9	10	10	10	10	10	9	8	7
L	3	3	3	3	5	5	7	7	7	7	8	8	8	8	9	9	9	10	10	10	9	9	8
I	3	3	3	3	5	5	7	7	7	7	8	8	8	8	9	9	9	10	10	10	9	9	8
Y	2	2	3	3	4	4	6	6	6	6	7	7	7	7	8	8	9	9	9	9	10	10	8
F	1	1	2	2	4	4	6	6	6	6	7	7	7	7	8	8	8	8	9	9	10	10	9
W	0	0	1	1	3	3	4	4	4	5	5	5	5	5	6	7	7	7	8	8	8	9	10

如何选择合适的评分矩阵?

- 一般来说,在局部相似性搜索上， **BLOSUM 矩阵较PAM要好**
- **当比较距离相近的蛋白时，应选择低的PAM或高的BLOSUM矩阵；当比较距离较远的蛋白时，应选择高的PAM或低的BLOSUM矩阵**
- 对于数据库搜索来说一般选择**BLOSUM62矩阵**
- **PAM矩阵可用于寻找蛋白质的进化起源，BLOSUM矩阵用于发现蛋白质的保守域**

- 经验法则（针对蛋白质序列）：

① 如果两个序列的长度都大于100，在适当地加入空位之后，它们配对的相同率达到25%以上，则两个序列相关；

② 如果配对的相同率小于15%，则不管两个序列的长度如何，它们都不可能相关；

③ 如果两个序列的相同率在15%~25%之间，它们可能是相关的。

第5节：

数据库的搜索简介

数据库查询为生物学研究提供了一个重要工具，在实际工作中经常使用。在分子生物学研究中，对于新测定的碱基序列或由此翻译得到的氨基酸序列，往往需要通过数据库搜索，找出具有一定相似性的同源序列，以推测该未知序列可能属于哪个基因家族，具有哪些生物学功能。对于氨基酸序列来说，有可能找到已知三维结构的同源蛋白质而推测其可能的空间结构。因此，数据库搜索与数据库查询一样，是生物信息学研究中的一个重要工具。

数据库搜索的基础是序列的相似性比对，即双序列比对(pairwise alignment)。

新测定的、希望通过数据库搜索确定其性质或功能的序列称作检测序列(probe sequence)；通过数据库搜索得到的和检测序列具有一定相似性的序列称目标序列(subject sequence)。

为了确定检测序列和一个已知基因家族之间的进化关系，在通过数据库搜索得到某些相似序列后，还需要判断其序列相似性程度。如果检测序列和目标序列的相似性程度很低，还必须通过其他方法或实验手段才能确定其是否属于同一基因家族 。

一、 BLAST 简介

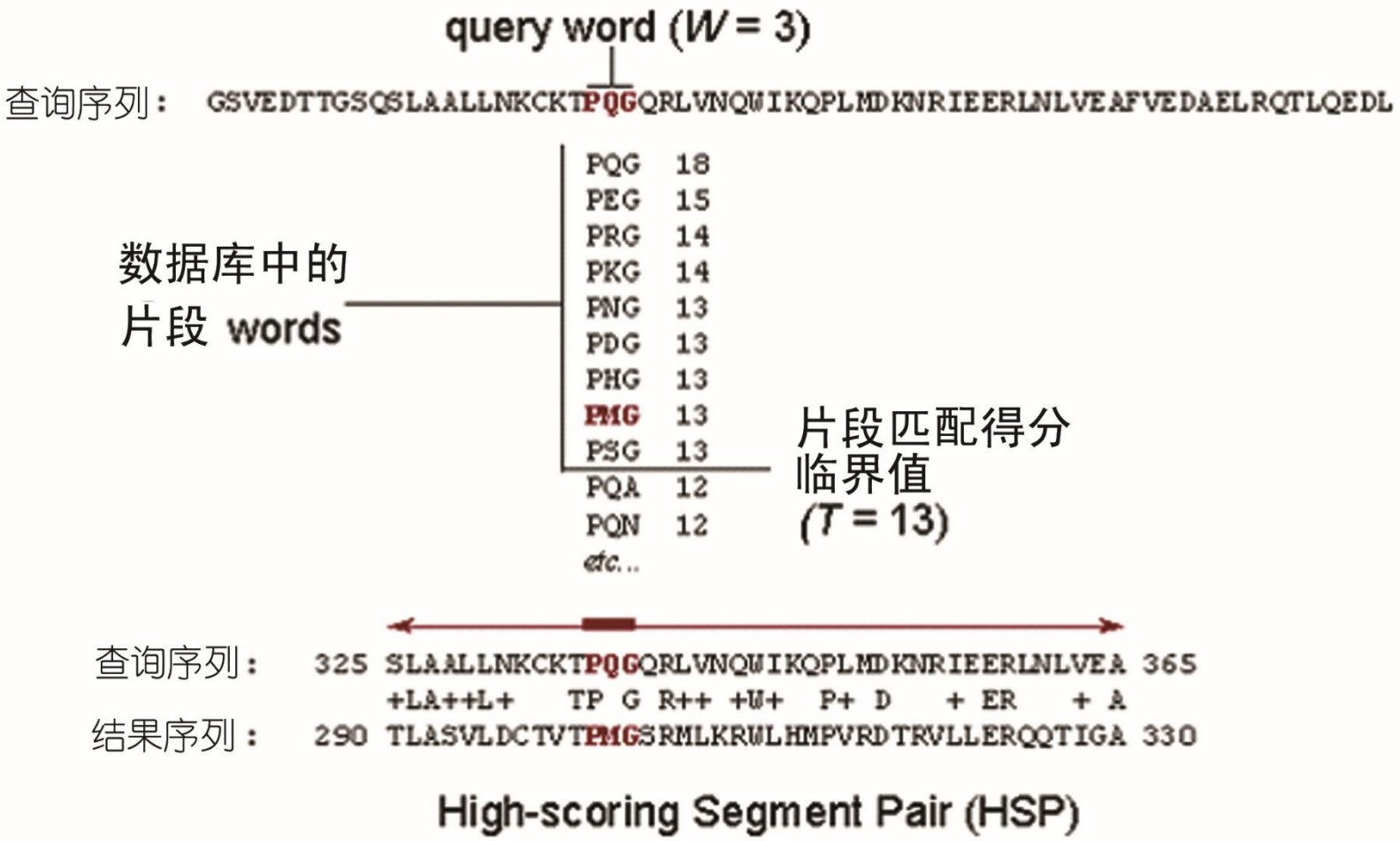
BLAST程序是目前最常用的基于局部相似性的数据库搜索程序，它们都基于查找完全匹配的短小序列片段，并将它们延伸得到较长的相似性匹配。它们的优势在于可以在普通的计算机系统上运行，而不必依赖计算机硬件系统而解决运行速度问题。

BLAST数据库搜索策略

通过部分而不是全部序列计算最适联配值

——赢得搜索速度

BLAST 算法


query word (W = 3)
查询序列： GSVEDTTGSQSLAALLNKCKTPQGQRLVNQWIKQPLMDKNRIEERLNLVEAFVEDAELRQTLQEDL
数据库中的
片段 words
PQG 18
PEG 15
PRG 14
PKG 14
PNG 13
PDG 13
PHG 13
PMG 13
PSG 13
PQA 12
PQN 12
etc...
片段匹配得分
临界值
(T = 13)
查询序列： 325 SLAALLNKCKTPQGQRLVNQWIKQPLMDKNRIEERLNLVEA 365
+LA++L+ TP G R++ +W+ P+ D + ER + A
结果序列： 290 TLASVLDCTVTPMGSRMLKRWLHMPVRDTRVLLERQQTIGA 330
High-scoring Segment Pair (HSP)

比对统计学意义的评价——*E*值(E-Value)

*P*值(*P*-Value)(概率值)

BLAST程序中使用了*E*值而非*P*值，这主要是从直观和便于理解的角度考虑。比如*E*值等于5和10，总比*P*值等于0.993和0.99995更直观。但是当*E*<0.01时，*P*值与*E*值接近相同

参数*K*和λ可分别被简单地视为搜索步长(search spacesize)和计分系统(scoring system)的特征数

BLAST软件包实际上是综合在一起的一组程序，不仅可用于直接对蛋白质序列数据库和核酸序列数据库进行搜索，而且可以将检测序列翻译成蛋白质或将数据库翻译成蛋白质后再进行搜索，以提高搜索结果的灵敏度。

BLAST程序检测序列和数据库类型

程序名	检测序列	数据库类型	方 法
Blastp	蛋白质	蛋白质	用检测序列蛋白质搜索蛋白质序列数据库
Blastn	核酸	核酸	用检测序列核酸搜索核酸序列数据库
Blastx	核酸	蛋白质	将核酸序列按6条链翻译成蛋白质序列后搜索蛋白质序列数据库
Tblastn	蛋白质	核酸	用检测序列蛋白质搜索由核酸序列数据库按6条链翻译成的蛋白质序列数据库
Tblastx	核酸	核酸	将核酸序列按6条链翻译成蛋白质序列后搜索由核酸序列数据库按6条链翻译成的蛋白质序列数据库

对一般用户来说，目前常用的办法是通过NCBI、EBI等国际著名生物信息中心的BLAST服务器进行搜索。需要说明的是，**各生物信息中心BLAST用户界面有所不同**，所提供的数据库也可能不完全相同，使用前最好先进行适当的选择。

BLAST 应用实例

- 多结构域蛋白 **(H1N1)**
- 脂质运载蛋白

多结构域蛋白 (H1N1) 的BLAST检索

多结构域蛋白 (H1N1) 的BLAST检索

H1N1聚合酶序列

>gi|224983683|pdb|3GBN|B Chain B, Crystal Structure Of Fab Cr6261 In Complex With The 1918 H1n1 Influenza Virus Hemagglutinin

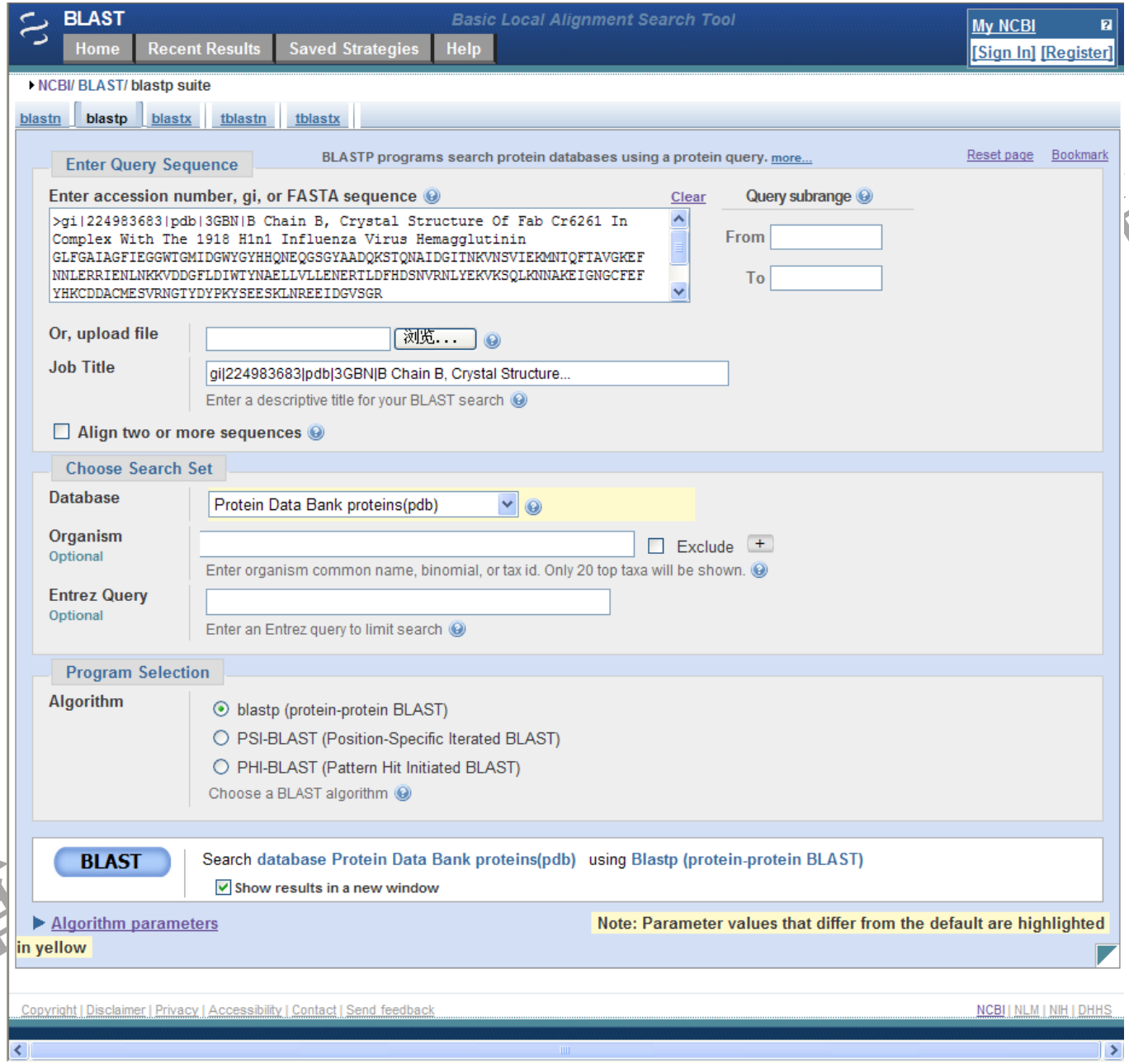
GLFGAIAGFIEGGWTGMIDGWYGYHHQNEQGSGYAADQKS

TQNAIDGITNKVNSVIEKMNTQFTAVGKEF

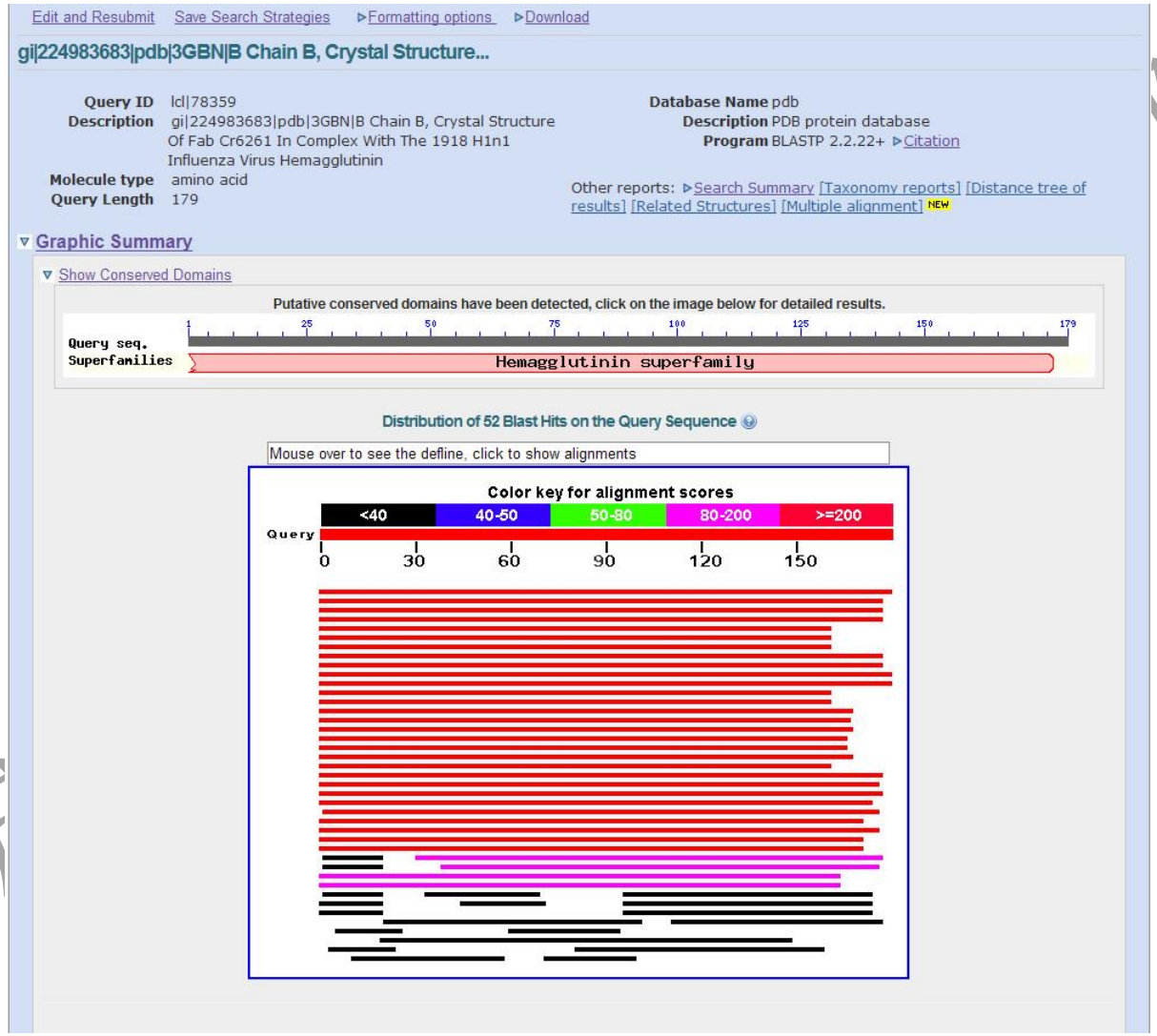
NNLERRIENLNKKVDDGFLDIWTYNAELLVLLENERTLDFH

DSNVRNLYEKVKSQLKNNAKEIGNGCFEF

YHKCDDACMESVRNGTYDYPKYSEESKLNREEIDGVSGR


BLAST
Basic Local Alignment Search Tool
Home
Recent Results
Saved Strategies
Help
My NCBI
[Sign In] [Register]
NCBI/ BLAST/ blastp suite
blastn
blastp
blastx
tblastn
tblastx
Enter Query Sequence
BLASTP programs search protein databases using a protein query. more...
Reset page
Bookmark
Enter accession number, gi, or FASTA sequence
Clear
>gi|224983683|pdb|3GBN|B Chain B, Crystal Structure Of Fab Cr6261 In Complex With The 1918 H1n1 Influenza Virus Hemagglutinin
GLFGAIAGFIEGGWTGMIDGWYGYHHQNEQGSGYAADQKSTQNAIDGITNKVNSVIEKMNTQFTAVGKEF
NNLERRIENLNKKVDDGFLDIWTYNAELLVLLENERTLDFHDSNVRNLYEKVKSQLKNNAKEIGNGCFEF
YHKCDDACMESVRNGTYDYPKYSEESKLNREEIDGVSGR
Query subrange
From
To
Or, upload file
浏览...
Job Title
gi|224983683|pdb|3GBN|B Chain B, Crystal Structure...
Enter a descriptive title for your BLAST search
Align two or more sequences
Choose Search Set
Database
Protein Data Bank proteins(pdb)
Organism
Optional
Exclude
Enter organism common name, binomial, or tax id. Only 20 top taxa will be shown.
Entrez Query
Optional
Enter an Entrez query to limit search
Program Selection
Algorithm
blastp (protein-protein BLAST)
PSI-BLAST (Position-Specific Iterated BLAST)
PHI-BLAST (Pattern Hit Initiated BLAST)
Choose a BLAST algorithm
BLAST
Search database Protein Data Bank proteins(pdb) using Blastp (protein-protein BLAST)
Show results in a new window
Algorithm parameters
Note: Parameter values that differ from the default are highlighted in yellow
Copyright | Disclaimer | Privacy | Accessibility | Contact | Send feedback
NCBI | NLM | NIH | DHHS

BLAST结果综述


Edit and Resubmit
Save Search Strategies
Formatting options
Download
gi|224983683|pdb|3GBN|B Chain B, Crystal Structure...
Query ID lcl|78359
Description gi|224983683|pdb|3GBN|B Chain B, Crystal Structure Of Fab Cr6261 In Complex With The 1918 H1n1 Influenza Virus Hemagglutinin
Molecule type amino acid
Query Length 179
Database Name pdb
Description PDB protein database
Program BLASTP 2.2.22+ Citation
Other reports: Search Summary [Taxonomy reports] [Distance tree of results] [Related Structures] [Multiple alignment] NEW
Graphic Summary
Show Conserved Domains
Putative conserved domains have been detected, click on the image below for detailed results.
Query seq.
Superfamilies
Hemagglutinin superfamily
Distribution of 52 Blast Hits on the Query Sequence
Mouse over to see the defline, click to show alignments
Color key for alignment scores
<40
40-50
50-80
80-200
>=200
Query
0
30
60
90
120
150

BLAST结果表述

Alignments Select All Get selected sequences Distance tree of results Multiple alignment NEW

>pdb|3GBN|B Chain B, Crystal Structure Of Fab Cr6261 In Complex With The 1918 H1n1 Influenza Virus Hemagglutinin
Length=179

Score = 370 bits (951), Expect = 1e-103, Method: Compositional matrix adjust.
Identities = 179/179 (100%), Positives = 179/179 (100%), Gaps = 0/179 (0%)

```
Query  1    GLFGAIAGFIEGGWTGMIDGWYGYHHQNEQGSGYAADQKSTQNAIDGITNKVNSVIEKMN  60
            GLFGAIAGFIEGGWTGMIDGWYGYHHQNEQGSGYAADQKSTQNAIDGITNKVNSVIEKMN
Sbjct  1    GLFGAIAGFIEGGWTGMIDGWYGYHHQNEQGSGYAADQKSTQNAIDGITNKVNSVIEKMN  60

Query  61   TQFTAVGKEFNNLERRIENLNKKVDDGFLDIWTYNAELLVLLENERTLDFHDSNVRNLYE  120
            TQFTAVGKEFNNLERRIENLNKKVDDGFLDIWTYNAELLVLLENERTLDFHDSNVRNLYE
Sbjct  61   TQFTAVGKEFNNLERRIENLNKKVDDGFLDIWTYNAELLVLLENERTLDFHDSNVRNLYE  120

Query  121  KVKSQLKNNAKEIGNGCFEFYHKCDDACMESVRNGTYDYPKYSEESKLNREEIDGVSGR  179
            KVKSQLKNNAKEIGNGCFEFYHKCDDACMESVRNGTYDYPKYSEESKLNREEIDGVSGR
Sbjct  121  KVKSQLKNNAKEIGNGCFEFYHKCDDACMESVRNGTYDYPKYSEESKLNREEIDGVSGR  179
```

>pdb|2WRG|I Chain I, Structure Of H1 1918 Hemagglutinin With Human Receptor
pdb|2WRG|K Chain K, Structure Of H1 1918 Hemagglutinin With Human Receptor
pdb|2WRG|M Chain M, Structure Of H1 1918 Hemagglutinin With Human Receptor
Length=222

Score = 366 bits (940), Expect = 2e-102, Method: Compositional matrix adjust.
Identities = 176/176 (100%), Positives = 176/176 (100%), Gaps = 0/176 (0%)

```
Query  1    GLFGAIAGFIEGGWTGMIDGWYGYHHQNEQGSGYAADQKSTQNAIDGITNKVNSVIEKMN  60
            GLFGAIAGFIEGGWTGMIDGWYGYHHQNEQGSGYAADQKSTQNAIDGITNKVNSVIEKMN
Sbjct  1    GLFGAIAGFIEGGWTGMIDGWYGYHHQNEQGSGYAADQKSTQNAIDGITNKVNSVIEKMN  60

Query  61   TQFTAVGKEFNNLERRIENLNKKVDDGFLDIWTYNAELLVLLENERTLDFHDSNVRNLYE  120
            TQFTAVGKEFNNLERRIENLNKKVDDGFLDIWTYNAELLVLLENERTLDFHDSNVRNLYE
Sbjct  61   TQFTAVGKEFNNLERRIENLNKKVDDGFLDIWTYNAELLVLLENERTLDFHDSNVRNLYE  120

Query  121  KVKSQLKNNAKEIGNGCFEFYHKCDDACMESVRNGTYDYPKYSEESKLNREEIDGV  176
            KVKSQLKNNAKEIGNGCFEFYHKCDDACMESVRNGTYDYPKYSEESKLNREEIDGV
Sbjct  121  KVKSQLKNNAKEIGNGCFEFYHKCDDACMESVRNGTYDYPKYSEESKLNREEIDGV  176
```

>pdb|1RD8|B Chain B, Crystal Sructure Of The 1918 Human H1 Hemagglutinin Precursor (Ha0)
pdb|1RD8|D Chain D, Crystal Sructure Of The 1918 Human H1 Hemagglutinin Precursor (Ha0)

BLAST结果逐条显示

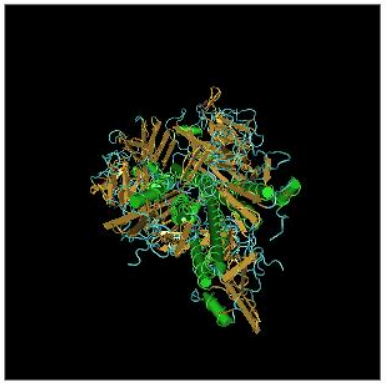
```
>gb|ACS77963.1|  polymerase PA [Influenza A virus (A/New York/3413/2009(H1N1))]
Length=716

 Score = 1489 bits (3854),  Expect = 0.0, Method: Compositional matrix adjust.
 Identities = 715/716 (99%), Positives = 715/716 (99%), Gaps = 0/716 (0%)

Query  1    MEDFVRQCFNPMIVELAEKAMKEYGEDPKIETNKFAAICTHLEVCFMYSDFHFIDERGES  60
            MED VRQCFNPMIVELAEKAMKEYGEDPKIETNKFAAICTHLEVCFMYSDFHFIDERGES
Sbjct  1    MEDLVRQCFNPMIVELAEKAMKEYGEDPKIETNKFAAICTHLEVCFMYSDFHFIDERGES  60

Query  61   IIVESGDPNALLKHRFEIIEGRDRIMAWTVVNSICNTTGVEKPKFLPDLYDYKENRFIEI  120
            IIVESGDPNALLKHRFEIIEGRDRIMAWTVVNSICNTTGVEKPKFLPDLYDYKENRFIEI
Sbjct  61   IIVESGDPNALLKHRFEIIEGRDRIMAWTVVNSICNTTGVEKPKFLPDLYDYKENRFIEI  120

Query  121  GVTRREVHIYYLEKANKIKSEKTHIHIFSFTGEEMATKADYTLDEESRARIKTRLFTIRQ  180
            GVTRREVHIYYLEKANKIKSEKTHIHIFSFTGEEMATKADYTLDEESRARIKTRLFTIRQ
Sbjct  121  GVTRREVHIYYLEKANKIKSEKTHIHIFSFTGEEMATKADYTLDEESRARIKTRLFTIRQ  180

Query  181  EMASRSLWDSFRQSERGEETIEEKFEITGTMRKLADQSLPPNFSSLENFRAYVDGFEPNG  240
            EMASRSLWDSFRQSERGEETIEEKFEITGTMRKLADQSLPPNFSSLENFRAYVDGFEPNG
Sbjct  181  EMASRSLWDSFRQSERGEETIEEKFEITGTMRKLADQSLPPNFSSLENFRAYVDGFEPNG  240

Query  241  CIEGKLSQMSKEVNAKIEPFLRTTPRPLRLPDGPLCHQRSKFLLMDALKLSIEDPSHEGE  300
            CIEGKLSQMSKEVNAKIEPFLRTTPRPLRLPDGPLCHQRSKFLLMDALKLSIEDPSHEGE
Sbjct  241  CIEGKLSQMSKEVNAKIEPFLRTTPRPLRLPDGPLCHQRSKFLLMDALKLSIEDPSHEGE  300

Query  301  GIPLYDAIKCMKTFFGWKEPNIVKPHEKGINPNYLMAWKQVLAELQDIENEEKIPRTKNM  360
            GIPLYDAIKCMKTFFGWKEPNIVKPHEKGINPNYLMAWKQVLAELQDIENEEKIPRTKNM
Sbjct  301  GIPLYDAIKCMKTFFGWKEPNIVKPHEKGINPNYLMAWKQVLAELQDIENEEKIPRTKNM  360

Query  361  KRTSQLKWALGENMAPEKVDFDDCKDVGDLKQYDSDEPEPRSLASWVQNEFNKACELTDS  420
            KRTSQLKWALGENMAPEKVDFDDCKDVGDLKQYDSDEPEPRSLASWVQNEFNKACELTDS
Sbjct  361  KRTSQLKWALGENMAPEKVDFDDCKDVGDLKQYDSDEPEPRSLASWVQNEFNKACELTDS  420

Query  421  SWIELDEIGEDVAPIEHIASMRRNYFTAEVSHCRATEYIMKGVYINTALLNASCAAMDDF  480
```

BLAST结果逐条显示

NCBI

Structure Summary
MMDB

HOME | SEARCH | SITE MAP | Entrez | Structure | Protein | CDD | PubMed | Taxonomy | PubChem | Help | Cn3D



MMDB ID: 26944 **PDB ID:** 1RV0

Search PDB or MMDB ID

Reference: Gamblin SJ, Haire LF, Russell RJ, Stevens DJ, Xiao B, Ha Y, Vasisht N, Steinhauer DA, Daniels RS, Elliot A, Wiley DC, Skehel JJ *The structure and receptor binding properties of the 1918 influenza hemagglutinin* Science v303, p.1838-1842

The 1918 influenza pandemic resulted in about 20 million deaths. This enormous impact, coupled with renewed interest in emerging infections, makes characterization of the virus involved a priority. Receptor binding, the initial event in virus infection, is a major determinant of virus transmissibility that, for influenza viruses, is mediated by the hemagglutinin (HA) membrane glycoprotein....

» View full abstract

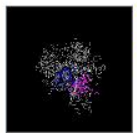
Description: 1930 Swine H1 Hemagglutinin Complexed With Lsta.
Deposition: 2003/12/12
Taxonomy: Influenza A virus
Related Structure: VAST

Structure View in Cn3D | Structure View in RasMol

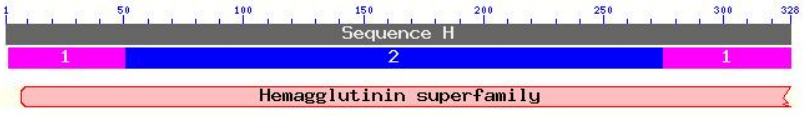
Tasks: Display Drawing: All Atoms

Download Cn3D | View Cn3D Tutorial

Molecular components in the MMDB structure are listed below and may include macromolecular chains, 3D domains, protein classifications (domain families), and ligands, as available. Mouse over each icon for more information on the component.



Protein
3d Domains
Domain Families
Super Families



Score得分值越高说明同源性越好；

Expect期望值越小，则比对结果越好，说明因某些原因引起的误差越小；

Identities是一致性。

E值适合于有一定长度、且复杂度不能太低的序列。

当E小于10^{-5}时，表明两序列有较高的同源性，而不是因为计算错误；

当E小于10^{-6}时，表明两序列的同源性非常高，而且几乎没有必要再做确认。

BLAST：改变打分矩阵的作用

脂质运载蛋白

序列

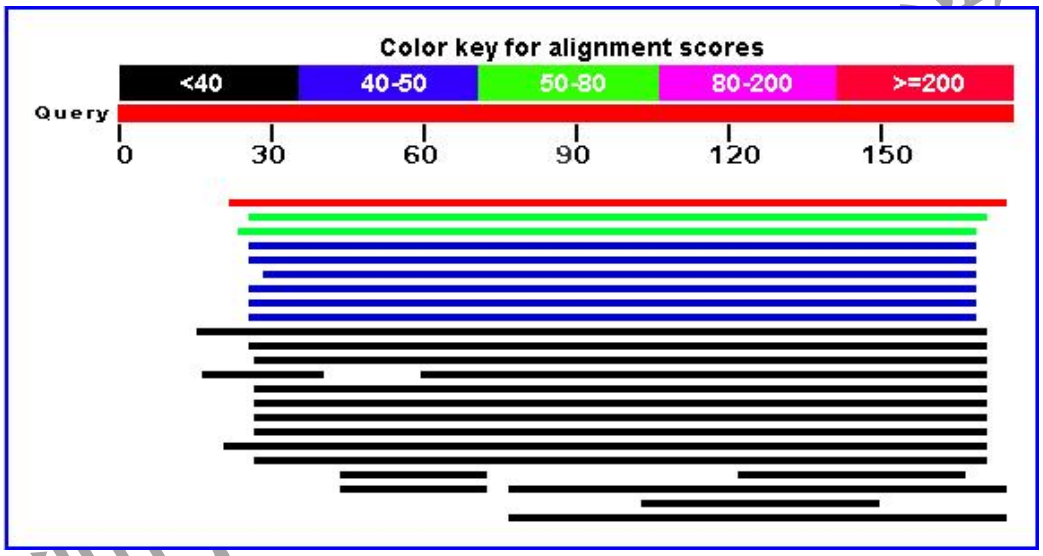
>sp|P31025|LCN1_HUMAN Lipocalin-1 OS=Homo sapiens GN=LCN1 PE=1 SV=1

MKPLLLAVSLGLIAALQAHHLLASDEEIQDVSGTWYLKAMTVDREFPEMNLESVTPMTLT

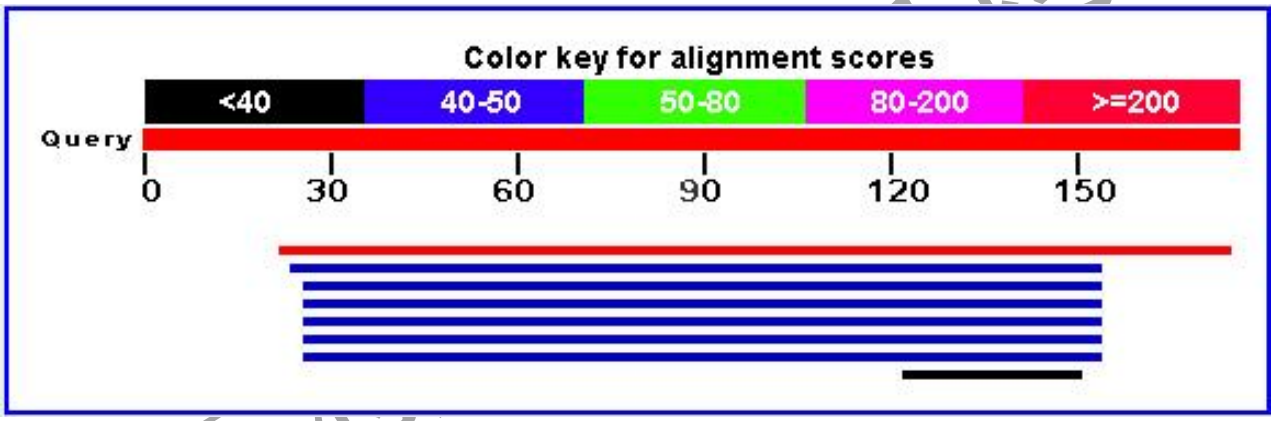
TLEGGNLEAKVTMLISGRCQEVKAVLEKTDEPGKYTADGGKHVAYIIRSHVKDHYIFYCE

GELHGKPVRGVKLVGRDPKNNLEALEDFEKAAGARGLSTESILIPRQSETCSPGSD

使用Blosum62矩阵搜索



使用PAM30矩阵搜索



FastA简介

FastA算法是由**Lipman**和**Pearson**于**1985**年发表的（**Lipman**和**Pearson**，**1985**）。**FastA**的基本思路是识别与代查序列相匹配的很短的序列片段，称为**k-tuple**。

蛋白质序列数据库搜索时，短片段的长度一般是1～2个残基长；DNA序列数据库搜索时，通常采用稍大点的值，最多为6个碱基。通过比较两个序列中的短片段及其相对位置，可以构成一个动态规划矩阵的对角线方向上的一些匹配片段。

FastA程序采用渐进（heuristic approach）算法将位于同一对角线上相互接近的短片段连接起来。也就是说，通过不匹配的残基将这些匹配残基片段连接起来，以便得到较长的相似性片段。这就意味着，FastA输出结果中允许出现不匹配残基。这和BLAST程序中的成对片段类似。如果匹配区域很多，FastA利用动态规划算法在这些匹配区域间插入空位。

由**FastA**搜索产生的典型输出结果的第一行列出程序名称和版本号，以及该程序发表的杂志。接下来列出所提交的序列，然后是所用参数和运行时间，紧跟这些一般信息的是数据库搜索结果。

首先列出搜索得到的目标序列简单说明，其数目可由用户定义。所列出的目标序列的信息包括：序列所在数据库名称的缩写，目标序列的标识码、序列号和序列名等部分信息。括号中标明匹配部分的残基数。紧接着是由程序计算得到的初始化和优化后的分数值。最后一列是期望值即***E***值，用来判断比对结果的置信度。接近于**0**的***E***值表明两序列的匹配不大可能是由随机因素造成的。

Software	Description	Type
Biostrings	Efficient manipulation of biological strings, Pairwise/Multiple Sequence alignment	R package
seqinr	Biological Sequences Retrieval and Analysis	R package
msa	Multiple Sequence Alignment	R package
DECIPHER	Decipher and manage biological sequences; Predict coding/non-coding genes	R package
ggmsa	Plot multiple sequence alignment using ggplot2	R package
AlignStat	Statistical comparison of alternative multiple sequence alignments	R package
BALCONY	MSA and functional compartments of protein variability analysis	R package
kmer	Fast alignment-free clustering of biological sequences	R package
muscle	Multiple Sequence Alignment with MUSCLE	R package
SubVis	Exploring the Effects of Multiple Substitution Matrices on Pairwise Sequence Alignment	R package
odseq	Outlier detection in a multiple sequence alignment	R package
SDSparsimony	Detecting Specificity Determining Sites in a multiple sequence alignment	R package
SequenceBouncer	Remove outlier entries from a multiple sequence alignment	Python package

Please see more by https://bioinformaticshome.com/tools/msa/msa.html

https://en.wikipedia.org/wiki/List_of_RNA-Seq_bioinformatics_tools

https://statsandr.com/blog/top-r-resources-on-covid-19-coronavirus/

Software	Description	Type
UniprotR	Retrieving and visualizing protein sequence and functional information from Universal Protein Resource	R package
spruceup	Flexible identification, visualization, and removal of abnormal sequences from multiple sequence alignments	Python package
rtracklayer	R interface to genome annotation files and the UCSC genome browser	R package
UCSCXenaTools	Accessing Genomics Data from UCSC Xena platform, from Cancer Multi-omics to Single-cell RNA-seq	R package
igvR	igvR: integrative genomics viewer	R package
ggtranscript	visualization and interpretation of transcript isoforms using ggplot2	R package
R-CHIE	Visualizing RNA secondary structures	R package
karyoploteR	Plot customizable linear genomes displaying arbitrary data	R package
XLmap	visualize and score protein structure models based on sites of protein cross-linking	R package
Bio3d	Comparative analysis of protein structures	R package
HPAanalyze	facilitates the retrieval and analysis of the Human Protein Atlas data	R package
MethFinder	prediction of human tissue-specific methylation status of CpG islands	
packFinder	a simple tool for the prediction of potential Pack-TYPE elements	R package
hpar	a simple R interface to and data from the Human Protein Atlas project.	R package
HPAStainR	query protein expression patterns in the Human Protein Atlas	Shiny app

Software	Description	Type
GenProSeq	Generating Protein Sequences with Deep Generative Models	R Package
proteinProfiles	Significance assessment for distance measures of time-course protein profiles	R Package
bcSeq	Fast Sequence Alignment for High-Throughput shRNA and CRISPR Screens (R)	R Package
transite	RNA-binding protein motif analysis	R Package
cisPath	Visualization and management of the protein-protein interaction networks.	R Package
Path2PPI	Prediction of pathway-related protein-protein interaction networks	R Package
customProDB	Generate customized protein database from NGS data, with a focus on RNA-Seq data, for proteomics search	R Package
DAPAR	Tools for the Differential Analysis of Proteins Abundance with R	R Package
drawProteins	allow the creation of protein schematics based on the data obtained from the Uniprot Protein Database.	R Package
GeneDMRs	An R Package for Gene-Based Differentially Methylated Regions Analysis	R Package
methylKit	a comprehensive R package for the analysis of genome-wide DNA methylation profiles	R Package
BOG	R-package for Bacterium and virus analysis of Orthologous Groups	

重庆师范大学
CHONG QING NORMAL UNIVERSIYT

Thanks for your attention!

Acknowledgement

College of Life Sciences, Chongqing Normal University

2022, Chongqing of P. R. C