一．碱基比对有三种情况：

- str1：SS_
- str2：T_T

-str1 的字符和 str2 的字符相互配对
-str1 的字符和 str2 的 gap 相互配对
-str1 的 gap 和 str2 的字符相互配对

二．比对原则：

str1 和 str2 根据*打分矩阵*进行相关匹配。

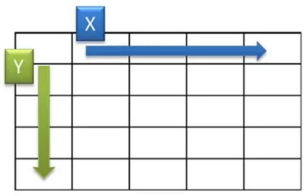
最佳匹配分数 = Σ 单独残基比分

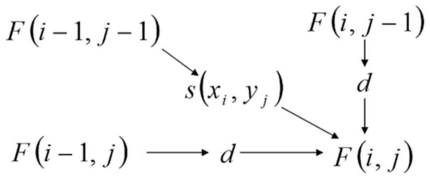
最好比分 = 之前最好 + 当前最好（*动态规划）
***动态规划**：局部最优解组合即为全局最优解

三．定义与 Needleman_Wunch 公式：

F(i,j) = 第一序列 X 从 1 到 i
} 的最好的比对分数
第二序列 Y 从 1 到 j

F(i,j)= max of
F(i-1,j-1) + s(X_i,Y_j) 【str1 的字符和 str2 的字符相互配对】
F(i-1,j) + d 【str1X 比对到 Gap】
F(i,j-1) + d 【str2Y 比对到 Gap】
F(0,0)=0
*Gap 分为 open gap 和 extending gap，这里中先都用 d 代替





例：

（）（）	A	A	G
（）	0	-5	-10	-15
A	-5			
G	-10			
C	-15			

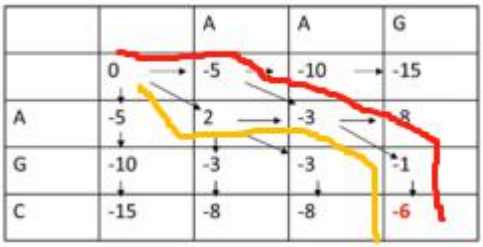
【打分矩阵】

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

通过打分矩阵结合 Needleman_Wunsch 公式进行计算。

根据结果-6 进行反推可以了解到有两条通路：

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



所以确定最佳比对格式，都是**最佳**。

红：　　黄：

AAG_　　AAG_

_AGC　　A_GC

四．过度

全局比对要死的特点是无法去掉内含子

序列比对的大部分目的是观察外显子，所以出现了局部比对的需求。

Smith_waterman 就是经典的局部比对算法。这是对 Needleman_Wunsch 的改进。

五．Smith_Waterman 算法

F(i,j)= max of

F(i-1,j-1) + s(X_i,Y_j)【str1 的字符和 str2 的字符相互配对】

F(i-1,j) + d　　【str1X 比对到 Gap】

F(i,j-1) + d　　【str2Y 比对到 Gap】

0【0 的作用实际是止损，一旦差异过大，0 就来止损了】

F(0,0)=0

例：

（）	（）	A	A	G
（）	0	0	0	0
A	0			
G	0			
C	0			

【打分矩阵】

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

通过打分矩阵结合 Smith_Waterman 公式进行计算

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

结果：

AG　　A

AG　　A

六 再次改进

由于 gap 分为 open gap 和 extending gap。因此引入*有限自动机*进行详解

有限自动机的三个状态：
M：彼此对上，但不一定相同
X：X 的残基对上了空位
Y：Y 的残基对上了空位

自动机：
M(i,j)=max of

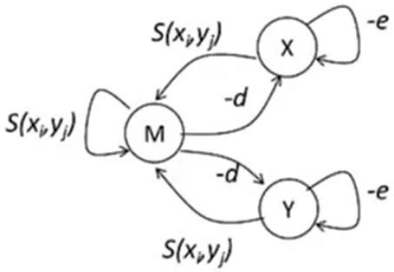
$M(i-1,j-1) + S(X_i,Y_j)$ after a match
$X(i-1,j-1) + S(X_i,Y_j)$ after a gap
$Y(i-1,j-1) + S(X_i,Y_j)$ after a gap

X_i aligned to a gap X(i,j) = max of

M(i-1,j) - d
M(i-1,j) - e

Y_j aligned to a gap Y(i,j) = max of

M(i,j-1) - d
M(i,j-1) - e



d：open gap 罚分
e：extend gap 罚分