Hidden Markov Models

Laboratory of Bioinformatics I
Module 2

March 23, 2018

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 $\{a_{kj}\}$

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

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

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

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

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

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

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

s: sequence, **π**: path through the states

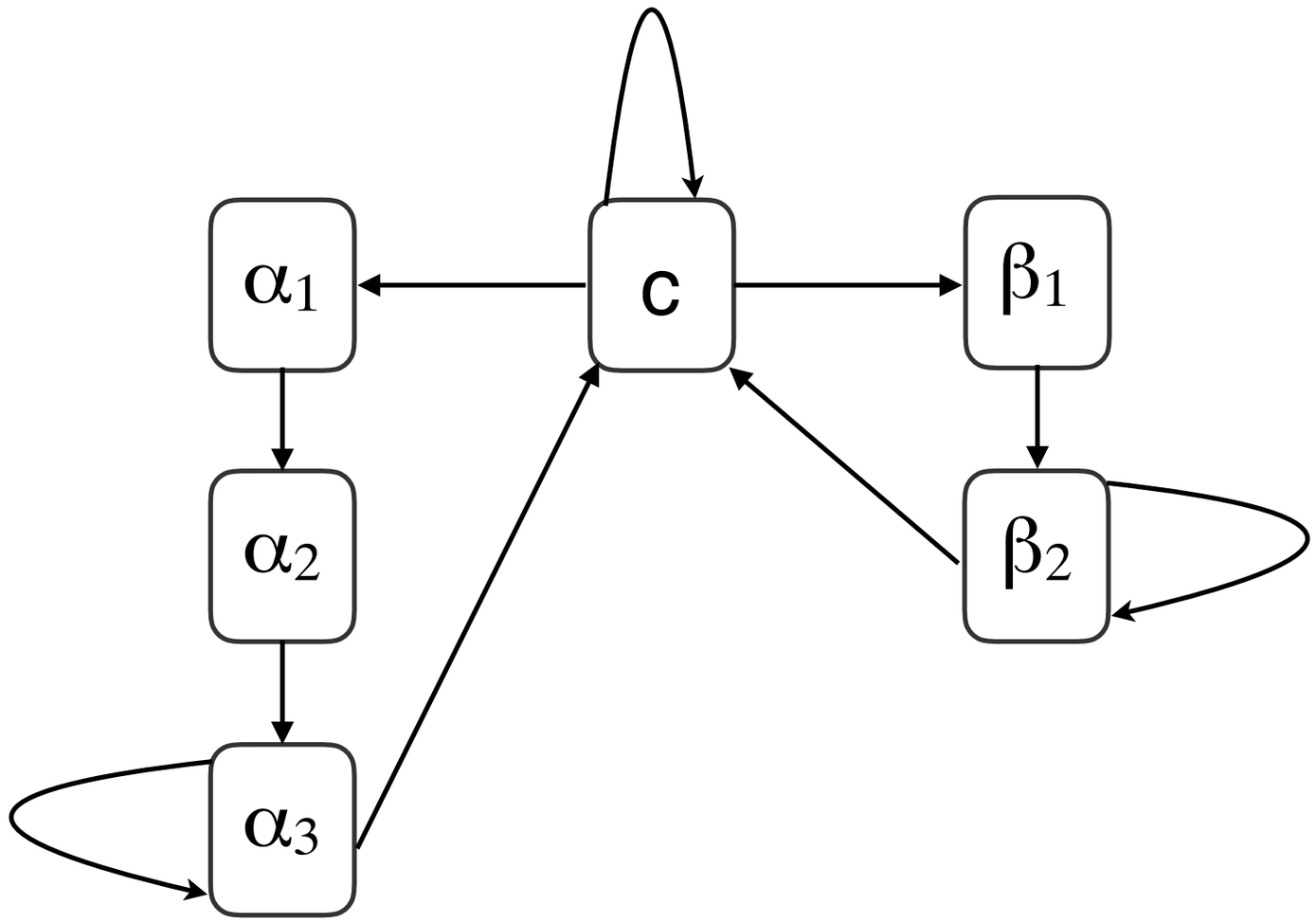
Hidden Markov Models

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

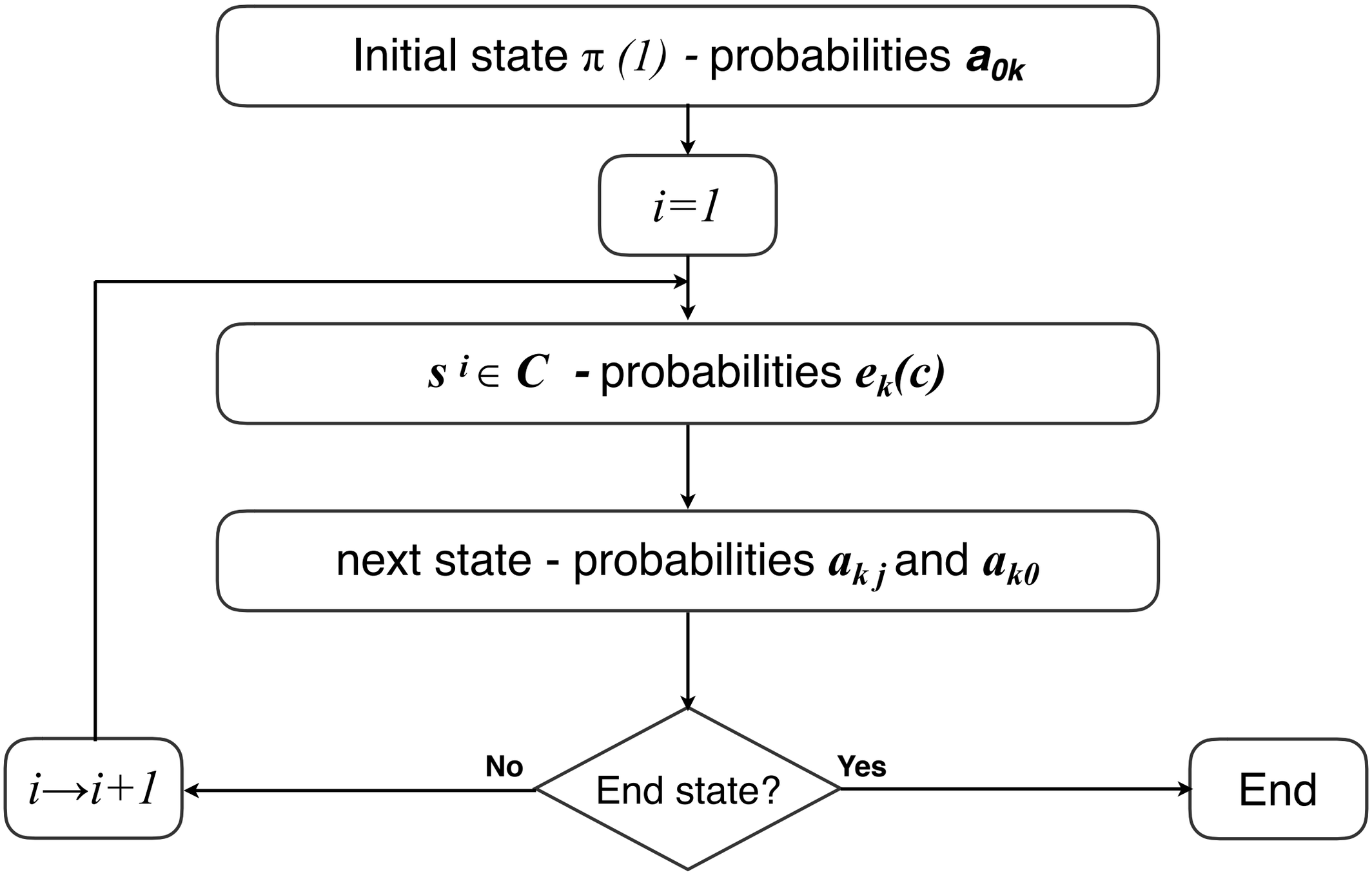
Emission probabilities cast the propensity of observable symbols in each state

Secondary Structure



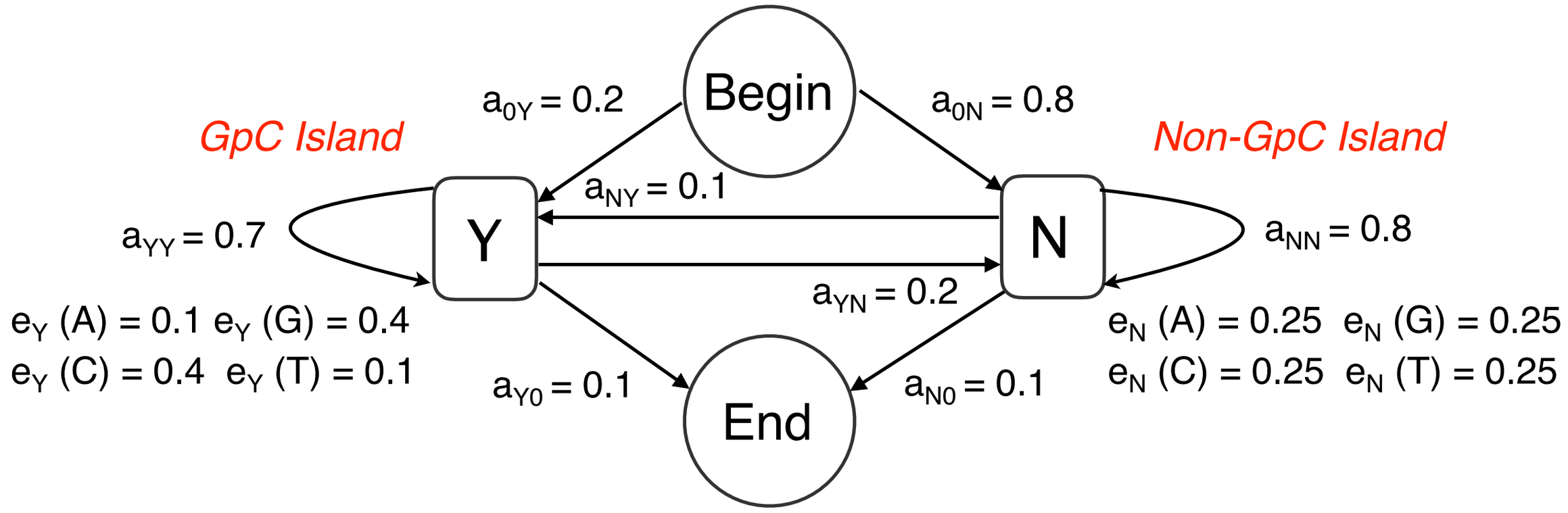
S	**A**	**L**	**K**	**M**	**N**	**Y**	**T**	**R**	**E**	**I**	**M**	**V**	**A**	**S**	**N**	**Q**	s: sequence
c	α_1	α_2	α_3	α_3	α_3	α_3	c	c	c	c	β_1	β_2	β_2	β_2	c	c	π: path
c	α	α	α	α	α	α	c	c	c	c	β	β	β	β	c	c	Y(π): labels

Generating HMM Sequence



GpC Islands Model

Probability of a sequence s with a given path π



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: $0.1 \times 0.4 \times 0.4 \times 0.4 \times 0.4 \times 0.4 \times 0.1 \times 0.25 \times 0.25 \times 0.25 \times 0.25 \times 0.25 \times 0.25$

Transition: $0.2 \times 0.7 \times 0.7 \times 0.7 \times 0.7 \times 0.7 \times 0.7 \times 0.2 \times 0.8 \times 0.8 \times 0.8 \times 0.8 \times 0.8 \times 0.1$

Joint Probability

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

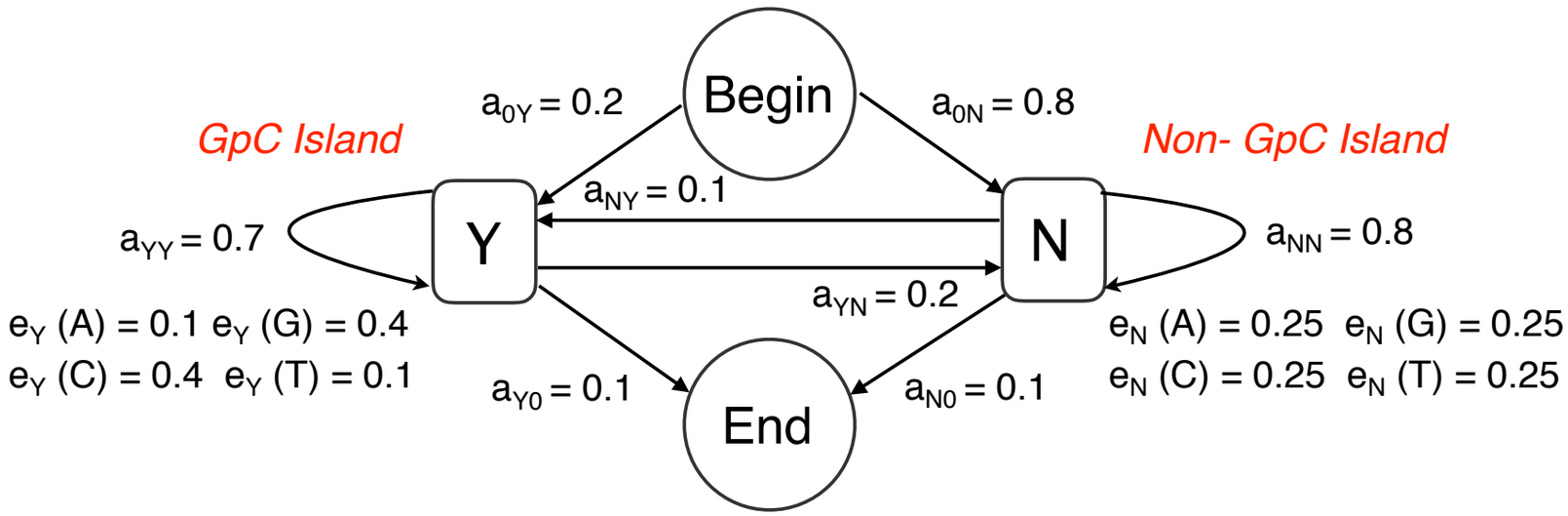
$$P(s,\pi \mid M) = P(s \mid \pi, M) \cdot P(\pi \mid M)$$

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

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

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

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

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

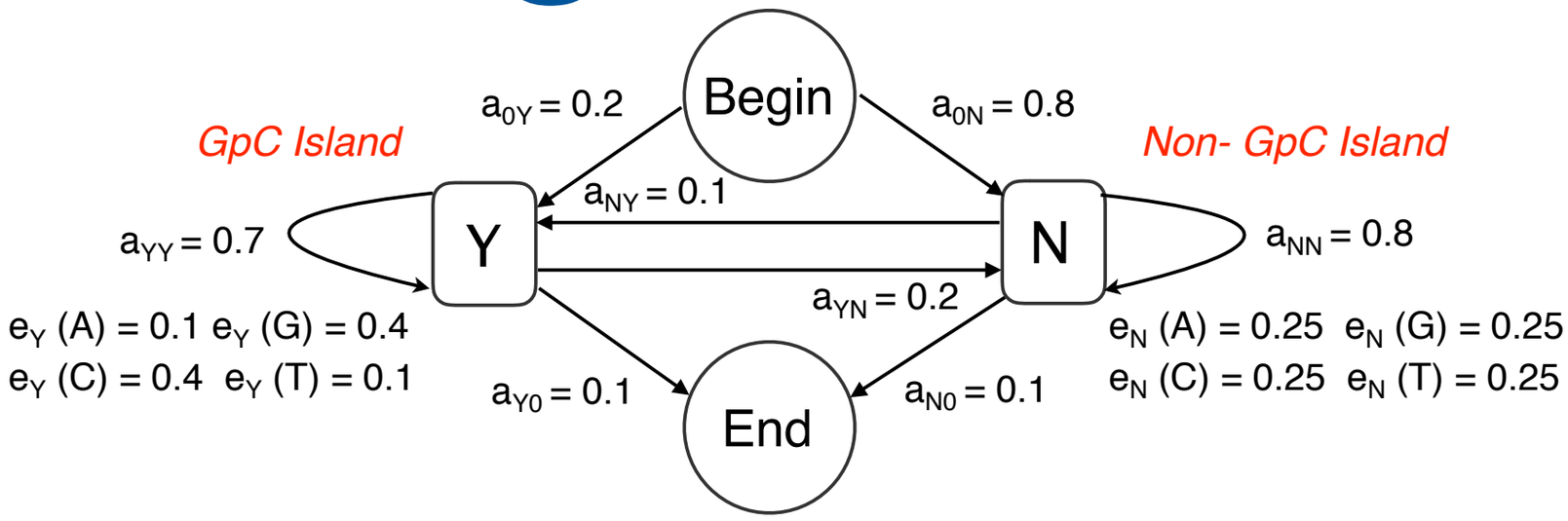
$= \mathrm{e}_{\,l}(\, s^{\,i+1}\,) \cdot \Sigma_k F_k\,(\, i\,) \cdot \mathrm{a}_{\,k\,l}$

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

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

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

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

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

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

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

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

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

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 $O(T \cdot N^T)$

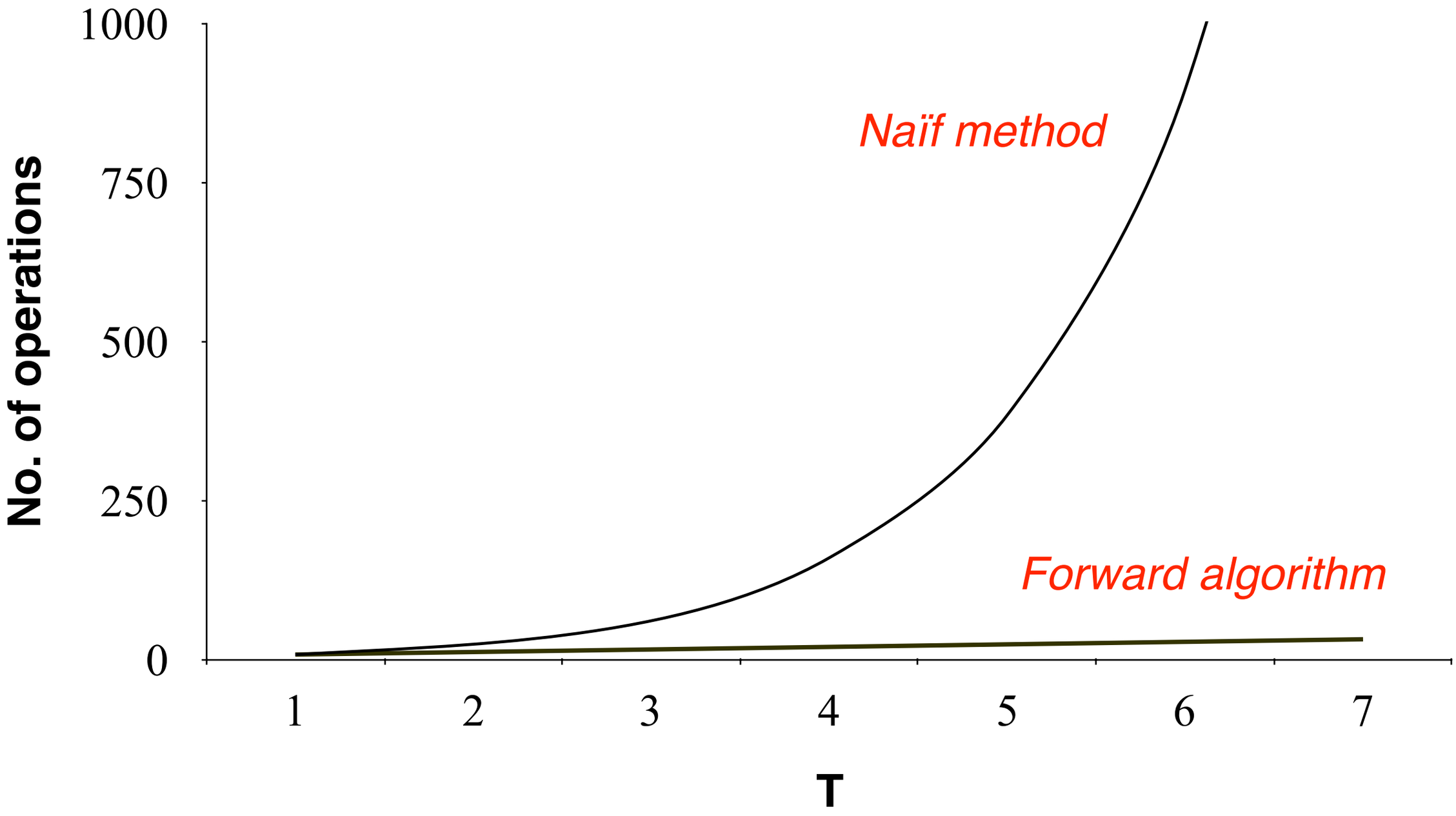
Forward Algorithm

T positions, N values for each position

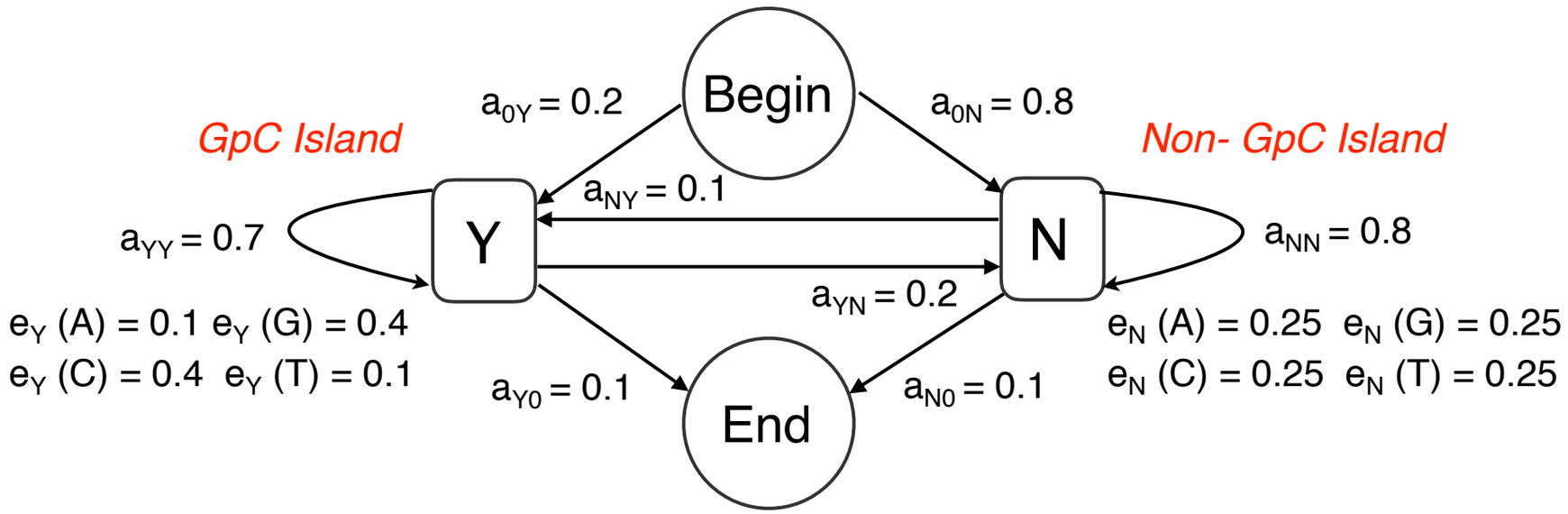
Each element requires about $2 \cdot N$ product and **1** sum

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

Complexity Plot



Hidden Paths



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

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

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

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

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

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

Recurrence: $V_l(i+1) = e_l(s^{i+1}) \cdot \text{Max}_k(V_k(i) \cdot a_{kl})$

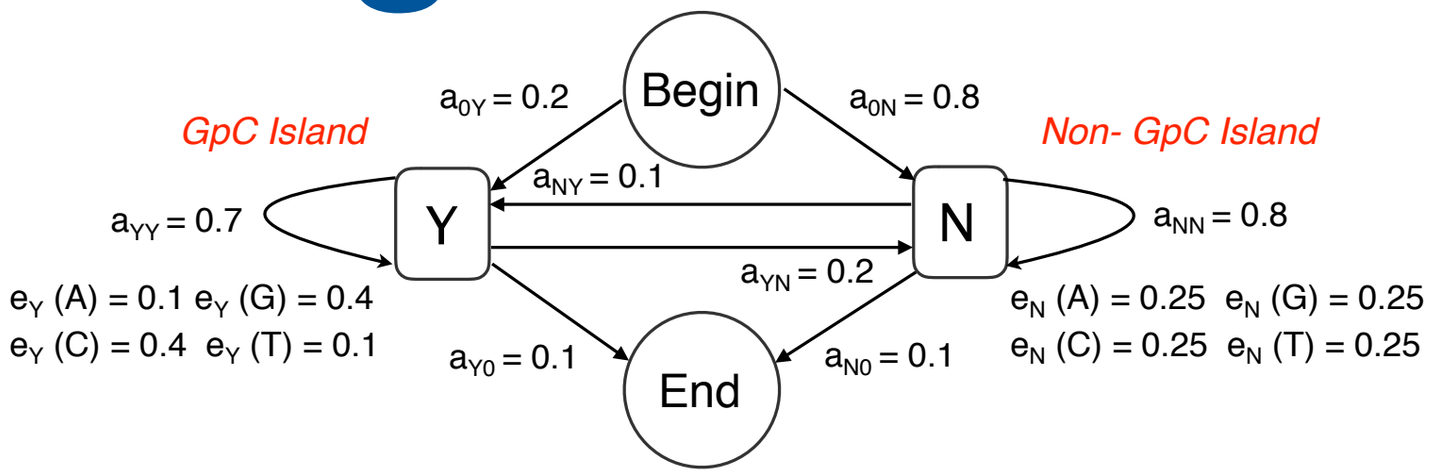
$\text{ptr}_i(l) = \text{argmax}_k(V_k(i) \cdot a_{kl})$

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

$\pi^*(T) = \text{argmax}_k(V_k(T) \cdot a_{k0})$

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

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**

A Posteriori Decoding

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

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

How to compute $\mathrm{P}(\,\pi\,(i) = k|\,s,\,M\,)$ for any state *k* and any position *i*?

$$P(\pi(i)=k\,|\,s,M)=\frac{P(\pi(i)=k,s\,|\,M)}{P(s\,|\,M)}$$

$$P(\pi(i)=k,s\,|\,M)=P(s^1s^2\ldots s^i,\pi(i)=k\,|\,M)\cdot P(s^{i+1},s^{1+2},..s^T\,|\,\pi(i)=k,M)=$$

$$=F_k(i)\cdot B_k(i)$$

$$P(\pi(i)=k\,|\,s,M)=\frac{F_k(i)\cdot B_k(i)}{P(s\,|\,M)}$$

Elements of the Forward and Backward matrices

Computed with Forward or Backward algorithm termination steps

Exercise

Using the BLAST tool at Uniprot, retrieve all the SwissProt sequences that are similar with an E-value <0,001 to the Rhodopseudomonas cytochrome C (P00091) .

Download the sequences in Fasta format and align with ClustalW, Muscle or T-Coffee

Analyse the conserved positions in the alignments

Repeat with the Arabidopsis (Q93VA3) and the human (P99999) sequences

Compare the results, an in particular the pattern of conserved residues