Bioinformatika
Hidden Markov Models

Michael Andel
(some slides are courtesy of Mark Craven, U. of Wisconsin)

Motivation

- What is a gene?

Motivation

- What is a gene?
- Can you formalize a gene as a regular expression?

Motivation

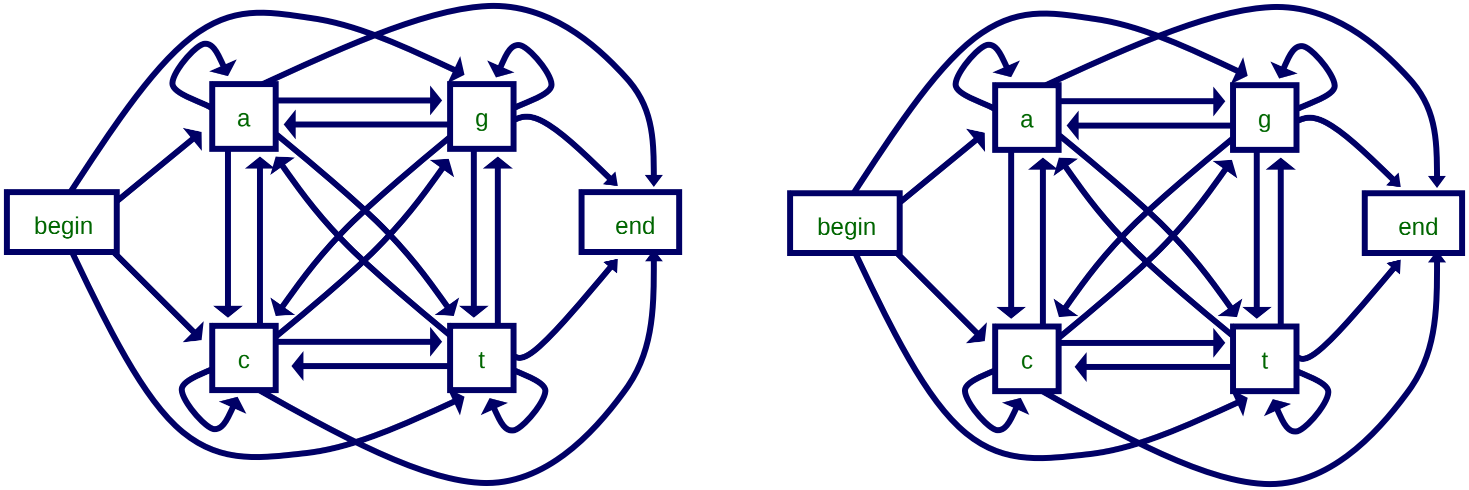
- What is a gene?
- Can you formalize a gene as a regular expression?
- What is HMM?

Motivation

- What is a gene?
- Can you formalize a gene as a regular expression?
- What is HMM?
- What are the general tasks with HMMs?

Motivation

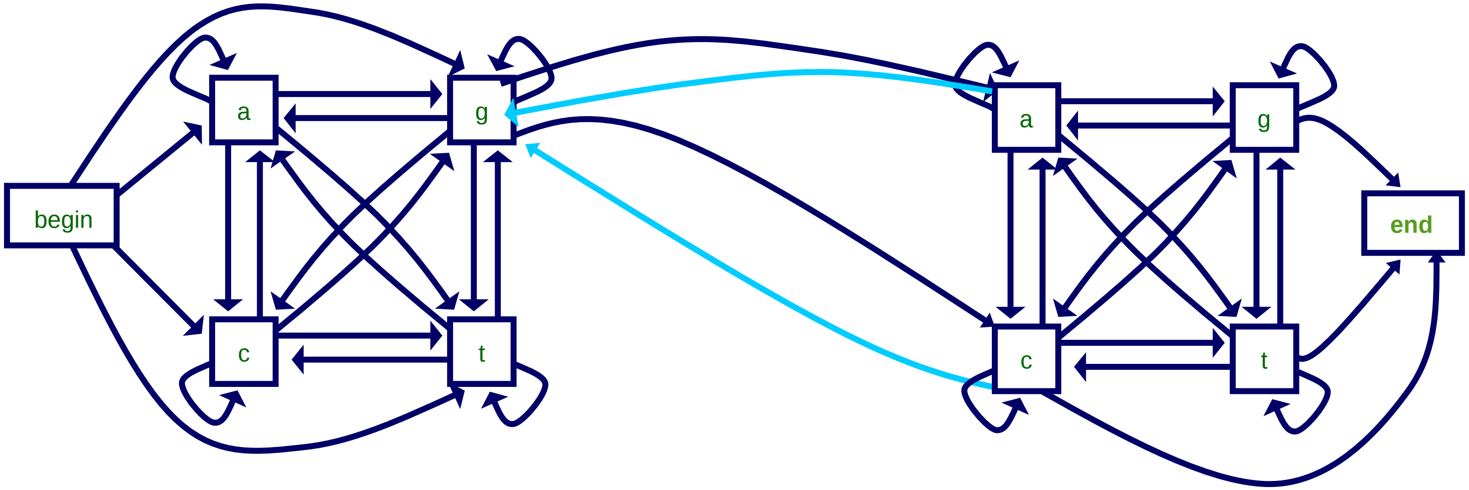
1. Train two MMs: one to represent represent CpG regions, the other to the background (nCpG)



- ➔ Given a new sequence, use two models to *classify* the sequence (CpG or nCpG).
- ➔ Given a new sequence, find the CpG islands within (**?!?**)

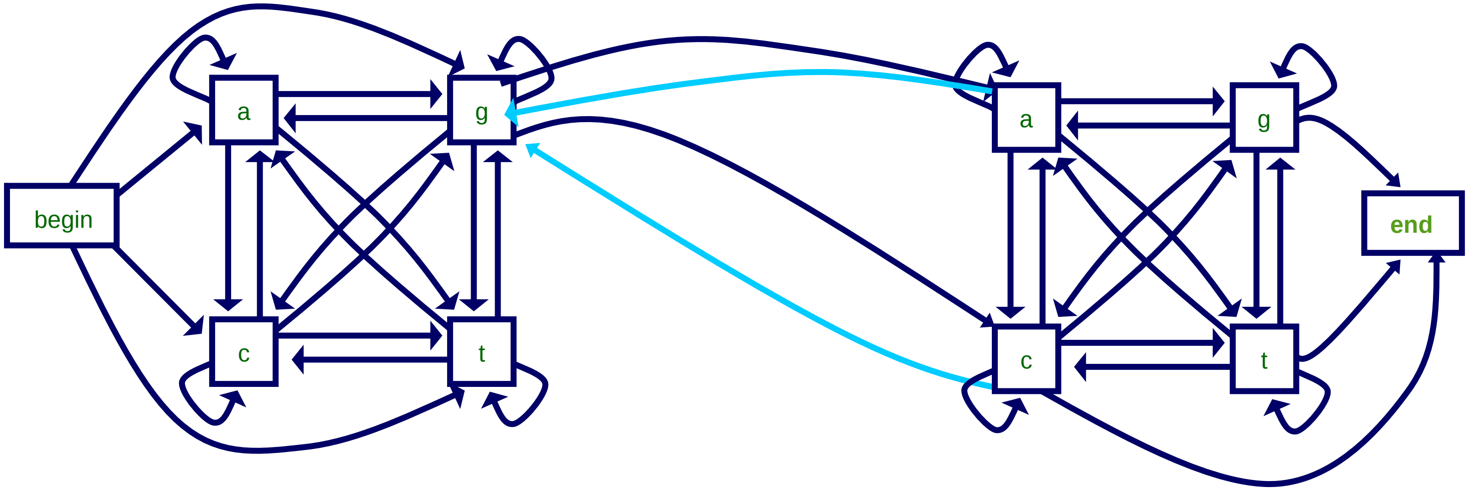
Motivation

1. Train two MMs: one to represent represent CpG regions, the other to the background (nCpG)



Motivation

1. Train two MMs: one to represent represent CpG regions, the other to the background (nCpG)



2. Join the 2 models into one HMM:
   - {a, c, t, g} → {a_{CpG}, a_{nCpG}, c_{CpG}, c_{nCpG}, t_{CpG}, t_{nCpG}, g_{CpG}, g_{nCpG}}

3. Segment a sequence as a maximum likely walk through the state space.

Hidden Markov Model

$M = (A, S, P_t, P_e)$

- $A = \{a, c, t, g\}$
- $S = \{s_1, \dots, s_K\}$
- $P_t : S \times S \rightarrow [0, 1]$
- $P_e : S \times A \rightarrow [0, 1]$

$P(x_1, \dots, x_L; s_1, \dots, s_L) =$

$= P(s_1) \cdot P(x_1|s_1) \cdot P(x_2|s_2) \cdot P(s_2|s_1) \cdot$

$\cdot \dots \cdot P(x_L|s_L) \cdot P(s_L|s_{L-1})$

with $x_i \in A$, $s_i \in S$

Sequence Annotation

Given:

- observed sequence $\mathbf{x} \in \{a, c, t, g\}^L$
- model $M = (A, S, P_t, P_e)$

Find:

- max. likely labeling $\mathbf{s} \in S^L$ → Viterbi alg.

Sequence Annotation

Given:

- observed sequence $\mathbf{x} \in \{a, c, t, g\}^L$
- model $M = (A, S, P_t, P_e)$

Find:

- max. likely labeling $\mathbf{s} \in S^L$ → Viterbi alg.

But how have the P_t, P_e been learnt??

Sequence Annotation

Given:

- observed sequence $\mathbf{x} \in \{a, c, t, g\}^L$
- model $M = (A, S, P_t, P_e)$

Find:

- max. likely labeling $\mathbf{s} \in S^L$ → Viterbi alg.

But how have the P_t, P_e been learnt??

- Supervised: $T = \{(\mathbf{x}_i, \mathbf{s}_i)\}_{i=1\ldots N}$ where $\mathbf{x}_i \in A^*$, $\mathbf{s}_i \in S^*$

Sequence Annotation

Given:

- observed sequence $\mathbf{x} \in \mathbf{A}^L$
- model $M = (A, S, P_t, P_e)$

Find:

- max. likely labeling $\mathbf{s} \in S^L$ → Viterbi alg.

But how have the P_t, P_e been learnt??

- Supervised: $T = \{(\mathbf{x}_i, \mathbf{s}_i)\}_{i=1...N}$ where $\mathbf{x}_i \in A^*$, $\mathbf{s}_i \in S^*$
- Unsupervised: $T = \{\mathbf{x}_i\}_{i=1...N}$ where $\mathbf{x}_i \in A^*$
  – Expectation-Maximization → Baum-Welsh alg. (**later**)

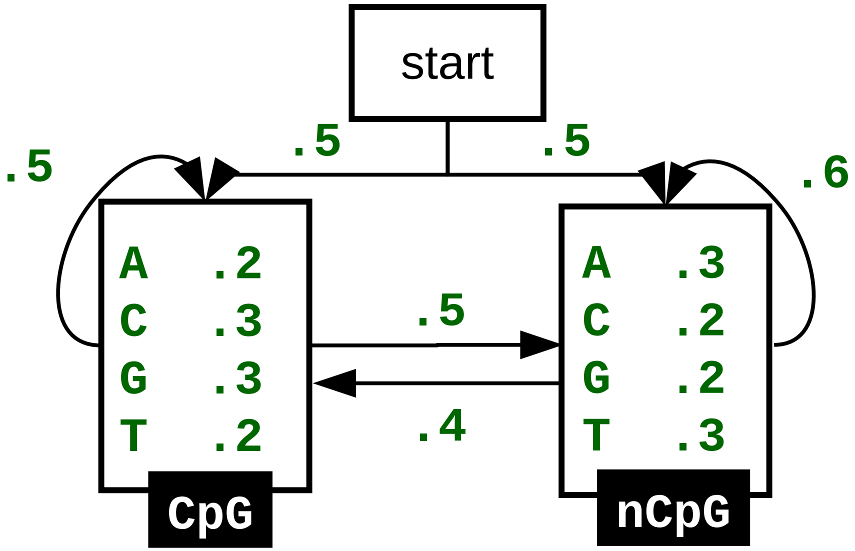
Viterbi algorithm

Ex: Naive model of CpG detection

$$\mathbf{s}^* = \arg\max_{s_0 \ldots s_N \in S^N} p(x_0 \ldots x_N ; s_0 \ldots s_N)$$

$$p(x_i \ldots x_N ; s_i \ldots s_N) = \prod_{i=1}^{N} p(x_i | s_i)\, p(s_i | s_{i-1}),$$

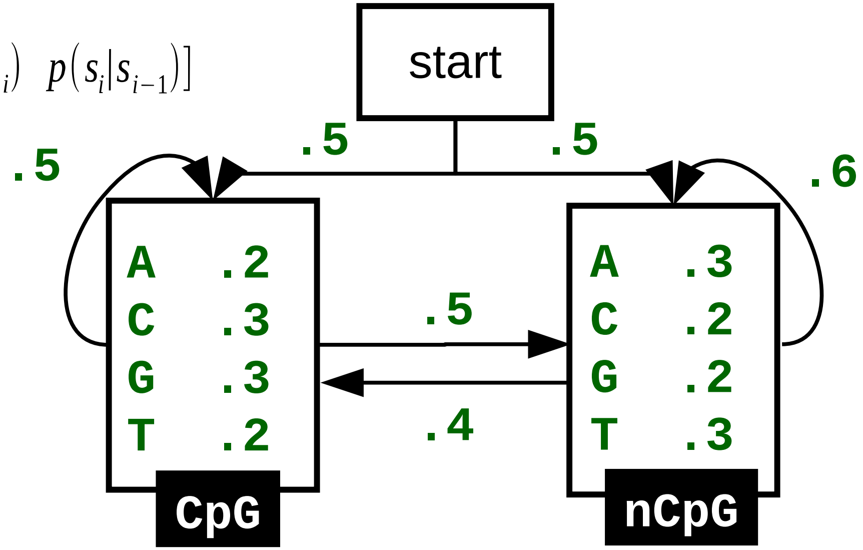
$$p(s_0) = 1$$



Viterbi algorithm (ex.)

	ε	A	T	G	G	C	A	C	T	A
START	1	0	0	0	0	0	0	0	0	0
CpG	0									
nCpG	0									

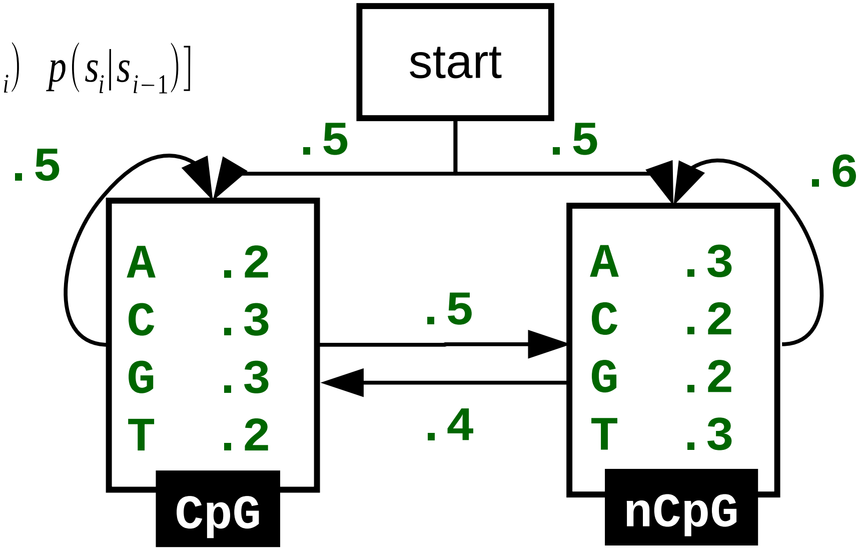
$$\max_{s_i\in S} p(x_0\ldots x_i|s_i) = \max_{s_{i-1}\in S}\left[p(x_0\ldots x_{i-1}|s_{i-1}) \max_{s_i\in S} p(x_i|s_i) \; p(s_i|s_{i-1})\right]$$



Viterbi algorithm (ex.)

	ε	A	T	G	G	C	A	C	T	A
START	1	0	0	0	0	0	0	0	0	0
CpG	0	**1** x .2 x .5 0 x .2 x .5 0 x .2 x .4 .1								
nCpG	0	**1** x .3 x .5 0 x .3 x .5 0 .3 xx .6 .15								

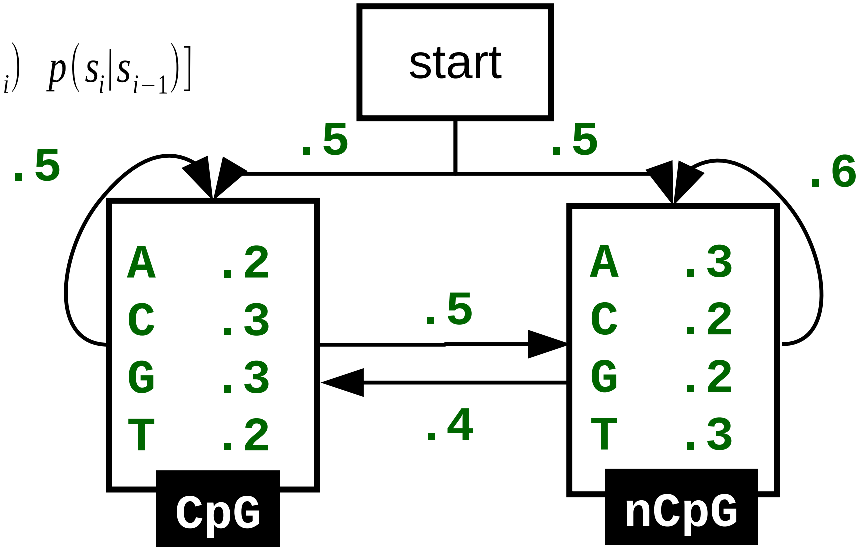
$$\max_{s_i\in S} p\left(x_0\ldots x_i|s_i\right) = \max_{s_{i-1}\in S}\left[p\left(x_0\ldots x_{i-1}|s_{i-1}\right) \;\max_{s_i\in S} p\left(x_i|s_i\right)\;\; p\left(s_i|s_{i-1}\right)\right]$$



Viterbi algorithm (ex.)

	ε	A	T	G	G	C	A	C	T	A
START	1	0	0	0	0	0	0	0	0	0
CpG	0	**1** x .2 x .5 0 x .2 x .5 0 x .2 x .4 .1	**0** x .2 x .5 .1 x .2 x .5 .15 x .2 x .4 .012							
nCpG	0	**1** x .3 x .5 0 x .3 x .5 0 .3 xx .6 .15	**0** x .3 x .5 .1 x .3 x .5 .15 x .3 x .6 .027							

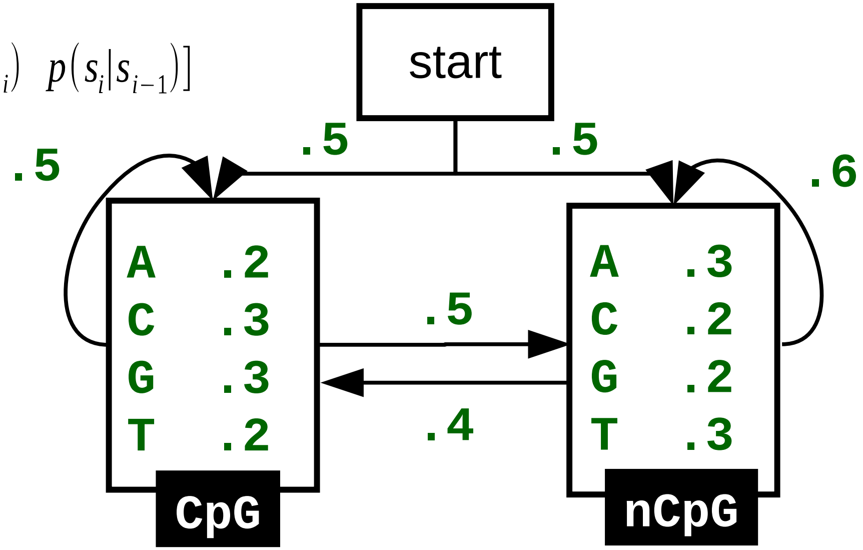
$$\max_{s_i \in S} p(x_0 \ldots x_i | s_i) = \max_{s_{i-1} \in S} [p(x_0 \ldots x_{i-1} | s_{i-1}) \ \max_{s_i \in S} p(x_i | s_i) \ \ p(s_i | s_{i-1})]$$



Viterbi algorithm (ex.)

	ε	A	T	G	G	C	A	C	T	A
START	1	0	0	0	0	0	0	0	0	0
CpG	0	**1** x .2 x .5 0 x .2 x .5 0 x .2 x .4 .1	**0** x .2 x .5 .1 x .2 x .5 .15 x .2 x .4 .012	0 .012 x .3 x .5 .027 x .3 x .4 .0032	0 .0032 x .3 x .5 .0032 x .3 x .4 5e-4	0 .012 x.3 x .5 .027 x .3 x .4 5e-5				
nCpG	0	**1** x .3 x .5 0 x .3 x .5 0 .3 xx .6 .15	**0** x .3 x .5 .1 x .3 x .5 .15 x .3 x .6 .027	0 .012 x .2 x .5 .027 x .2 x .6 .0032	0 .0032 x .2 x .5 .0032 x .2 x .6 4e-4	0 .012 x .2 x .5 .027 x .2 x .6 4e-5				

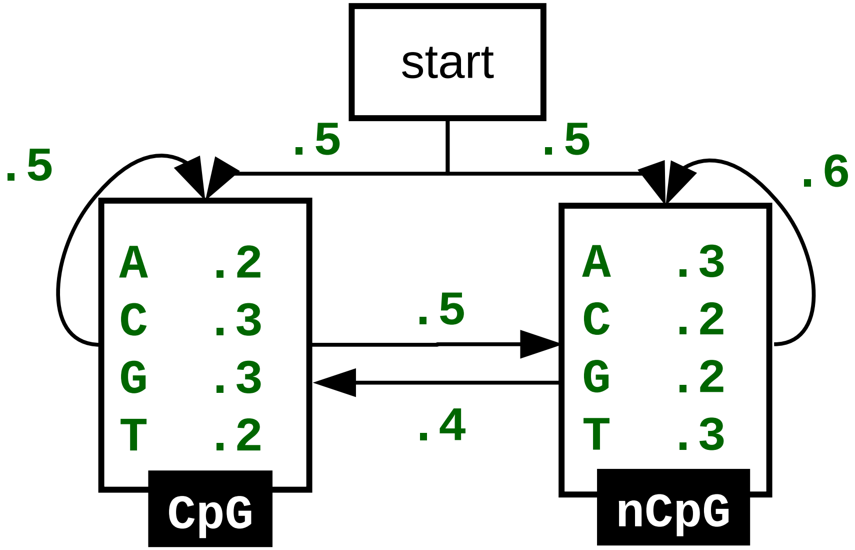
$$\max_{s_i \in S} p(x_0 \ldots x_i | s_i) = \max_{s_{i-1} \in S} [p(x_0 \ldots x_{i-1} | s_{i-1}) \ \max_{s_i \in S} p(x_i | s_i) \ \ p(s_i | s_{i-1})]$$



Viterbi algorithm (ex.)

	ε	A	T	G	G	C	A	C	T	A
START	0	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf
CpG	-inf	ln.2+**0**+ln.5 ln.2+-inf+ln.5 ln.2+-inf+ln.4 -2.30	ln.2+**-inf**+ln.5 ln.2+0+ln.5 ln.2+ln.15+ln.4 -2.3							
nCpG	-inf	ln.3+**0**+ln.5 ln.3+-inf+ln.5 ln.3+-inf+ln.6 -1.9	**-inf** ln.3+ln.1+ln.5 ln.3+ln.15+ln.6 -1.9							

$$\arg\max_{s_i\in S} p(x_0\ldots x_i|s_i) = \arg\max_{s_i\in S} \log p(x_0\ldots x_i|s_i)$$



Viterbi alg. – pseudocode

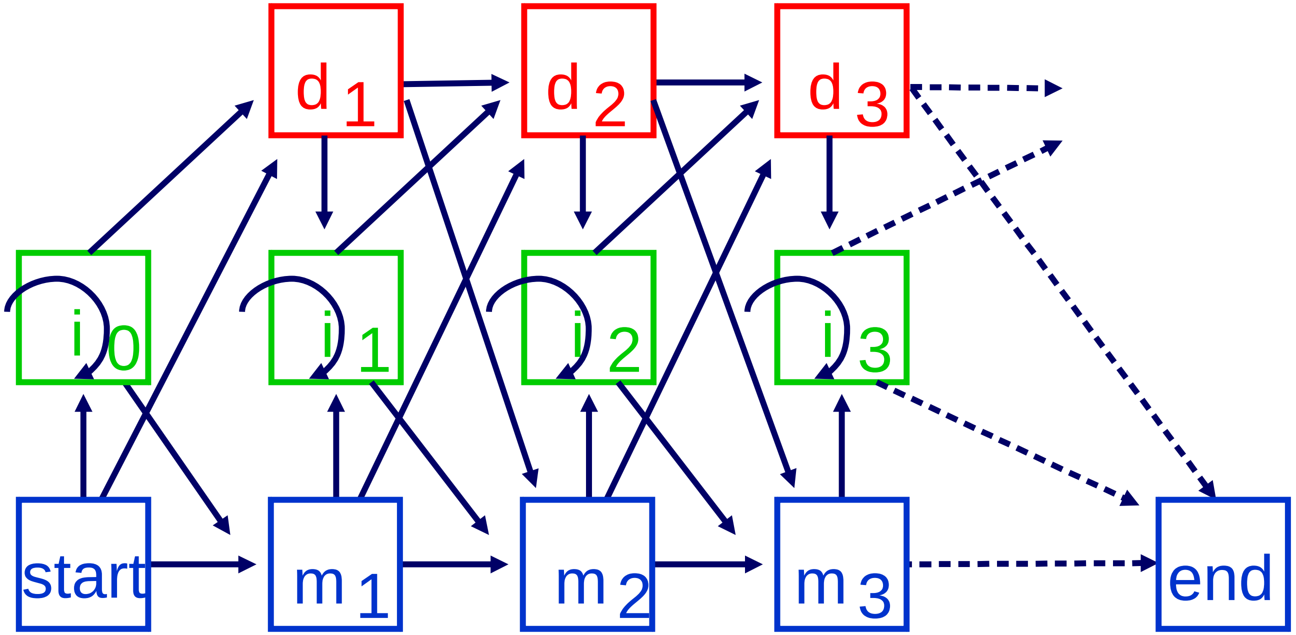
```
function VITERBI( O, S, π, Y, A, B ) : X
    for each state si do
        T1[i,1] ← πi·Biy1
        T2[i,1] ← 0
    end for
    for i ← 2,3,...,T do
        for each state sj do


        end for
    end for

    xT ← szT
    for i ← T,T-1,...,2 do
        zi-1 ← T2[zi,i]
        xi-1 ← szi-1
    end for
    return X
end function
```

Assignment – Gene Finding

- http://www.biostat.wisc.edu/~craven/776/hw3.html
- **You can use an existing implementation of Viterbi alg.**
- 15 pt.

Profile HMM

```
ATTGCC- A TT--
ATGGCC- A TT--
ATC-CA- A TTTT
ATCTTC- - TT--
ATTGCCG A TT--
```



Delete states (silent)

Insert states

Match states

Profile HMM – Excercise

```
bat    AG---C
rat    A-AA-C
cat    AG-AA-
dog    -GAC-C
fox    AC---G
       xxxxxx
```

Given the alignment:

- Make a profile HMM

Profile HMM – Excercise

```
bat    AG---C
rat    A-AA-C
cat    AG-AA-
dog    -GAC-C
fox    AC---G
       12-3-4
```

- **Heuristic:**

```
if #('-') < #('X')-1:
    if pos == '-':
        pos = '-'
    else:
        pos = 'X'
else:
        pos = '*'
```

Profile HMM – Excercise

```
bat   AG---C
rat   A-AA-C
cat   AG-AA-
dog   -GAC-C
fox   AC---G
      12---3
```

- **… more regularized:**

```
if #('-') < #('X')-1:
    if pos == '-':
        pos = '-'
    else:
        pos = 'X'
else:
        pos = '*'
```

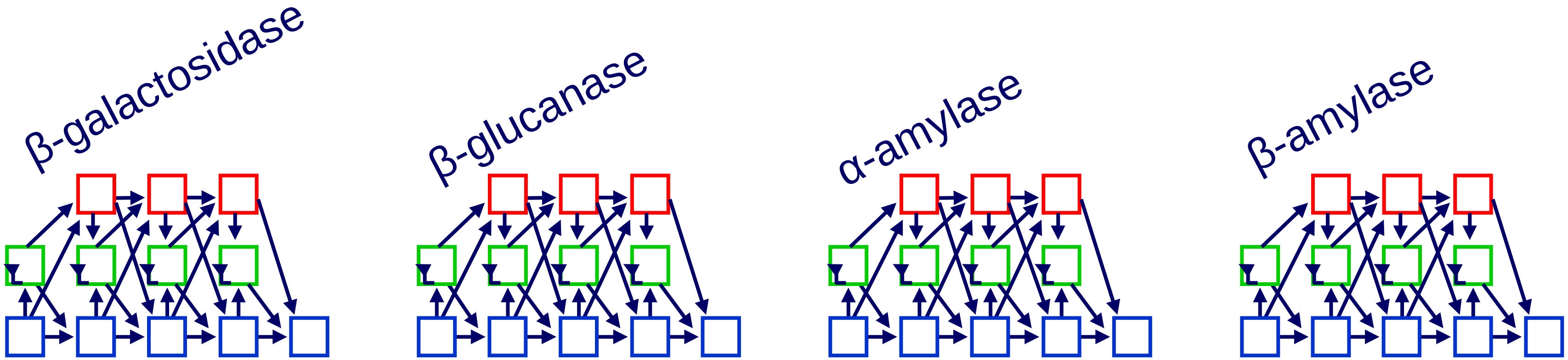
Profile HMM – Excercise

bat	AG---C
rat	A-AA-C
cat	AG-AA-
dog	-GAC-C
fox	AC---G
	12---3
ass	AGG

- Make a profile model *M*
- Compute the probability that a new (non-aligned) sequence has been generated by *M*:

$$P(AGG|M) = p_M(A|s_1)p_M(s_1)p_M(G|s_2)p_M(s_2|s_1)p_M(G|s_3)p_M(s_3|s_2)$$

Sequence Categorization



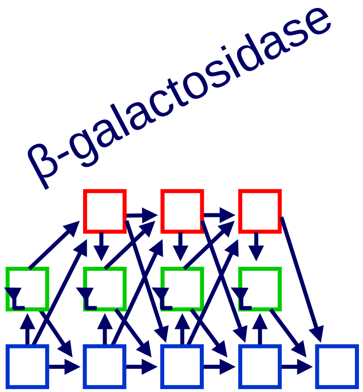
Given:

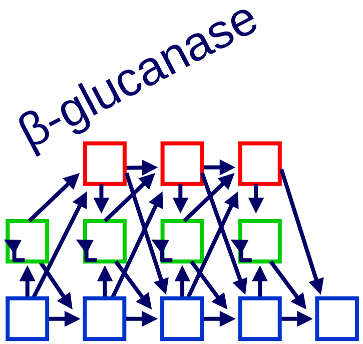
- observed sequence **x** ∈ A^L
- Set of *K* models {M_k = (A, S, P_t, P_e)} of *K* families

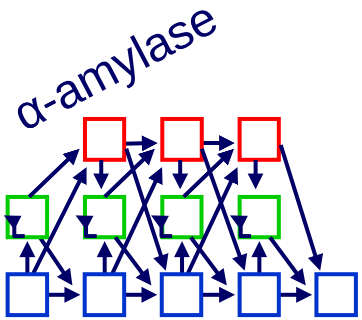
Do:

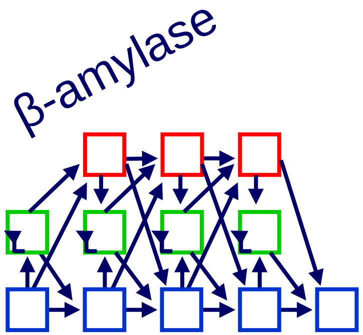
- categorize **x** into one of the families

Sequence Categorization









Given:

- observed sequence **x** ∈ {a, c, t, g}^{L}
- Set of *K* models {M_k = (A, S, P_t, P_e)} of *K* families

Do:

- categorize **x** into one of the families

$$p(\alpha-\text{amyl}|x_1\ldots x_L) \;<\; p(\beta-\text{gluc}|x_1\ldots x_L)$$

$$p(\alpha-\text{amyl})\,p(x_1\ldots x_L|\alpha-\text{amyl}) \;<\; p(\beta-\text{gluc})\,p(x_1\ldots x_L|\beta-\text{gluc})$$

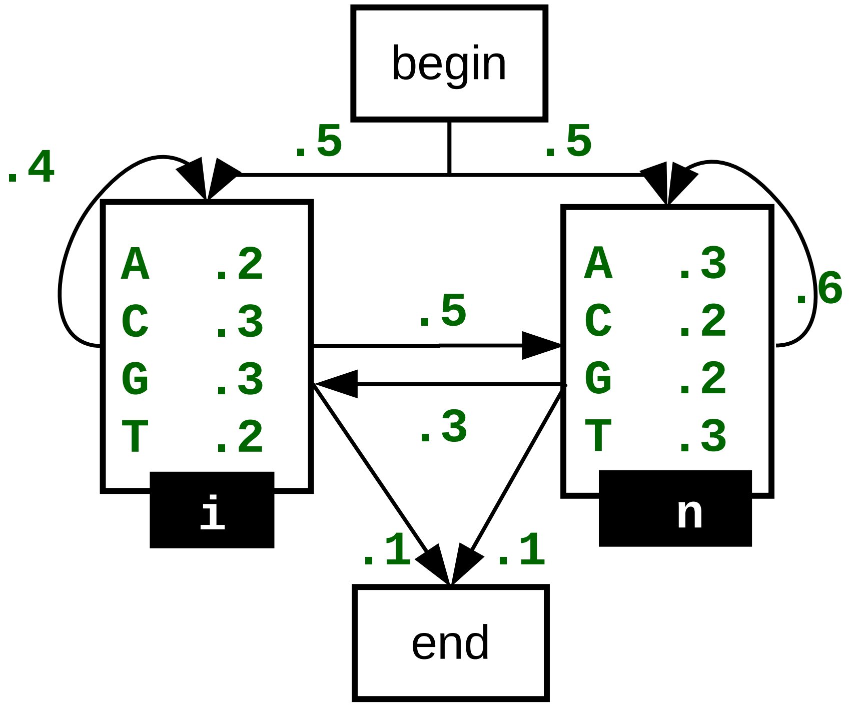
$$p(x_1\ldots x_L|c) \;=\; \sum_{s_1\ldots s_L\in S^L} p(x_1\ldots x_L; s_1\ldots s_L|c) \;=\; \sum_{s_1\ldots s_L\in S^L} p_c(x_1|s_1)\,p_c(s_1)\prod_{i=1}^{L} p_c(x_i|s_i)\,p(s_i|s_{i-1})$$

Forward algorithm (ex.)

$$\sum_{s_1 s_2 s_3 \in S^3} p(\mathrm{CAG}, s_1 s_2 s_3) = \sum_{s_1 \in \{i,n\}} p(\mathrm{C}|s_1)\, p(s_1|\mathrm{b}) \sum_{s_2 \in \{i,n\}} p(\mathrm{G}|s_2)\, p(s_2|s_1) \sum_{s_3 \in \{i,n\}} p(\mathrm{A}|s_3)\, p(s_3|s_2)\, p(\mathrm{e}|s_3)$$

$$T[i,j] = \sum_k T[k,i-1] * p(s_i|s_k) * p(x_i|s_i)$$

	ε	C	G	A	ε
begin	1	0	0	0	0
I	0	0	0	0	0
N	0	0	0	0	0
end		0	0	0	0

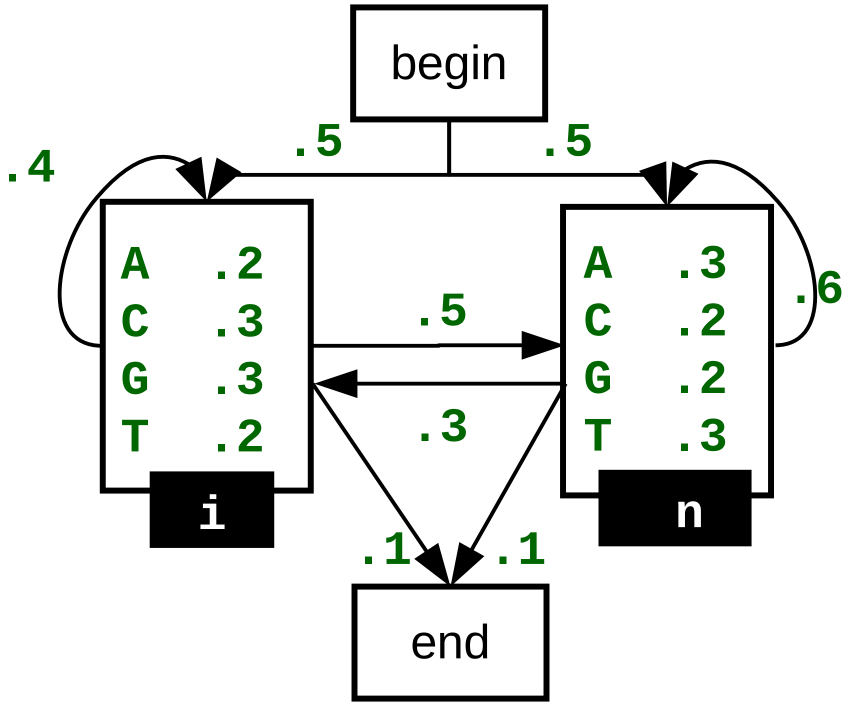


Forward algorithm (ex.)

$$\sum_{s_1 s_2 s_3 \in S^3} p(\mathrm{CAG}, s_1 s_2 s_3) = \sum_{s_1 \in \{i,n\}} p(\mathrm{C}|s_1)\, p(s_1|\mathrm{b}) \sum_{s_2 \in \{i,n\}} p(\mathrm{G}|s_2)\, p(s_2|s_1) \sum_{s_3 \in \{i,n\}} p(\mathrm{A}|s_3)\, p(s_3|s_2)\, p(\mathrm{e}|s_3)$$

$$T[i,j] = \sum_k T[k,i-1] * p(s_i|s_k) * p(x_i|s_i)$$

	ε	C	G	A	ε
begin	**1**	0	0	0	0
I	0	**1** x .5 x .3 0 x .4 x .3 0 x .3 x .3 .15	0	0	0
N	0	0	0	0	0
end		0	0	0	0

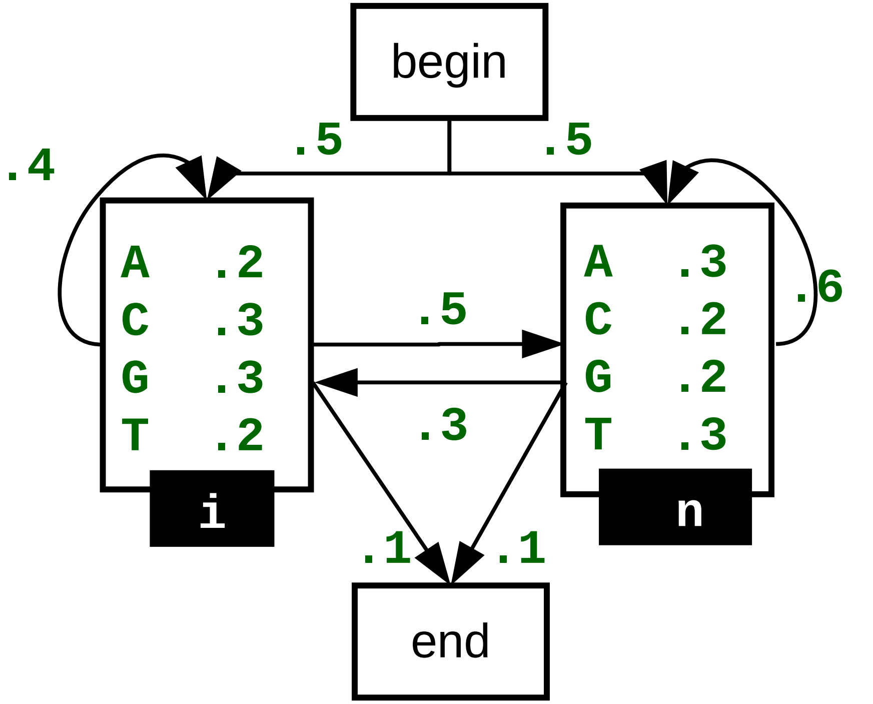


Forward algorithm (ex.)

$$\sum_{s_1 s_2 s_3 \in S^3} p(\mathrm{CAG}, s_1 s_2 s_3) = \sum_{s_1 \in \{i,n\}} p(\mathrm{C}|s_1)\, p(s_1|\mathrm{b}) \sum_{s_2 \in \{i,n\}} p(\mathrm{G}|s_2)\, p(s_2|s_1) \sum_{s_3 \in \{i,n\}} p(\mathrm{A}|s_3)\, p(s_3|s_2)\, p(\mathrm{e}|s_3)$$

$$T[i,j] = \sum_k T[k,i-1] * p(s_i|s_k) * p(x_i|s_i)$$

	ε	C	G	A	ε
begin	**1**	0	0	0	0
I	0	**1** x .5 x .3 0 x .4 x .3 0 x .3 x .3 .15	0	0	0
N	0	**1** x .5 x .2 0 x .5 x .2 0 x .6 x .2 .1	0	0	0
end		0	0	0	0

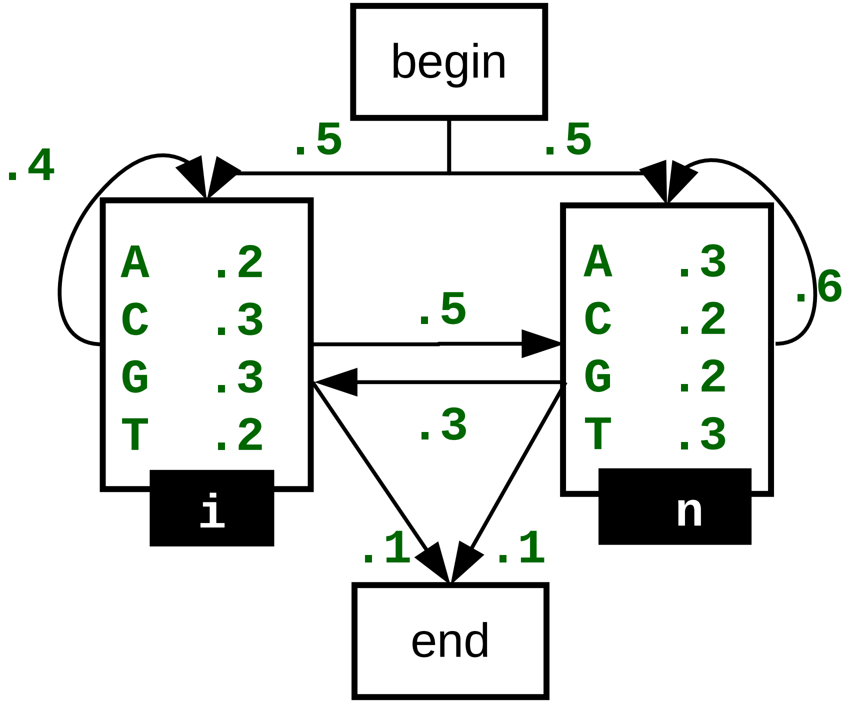


Forward algorithm (ex.)

$$\sum_{s_1 s_2 s_3 \in S^3} p(\mathrm{CAG}, s_1 s_2 s_3) = \sum_{s_1 \in \{\mathrm{i},\mathrm{n}\}} p(\mathrm{C}|s_1)\, p(s_1|\mathrm{b}) \sum_{s_2 \in \{\mathrm{i},\mathrm{n}\}} p(\mathrm{G}|s_2)\, p(s_2|s_1) \sum_{s_3 \in \{\mathrm{i},\mathrm{n}\}} p(\mathrm{A}|s_3)\, p(s_3|s_2)\, p(\mathrm{e}|s_3)$$

$$T[i,j] = \sum_k T[k,i-1] * p(s_i|s_k) * p(x_i|s_i)$$

	ε	C	G	A	ε
begin	**1**	0	0	0	0
I	0	**1** x .5 x .3 0 x .4 x .3 0 x .3 x .3 .15	**0** x .5 x .3 .15 x .4 x .3 ,1 x .3 x .3 .026	0	0
N	0	**1** x .5 x .2 0 x .5 x .2 0 x .6 x .2 .1	**0** x .5 x .2 .15 x .5 x .2 .1 x .6 x .2 .027	0	0
end		0	0	0	0

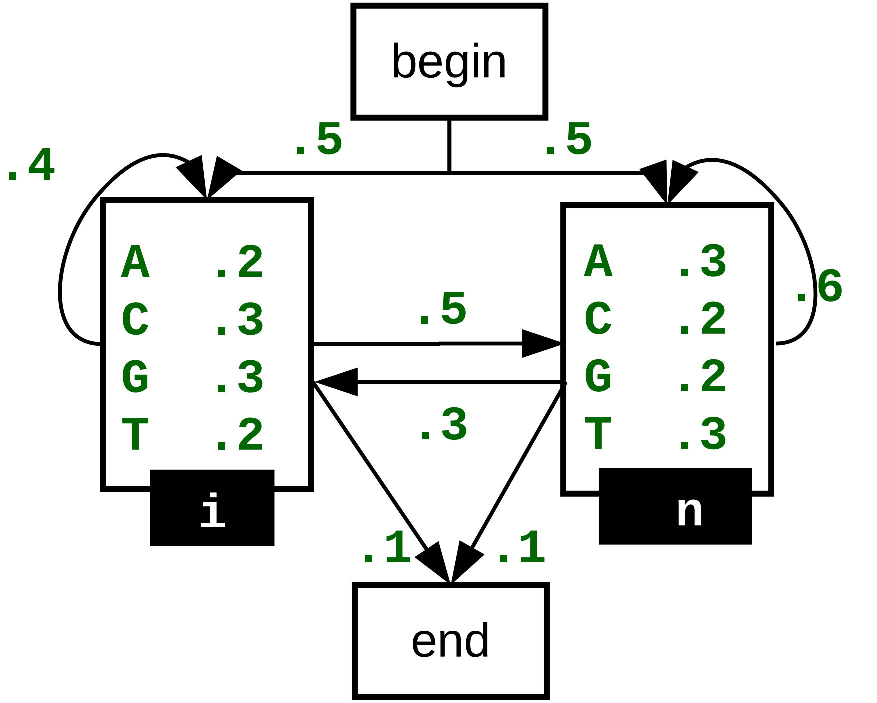


Forward algorithm (ex.)

$$\sum_{s_1 s_2 s_3 \in S^3} p(\mathrm{CAG}, s_1 s_2 s_3) = \sum_{s_1 \in \{i,n\}} p(\mathrm{C}|s_1)\, p(s_1|\mathrm{b}) \sum_{s_2 \in \{i,n\}} p(\mathrm{G}|s_2)\, p(s_2|s_1) \sum_{s_3 \in \{i,n\}} p(\mathrm{A}|s_3)\, p(s_3|s_2)\, p(\mathrm{e}|s_3)$$

$$T[i,j] = \sum_k T[k,i-1] * p(s_i|s_k) * p(x_i|s_i)$$

	ε	C	G	A	ε
begin	**1**	0	0	0	0
I	0	**1** x .5 x .3 0 x .4 x .3 0 x .3 x .3 .15	**0** x .5 x .3 .15 x .4 x .3 ,1 x .3 x .3 .026	**0** x .5 x .2 0.026 x .4 x .2 .027 x .3 x .2 .004	0
N	0	**1** x .5 x .2 0 x .5 x .2 0 x .6 x .2 .1	**0** x .5 x .2 .15 x .5 x .2 .1 x .6 x .2 .027	**0** x .5 x .3 .026 x .5 x .3 .027 x .6 x .3 .009	0
end		0	0	0	0

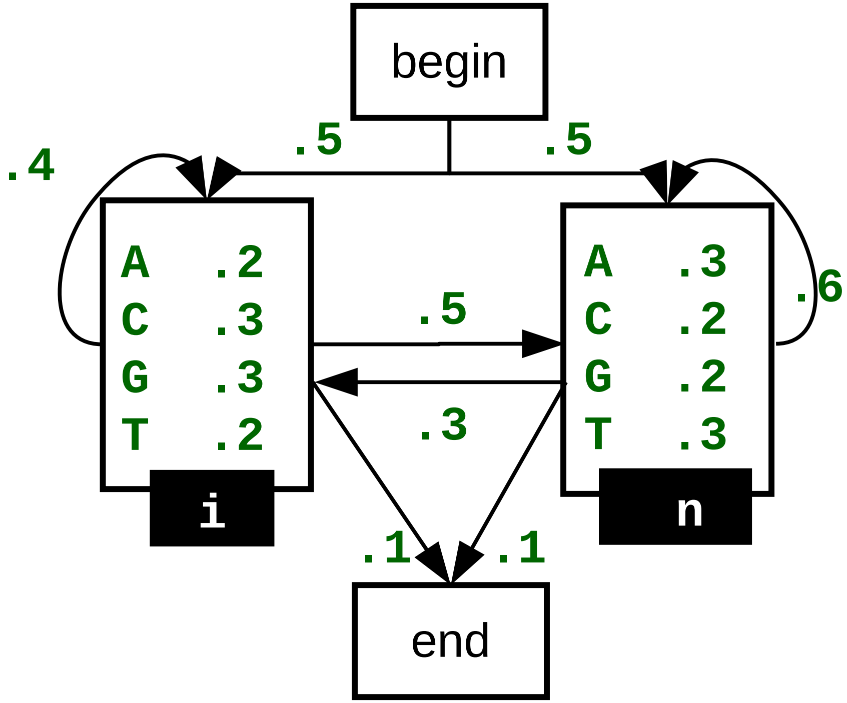


Forward algorithm (ex.)

$$\sum_{s_1 s_2 s_3 \in S^3} p(\mathrm{CAG}, s_1 s_2 s_3) = \sum_{s_1 \in \{i,n\}} p(\mathrm{C}|s_1)\,p(s_1|\mathrm{b}) \sum_{s_2 \in \{i,n\}} p(\mathrm{G}|s_2)\,p(s_2|s_1) \sum_{s_3 \in \{i,n\}} p(\mathrm{A}|s_3)\,p(s_3|s_2)\,p(\mathrm{e}|s_3)$$

$$T[i,j] = \sum_k T[k,i-1] * p(s_i|s_k) * p(x_i|s_i)$$

	ε	C	G	A	ε
begin	**1**	0	0	0	0
I	0	**1** x .5 x .3 0 x .4 x .3 0 x .3 x .3 .15	**0** x .5 x .3 .15 x .4 x .3 .1 x .3 x .3 .026	**0** x .5 x .2 0.026 x .4 x .2 .027 x .3 x .2 .004	0
N	0	**1** x .5 x .2 0 x .5 x .2 0 x .6 x .2 .1	**0** x .5 x .2 .15 x .5 x .2 .1 x .6 x .2 .027	**0** x .5 x .3 .026 x .5 x .3 .027 x .6 x .3 .009	0
end		0	0	0	**0** x .5 .004 x .1 .009 x .1 .0013



Profile HMM – Excercise

bat	AG---C
rat	A-AA-C
cat	AG-AA-
dog	-GAC-C
fox	AC---G
	12---3
ass	AGG

- Make a profile model *M*
- Compute the probability that a new (non-aligned) sequence has been generated by *M*:

$$P(AGG|M) = p_M(A|s_1)p_M(s_1)p_M(G|s_2)p_M(s_2|s_1)p_M(G|s_3)p_M(s_3|s_2)$$

$$T[i,j] = \sum_k T[k,i-1] * p(s_i|s_k) * p(x_i|s_i)$$

$$T[i,j] = \sum_k T[k,i] * p(s_i|s_k)$$

Profile HMM – Excercise

```
bat    AG----
rat    A-AA-C
cat    AG-AA-
dog    -GAC-C
fox    AC---G
       12---3


ass    AGG
```

- Make a profile model *M*
- Employ M to align the sequence.

$$P(AGG|M) = p_M(A|s_1)p_M(s_1)p_M(G|s_2)p_M(s_2|s_1)p_M(G|s_3)p_M(s_3|s_2)$$

$$\begin{aligned} T[i,j] &= max_k T[k,i-1] + \log p(s_i|s_k) + \log p(x_i|s_i) \\ S[i,j] &= argmax_k S[k,i-1] + \log p(s_i|s_k) + \log p(x_i|s_i) \end{aligned}$$

$$\begin{aligned} T[i,j] &= max_k T[k,i] + \log p(s_i|s_k) \\ S[i,j] &= max_k T[k,i] + \log p(s_i|s_k) \end{aligned}$$

Sum-up

- Sequence categorization into family of sequences (Forward alg.)
- Sequence anotation: CpG detection, gene finding (Viterbi alg.)
- Learning ***hidden*** parameters (Baum-Welsh alg.)