Pairwise HMMs and Sequence Alignment

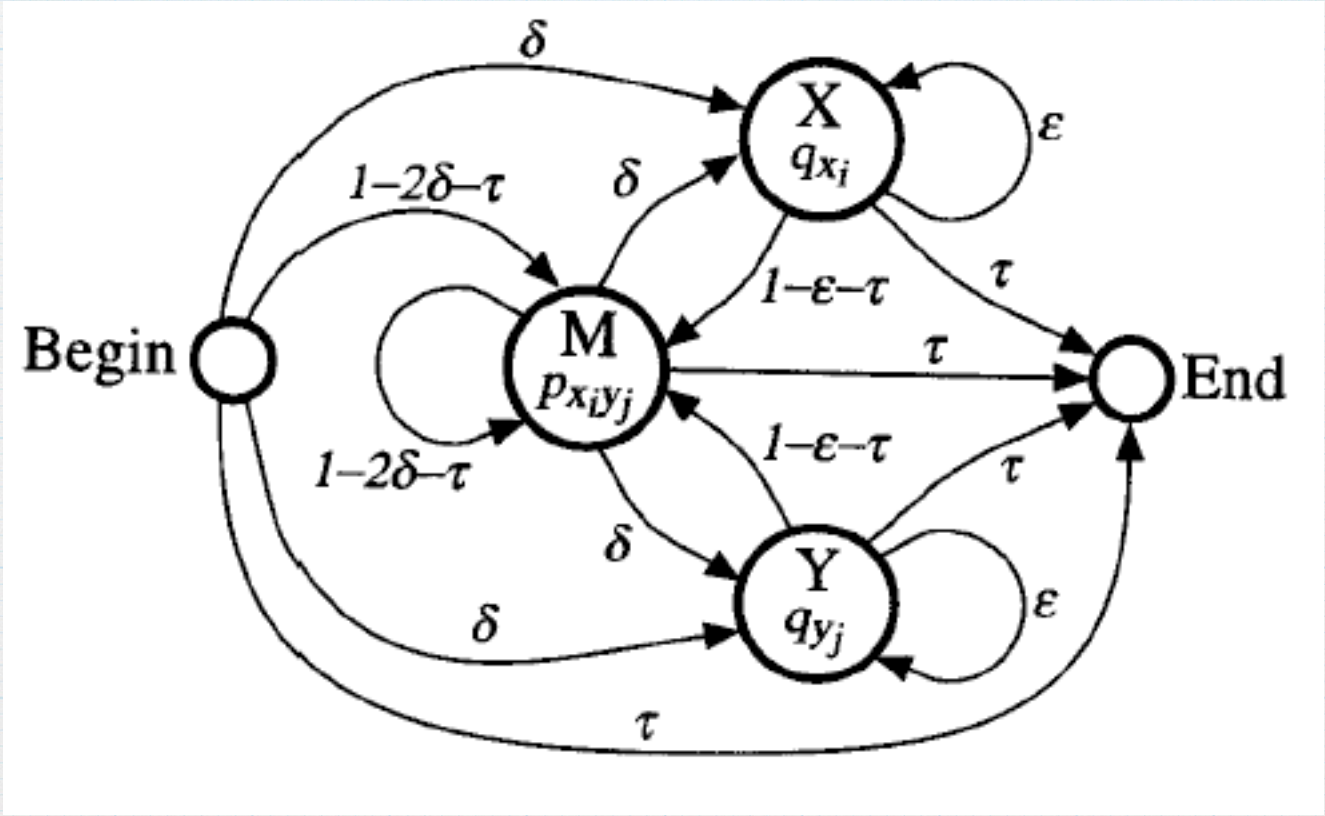
COMP 571 - Spring 2015

Luay Nakhleh, Rice University

Pair HMMs

* Match state M: emission probability p_{ab} for emitting an aligned pair *a:b*
* States X and Y: emission probabilities q_a for emitting symbol *a* against a gap
* Emits a pairwise alignment instead of a single sequence

Pair HMMs


Begin
M
$p_{x_i y_j}$
X
q_{x_i}
Y
q_{y_j}
End
δ
$1-2\delta-\tau$
δ
ε
$1-\varepsilon-\tau$
τ
τ
$1-2\delta-\tau$
$1-\varepsilon-\tau$
τ
δ
δ
ε
τ

Pair HMMs And Alignments

- Start in the *Begin* state and repeat the following two steps:

  (1) Pick the next state according to the transition probabilities leaving the current state

  (2) Pick a symbol pair to be added to the alignment according to the emission probabilities in the new state

Viterbi Algorithm For Pair HMMs

Initialization:

$$v^M(0,0) = 1.\ V^X(0,0) = v^Y(0,0) = 0, \text{ and } v^*(-1,j) = v^*(i,-1) = 0.$$

Recurrence: $i = 0, \ldots, n, j = 0, \ldots, m$:

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

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

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

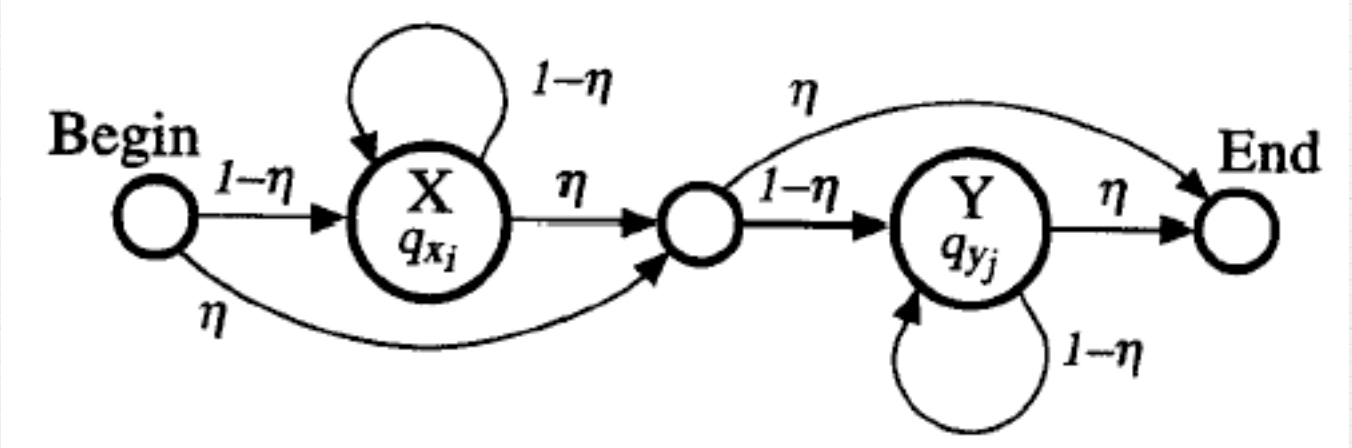
Termination:

$$v^E = \tau \max(v^M(n,m), v^X(n,m), v^Y(n,m)).$$

Pairwise Alignment Using HMMs

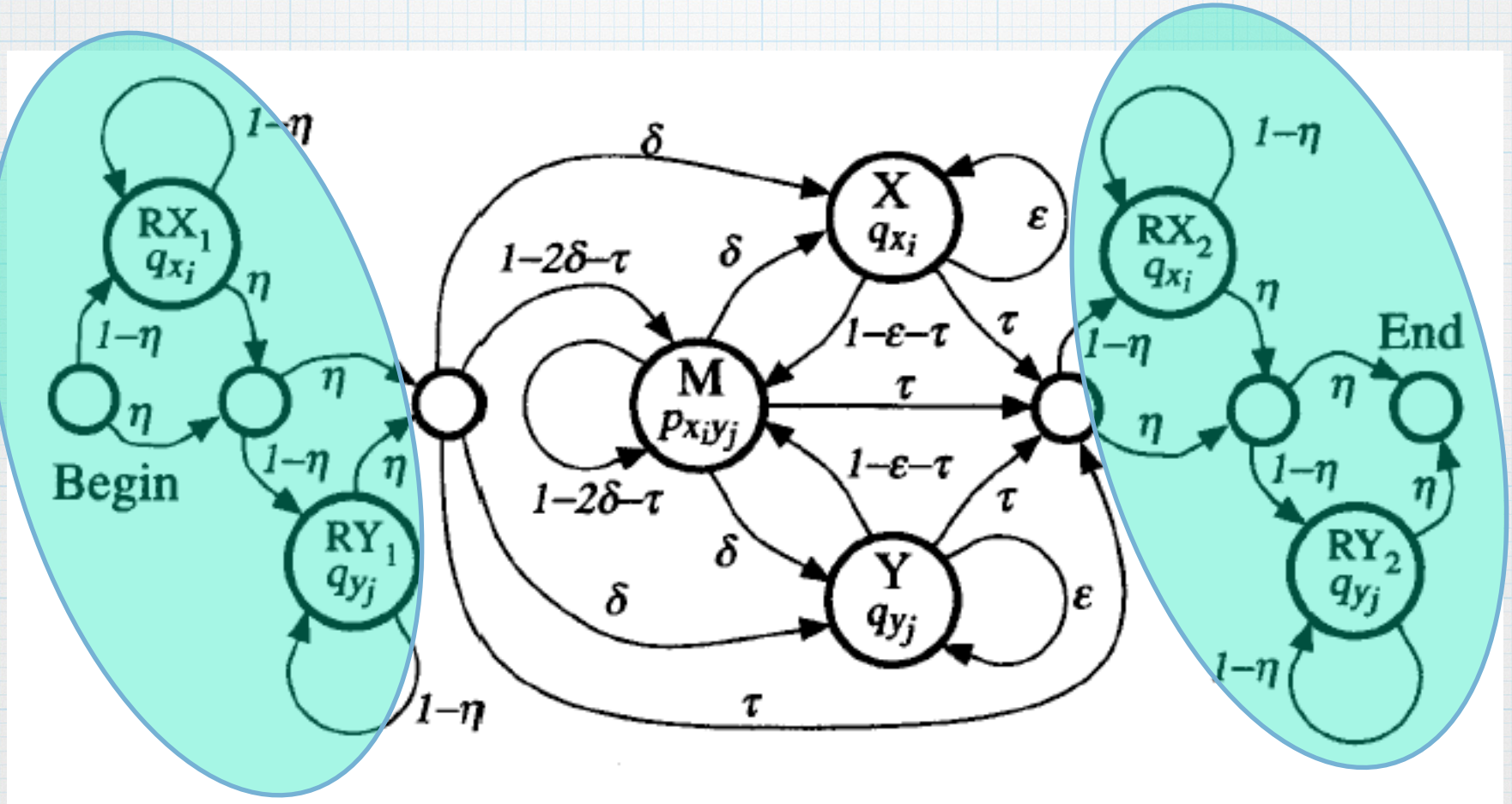
* To find the best alignment, we keep pointers and trace back as usual
* To get the alignment itself, we keep track of which residues are emitted at each step in the path during the traceback

A Pair HMM For Local Alignment



* We need an HMM "component" that models the "irrelevant" (low score) parts, which are not part of the local alignment

A Pair HMM For Local Alignment



Full Probability Of The Two Sequences

* A significant advantage of HMM approaches to alignment over standard DP approaches, is that 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

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

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:

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

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

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

Termination:

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

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:

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

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

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

Termination:

P(x,y) →

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

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 path corresponding to that alignment

How Correct Is The Alignment

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

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

Probability that the optimal scoring alignment is correct:

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

How Correct Is The Alignment

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

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

Probability that the optimal scoring alignment is correct:

Viterbi algorithm

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

How Correct Is The Alignment

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

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

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 $P(x_i \Diamond y_j | x, y)$

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

* We have

$$P(x, y, x_i \Diamond y_j) = P(x_{1\ldots i}, y_{1\ldots j}, x_i \Diamond y_j) P(x_{i+1\ldots n}, y_{j+1\ldots m} \mid x_i \Diamond y_j)$$

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

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

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

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