BLAST算法介绍

亢雨笺

2013年11月1日

序列比对

Nothing in Biology Makes Sense
Except in the Light of Evolution

Theodosius Dobzhansky (1900-1975)

生物中最被关注的DNA、RNA、蛋白质，都具有线性的序列信息

研究序列相似性

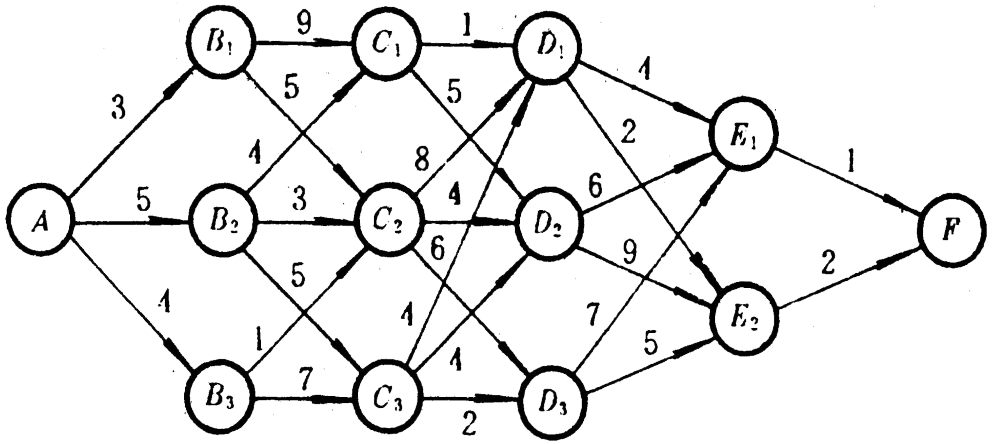
关注其功能、演化历史

全局比对：Needleman-Wunsch

- 无法通过穷举得到所有的两两比对结果
- **动态规划**

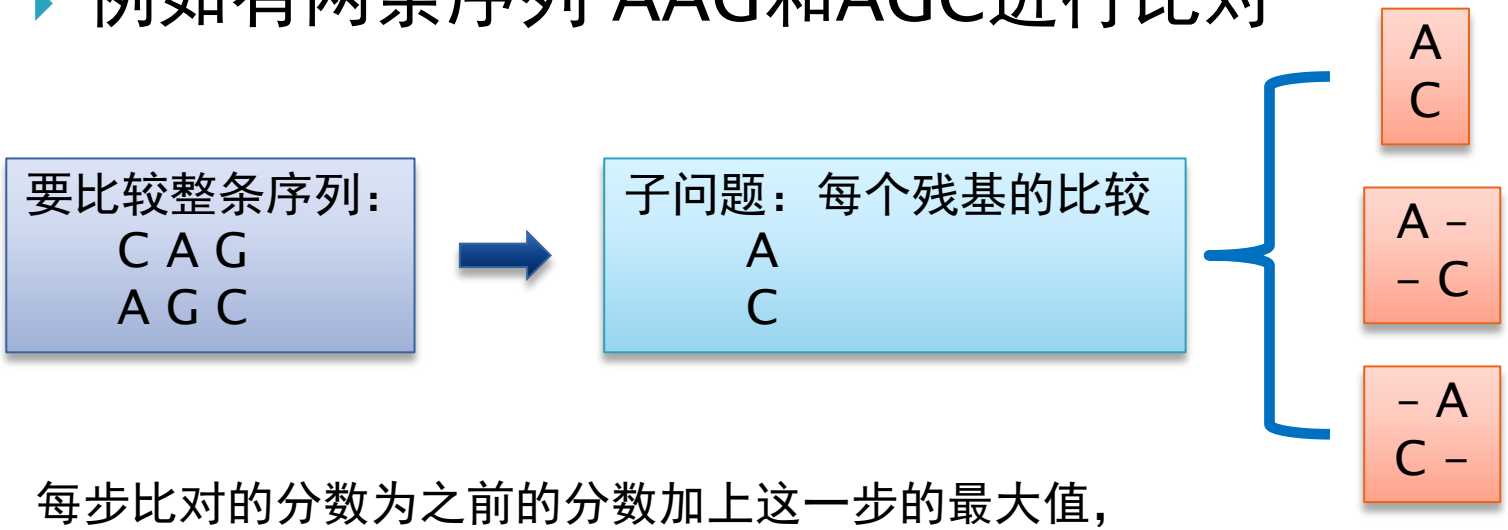
一个大问题可以分成若干个子问题

寻找每个子问题的最优解，就是最终的最优解



全局比对：Needleman-Wunsch

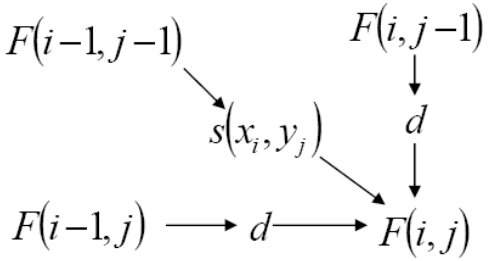
- 例如有两条序列 AAG和AGC进行比对



每步比对的分数为之前的分数加上这一步的最大值，
对应公式为：

$$F(i,j) = \max \begin{cases} F(i-1,j-1) + s(x_i, y_j), \\ F(i-1,j) - d, \\ F(i,j-1) - d. \end{cases}$$

全局比对：Needleman-Wunsch



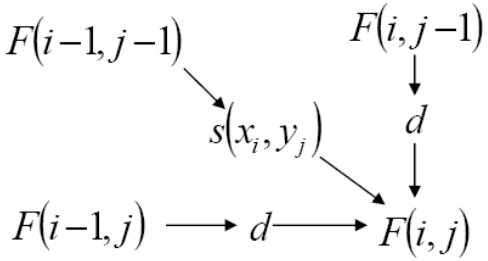
假设比较序列　A A G
A G C

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

		A	A	G
	0			
A				
G				
C				

gap open = -5　　gap extension = -5

全局比对：Needleman-Wunsch



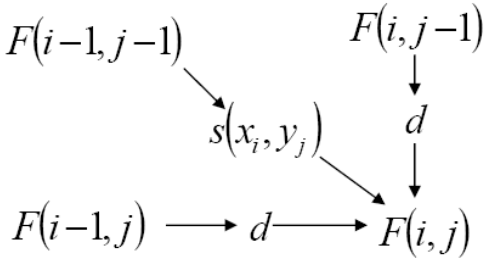
假设比较序列　A A G
A G C

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

		A	A	G
	0 →	-5 →	-10 →	-15
A	-5			
G	-10			
C	-15			

gap open = -5　　gap extension = -5

全局比对：Needleman-Wunsch



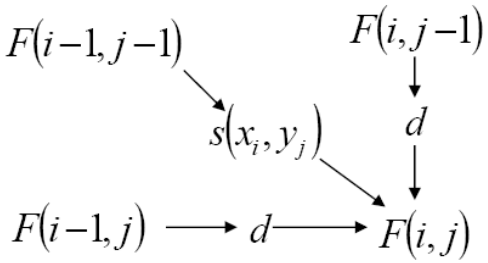
	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

假设比较序列　A A G
A G C

		A	A	G
	0 →	-5 →	-10 →	-15
A	-5	2 →	-3 →	-8
G	-10	-3	-3	-1
C	-15	-8	-8	**-6**

gap open = -5　　gap extension = -5

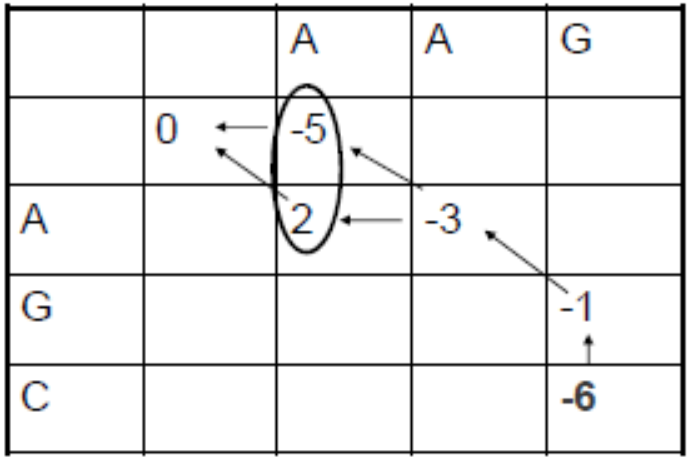
全局比对：Needleman-Wunsch



A A G –
A – G C

A A G –
– A G C

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2



gap open = -5 gap extension = -5

局部比对： Smith-Waterman

- 需要比对序列的结构域，而不是整条序列时，Needleman-Wunsch算法并不适用

$$F(0,0)=0$$

$$F(i,j)=\max\begin{cases}F(i-1,j-1)+s(x_i,y_j)\\F(i-1,j)+d\\F(i,j-1)+d\\0\end{cases}$$

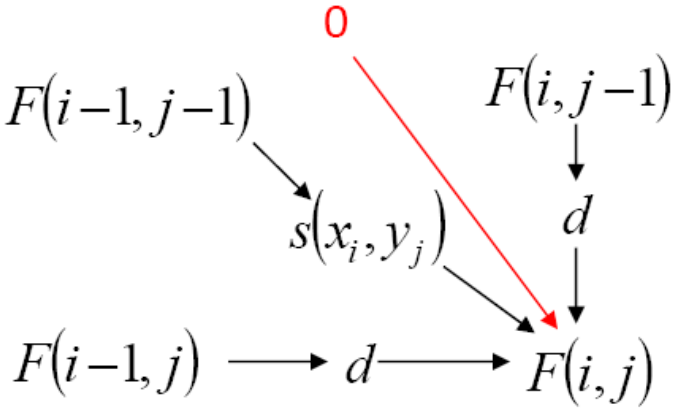
同样考虑比较序列

A A G

A G C

		A	A	G
A				
G				
C				

局部比对： Smith-Waterman



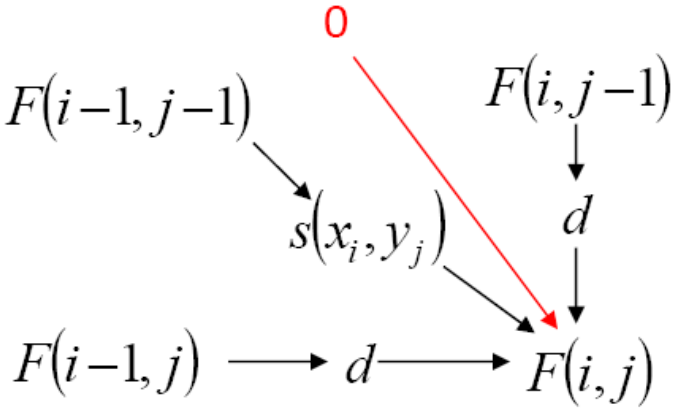
假设比较序列　A A G
A G C

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

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

gap open = −5　　gap extension = −5

局部比对： Smith-Waterman



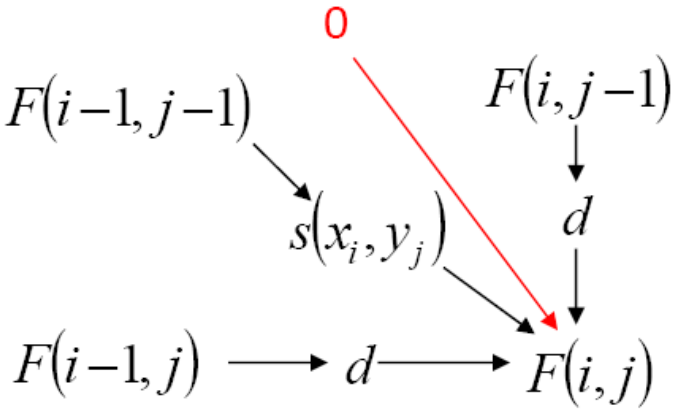
假设比较序列 A A G
A G C

	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

		A	A	G
	0	0	0	0
A	0	2	2	0
G	0	0	0	4
C	0	0	0	0

gap open = −5 gap extension = −5

局部比对： Smith-Waterman



	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

A G
A G

A
A

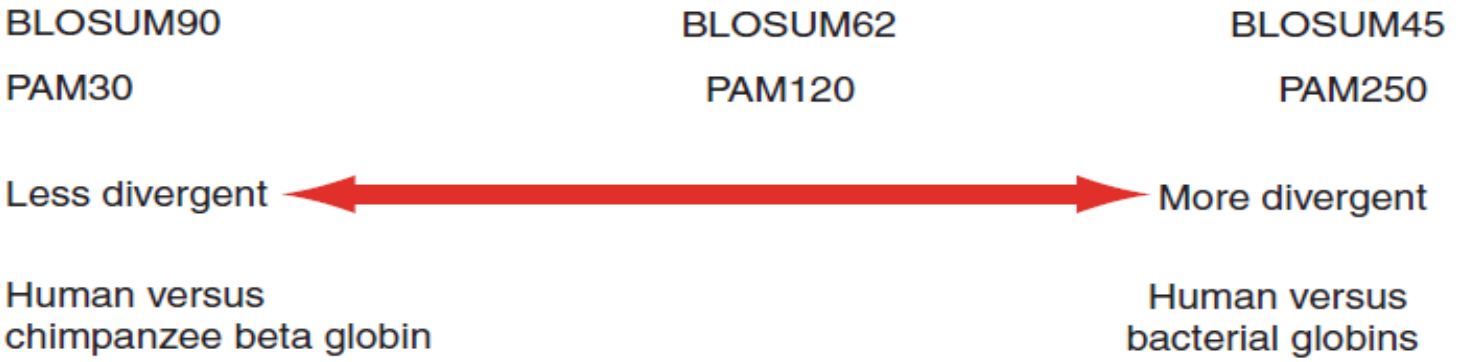
		A	A	G
	0	0	0	0
A	0	2	2	0
G	0	0	0	4
C	0	0	0	0

gap open = −5 gap extension = −5

计分矩阵：PAM和BLOSUM

- PAM矩阵是基于近相关蛋白数据的，并且假设高度相关蛋白的取代概率可以外推到远相关蛋白的概率
- 而BLOSUM矩阵是基于实际观测到的远相关蛋白构建的

因此在比对较远蛋白时，应选BLOSUM矩阵



(Jonathan Pevsner, *Bioinformatics and Functional Genomics)*

计分矩阵：BLOSUM矩阵

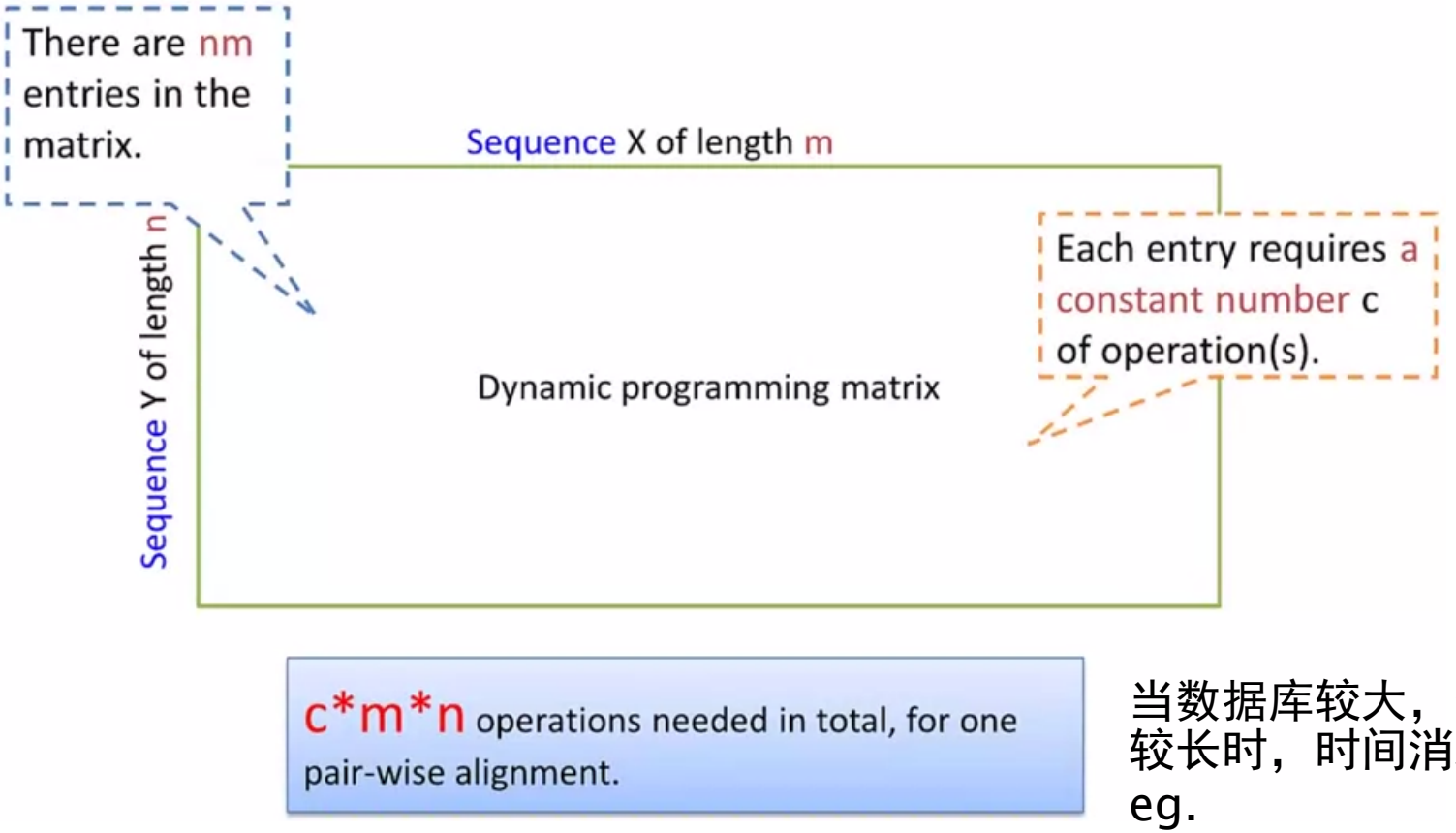
	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	B	Z	X	*
A	4	-1	-2	-2	0	-1	-1	0	-2	-1	-1	-1	-1	-2	-1	1	0	-3	-2	0	-2	-1	0	-4
R	-1	5	0	-2	-3	1	0	-2	0	-3	-2	2	-1	-3	-2	-1	-1	-3	-2	-3	-1	0	-1	-4
N	-2	0	6	1	-3	0	0	0	1	-3	-3	0	-2	-3	-2	1	0	-4	-2	-3	3	0	-1	-4
D	-2	-2	1	6	-3	0	2	-1	-1	-3	-4	-1	-3	-3	-1	0	-1	-4	-3	-3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

BLOSUM62

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	B	Z	X	*
A	7	-3	-3	-3	-1	-2	-2	0	-3	-3	-3	-1	-2	-4	-1	2	0	-5	-4	-1	-3	-2	-1	-8
R	-3	9	-1	-3	-6	1	-1	-4	0	-5	-4	3	-3	-5	-3	-2	-2	-5	-4	-4	-2	0	-2	-8
N	-3	-1	9	2	-5	0	-1	-1	1	-6	-6	0	-4	-6	-4	1	0	-7	-4	-5	5	-1	-2	-8
D	-3	-3	2	10	-7	-1	2	-3	-2	-7	-7	-2	-6	-6	-3	-1	-2	-8	-6	-6	6	1	-3	-8
C	-1	-6	-5	-7	13	-5	-7	-6	-7	-2	-3	-6	-3	-4	-6	-2	-2	-5	-5	-2	-6	-7	-3	-8
Q	-2	1	0	-1	-5	9	3	-4	1	-5	-4	2	-1	-5	-3	-1	-1	-4	-4	-4	-1	5	-2	-8
E	-2	-1	-1	2	-7	3	8	-4	0	-6	-6	1	-4	-6	-2	-1	-1	-5	-4	-4	0	6	-2	-8
G	0	-4	-1	-3	-6	-4	-4	9	-4	-7	-7	-3	-5	-6	-5	-1	-3	-6	-5	-6	-1	-2	-3	-8
H	-3	0	1	-2	-7	1	0	-4	12	-6	-5	-1	-4	-2	-4	-3	-3	-4	-3	-5	0	1	-3	-8
I	-3	-5	-6	-7	-2	-5	-6	-7	-6	7	2	-5	2	-1	-5	-4	-4	-4	-2	4	-6	-5	-2	-8
L	-3	-4	-6	-7	-3	-4	-6	-7	-5	2	6	-4	3	0	-5	-4	-3	-3	-2	1	-7	-4	-2	-8
K	-1	3	0	-2	-6	2	1	-3	-1	-5	-4	8	-3	-5	-2	-1	-1	-6	-4	-4	0	1	-2	-8
M	-2	-3	-4	-6	-3	-1	-4	-5	-4	2	3	-3	9	0	-4	-3	-2	-4	-3	2	-5	-3	-2	-8
F	-4	-5	-6	-6	-4	-5	-6	-6	-2	-1	0	-5	0	10	-6	-4	-4	0	4	-2	-6	-5	-3	-8
P	-1	-3	-4	-3	-6	-3	-2	-5	-4	-5	-5	-2	-4	-6	12	-2	-3	-7	-6	-4	-4	-2	-3	-8
S	2	-2	1	-1	-2	-1	-1	-1	-2	-4	-4	-1	-3	-4	-2	7	2	-6	-3	-3	0	-1	-1	-8
T	0	-2	0	-2	-2	-1	-2	-3	-3	-2	-3	-1	-1	-4	-3	2	8	-5	-3	0	-1	-2	-1	-8
W	-5	-5	-7	-8	-5	-4	-6	-6	-4	-5	-4	-6	-3	0	-7	-6	-5	16	3	-5	-8	-5	-5	-8
Y	-4	-4	-4	-6	-5	-3	-5	-6	3	-3	-2	-4	-3	4	-6	-3	-3	3	11	-3	-5	-4	-3	-8
V	-1	-4	-5	-6	-2	-4	-4	-6	-5	4	1	-4	1	-2	-4	-3	0	-5	-3	7	-6	-4	-2	-8
B	-3	-2	5	6	-6	-1	1	-2	-1	-6	-7	-1	-5	-6	-4	0	-1	-8	-5	-6	6	0	-3	-8
Z	-2	0	-1	1	-7	5	6	-4	0	-6	-5	1	-3	-6	-2	-1	-2	-5	-4	-4	0	6	-1	-8
X	-1	-2	-2	-3	-4	-2	-2	-3	-2	-2	-2	-2	-2	-3	-3	-1	-1	-5	-3	-2	-3	-1	-2	-8
*	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	1

BLOSUM80

Smith-Waterman算法的局限性



当数据库较大，查询序列较多或较长时，时间消耗太大

eg.

142*192206270*1μs≈7.5hrs

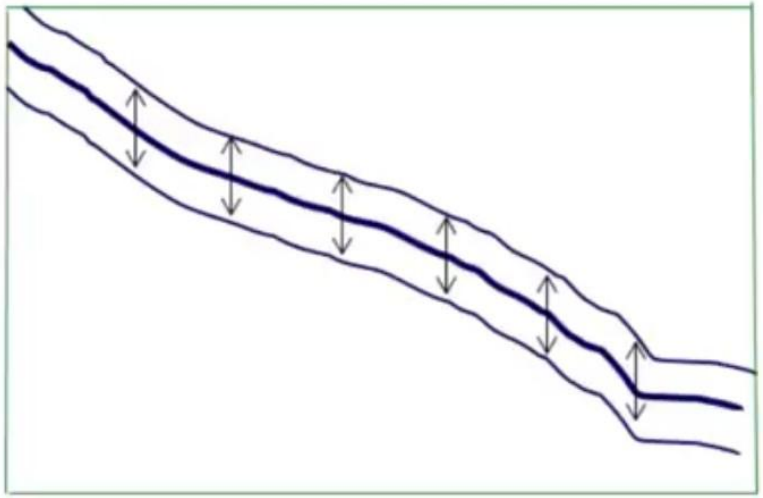
HBA_HUMAN　SwissProt氨基酸数

启发式(heuristic)算法：BLAST

- FASTA (Pearson和Lipman,1988)
- Basic Local Alignment Search Tool (Altschul等,1990)

△启发式算法：以牺牲灵敏度（sensitivity）为代价，提升计算速度
△与Smith-Waterman算法不同，不能保证找到最佳匹配

		A	A	G
	0	0	0	0
A	0	2	2	0
G	0	0	0	4
C	0	0	0	0



BLAST核心：seeding and extending

找种子序列

↓

在数据库中定位种子

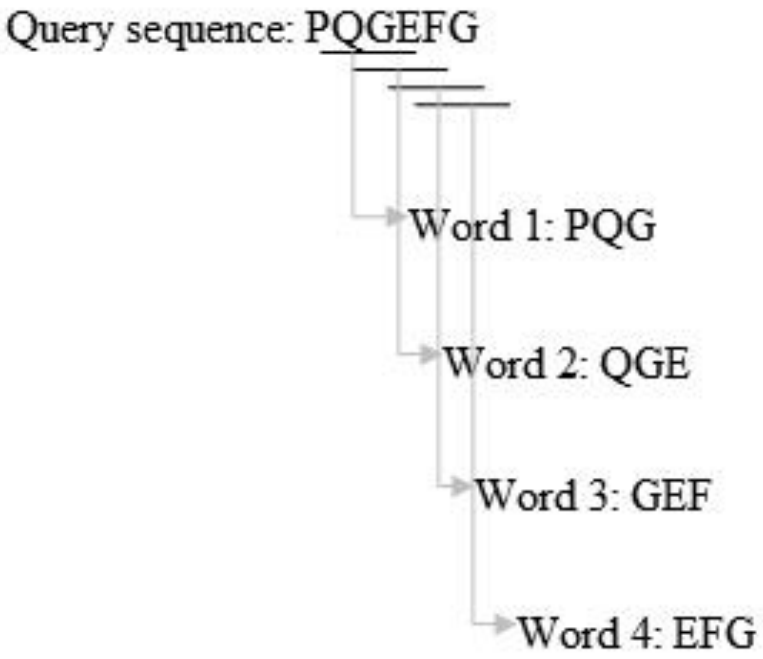
↓

延伸匹配

Seeding：划分查询序列

- 把查询序列划成kmer,找所有覆盖到的种子序列

  k-mer words长度w,最后得到n-w+1个字串

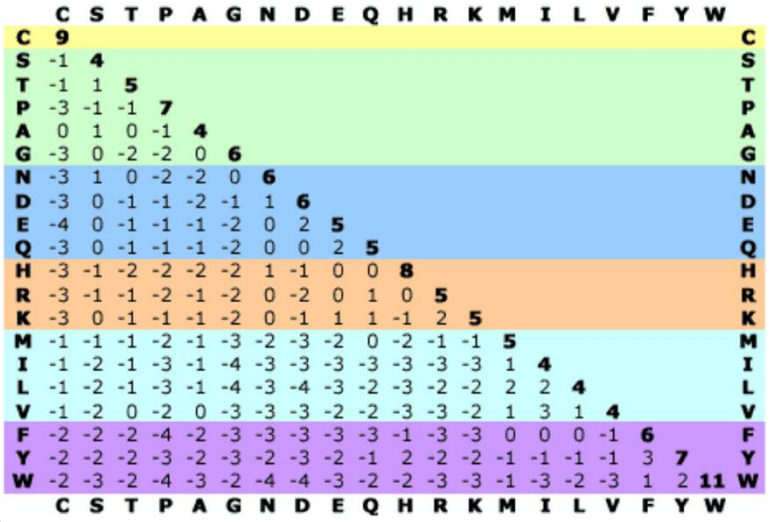


一般来说：
对蛋白w=3，核酸w=11

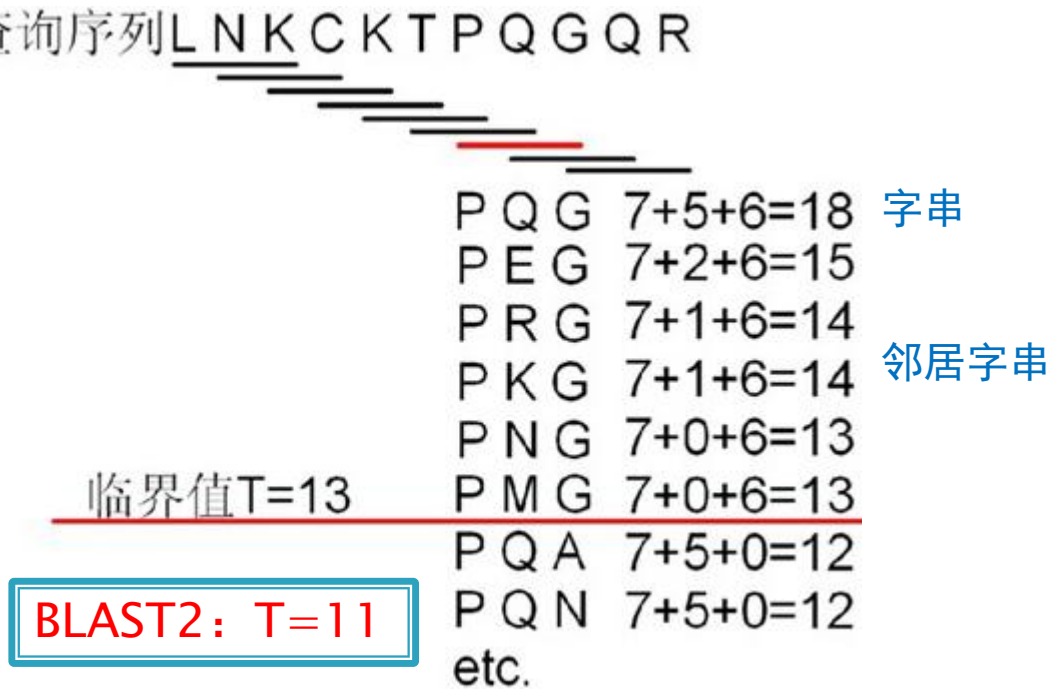
种子越短：
灵敏度越高
计算速度越慢

Seeding：创建序列的查询列表

1.根据查询序列划分出的字串
2.这些单词分数高于neighborhood word score threshold（T）的邻居字串
（这个分数根据计分矩阵得到，我们这里以BLOSUM62矩阵为例）

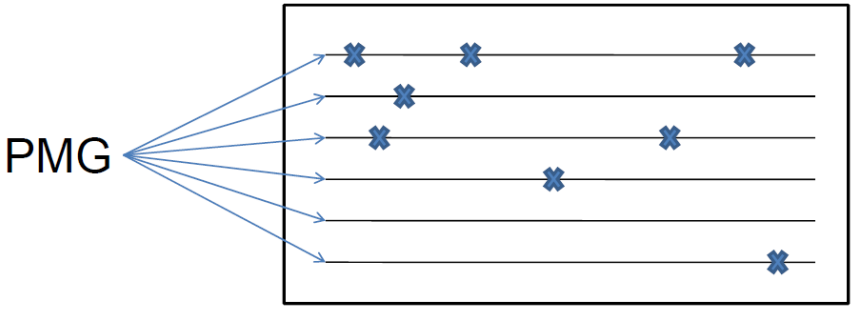


	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W	
C	9																				C
S	-1	4																			S
T	-1	1	5																		T
P	-3	-1	-1	7																	P
A	0	1	0	-1	4																A
G	-3	0	-2	-2	0	6															G
N	-3	1	0	-2	-2	0	6														N
D	-3	0	-1	-1	-2	-1	1	6													D
E	-4	0	-1	-1	-1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



Seeding：在数据库中定位种子

- 对于单词列表中的每一个字串，在所有的数据库序列中找到其出现的每一个位置。



由于数据库预先有建立索引，因此查询种子找到match是非常快的。

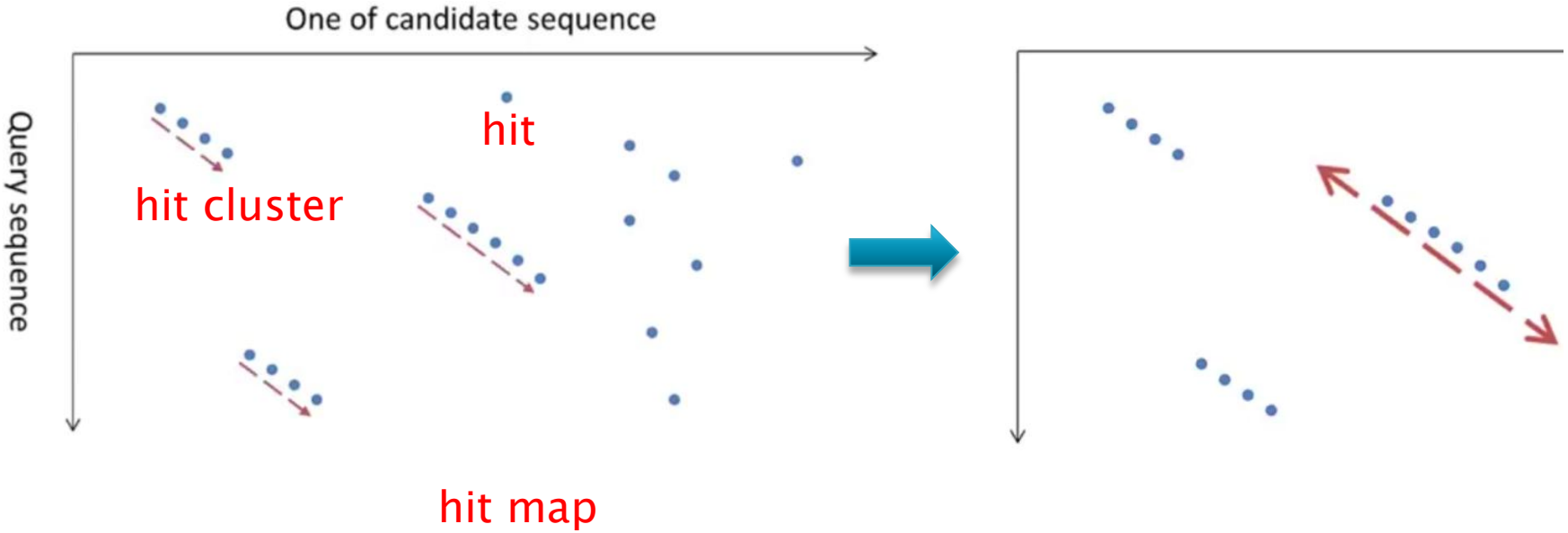
可以利用如下方法：

利用hash建index

后缀树

Aho-Corasick自动机算法

Extending：从hit到HSP



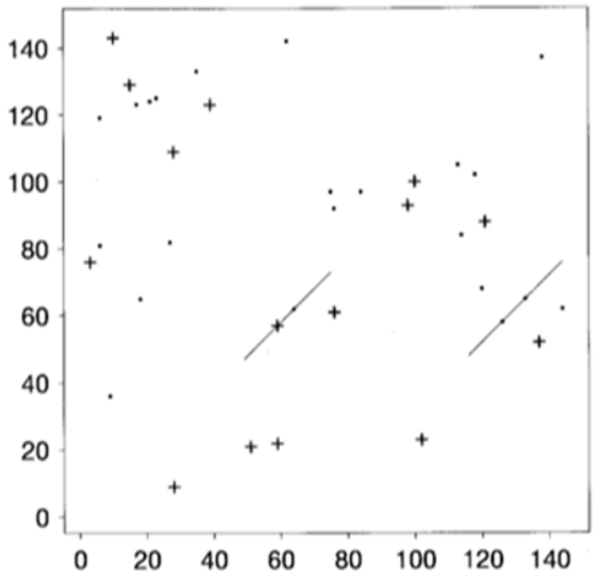
- 利用打分矩阵沿左右两个方向延伸hit cluster直到打分低于一个临界值，得到的结果称为高分片段对（high-scoring segment pair, HSP）。

Extending：hit延伸的条件

- 两个hits没有overlap
- 两个hits有同样的diagonal

Diagonal：
某个word在数据库序列中起始位置是x_1，在查询序列中起始位置是x_2，$x_1 - x_2$即为diagonal

- 两个hits在同一个window内（window length一般为40）



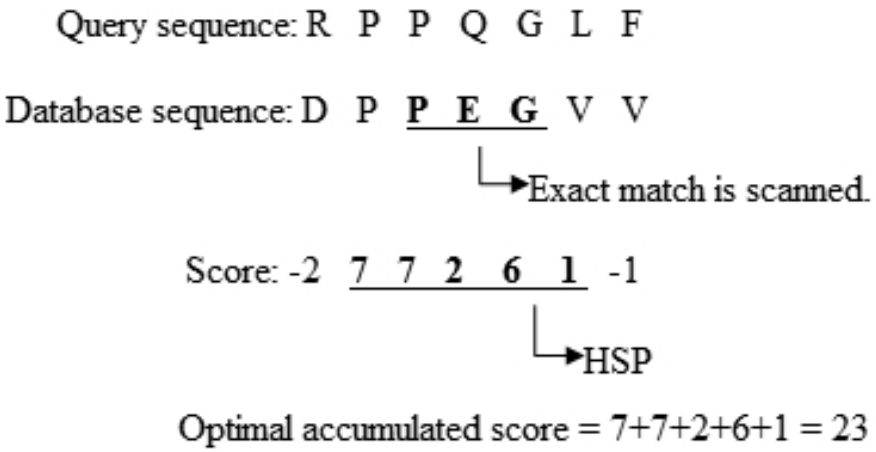
Extending：动态规划得到HSP

- 通过局部实现Smith-Waterman算法，将之前得到的hit cluster的匹配进行延伸，直到分值降低到某个阈值时停止。从而得到为高分片段对（HSP）

- 有时也将一个数据库序列中的多个HSP区域结合成一个更长的比对结果

$$F(0,0)=0$$

$$F(i,j)=\max\begin{cases}F(i-1,j-1)+s(x_i,y_j)\\F(i-1,j)+d\\F(i,j-1)+d\\0\end{cases}$$



Extending：合并HSP为较长比对

- 对于相邻或距离较近的HSP，可以把它们合并

当需要比较这些结合区域之间分值高低时，有以下两种方法：

1. Poisson法则（Poisson method）（old）
2. 总分法则（sum-of scores method）

Eg.（65, 40）和（52, 45）

Poisson法则：（52, 45） →45>40

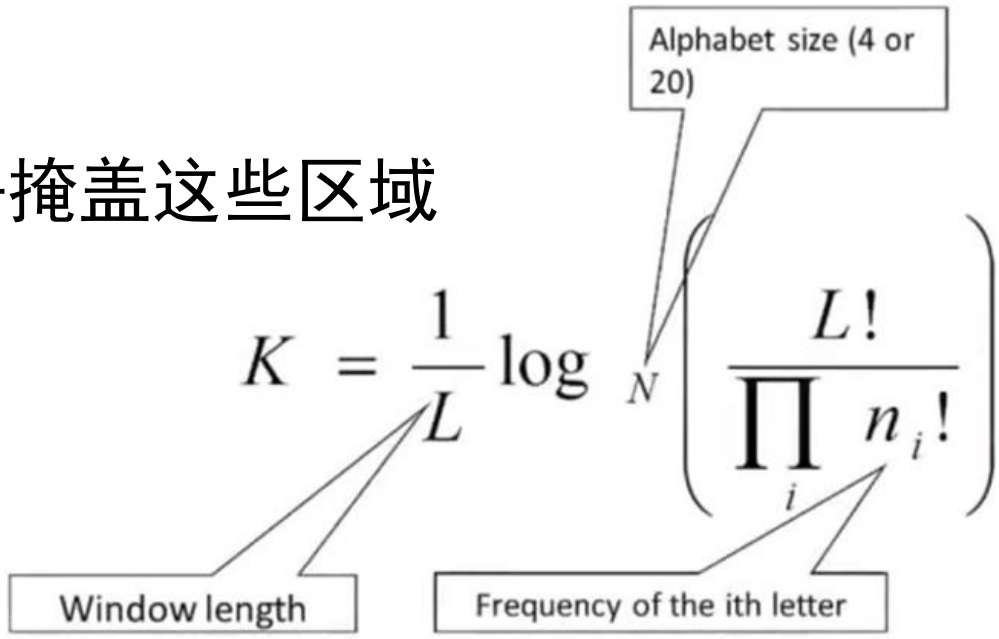
总分法则：（65, 40）

→ 65+40（105）>52+45（97）

Speedup:移除低复杂度区域

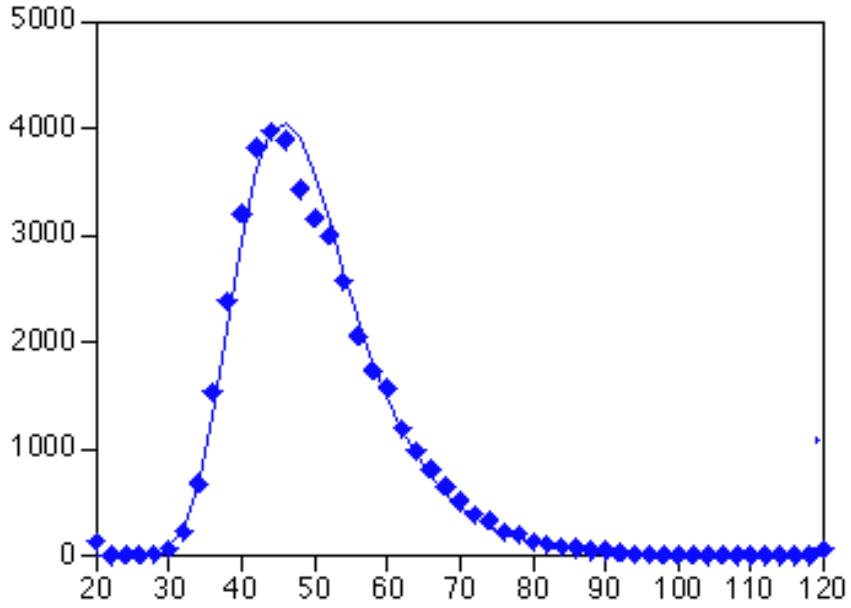
- 查询序列中一些低复杂度区域会带来假阳性，例如：微卫星序列
- CACACACACACACACA
- AAAAAAAAAAAA
- KLKLKLKLKLKLKL

- 因此需要用如下字母去掩盖这些区域
- Ns（核酸）
- Xs（蛋白）



E value：比对结果的统计显著性

- 将查询序列与一系列统一长度的随机序列进行比对时，分值通常符合Gumbel极值分布。



累积概率分布：

$$F(x;\mu,\beta) = e^{-e^{-(x-\mu)/\beta}}.$$

E value：比对结果的统计显著性

- 因此对于序列m和n，观察到一个大于等于x的比对分数S的概率：

$$p(S \geq x) = 1 - \exp\left(-e^{-\lambda(x-\mu)}\right)$$

要使用该式，必须知道λ和μ。

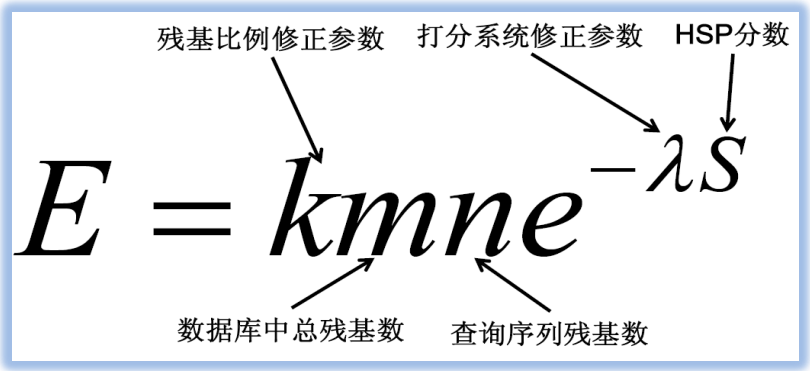
$\mu = [\log(Kmn)]/\lambda \Rightarrow P(S \geq x) = 1 - \exp(-Kmne^{-\lambda x})$

K、 λ：与打分矩阵相关的参数
m: 查询序列长度
n:数据库大小

改进后BLAST有将m、n进行矫正用m'和n'（有效长度）来计算

E value：比对结果的统计显著性

- E value：在随机情况下，获得比当前比对分数相等或更高的可能比对条数。（期望值）



- Bit score S’（比特分数）

$$E = Kmn\,e^{-\lambda S} \xrightarrow{S' = \frac{\lambda S - \ln K}{\ln 2}} E = mn\,2^{-S'}$$

将原始分数对打分系统的变量进行归一化，使不同的BLAST搜索结果可以进行比较

E值与P值的关系

- P value：分值大于等于要求分值的比对的随机发生概率

$P \rightarrow 0$,显著性越高

- 将随机序列（长度与实际查询序列一致）作为查询序列了搜索数据库，联系实际查询序列得到的比对分值得到：

$$P = 1 - e^{-E}$$

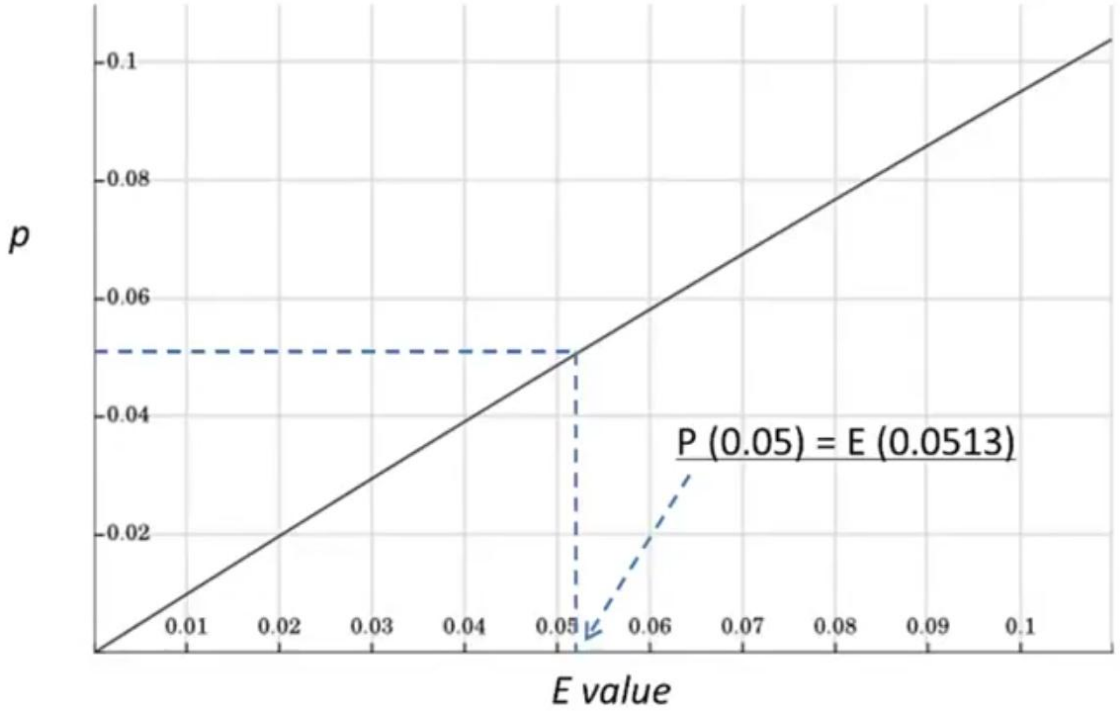
- P value和E value是反映比对显著性的两种不同方式

E值与P值的关系

当E < 0.1：E ≈ P

（BLAST的结果只列出了E值）

$$P = 1 - e^{-E}$$



E值和P值关系：

E	P	E	P
10	0.99995460	0.1	0.09516258
5	0.99326205	0.05	0.04877058
2	0.86466472	0.001	0.00099950
1	0.63212056	0.0001	0.0001000

(Jonathan Pevsner, *Bioinformatics and Functional Genomics)*

PSI-BLAST：位点特异性反复比对

- 数据库中某些蛋白相关性较小，搜索效果差
- PSI-BLAST比常规算法更敏感，主要用于搜索与我们感兴趣蛋白远缘相关的蛋白。

PSI-BLAST：位点特异打分矩阵

- 传统BLAST对计分矩阵依赖较大
  HSP的分值均依赖于固定的计分矩阵的均值

- **每次比对都特异建立一个新的计分矩阵**
  位点特异计分矩阵（PSSM）

使不能被搜索到的远缘蛋白被比对上

PSI-BLAST：位点特异打分矩阵

- 位点特异计分矩阵（PSSM）

行：20个氨基酸

列：查询序列

分数：

每个位点的$\lg(Q_i/P_i)$

Q_i- 残基i在该位点出现的估计概率

P_i - 这个残基的背景概率

(A)

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
1 V	-3	-5	-5	-5	-3	-5	-5	-6	-5	4	0	-5	-1	-3	-5	-4	-2	-5	-3	6
2 N	-1	-3	7	0	-4	-2	-2	-3	-2	-4	-1	-1	-3	-5	-4	2	-2	-6	-4	-1
3 P	0	-4	-4	-3	-5	-3	0	-4	-4	-4	-5	-3	-5	-6	8	0	-3	-6	-5	-5
4 K	-2	0	-4	-4	-5	-3	-2	-5	-4	-4	-3	6	-1	-3	-5	-2	-2	-5	-3	-2
5 A	5	-4	-4	-5	-4	-4	-4	-3	-3	-1	-4	-4	-4	-3	-1	-1	-1	-3	-5	-2
6 Y	-2	-4	-2	-3	-5	-1	-3	-4	-3	-1	-3	-2	-4	-2	-3	-0	-1	-4	-3	-2
7 P	-2	-5	-5	-3	-6	-4	-2	-5	-5	-5	-1	-4	-5	-6	-8	-3	-4	-6	-6	-5
8 L	-3	-4	-5	-4	-1	-0	-4	-6	-4	-2	-4	-2	-1	-2	-5	-0	-3	-5	-4	-1
9 A	-5	-4	-4	-1	-4	-0	-2	-2	-4	-4	-3	-3	-2	-4	-1	-0	-4	-5	-5	-2
10 D	-2	-1	-1	-4	-6	-2	-1	-0	-4	-5	-4	-2	-1	-6	-3	-1	-2	-6	-5	-4
11 A	-3	-0	-3	-1	-5	-2	-2	-1	-1	-5	-4	-2	-0	-6	-2	-1	-2	-6	-5	-4
12 H	-0	-0	-1	-1	-6	-1	-4	-4	-1	-5	-2	-4	-1	-6	-4	-1	-0	-6	-5	-5
13 L	-1	-2	-5	-5	-5	-2	-4	-2	-5	-1	-4	-3	-3	-4	-3	-1	-0	-5	-4	-0
14 T	-2	-0	-2	-4	-0	-1	-0	-4	-1	-3	-3	-3	-3	-5	-0	-0	-4	-6	-5	-3
15 K	-1	-2	-0	-1	-6	-3	-2	-1	-1	-3	-4	-5	-4	-6	-4	-2	-0	-6	-5	-4
16 K	-0	-4	-3	-2	-6	-2	-0	-4	-3	-5	-4	-6	-1	-6	-3	-0	-2	-6	-5	-4
17 L	-0	-4	-5	-3	-3	-5	-2	-4	-6	-3	-4	-3	-1	-3	-2	-3	-2	-5	-4	-1
18 L	-1	-2	-5	-4	-5	-1	-2	-4	-1	-2	-4	-2	-0	-1	-5	-3	-1	-3	-4	-2
19 D	-0	-1	-0	-3	-5	-1	-3	-4	-4	-5	-5	-5	-5	-6	-3	-0	-1	-6	-3	-4
20 L	-1	-0	-5	-4	-0	-3	-2	-3	-3	-1	-4	-0	-1	-2	-2	-2	-0	-3	-4	-2
21 V	-2	-4	-5	-4	-3	-5	-3	-3	-5	-3	-2	-3	-1	-0	-3	-2	-2	-5	-4	-4
22 Q	-2	-2	-0	-2	-6	-4	-2	-4	-1	-5	-5	-5	-4	-6	-2	-0	-3	-6	-5	-4
23 Q	-1	-1	-1	-2	-6	-4	-0	-4	-1	-2	-3	-5	-2	-6	-3	-1	-1	-6	-5	-2

(B)

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
1 V	-6	-12	-6	-12	-17	-13	-9	-18	-15	0	-9	-3	-10	-11	-6	-4	-9	-22	-15	-2
2 N	-1	-16	-10	-17	-21	-17	-13	-21	-18	-1	-13	-7	-14	-14	-5	-5	-10	-25	-10	-4
3 P	-3	-12	-8	-20	-26	-16	-16	-25	-21	0	-16	-5	-18	-12	-8	-5	-11	-21	-7	-6
4 K	-5	-17	-13	-23	-32	-20	-18	-30	-26	-5	-17	-9	-23	-18	0	-8	-15	-27	-13	-11
5 A	-5	-16	-13	-22	-30	-15	-9	-30	-25	-7	-13	-6	-21	-17	0	-4	-16	-27	-14	-18
6 Y	1	-17	-24	-23	-30	-13	-9	-29	-27	-7	-15	-8	-20	-16	3	-6	-10	-26	-15	-19
7 P	-1	-14	-19	-16	-31	-12	-8	-25	-27	-8	-14	-3	-14	-16	-2	-5	-9	-26	-15	-18
8 L	4	-14	-18	-11	-31	-7	-4	-21	-24	-17	-15	-7	-13	-25	5	-4	-9	-27	-17	-20
9 A	-1	-10	-15	-5	-33	-2	4	-22	-22	-21	-13	1	-10	-28	0	-4	-8	-30	-27	-23
10 D	0	-16	-22	-7	-33	-5	3	-20	-24	-23	-6	2	-3	-34	0	-5	-7	-39	-34	-21
11 A	4	-11	-15	-8	-27	0	5	-19	-20	-21	-8	9	-1	-33	-8	-2	1	-39	-33	-19
12 H	6	-5	-10	-5	-32	3	3	-14	-17	-26	-16	16	-6	-37	-7	-4	4	-40	-34	-22
13 L	1	3	-9	-6	-34	5	5	-16	-16	-27	-17	25	-5	-39	-9	-4	-2	-41	-34	-24
14 T	3	0	-15	-13	-25	2	2	-20	-18	-19	-9	20	-7	-36	-14	-8	-2	-40	-33	-19
15 K	-1	-2	-17	-18	-25	1	-2	-23	-16	-16	-1	16	-7	-31	-21	-10	-1	-31	-24	-17
16 K	-1	-3	-16	-16	-24	1	-3	-23	-19	-16	-4	17	-11	-31	-20	-9	0	-31	-22	-16
17 L	1	-1	-16	-15	-19	0	-1	-24	-17	-16	-4	14	-13	-29	-19	-10	0	-29	-22	-14
18 L	1	-5	-23	-15	-22	-6	-4	-23	-21	-10	1	8	-9	-24	-22	-12	-6	-28	-21	-7
19 D	0	-5	-23	-16	-22	-5	-4	-26	-21	-12	0	8	-9	-24	-20	-10	-9	-28	-21	-7
20 L	-1	-8	-21	-16	-22	-3	-4	-26	-19	-9	1	7	-10	-24	-20	-9	-8	-28	-21	-5
21 V	4	-7	-20	-17	-27	0	-4	-24	-10	-14	-5	10	-11	-26	-16	-5	-7	-28	-19	-8
22 Q	7	-4	-18	-17	-21	-2	-5	-23	-14	-12	-9	13	-12	-27	-15	0	-6	-32	-21	-5
23 Q	7	-2	-16	-20	-20	-3	-6	-21	-9	-8	-6	11	-10	-26	-16	-1	-5	-31	-22	-2

20 X 20 → L X 20

Reference

- Altschul, Basic Local Alignment Tool, *J. Mol. Biol*., 1990
- Altschul, Gapped BLAST and PSI-BLAST,*J. Mol. Biol*.,1997
- Jonathan Pevsner, Bioinformatics and Functional Genomics, 2006
- Slides from Gao G