BLAST介绍

BLAST的前世今生

- BLAST 全称为basic local alignment search tool，是一个局部比对的搜索工具。

在SF ALTSCHUL于1990年发表的文章中我们可以读到，第一代的BLAST是一个不含gap思想的搜索算法，而同一作者1997年发表的文章中提出来重大的改进，得到现在我们常用的gapped blast和PSI—blast。

基本概念以及数学准备知识

- MSP score： maximal segment pair (MSP) score.
- HSPs: high-scoring pair(s).
- Seed: 种子，指精确匹配后得到的高分HSP，以此为核心，向两边（左右）延伸以便得到高分段。
- Hit：启发点。注意one-hit以及two-hit的概念，前者是前世blast所用，后者为今生blast所用。

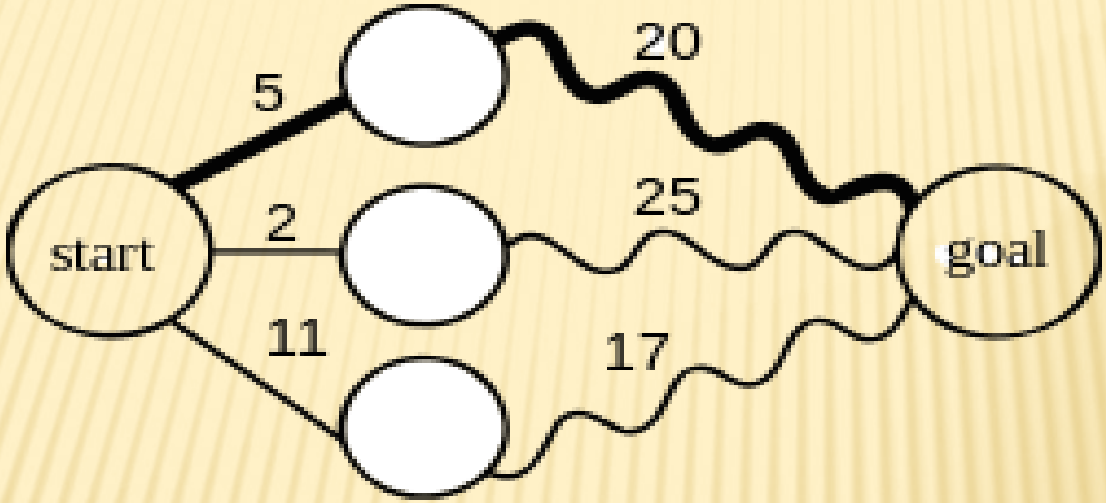
- Substitution matrix：代换矩阵，是这个矩阵的元素用于代换给出的线性方程组的未知元。数学上一般用back substitution来称呼这一操作。
- inverse 称为矩阵的逆
- Permutation 为转置矩阵，指交换指定矩阵的行与列。

数学知识ABC

- Introduction：
- BLAST与needle的区别，BLAST首先建立了一系列的严格统计学概念。从BLAST的应用可以看到，给出一个query，然后指定了搜索的库之后，将得到按照匹配度的大小排列显示的结果序列。所以，当给出了一个有效的算法，需要统计学给出的预期值（阈值），指导搜索的范围和何时该停止搜索一个序列。并考虑到内存和运算时间，协调搜索的步骤数和得到的结果的精确性等原因。

动态规划算法

- BLAST应用了动态规划的基本思想，引入启发式算法的思想，节省了时间。启发式的动态规划算法在精确性上不如纯粹动态规划（如needle算法）但在速度上却是其的50倍左右。这一点保证了BLAST比对大规模数据库的可行性。
- 动态规划简单的讲就是如果一个问题能够有最优化的子结构，那么它就能很好的被递归的方法解决。



如图，从start至goal，一共是三个路线，显然的是，粗线所示的是最佳路径。这里反映了动态规划的一个基本想法，那就是，先将一个问题分解为若干子问题，然后对所有子问题进行计算，最后得出最佳的子问题的一个线性组合。

- 刚才我们讲到了子问题，那么什么样的子问题是可以考虑的，或者说什么是有效的子问题呢？这就牵涉到动态规划的基本操作原则，递归。也就是说那些能产生单向递归关系的子问题才是好问题。

- 举个反例：解决一个K—K（n）的问题，我们分解为k（0），k（1），k（2）.....k（n-1).如果说在这个子问题串中出现了k（x）推出k（y）再推出k（z），但是此时k（z)却为了得到下一个递归式，用到了k（x），这样就说明这个地方出现了环路，一旦出现这种环路，动态规划就不好用了，专业术语为出现了后效性。简言之，动态规划中严格遵守后续状态的推导与前一个状态的得到无关。

- 由于确定一个子问题是否是最优化的，是核心问题，但是表达起来过于复杂，我们在这里假定，所有的讨论都基于子问题是最优化的子结构。动态规划算法利用了子问题的重叠性来缩短计算时间，也就是说，在得到一个子问题的最优解，进行下一个子问题最优解的计算时，这个原来的子问题将会被用到，如果我们把它保存在一个表格里，我们就可以直接来应用。一般来说，计算时间随着加入的子问题的增多而呈现多项式时间（步骤）的增加，所以动态规划的子问题重叠性能保证其计算在一个多项式时间内，从而避免出现NP—hardness。一旦递归式构造成功了，我们需要采用回溯的办法来得到最优解的结构。

用NEEDLE算法做例子说明

	A	G	C	T
A	10	−1	−3	−4
G	−1	7	−5	−3
C	−3	−5	9	0
T	−4	−3	0	8

```
AGACTAGTTAC
CGA———GACGT
```

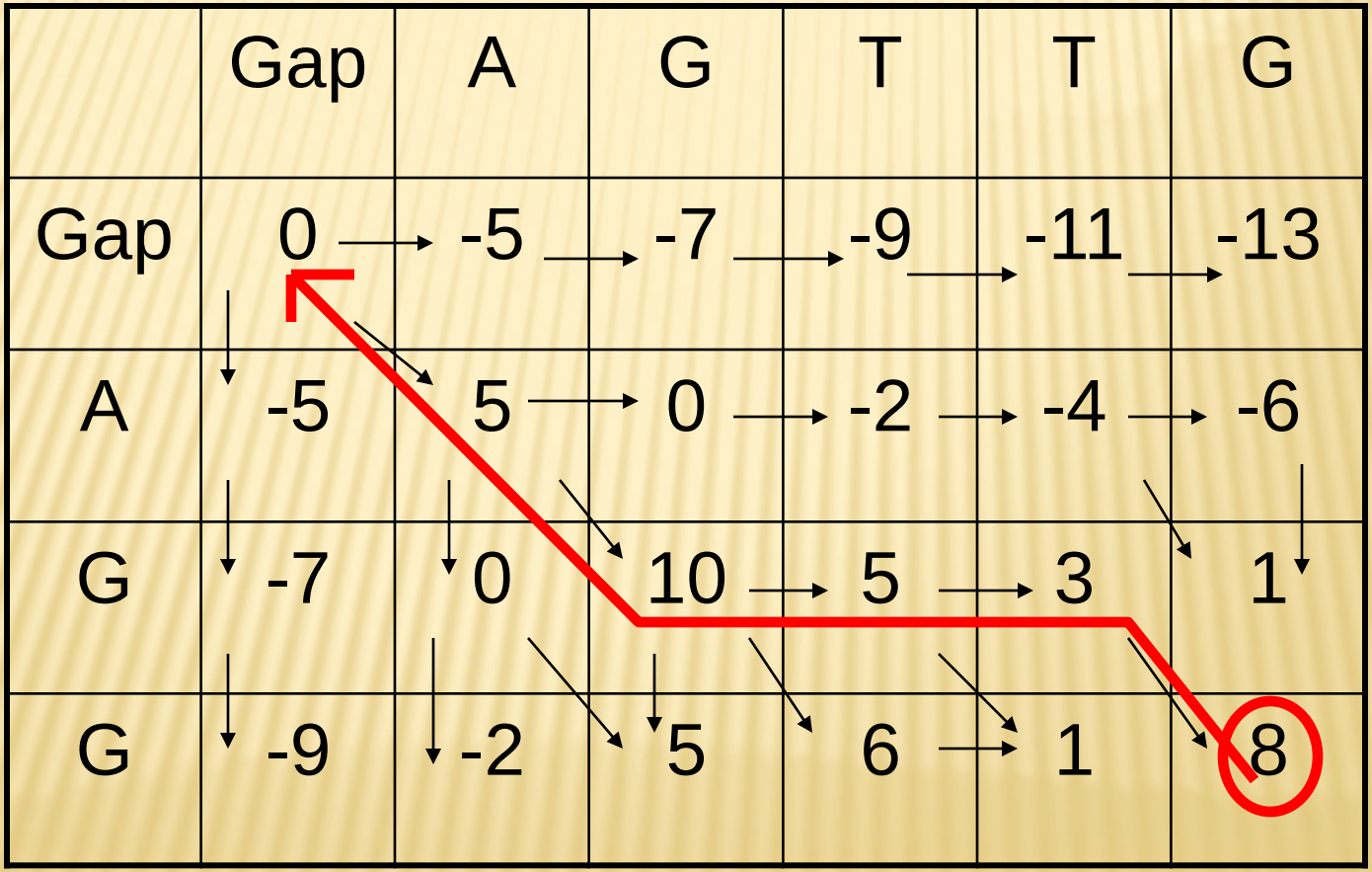
$$S(A,C)+S(G,G)+S(A,A)+(3\times d)+S(G,G)+S(T,A)+S(T,C)+S(A,G)+S(C,T)$$
$$= -3+7+10-(3\times 5)+7+-4+0+-1+0=1$$

- 上面的图标给出了一个相似度矩阵，一个序列（已经插空），以及一个算式。
- 插空是核心步骤之一的子问题最优化，假设这个插空已经完美的使得求解整个序列得分最优化，那么我们要做的就是在列举了所有得分的矩阵中找到相应的得分，然后再相加，这就是回溯。
- 如果把这个线性图改为矩阵的形式，便是熟悉的形式了。下面举一个写成矩阵的形式。

例

序列1：AGTTG

序列2：AG__G



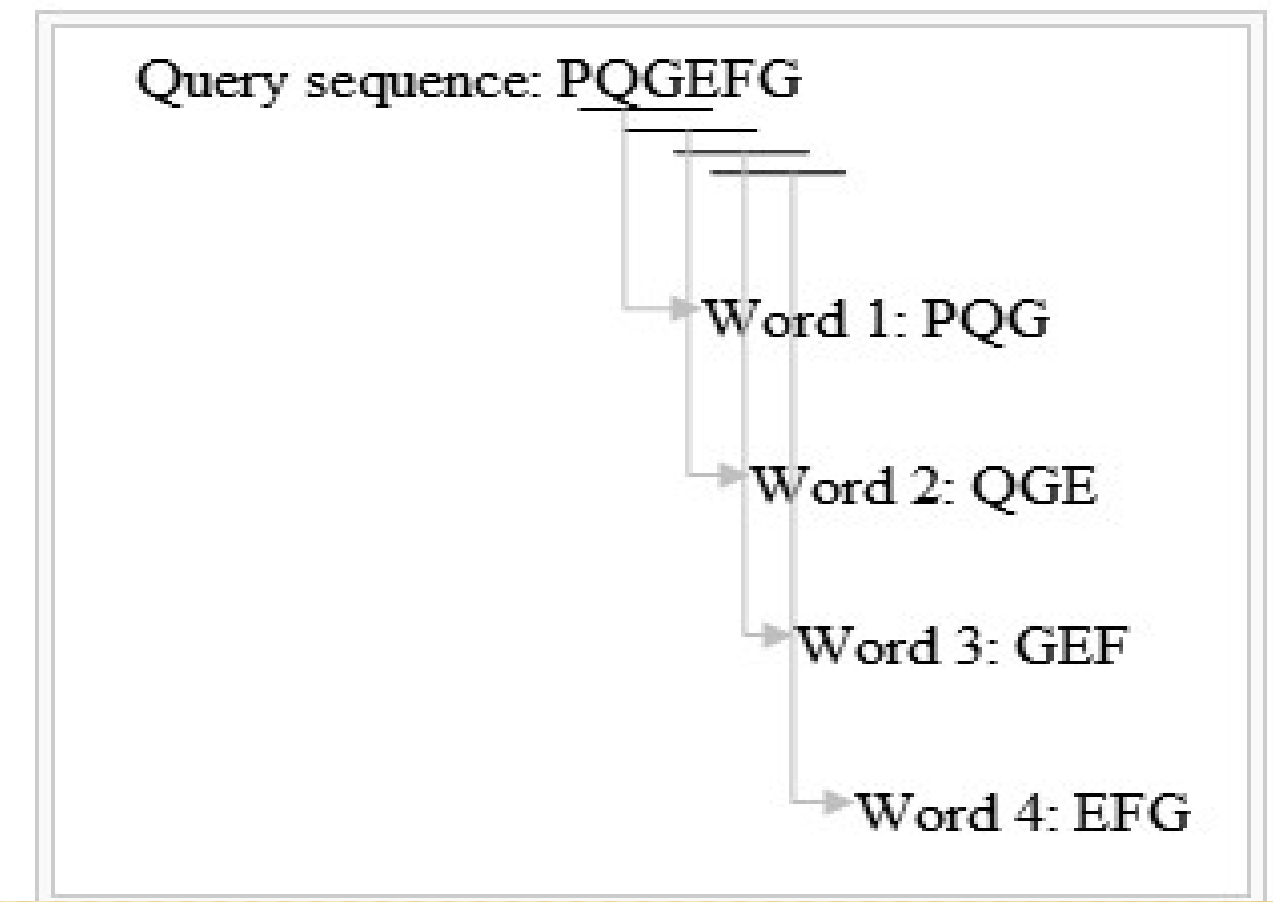
- 我们把序列1写为row，将序列2写为column。最关键的是在首row和首column处添加了2个gap，这是非常漂亮的做法，这样其实就涵盖了所有的针对这两条序列的插空，且明显可推广。
- 从第一个0开始，所有的小箭头都最终指向一个末尾，也就是说这就是所有的子问题构成的一个路径树。当所有的路径都被找到了，那么怎么在所有递归的子问题里找到我们的最终唯一的路径呢?
- 计算机通过做多次上面列举的求和之后，可以得到，沿着红线的方向就能得到最优解。于是，我们就得到了回溯的最优解。
- 这里引入了两个重要的概念，top-down以及bottom-up。前者求出所有子问题递归形式，后者求出唯一最优化的路径。

- Blast的算法

- BLAST其实是一个动态规划的方法，关键在于做了近似，便称之为启发式算法。为近何似？就是说，针对一个字符串，搜索了数据库给出的一个随机序列的一个区域，这个区域里相同（也许也包括相似的，只是赋权不同）的字符数通过计算和我预先给定的理论期待值不冲突（大于或等于）那么我就宣称找到了一个匹配。
- 那么这是什么样的阀值，如何建立这个阀值，具体如何操作呢？我们需要引入统计学的知识。

- 我们理清思路，也就是说，我们先设计一系列合理的步骤，能最终高效的得到局部比对结果。
- 1.析出数据库中的看起来毫无希望的序列。这里指的是那些低复杂度，或者说明显具有重复BP的序列。这样做有利于节省运算时间。我们有诸如SEG,DUST等软件可以进行操作。

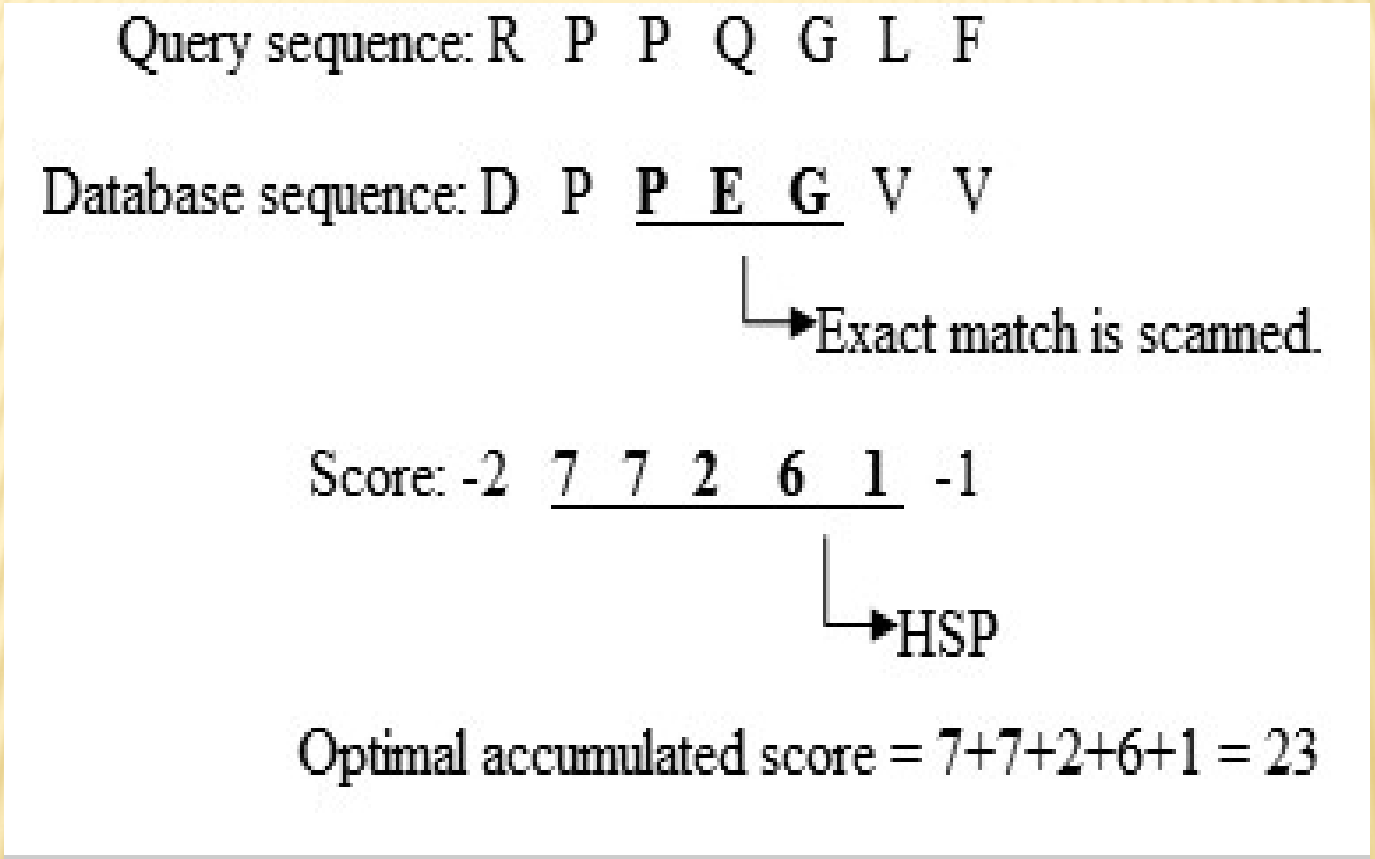
- 2.引入w-letters以及threshold-value的概念。
- 前者指在我们给出的query中选取具有w长度的字符串，比如一个残基序列ACGEKM，我们选取ACG，这时w=3.
- 后者缩写为T-value，这个值是说，在选取w长度后，query的w-letter按照substitution matrix给出的分数进行打分后，把低于T值的w字符串舍弃掉。


Query sequence: PQGEFG
Word 1: PQG
Word 2: QGE
Word 3: GEF
Word 4: EFG

- 3. 将所有的w-letters的可能性全部列举出来以后，就要对得到结果进行考察。根据substitution matrix进行计分，在前述的T值的参与下，进一步筛选出我们需要的高分字串。
- 4.将T值允许的比对片段保留下来，并构建为一个搜索树，从而能极大地提高效率。
- 5.如上图所示我们看到当w=3时，有四个字符串（默认都已经通过了T值的考察），那么我们要做的就是将每个字符串都确保进行一次3-4步骤。

- 6. 对那些筛选出来的高分字符串拿到数据库中去和包含其中的随机序列去进行覆盖比对。这个时候只要w-letters里的一个字符串与目标序列的一个区域发生了一个准确配对，那么就称bingo（hit），并以此为启发点（或者种子，seed）进行下一步工作。

- 7.当得到种子以后，我们就会以这个种子为中心向两边开始延伸，延伸的终止点就是这个序列的累积得分开始变小，也就是说出现了负数的匹配得分。


Query sequence: R P P Q G L F
Database sequence: D P P E G V V
Exact match is scanned.
Score: -2 7 7 2 6 1 -1
HSP
Optimal accumulated score = 7+7+2+6+1 = 23

- 8.必须首先引入一个以经验为主的值S，这个值称之为切断分（cutoff score），顾名思义，就是一个给HSPs的阀值，凡低于这个S的HSPs将无法进入我们的下一步的工作。相反，我们将所有及格的HSPs列举出来。

- 9.当我们得到了这些需要的HSPs片段后，并不能直接将这些所谓的HSPs所在的序列就这么输出给程序的使用者，因为太多了，所以必须进一步筛选那些含有HSPs的序列。为此我们需要引入耿贝尔极值分布的概念（gumbel extreme distribution）：

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

- 这个极值分布式给出了一个所谓的概率p，指cutoff score 不小于x这个预期值的概率。应用这个p值（因为着重是为了引出下面E值的概念，所以并不对这个艰难的数学知识做过多的解释），我们可以引入expect value—E。它的数学表述如下：

$$E \approx 1 - e^{-p(s>x)D}$$

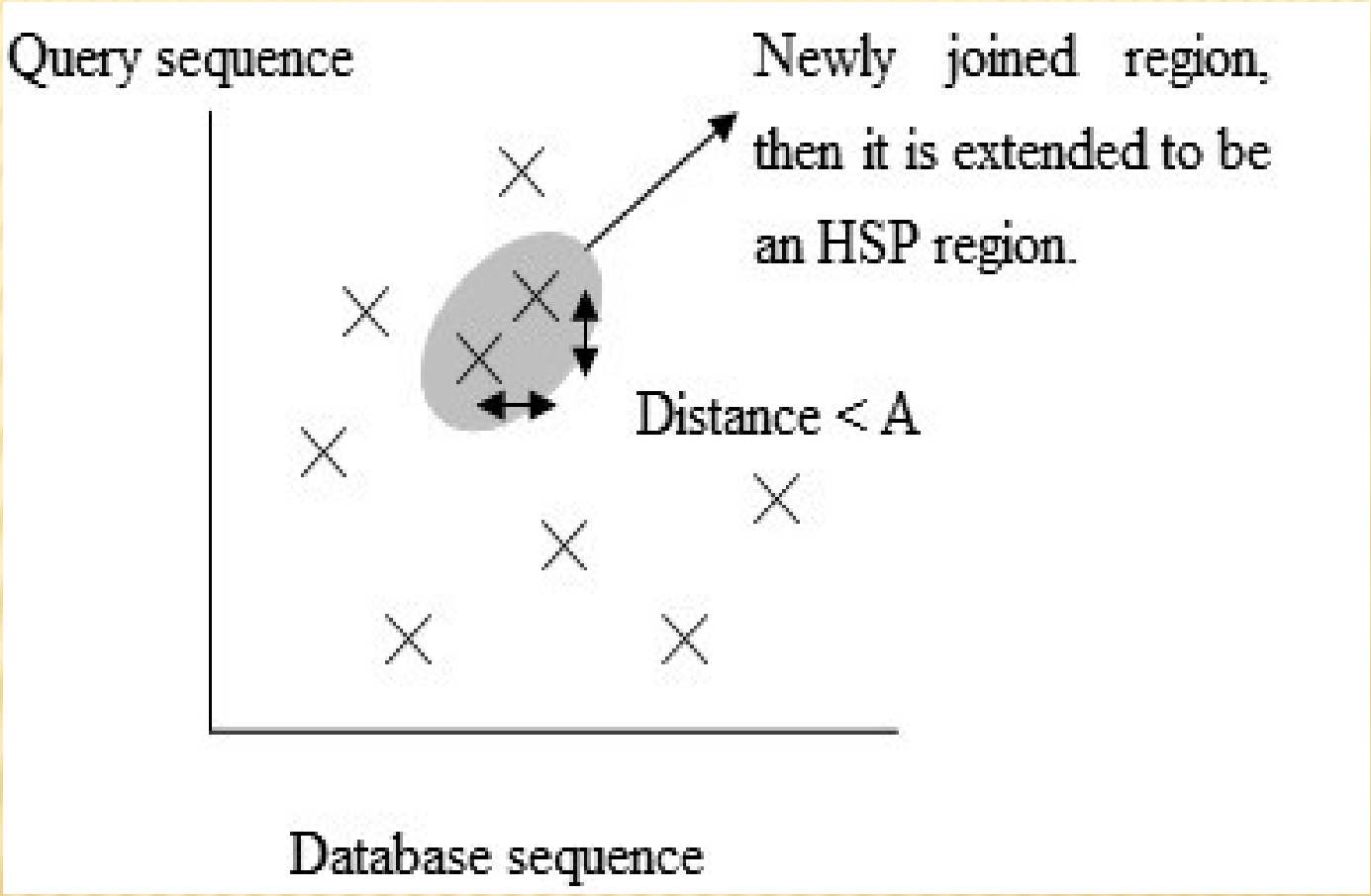
$$E \approx pD$$

- 其中，D指的是所搜索的数据库中有多少个序列。后者是前者的一个简化，在p值小于0.1时认为p足够小到服从泊松分布，从而得到最简式。E值的出现，让我们输出的结果更令人满意，因为设定了E值将对所有的原始HSPs进行一个筛选，只有那些小于或等于E值的HSPs才会输出给我们进行研究。
- $$P(X = k) = \frac{e^{-\lambda}\lambda^{k}}{k!}$$
- 至此，我们已经将第一代blast的原理讲的比较清楚了。

- BLAST2的介绍

- BLAST2是第一代开发者时过7年后开发的新一代BLAST，这个BLAST引入了gap 的概念。从第一代单启发点增加到了两个启发点，同时设置了引发gap的阀值，这在现如今的blast软件里分别成为打开一个gap的罚分和延展一个gap 的罚分设置项。

- ABCDE　　ABCDE
- ACD ...　　A_CD_
- 左边是第一代单启发点blast，不允许gap的存在，只能是ACD作为HSP去延展。右边是第二代blast，采用了双启发点，允许了gap作为HSP的一部分去进行延展。当双启发点进行延展后就会连成一个所谓的HSP区域。


Query sequence
Newly joined region, then it is extended to be an HSP region.
Distance < A
Database sequence

- 参考文献：

- 1. Basic local alignment search tool
- 2.Applications and statistics for multiple high-scoring segments in molecular sequences
- 3.Method for assessing the statical significance of molecular sequence features by using general scoring schemes
- 4.Gapped BLAST and PSI-BLAST: a new generation of protein database search programs