

Pair Hidden Markov Model

Three kinds of pair HMMs (PHMMs)

- PHMM for pairwise sequence alignment
 - BSA Chapter 4
- PHMM for the analysis (e.g. gene prediction) on two aligned sequences (i.e. the **pre-calculated pairwise alignments**)
 - Twinscan
- PHMM for simultaneously pairwise alignment and analysis
 - SLAM

Pairwise sequence alignment

Given two sequences over an alphabet (4 nucleotides or 20 amino acids):

ATGTTAT and ATCGTAC

By inserting '-'s and shifting two sequences, they can be aligned into a table of two rows with the same length:

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 Matrix: Example

	A	R	N	K
A	5	-2	-1	-1
R	-	7	-1	3
N	-	-	7	0
K	-	-	-	6

AKRANR

KAAANK
 $-1 + (-1) + (-2) + 5 + 7 + 3 = 11$

- Notice that although R and K are different amino acids, they have a positive score.
- Why? They are both positively charged amino acids → will not greatly change function of protein.

Scoring matrices

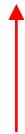
- Amino acid substitution matrices
 - PAM
 - BLOSUM
- DNA substitution matrices
 - DNA is less conserved than protein sequences
 - Less effective to compare coding regions at nucleotide level

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:

ATA__GC

ATATTGC



This is more
likely.

ATAG_GC

AT_GTGC



Normal scoring would
give the same score
for both alignments



This is less
likely.

Accounting for Gaps

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

- Score for a gap of length x is:

$$-(\rho + \sigma x)$$

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

gap opening penalty

ρ will be large relative to σ :

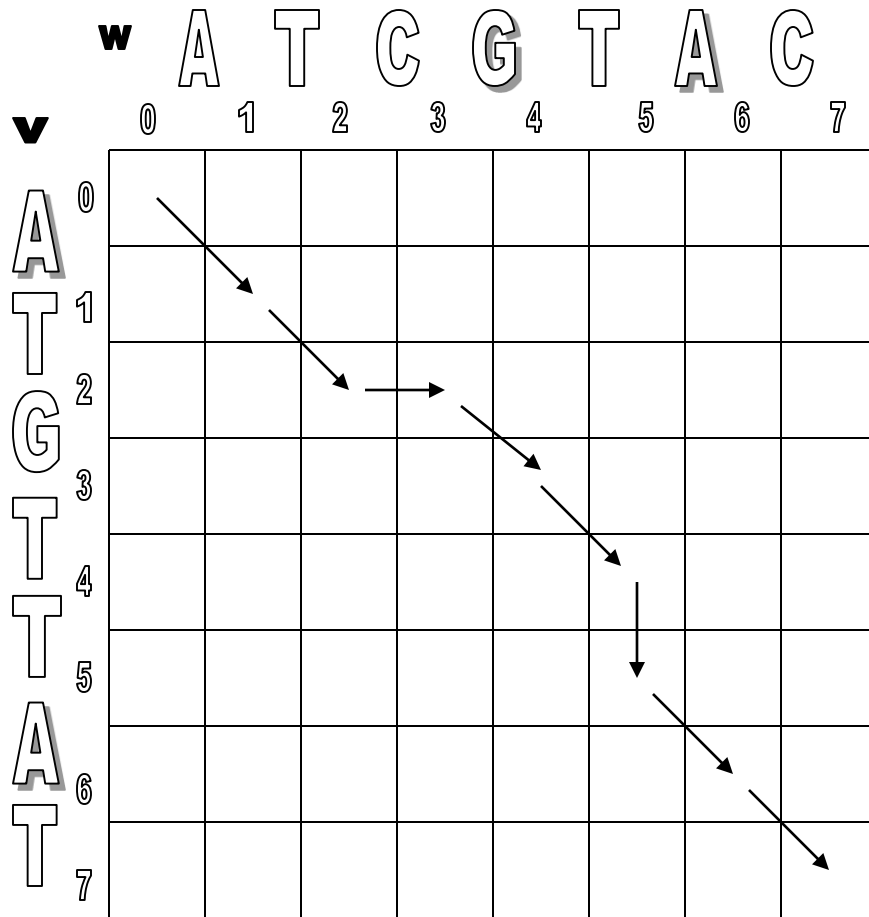
gap extension penalty

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

Affine Gap Penalties

- Gap penalties:
 - $-\rho - \sigma$ when there is 1 indel
 - $-\rho - 2\sigma$ when there are 2 indels
 - $-\rho - 3\sigma$ when there are 3 indels, etc.
 - $-\rho - x \cdot \sigma$ (-gap opening - x gap extensions)
- Somehow reduced penalties (as compared to naive scoring) are given to runs of horizontal and vertical edges

Alignment: a path in the Alignment Graph

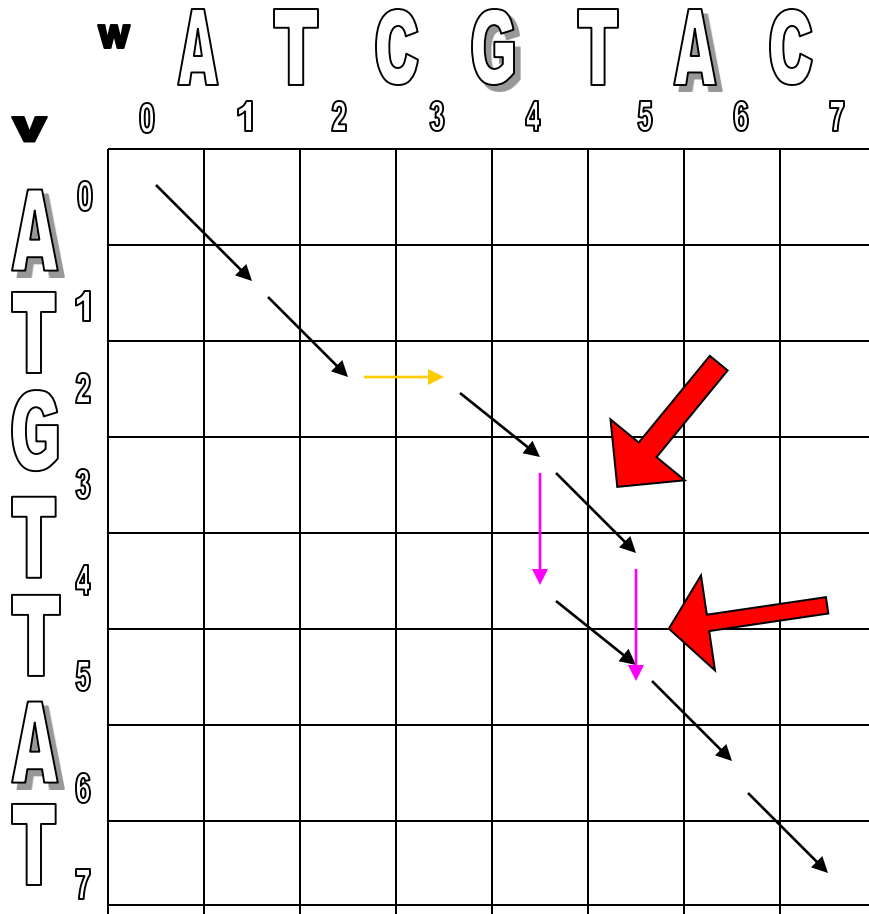


0	1	2	2	3	4	5	6	7
	A	T	-	G	T	T	A	T
	A	T	C	G	T	-	A	C
0	1	2	3	4	5	5	6	7

- Corresponding path -

(0,0) , (1,1) , (2,2), (2,3),
 (3,4), (4,5), (5,5), (6,6),
 (7,7)

Alignment as a Path in the Edit Graph



Old Alignment

012234567

x= AT_GTTAT

y= ATCGT_AC

012345567

New Alignment

012234567

x= AT_GTTAT

y= ATCG_TAC

012344567

Representing sequence alignment using pair HMM

HMM for sequence alignment, which incorporates affine gap scores.

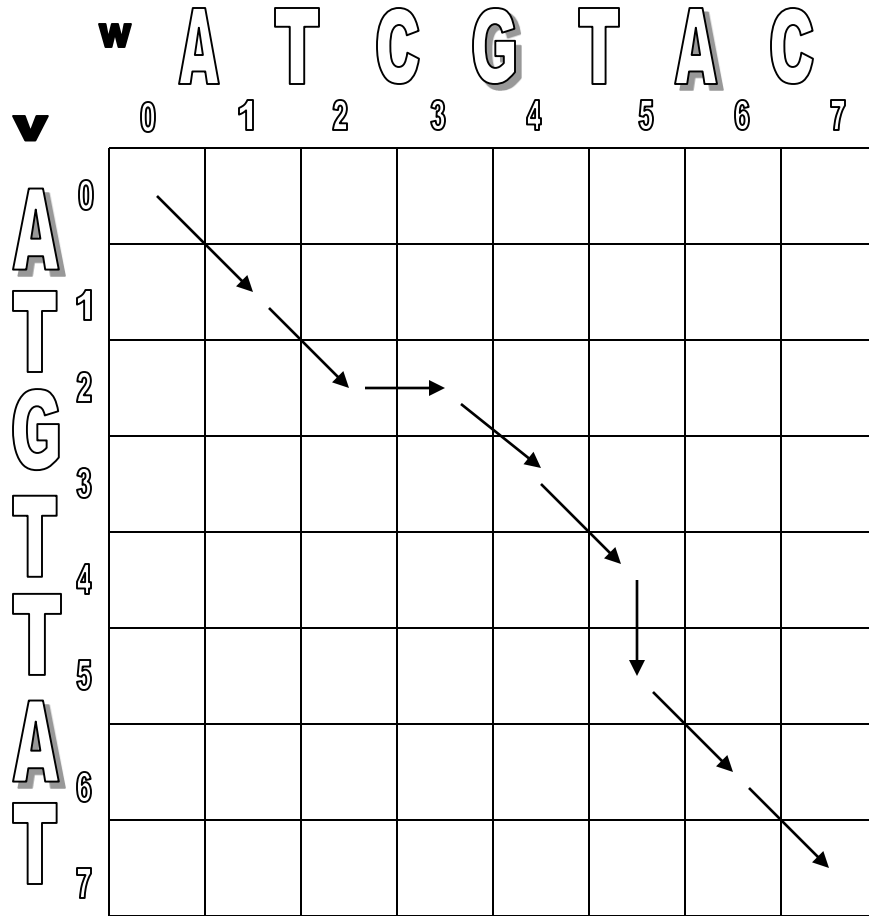
“Hidden” States

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

Observation Symbols

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

Alignment: a path \rightarrow a hidden state sequence

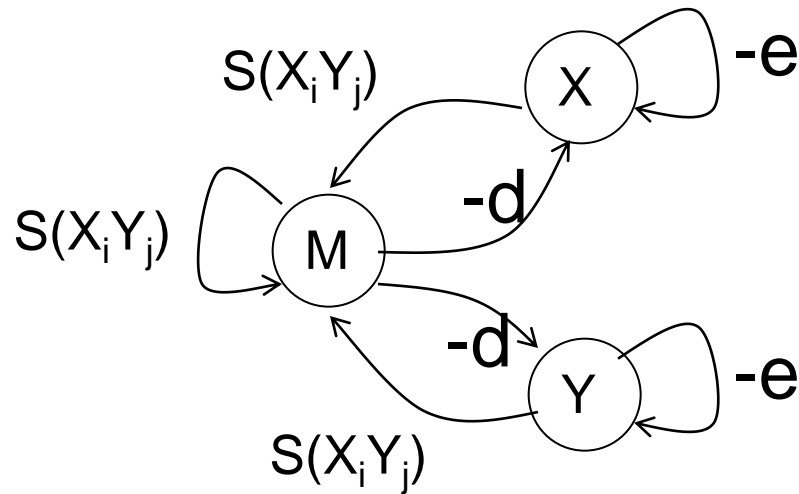


A T - G T T A T

A T C G T - A C

M M Y M M X M M

Representing sequence alignment using pair HMM

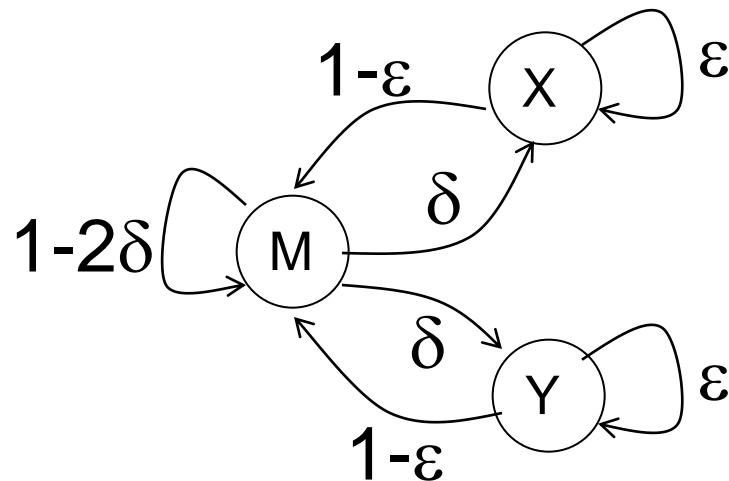


Finite State Machine:

M: (+1,+1)

X: (+1,0)

Y: (0,+1)



Emission probabilities:

M: P_{x_i, y_j}

X: q_{x_i}

Y: q_{y_j}

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).

◆ δ = probability for 1st gap

◆ ε = probability for extending gap.

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

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

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

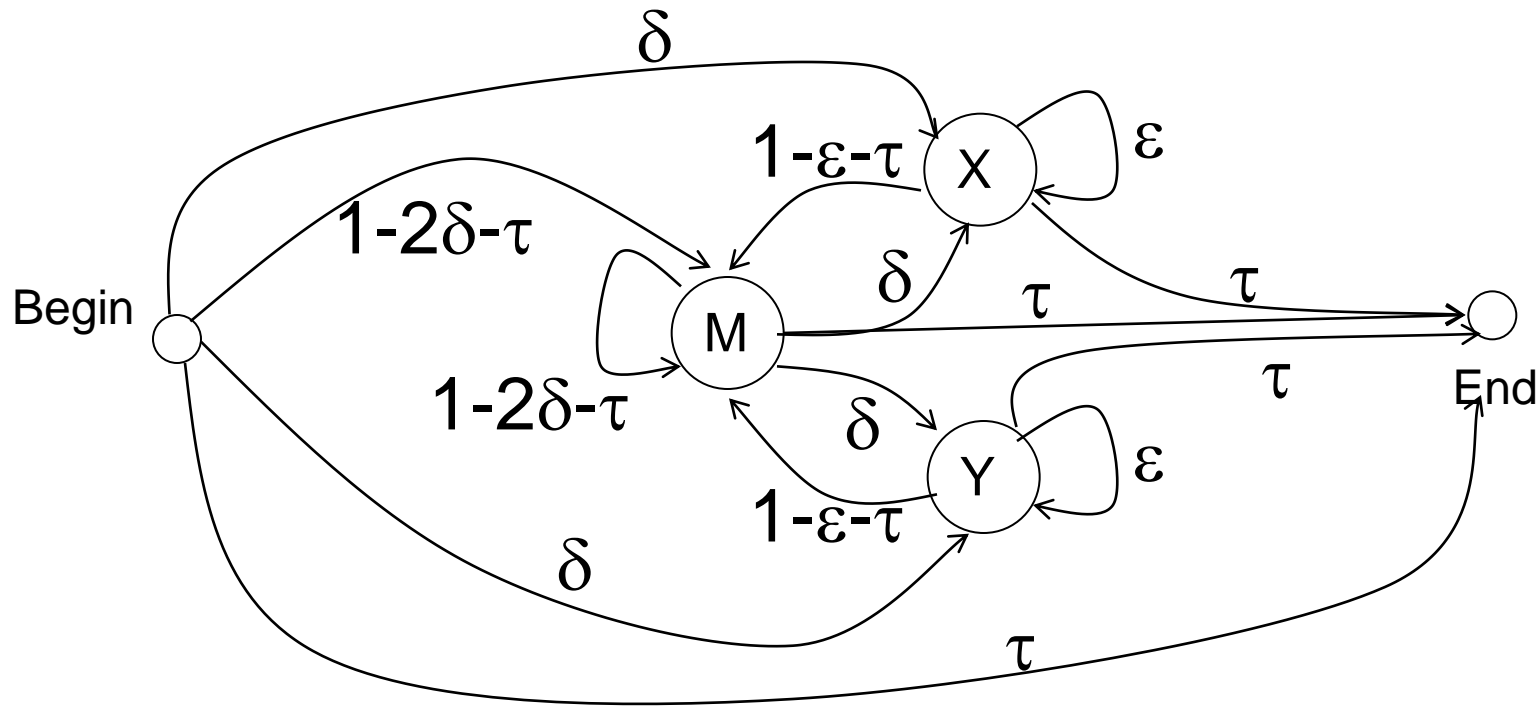
Different alignments of x and y may have different lengths. To get a coherent probabilistic model we need to define a probability distribution over sequences of different lengths.

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

The last transition in each alignment is to the END state, with probability τ

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

Representing sequence alignment using pair HMM



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 a random sequence alignment

The transition probabilities for the random model, with termination probability η :

(x is the start state)

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

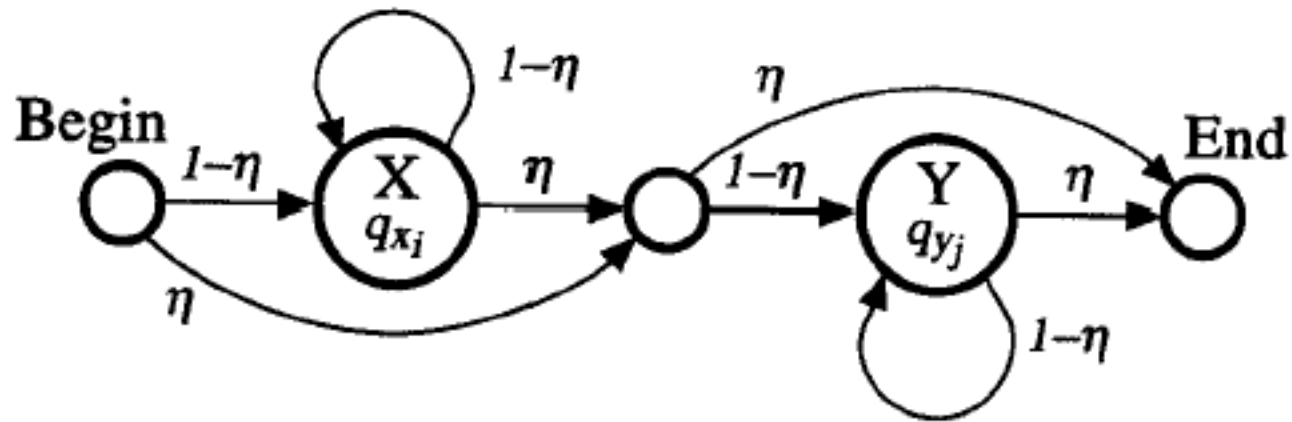
The emission probability for a is q_a .
Thus the probability of x (of length n) and y (of length m) being random is:

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

And the corresponding score is:

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

HMM for random sequence alignment



Markov Chains for “Random” and “Model”

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

“Model”

“Random”

	X	Y	END
X	$1-\eta$	η	
Y		$1-\eta$	η
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

(assuming that letters at insertions/deletions are selected by the random model)

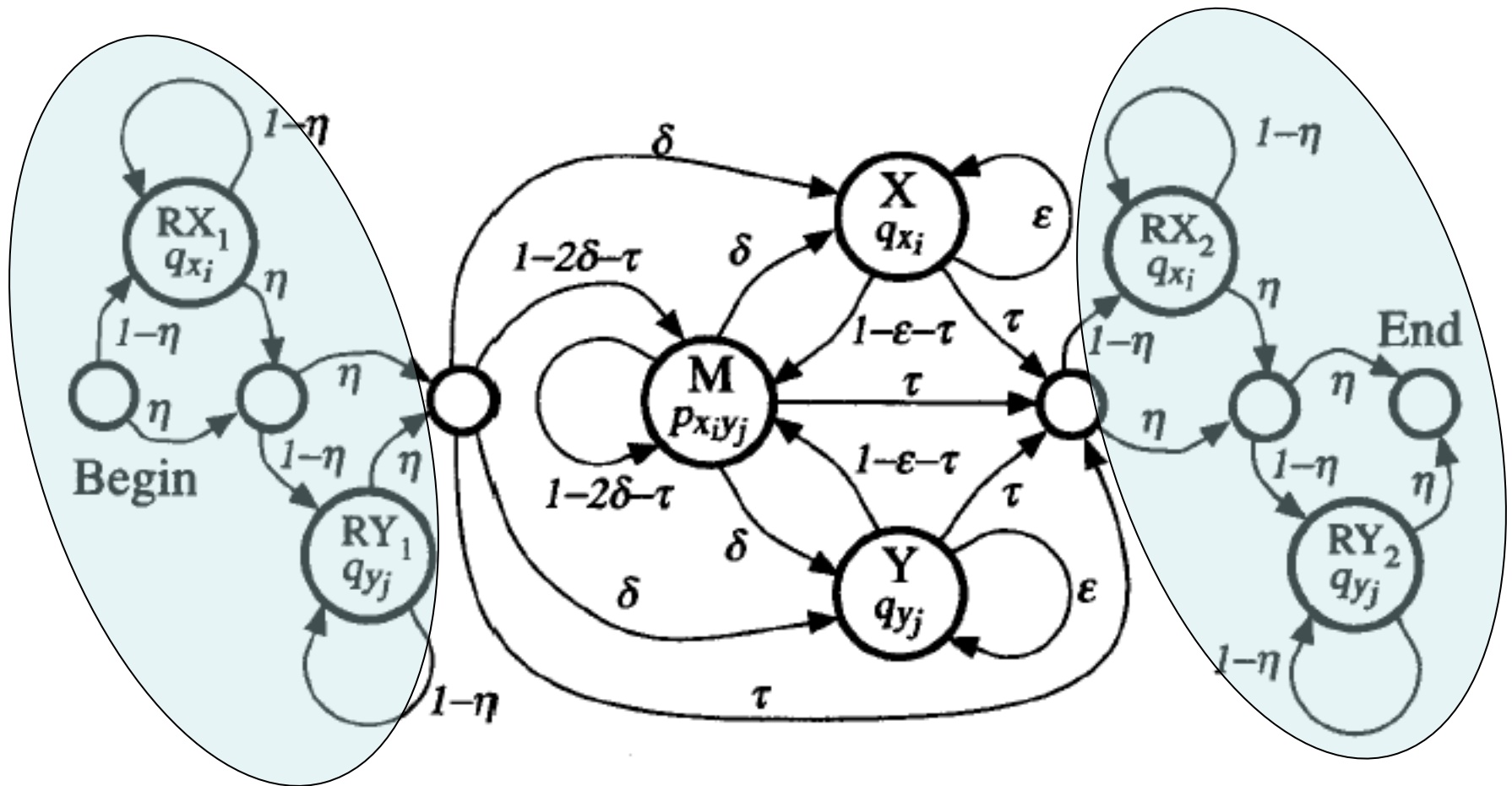
$$V^M[i, j] = \log \frac{p_{x_i y_j}}{q_{x_i} q_{y_j}} + \max \begin{pmatrix} \log(1 - 2\delta - \tau) + V^M[i - 1, j - 1] \\ \log(1 - \varepsilon - \tau) + V^X[i - 1, j - 1] \\ \log(1 - \varepsilon - \tau) + V^Y[i - 1, j - 1]_b \end{pmatrix} - 2\log(1 - \eta)$$

$$V^X[i, j] = \max \begin{pmatrix} \log \delta + V^M[i - 1, j] \\ \log \varepsilon + V^X[i - 1, j] \end{pmatrix} - \log(1 - \eta)$$

$$V^Y[i, j] = \max \begin{pmatrix} \log \delta + V^M[i, j - 1] \\ \log \varepsilon + V^Y[i, j - 1] \end{pmatrix} - \log(1 - \eta)$$

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

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

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

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

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

Probability that the optimal scoring alignment is correct:

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

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));

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

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

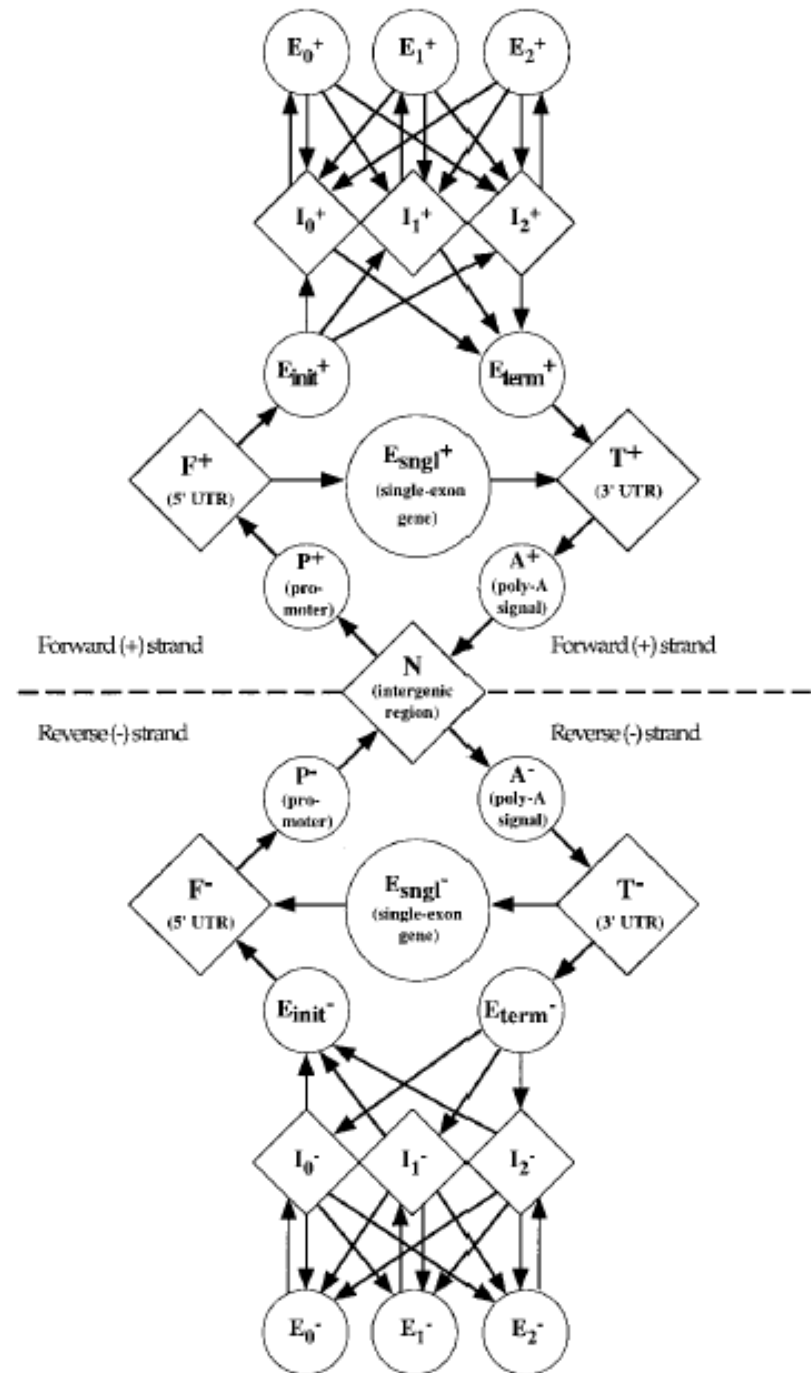
$$b^Y(i, j) = (1 - \varepsilon - \tau)p_{x_{i+1}y_{j+1}}b^M(i + 1, j + 1) + \varepsilon q_{y_{j+1}}b^Y(i + 1, j).$$

Pair HMM for gene finding (Twinscan)

- Twinscan is an augmented version of the GHMM used in Genscan.

Genscan Model

- Genscan considers the following:
 - Promoter signals
 - Polyadenylation signals
 - Splice signals
 - Probability of coding and non-coding DNA
 - Gene, exon and intron length



Twinscan Algorithm

1. Align the two sequences (eg. from human and mouse);
2. The similar hidden states as Genscan;
3. New “alphabet” for observation symbols: $4 \times 3 = 12$ symbols:

$$\Sigma = \{ A-, A:, A|, C-, C:, C|, G-, G:, G|, U-, U:, U| \}$$

Mark each base as gap (-), mismatch (:), match (|)

Twinscan Algorithm

Run Viterbi using emissions $e_k(b)$, where $b \in \{ A-, A:, A|, \dots, T| \}$

Note:

Emission distributions $e_k(b)$ estimated from the alignment of real gene pairs from human/mouse

$e_I(x|) < e_E(x|)$: matches favored in exons

$e_I(x-) > e_E(x-)$: gaps (and mismatches) favored in introns

Example

Human : **ACGGCGACUGUGCACGU**

Mouse : **ACUGUGAC GUGCACUU**

Align : **||:|:| |-||| |:|**

Input to Twinscan HMM:

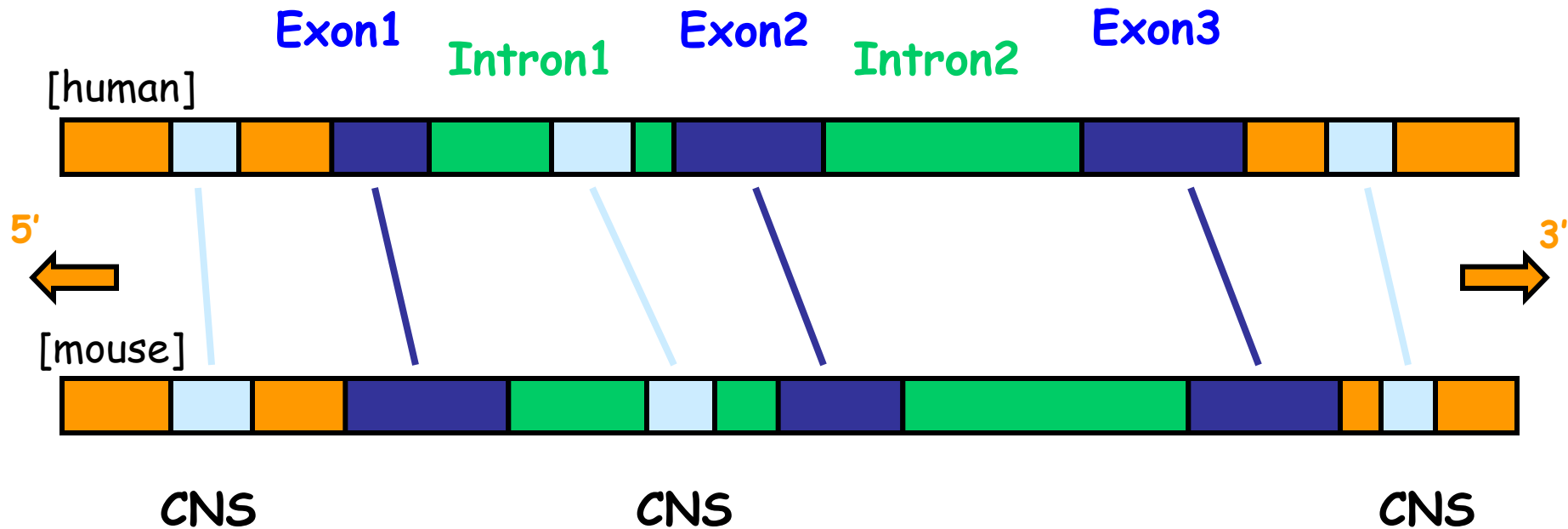
A| C| G: G| C: G| A| C| U- G| U| G| C| A| C| G: U|

Recall, $e_E(A|) > e_I(A|)$

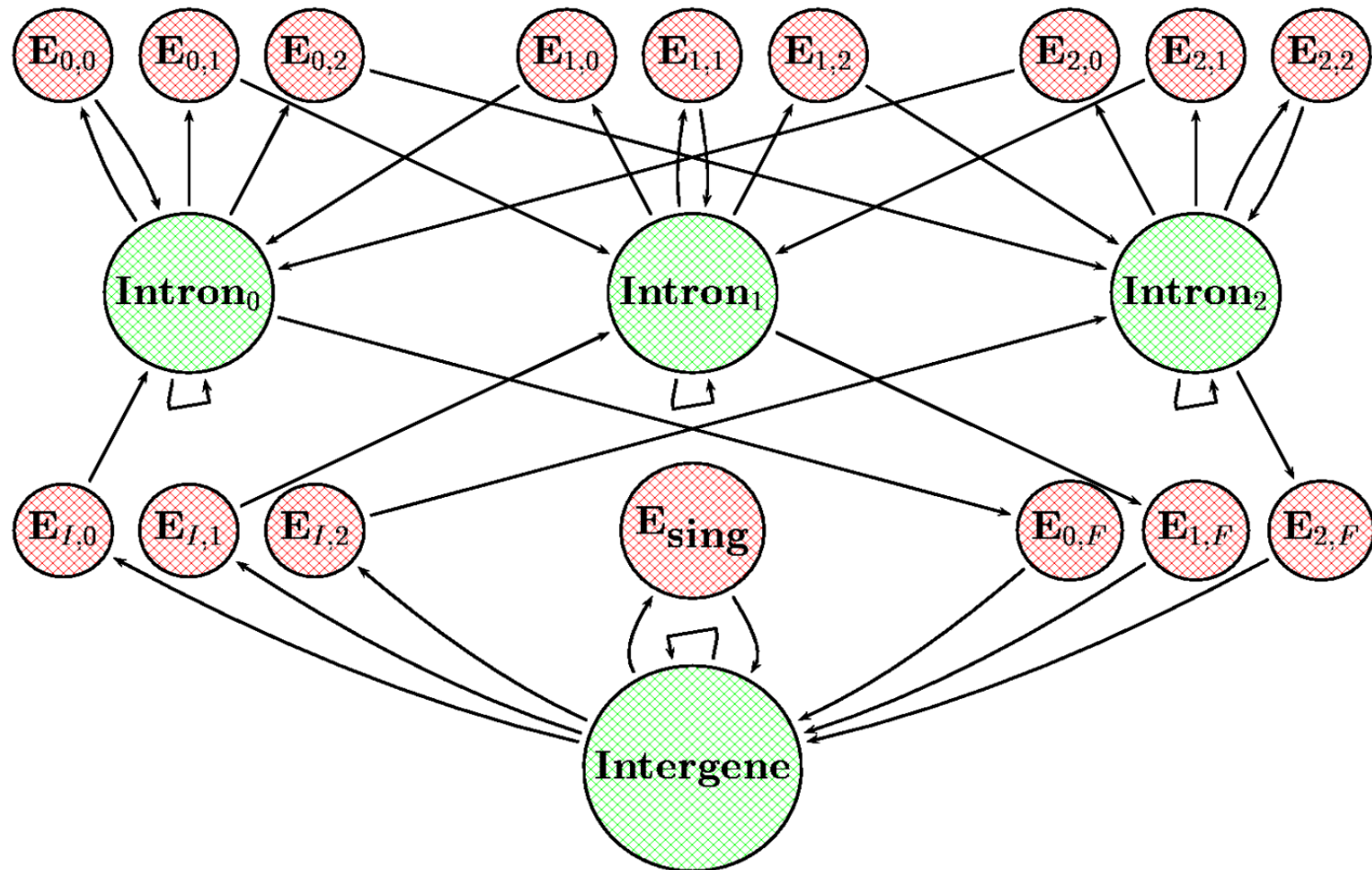
$e_E(A-) < e_I(A-)$

Likely exon

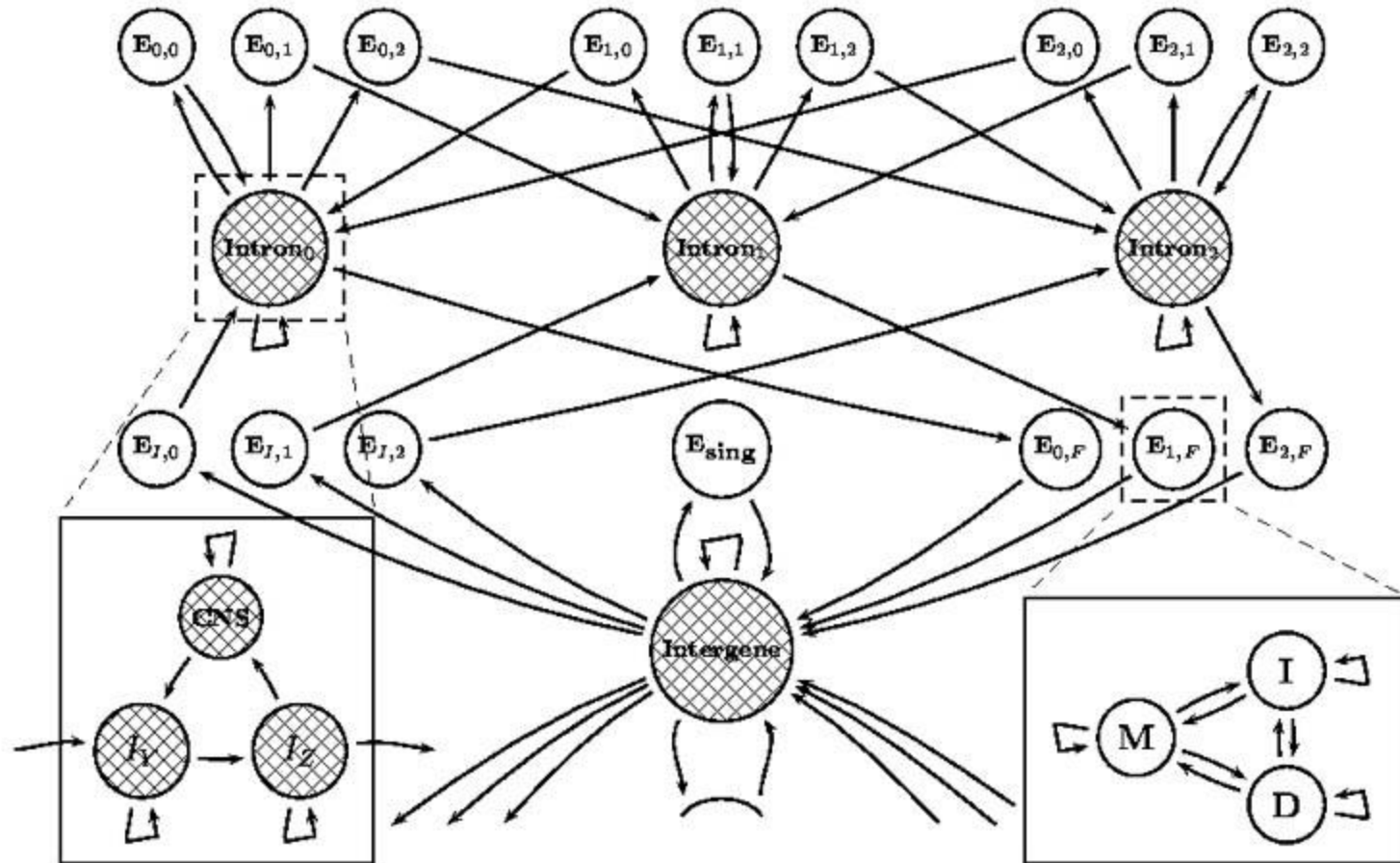
HMMs for simultaneous alignment and gene finding (SLAM)



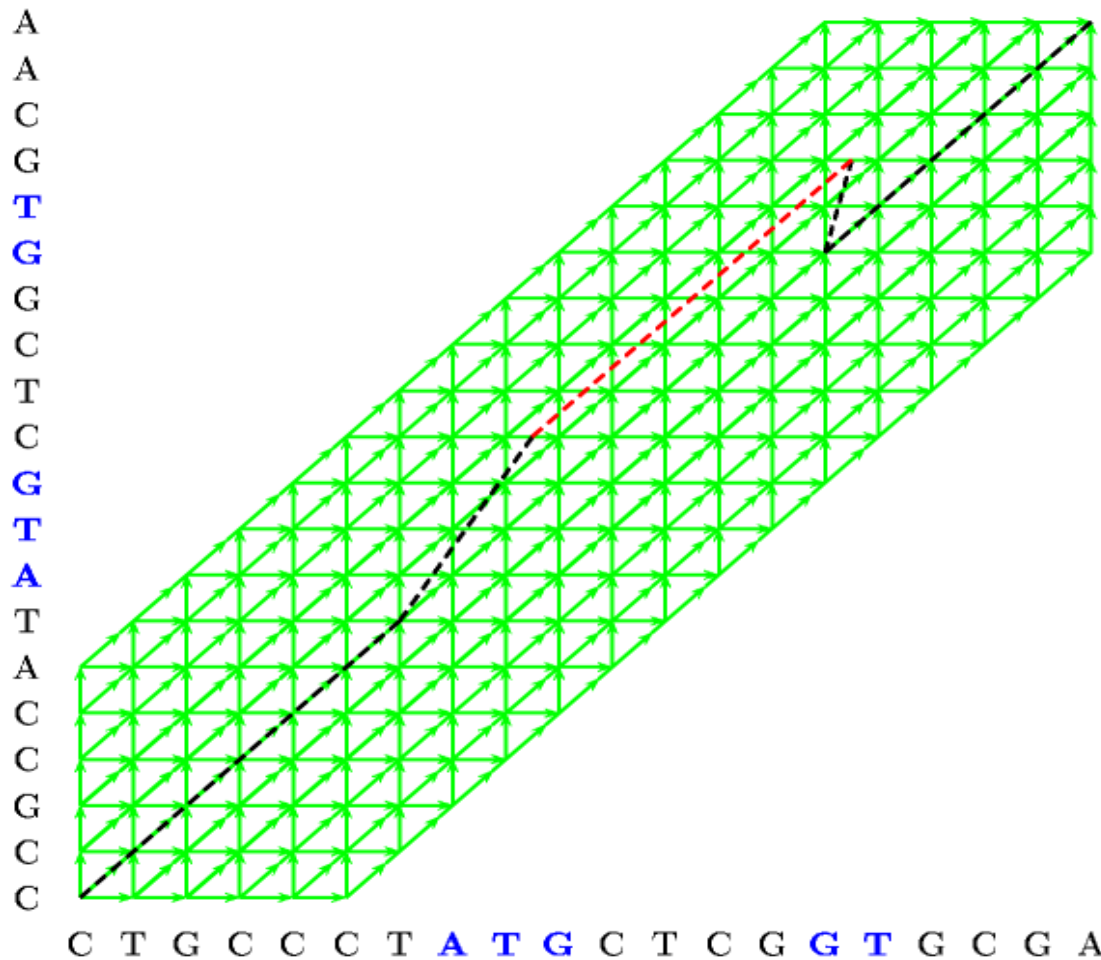
Generalized Pair HMMs



Generalized Pair HMMs (SLAM)



Gapped alignment



Measuring Performance

Test set	Nucleotide level			Exon level				
	SN	SP	AC	SN	SP	(SN+SP)/2	ME	WE
The ROSETTA set								
ROSETTA	0.935	0.978	0.949	0.833	0.829	0.831	0.048	0.047
SGP-1	0.940	0.960	0.940	0.700	0.760	0.730	0.120	0.040
SLAM	0.951	0.981	0.960	0.783	0.755	0.769	0.038	0.057
TWINSKAN.p	0.960	0.941	0.940	0.855	0.824	0.840	0.045	0.081
TWINSKAN	0.984	0.889	0.923	0.889	0.767	0.803	0.034	0.118
GENSCAN	0.975	0.908	0.929	0.817	0.770	0.793	0.057	0.107
HoxA								
SLAM	0.852	0.896	0.864	0.727	0.533	0.630	0.000	0.333
TWINSKAN.p	0.976	0.829	0.896	0.773	0.531	0.652	0.000	0.312
TWINSKAN	0.949	0.511	0.704	0.591	0.173	0.382	0.000	0.707
SGP-2	0.640	0.637	0.619	0.409	0.173	0.291	0.091	0.596
GENSCAN	0.932	0.687	0.796	0.545	0.235	0.390	0.000	0.569
Elastin								
SLAM	0.876	0.981	0.926	0.802	0.859	0.831	0.121	0.059
TWINSKAN.p	0.942	0.950	0.945	0.879	0.889	0.884	0.096	0.056
TWINSKAN	0.933	0.877	0.903	0.835	0.826	0.831	0.110	0.120
SGP-2	0.755	0.908	0.873	0.593	0.900	0.291	0.352	0.017
GENSCAN	0.947	0.766	0.852	0.835	0.731	0.783	0.121	0.231