Hidden Markov Models

Laboratory of Bioinformatics I
Module 2

Emidio Capriotti
http://biofold.org/

Biomolecules
Folding and
Disease

Department of Pharmacy and
Biotechnology (FaBiT)
University of Bologna

Formal Definition

A HMM is a stochastic generator of sequences characterized by:

- *N* states
- A set of transition probabilities between two states $\{\boldsymbol{a_{kj}}\}$

  $\boldsymbol{a_{kj}} = \mathrm{P}(\ \pi\ (i) = \boldsymbol{j}\ |\ \pi\ (i\text{-}1) = \boldsymbol{k}\ )$
- A set of starting probabilities $\{\boldsymbol{a_{0k}}\}$

  $\boldsymbol{a_{0k}} = \mathrm{P}(\ \pi\ (1) = \boldsymbol{k}\ )$
- A set of ending probabilities $\{\boldsymbol{a_{k0}}\}$

  $\boldsymbol{a_{k0}} = \mathrm{P}(\ \pi\ (i) = \mathrm{END}\ |\ \pi\ (i\text{-}1) = \boldsymbol{k}\ )$
- An alphabet ***C*** with *M* characters.
- A set of emission probabilities for each state $\{\boldsymbol{e_k\,(c)}\}$

  $\boldsymbol{e_k\,(c)} = \mathrm{P}(\ s^{\ i} = \boldsymbol{c}\ |\ \pi\ (i) = \boldsymbol{k}\ )$
- Constraints:

  $\Sigma_k\ \ \boldsymbol{a_{0\,k}} = 1$

  $\boldsymbol{a_{k0}} + \Sigma_j\ \ \boldsymbol{a_{k\,j}} = 1 \qquad \forall\ \boldsymbol{k}$

  $\Sigma_{c \in C}\ \ \boldsymbol{e_k\,(c)} = 1 \qquad \forall\ \boldsymbol{k}$

s: sequence, $\boldsymbol{\pi}$: path through the states

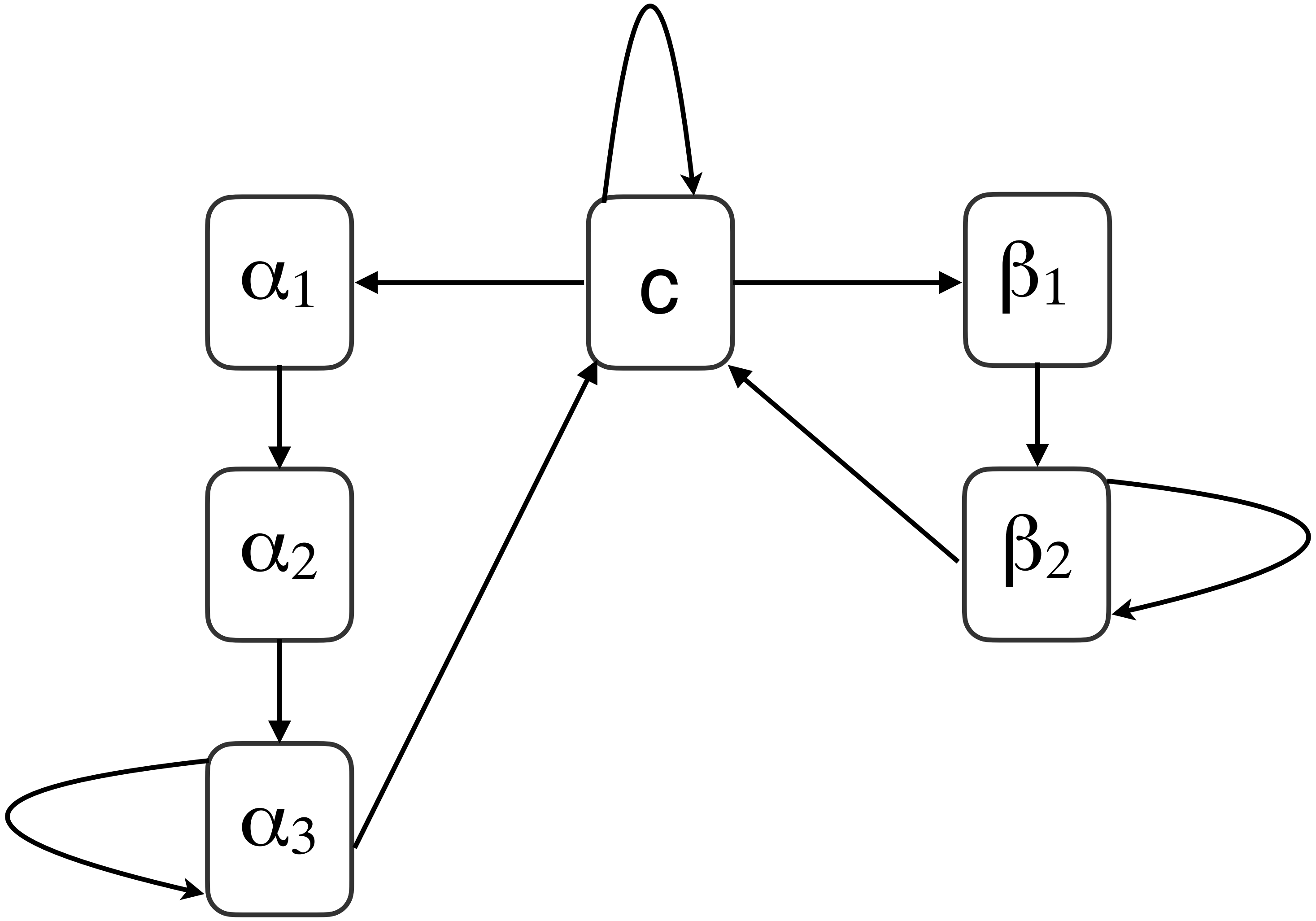
Hidden Markov Models

HMMs interpret an observable sequence (residue sequence or DNA/RNA sequence) as «generated» by an underlying (hidden) process.

Transition topology and probabilities define a global grammar

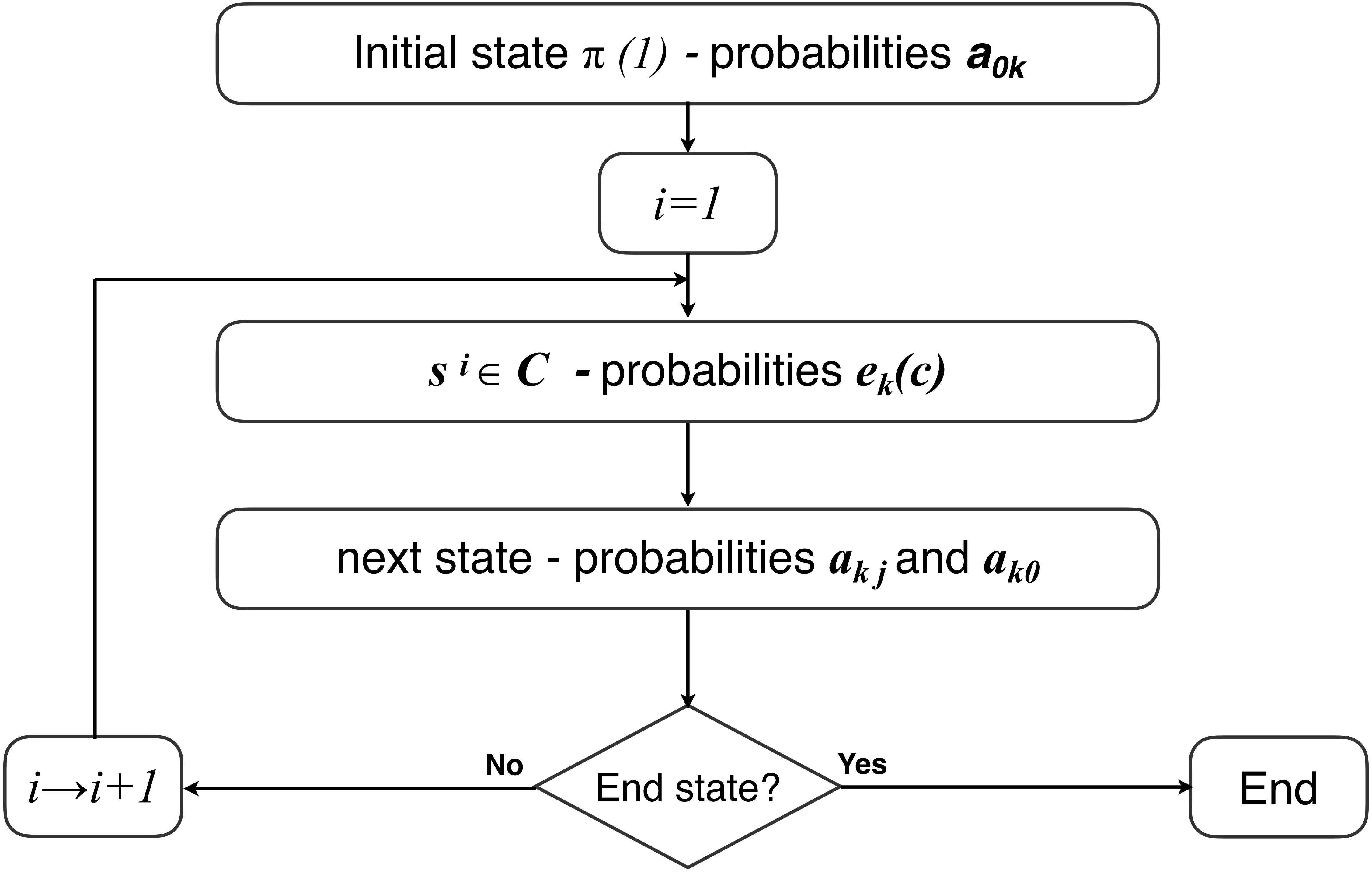
Emission probabilities cast the propensity of observable symbols in each state

Secondary Structure



S	**A**	**L**	**K**	**M**	**N**	**Y**	**T**	**R**	**E**	**I**	**M**	**V**	**A**	**S**	**N**	**Q**	s: sequence
c	α_1	α_2	α_3	α_3	α_3	α_3	c	c	c	c	β_1	β_2	β_2	β_2	c	c	π: path
c	α	α	α	α	α	α	c	c	c	c	β	β	β	β	c	c	$Y(\pi)$: labels

Generating HMM Sequence



Initial state π (1) - probabilities a_{0k}

$i=1$

$s^{i} \in C$ - probabilities $e_k(c)$

next state - probabilities a_{kj} and a_{k0}

End state?

No

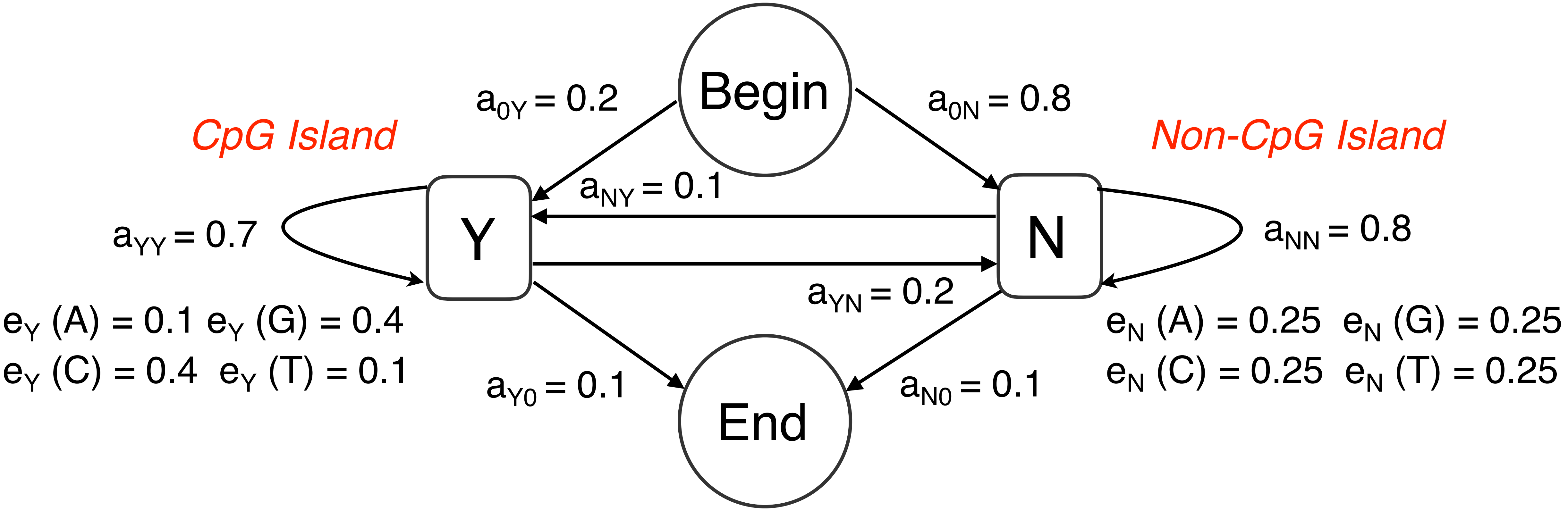
Yes

$i \rightarrow i+1$

End

CpG Islands Model

Probability of a sequence s with a given path π



Begin

$a_{0Y} = 0.2$

$a_{0N} = 0.8$

CpG Island

Non-CpG Island

Y

N

$a_{NY} = 0.1$

$a_{YY} = 0.7$

$a_{NN} = 0.8$

$a_{YN} = 0.2$

e_Y (A) = 0.1 e_Y (G) = 0.4

e_Y (C) = 0.4 e_Y (T) = 0.1

e_N (A) = 0.25 e_N (G) = 0.25

e_N (C) = 0.25 e_N (T) = 0.25

$a_{Y0} = 0.1$

$a_{N0} = 0.1$

End

s : **A G C G C G T A A T C T G**

π : **Y Y Y Y Y Y Y N N N N N N**

Emission: 0.1 × 0.4 ×0.4 × 0.4 ×0.4 ×0.4 ×0.1×0.25×0.25×0.25×0.25×0.25×0.25

Transition: 0.2 × 0.7 × 0.7 × 0.7 × 0.7 × 0.7 × 0.7 × 0.2×0.8 × 0.8 ×0.8×0.8× 0.8 × 0.1

Joint Probability

Calculate the joint probability of the sequence (s) ad the path (π) given the model (M)

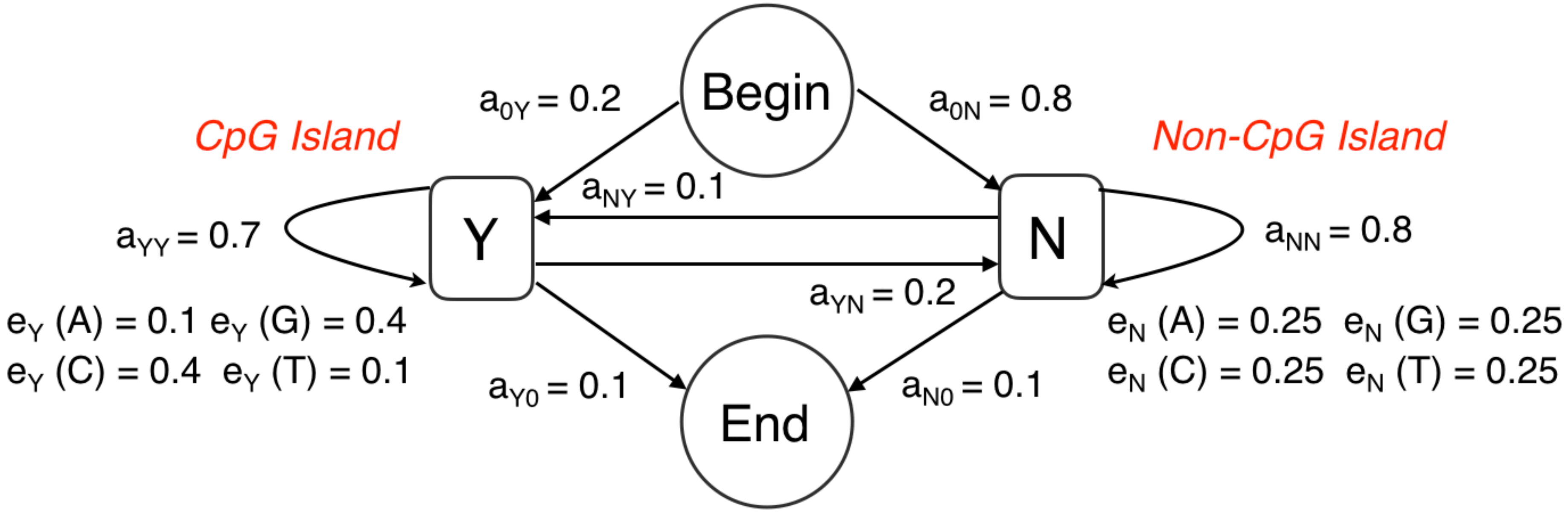
$$P(s,\pi \mid M) = P(s \mid \pi, M) \cdot P(\pi \mid M)$$

$$P(\pi \mid M) = a_{0\pi(1)} \cdot \prod_{i=2}^{T} a_{\pi(i-1)\pi(i)} \cdot a_{\pi(T)0}$$

$$P(s \mid \pi, M) = \prod_{i=1}^{T} e_{\pi(i)}(s^{i})$$

$$P(s,\pi \mid M) = a_{\pi(T)0} \cdot \prod_{i=1}^{T} a_{\pi(i-1)\pi(i)} \cdot e_{\pi(i)}(s^{i})$$

Sequence Probability



$$P (s \mid M) = \Sigma_\pi P(s, \pi \mid M)$$

2^{13} different paths
Summing over all the path will give the probability of having the sequence

s :	A	G	C	G	C	G	T	A	A	T	C	T	G
π_1:	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
π_2:	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N
π_3:	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y
π_4:	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N
π_5:	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y

Forward Algorithm

On the basis of preceding observations the computation of P(s I M) can be decomposed in simplest problems

For each state k and each position i in the sequence, we compute:

$$F_k(i) = \mathrm{P}(\, s^1 s^2 s^3 \ldots\ldots s^{\,i},\ \pi\,(i) = k \mid M)$$

Initialization: $F_{BEGIN}\,(0) = 1 \qquad F_i\,(0) = 0 \qquad \forall\ \ i \neq \mathrm{BEGIN}$

Recurrence: $F_l(\, i{+}1) = \mathrm{P}(\, s^1 s^2 \ldots s^{\,i} s^{\,i+1},\ \pi\,(i + 1) = l\,)\ =$

$$= \Sigma_k\ \mathrm{P}(\, s^1 s^2\ \ldots s^{\,i},\ \pi\,(i) = k\,) \cdot \mathrm{a}_{\,k\,l} \cdot \mathrm{e}_{\,l}(\, s^{\,i+1}\,) =$$

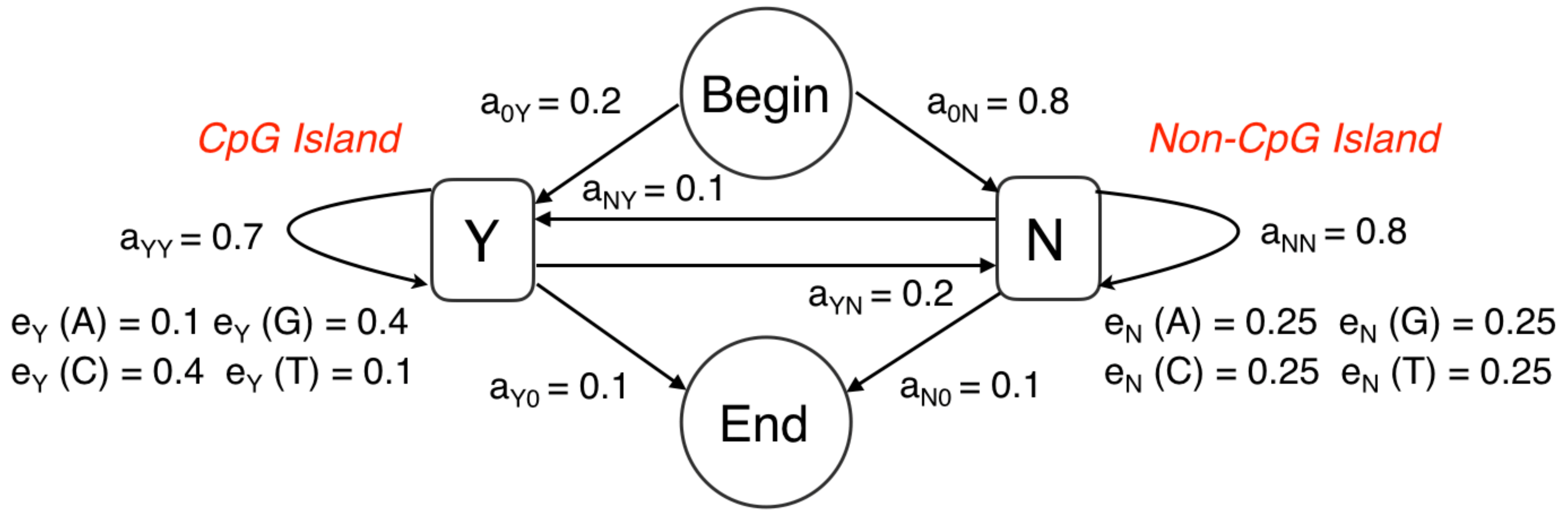
$$= \mathrm{e}_{\,l}(\, s^{\,i+1}\,) \cdot \Sigma_k F_k\ (\,i\,) \cdot \mathrm{a}_{\,k\,l}$$

Termination: $\mathrm{P}(\, s\,) = \mathrm{P}(\, s^1 s^2 s^3 \ldots\ldots s^{\,T},\ \pi\,(T + 1) = END\,) =$

$$= \Sigma_k\ \mathrm{P}(\, s^1 s^2\ \ldots s^{\,T},\ \pi\,(T) = k\,) \cdot \mathrm{a}_{\,k0}$$

$$=\ \Sigma_k F_k\,(\,T\,) \cdot \mathrm{a}_{\,k\,0}$$

Forward Algorithm: Example



S: ATGCG *Initialization*: $F_{BEGIN}(0) = 1 \; F_i(0) = 0 \; \forall \; i \neq \text{BEGIN}$

Recurrence: $F_l(\text{i}+1) = \text{e}_l(s^{\text{i}}) \cdot \Sigma_k F_k(\text{i}) \cdot \text{a}_{kl}$ *Termination*: $\text{P}(s) = \Sigma_k F_k(T) \cdot \text{a}_{k0}$

	-	A	T	G	C	G	-
Begin	1	0	0	0	0	0	0
Y	0	0.2x0.1	2e-2x0.7x0.1+ +0.2x0.1x0.1= =3.4e-3	3.4e-3x0.7x0.4+ +4.1e-2x0.1x0.4= =2.59e-3	2.59e-3x0.7x0.4+ +8.37e-3x0.1x0.4= =1.06056e-3	1.06056e-3x0.7x0.4+ +1.8036e-3x0.1x0.4= =3.691008e-4	
N	0	0.8x0.25	2e-2x0.2x0.25+ +0.2x0.8x0.25= =4.1e-2	3.4e-3x0.2x0.25+ +4.1e-2x0.8x0.25= =8.37e-3	2.592e-3x0.2x0.25+ +8.37e-3x0.8x0.25= =1.8036e-3	1.06056e-3x0.2x0.25+ +1.8036e-3x0.8x0.25= =4.13748e-4	
End	0	0	0	0	0	0	3.69e-4x0.1+ +4.13e-4x0.1= =7.82e-5

Backward Algorithm

Similar to the Forward algorithm: it computes P(s I M), reconstructing the sequence from the end

For each state k and each position i in the sequence, we compute:

$$B_k(i) = \mathrm{P}(\, s^{\,i+1} s^{\,i+2} s^{\,i+3} \ldots\ldots s^{\,T} \mid \pi\,(i) = k\,)$$

Initialization: $B_k\,(T) = \mathrm{P}(\pi\,(T{+}1) = \mathrm{END} \mid \pi\,(T) = k\,) = \mathrm{a}_{k0}$

Recurrence: $B_l(\,i\text{-}1) = \mathrm{P}(s^{\,i} s^{\,i+1} \ldots s^{\,T} \mid \pi\,(i - 1) = l\,) =$

$$= \Sigma_k\, \mathrm{P}(s^{\,i+1} s^{\,i+2} \ldots s^{\,T} \mid \pi\,(i) = k) \cdot \mathrm{a}_{\,l\,k} \cdot \mathrm{e}_{\,k}(s^{\,i}) =$$

$$= \Sigma_k B_k\;(\,i\,) \cdot \mathrm{e}_{\,k}(\,s^{\,i}\,) \cdot \mathrm{a}_{\,l\,k}$$

Termination: $\mathrm{P}(\,s\,) = \mathrm{P}(\,s^1 s^2 s^3 \ldots\ldots s^{\,T} \mid \pi\,(0) = BEGIN\,) =$

$$= \Sigma_k\, \mathrm{P}(\,s^2 \ldots s^{\,T} \mid \pi\,(1) = k\,) \cdot \mathrm{a}_{\,0\,k} \cdot \mathrm{e}_{\,k}(\,s^{\,1}\,) =$$

$$= \Sigma_k\, B_k\,(\,1\,) \cdot \mathrm{a}_{\,0k} \cdot \mathrm{e}_{\,k}(\,s^{\,1}\,)$$

Computational Complexity

Naïf method

$$\mathrm{P}(s \mid M) = \Sigma_{\pi} \mathrm{P}(s, \pi \mid M)$$

There are N^T possible paths.

Each path requires about $2 \cdot T$ operations.

The time for the computation is $O(T \cdot N^T)$

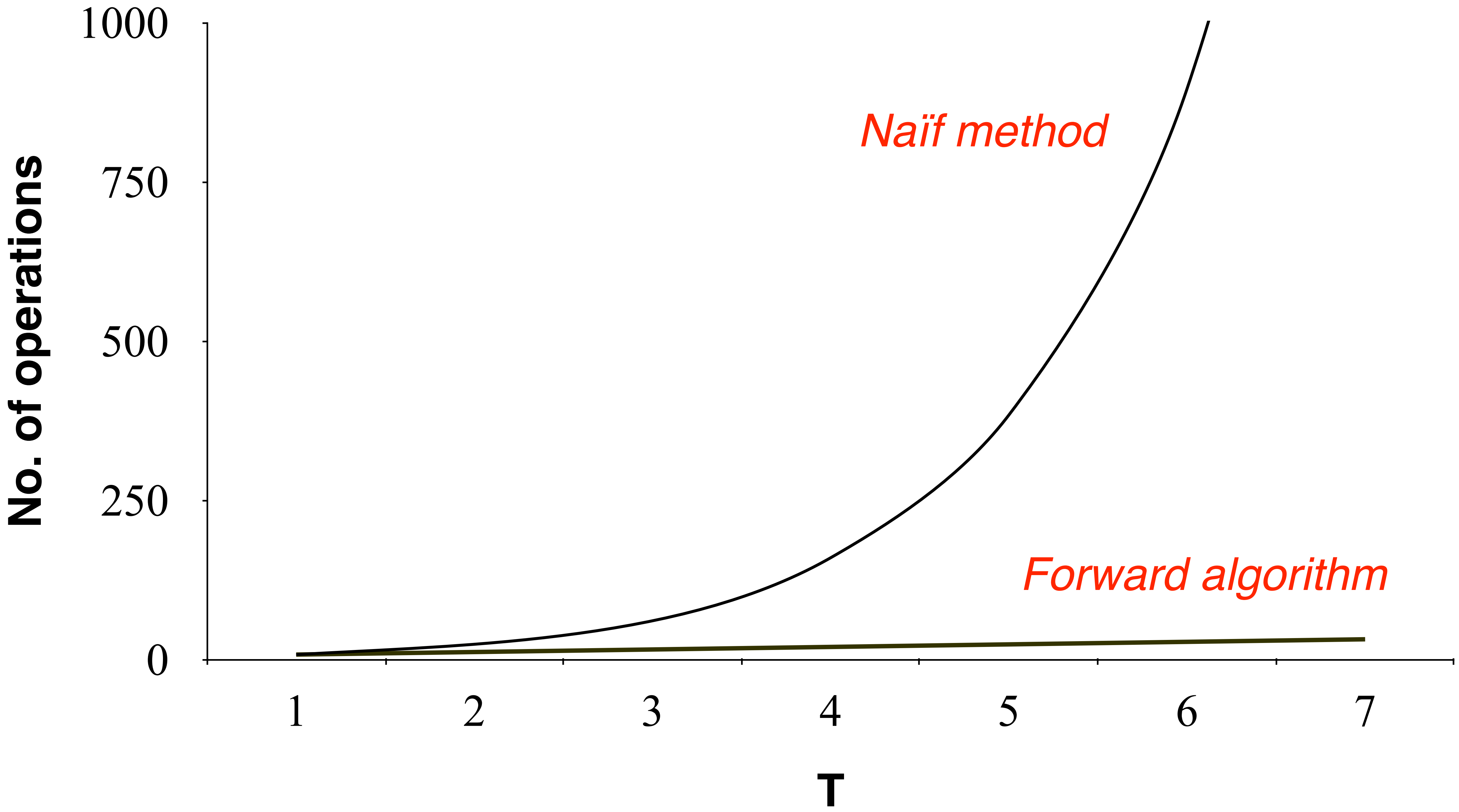
Forward Algorithm

T positions, ***N*** values for each position

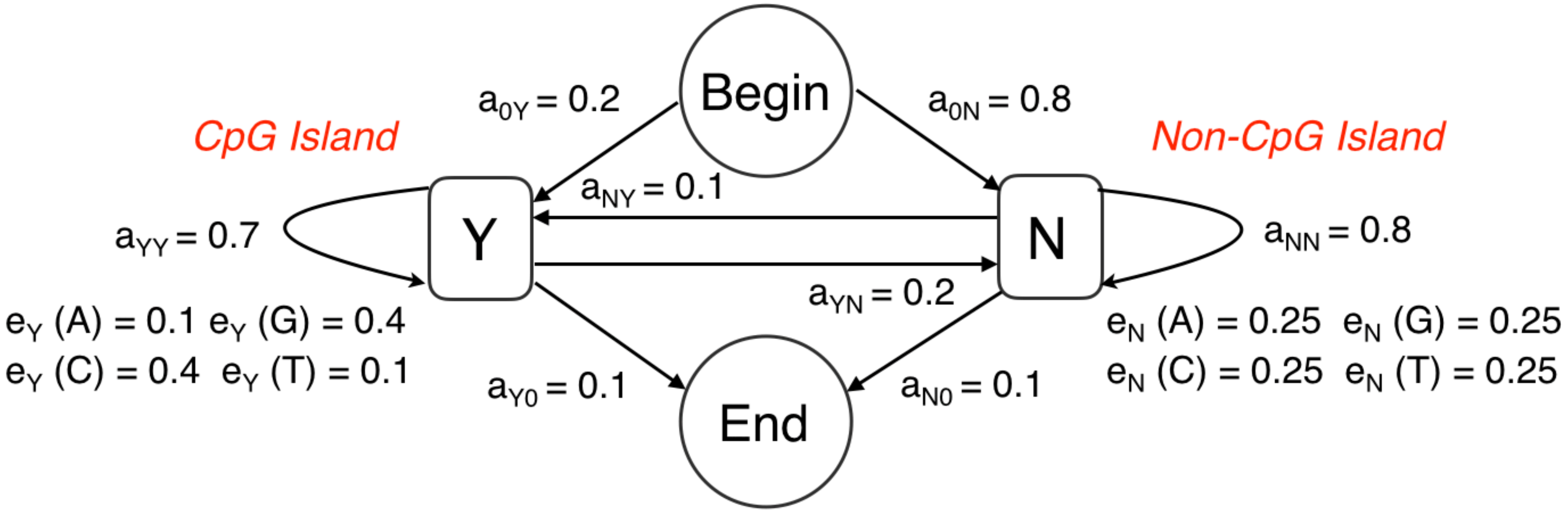
Each element requires about $2 \cdot N$ product and **1** sum

The time for the computation is $O(T \cdot N^2)$

Complexity Plot



Hidden Paths



$$\pi^* = \operatorname{argmax}_{\pi} [\, P(\pi \mid s, M)\,] \\ = \operatorname{argmax}_{\pi} [\, P(\pi, s \mid M)\,]$$

2^{13} different paths

Viterbi path: path that gives the best joint probability

s :	A	G	C	G	C	G	T	A	A	T	C	T	G
π_1:	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
π_2:	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N
π_3:	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y
π_4:	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N
π_5:	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y

Searching the Hidden Path

Viterbi decoding

Among all the possible path, choose the path π^* that maximizes the $P(\pi \mid s, M)$

$$\pi^* = \text{argmax}_{\pi} [P(\pi \mid s, M)] = \text{argmax}_{\pi} [P(\pi , s \mid M)]$$

A Posteriori decoding

For each position choose the state $\underline{\pi} (i)$:

$$\underline{\pi} (i) = \text{argmax}_{k} [P(\pi (i) = k \mid s, M)]$$

The contribution to this probability derives from all the paths that go through the state *k* at position *i.*

The A posteriori path can be a non-sense path (it may not be a legitimate path if some transitions are not permitted in the model)

Viterbi Algorithm

$\pi^* = \text{argmax}_{\pi}\ [\ \mathrm{P}(\ \pi\ ,\ s\ |\ M\ )\ ]$

The computation of $\mathrm{P}(s,\pi^*|\ M)$ can be decomposed in simplest problems

Let $V_k(i)$ be the probability of the most probable path for generating the subsequence $s^1s^2s^3\ldots\ldots s^{\ i}$ ending in the state *k* at iteration *i.*

Initialization: $V_{BEGIN}\,(0) = 1 \qquad V_i\,(0) = 0 \qquad \forall\ \ i \neq \mathrm{BEGIN}$

Recurrence: $V_l(\ i{+}1) = \mathrm{e}_{\,l}(\ s^{\,i+1}\ ) \cdot \mathrm{Max}_{\ k}(\ V_k\ \ (\ i\ ) \cdot \mathrm{a}_{\ k\,l}\ )$

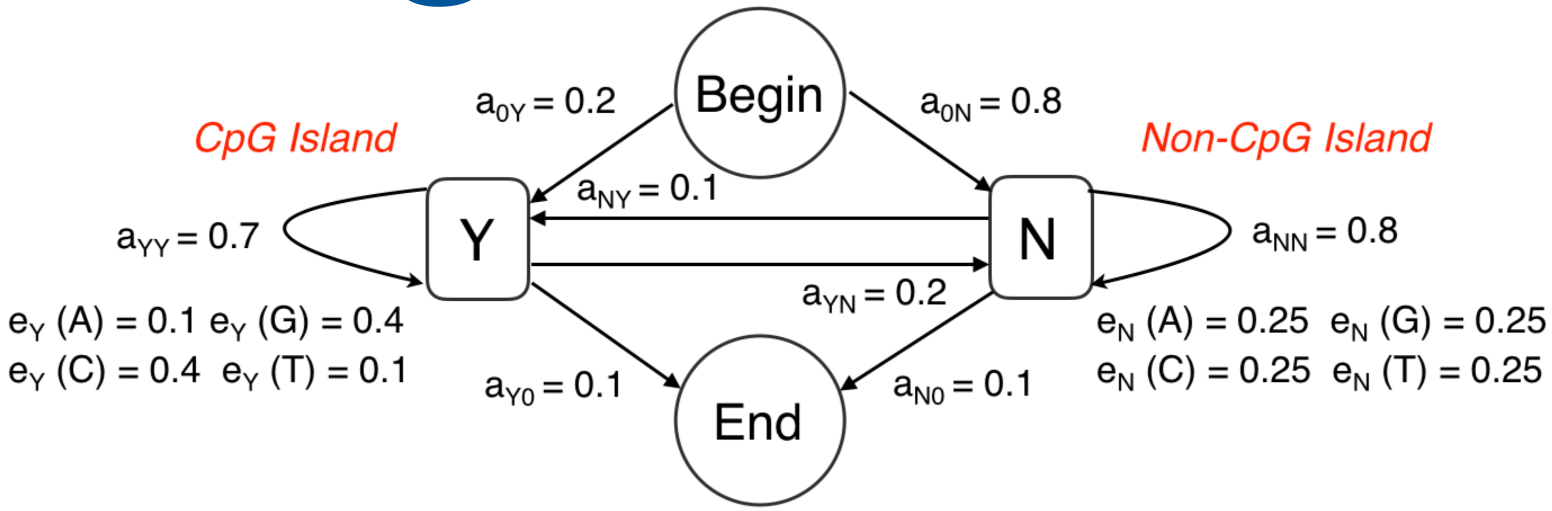
$\mathrm{ptr}_{\ i}(\ l\ ) = \text{argmax}_{\ k}\ (\ V_k\ \ (\ i\ ) \cdot \mathrm{a}_{\ k\,l}\ )$

Termination: $\mathrm{P}(\ s,\ \pi^*\ ) = \mathrm{Max}_k\ (V_k\,(\ T\ ) \cdot \mathrm{a}_{\ k\,0}\ )$

$\pi^*\ (\ T\ ) = \text{argmax}_{\ k}\ (V_k\,(\ T\ ) \cdot \mathrm{a}_{\ k\,0}\ )$

Traceback: $\pi^*\ (\ i\text{-}1\ ) = \mathrm{ptr}_{\ i}\,(\pi^*\ (\ i\ ))$

Viterbi Algorithm: Example



S: ATGCG *Initialization*: $V_{BEGIN}(0) = 1 \;\; V_i(0) = 0 \;\; \forall \;\; i \neq \text{BEGIN}$

Recurrence: $V_l(\text{i}) = \text{e}_l(s^{\text{i}}) \cdot \text{Max}_k(V_k(\text{i-1}) \cdot \text{a}_{kl})$ — $ptr_i(l) = argmax_k(V_k(i\text{-}1) \cdot a_{kl})$

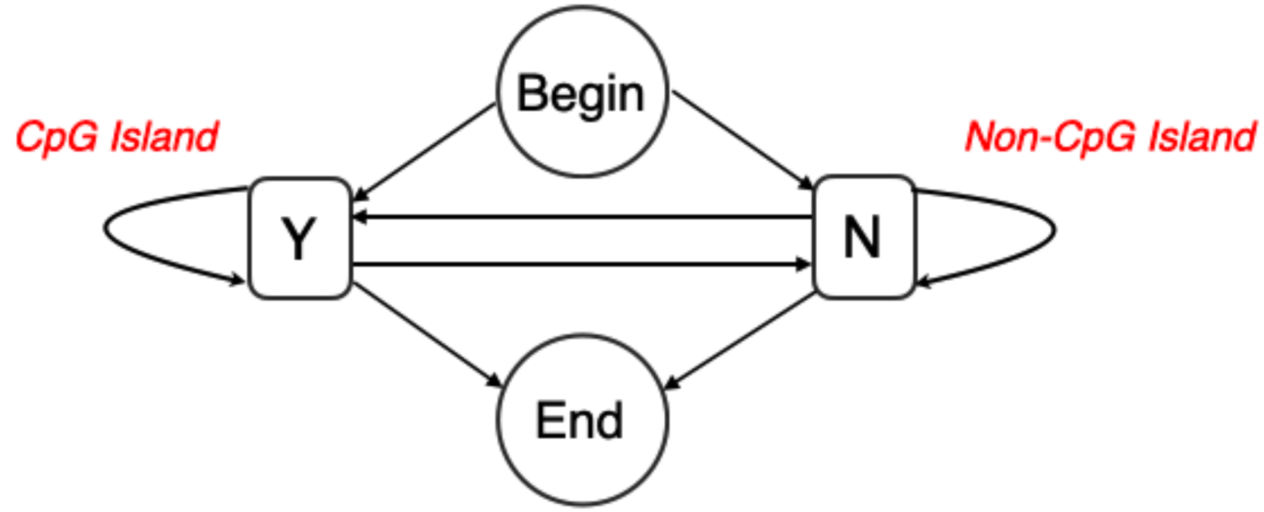
Termination: $\text{P}(s, \pi^*) = \text{Max}_k(V_k(T) \cdot \text{a}_{k0})$ — $\pi^*(T) = argmax_k(V_k(T) \cdot a_{k0})$

Traceback: $\pi^*(i\text{-}1) = \text{ptr}_i(\pi^*(i))$

	-	A	T	G	C	G	-
Begin	1	0	0	0	0	0	0
Y	0	0.2x0.1= =2e-2 ptr=Begin	Max(2e-2x0.7x0.1; 0.2x0.1x0.1) 2e-3; ptr=N	Max(2e-3x0.7x0.4; 1.6e-2x0.1x0.4) 6.4e-4; ptr=**N**	Max(6.4e-4x0.7x0.4; 3.2e-4x0.1x0.4) 1.79e-4; ptr=**Y**	Max(1.79e-4x0.7x0.4; 6.4e-5x0.1x0.4) 5.02e-5; ptr=**Y**	
N	0	0.8x0.25= =0.2 ptr=Begin	Max(2e-2x0.2x0.25; 0.2x0.8x0.25) 1.6e-2; ptr=**N**	Max(2e-3x0.2x0.25; 1.6e-2x0.8x0.25) 3.2e-4; ptr=N	Max(6.4e-4x0.2x0.25; 3.2e-4x0.8x0.25) 6.4e-5; ptr=N	Max(1.79e-4x0.2x0.25 ;6.4e-5x0.8x0.25) 1.28e-5; ptr=N	
End	0	0	0	0	0	0	Max(5.01e-5x0.1; 1.28e-5x0.1) 5.02e-6; ptr=**Y**

Exercise

Build an HMM modeling CpG islands sequences suing the following model where the states Y and N can emit the letters representing the 4 nucleotides.



For this exercise we consider as a training sequence the human chromosome 21 downloaded from the ucsc genome browser. For the GpC Island annotation refer to the cpgIslandExt.txt.gz file.