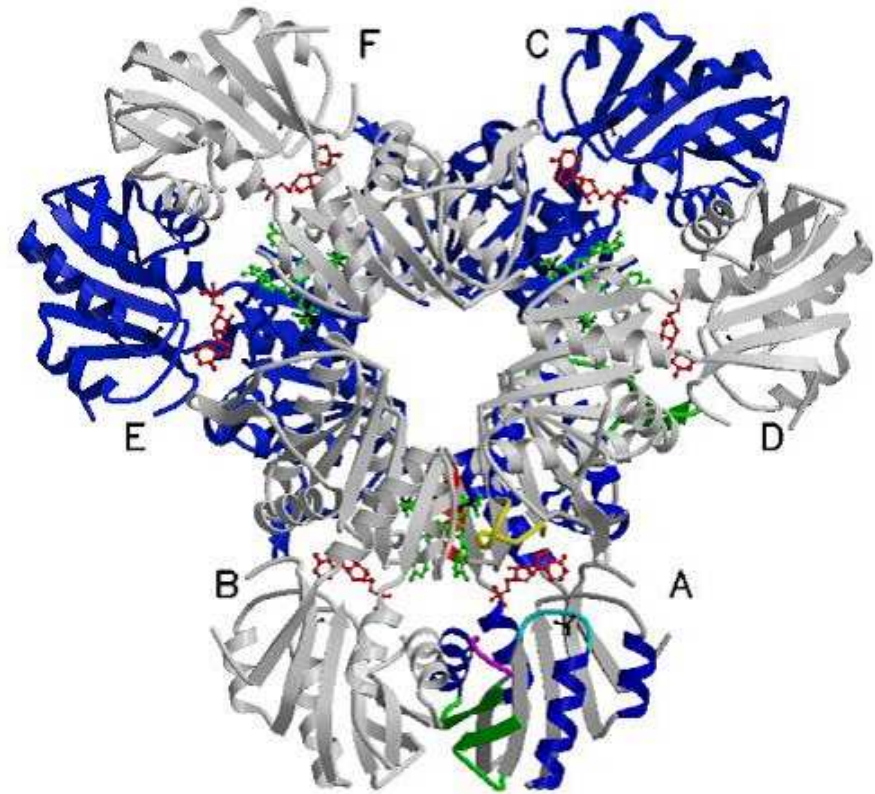


HMMs for Pairwise Sequence Alignment

based on Ch. 4 from
Biological Sequence Analysis
by R. Durbin et al., 1998

Acknowledgement:
M.Sc. student Oana Răţoi



[*B. subtilis* PRPP synthase]

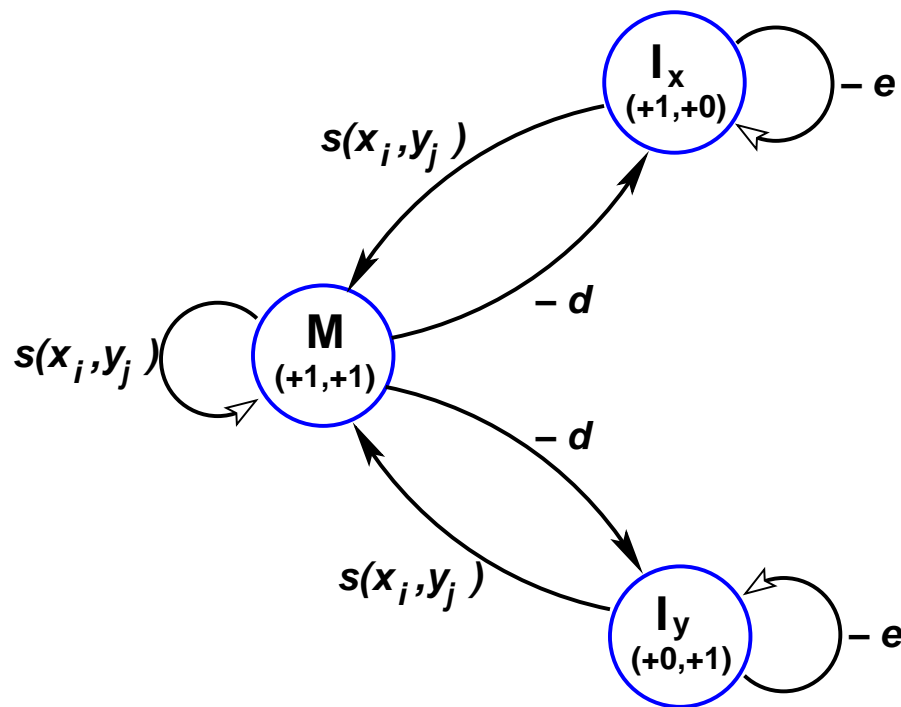
PLAN

- 1 How to build a **pair HMM** for pairwise alignment with affine gap penalties ...starting from the **FSA** for more complex pairwise alignments introduced in Ch. 2.

Advantages of using such a **full probabilistic model for pairwise alignment**:

- 2 evaluate the **similarity of two sequences independently of any specific alignment**, by weighting all alternatives, and assess the **reliability of the alignment** obtained by dynamic programming
- 3 explore **suboptimal alignments** by **probabilistic sampling**
- 4 evaluate the **posterior probability** that x_i is aligned to y_j ;
define the **overall accuracy of an alignment**;
design an **algorithm** for getting the maximal overall accuracy alignment

Remember: A simple FSA for global pairwise sequence alignment with affine gap penalties



Example: VLSPAD-K
HL--AESK

Computing the values of the states

- Initialisations:**

$$V^M(0,0) = 0, V^X(0,0) = V^Y(0,0) = 0$$

- Recurrence relations:**

$$V^M(i,j) = s(x_i, y_j) + \max \begin{cases} V^M(i-1, j-1), \\ V^X(i-1, j-1), \\ V^Y(i-1, j-1), \end{cases}$$

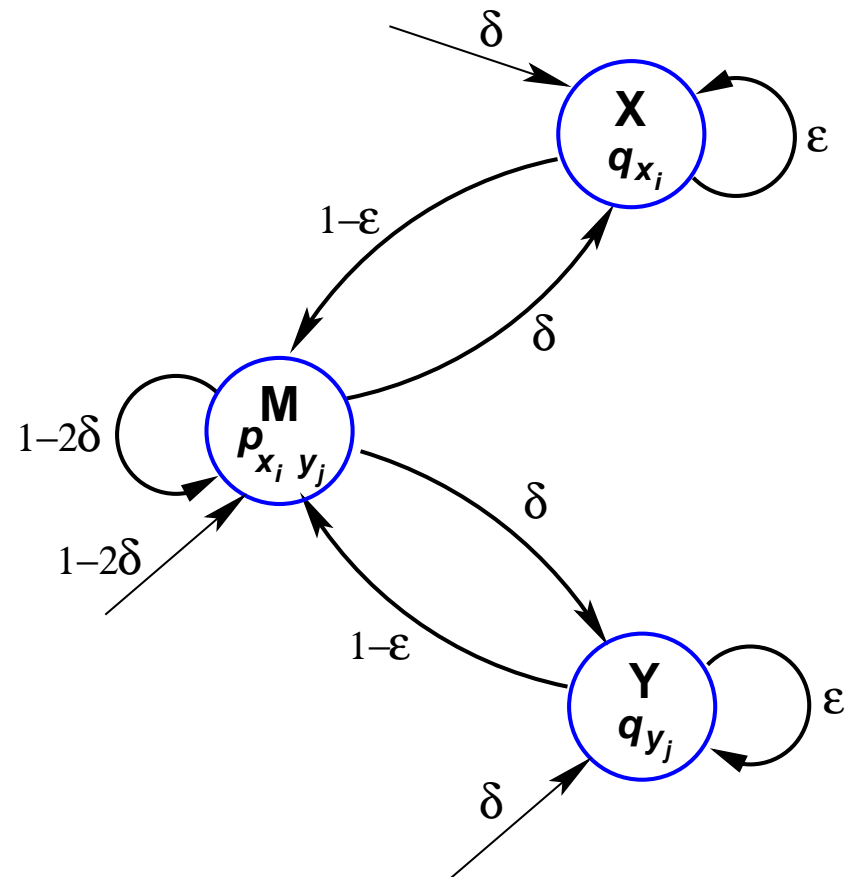
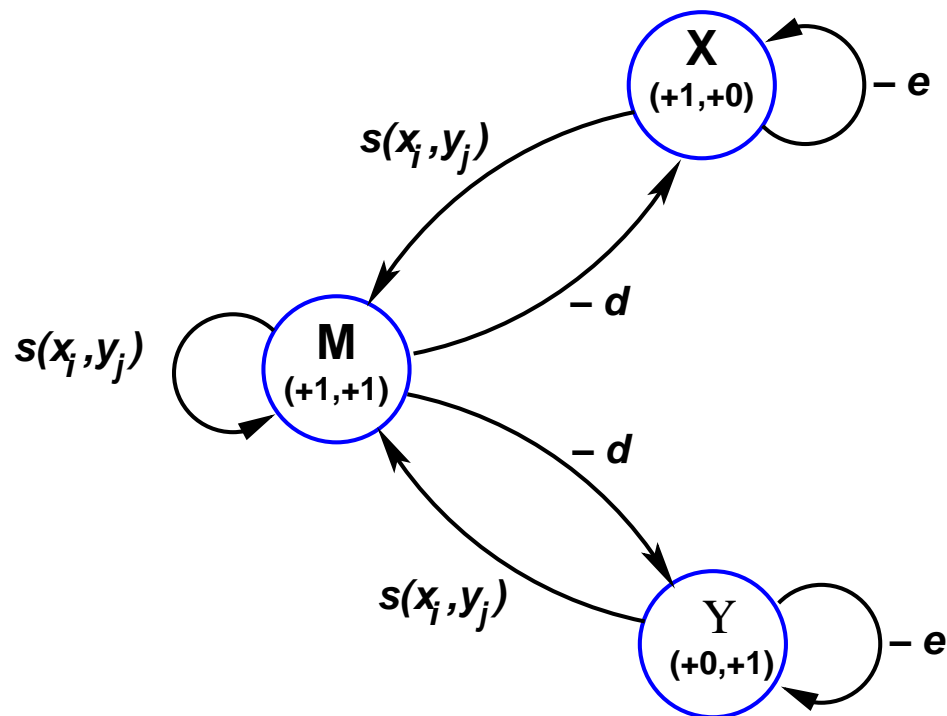
$$V^X(i,j) = \max \begin{cases} V^M(i-1, j) - d, \\ V^X(i-1, j) - e, \end{cases}$$

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

1 How to build a Pair HMM?

A first, FSA-inspired version

3.



The parameters

- Set probabilities for **transitions** between states
 - transition from M to an insert state: δ
 - probability of staying in an insert state: ϵ
- Set probabilities for **emissions** of symbols from the states
 - p_{ab} for emitting an aligned pair $a : b$ from a M state
 - q_a for emitting symbol a against a gap
- Set the initial transitions to each state to be the same as from the M state.

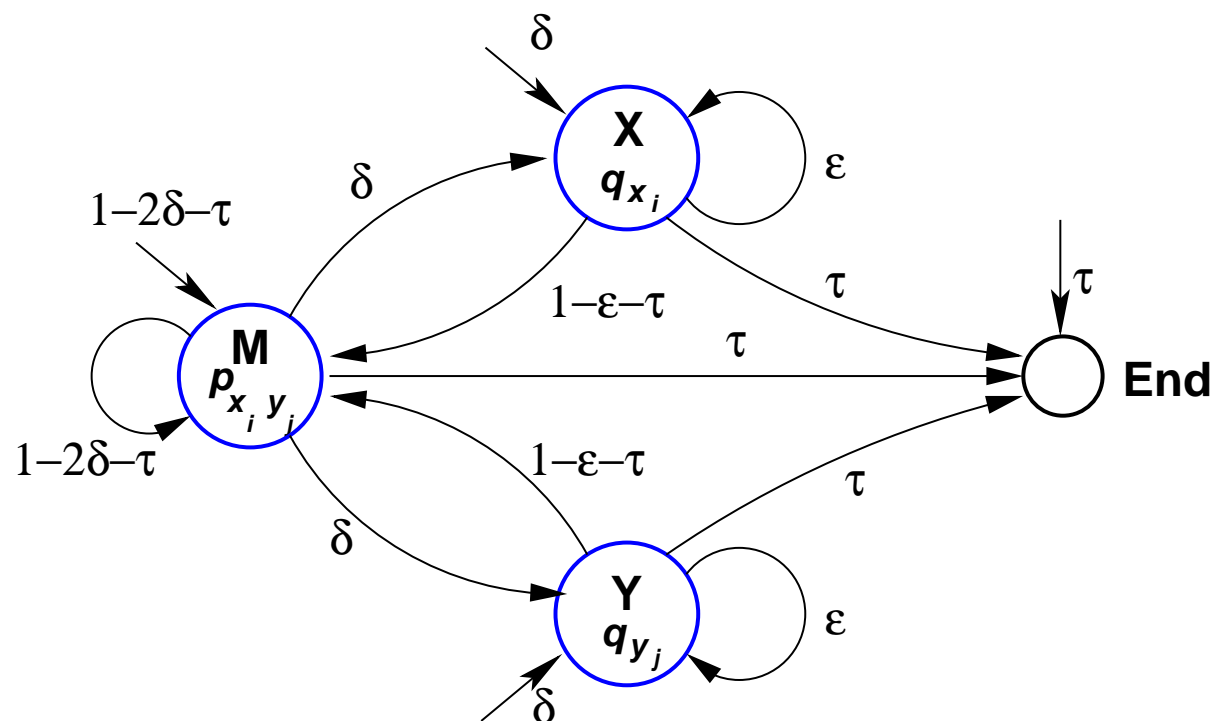
Remark: The HMM shown on the previous slide generates pair alignments, except the pair of empty sequences.

To get a **full probabilistic version**:

- Define a **new state** End, and add a **new parameter** τ for the probability of the transition(s) into the End state.

Pair HMM: a full probabilistic version

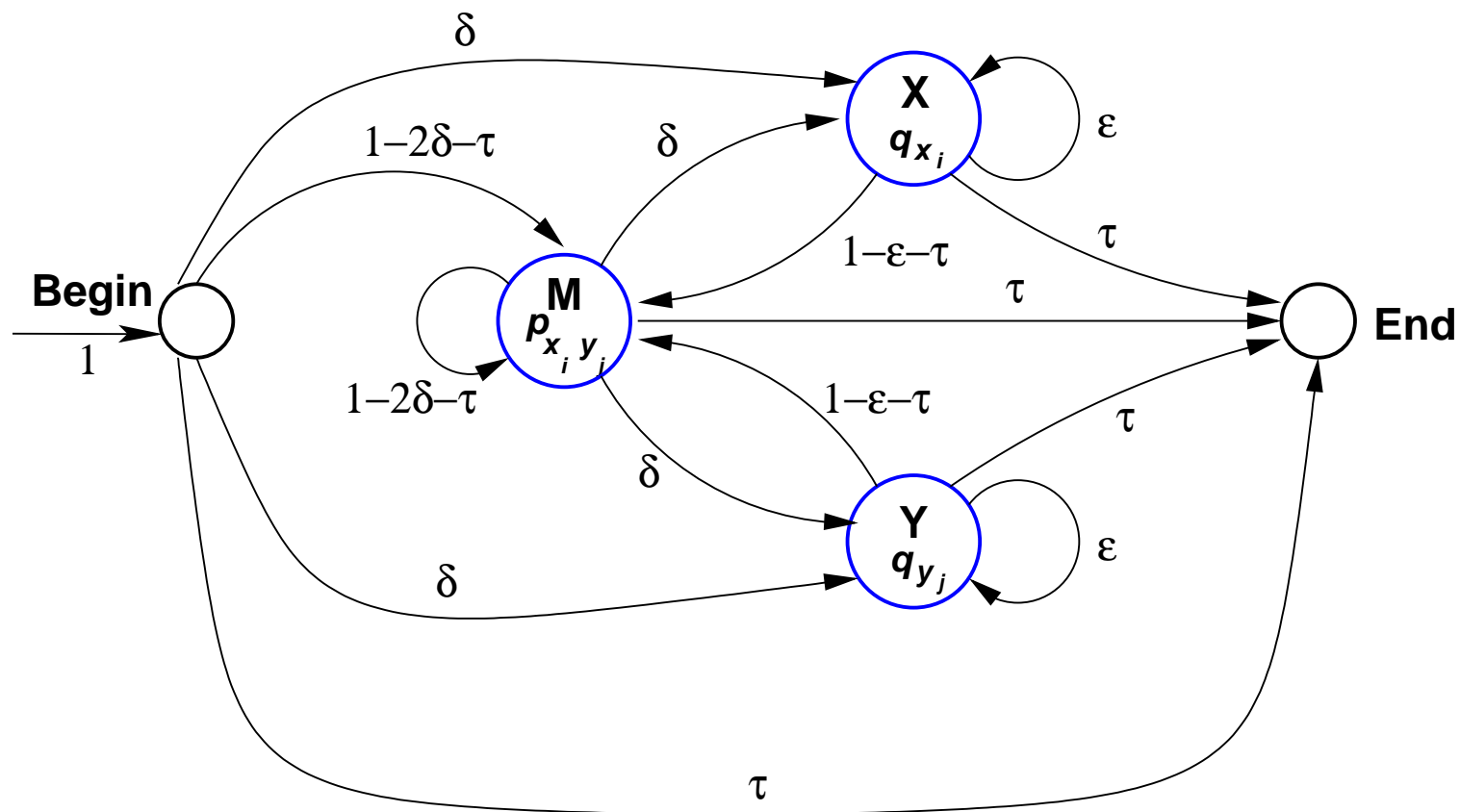
5.



In this HMM, the emission of (pairs of) symbols is done when reaching the (non-End) states.

To simplify the initialisation relations in the pair HMM algorithms (and also for symmetry with the End state), we will also add a silent state Begin.

Pair HMM: the final version



How does this pair HMM work?

- start in the Begin state;
- pick the next state according to the distribution of transition probabilities leaving the current state;
- pick a symbol pair to be added to the alignment according to the emission distribution in the new state;
- cycle over the previous two steps, while the End state is not reached.

Remark

Throughout this chapter and also in the next one (Profile HMMs), the **emission probabilities**, that have been denoted b_{ijk} in [Manning & Schütze, 2000] Ch. 9, are considered independent of the origin state i .

As a consequence, **slightly different definitions** (and notations) will be used **for forward, backward and Viterbi probabilities**, compared to our HMM slides, which were based on [Manning & Schütze, 2000].

$$\begin{aligned}f_k(i) &= P(x_1 \dots x_i, \pi_i = k) \\b_k(i) &= P(x_{i+1} \dots x_L \mid \pi_i = k) \\v_k(i) &= \max_{\pi_1 \dots \pi_i} P(x_1 \dots x_i, \pi_1 \dots \pi_i, \pi_i = k)\end{aligned}$$

See also [Durbin et al, 1998], pages 56–59.

The Viterbi algorithm for pair HMMs

- **Notation:** $v^k(i, j) = \max_{\pi} P(x_1 \dots x_i, y_1 \dots y_j, \pi_{ij} = k)$ will correspond to the most probable alignment up to the positions i, j , ending in state k .

Remark: To simplify the equations to be given below, the Begin state is defined as being the M state.

- **Initialisation:** $v^M(0, 0) = 1$, and all other $v^\bullet(i, 0) = 0$, $v^\bullet(0, j) = 0$

(See Remark on page 116 in [Borodowsky and Ekisheva, 2006].)

- **Recurrence relations:**

$$v^M(i, j) = p_{x_i y_j} \max \begin{cases} (1 - 2\delta - \tau) v^M(i - 1, j - 1) \\ (1 - \epsilon - \tau) v^X(i - 1, j - 1) \\ (1 - \epsilon - \tau) v^Y(i - 1, j - 1) \end{cases} \quad \text{for } i \geq 1, j \geq 1$$

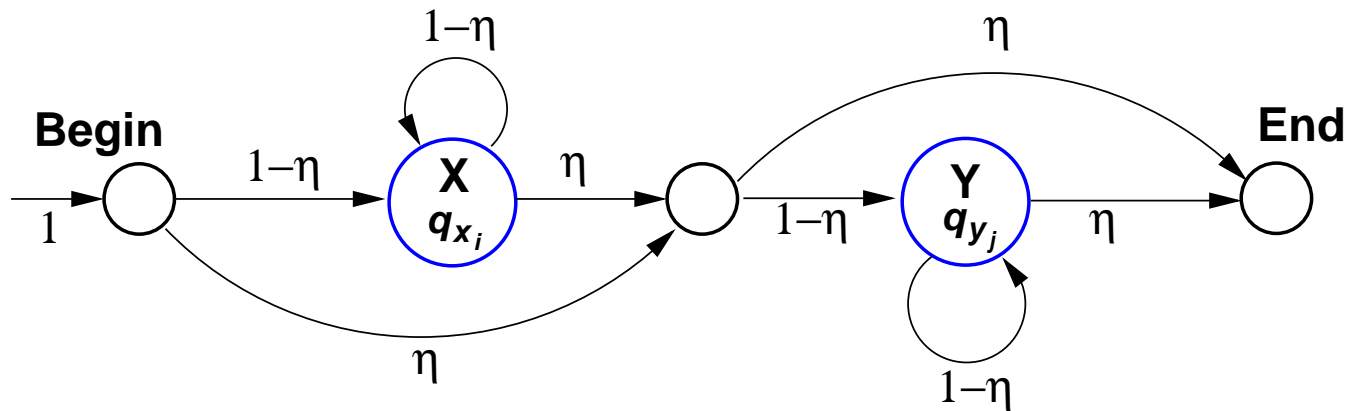
$$v^X(i, j) = q_{x_i} \max \begin{cases} \delta v^M(i - 1, j) \\ \epsilon v^X(i - 1, j) \end{cases} \quad \text{for } i \geq 1, j \geq 0$$

$$v^Y(i, j) = q_{y_j} \max \begin{cases} \delta v^M(i, j - 1) \\ \epsilon v^Y(i, j - 1) \end{cases} \quad \text{for } i \geq 0, j \geq 1$$

- **Termination:** $v^E = \tau \max(v^M(n, m), v^X(n, m), v^Y(n, m))$

An independence model for pairwise sequence alignment

10.



- The **main states** are X and Y, which emit the two sequences in turn, independently of each other.
- There is a **silent state** between X and Y, that doesn't emit any symbols.
- The **probability of a pair of sequences x and y** :

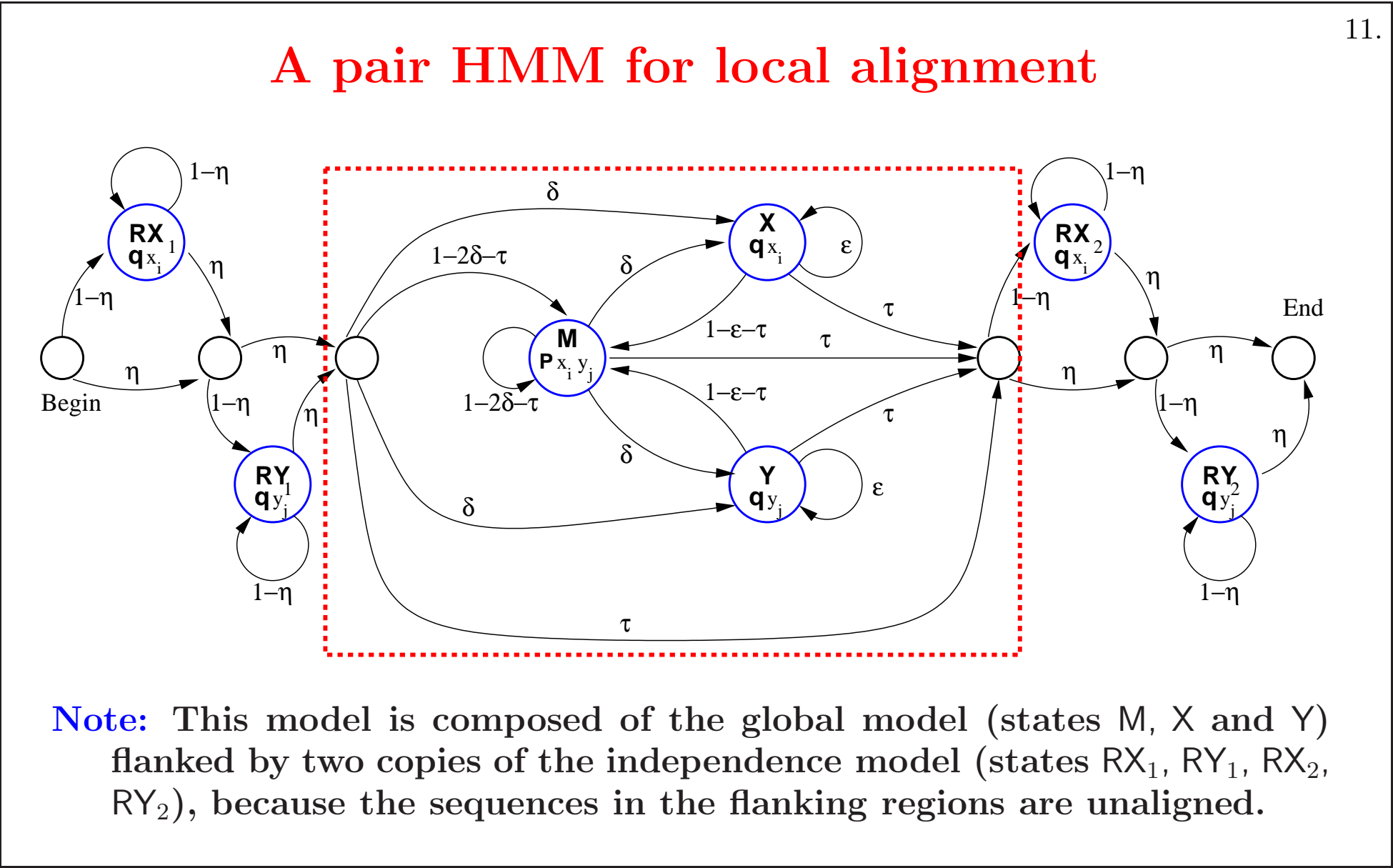
$$P(x, y|R) = \eta(1 - \eta)^n \prod_{i=1}^n q_{x_i} \eta(1 - \eta)^m \prod_{j=1}^m q_{y_j} = \eta^2(1 - \eta)^{n+m} \prod_{i=1}^n q_{x_i} \prod_{j=1}^m q_{y_j}$$

A pair HMM for local alignment

The diagram illustrates a pair HMM for local alignment. It consists of several states and transitions:

- States:**
 - Begin** (white circle)
 - RX₁** (blue circle, labeled q_{x_1})
 - Transition state** (white circle)
 - M** (blue circle, labeled $p_{x_i y_j}$)
 - X** (blue circle, labeled q_{x_i})
 - Y** (blue circle, labeled q_{y_j})
 - Transition state** (white circle)
 - RX₂** (blue circle, labeled q_{x_i})
 - End** (white circle)
- Transitions and Probabilities:**
 - Begin to RX₁:** η
 - RX₁ to Transition state:** η
 - Transition state to M:** δ
 - M to X:** δ
 - M to Y:** δ
 - X to Transition state:** τ
 - Y to Transition state:** τ
 - Transition state to RX₂:** η
 - RX₂ to End:** η
 - End to End:** η
 - Self-loops:**
 - RX₁:** $1-\eta$
 - RX₂:** $1-\eta$
 - X:** ϵ
 - Y:** ϵ
- Red Dashed Box:** Encloses the central part of the model, including states M, X, Y and the transition states.

Note: This model is composed of the global model (states M, X and Y) flanked by two copies of the independence model (states RX₁, RY₁, RX₂, RY₂), because the sequences in the flanking regions are unaligned.



A pair HMM for local alignment

The diagram illustrates a pair HMM for local alignment. It consists of several states and transitions:

- States:**
 - Begin** (white circle)
 - RX₁** (blue circle, labeled q_{x_1})
 - Transition state** (white circle)
 - M** (blue circle, labeled $p_{x_i y_j}$)
 - X** (blue circle, labeled q_{x_i})
 - Y** (blue circle, labeled q_{y_j})
 - Transition state** (white circle)
 - RX₂** (blue circle, labeled q_{x_i})
 - End** (white circle)
- Transitions and Probabilities:**
 - Begin to RX₁:** η
 - RX₁ to Transition state:** η
 - Transition state to M:** δ
 - M to X:** δ
 - M to Y:** δ
 - X to Transition state:** τ
 - Y to Transition state:** τ
 - Transition state to RX₂:** η
 - RX₂ to End:** η
 - End to End:** η
 - Self-loops:**
 - RX₁:** $1-\eta$
 - RX₂:** $1-\eta$
 - X:** ϵ
 - Y:** ϵ
- Red Dashed Box:** Encloses the central part of the model, including states M, X, Y and the transition states.

Note: This model is composed of the global model (states M, X and Y) flanked by two copies of the independence model (states RX₁, RY₁, RX₂, RY₂), because the sequences in the flanking regions are unaligned.

2 The full probability of aligning x and y , summing over all paths

Problem:

When two sequences have a **weak similarity**, it is hard to identify the correct (biologically meaningful) alignment.

Alternative:

Using the pair HMM **forward algorithm**, we can calculate the probability that a given pair of sequences are related by *any* alignment.

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

Note: If there is an unambiguous best/Viterbi alignment π^* , almost all of the probability $P(x, y)$ will be contributed by $P(x, y, \pi^*)$. But when there are many comparable alternative alignments or alternative variations, the probability $P(x, y)$ can be significantly different from $P(x, y, \pi^*)$.

Algorithm: Forward calculation for pair HMM

- **Notation:** $f^k(i, j)$ is the combined probability of all alignments up to the positions i, j , ending in state k : $f^k(i, j) = P(x_1 \dots x_i, y_1 \dots y_j, \pi_{ij} = k)$.

- **Initialisation:**

$$f^M(0, 0) = 1 \text{ and all other } f^\bullet(i, 0) = 0, f^\bullet(0, j) = 0$$

- **Recursion relations:**

$$f^M(i, j) = p_{x_i y_j} [(1 - 2\delta - \tau)f^M(i-1, j-1) + (1 - \epsilon - \tau)(f^X(i-1, j-1) + f^Y(i-1, j-1))], \text{ for } i \geq 1, j \geq 1$$

$$f^X(i, j) = q_{x_i} [\delta f^M(i-1, j) + \epsilon f^X(i-1, j)], \text{ for } i \geq 1, j \geq 0$$

$$f^Y(i, j) = q_{y_j} [\delta f^M(i, j-1) + \epsilon f^Y(i, j-1)], \text{ for } i \geq 0, j \geq 1$$

- **Termination:** $f^E(n, m) = \tau(f^M(n, m) + f^X(n, m) + f^Y(n, m))$

Note

The log-odds ratio of the full probability $P(x, y) = f^E(n, m)$ to the independence model probability is a measure of the **likelihood that the two sequences are related** to each other by some unspecified alignment, as opposed to being unrelated.

The posterior probability $P(\pi \mid x, y)$ over alignments π , given x and y

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

In particular, for the **Viterbi** path:

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

Note: As already mentioned, this probability can be very small.

For **example**, for the alignment of two highly diverged proteins shown below — the human alpha globin and the leghaemoglobin from yellow lupin —, this probability is 4.6×10^{-6} .

HBA_HUMAN	GSAQVKGHGKKVADALTNAVAHV---D--DMPNALSALSDDLHAHKL
	++ ++++H+ KV + +A ++ +L+ L+++H+ K
LGB2_LUPLU	NNPELQAHAGKVFKLVYEAAIQLQVTGVVVTDLTKNLGSVHVSKG

3 Suboptimal alignment by probabilistic sampling

Idea: [How to generate a sample alignment]

While tracing back through the matrix of $f^k(i, j)$ values instead of taking the highest choice at each step, make a probabilistic choice based on the relative strengths of the three components.

Probabilistic sampling of alignments: Technical details

- Suppose that at a certain step in this traceback, we are in state **M** at position (i, j) :

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

$$\text{Then choose: } \begin{cases} M(i - 1, j - 1) & \text{with probability } \frac{p_{x_i y_j} (1 - 2\delta - \tau) f^M(i - 1, j - 1)}{f^M(i, j)} \\ X(i - 1, j - 1) & \text{with probability } \frac{p_{x_i y_j} (1 - \epsilon - \tau) f^X(i - 1, j - 1)}{f^M(i, j)} \\ Y(i - 1, j - 1) & \text{with probability } \frac{p_{x_i y_j} (1 - \epsilon - \tau) f^Y(i - 1, j - 1)}{f^M(i, j)} \end{cases}$$

- If we were in the state **X** at cell (i, j) ,

$$\text{then choose: } \begin{cases} M(i - 1, j) & \text{with probability } \frac{q_{x_i} \delta f^M(i - 1, j)}{f^X(i, j)} \\ X(i - 1, j) & \text{with probability } \frac{q_{x_i} \epsilon f^X(i - 1, j)}{f^X(i, j)} \end{cases}$$

- Similarly if we were in the state **Y** at cell (i, j) .

A set of sample global alignments

HEAGAWGHEE	HEAGAWGHEE	HEAGAWGHEE	HEAGAWGHEE
-P-A-WHEAE	-P--AWHEAE	P---AWHEAE	-P-A-WHEAE

HEAGAWGHE-E	HEAGAWGHE-E	HEAGAWGHE-E	HEAGAWGHEE
-PA--W-HEAE	--P-AW-HEAE	-P--AW-HEAE	--PA-WHEAE

Note: The rightmost alignment on the top row coincides with the first alignment.
Such a repetition is possible when doing probabilistic sampling.

4 The posterior probability that x_i is aligned to y_j

Gives a reliability measure for each part of an alignment.

- The combined probability of all the alignments that pass through a specified matched pair of residues (x_i, y_j) :

$$\begin{aligned}
 P(x, y, x_i \diamond y_j) &= P(x_{1\dots i}, y_{1\dots j}, x_i \diamond y_j) P(x_{i+1} \dots n, y_{j+1} \dots m \mid x_{1\dots i}, y_{1\dots j}, x_i \diamond y_j) \\
 &= P(x_{1\dots i}, y_{1\dots j}, x_i \diamond y_j) P(x_{i+1} \dots n, y_{j+1} \dots m \mid x_i \diamond y_j) \\
 &= f^M(i, j) b^M(i, j)
 \end{aligned}$$

where $f^M(i, j)$ and $b^M(i, j)$ are computed by the forward algorithm and respectively the **backward algorithm**.

- Then one can compute the posterior probability that x_i is aligned to y_j :

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

and similarly the posterior probabilities of using specific insert states.

- If the ratio is close to 1, then the match is highly reliable. If it is near to 0, the match is unreliable.

Algorithm: Backward calculation for pair HMMs

Notation: $b^k(i, j)$: the combined probability of all alignments starting from the positions $i + 1, j + 1$, assuming that we begin from state k :
 $b^k(i, j) = P(x_{i+1} \dots x_n, y_{j+1} \dots y_m \mid \pi_{ij} = k)$.

Initialisation:

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

Recursion relations:

For all $i \leq n + 1, j \leq m + 1$ except $(n + 1, m + 1)$:

$$\begin{aligned} b^M(i, j) &= (1 - 2\delta - \tau)p_{x_{i+1}y_{j+1}}b^M(i + 1, j + 1) + \\ &\quad \delta [q_{x_{i+1}}b^X(i + 1, j) + q_{y_{j+1}}b^Y(i, j + 1)] \text{ for } i \geq 0, j \geq 0 \\ b^X(i, j) &= (1 - \epsilon - \tau)p_{x_{i+1}y_{j+1}}b^M(i + 1, j + 1) + \epsilon q_{x_{i+1}}b^X(i + 1, j) \text{ for } i \geq 1, j \geq 0 \\ b^Y(i, j) &= (1 - \epsilon - \tau)p_{x_{i+1}y_{j+1}}b^M(i + 1, j + 1) + \epsilon q_{y_{j+1}}b^Y(i, j + 1) \text{ for } i \geq 0, j \geq 1 \end{aligned}$$

Termination:

There is no special termination, because we need the $b^\bullet(i, j)$ values for $i, j \geq 1$

The expected accuracy of an alignment

- A natural measure of the **overall accuracy of an alignment** π : the expected number of correct matches in π :

$$\mathcal{A}(\pi) = \sum_{(i,j) \in \pi} P(x_i \diamond y_j \mid x, y)$$

- An **algorithm** that obtains a complete **alignment with maximal overall accuracy** uses standard dynamic programming with score values given by the posterior probabilities of pair matches, without gap costs.

$$A(i, j) = \max \begin{cases} A(i-1, j-1) + P(x_i \diamond y_j \mid x, y), \\ A(i-1, j), \\ A(i, j-1), \end{cases}$$

The standard traceback will produce the best alignment, which is a legitimate alignment (not necessarily the Viterbi alignment; see the example on the next slides).

Remark: The algorithm works for any sort of gap scores!

Example

The next three slides show the posterior probabilities for the example data used in Ch. 2. The tables correspond to M, X and Y states respectively. Values are shown as percentages, i.e. 100 times the relevant probability rounded to the nearest integer. The path indicated is the **optimal accuracy path** produced by the algorithm, as shown in the fourth slide.

23.

Match:

$P(x_i \diamond y_j \mid x, y) = \frac{f^M(i,j) b^M(i,j)}{f^E(n,m)}$

$y \setminus x$		H	E	A	G	A	W	G	H	E	E
	87 ←	0	0	0	0	0	0	0	0	0	0
P	0	24 ↖	36 ←	18 ←	7	0	0	0	0	0	0
A	0	0	2	26 ↖	15	43	0	0	0	0	0
W	0	0	0	0	0	0	85 ←	1	0	0	0
H	0	0	0	0	0	0	0	12 ↖	73	0	0
E	0	0	0	0	0	0	0	1	8 ↖	65	0
A	0	0	0	0	0	0	0	0	1	↑ 21	0
E	0	0	0	0	0	0	0	0	0	0 ↖	86

23.

Match:

$P(x_i \diamond y_j \mid x, y) = \frac{f^M(i,j) b^M(i,j)}{f^E(n,m)}$

$y \setminus x$		H	E	A	G	A	W	G	H	E	E
	87 ←	0	0	0	0	0	0	0	0	0	0
P	0	24 ↖	36 ←	18 ←	7	0	0	0	0	0	0
A	0	0	2	26 ↖	15	43	0	0	0	0	0
W	0	0	0	0	0	0	85 ←	1	0	0	0
H	0	0	0	0	0	0	0	12 ↖	73	0	0
E	0	0	0	0	0	0	0	1	8 ↖	65	0
A	0	0	0	0	0	0	0	0	1	↑ 21	0
E	0	0	0	0	0	0	0	0	0	0 ↖	86

[illegible]

24.

X Insert:

$P(x_i \diamond - \mid x, y) = \frac{f^X(i,j) b^X(i,j)}{f^E(n,m)}$

$y \setminus x$		H	E	A	G	A	W	G	H	E	E
	0	\leftarrow 62	26	7	0	0	0	0	0	0	0
P	0	0	\nwarrow 22	\leftarrow 32	\leftarrow 36	0	0	0	0	0	0
A	0	0	0	2	28	42	0	0	0	0	0
W	0	0	0	0	0	0	\nwarrow 0	\leftarrow 72	0	0	0
H	0	0	0	0	0	0	0	0	\nwarrow 3	0	0
E	0	0	0	0	0	0	0	0	0	\nwarrow 0	0
A	0	0	0	0	0	0	0	0	0	\uparrow 1	0
E	0	0	0	0	0	0	0	0	0	0	\nwarrow 2

25.

Y Insert: $P(-\diamond y_j \mid x, y) = \frac{f^Y(i,j) b^Y(i,j)}{f^E(n,m)}$

$y \setminus x$		H	E	A	G	A	W	G	H	E	E
	$0 \leftarrow$	0	0	0	0	0	0	0	0	0	0
P	0	0	\nwarrow 0	\leftarrow 0	\nwarrow 0	0	0	0	0	0	0
A	0	0	0	0	0	\nwarrow 0	0	0	0	0	0
W	0	0	0	0	0	0	\nwarrow 0	\leftarrow 0	0	0	0
H	0	0	0	0	0	0	1	\nwarrow 0	0	0	0
E	0	0	0	0	0	0	0	0	12	\nwarrow 0	0
A	0	0	0	0	0	0	0	0	0	\uparrow 64	0
E	0	0	0	0	0	0	0	0	0	0	\nwarrow 10

25.

Y Insert: $P(-\diamond y_j \mid x, y) = \frac{f^Y(i,j) b^Y(i,j)}{f^E(n,m)}$

$y \setminus x$		H	E	A	G	A	W	G	H	E	E
	$0 \leftarrow$	0	0	0	0	0	0	0	0	0	0
P	0	0	\nwarrow 0	\leftarrow 0	\nwarrow 0	0	0	0	0	0	0
A	0	0	0	0	0	\nwarrow 0	0	0	0	0	0
W	0	0	0	0	0	0	\nwarrow 0	\leftarrow 0	0	0	0
H	0	0	0	0	0	0	1	\nwarrow 0	0	0	0
E	0	0	0	0	0	0	0	0	12	\nwarrow 0	0
A	0	0	0	0	0	0	0	0	0	\uparrow 64	0
E	0	0	0	0	0	0	0	0	0	0	\nwarrow 10

Overall accuracy alignment

$y \setminus x$		H	E	A	G	A	W	G	H	E	E
		87 ← 87	87	87	87	87	87	87	87	87	87
P	87	111	123 ← 123 ← 123	123	123	123	123	123	123	123	123
A	87	111	123	149	149	166	166	166	166	166	166
W	87	111	123	149	149	166	251 ← 251	251	251	251	251
H	87	111	123	149	149	166	251	263	324	324	324
E	87	111	123	149	149	166	251	263	324	389	389
A	87	111	123	149	149	166	251	263	324	389	389
E	87	111	123	149	149	166	251	263	324	389	475

$H \ E \ A \ G \ A \ W \ G \ H \ E \ - \ E$
 $- \ P \ - \ - \ A \ W \ - \ H \ E \ A \ E$

Conclusion

Probabilistic methods like HMMs may underperform (LC: w.r.t. to efficiency) the standard alignment methods when searching for optimal paths, but they have the advantage to provide complete alignment scores independent of any specific path.

ADDENDA 1

(see [Durbin et al, 1998], pag. 85)

We show how **for any pair HMM for global alignment** (as given in the beginning) we can derive **an equivalent FSA for obtaining the most probable alignment**.

This allows us to see a rigorous probabilistic-based interpretation for the terms used in sequence alignment (Ch. 2).

Note: To do the reverse, i.e. to go from a DP algorithm expressed as an FSA to a pair HMM is more complicated:

- a new parameter λ is needed, acting as a scaling factor for the scores $s(a, b)$, and
- for any given set of scores there may be constraints on the choice of η and τ .

The log likelihood of aligning sequences x and y

By putting in correspondence the **recurrence relations** in the **pair HMM**:

$$v^M(i, j) = p_{x_i y_j} \max \begin{cases} (1 - 2\delta - \tau) v^M(i - 1, j - 1) \\ (1 - \epsilon - \tau) v^X(i - 1, j - 1) \\ (1 - \epsilon - \tau) v^Y(i - 1, j - 1) \end{cases}$$

$$v^X(i, j) = q_{x_i} \max \begin{cases} \delta v^M(i - 1, j) \\ \epsilon v^X(i - 1, j) \end{cases} \quad v^Y(i, j) = q_{y_j} \max \begin{cases} \delta v^M(i, j - 1) \\ \epsilon v^Y(i, j - 1) \end{cases}$$

with the **probability of a pair of sequences** x and y in the **independence model**:

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

we obtain **log-odds scores for emissions and transitions** that correspond to the standard terms used in sequence alignment by DP:

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

$$d = -\log \frac{\delta}{1 - \eta} \quad \frac{1 - \epsilon - \tau}{1 - 2\delta - \tau} \quad e = -\log \frac{\epsilon}{1 - \eta}$$

The log-odds version of Viterbi algorithm

30.

- **Initialisation:** $V^M(0, 0) = -2 \log \eta$ and all other $V^X(i, 0) = V^Y(0, j) = -\infty$

- **Recursion relations:**

$$V^M(i, j) = s(x_i, y_j) + \max \begin{cases} V^M(i-1, j-1) \\ V^X(i-1, j-1) \\ V^Y(i-1, j-1) \end{cases} \quad \text{for } i \geq 1, j \geq 1$$

$$V^X(i, j) = \max \begin{cases} V^M(i-1, j) - d \\ V^X(i-1, j) - e \end{cases} \quad \text{for } i \geq 1, j \geq 0$$

$$V^Y(i, j) = \max \begin{cases} V^M(i, j-1) - d \\ V^Y(i, j-1) - e \end{cases} \quad \text{for } i \geq 0, j \geq 1$$

- **Termination:** $V = \max(V^M(n, m), V^X(n, m) + c, V^Y(n, m) + c)$

Notes:

1. $V^M(0, 0)$ is set to $2 \log \eta$ in order to compensate for the term η^2 in the final form of $P(x, y | R)$.
2. The constant $c = \log \frac{1-2\delta-\tau}{1-\epsilon-\tau}$ is needed to compensate for the adjustment made when defining d (see previous slide).

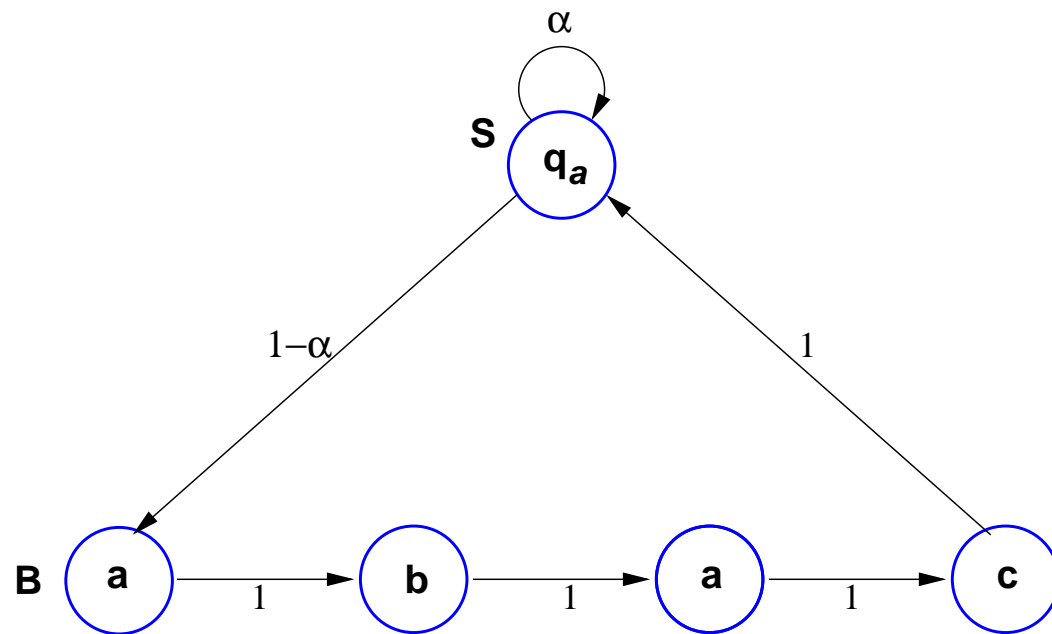
Notes

- The most probable path in the pair HMM is the optimal alignment.
- There is a close (but not exact) correspondence between the values of the standard DP matrix obtained by using the FSA (shown in the beginning) and the log-odds Viterbi values of the pair HMM, due to
 - the way $s(a, b)$, d and e were defined, and
 - the fact that only the initialisation and the termination relations differ in the two algorithms (see slides 2 and 12).
- If the exit transition probabilities to the M state are equal ($1 - \epsilon - \tau = 1 - 2\delta - \tau$), then $c = 0$ and $V = \max(V^M(n, m), V^X(n, m), V^Y(n, m))$.

ADDENDA 2

**Two drawbacks of FSA-based methods (like in Ch. 2)
that search for optimal alignment paths**

1. FSA-based methods do not (and even: they may be unable(!) to) account for the full probability $P(x, y)$ of aligning x and y .



Example: Suppose we have data generated by this simple (probabilistic) FSA.

If $\alpha^4 q_a q_b q_a q_c > 1 - \alpha$, then the Viterbi path will only use the state S , which doesn't give a full account of the data model.

Note that decreasing α so as to increase the probability of transition to B doesn't solve the problem.

2. The parameters of a FSA as shown below (that does local alignment) cannot be readily transformed into probabilities, since transition and emission probabilities are non null. [However, using two probabilistic models like the HMMs in the next slide may solve the problem, namely by doing Bayesian model comparison by computing log-odds ratios.]

