

# 第3章：序列比对(Alignment)

- 比对的数学问题
- Pairwise 比对
- Multiple 比对

部分Slides修改自Indiana University 汤海旭 (Haixu Tang) 2007年课程 “Biological sequence analysis” 课件。

# 比对的数学模型

- 设有序列:  $X=(x_1, x_2, \dots, x_m)$  与  $Y=(y_1, y_2, \dots, y_n)$ , 长度分别为  $m$  和  $n$ . 现在要找出最优的全局比对方案, 即在  $X$  与  $Y$  中分别加入若干插入, 使得符合得最好。
- 问题: 好的度量(比对好的标准)与表达式;  
好的算法。

$X=ATGTTAT, \quad Y=ATCGTAC$

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

# Scoring a Pairwise Alignment

- Mismatches are penalized by  $-\mu$ , indels are penalized by  $-\sigma$ , and matches are rewarded with  $+1$ , the resulting score is:

$$\#matches - \mu(\#mismatches) - \sigma(\#indels)$$

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

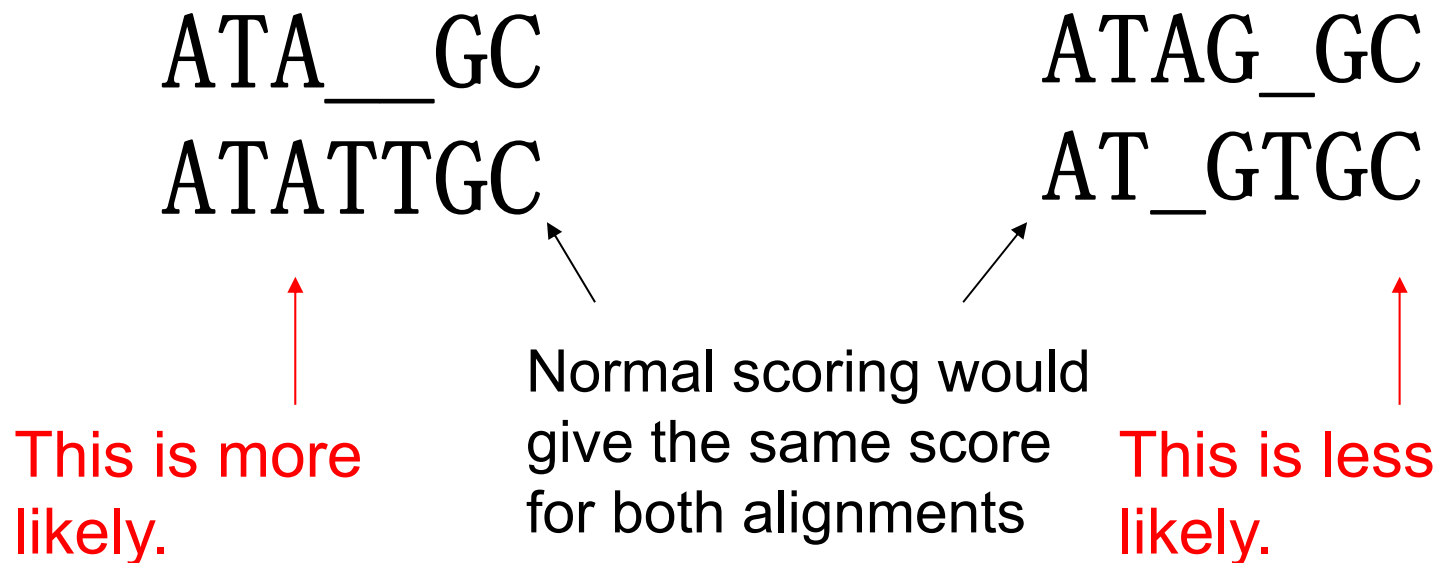
$$5 - \mu - 2\sigma$$

# Scoring Matrices

- Amino acid substitution matrices
  - PAM
  - BLOSUM
- DNA substitution matrices
  - DNA is less conserved than protein sequences
  - Less effective to compare coding regions at nucleotide level

# Affine Gap Penalties

- In nature, a series of  $k$  indels often come as a single event rather than a series of  $k$  single nucleotide events:



# Accounting for Gaps

- *Gaps*- contiguous sequence of spaces in one of the rows

- Score for a gap of length  $x$  is:

$$-[\rho + \sigma(x-1)]$$

where  $\rho > 0$  is the penalty for introducing a gap:

gap opening penalty

$\rho$  will be large relative to  $\sigma$ :

gap extension penalty

because you do not want to add too much of a penalty for extending the gap.

# Part I

## Pairwise Alignment

# Global Alignment

## Needleman-Wunsch 算法

- 设有序列:  $X=(x_1, x_2, \dots, x_m)$  与  $Y=(y_1, y_2, \dots, y_n)$ , 长度分别为  $m$  和  $n$ , 现要找出最优的全局比对方案, 使得对在给定的打分函数下得到最高分。
- 例:  $X=\text{“ACGAA”}$   $Y=\text{“AACAGAC”}$   
打分策略为每个正确匹配加 1 分  
每个配错位置减 1 分, 每个插入位置减 1 分。



# 递推变量

- 设矩阵  $F=(F(i,j))$  为  $X, Y$  的子序列  $X_{1,i}$  和  $Y_{1,j}$  的最佳比对得分, 其中  $X_{1,i}=(x_1, x_2, \dots, x_i)$ ,  $Y_{1,j}=(y_1, y_2, \dots, y_j)$ , 在我们的例子里  $F(2,3)$  就是

$$X_{1,i} = \text{“AC”}, Y_{1,j} = \text{“AAC”}$$

的最佳比对得分。

- 我们可以用递推的方法把  $F$  求出来, 而  $F(m,n)$  和与  $F(m,n)$  相对应的比对方案正是我们想要的结果。

# 递推算法

- 我们可以根据  $F(i-1, j)$ ,  $F(i-1, j-1)$ ,  $F(i, j-1)$  算出  $F(i, j)$ 。因为到达  $(i, j)$  的路径必经  $(i-1, j-1)$ ,  $(i, j-1)$ ,  $(i-1, j)$  三者之一，因而到达  $(i, j)$  的最佳路径必为到达它们三者之一的最佳路径再加上最后一步的分值中最优的路径。

$$F(i, j) = \max \left\{ \begin{array}{l} F(i-1, j) - d \\ F(i, j-1) - d \\ F(i-1, j-1) + d_{ij} \end{array} \right\}$$

$$d_{ij} = \begin{cases} 1 & \text{if } x_i = y_j \\ -1 & \text{if } x_i \neq y_j \end{cases}$$

# 递推演示

	Y	A	A	C	A	G	A	C
X	0	-1	-2	-3	-4	-5	-6	-7
A	-1							
C	-2							
G	-3							
A	-4							
A	-5							

# 递推演示

	Y	A	A	C	A	G	A	C
X	0	← -1	← -2	← -3	← -4	← -5	← -6	← -7
A	-1	1						
C	-2							
G	-3							
A	-4							
A	-5							

# 递推演示

	Y	A	A	C	A	G	A	C
X	0	-1	-2	-3	-4	-5	-6	-7
A	-1	1	0					
C	-2	0						
G	-3							
A	-4							
A	-5							

The diagram illustrates a recursive calculation process. The grid shows values for different combinations of X and Y. The 'X' row contains values from 0 to -7. The 'A' row contains values from -1 to 0. The 'C' row contains values from -2 to 0. The 'G' row contains values from -3 to -5. The 'A' row contains values from -4 to -5. Arrows indicate the flow of values from the 'X' row to the 'A' row, and from the 'A' row to the 'C' row, and so on, illustrating the iterative nature of the calculation.

# 递推演示

	Y	A	A	C	A	G	A	C
X	0	-1	-2	-3	-4	-5	-6	-7
A	-1	1	0	-1				
C	-2	0	0					
G	-3	-1						
A	-4							
A	-5							

# 递推演示

	Y	A	A	C	A	G	A	C
X	0 ←	-1 ←	-2 ←	-3 ←	-4 ←	-5 ←	-6 ←	-7
A	-1 ↑	1 ←	0 ←	-1 ←	-2			
C	-2 ↑	0 ↑	0 ↑	1				
G	-3 ↑	-1 ↑	-1 ↑					
A	-4 ↑	-2						
A	-5							

# 递推演示

	Y	A	A	C	A	G	A	C
X	0	-1	-2	-3	-4	-5	-6	-7
A	-1	1	0	-1	-2	-3	-4	-5
C	-2	0	0	1	0	-1	-2	-3
G	-3	-1	-1	0	0	1	0	-1
A	-4	-2	0	-1	1	0	2	1
A	-5	-3	-1	-1	0	0	1	1



	Y	A	A	C	A	G	A	C
X	0	-1	-2	-3	-4	-5	-6	-7
A	-1	1	0	-1	-2	-3	-4	-5
C	-2	0	0	1	0	-1	-2	-3
G	-3	-1	-1	0	0	1	0	-1
A	-4	-2	0	-1	1	0	2	1
A	-5	-3	-1	-1	0	0	1	1

<b>X</b>	--	A	C	--	G	A	A
<b>Y</b>	A	A	C	A	G	A	C

<b>X</b>	A	--	C	--	G	A	A
<b>Y</b>	A	A	C	A	G	A	C

# 仿射罚分下的比对

- 罚分函数:  $\gamma(g) = -d - (g - 1)e, d > e$
- 仍然可以采用动态规划算法，其核心是：  
只要记住最后的位置是否有过空格，就可以决定如何打分

# 仿射罚分下的比对

- 迭代算法

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

$$I_x(i, j) = \max \begin{cases} M(i-1, j) - d \\ I_x(i-1, j) - e \end{cases}$$

$$I_y(i, j) = \max \begin{cases} M(i, j-1) - d \\ I_y(i, j-1) - e \end{cases}$$

# 局部比对(Local Alignment)

- 两个序列的局部比对，是指在两个序列中分别找出一段子序列，使得它们的全局比对符合度最高。
- 局部比对的算法与全局比对有两点不同：  
局部比对比全局比对又增加了两个选择参数:序列的位置和长度,因而
  - $F(i,j)$ 的初始化和递推公式不同;
  - 回溯开始位置不同。

# Smith-Waterman 算法

- 初始化:  $F(i, 0)$  和  $F(0, j)$  均为零
- 递推公式:

$$F(i, j) = \max \left\{ \begin{array}{l} 0 \\ F(i-1, j) - d \\ F(i, j-1) - d \\ F(i-1, j-1) + d_{ij} \end{array} \right\}$$

$$d_{ij} = \begin{cases} 1 & \text{if } x_i = y_j \\ -1 & \text{if } x_i \neq y_j \end{cases}$$

# Smith-Waterman 算法 (II)

- 四者之间的最大者，多了一个取零的可能。取零意味着从当前位置开始一个新的比对，也就是说若比对到某处分数为负，那就索性从头开始新的比对。

<b>X</b>	--	A	C	--	G	A	A
<b>Y</b>	A	A	C	A	G	A	C

# 回溯

	Y	A	A	C	A	G	A	C
X	0	0	0	0	0	0	0	0
A	0	1	1	0	1	0	1	0
C	0	0	0	2	1	0	0	2
G	0	0	0	1	1	2	1	1
A	0	1	1	0	2	1	<b>3</b>	2
A	0	1	2	1	1	1	2	2

- 回溯时,由于局部比对可能到任何位置结束, 我们从整个矩阵中找到最大值, 从整个矩阵中最大值处开始回溯到0为止。

于是得到最佳局部比对

<b>X</b>	A	A	C	A	G	A	C
<b>Y</b>		A	C	---	G	A	A



# Representing Sequence Alignment Using Pair HMM

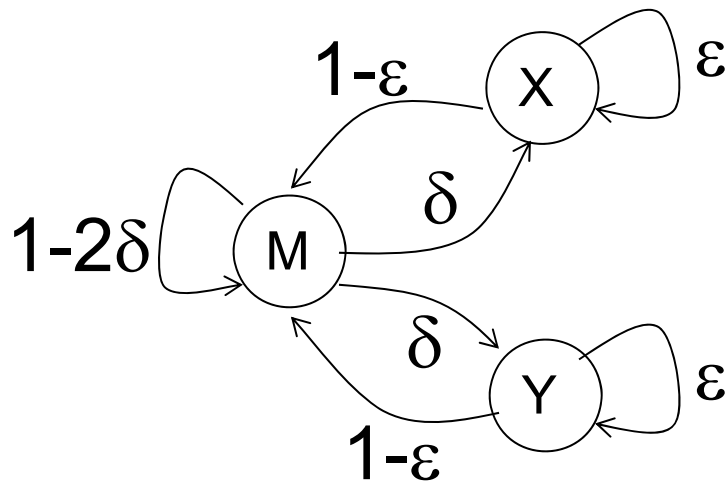
## “Hidden” States

- Match (M)
- Insertion in  $x$  (X)
- insertion in  $y$  (Y)

## Observation Symbols

- Match (M):  $\{(a,b) \mid a,b \text{ in } \Sigma \}$ .
- Insertion in  $x$  (X):  $\{(a,-) \mid a \text{ in } \Sigma \}$ .
- Insertion in  $y$  (Y):  $\{(-,a) \mid a \text{ in } \Sigma \}$ .

# Representing Sequence Alignment Using Pair HMM



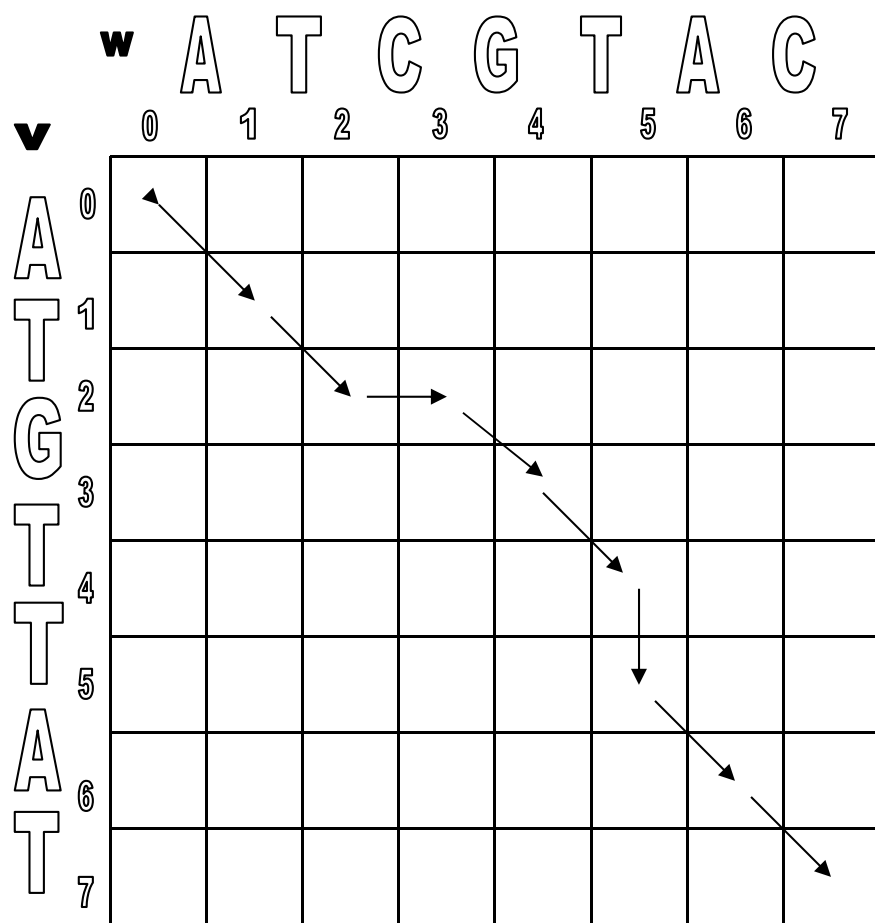
Emission probabilities:

M:  $P_{x_i, y_j}$

X:  $q_{x_i}$

Y:  $q_{y_j}$

# Alignment



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

M M Y M M X M M

Path= hidden state sequence

# Sequence Alignment Using Pair HMM

- Based on the HMM, each alignment of two DNA/protein sequences can be assigned with a probability score;
- Each “observation symbol” of the HMM is an aligned pair of two letters, or of a letter and a gap.
- The Markov chain of hidden states should represent a scoring scheme reflecting an evolutionary model.
- Transition and emission probabilities define the probability of each aligned pair of sequences.
- Given two input sequences, we look for an alignment of these two sequences of maximum probability.

# Transitions and Emission Probabilities

## Transitions probabilities

(note the forbidden ones).

◆  $\delta$  = probability for 1<sup>st</sup> gap

◆  $\varepsilon$  = probability for extending gap.

	M	X	Y
M	$1-2\delta$	$\delta$	$\delta$
X	$1-\varepsilon$	$\varepsilon$	0
Y	$1-\varepsilon$	0	$\varepsilon$

## Emission Probabilities

- Match:  $(a,b)$  with  $p_{ab}$  – only from M states
- Insertion in  $x$ :  $(a,-)$  with  $q_a$  – only from X state
- Insertion in  $y$ :  $(-,a)$ .with  $q_a$  - only from Y state.

# Scoring Alignments

- For each pair of sequences  $x$  (of length  $m$ ) and  $y$  (of length  $n$ ), there are many alignments of  $x$  and  $y$ , each corresponds to a different state sequence (with the length between  $\max\{m, n\}$  and  $m+n$ ).
- Given the transmission and emission probabilities, each alignment has a defined score – the product of the corresponding probabilities.
- An alignment is “most probable”, if it maximizes this score.

# Finding the Most Probable Alignment-Viterbi Algorithm

Let  $v^M(i,j)$  be the probability of the most probable alignment of  $x(1..i)$  and  $y(1..j)$ , which ends with a match (state M). Similarly,  $v^X(i,j)$  and  $v^Y(i,j)$ , the probabilities of the most probable alignment of  $x(1..i)$  and  $y(1..j)$ , which ends with states X or Y, respectively.

$$v^M[i, j] = p_{x_i y_j} \max \begin{pmatrix} (1 - 2\delta)v^M(i-1, j-1) \\ (1 - \varepsilon)v^X(i-1, j-1) \\ (1 - \varepsilon)v^Y(i-1, j-1) \end{pmatrix}$$

# Most Probable Alignment

Similar argument for  $v^X(i,j)$  and  $v^Y(i,j)$ , the probabilities of the most probable alignment of  $x(1..i)$  and  $y(1..j)$ , which ends with an insertion to  $x$  or  $y$ , are:

$$v^X[i, j] = q_{x_i} \max \left( \begin{array}{l} \delta v^M(i-1, j) \\ \varepsilon v^X(i-1, j) \end{array} \right)$$

$$v^Y[i, j] = q_{y_j} \max \left( \begin{array}{l} \delta v^M(i, j-1) \\ \varepsilon v^Y(i, j-1) \end{array} \right)$$



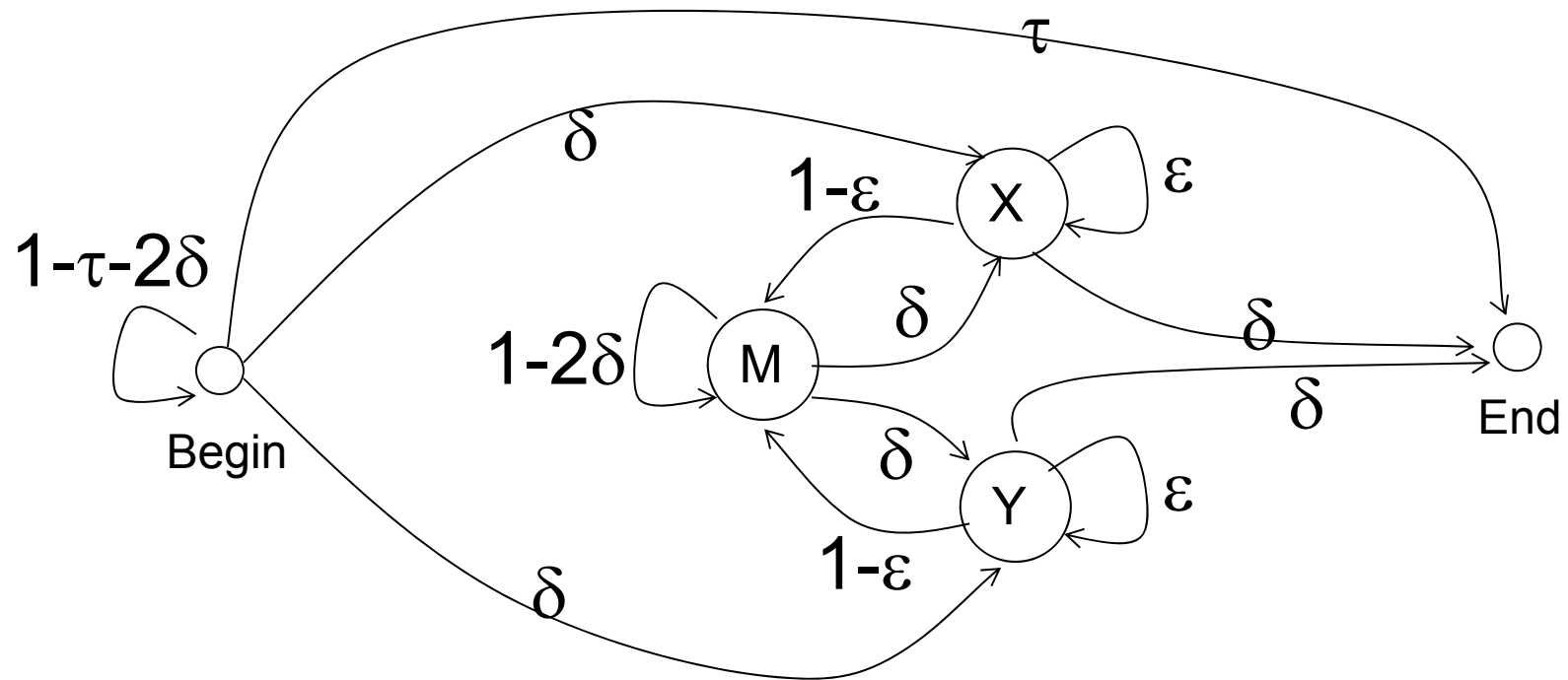
# Adding Termination Probabilities

The last transition in each alignment is to the END state, with probability  $\tau$

For this, an END state is added, with transition probability  $\tau$  from any other state to END. This assumes expected sequence length of  $1/\tau$ .

	M	X	Y	END
M	$1-2\delta - \tau$	$\delta$	$\delta$	$\tau$
X	$1-\varepsilon - \tau$	$\varepsilon$		$\tau$
Y	$1-\varepsilon - \tau$		$\varepsilon$	$\tau$
END				1

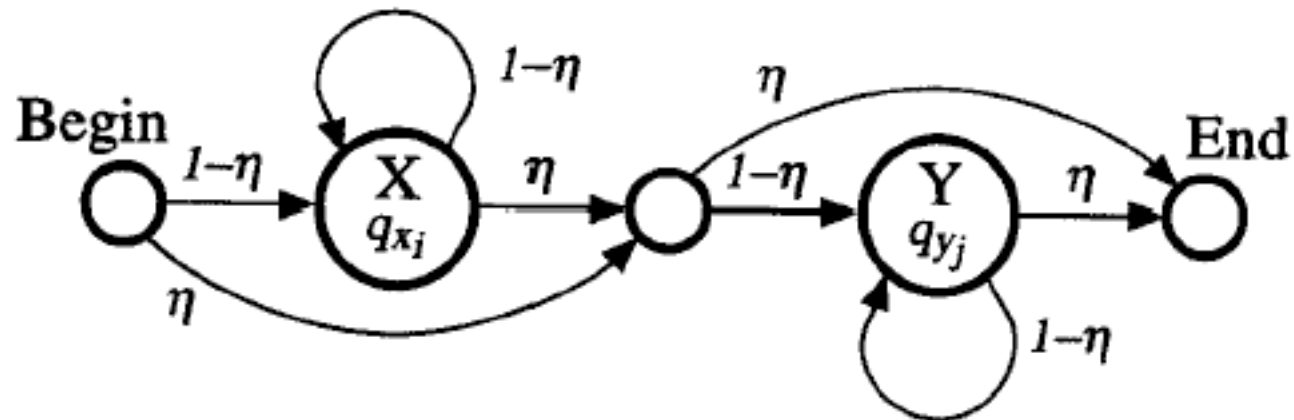
# Full Pair HMM Model



# The log-odds Scoring Function

- We wish to know if the alignment score is above or below the score of random alignment of sequences with the same length.
  - Model comparison
- We need to model random sequence alignment by HMM, with end state. This model assigns probability to each pair of sequences  $x$  and  $y$  of arbitrary lengths  $m$  and  $n$ .

# HMM for Random Sequence Alignment



# HMM for a Random Sequence Alignment

The transition probabilities for the random model, with termination probability  $\eta$ :  
( $x$  is the start state)

	$X$	$Y$	$END$
$X$	$1 - \eta$	$\eta$	0
$Y$	0	$1 - \eta$	$\eta$
$END$	0	0	1

The emission probability for  $a$  is  $q_a$ .  
Thus the probability of  $x$  (of length  $n$ )  
and  $y$  (of length  $m$ ) being random is:

$$p(x, y \mid \text{Random}) = \eta^2 (1 - \eta)^{n+m} \prod_{i=1}^n q_{x_i} \prod_{j=1}^m q_{y_j}$$

And the corresponding score is:

$$\log p(x, y \mid \text{Random}) = 2 \log \eta + (n + m) \log(1 - \eta) + \sum_{i=1}^n \log q_{x_i} + \sum_{i=1}^m \log q_{y_i}$$

# Markov Chains for “Random” and “Model”

	$M$	$X$	$Y$	$END$
$M$	$1-2\delta-\tau$	$\delta$	$\delta$	$\tau$
$X$	$1-\varepsilon-\tau$	$\varepsilon$		$\tau$
$Y$	$1-\varepsilon-\tau$		$\varepsilon$	$\tau$
$END$				$1$

“Model”

“Random”

	$X$	$Y$	$END$
$X$	$1-\eta$	$\eta$	
$Y$		$1-\eta$	$\eta$
$END$			$1$

# Combining Models in the Log-odds Scoring Function

In order to compare the  $M$  score to the  $R$  score of sequences  $x$  and  $y$ , we can find an optimal  $M$  score, and then subtract from it the  $R$  score.

This is insufficient when we look for local alignments, where the optimal substrings in the alignment are not known in advance. A better way:

1. Define a log-odds scoring function which keeps track of the difference Match-Random scores of the partial strings during the alignment.
2. At the end add to the score  $(\log \tau - 2 \log \eta)$  to compensate for the end transitions in both models.

# The Log-odds Scoring Function

$$V_M(i, j) = \log \frac{p_{x_i y_j}}{q_{x_i} q_{y_j}} + \max \begin{pmatrix} \log(1 - 2\delta - \tau) + V_M(i - 1, j - 1) \\ \log(1 - \epsilon - \tau) + V_X(i - 1, j - 1) \\ \log(1 - \epsilon - \tau) + V_Y(i - 1, j - 1) \end{pmatrix} - 2\log(1 - \eta)$$

$$V_X(i, j) = \max \begin{pmatrix} \log \delta + V_M(i - 1, j) \\ \log \epsilon + V_X(i - 1, j) \end{pmatrix} - \log(1 - \eta)$$

$$V_Y(i, j) = \max \begin{pmatrix} \log \delta + V_M(i, j - 1) \\ \log \epsilon + V_Y(i, j - 1) \end{pmatrix} - \log(1 - \eta)$$

And at the end add to the score  $(\log \tau - 2\log \eta)$ .



# 对应关系

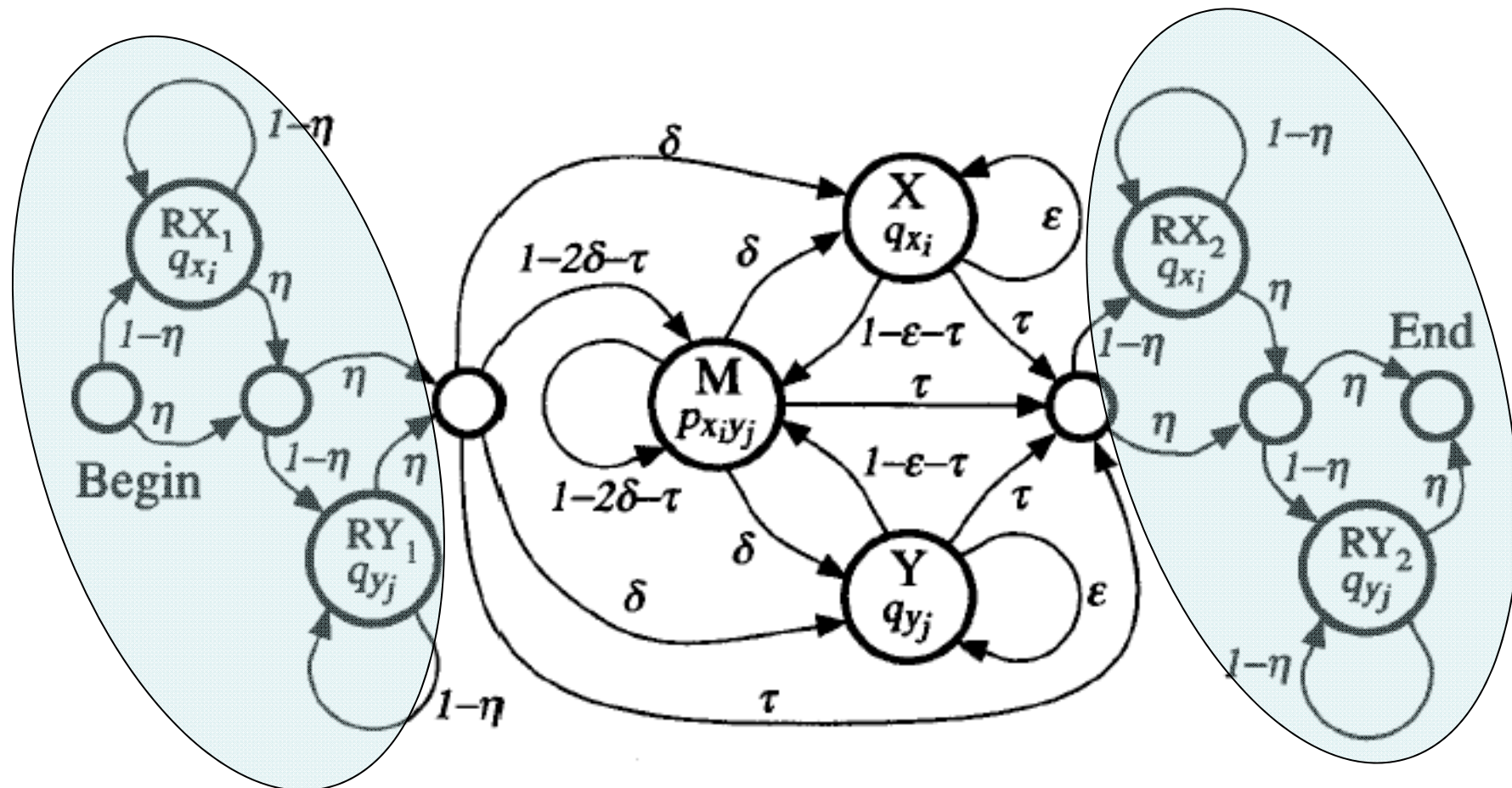
- 对比前面的仿射罚分下的动态规划算法，  
令

$$S(a, b) = \log \frac{p_{ab}}{q_a q_b} + \log \frac{(1 - 2\delta - \tau)}{(1 - \eta)^2}$$

$$d = -\log \frac{\delta(1 - \epsilon - \tau)}{(1 - \eta)(1 - 2\delta - \tau)}$$

$$e = -\log \frac{\epsilon}{1 - \tau}$$

# A Pair HMM For Local Alignment



# Full Probability Of The Two Sequences

- HMMs allow for calculating the probability that a given pair of sequences are related according to the HMM by any alignment
- This is achieved by summing over all alignments

$$P(x, y) = \sum_{\text{alignment } \pi} P(x, y, \pi)$$

# Full Probability Of The Two Sequences

- The way to calculate the sum is by using the forward algorithm
- $f^k(i,j)$  : the combined probability of all alignments up to  $(i,j)$  that end in state  $k$

# Forward Algorithm For Pair HMMs

## Initialization:

$$f^M(0, 0) = 1. f^X(0, 0) = f^Y(0, 0) = 0.$$

All  $f^*(i, -1), f^*(-1, j)$  are set to 0.


## Recursion:

$$f^M(i, j) = p_{x_i y_j} \left[ (1 - 2\delta - \tau) f^M(i - 1, j - 1) + (1 - \varepsilon - \tau)(f^X(i - 1, j - 1) + f^Y(i - 1, j - 1)) \right].$$

$$f^X(i, j) = q_{x_i} \left[ \delta f^M(i - 1, j) + \varepsilon f^X(i - 1, j) \right].$$

$$f^Y(i, j) = q_{y_j} \left[ \delta f^M(i, j - 1) + \varepsilon f^Y(i, j - 1) \right].$$

## Termination:

$P(x, y)$  

$$f^E(n, m) = \tau \left[ f^M(n, m) + f^X(n, m) + f^Y(n, m) \right].$$

# Full Probability Of The Two Sequences

- $P(x,y)$  gives the likelihood that  $x$  and  $y$  are related by some unspecified alignment, as opposed to being unrelated
- If there is an unambiguous best alignment,  $P(x,y)$  will be “dominated” by the single hidden state sequence corresponding to that alignment

# How Correct is the Alignment

- Define a posterior distribution  $P(s|x,y)$  over all alignments given a pair of sequences  $x$  and  $y$

$$P(s \mid x, y) = \frac{P(x, y, s)}{P(x, y)}$$

Probability that the optimal scoring alignment is correct:

$$P(\pi^* \mid x, y) = \frac{P(x, y, \pi^*)}{P(x, y)} = \frac{v^E(n, m)}{f^E(n, m)}$$

Viterbi algorithm  
Forward algorithm

- Usually the probability that the optimal scoring alignment is correct, is extremely small!
- Reason: there are many small variants of the best alignment that have nearly the same score.



# The Posterior Probability That Two Residues Are Aligned

- If the probability of any single complete path being entirely correct is small, can we say something about the local accuracy of an alignment?
- It is useful to be able to give a reliability measure for each part of an alignment

# The posterior probability that two residues are aligned

- The idea is:
  - calculate the probability of all the alignments that pass through a specified matched pair of residues  $(x_i, y_j)$
  - Compare this value with the full probability of all alignments of the pair of sequences
  - If the ratio is close to 1, then the match is highly reliable
  - If the ratio is close to 0, then the match is unreliable

# The Posterior probability that Two Residues are aligned

- Notation:  $x_i \diamond y_j$  denotes that  $x_i$  is aligned to  $y_j$
- We are interested in  $P(x_i \diamond y_j | x, y)$

- We have

$$P(x_i \diamond y_j | x, y) = \frac{P(x, y, x_i \diamond y_j)}{P(x, y)}$$

$$P(x, y, x_i \diamond y_j) = P(x_{1..i}, y_{1..j}, x_i \diamond y_j) P(x_{i+1..n}, y_{j+1..m} | x_i \diamond y_j)$$

- $P(x, y)$  is computed using the forward algorithm
- $P(x, y, x_i \diamond y_j)$ : the first term is computed by the forward algorithm, and the second is computed by the backward algorithm ( $= b^M(i, j)$  in the backward algorithm)

# Backward Algorithm For Pair HMMs

## Initialization:

$$b^M(n, m) = b^X(n, m) = b^Y(n, m) = \tau.$$

All  $b^*(i, m + 1)$ ,  $b^*(n + 1, j)$  are set to 0.

**Recursion:**  $i = n, \dots, 1, j = m, \dots, 1$  (except  $(n, m)$ );

$$b^M(i, j) = (1 - 2\delta - \tau)p_{x_{i+1}y_{j+1}}b^M(i + 1, j + 1) + \delta \left[ q_{x_{i+1}}b^X(i + 1, j) + q_{y_{j+1}}b^Y(i, j + 1) \right].$$

$$b^X(i, j) = (1 - \varepsilon - \tau)p_{x_{i+1}y_{j+1}}b^M(i + 1, j + 1) + \varepsilon q_{x_{i+1}}b^X(i + 1, j).$$

$$b^Y(i, j) = (1 - \varepsilon - \tau)p_{x_{i+1}y_{j+1}}b^M(i + 1, j + 1) + \varepsilon q_{y_{j+1}}b^Y(i + 1, j).$$

# Part II

## Multiple Alignment

# Multiple Sequence Alignment (Globin family)

```

Helix      AAAAAAAAAAAAAAAAAA  BBBB BBBB BBBB BBBB BBBBBBBBBB
HBA_HUMAN  -----VLSPADKTNVKAAWGKVGA--HAGEYGAEALERMFLSFPTTKTYFPHF
HBB_HUMAN  -----VHLTPEEKSAVTALWGKV---NVDEVGGEALGRLLVVYPWTQRFFESF
MYG_PHYCA  -----VLSEGEWQLVLHVWAKVEA--DVAGHGQDILIRLFKSHPETLEKFDRF
GLB3_CHITP -----LSADQISTVQASFDKVKG-----DPVGILYAVFKADPSIMAKFTQF
GLB5_PETMA PIVDTGSGVAPLSAAEKTIRSAAWAPVYS--TYETSGVDILVKFFTSTPAAQEFFPKF
LGB2_LUPLU -----GALTESQAALVKSSWEEFNA--NIPKHTHRFFILVLEIAPAAKDLFS-F
GLB1_GLYDI -----GLSAAQRQVIAATWKDIAGADNGAGVGKDLIKFLSAHPQMAAVFG-F
Consensus  Ls.... v a W kv . . g . L.. f . P . F F

```

```

Helix      DDDDDDDDEEEEEEEEEEEEEEEEEEEEEEE  FFFFFFFFFFFFFFFF
HBA_HUMAN  -DLS-----HGSAQVKGHGKKVADALTNAVAHV---D--DMPNALSALSDLHAHKL-
HBB_HUMAN  GDLSTPDVAVMGNPKVKAHGKKVLGAFSDGLAHL---D--NLKGTFFATLSELHCDKL-
MYG_PHYCA  KHLKTEAEMKASEDLKKHGVTVLTALGAILKK---K-GHHEAELKPLAQSHATKH-
GLB3_CHITP AG-KDLESIKGTAPFETHANRIVGFFSKIIGEL--P---NIEADVNTFVASHKPRG-
GLB5_PETMA KGLTTADQLKKSADVRWHAERIINAVNDAVASM--DDTEKMSMKLRDLSGKHAKSF-
LGB2_LUPLU LK-GTSEVPQNNPELQAHAGKVFCLVYEAAIQLQVTGVVVTDATLKNLGSVHVSKG-
GLB1_GLYDI SG----AS---DPGVAALGAKVLAQIGVAVSHL--GDEGKMVAQMKAVGVRHKG YGN
Consensus  . t . . . . v..Hg kv. a a...l d . a l. l H .

```

```

Helix      FFGGGGGGGGGGGGGGGGGGGGGG  HHHHHHHHHHHHHHHHHHHHHHHHHHHHH
HBA_HUMAN  -RVDPVNFKLLSHCLLVTLAAHLPAEFTPAVHASLDKFLASVSTVLTSKYR-----
HBB_HUMAN  -HVDPENFRLLGNVLVCVLAHHFGKEFTPPVQAAYQKVAVAGVANALAHKYH-----
MYG_PHYCA  -KIPIKYLEFISEAIIHVLHSRHPGDFGADAQGAMNKALELFRKDIAAKYKELGYQG
GLB3_CHITP --VTHDQLNNFRAGFVS YMK AHT--DFA-GAEAAWGATLDTFFGMIFSKM-----
GLB5_PETMA -QVDPQYFKVLA AVIADTVAAG-----DAGFEKLMSMICILLRSAY-----
LGB2_LUPLU --VADAHFPVVKEAILKTIKEVVGAKWSEELNSAWTIAYDELAIVIKKEMNDAA---
GLB1_GLYDI KHKAQYFEPLGASLLSAMEHRIGGKMNA AAKDAWAAAYADISGALISGLQS----
Consensus  v. f l . . . . . f . aa. k. . l sky

```

# Profile Model (PSSM)

- A natural probabilistic model for a conserved region would be to specify independent probabilities  $e_i(a)$  of observing nucleotide (amino acid)  $a$  in position  $i$
- The probability of a new sequence  $x$  according to this model is

$$P(x \mid M) = \prod_{i=1}^L e_i(x_i)$$

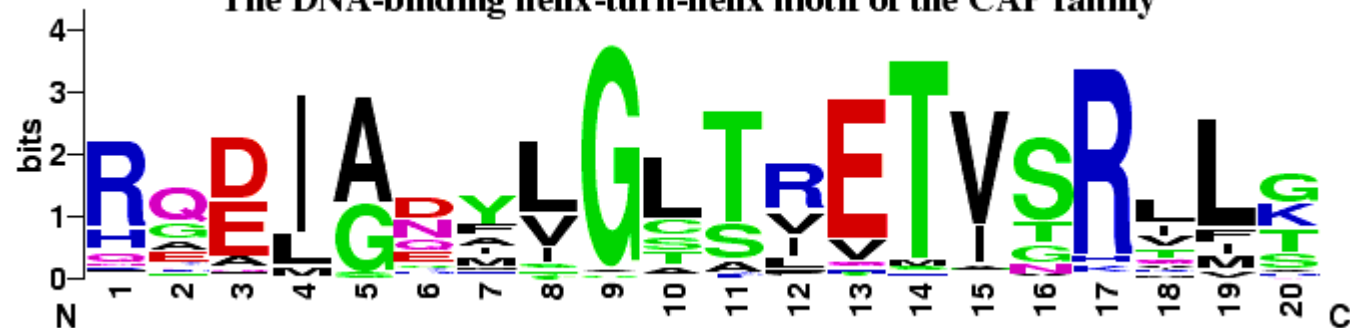
# Profile

- Position Specific Score Matrix (PSSM)

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

A		1				1			.8				
C	.6			1			.4	1		.6	.2		
G			1	.2					.2			.4	1
T	.2				1		.6					.2	
-	.2		.8						.4	.8	.4		

The DNA-binding helix-turn-helix motif of the CAP family





# Searching Profiles: Inference

- Give a sequence  $S$  of length  $L$ , compute the likelihood ratio of being generated from this profile vs. from background model:

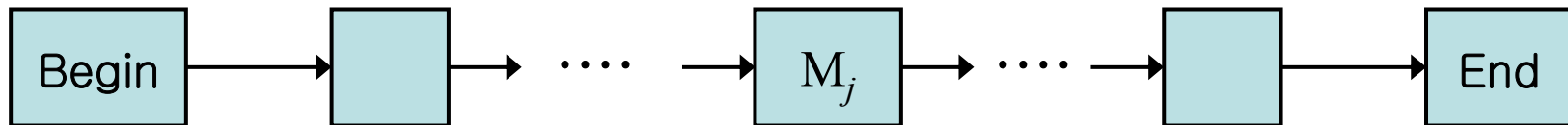
- $R(S|P) = \prod_{i=1}^L \frac{e_i(x_i)}{b_s}$

- Searching motifs in a sequence: sliding window approach

# Match States for Profile HMMs

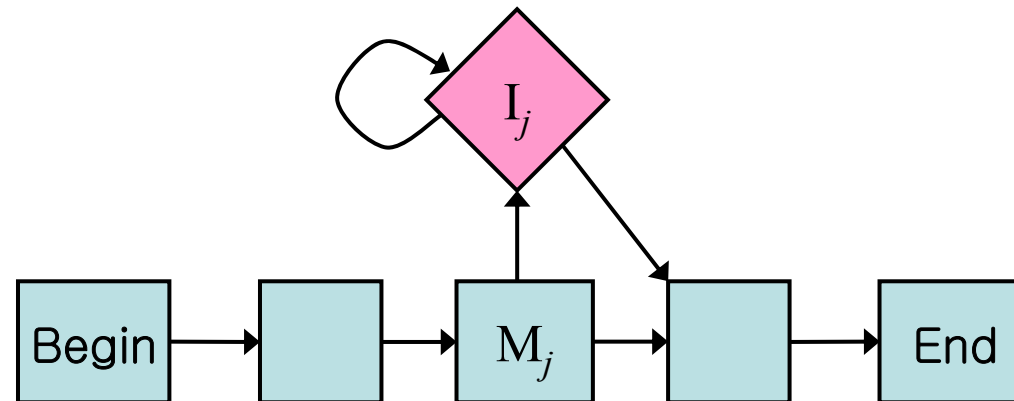
- Match states
  - Emission probabilities

$$e_{M_i}(a)$$



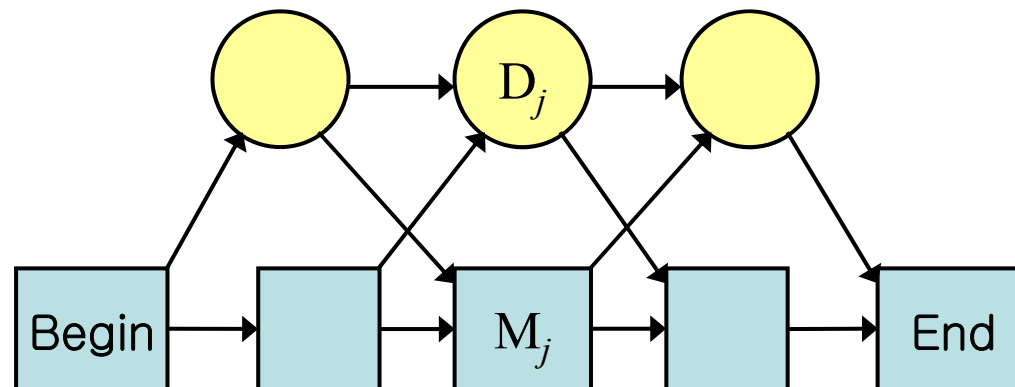
# Components of Profile HMMs

- Insert states  $e_{I_i}(a)$ 
  - Emission prob.
    - Usually back ground distribution  $q_a$ .
  - Transition prob.
    - $M_i$  to  $I_i$ ,  $I_i$  to itself,  $I_i$  to  $M_{i+1}$



# Components of Profile HMMs

- **Delete states** (跳过某些Match State, 而直接过渡到后面的Match State)
  - No emission prob.
  - Cost of a deletion
    - $M \rightarrow D$ ,  $D \rightarrow D$ ,  $D \rightarrow M$
    - Each  $D \rightarrow D$  might be different

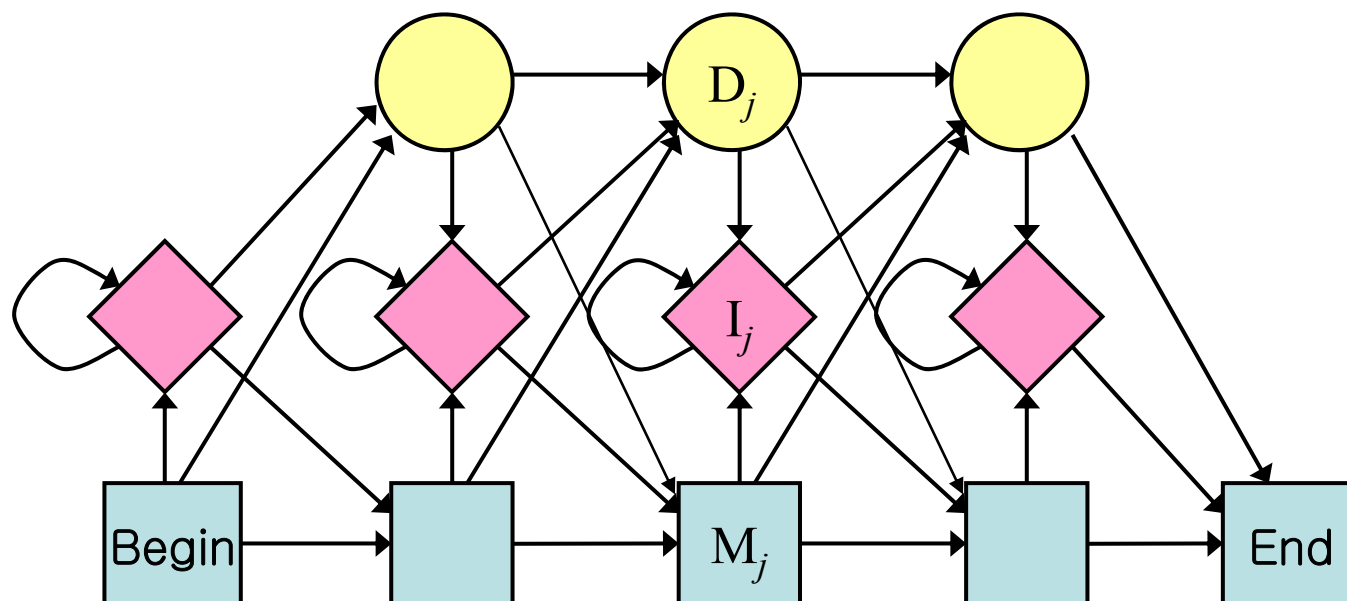


# Why “Delete” State?

- 对序列HBB\_HUMAN, 出现了两个“Delete”, 而直接跳到了M4.
- 避免多个Match状态之间的概率转移, 简化模型参数;

```
HBA_HUMAN    ...VGA--HAGEY...
HBB_HUMAN    ...V----NVDEV...
MYG_PHYCA    ...VEA--DVAGH...
GLB3_CHITP   ...VKG-----D...
GLB5_PETMA   ...VYS--TYETS...
LGB2_LUPLU   ...FNA--NIPKH...
GLB1_GLYDI   ...IAGADNGAGV...
              ***      *****
```

# Full Structure of Profile HMMs



1. 注意到D- $\rightarrow$  I, I- $\rightarrow$  D这样的转移往往很少发生;
2. **Number of match states** 通常取成average sequence length in the family

# Deriving HMMs from Multiple Alignments

- Key idea behind profile HMMs
  - Model representing the consensus for the alignment of sequence from the same family
  - Not the sequence of any particular member

HBA_HUMAN	. . . VGA--HAGEY . . .
HBB_HUMAN	. . . V----NVDEV . . .
MYG_PHYCA	. . . VEA--DVAGH . . .
GLB3_CHITP	. . . VKG-----D . . .
GLB5_PETMA	. . . VYS--TYETS . . .
LGB2_LUPLU	. . . FNA--NIPKH . . .
GLB1_GLYDI	. . . IAGADNGAGV . . .
	*** *****

# Deriving HMMs from Multiple Alignments

- Basic profile HMM parameterization
  - Aim: making the higher probability for sequences from the family
- Parameters
  - the probabilities values : trivial if many of independent alignment sequences are given.

$$a_{kl} = \frac{A_{kl}}{\sum_{l'} A_{kl'}} \quad e_k(a) = \frac{E_k(a)}{\sum_{a'} E_k(a')}$$

- length of the model: heuristics or systematic way



# Estimation of Prob.

- Maximum likelihood (ML) estimation
  - given observed freq.  $c_{ja}$  of residue  $a$  in position  $j$ .

$$e_{M_j}(a) = \frac{c_{ja}}{\sum_{a'} c_{ja'}}$$

- Simple pseudocounts (Dirichlet Prior)
  - $q_a$ : background distribution
  - $A$ : weight factor

$$e_{M_j}(a) = \frac{c_{ja} + Aq_a}{A + \sum_{a'} c_{ja'}}$$

# Searching with Profile HMMs

- Main usage of profile HMMs
  - Detecting potential sequences in a family
  - Matching a sequence to the profile HMMs
    - Viterbi algorithm or forward algorithm
  - Comparing the resulting probability with random model

$$P(x | R) = \prod_i q_{x_i}$$

# Searching with Profile HMMs

- Viterbi algorithm (optimal log-odd alignment)

$$V_j^M(i) = \log \frac{e_{M_j}(x_i)}{q_{x_i}} + \max \begin{cases} V_{j-1}^M(i-1) + \log a_{M_{j-1}M_j}, \\ V_{j-1}^I(i-1) + \log a_{I_{j-1}M_j}, \\ V_{j-1}^D(i-1) + \log a_{D_{j-1}M_j}; \end{cases}$$

$$V_j^I(i) = \log \frac{e_{I_j}(x_i)}{q_{x_i}} + \max \begin{cases} V_j^M(i-1) + \log a_{M_jI_j}, \\ V_j^I(i-1) + \log a_{I_jI_j}, \\ V_j^D(i-1) + \log a_{D_jI_j}; \end{cases}$$

$$V_j^D(i) = \max \begin{cases} V_{j-1}^M(i) + \log a_{M_{j-1}D_j}, \\ V_{j-1}^I(i) + \log a_{I_{j-1}D_j}, \\ V_{j-1}^D(i) + \log a_{D_{j-1}D_j}; \end{cases}$$

# Viterbi算法说明

- 通常取Insert状态下的Emission probability为背景概率分布，因此上面的第二个式子中不Emission项为0;
- 通常D->I, I->D的状态转移都非常小，第三式子中可能只有D-> M 和M-> D的转移。

# Searching with Profile HMMs

- Forward algorithm: summing over all potent alignments

$$F_j^M(i) = \log \frac{e_{M_j}(x_i)}{q_{x_i}} + \log[a_{M_{j-1}M_j} \exp(F_{j-1}^M(i-1)) \\ + a_{I_{j-1}M_j} \exp(F_{j-1}^I(i-1)) + a_{D_{j-1}M_j} \exp(F_{j-1}^D(i-1))];$$

$$F_j^I(i) = \log \frac{e_{I_j}(x_i)}{q_{x_i}} + \log[a_{M_jI_j} \exp(F_j^M(i-1)) \\ + a_{I_jI_j} \exp(F_j^I(i-1)) + a_{D_jI_j} \exp(F_j^D(i-1))];$$

$$F_j^D(i) = \log[a_{M_{j-1}D_j} \exp(F_{j-1}^M(i)) + a_{I_{j-1}D_j} \exp(F_{j-1}^I(i)) \\ + a_{D_{j-1}D_j} \exp(F_{j-1}^D(i))];$$

# An Example

A C A - - - A T G

T C A A C T A T C

A C A C - - A G C

A G A - - - A T C

A C C G - - A T C

How could we characterize this (hypothetical) family of nucleotide sequences?

- Keep the Multiple Alignment

- Try a regular expression

[AT] [CG] [AC] [ACTG]\* A [TG]  
[GC]

- But what about?

- T G C T - - A G G *vrs*

- A C A C - - A T C

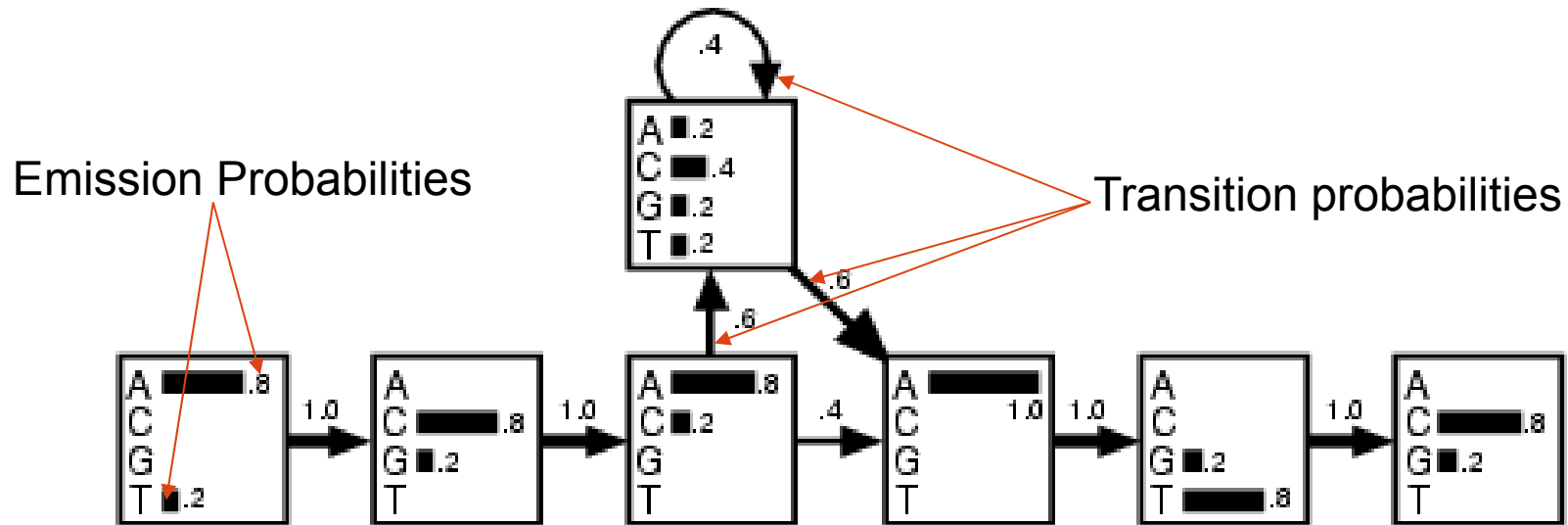
- Try a consensus sequence:

A C A - - - A T C

- Depends on distance measure

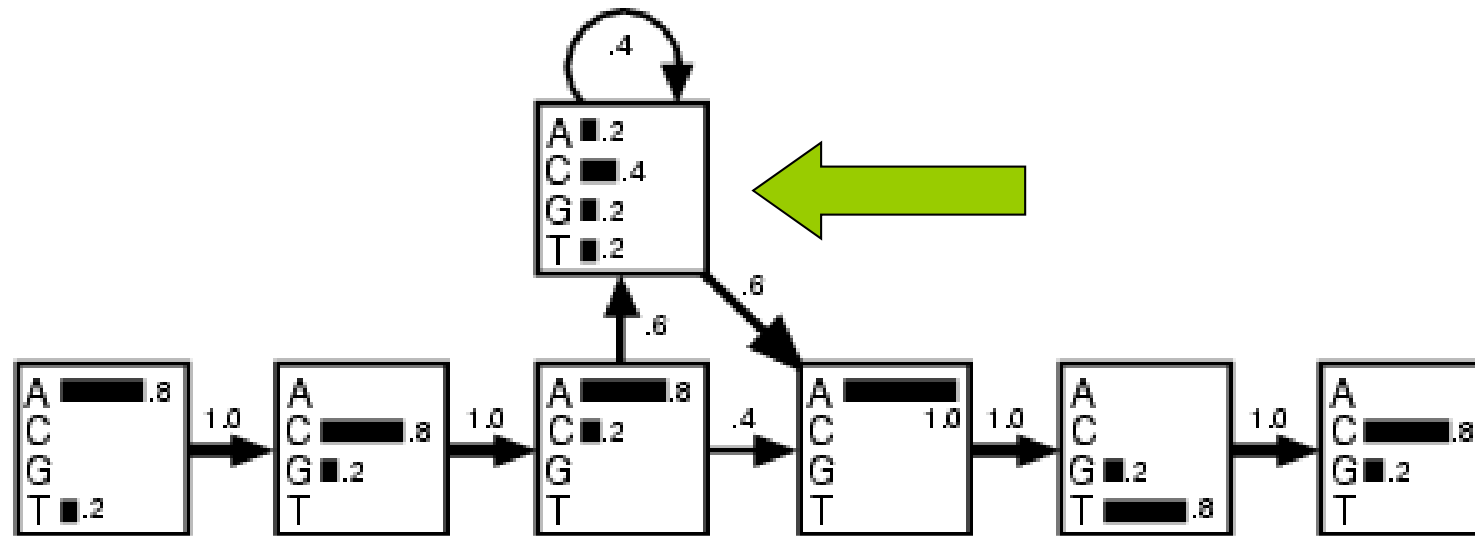
Example borrowed from Salzberg, 1998

# An Example



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

# Insert (Loop) States



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

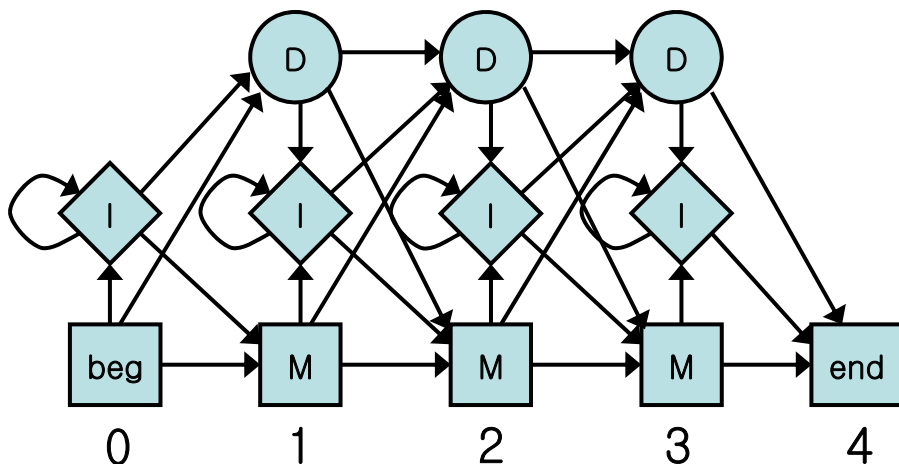


# Optimal Model Construction: Mark Columns

(a) Multiple alignment:

	x	x	.	.	.	x
bat	A	G	-	-	-	C
rat	A	-	A	G	-	C
cat	A	G	-	A	A	-
gnat	-	-	A	A	A	C
goat	A	G	-	-	-	C
	1	2	.	.	.	3

(b) Profile-HMM architecture:



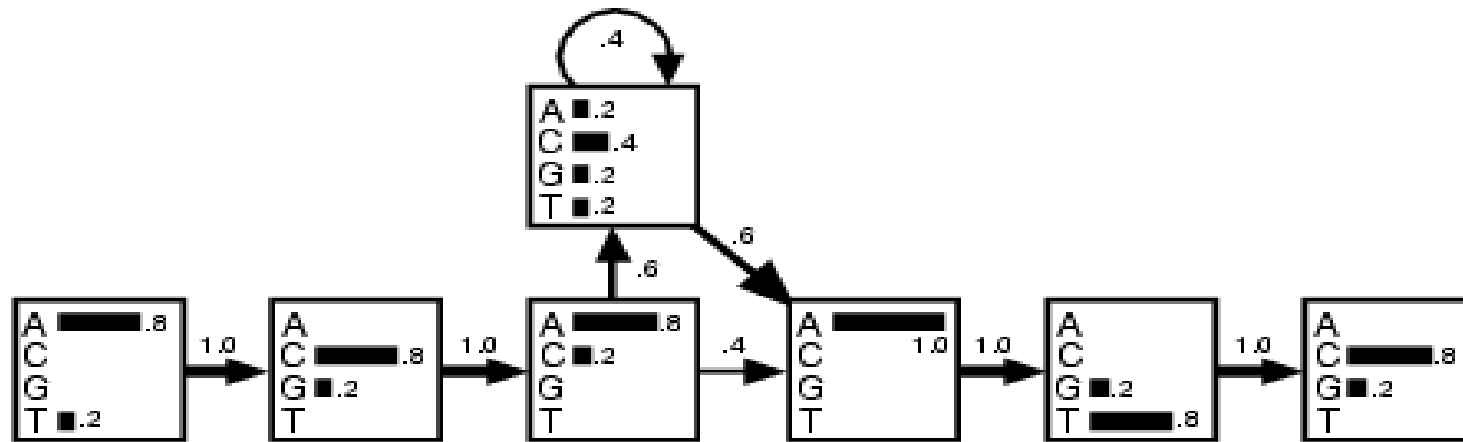
(c) Observed emission/transition counts

		0	1	2	3
match emissions	A	-	4	0	0
	C	-	0	0	4
	G	-	0	3	0
	T	-	0	0	0
insert emissions	A	0	0	6	0
	C	0	0	0	0
	G	0	0	1	0
	T	0	0	0	0
state transitions	M-M	4	3	2	4
	M-D	1	1	0	0
	M-I	0	0	1	0
	I-M	0	0	2	0
	I-D	0	0	1	0
	I-I	0	0	4	0
	D-M	-	0	0	1
	D-D	-	1	0	0
	D-I	-	0	2	0

# Align Sequences to Profile HMM

- Align a sequence to a profile HMM → Viterbi algorithm
- Construction a multiple alignment just requires calculating a Viterbi alignment for each individual sequence.
  - Residues aligned to the same match state in the profile HMM should be aligned in the same columns.

# Scoring the Simple HMM



- #1 - “T G C T - - A G G” vs: #2 - “A C A C - - A T C”
  - Regular Expression ([AT] [CG] [AC] [ACTG]\* A [TG] [GC]):
    - #1 = Member
    - #2: Member
  - HMM:
    - #1 = Score of 0.0023%    #2 Score of 4.7% (Probability)
    - #1 = Score of -0.97      #2 Score of 6.7 (Log odds)

# Multiple Alignment with a Known Profile HMM

- | Position | 1 | 2 | 3 | 4 | 5 | 6 | insert  | 7 | 8 | 9 | 10 | 11 |
|----------|---|---|---|---|---|---|---------|---|---|---|----|----|
|          | F | P | H | F | – | D | LS      | H | G | S | A  | Q  |
|          | F | E | S | F | G | D | LSTPDAV | M | G | N | P  | K  |
|          | F | D | R | F | K | H | LKTEAEM | K | A | S | E  | D  |
|          | F | T | Q | F | A | G | KDLESI  | K | G | T | A  | P  |
|          | F | P | K | F | K | G | LTTADQL | K | K | S | A  | D  |
|          | F | S | – | F | L | K | GTSEVP  | Q | N | N | P  | E  |
|          | F | G | – | F | S | G | AS      | – | – | D | P  | G  |

# Profile HMM

## Training from Unaligned Sequences

- Harder problem
  - estimating both a model and a multiple alignment from initially unaligned sequences.
  - Initialization: Choose the length of the profile HMM and initialize parameters.
  - Training: estimate the model using the Baum-Welch algorithm (iteratively).
  - Multiple Alignment: Align all sequences to the final model using the Viterbi algorithm and build a multiple alignment as described in the previous section.

# Profile HMM

## Training from Unaligned Sequences

- Initial Model
  - The only decision that must be made in choosing an initial structure for Baum-Welch estimation is the length of the model  $M$ .
  - A commonly used rule is to set  $M$  be the average length of the training sequence.
  - We need some randomness in initial parameters to avoid local maxima.

# References

- S. Durbin, S. Eddy, A. Krogh and G. Mitchison. Biological Sequence Analysis—Probabilistic Models of Proteins and Nucleic Acids. 1998, Cambridge University Press.