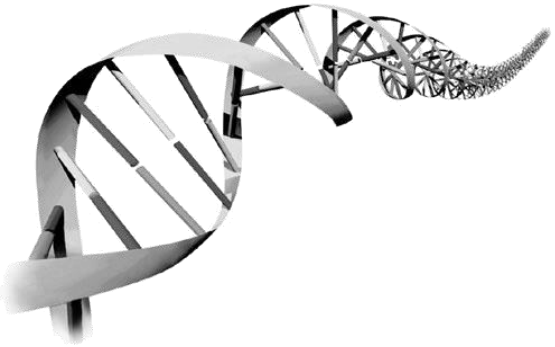


生物信息学

第六章　双序列比对

为什么要序列比对

- Sequence alignment
- 基于同源序列鉴定的功能预测

- 基本假设：

  序列的保守性 ⟷ 功能的保守性

- 注意：
  - 蛋白质一般在三级结构的层面上执行功能
  - 蛋白质序列的保守性决定于其编码DNA的保守性

序列同源模型中的进化假设

- 所有的生物都起源于同一个祖先
- 序列不是随机产生，而是在进化上，不断发生着演变
- 基本假设：

  序列保守性 → 结构保守性

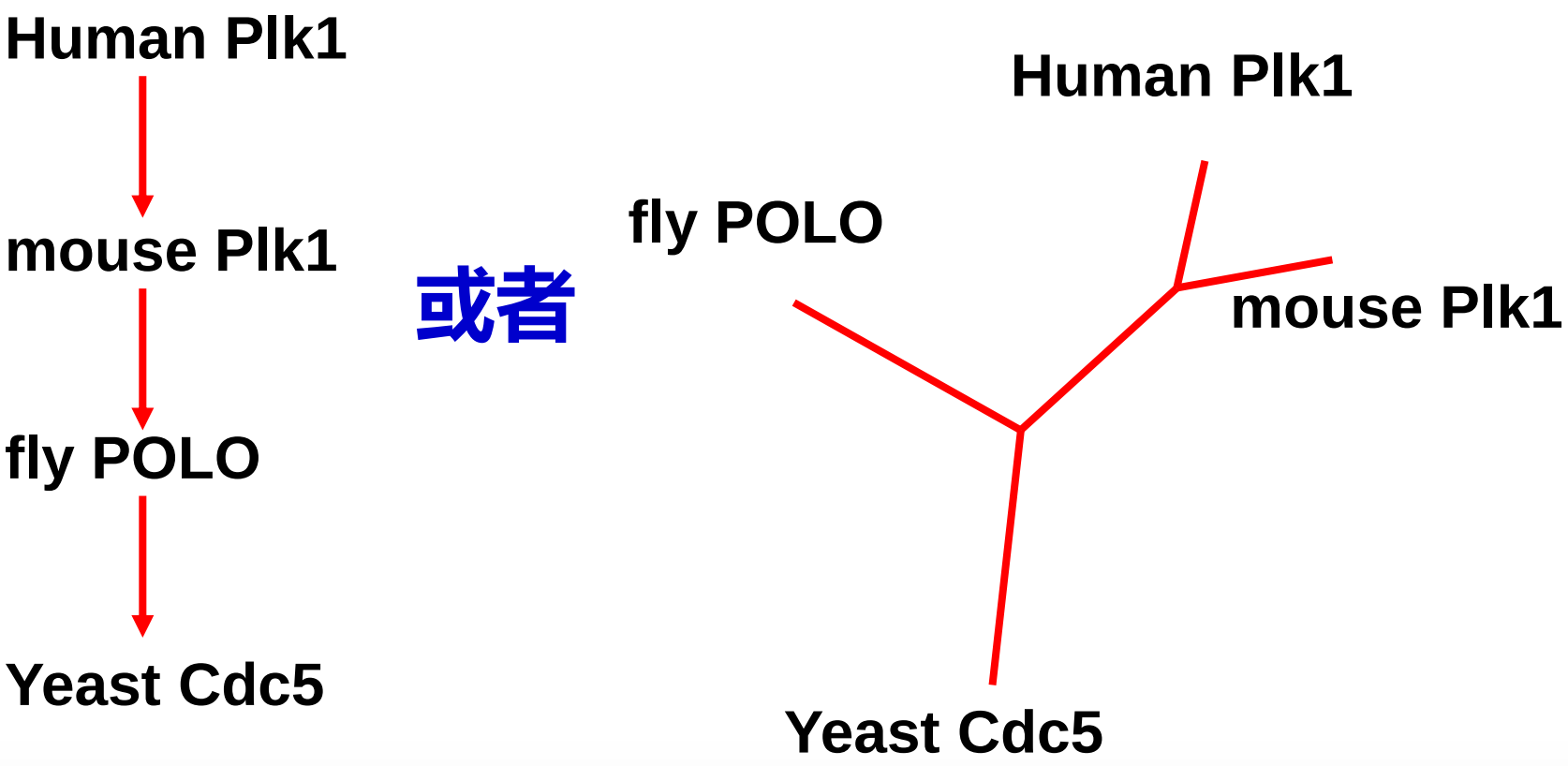
- 注意：反之可以不为真

  结构保守性 ⇏ 序列保守性

同源序列：定义

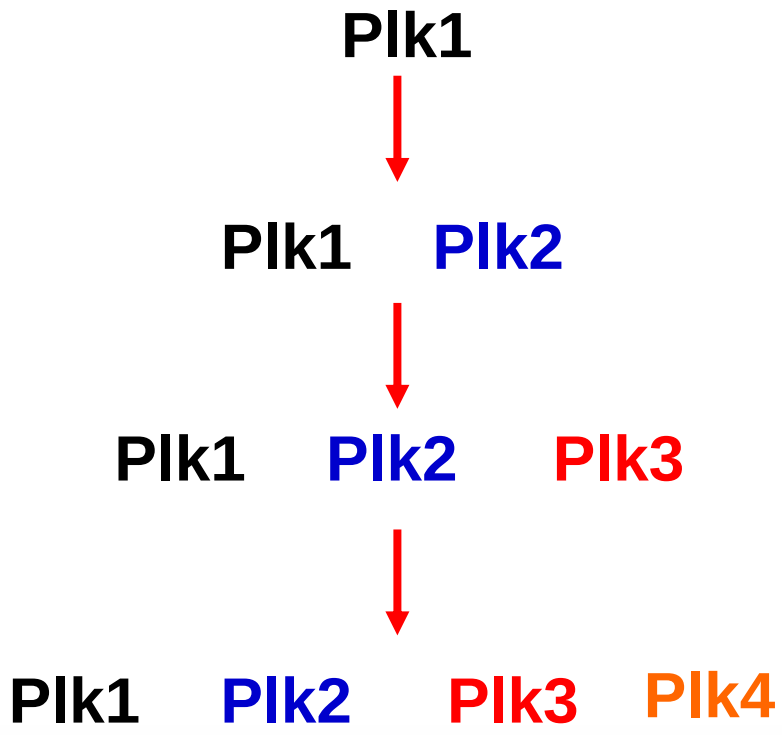
- Ortholog（直系同源）：两个基因通过物种形成的事件而产生，或源于不同物种的最近的共同祖先的两个基因，或者两个物种中的同一基因，一般具有相同的功能
- Paralog（旁系同源）：两个基因在同一物种中，通过至少一次基因复制的事件而产生
- Xenolog（异同源）：由某一个水平基因转移事件而得到的同源序列

直系同源：物种形成

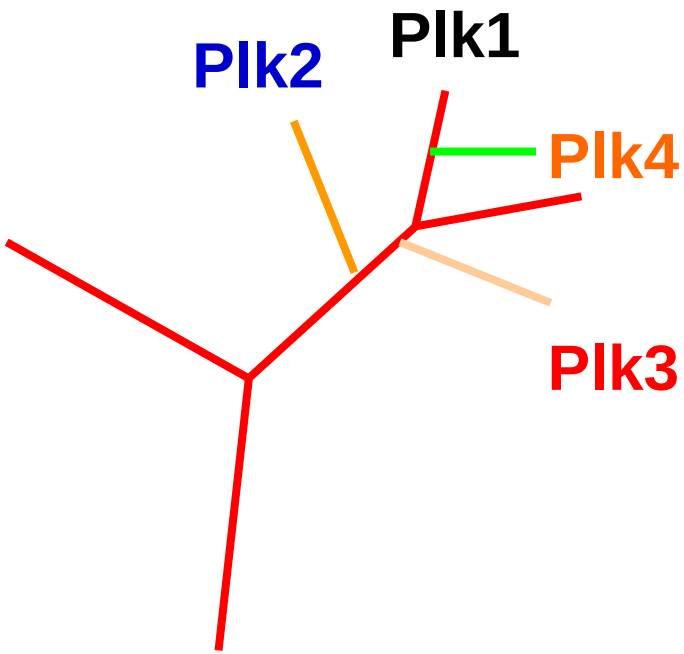

Human Plk1
mouse Plk1
fly POLO
Yeast Cdc5
或者
Human Plk1
fly POLO
mouse Plk1
Yeast Cdc5

旁系同源序列：基因复制

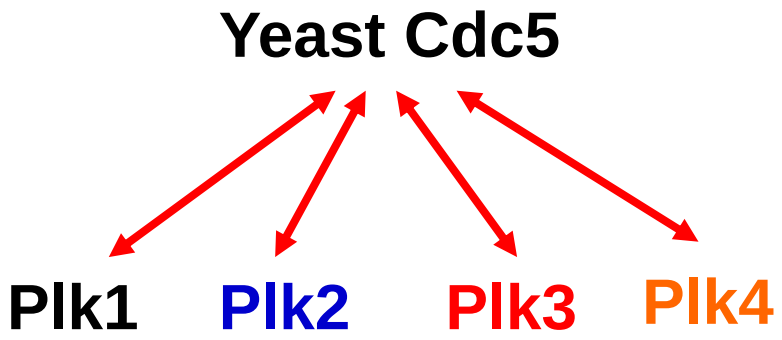
Human



或者



复杂问题



- 直系同源序列 *vs.* 旁系同源序列？

双序列比对的算法

- Dot Matrix，点阵法
- 动态规划算法：
  - Global: Needleman-Wunsch
  - Local: Smith-Waterman
- Word or *k*-tuple算法：FASTA, BLAST

动态规划算法

- 打分模型、替代矩阵以及空位罚分
- 比对算法：递归及动态规划算法
- 全局优化比对：Needleman-Wunsch
  - BLAST (Global Alignment), https://blast.ncbi.nlm.nih.gov/Blast.cgi
- 局部优化比对：Smith-Waterman
  - EMBOSS Water, https://www.ebi.ac.uk/Tools/psa/emboss_water/

好的 *vs.* 差的比对

- 两条序列的相似性 -> 相似/相同的生物学功能

Good

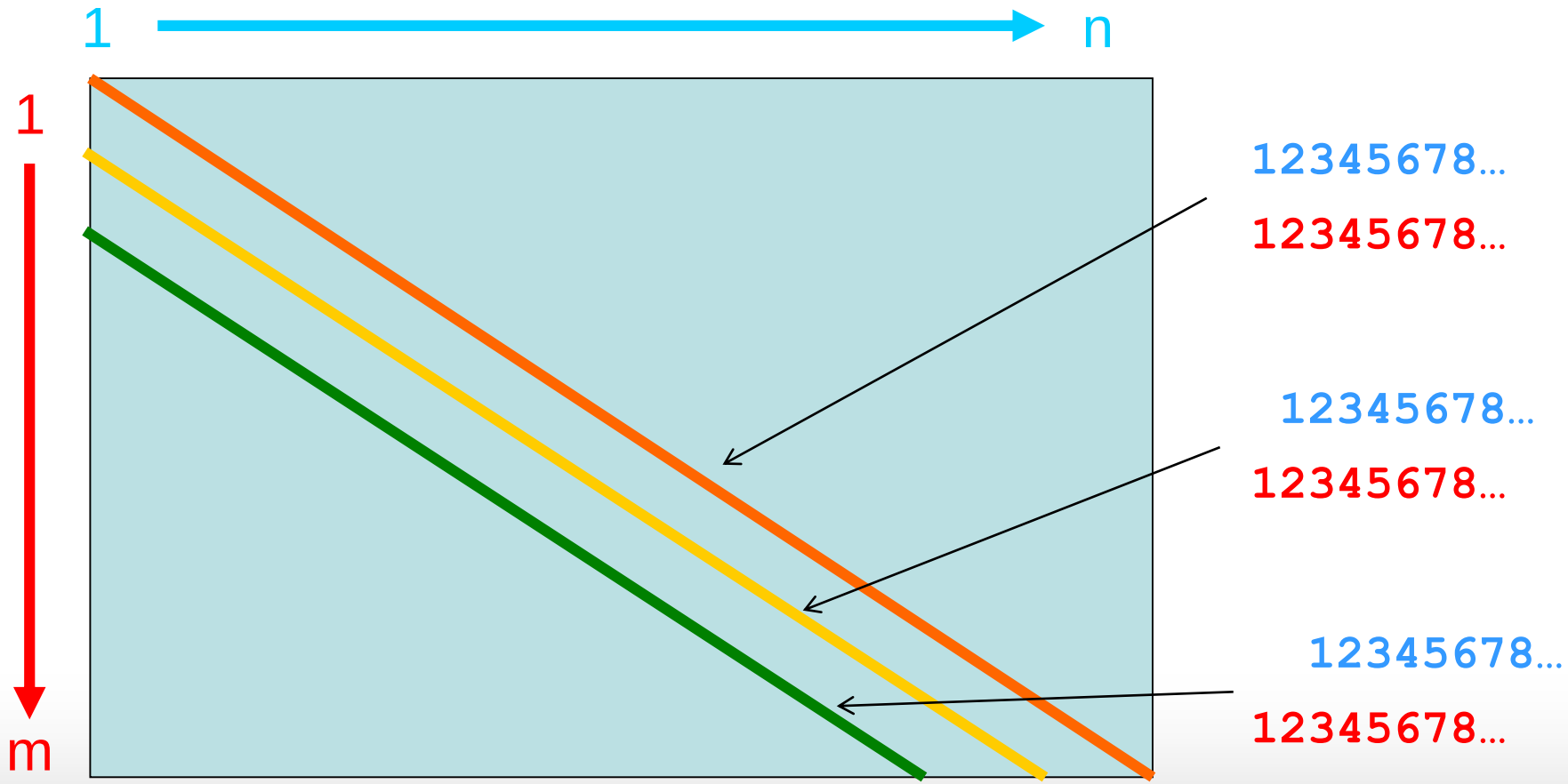
```
SUMO-1   5  EAKPSTEDLGDKKEGEYIKLKVIGQDSSEIHFKVKMTTHLKKLKESYCQRQGVPMNSLRF 64
            E KP     G K E ++I LKV GQD S + FK+K  T L KL ++YC+RQG+ M  +RF
SUMO-3   3  EEKPKE---GVKTENDHINLKVAGQDGSVVQFKIKRHTPLSKLMKAYCERQGLSMRQIRF 59

SUMO-1  65 LFEGQRIADNHTPKELGMEEEDVIEVYQEQTGG 97
            F+GQ I +  TP +L ME+ED I+V+Q+QTGG
SUMO-3  60 RFDGQPINETDTPAQLEMEDEDTIDVFQQQTGG 92
```

Bad

```
SUMO-1  1   MSDQEAKPSTEDLGDKKEGEYIKLKVIGQDSSEIHFK--VKMTTHLKKLKE 49
            +S    + S  D+G  K   Y+KL  +G+ +    FK   K+T +L  LKE
PCTK3   152 LSRMSRRASLSDIGFGKLETYVKLDKLGEGTYATVFKGRSKLTENLVALKE 202
```

无空位罚分的双序列比对

$O(n)=mn=n^2$



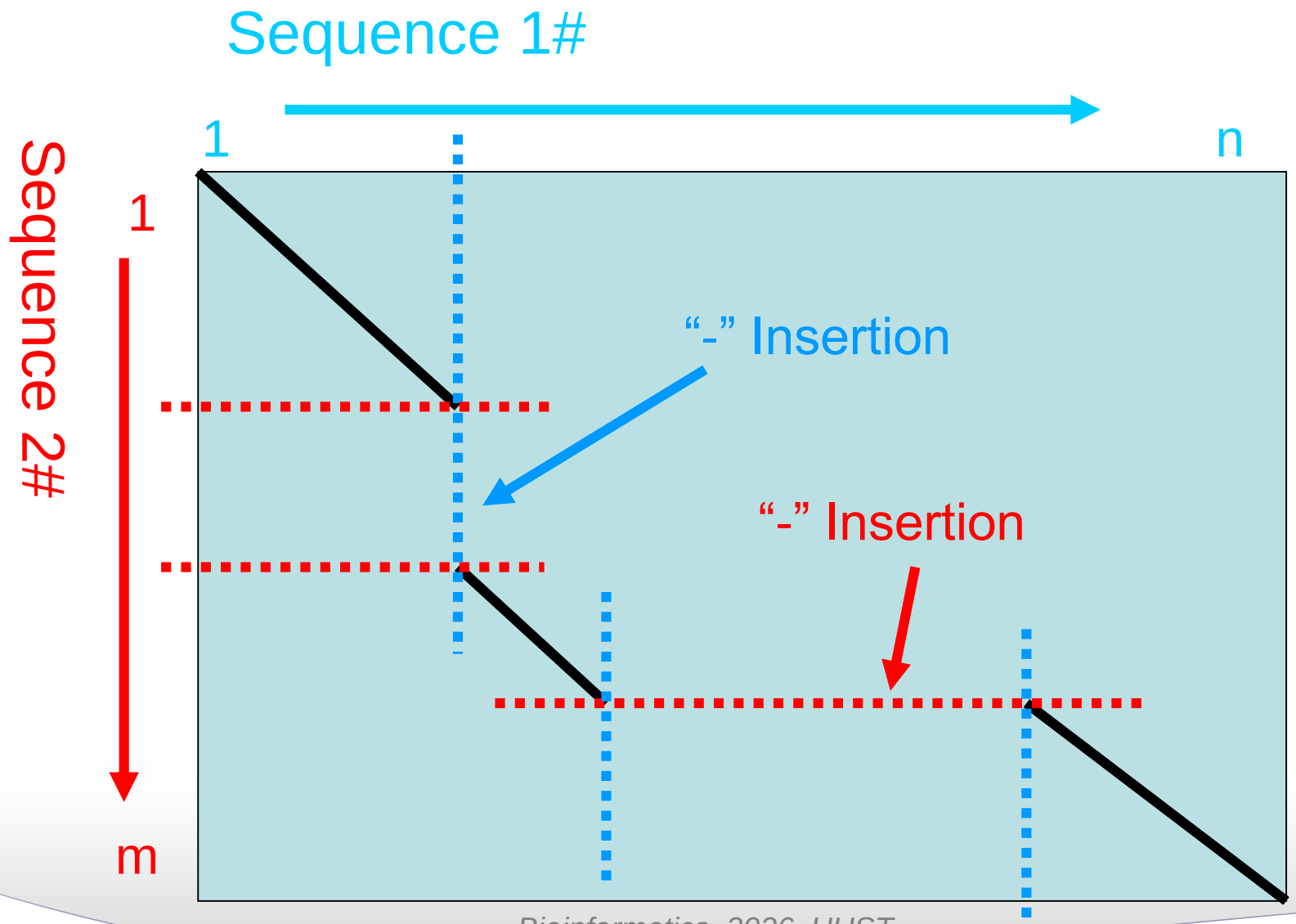
计算效率/计算复杂性

- 用CPU的计算时间和内存占用量来衡量
- O( )=…, 时间复杂度
- 对于需要解决的问题，其单位数量n运算的时间是一定的f(n)
- 如果需要解决的问题的大小与单位数量n的平方成正比，则$O(n)=n^2$
- 对于算法来说：$O(\log n) > O(n) > O(n^2)$

NP问题

- 一般的，$O(n^k)$, 当$k \leq 3$ 时，为多项式时间，较为容易处理
- 当$O(n^k)$= 指数级时间，则难以处理
- **NP难题**：无法找到能够在多项式时间复杂度内解决的问题
- 近似算法/优化算法，求近似解

有空位罚分的双序列比对



有空位罚分的双序列比对

- n=1, 3种比对
- n=2, 13种比对
- …
- 归纳法
- $a(n,n) \geq \frac{(n+2)(2n)!}{2(n!)^2}$
- n=1, $a(n, n) = 3$
- n=2, $a(n, n) = 12 < 13$

A B	A- -B	-A B-
AB CD	-AB CD-	-AB C-D	AB- -CD
	A-B CD-	A-B -CD	AB-- --CD
	A-B- -C-D	AB- C-D	--AB CD--
	-A-B C-D-	-AB- C--D	A--B -CD-

有空位罚分的双序列比对

- 两条序列比对，允许空位，时间复杂度为：

- $$\frac{(n+2)(2n)!}{2(n!)^2} \geq \frac{\sqrt{2\pi}(n+2)e^{-2\pi}(2n)^{2n+\frac{1}{2}}}{2e^{-2n+2}n^{2n+1}} = \frac{\sqrt{\pi}(n+2)2^{2n}}{e^2\sqrt{n}}$$

- NP-hard问题！

- 其中，斯特林公式：

$$x! = \sqrt{2\pi}x^{x+\frac{1}{2}}e^{-x}$$

打分模型

- 替代矩阵
  - 字符相同：identity
  - 字符替代：similarity，相似性，氨基酸/碱基之间的替代和突变
- 插入和缺失
- 空位罚分

替代矩阵的模型

- 考虑长度为*n*的序列*x*和长度为*m*的序列*y*
- 令x_i为*x*序列中的第*i*位；y_j为*y*序列中的第*j*位
- 对于不相关或者随机的模型R，假设x_i出现的频率为q_{xi}，y_j出现的频率为q_{yj}，则两条序列匹配的概率为：

$$P(x, y \mid R) = \prod_i q_{xi} \prod_j q_{yj}$$

替代矩阵的模型 (2)

- 对于另择假设/匹配模型M，两个字符匹配的概率为连接概率p_{ab}, 因此：

$$P(x, y \mid M) = \prod_i P_{xiyi}$$

- 两个似然性值之间的比值称为几率值(odds ratio):

$$\frac{P(x, y \mid M)}{P(x, y \mid R)} = \frac{\prod_i p_{xiyi}}{\prod_i q_{xi} \prod_i q_{yi}} = \prod_i \frac{p_{xiyi}}{q_{xi} q_{yi}}$$

替代矩阵的模型 (3)

- 连乘->连加；取对数，求对数几率值(log-odds ratio):

$$S = \sum_i s(x_i, y_i)$$

- 并且

$$s(a,b) = \log\left(\frac{P_{ab}}{q_a q_b}\right)$$

变连乘为连加

空位罚分

- 线性罚分：d, 每次罚分的分数；g，空位数

$$r(g) = -gd$$

- 修正的罚分：d, 第一次罚分的分数；g，空位数；e, 修正后的参数

$$r(g) = -d - (g-1)e$$

递归和动态规划算法 (2)

- 有空位的双序列比对，时间复杂度为：$O(2^{2n})$，指数增加，无法求最优解
- 动态规划算法：比较所有可能的字符对，考虑匹配、错配以及空位罚分，并且将比对次数控制在多项式时间内
- 替代矩阵：BLOSUM62，
  - 空位罚分：11
  - 延伸的空位罚分：1 (BLAST工具)

例：双序列比对

- 序列1： V D S – C Y
- 序列2： V E S L C Y
- 替代矩阵中的分数： 4 2 4 -11 9 7

- 两序列比对的总分：
- Score=Σ(AA pair scores) – gap penalty = 15

BLOSUM62替代矩阵

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

动态规划算法：全局比对

	Gap	V	D	S	C	Y
Gap	0	1gap	2gap	…		
V	1gap					
E	2gap					
S	…					
L						
C						
Y						

本例：线性罚分

$$r(g) = -gd$$

全局比对 (2)

	Gap	V	D	S	C	Y
Gap	0	-11	-22	-33	-44	-55
V	-11	S_{ij}				
E	-22					
S	-33					
L	-44					
C	-55					
Y	-66					

要求解S_{ij}的分数，我们必须先知道$S_{i-1, j-1}$, $S_{i-1, j}$, 以及$S_{i, j-1}$的分数，这种方法叫做递归算法；采用这种方法，可以把大的问题分割成小的问题逐一解决，即动态规划算法；需要存储如何得到S_{ij}分数的过程。

全局比对 (3)

	Gap	V	D	S	C	Y
Gap	0	-11	-22	-33	-44	-55
V	-11	S_{ij}				
E	-22					
S	-33					
L	-44					
C	-55					
Y	-66					

Needleman-Wunsch算法；

时间复杂度$O(n^2)$；

$$S_{ij} = \max \text{ of } \begin{cases} S_{i-1,\,j-1} + \sigma(x_i, y_j) \\ S_{i-1,\,j} - d \text{ (从左到右)} \\ S_{i,\,j-1} - d \text{ (从上到下)} \end{cases}$$

全局比对 (4)

	Gap	V	D	S	C	Y
Gap	0	-11	-22	-33	-44	-55
V	-11	S_{ij}				
E	-22					
S	-33					
L	-44					
C	-55					
Y	-66					

4

-11

-11

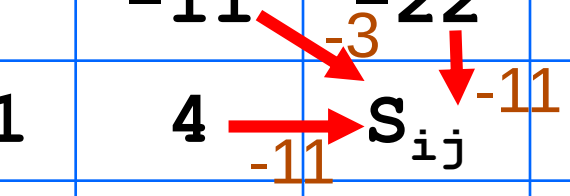
Needleman-Wunsch算法；

时间复杂度$O(n^2)$；

$$S_{ij} = \max \text{ of } \begin{cases} S_{i-1,\,j-1} + \sigma(x_i, y_j) \\ S_{i-1,\,j} - d \text{ (从左到右)} \\ S_{i,\,j-1} - d \text{ (从上到下)} \end{cases}$$

全局比对 (6)

	Gap	V	D	S	C	Y
Gap	0	-11	-22	-33	-44	-55
V	-11	4	S_{ij}			
E	-22					
S	-33					
L	-44					
C	-55					
Y	-66					



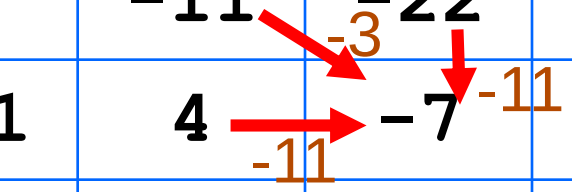
Needleman-Wunsch算法；

时间复杂度$O(n^2)$；

$$S_{ij} = \max \text{ of } \begin{cases} S_{i-1,\,j-1} + \sigma(x_i, y_j) \\ S_{i-1,\,j} - d \text{ (从左到右)} \\ S_{i,\,j-1} - d \text{ (从上到下)} \end{cases}$$

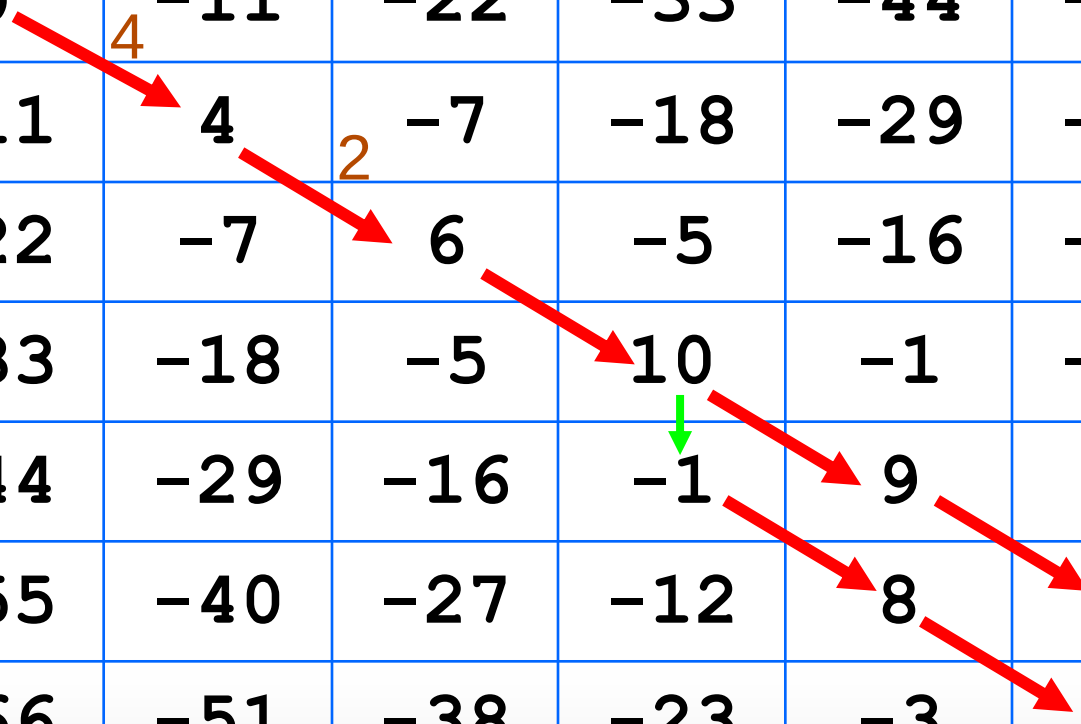
全局比对 (7)

	Gap	V	D	S	C	Y
Gap	0	-11	-22	-33	-44	-55
V	-11	4	-7			
E	-22					
S	-33					
L	-44					
C	-55					
Y	-66					



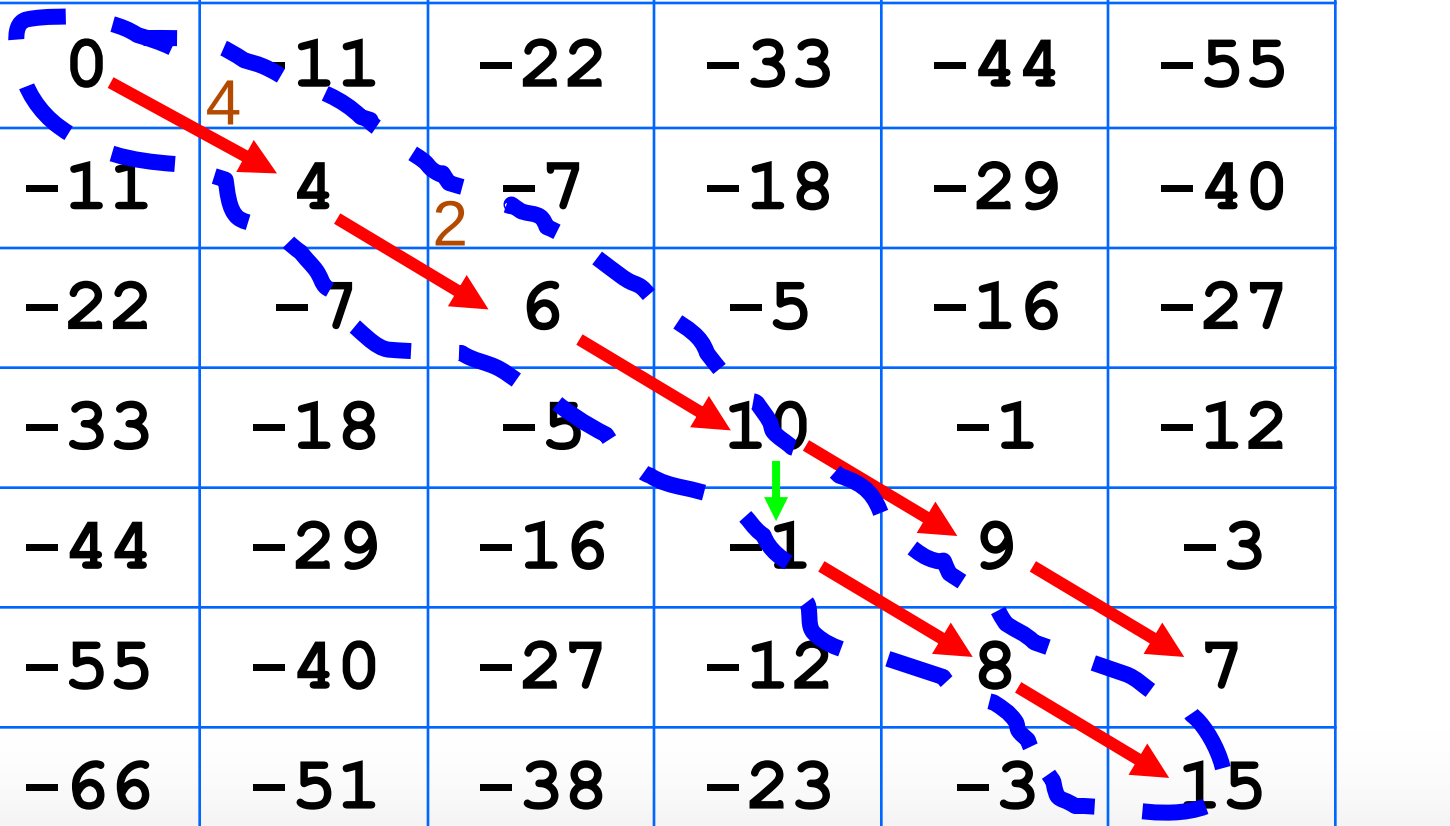
全局比对 (8)

	Gap	V	D	S	C	Y
Gap	0	-11	-22	-33	-44	-55
V	-11	4	-7	-18	-29	-40
E	-22	-7	6	-5	-16	-27
S	-33	-18	-5	10	-1	-12
L	-44	-29	-16	-1	9	-3
C	-55	-40	-27	-12	8	7
Y	-66	-51	-38	-23	-3	15



回溯：比对结果

	Gap	V	D	S	C	Y
Gap	0	-11	-22	-33	-44	-55
V	-11	4	-7	-18	-29	-40
E	-22	-7	6	-5	-16	-27
S	-33	-18	-5	10	-1	-12
L	-44	-29	-16	-1	9	-3
C	-55	-40	-27	-12	8	7
Y	-66	-51	-38	-23	-3	15

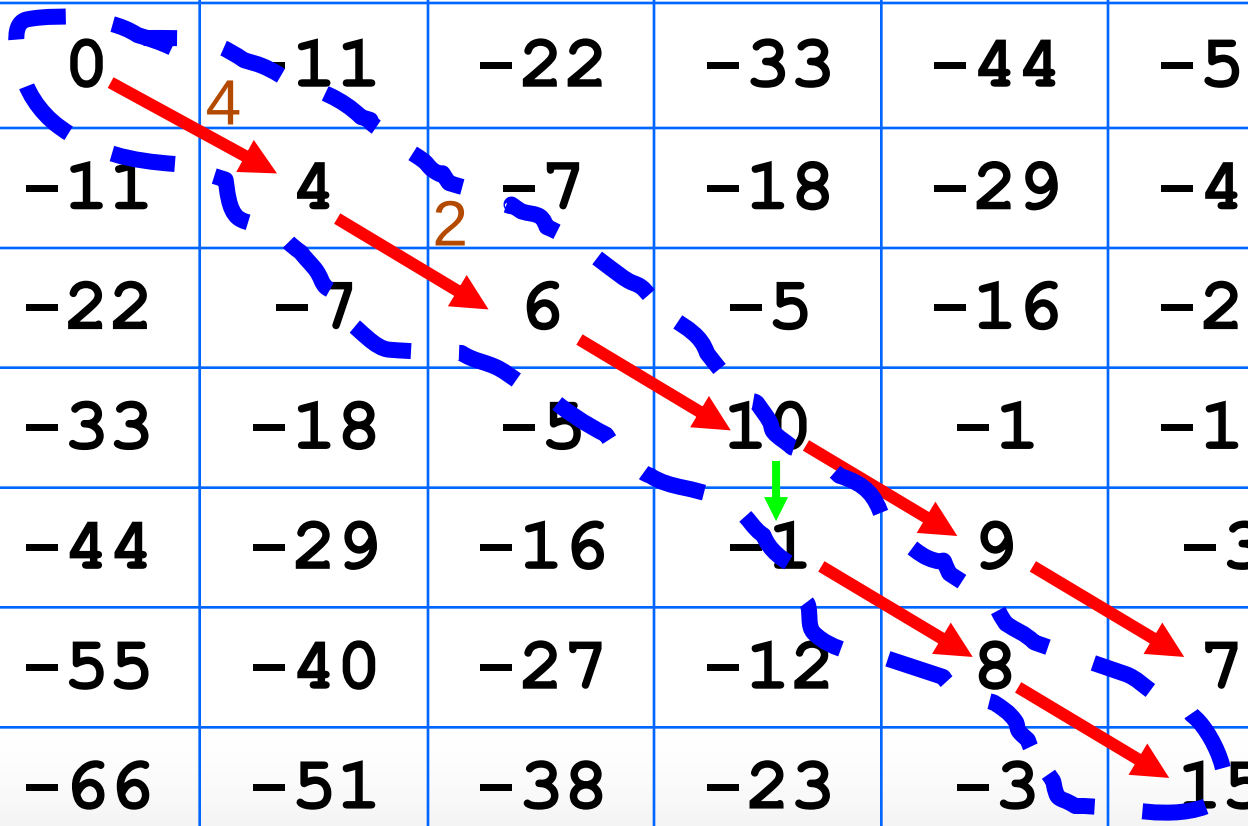


比对结果：

V D S - C Y

V E S L C Y

	Gap	V	D	S	C	Y
Gap	0	-11	-22	-33	-44	-55
V	-11	4	-7	-18	-29	-40
E	-22	-7	6	-5	-16	-27
S	-33	-18	-5	10	-1	-12
L	-44	-29	-16	-1	9	-3
C	-55	-40	-27	-12	8	7
Y	-66	-51	-38	-23	-3	15



局部优化比对

- 下例：局部优化打分
- 两条序列如下：

L D S - C H

G E S L C K

- 目标：使用局部优化算法寻找比对的结果

局部优化比对 (2)

- Smith-Waterman算法
- 时间复杂度$O(n^2)$
- S_{ij} = max of 0
- $S_{i-1,j-1} + \sigma(x_i, y_j)$
- $S_{i-1,j}$ - A (从左到右)
- $S_{i,j-1}$ - A (从上到下)
- 本例中：gap: 12，线性罚分模型

局部优化比对 (3)

	Gap	L	D	S	C	H
Gap	0	0	0	0	0	0
G	0	S_{ij}				
E	0					
S	0					
L	0					
C	0					
K	0					

Smith-Waterman算法；

$$S_{ij} = \max \text{ of } \begin{cases} S_{i-1,\,j-1} + \sigma(x_i, y_j) \\ S_{i-1,\,j} - d \text{ (从左到右)} \\ S_{i,\,j-1} - d \text{ (从上到下)} \\ 0 \end{cases}$$

局部优化比对 (5)

	Gap	L	D	S	C	H
Gap	0	0	0	0	0	0
G	0	0	0			
E	0					
S	0					
L	0					
C	0					
K	0					

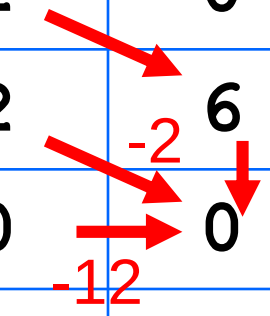
-1

-12

-12

局部优化比对 (6)

	Gap	L	D	S	C	H
Gap	0	0	0	0	0	0
G	0	0	0	0	0	0
E	0	0	2	0	0	0
S	0	0	2	6	0	0
L	0	4	0	0	5	0
C	0	0	1	0	9	2
K	0	0	0	0	0	8



局部优化比对 (7)

	Gap	L	D	S	C	H
Gap	0	0	0	0	0	0
G	0	0	0	0	0	0
E	0	0	2	0	0	0
S	0	0	2	6	0	0
L	0	4	0	0	5	0
C	0	0	1	0	9	2
K	0	0	0	0	0	8

比对结果：

L D S – C H

G E S L C K

	Gap	L	D	S	C	H
Gap	0	0	0	0	0	0
G	0	0	0	0	0	0
E	0	0	2	0	0	0
S	0	0	2	6	0	0
L	0	4	0	0	5	0
C	0	0	1	0	9	2
K	0	0	0	0	0	8

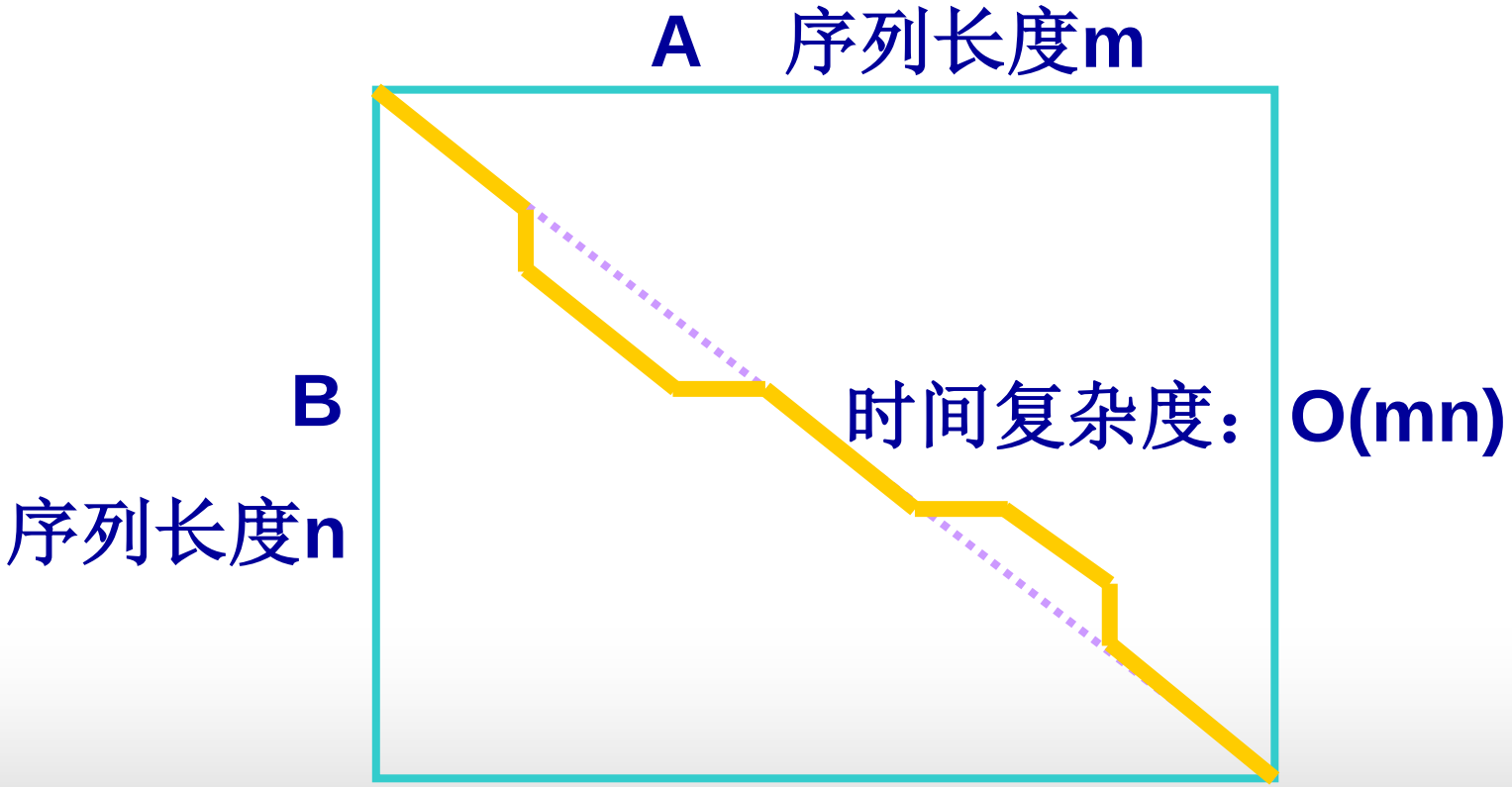
打分有何不同？

```
L  D  S  -  C  H
G  E  S  L  C  K
```

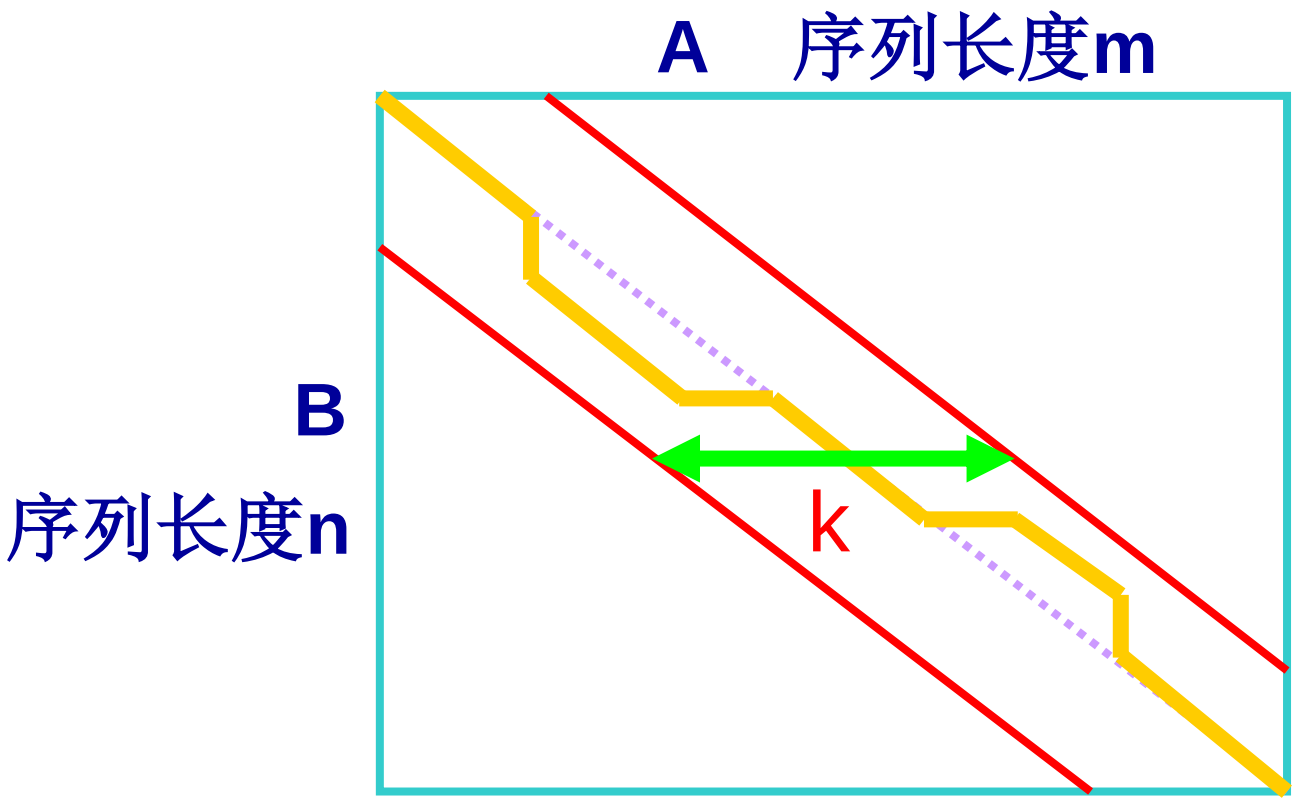
- Smith-waterman算法打分：9分
- 直接打分：-4+2+4-12+9-1=-2
- 为何不同？

K-tup算法原理

- 对于两条序列A, B，若包含少量gap，则最优比对趋近对角线



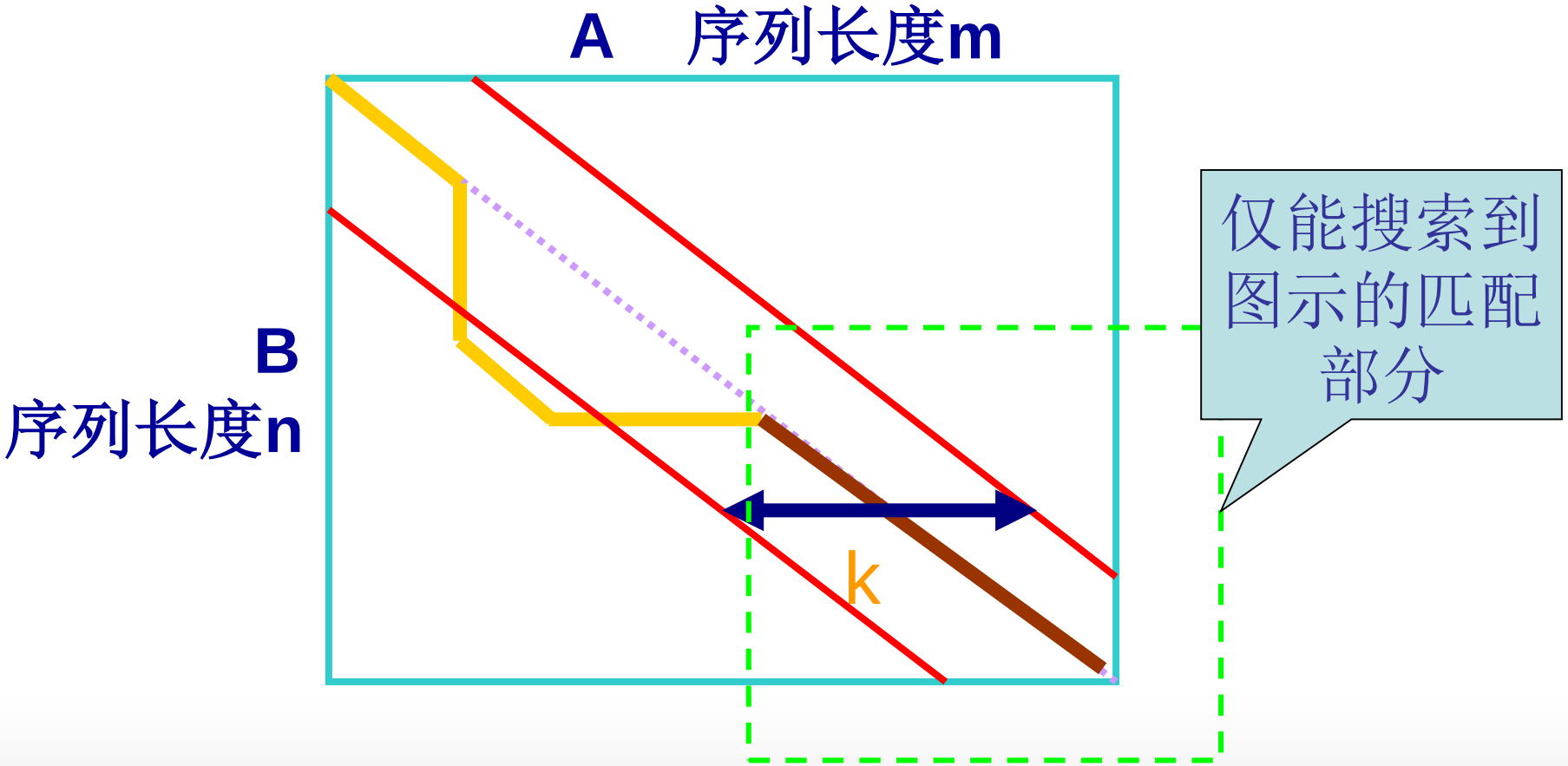
*K-tup*算法原理 (2)



令**k**为一常数，搜索限定区域内的最优比对

时间复杂度：O(kn)

*K-tup*算法原理：缺点



BLAST

- Word size: DNA, 11nt; 蛋白质，3aa
- 蛋白质序列数据库，构建由3aa组成的分值表,采用BLOSUM62矩阵打分
- 待查询序列，打断成3aa的片段，在上述数据库中的分值表中进行查询
- 保留高于域值的结果，并进行两端的延伸，HSP: high-scoring segment pair
- Nothing can be worse: 牺牲灵敏度，提高计算速度

Nucleotide BLAST三个program

https://www.ncbi.nlm.nih.gov/Class/MLACourse/Modules/BLAST/nucleotide_blast.html

Nucleotide-Nucleotide BLAST (blastn)

Now that we have explored the program and database options, let's do a basic **blastn** search with the Jurassic Park sequence that you have copied/pasted into memory. If you haven't already copied the query sequence into memory, please do it now.

One more note before we do the search...

The nucleotide BLAST page provides a selection of three programs that vary in their sensitivity and speed: **megablast** (default), **discontiguous megablast**, and **blastn**.

For our sample search, use the traditional **blastn** program.

Footnote, for your **future reference**. Some of the differences between the algorithms are highlighted below.

Megablast	Retrieves **highly similar sequences** and is very **fast**. It efficiently find long alignments between very similar sequences -- it is intended for comparing a query to closely related sequences and works best if the target percent identity is 95% or more. (word size* is 28 base pairs). learn more...
Discontiguous megablast	Retrieves **more dissimilar sequences** than megablast, but is more sensitive than blastn. It uses an initial seed that ignores some bases (allowing mismatches) and is intended for cross-species comparisons -- the third base wobbling is taken into consideration by focusing on finding matches at the first and second codon positions while ignoring the mismatches in the third position. (word size* can be set only at 11 or 12 base pairs.) learn more...
Blastn	Retrieves **somewhat similar sequences**, so **can find more distantly related sequences**, but is **slower** than megablast and discontiguous megablast. (default word size* can range from 7 base pairs to 11 (default) base pairs) learn more...

* **Word Size** is discussed later in the module in the slide on how did BLAST work. It is mentioned here only so this slide can serve as a useful reference after the course.

BLAST:索引表构建

- formatdb命令，将fasta格式的序列文件转换成blast能够识别的文件格式
- 构建索引表：

PQG

PQG 7+5+6=18

PEG 7+2+6=15

PWG 7-2+6=11

SQG -1+5+6=10

BLAST: 序列匹配

- 两条蛋白质序列
- Protein1：IVPQGRL
- Protein2：VAPEGKL

- Protein1：I V P Q G R L
- Protein2：V A P E G K L

<Word>

7 2 6

3 0 2 4

两边延伸

HSP分值：3+0+15+2+4=24

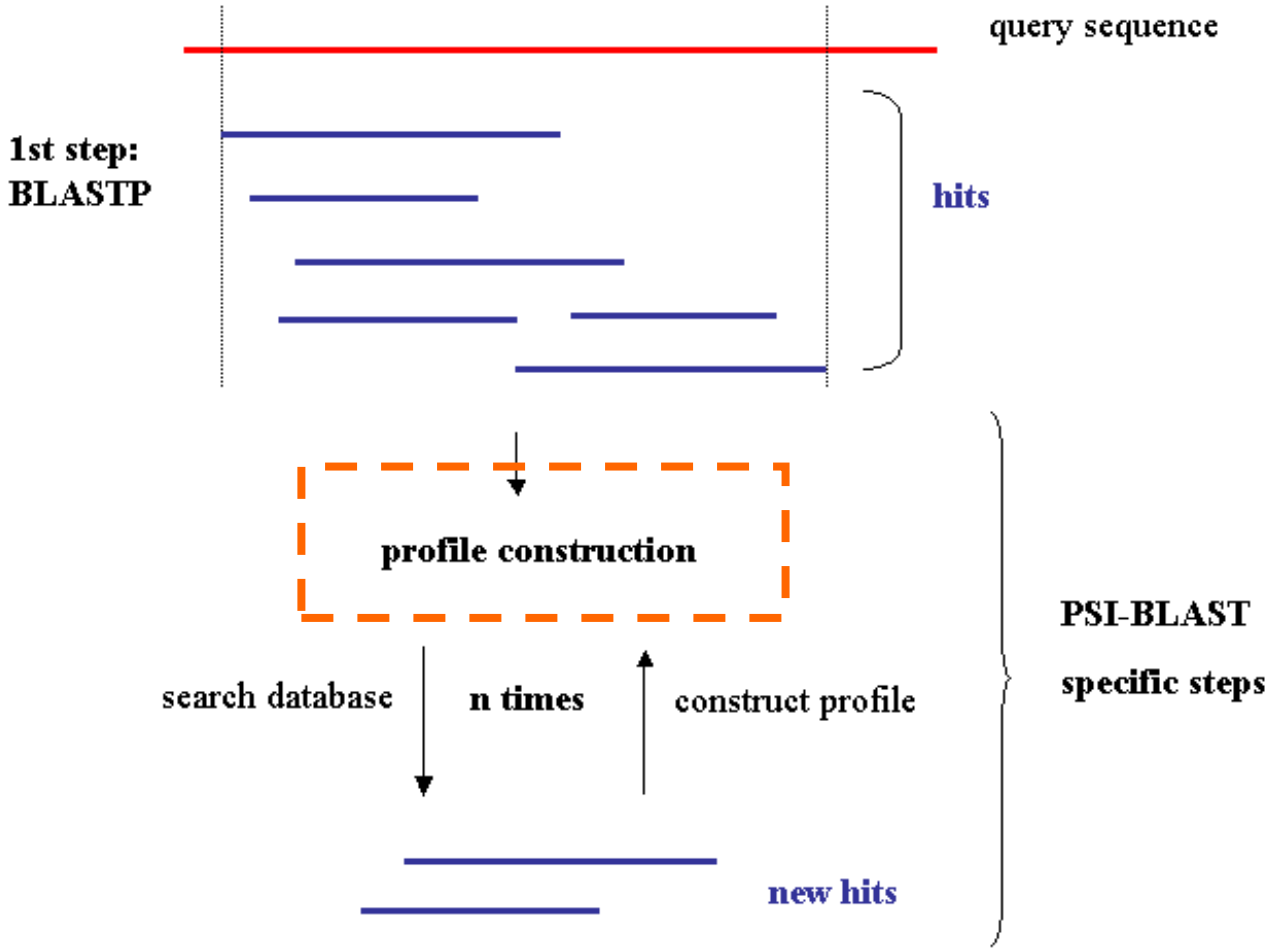
BLAST：发展与改进

- 早期的BLAST版本：无空位罚分
- 新版本：Gap Penalties: Existence: 11, Extension: 1
- Psi-BLAST: 构建位点特异性矩阵
- Phi-BLAST：包含特定模体的序列相似性搜索

Psi-BLAST ：迭代搜索

- 第一步，使用普通的blast算法进行搜索
- 第二步，将搜索得到的序列，包括输入的序列放在一起，构建位点特异性的矩阵(Position Specific Matrix)
- 第三步，利用上面得到的矩阵谱(profile)，再次在数据库中进行搜索
- 重复2，3步，直到不再有新的序列出现
- 优点：能够发现序列相似性非常低的同源序列
- 缺点：常常得到假阳性的结果

Psi-BLAST ：迭代搜索 (2)



打分矩阵及其含义

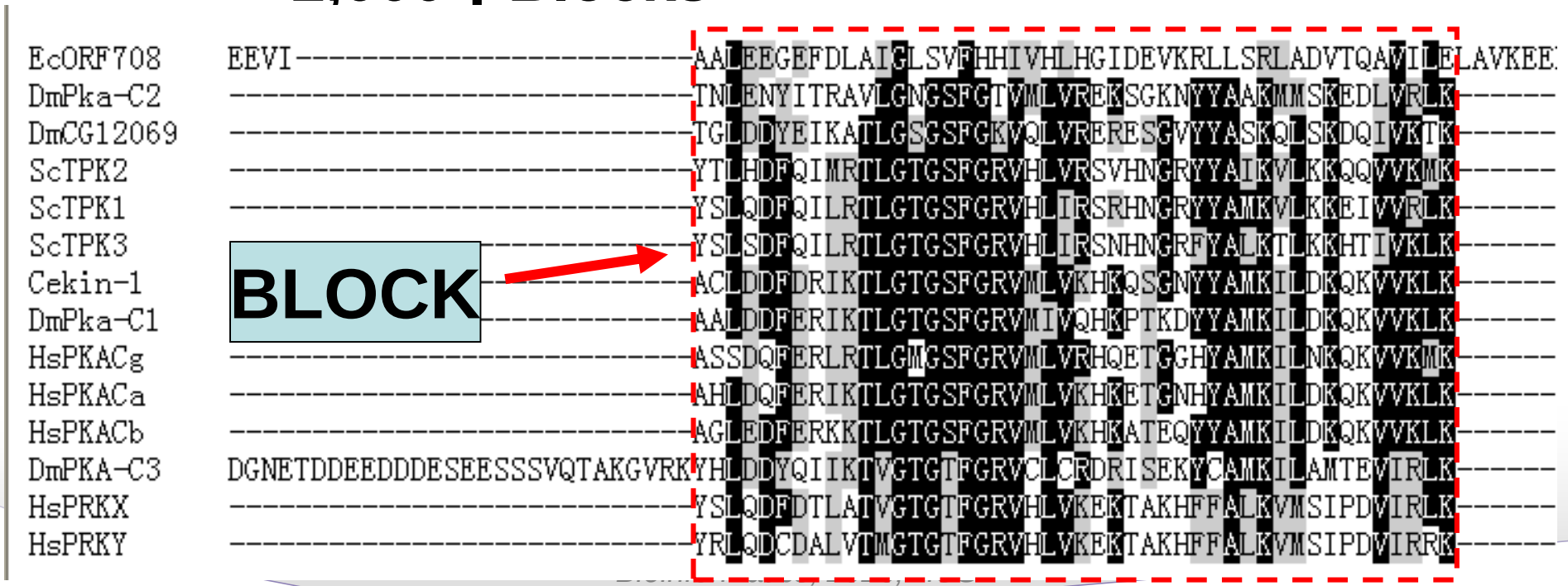
- Dayhoff: PAM系列矩阵
- **Henikoff: BLOSUM系列矩阵**
- **常用氨基酸打分矩阵：BLOSUM62**

BLOSUM系列矩阵

- BLOCK: 蛋白质家族保守的一段氨基酸，无gap，一般几个~上百个氨基酸
- Prosite家族：至少有一个BLOCK存在于该家族的所有蛋白质序列中
- BLOSUM62: 序列的平均相似性为62%的BLOCK构建的打分矩阵
- 最被广泛使用的氨基酸打分矩阵系列

BLOSUM系列矩阵

- 1992，Steven Henikoff
- 利用”Block” (模块) 计算矩阵
  - ~500组相似的蛋白质
  - ~2,000个Blocks



计算方法

- 例：给定一列数据，9个A，1个S
  - f_{AA}= (8+1)*8/2=36; f_{AS}/f_{SA}=9; f_{SS}=0
- $$q_{ij} = f_{ij} / \sum_{i=1}^{20} \sum_{j=1}^{i} f_{ij}$$
  - q_{AA}=36/45=0.8, q_{AS}=9/45=0.2
- $$p_i = q_{ii} + \sum_{j \neq i} q_{ij}/2$$
  - p_A=(36+9/2)/45=0.9; p_S=0.1
  - $e_{AS}=p_A p_S + p_S p_A$=0.9*0.1*2=0.18
- $$s_{ij} = \log_2(q_{ij}/e_{ij})$$
  - $S_{as}=\log_2(0.2/0.18)$=0.152

Relative entropy

$$H = \sum_{i=1}^{20} \sum_{j=1}^{i} q_{ij} \times s_{ij}$$

Expected Score

$$E = \sum_{i=1}^{20} \sum_{j=1}^{i} p_i \times p_j \times s_{ij}$$

多列数据

- 两列数据：16个A，4个S
- p_A=16/20=0.8; p_S=0.2
- $e_{AS}=p_Ap_S+p_Ap_S=0.8*0.2*2=0.32$
- q_{AS}=(9+21)/90 =0.333
- $S_{AS}=\log_2(0.333/0.32)=0.059$

AA
AA
AA
AA
AA
AA
AA
AS
AS
SS

AA
AA
AF
FA
FA
AA
AA
AF
AS
SS

- 两列数据：13个A，3个S
- p_A=13/20=0.65; p_S=0.15
- $e_{AS}=p_Ap_S+p_Ap_S=0.65*0.15*2=0.195$
- q_{AS}=(7+12)/90 =0.211
- $S_{AS}=\log_2(0.195/0.211)=-0.11$

BLOSUM62

```
#  Matrix made by matblas from blosum62.iij
#  * column uses minimum score
#  BLOSUM Clustered Scoring Matrix in 1/2 Bit Units
#  Blocks Database = /data/blocks_5.0/blocks.dat
#  Cluster Percentage: >= 62
#  Entropy =   0.6979, Expected =  -0.5209
   A  R  N  D  C  Q  E  G  H  I  L  K  M  F  P  S  T  W  Y  V  B  Z  X  *
A  4 -1 -2 -2  0 -1 -1  0 -2 -1 -1 -1 -1 -2 -1  1  0 -3 -2  0 -2 -1  0 -4
R -1  5  0 -2 -3  1  0 -2  0 -3 -2  2 -1 -3 -2 -1 -1 -3 -2 -3 -1  0 -1 -4
N -2  0  6  1 -3  0  0  0  1 -3 -3  0 -2 -3 -2  1  0 -4 -2 -3  3  0 -1 -4
D -2 -2  1  6 -3  0  2 -1 -1 -3 -4 -1 -3 -3 -1  0 -1 -4 -3 -3  4  1 -1 -4
C  0 -3 -3 -3  9 -3 -4 -3 -3 -1 -1 -3 -1 -2 -3 -1 -1 -2 -2 -1 -3 -3 -2 -4
Q -1  1  0  0 -3  5  2 -2  0 -3 -2  1  0 -3 -1  0 -1 -2 -1 -2  0  3 -1 -4
E -1  0  0  2 -4  2  5 -2  0 -3 -3  1 -2 -3 -1  0 -1 -3 -2 -2  1  4 -1 -4
G  0 -2  0 -1 -3 -2 -2  6 -2 -4 -4 -2 -3 -3 -2  0 -2 -2 -3 -3 -1 -2 -1 -4
H -2  0  1 -1 -3  0  0 -2  8 -3 -3 -1 -2 -1 -2 -1 -2 -2  2 -3  0  0 -1 -4
I -1 -3 -3 -3 -1 -3 -3 -4 -3  4  2 -3  1  0 -3 -2 -1 -3 -1  3 -3 -3 -1 -4
L -1 -2 -3 -4 -1 -2 -3 -4 -3  2  4 -2  2  0 -3 -2 -1 -2 -1  1 -4 -3 -1 -4
K -1  2  0 -1 -3  1  1 -2 -1 -3 -2  5 -1 -3 -1  0 -1 -3 -2 -2  0  1 -1 -4
M -1 -1 -2 -3 -1  0 -2 -3 -2  1  2 -1  5  0 -2 -1 -1 -1 -1  1 -3 -1 -1 -4
F -2 -3 -3 -3 -2 -3 -3 -3 -1  0  0 -3  0  6 -4 -2 -2  1  3 -1 -3 -3 -1 -4
P -1 -2 -2 -1 -3 -1 -1 -2 -2 -3 -3 -1 -2 -4  7 -1 -1 -4 -3 -2 -2 -1 -2 -4
S  1 -1  1  0 -1  0  0  0 -1 -2 -2  0 -1 -2 -1  4  1 -3 -2 -2  0  0  0 -4
T  0 -1  0 -1 -1 -1 -1 -2 -2 -1 -1 -1 -1 -2 -1  1  5 -2 -2  0 -1 -1  0 -4
W -3 -3 -4 -4 -2 -2 -3 -2 -2 -3 -2 -3 -1  1 -4 -3 -2 11  2 -3 -4 -3 -2 -4
Y -2 -2 -2 -3 -2 -1 -2 -3  2 -1 -1 -2 -1  3 -3 -2 -2  2  7 -1 -3 -2 -1 -4
V  0 -3 -3 -3 -1 -2 -2 -3 -3  3  1 -2 -1 -1 -2 -2  0 -3 -1  4 -3 -2 -1 -4
B -2 -1  3  4 -3  0  1 -1  0 -3 -4  0 -3 -3 -2  0 -1 -4 -3 -3  4  1 -1 -4
Z -1  0  0  1 -3  3  4 -2  0 -3 -3  1 -1 -3 -1  0 -1 -3 -2 -2  1  4 -1 -4
X  0 -1 -1 -1 -2 -1 -1 -1 -1 -1 -1 -1 -1 -1 -2 -1 -1 -2 -1 -1 -1 -1 -1 -4
* -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4  1
```

Lod ratios are multiplied by a scaling factor of 2 and then rounded to the nearest integer value

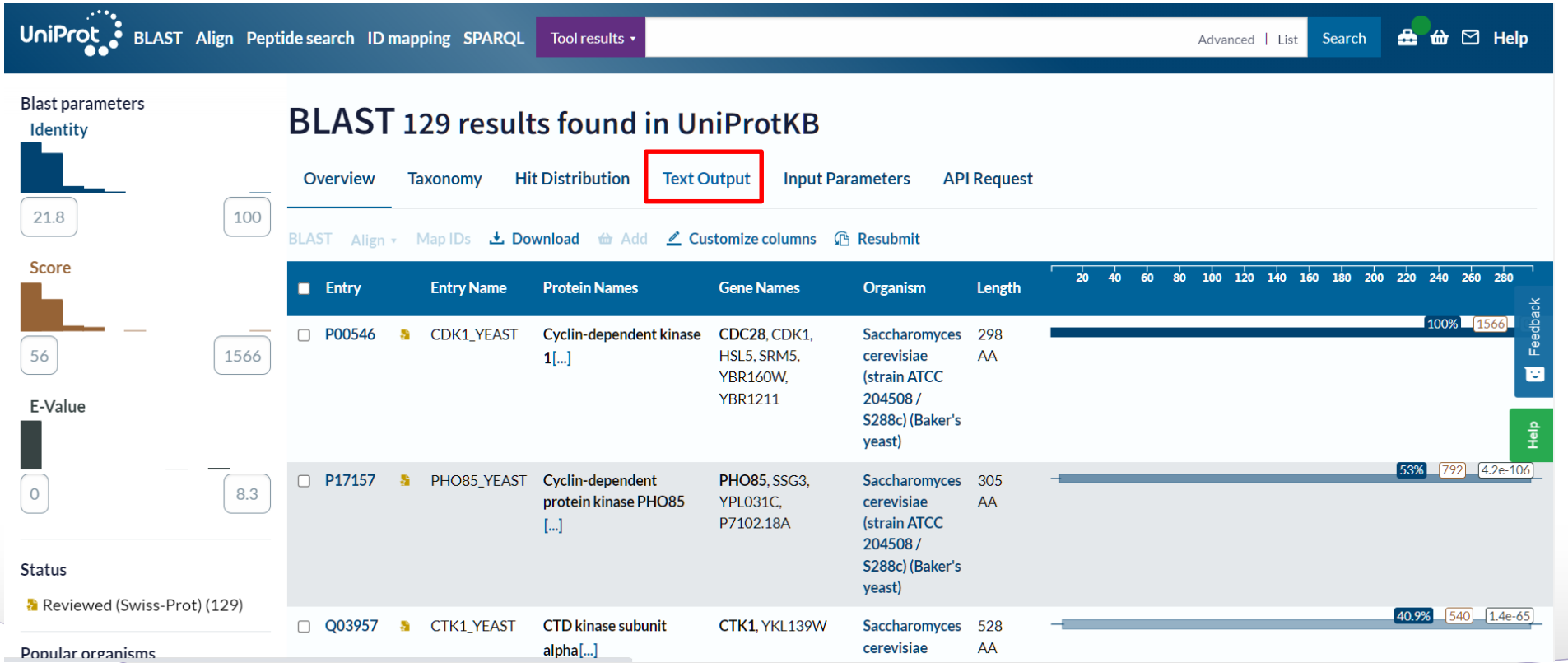
序列比对的显著性检验

- 两条序列在进化上显著相关：
- Sander & Schneider, 1991, 序列相似性的显著性于匹配的序列长度*k*有一定关系：
  - (1) 当匹配序列$k<10$时，无相似性
  - (2) $10<k<80$时，域值为$290.15k^{-0.562}$%
  - (3) $k>80$时，域值为24.8%
- Brenner,1998，
  - (1) k=~150，>25%
  - (2) k=~70，>40%
  - (3) 10%~20%时，无相似性

显著性计算

❑ 如何计算序列比对的显著性？

酵母CDK1在酵母中的序列比对结果



比对结果

SP:P21965MCK1_YEAST Protein kinase MCK1 OS=Saccharomyces cerevisi...	127	9e-35
SP:P19454CSK22_YEAST Casein kinase II subunit alpha' OS=Saccharom...	124	7e-34
SP:P15790CSK21_YEAST Casein kinase II subunit alpha OS=Saccharomy...	121	2e-32
SP:P32581IME2_YEAST Meiosis induction protein kinase IME2/SME1 OS...	117	6e-30
SP:P06782SNF1_YEAST Carbon catabolite-derepressing protein kinase...	111	7e-28
SP:Q12222YGK3_YEAST Glycogen synthase kinase-3 homolog YGK3 OS=Sa...	108	2e-27
SP:P22216RAD53_YEAST Serine/threonine-protein kinase RAD53 OS=Sac...	104	2e-25
SP:Q01389BCK1_YEAST Serine/threonine-protein kinase BCK1/SLK1/SSP...	104	2e-25
SP:P27636CDC15_YEAST Cell division control protein 15 OS=Saccharo...	100	3e-24

- SNF1_YEAST的结果：
  - Score: 111
  - E-value: 7e-28
- 问题：
  - 如何计算Score？
  - 如何计算E-value？该值是何意义？

贝叶斯方法：模型比较

- 考虑两个模型：两条序列无关的概率为P(R|x,y)以及两条序列相关的概率为P(M|x,y)
- 两个模型的前向概率为：P(M)和P(R)，存在：
  - P(R)=1-P(M)
- 因此：

$$P(M \mid x,y) = \frac{P(x,y \mid M)P(M)}{P(x,y)} = \frac{P(x,y \mid M)P(M)}{P(x,y \mid M)P(M) + P(x,y \mid R)P(R)}$$

$$= \frac{\frac{P(x,y \mid M)P(M)}{P(X,y \mid R)P(R)}}{1 + \frac{P(x,y \mid M)P(M)}{P(x,y \mid R)P(R)}}$$

贝叶斯方法：模型比较(2)

令 $S' = S + \log\left(\frac{P(M)}{P(R)}\right)$ 则 $S = \log\left(\frac{P(x,y|M)}{P(x,y|R)}\right)$

得 P(M|x,y)=σ(S'), 其中，

$$\sigma(x) = \frac{e^x}{1+e^x}$$ Sigmoid方程

求近似值

- E(S) =~ Kmne$^{-\lambda S}$ = mn2^{-S};

$$\sum_{a,b} q_a q_b e^{\lambda S(a,b)} = 1$$

- S: bit分值，有公式：

$$S = \frac{\lambda R - \ln K}{\ln(2)}$$

- R，raw分值，根据打分矩阵直接得到的分数

SNF1

```
>SP:P06782 SNF1_YEAST Carbon catabolite-derepressing protein kinase OS=Saccharomyces
cerevisiae (strain ATCC 204508 / S288c) OX=559292
GN=SNF1 PE=1 SV=1
Length=633

 Score = 111 bits (277),  Expect = 7e-28
 Identities = 87/296 (29%), Positives = 144/296 (49%), Gaps = 44/296 (15%)

Query  5    LANYKRLEKVGEGTYGVVYKALDLRPGQGQRVVAL-KKIRLESEDEGVPSTAIREISLLK  63
            + NY+ ++ +GEG++G V  A     GQ   +  + KK+  +S+ +G      REIS L+
Sbjct  52   IGNYQIVKTLGEGSFGKVKLAYHTTTGQKVALKIINKKVLAKSDMQG---RIEREISYLR  108

Query  64   ELKDDNIVRLYDIVHSDAHKLYLVFEFLDLDLKRYMEGIPKDQPLGADIVKKFMMQLCKG  123
             L+  +I++LYD++ S   ++ +V E+   +L  Y+  + +D+ +     ++F  Q+
Sbjct  109  LLRHPHIIKLYDVIKSK-DEIIMVIEYAGNELFDYI--VQRDK-MSEQEARRFFQQIISA  164

Query  124  IAYCHSHRILHRDLKPQNLLINKDGNLKLGDFGLARAF--GVPLRAYTHEIVTLWYRAPE  181
            + YCH H+I+HRDLKP+NLL++   N+K+ DFGL+     G  L+       +  Y APE
Sbjct  165  VEYCHRHKIVHRDLKPENLLLDEHLNVKIADFGLSNIMTDGNFLKT---SCGSPNYAAPE  221

Query  182  VLLGGKQYSTGVDTWSIGCI-FAEMCNRKPIFSGDSEIDQIFKIFRVLGTPNEAIWPDIV  240
            V+ G       VD WS G I +  +C R P    D  I  +FK           I   +
Sbjct  222  VISGKLYAGPEVDVWSCGVILYVMLCRRLPF--DDESIPVLFK----------NISNGVY  269
```

因此，上例

```
>SP:P06782 SNF1_YEAST Carbon catabolite-derepressing protein kinase OS=Saccharomyces
cerevisiae (strain ATCC 204508 / S288c) OX=559292
GN=SNF1 PE=1 SV=1
Length=633

 Score = 111 bits (277),  Expect = 7e-28
 Identities = 87/296 (29%), Positives = 144/296 (49%), Gaps = 44/296 (15%)
```

- R=277

上例 (2)

```
Lambda      K        H
   0.323    0.142    0.434

Gapped
Lambda      K        H
   0.267    0.0410   0.140

Effective search space used: 590525104


  Database: uniprotkb_swissprot
    Posted date:  Dec 14, 2022 02:19 PM
  Number of letters in database: 205,548,017
  Number of sequences in database:  568,744
```

- λ= 0.267
- K=0.0410
- m=298
- n=590,525,104

上例 (3)

$$S = \frac{0.267 * 277 - \ln(0.041)}{\ln(2)}$$

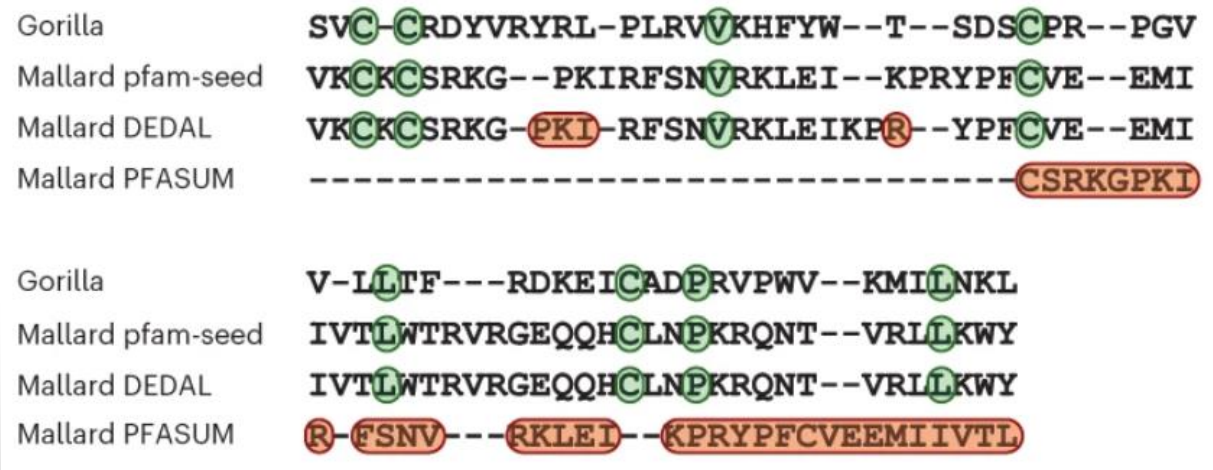
$$= \frac{73.959 - (-3.194)}{0.693} \approx 111$$

$$E = 298 * 590525104 * 2^{-111}$$

$$= 6.78e - 23 = 7e - 23$$

深度学习方法

- DEDAL (deep embedding and differentiable alignment)：深度嵌入与可微比对
  - 比对评分函数：根据每个序列对进行自适应调整，同时考虑错配发生的上下文环境
  - DEDAL训练：已知比对的序列对集合
  - 序列对特异性参数：Smith-Waterman算法打分



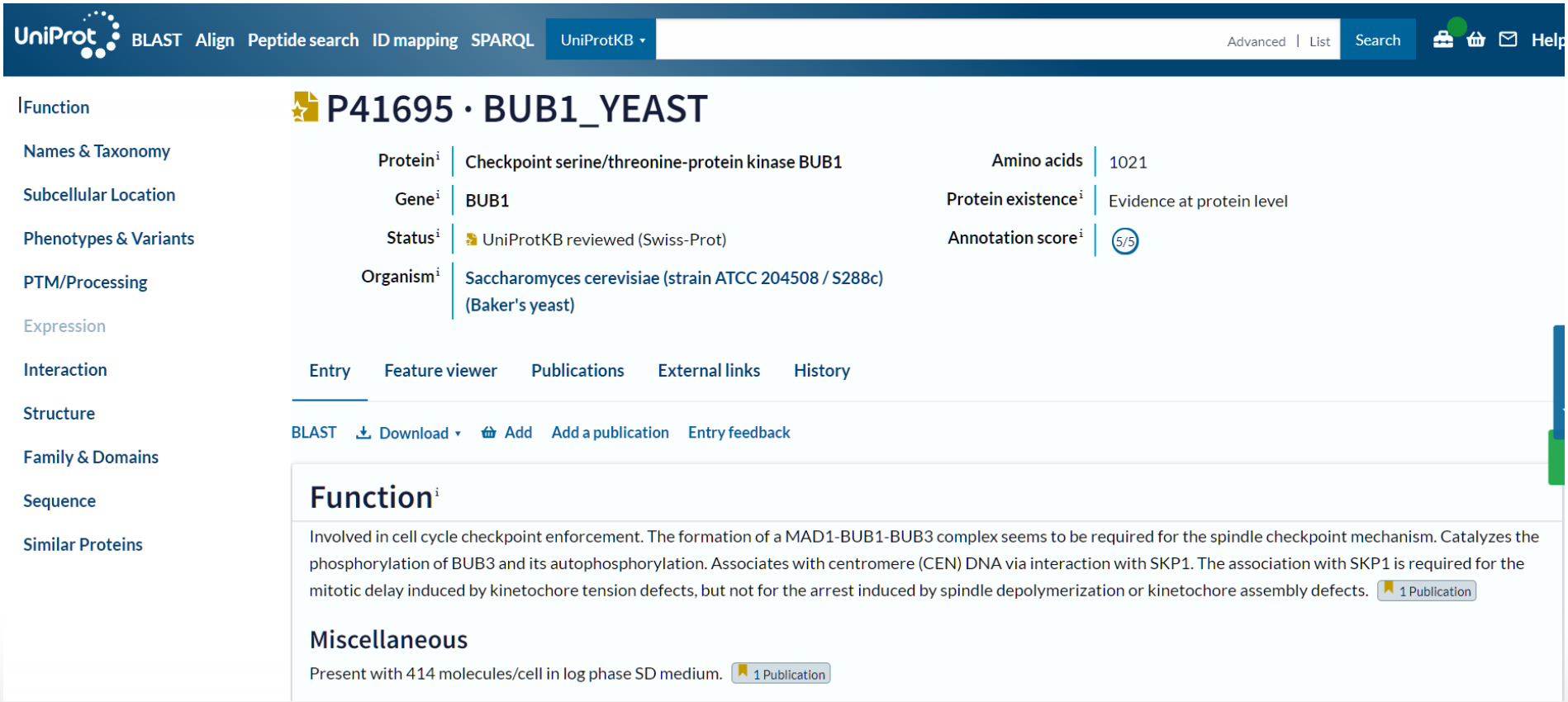
Nat Methods, 2023, 20, 40-41

同源搜索

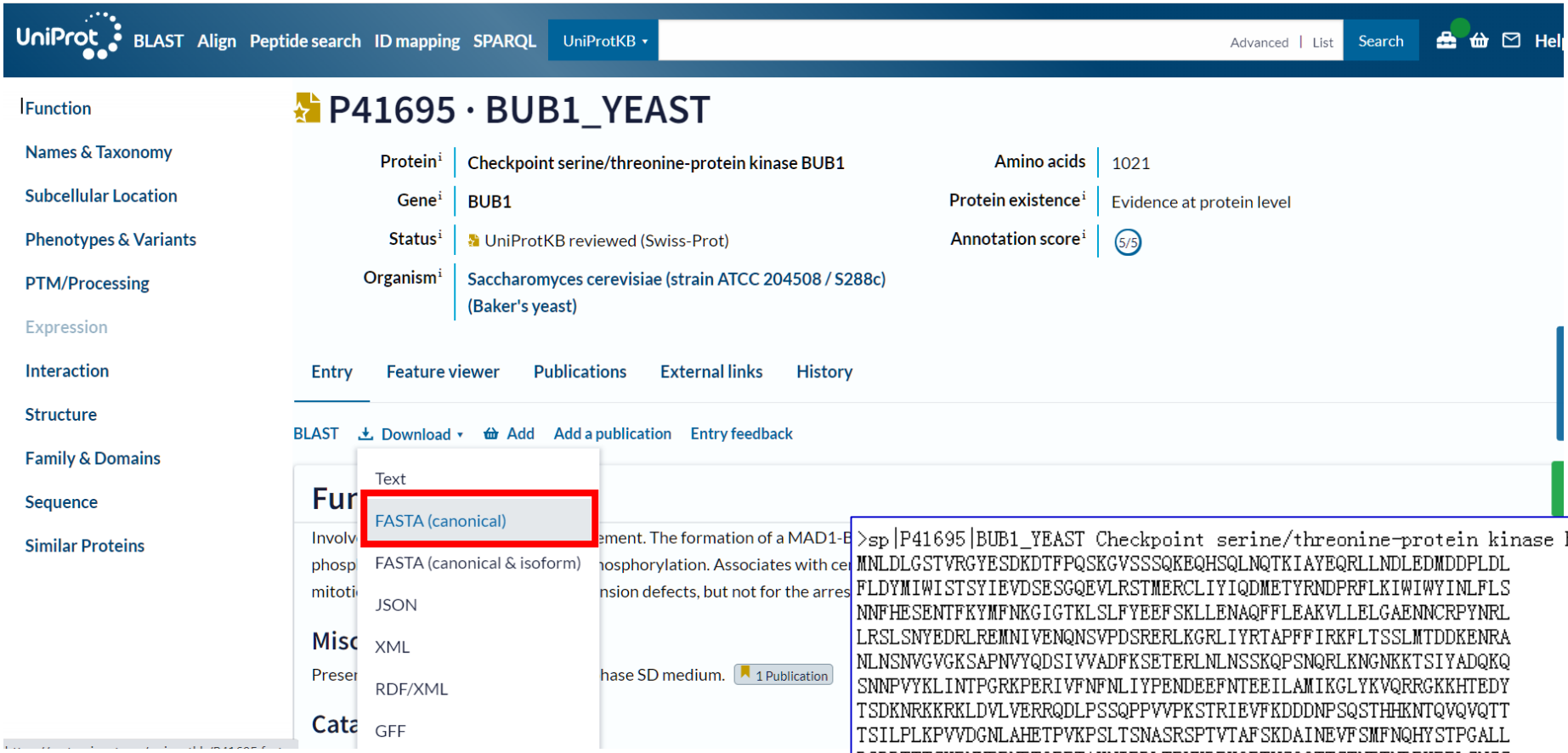
- 同源序列通常具有相似的生物学功能
- 同源关系的分析：直系同源 or 旁系同源？
- 直系同源的确定：Reciprocal Best Hits
- 旁系同源的确定：BLAST，序列比对及数据库搜索，至少存在一个共有的功能结构域
- 整体分析/蛋白质家族分析：系统发育树的构建

例：Bub1

- 芽殖酵母的Bub1:定位于动点，纺锤体检验点



获得FASTA序列




```
>sp|P41695|BUB1_YEAST Checkpoint serine/threonine-protein kinase BUB1
MNLDLGSTVRGYESDKDTFPQSKGVSSSQKEQHSQLNQTKIAYEQRLLNDLEDMDDPLDL
FLDYMIWISTSYIEVDSESGQEVLRSTMERCLIYIQDMETYRNDPRFLKIWIWYINLFLS
NNFHESENTFKYMFNKGIGTKLSLFYEEFSKLLENAQFFLEAKVLLELGAENNCRPYNRL
LRSLSNYEDRLREMNIVENQNSVPDSRERLKGRLIYRTAPFFIRKFLTSSLMTDDKENRA
NLNSNVGVGKSAPNVYQDSIVVADFKSETERLNLNSSKQPSNQRLKNGNKKTSIYADQKQ
SNNPVYKLINTPGRKPERIVFNFNLIYPENDEEFNTEEILAMIKGLYKVQRRGKKHTEDY
TSDKNRKKRKLDVLVERRQDLPSSQPPVVPKSTRIEVFKDDDNPSQSTHHKNTQVQVQTT
TSILPLKPVVDGNLAHETPVKPSLTSNASRSPTVTAFSKDAINEVFSMFNQHYSTPGALL
DGDDTTTSKFNVFENFTQEFTAKNIEDLTEVKDPKQETVSQQTTSTNETNDRYERLSNSS
TRPEKADYMTPIKETTETDVVPIIQTPKEQIRTEDKKSGDNTETQTQLTSTTIQSSPFLT
QPEPQAEKLLQTAEHSEKSKEHYPTIIPPFTKIKNQPPVIIENPLSNNLRAKFLSEISPP
LFQYNTFYNYNQELKMSSLLKKIHRVSRNENKNPIVDFKKTGDLYCIRGELGEGGYATVY
LAESSQGHLRALKVEKPASVWEYYIMSQVEFRLRKSTILKSIINASALHLFLDESYLVLN
YASQGTVLDLINLQREKAIDGNGIMDEYLCMFITVELMKVLEKIHEVGIIHGDLKPDNCM
IRLEKPGEPLGAHYMRNGEDGWENKGIYLIDFGRSFDMTLLPPGTKFKSNWKADQQDCWE
MRAGKPWSYEADYYGLAGVIHSMLFGKFIETIQLQNGRCKLKNPFKRYWKKEIWGVIFDL
LLNSGQASNQALPMTEKIVEIRNLIESHLEQHAENHLRNVILSIEEELSHFQYKGKPSRR
F
```

酵母的同源序列：旁系同源

BLAST

Find a protein sequence to run BLAST sequence similarity search by UniProt ID (e.g. P05067 or A4_HUMAN or UPI0000000001).

UniProt IDs

OR

Enter one or more sequences (20 max). You may also load from a text file.

```
>sp|P41695|BUB1_YEAST Checkpoint serine/threonine-protein kinase BUB1 OS=Saccharomyces cerevisiae (strain ATCC 204508 / S288c) OX=559292
MNLDLGSTVRGYESDKDTFPQSKGVSSSQKEQHSQLNQTKIAYEQRLLNDLEDMDDPLDL
FLDYMIWISTSYIEVDSESGQEVLRSTMERCLIYIQDMETYRNDPRFLKIWIWYINLFLS
NNFHESENTFKYMFNKGIGTKLSLFYEEFSKLLENAQFFLEAKVLLELGAENNCRPYNRL
LRSLSNYEDRLREMNIVENQNSVPDSRERLKGRLIYRTAPFFIRKFLTSSLMTDDKENRA
NLNSNVGVGKSAPNVYQDSIVVADFKSETERLNLNSSKQPSNQRLKNGNKKTSIYADQKQ
SNNPVYKLINTPGRKPERIVFNFNLIYPENDEEFNTEEILAMIKGLYKVQRRGKKHTEDY
TSDKNRKKRKLDVLVERRQDLPSSQPPVVPKSTRIEVFKDDDNPSQSTHHKNTQVQVQTT
```

Your input contains 1 sequence

Target database

UniProtKB Swiss-Prot

Restrict by taxonomy

Enter taxon names or IDs to include

Saccharomyces cerevisiae [4932] ×

重置 Run BLAST

Name your BLAST job

Mad3: 旁系同源

UniProt BLAST Align Peptide search ID mapping SPARQL Tool results Advanced | List Search Help

Blast parameters

Identity

17 100

Score

60 5323

E-Value

0 9.9

Status

BLAST 193 results found in UniProtKB

Overview Taxonomy Hit Distribution Text Output Input Parameters API Request

BLAST Align Map IDs Download Add Customize columns Resubmit

Entry	Entry Name	Protein Names	Gene Names	Organism	Length
P41695	BUB1_YEAST	Checkpoint serine/threonine-protein kinase BUB1[...]	**BUB1**, YGR188C, G7542	Saccharomyces cerevisiae (strain ATCC 204508 / S288c) (Baker's yeast)	1,021 AA
P47074	MAD3_YEAST	Spindle assembly checkpoint component MAD3[...]	**MAD3**, YJL013C, J1341	Saccharomyces cerevisiae (strain ATCC 204508 / S288c) (Baker's yeast)	515 AA

100 200 300 400 500 600 700 800 900 1,000

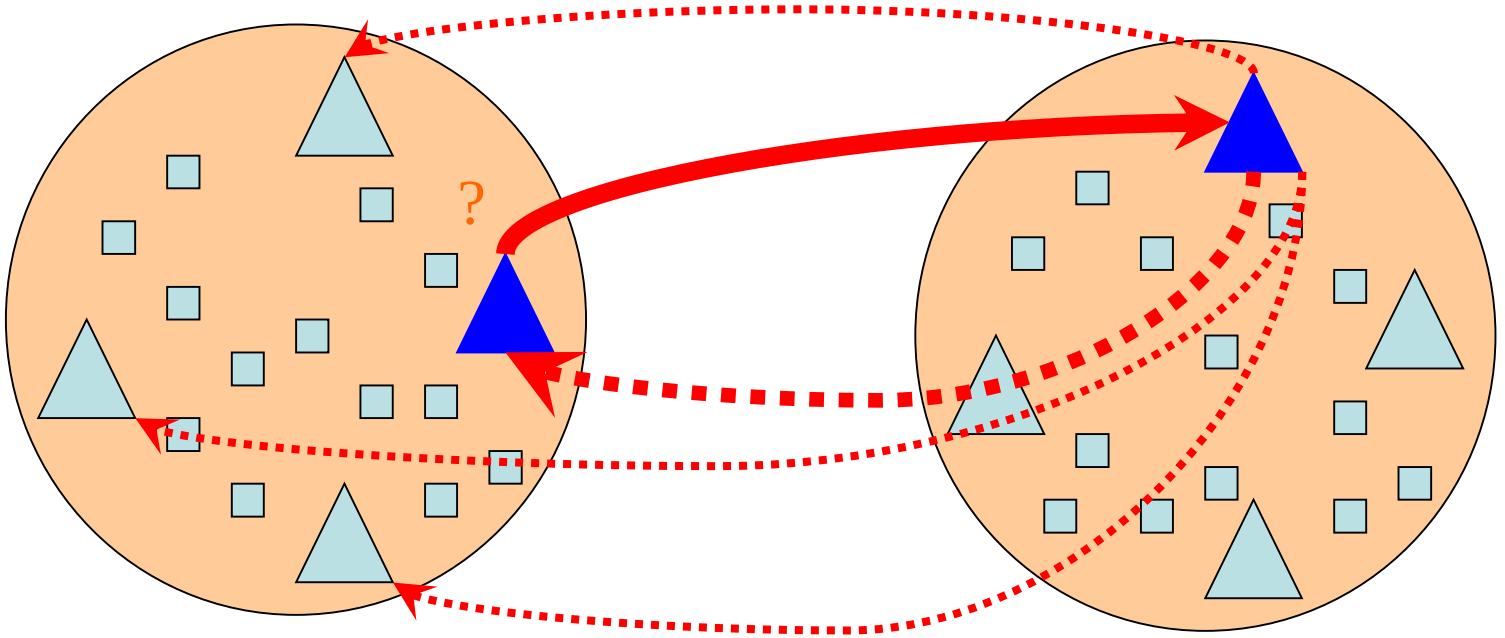
100% 5323

35% 495

Feedback Help

```
>SP:P47074 MAD3_YEAST Spindle assembly checkpoint component MAD3 OS=Saccharomyces
cerevisiae (strain ATCC 204508 / S288c) OX=559292
GN=MAD3 PE=1 SV=1
Length=515

 Score = 195 bits (495),  Expect = 6e-54
 Identities = 124/354 (35%), Positives = 196/354 (55%), Gaps = 33/354 (9%)

Query  35   QLNQTKIAYEQRLLNDLEDMDDPLDLFLDYMIWISTSYIEVDSESGQEVLRSTMERCLIY  94
            ++NQ K ++EQRL+++L  + DP+ L+L+Y+ W++ +Y +     S Q  + + +ERCL +
Sbjct  55   EINQVKSSFEQRLIDELPALSDPITLYLEYIKWLNNAYPQ-GGNSKQSGMLTLLERCLSH  113
```

Reciprocal Best Hits



直系同源序列：Reciprocal Best Hits

人类同源序列：直系同源

BLAST

Find a protein sequence to run BLAST sequence similarity search by UniProt ID (e.g. P05067 or A4_HUMAN or UPI0000000001).

UniProt IDs

OR

Enter one or more sequences (20 max). You may also load from a text file.

```
>sp|P41695|BUB1_YEAST Checkpoint serine/threonine-protein kinase BUB1 OS=Saccharomyces cerevisiae (strain ATCC 204508 / S288c) OX=559292
MNLDLGSTVRGYESDKDTFPQSKGVSSSQKEQHSQLNQTKIAYEQRLLNDLEDMDDPLDL
FLDYMIWISTSYIEVDSESGQEVLRSTMERCLIYIQDMETYRNDPRFLKIWIWYINLFLS
NNFHESENTFKYMFNKGIGTKLSLFYEEFSKLLENAQFFLEAKVLLELGAENNCRPYNRL
LRSLSNYEDRLREMNIVENQNSVPDSRERLKGRLIYRTAPFFIRKFLTSSLMTDDKENRA
NLNSNVGVGKSAPNVYQDSIVVADFKSETERLNLNSSKQPSNQRLKNGNKKTSIYADQKQ
SNNPVYKLINTPGRKPERIVFNFNLIYPENDEEFNTEEILAMIKGLYKVQRRGKKHTEDY
TSDKNRKKRKLDVLVERRQDLPSSQPPVVPKSTRIEVFKDDDNPSQSTHHKNTQVQVQTT
```

Your input contains 1 sequence

Target database

UniProtKB Swiss-Prot

Restrict by taxonomy

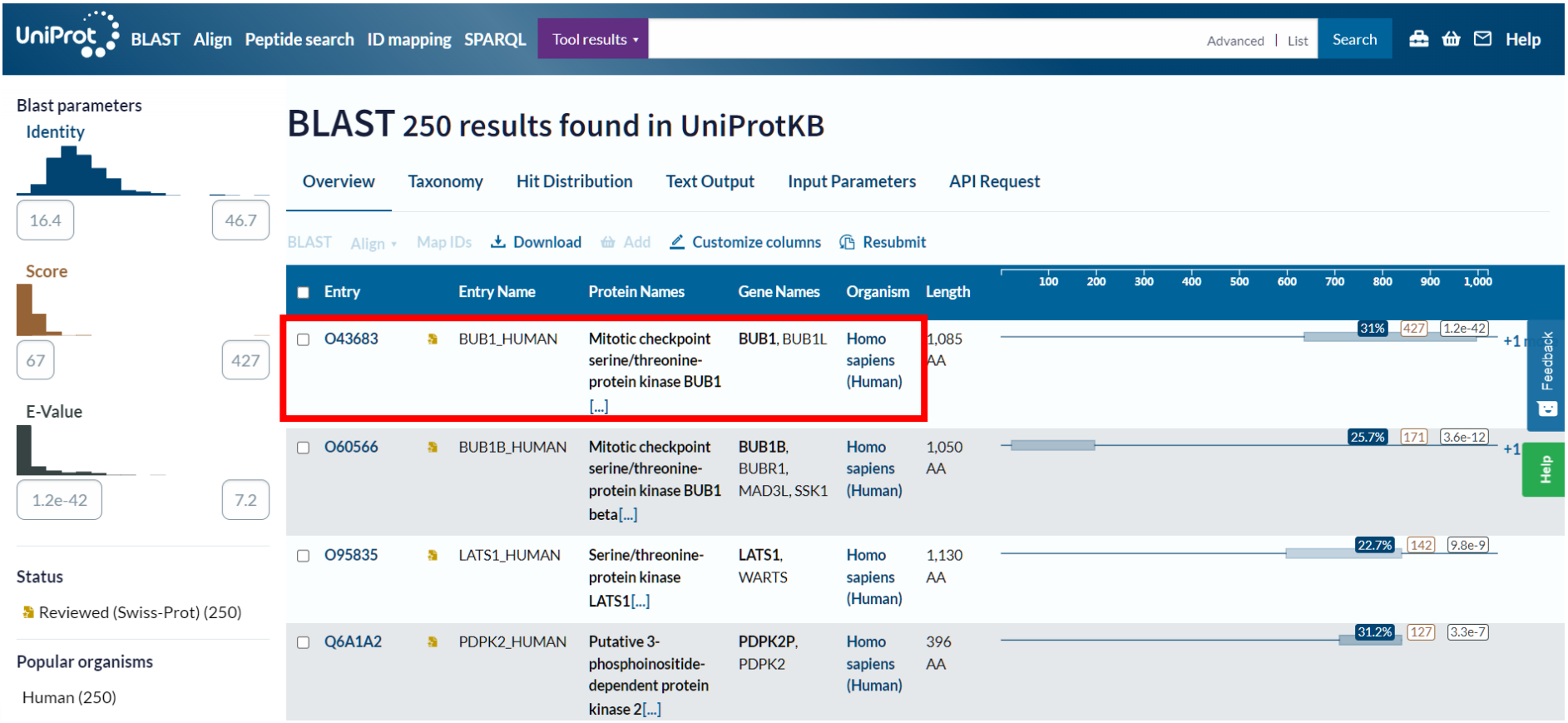
Enter taxon names or IDs to include

Homo sapiens [9606] ×

重置 Run BLAST

Name your BLAST job

人类Bub1？


UniProt
BLAST
Align
Peptide search
ID mapping
SPARQL
Tool results
Advanced | List
Search
Help
Blast parameters
Identity
16.4
46.7
Score
67
427
E-Value
1.2e-42
7.2
Status
Reviewed (Swiss-Prot) (250)
Popular organisms
Human (250)
BLAST 250 results found in UniProtKB
Overview
Taxonomy
Hit Distribution
Text Output
Input Parameters
API Request
BLAST
Align
Map IDs
Download
Add
Customize columns
Resubmit
Entry
Entry Name
Protein Names
Gene Names
Organism
Length
100
200
300
400
500
600
700
800
900
1,000
O43683
BUB1_HUMAN
Mitotic checkpoint serine/threonine-protein kinase BUB1 [...]
BUB1, BUB1L
Homo sapiens (Human)
1,085 AA
31%
427
1.2e-42
O60566
BUB1B_HUMAN
Mitotic checkpoint serine/threonine-protein kinase BUB1 beta[...]
BUB1B, BUBR1, MAD3L, SSK1
Homo sapiens (Human)
1,050 AA
25.7%
171
3.6e-12
O95835
LATS1_HUMAN
Serine/threonine-protein kinase LATS1[...]
LATS1, WARTS
Homo sapiens (Human)
1,130 AA
22.7%
142
9.8e-9
Q6A1A2
PDPK2_HUMAN
Putative 3-phosphoinositide-dependent protein kinase 2[...]
PDPK2P, PDPK2
Homo sapiens (Human)
396 AA
31.2%
127
3.3e-7
Feedback
Help

在酵母中做比对

BLAST

Find a protein sequence to run BLAST sequence similarity search by UniProt ID (e.g. P05067 or A4_HUMAN or UPI0000000001).

UniProt IDs

OR

Enter one or more sequences (20 max). You may also load from a text file.

```
>sp|O43683|BUB1_HUMAN Mitotic checkpoint serine/threonine-protein kinase BUB1 OS=Homo sapiens OX=9606 GN=BUB1 PE=1 SV=1
MDTPENVLQMLEAHMQSYKGNDPLGEWERYIQWVEENFPENKEYLITLLEHLMKEFLDKK
KYHNDPRFISYCLKFAEYNSDLHQFFEFLYNHGIGTLSSPLYIAWAGHLEAQGELQHASA
VLQRGIQNQAEPREFLQQQYRLFQTRLTETHLPAQARTSEPLHNVQVLNQMITSKSNPGN
NMACISKNQGSELSGVISSACDKESNMERRVITISKSEYSVHSSLASKVDVEQVVMYCKE
KLIRGESEFSFEELRAQKYNQRRKHEQWVNEDRHYMKRKEANAFEEQLLKQKMDELHKKL
HQVVETSHEDLPASQERSEVNPARMGPSVGSQQELRAPCLPVTYQQTPVNMEKNPREAPP
VVPPLANAISAALVSPATSQSIAPPVPLKAQTVTDSMFAVASKDAGCVNKSTHEFKPQSG
AFIKEGCETHKVANTSSEHTTPNTSLGMVQATPSKVQPSPTVHTKEALGEIMNMEQAPTL
```

Your input contains 1 sequence

Target database

UniProtKB Swiss-Prot

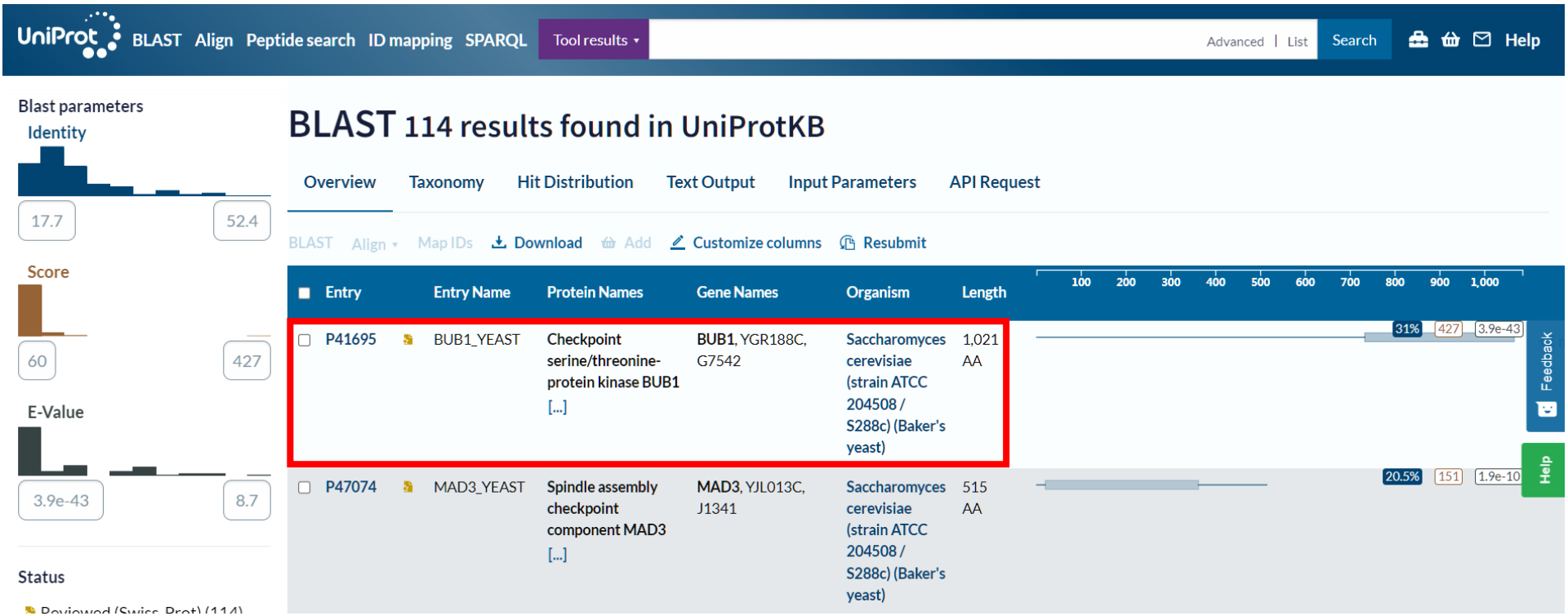
Restrict by taxonomy

Enter taxon names or IDs to include

Saccharomyces cerevisiae [4932] ×

重置 Run BLAST

Best Hit!




```
>SP:P41695 BUB1_YEAST Checkpoint serine/threonine-protein kinase BUB1 OS=Saccharomyces
cerevisiae (strain ATCC 204508 / S288c) OX=559292
GN=BUB1 PE=1 SV=2
Length=1021

 Score = 169 bits (427),  Expect = 4e-43
 Identities = 115/371 (31%), Positives = 190/371 (51%), Gaps = 46/371 (12%)

Query  734   PNFIVGNPWDDKLIFKLLSGLSKPVSSYPNTF---EWQCKLPAI----------KPK  777
             P  I+ NP  + L  K LS +S P+  Y NTF     + K+ ++              P
Sbjct  637   PPVIIENPLSNNLRAKFLSEISPPLFQY-NTFYNYNQELKMSSLLKKIHRVSRNENKNPI  695
```