Genome 540 Class 17

Chengxiang Qiu

HW7 questions?

Agenda

- HW8 questions?
- Baum-Welch (forward-backward) algorithm example
- GENSCAN
- HW9 Introduction

A general definition of HMM

$$HMM = (S, V, B, A, \Pi)$$

N states

$$S = \{s_1, \ldots, s_N\}$$

M symbols

$$V = \{v_1, \ldots, v_M\}$$

Initial state probability:

$$\Pi = \{\pi_1, \ldots, \pi_N\} \quad \sum_{i=1}^{N} \pi_i = 1$$

$$\pi_i : \textit{prob of starting at state } s_i$$

State transition probability:

$$A = \{a_{ij}\} \quad 1 \le i, j \le N \quad \sum_{j=1}^{N} a_{ij} = 1$$

$$a_{ij} : \textit{prob of going } s_i \rightarrow s_j$$

Output probability:

$$B = \{b_i(v_k)\} \quad 1 \le i \le N, 1 \le k \le M \quad \sum_{k=1}^{M} b_i(v_k) = 1$$

$$b_i(v_k) : \textit{prob of "generating" } v_k \textit{ at } s_i$$

Forward variables

Backward variables

Forward Algorithm

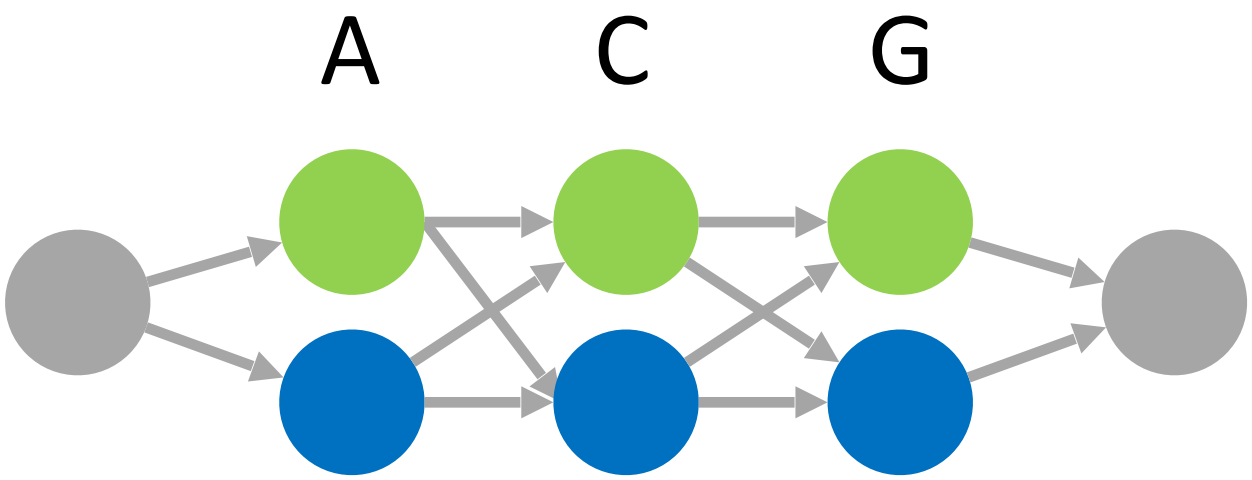
1. Initialization:

$$\alpha_1(i) = \pi_i b_i(O_1), \qquad 1 \le i \le N$$

2. Induction:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(O_{t+1}), \qquad 1 \le t \le T-1, 1 \le j \le N.$$

Build a dynamic programming table for these calculations



	A	C	G
State 1			
State 2			

Forward Algorithm

1. Initialization:

$$\alpha_1(i) = \pi_i b_i(O_1), \qquad 1 \le i \le N$$

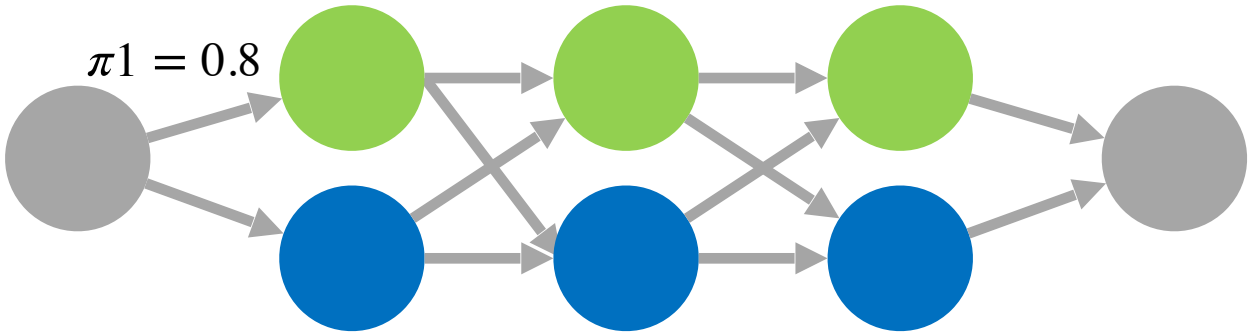
2. Induction:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(O_{t+1}), \qquad 1 \le t \le T-1, 1 \le j \le N.$$

Build a dynamic programming table for these calculations

b_1(A) = 0.4

A C G



$\alpha_1(1)$

	A	C	G
State 1	0.32		
State 2			

Forward Algorithm

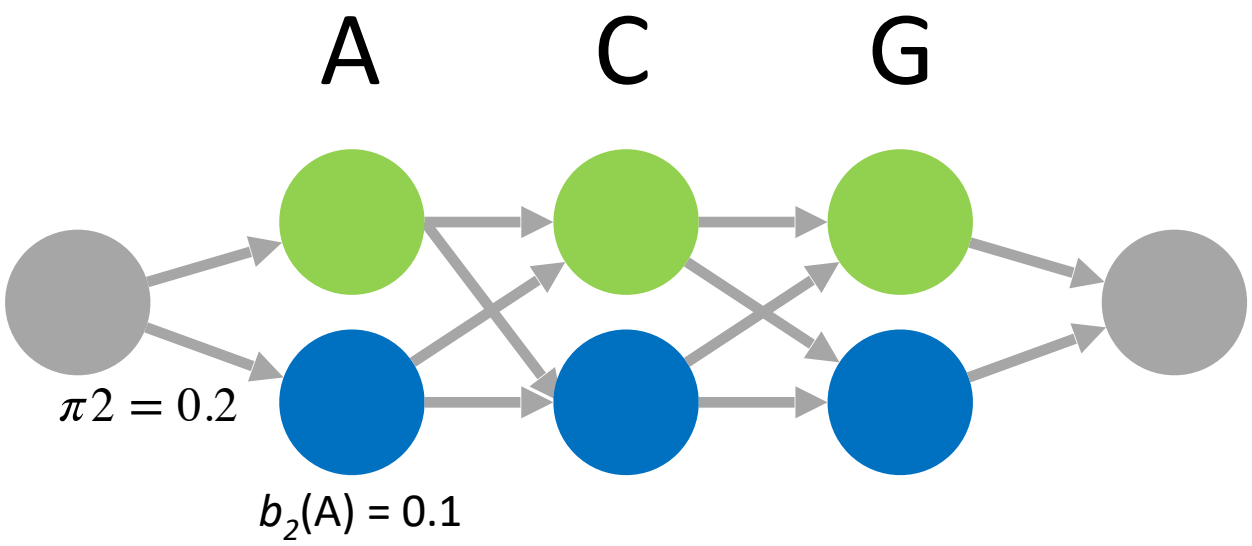
1. Initialization:

$$\alpha_1(i) = \pi_i b_i(O_1), \qquad 1 \le i \le N$$

2. Induction:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(O_{t+1}), \qquad 1 \le t \le T-1, 1 \le j \le N.$$

Build a dynamic programming table for these calculations



$\alpha_1(1)$

	A	C	G
State 1	0.32		
State 2	0.02		

$\alpha_1(2)$

Forward Algorithm

1. Initialization:

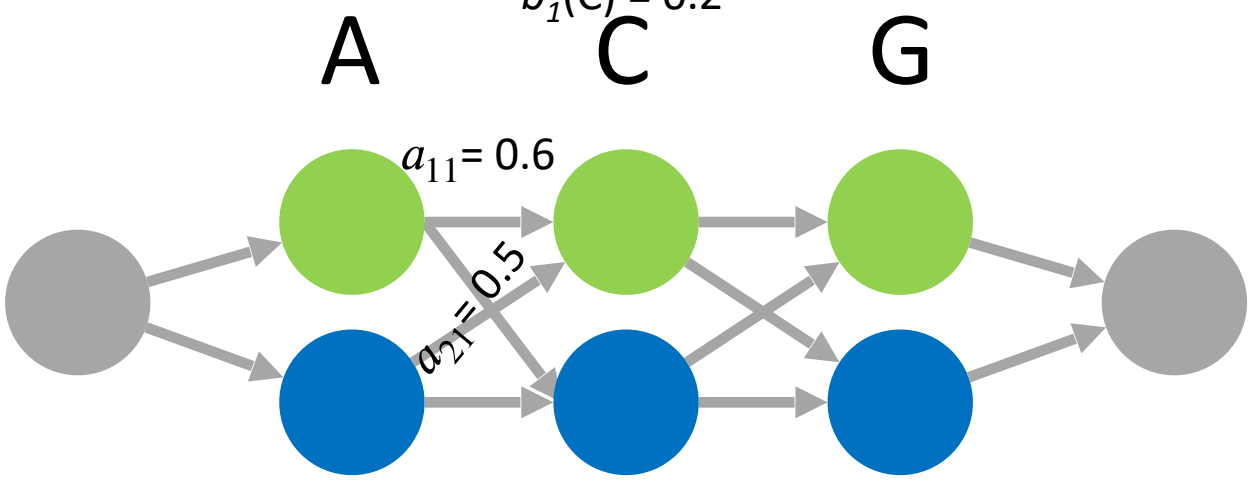
$$\alpha_1(i) = \pi_i b_i(O_1), \qquad 1 \le i \le N$$

2. Induction:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(O_{t+1}), \qquad 1 \le t \le T-1, 1 \le j \le N.$$

Build a dynamic programming table for these calculations

b_1(C) = 0.2



$\alpha_1(1)$ $\alpha_2(1)$

	A	C	G
State 1	0.32	0.0404	
State 2	0.02		

$\alpha_1(2)$

Forward Algorithm

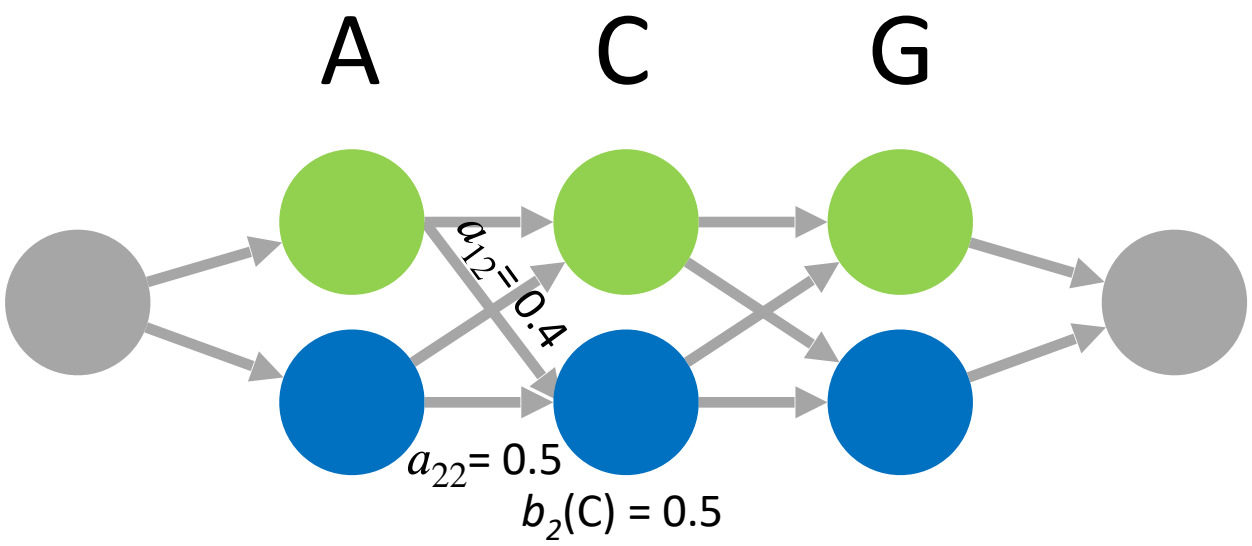
1. Initialization:

$$\alpha_1(i) = \pi_i b_i(O_1), \qquad 1 \le i \le N$$

2. Induction:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(O_{t+1}), \qquad 1 \le t \le T-1, 1 \le j \le N.$$

Build a dynamic programming table for these calculations



	$\alpha_1(1)$	$\alpha_2(1)$	$\alpha_3(1)$
	A	C	G
State 1	0.32	0.0404	
State 2	0.02	0.069	

	$\alpha_1(2)$	$\alpha_2(2)$	$\alpha_3(2)$

Backward Algorithm

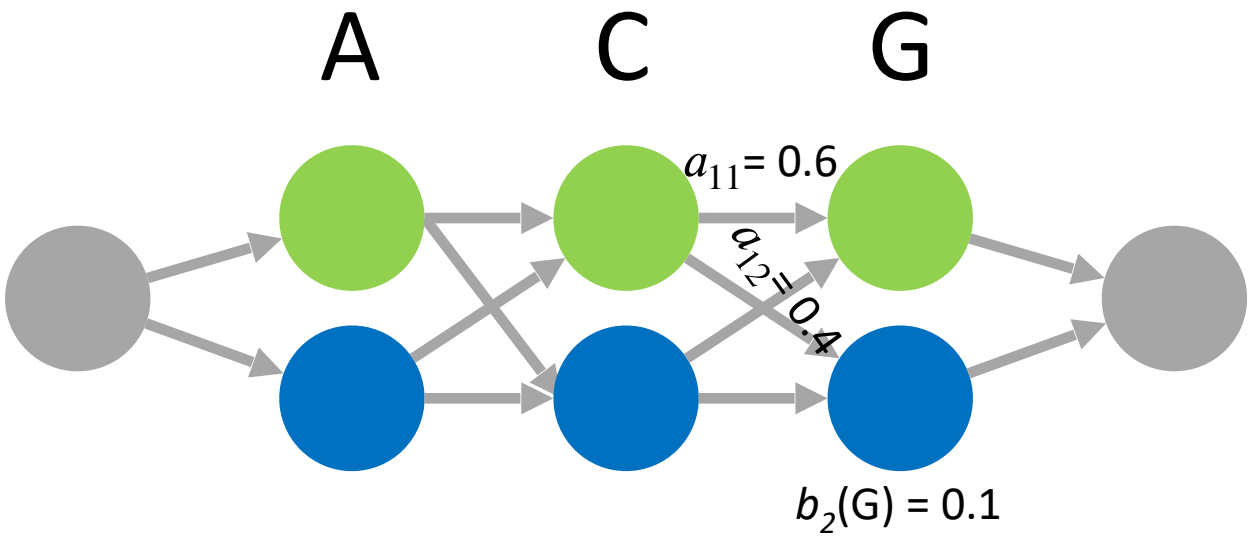
1. Initialization:

$$\beta_T(i) = 1, \qquad 1 \le i \le N$$

2. Induction:

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j), \qquad 1 \le t \le T-1, 1 \le j \le N.$$

Build a dynamic programming table for these calculations



	$\beta_1(1)$	$\beta_2(1)$	$\beta_3(1)$
	A	C	G
State 1		0.28	1
State 2			1
	$\beta_1(2)$	$\beta_2(2)$	$\beta_3(2)$

Backward Algorithm

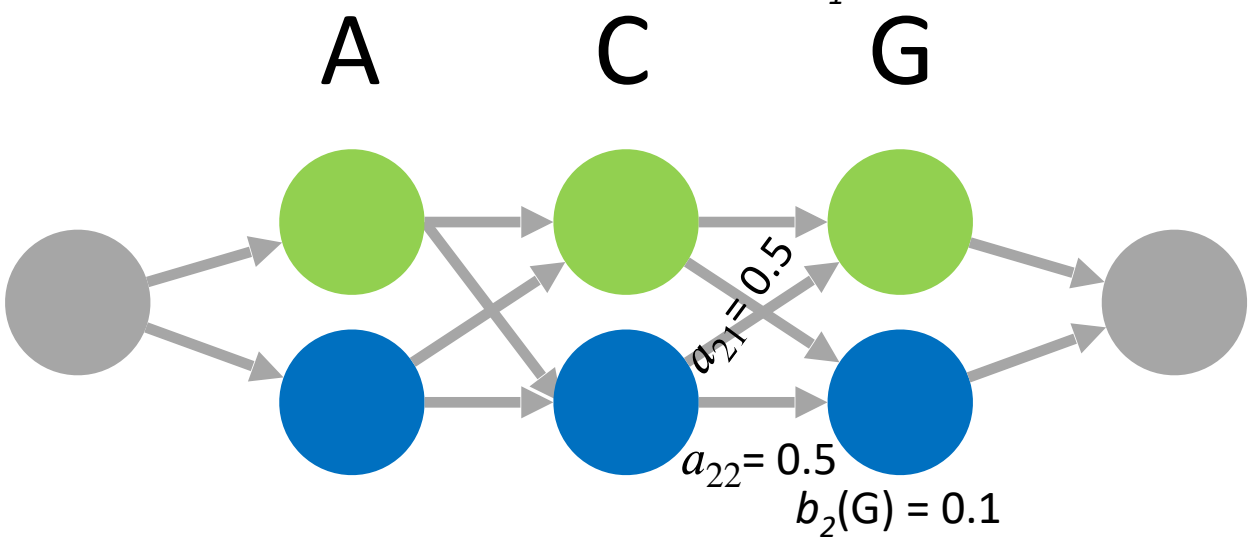
1. Initialization:

$$\beta_T(i) = 1, \qquad 1 \le i \le N$$

2. Induction:

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j), \qquad 1 \le t \le T-1, 1 \le j \le N.$$

Build a dynamic programming table for these calculations



	$\beta_1(1)$	$\beta_2(1)$	$\beta_3(1)$
	A	C	G
State 1		0.28	1
State 2		0.25	1
	$\beta_1(2)$	**$\beta_2(2)$**	$\beta_3(2)$

Scale

	A	C	G
State 1			
State 2			
	c1	c2	c3

Forward

- Initialization

$$\begin{aligned} \ddot{\alpha}_1(i) &= \alpha_1(i) \\ c_1 &= \frac{1}{\sum_{i=1}^{N} \ddot{\alpha}_1(i)} \\ \hat{\alpha}_1(i) &= c_1 \ddot{\alpha}_1(i) \end{aligned}$$

- Induction

$$\begin{aligned} \ddot{\alpha}_t(i) &= \sum_{j=1}^{N} \hat{\alpha}_{t-1}(j) a_{ji} b_i(O_t) \\ c_t &= \frac{1}{\sum_{i=1}^{N} \ddot{\alpha}_t(i)} \\ \hat{\alpha}_t(i) &= c_t \ddot{\alpha}_t(i) \end{aligned}$$

$$\hat{\alpha}_t(i) = \left(\prod_{\tau=1}^{t} c_\tau \right) \alpha_t(i).$$

$$\mathbf{C}_t = \prod_{\tau=1}^{t} c_\tau$$

$$\log[P(O|\lambda)] = -\sum_{t=1}^{T} \log c_t.$$

Scale

Backward

- Initialization

$$\begin{aligned} \ddot{\beta}_T(i) &= 1 \\ \hat{\beta}_T(i) &= c_T \ddot{\beta}_T(i) \end{aligned}$$

$$\hat{\beta}_t(i) = \left(\prod_{s=t}^{T} c_s \right) \beta_t(i) = \mathbf{D}_t \beta_t(i),$$

- Induction

$$\begin{aligned} \ddot{\beta}_t(i) &= \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \hat{\beta}_{t+1}(j) \\ \hat{\beta}_t(i) &= c_t \ddot{\beta}_t(i) \end{aligned}$$

Scale

$$
\begin{aligned}
\bar{a}_{ij} &= \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \\
&= \frac{\sum_{t=1}^{T-1} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{t=1}^{T-1} \alpha_t(i)\beta_t(i)} \\
&= \frac{\sum_{t=1}^{T-1} \hat{\alpha}_t(i)/\mathbf{C}_t \cdot a_{ij} b_j(O_{t+1}) \cdot \hat{\beta}_{t+1}(j)/\mathbf{D}_{t+1}}{\sum_{t=1}^{T-1} \hat{\alpha}_t(i)/\mathbf{C}_t \cdot \hat{\beta}_t(i)/\mathbf{D}_t} \\
&= \frac{\left(\sum_{t=1}^{T-1} \hat{\alpha}_t(i) \cdot a_{ij} b_j(O_{t+1}) \cdot \hat{\beta}_{t+1}(j)\right)/\mathbf{C}_T}{\left(\sum_{t=1}^{T-1} \hat{\alpha}_t(i) \cdot \hat{\beta}_t(i)/c_t\right)/\mathbf{C}_T} \\
&= \frac{\sum_{t=1}^{T-1} \hat{\alpha}_t(i) \cdot a_{ij} b_j(O_{t+1}) \cdot \hat{\beta}_{t+1}(j)}{\sum_{t=1}^{T-1} \hat{\alpha}_t(i) \cdot \hat{\beta}_t(i)/c_t}.
\end{aligned}
$$

$$
\begin{aligned}
\bar{b}_j(k) &= \frac{\sum_{t=1, O_t=v_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)} \\
&= \frac{\sum_{t=1, O_t=v_k}^{T} \hat{\alpha}_t(j) \cdot \hat{\beta}_t(j)/c_t}{\sum_{t=1}^{T} \hat{\alpha}_t(j) \cdot \hat{\beta}_t(j)/c_t}.
\end{aligned}
$$

initial prob
Pi(i) = alpha_hat 1(i) * beta_hat 1(i) / c1

$$
\log[P(O|\lambda)] = -\sum_{t=1}^{T} \log c_t.
$$

HW8 tips

- Calculate first few steps by hand and make sure your program matches (exactly!)
- Create other small test cases
- Avoid underflow
  - Scale
  - Take the logarithm
- Let me know if any questions!

HW9: Evolutionarily conserved segments
Due Sunday March-13 11:59pm

- ENCODE region 010 (chromosome 7)
- Multiple alignment of human, dog, and mouse
- 2 states: neutral (fast-evolving), conserved (slow-evolving)
- Emitted symbols are multiple alignment columns (e.g. ‘AAT’)
- Viterbi parse (no iteration)

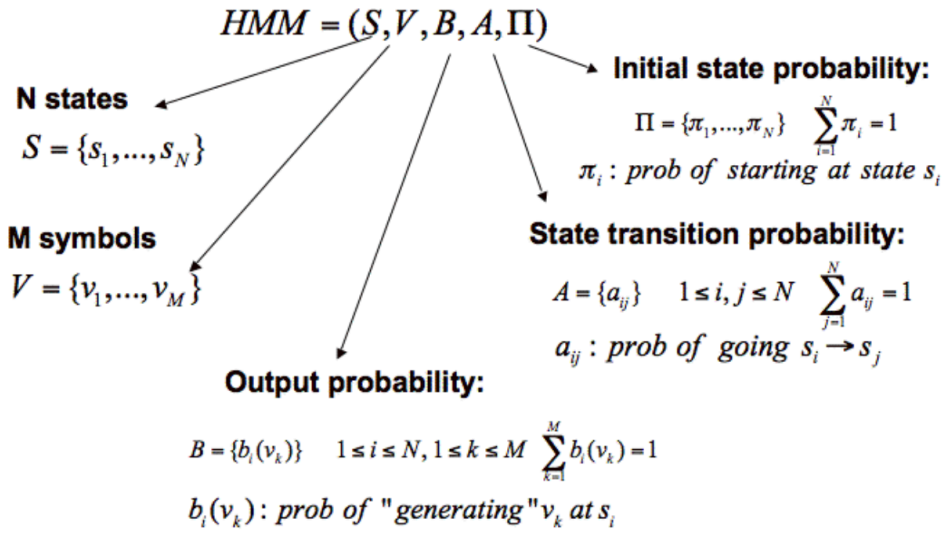
Input data

```
# chr7:26924045-26924056
hg18------TGCTCACATTTT
canFam2---CTCACAGTTT
mm9-------CGCTT-

# chr7:26924057-26924120
hg18------CTAGAAGGATTAATGTTCTGTAGATCTATTGATCTTCTACAT
canFam2-TCAGAGGGATTAGTGTTCTGTGGATCTATTGATCTTCTGCAC
mm9-CCAGAGGGAGTGGTGTTCTGTAGATCTATCGACCTTC--CACGCAG

# chr7:26924121-26924289
hg18------ATCATTAACAATACTTTGTTTTGATTTACTTGCCTGGTGTCT
canFam2-ATCATTAGCAACACTTTGTTCTGATCTACTTGCCTGTCATCC
mm9-------------ACTTCGCTCTGCTCCACTTGCCTGACATCCAAGG

# chr7:26924290-26924313
hg18------AATCTAATGTTTAGATTAGGGTTA
canFam2-------------------------
mm9-----------TTAGA-------TA
```

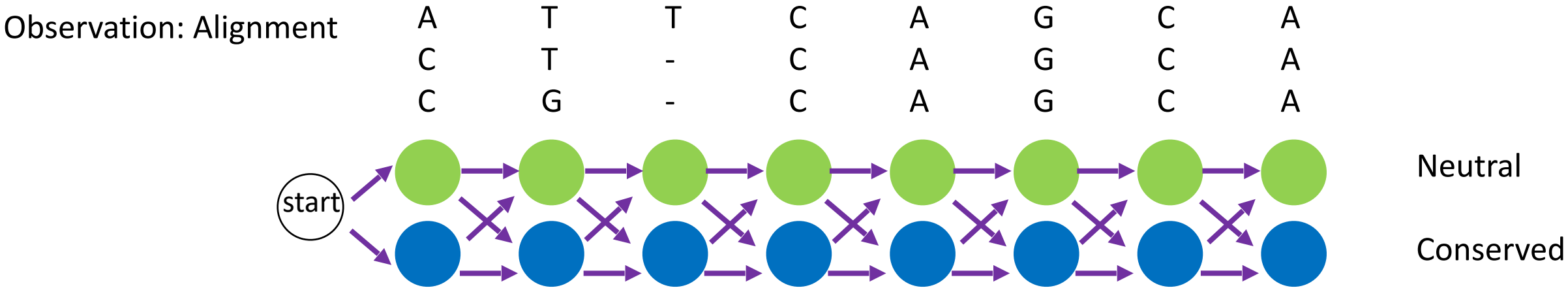


N = 2 states

M = 100 symbols

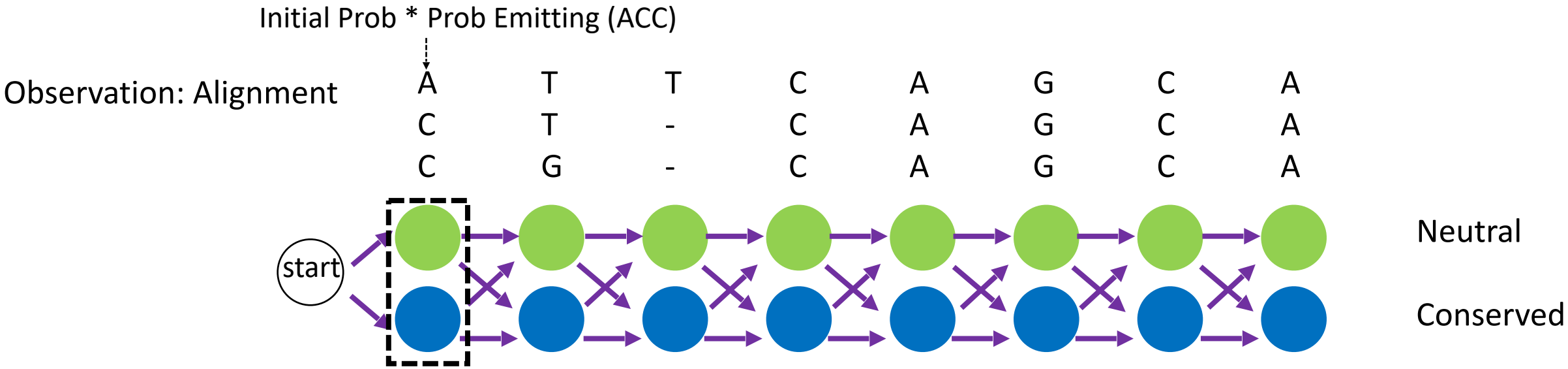
Finding the most likely series of hidden states (Viterbi Path)

- Step 1: given an observed alignment, determine the most probable series of states
  - This depends on the specified probabilities:
    - Initiation
    - Transition
    - Emission
  - Process nodes in a sliding window



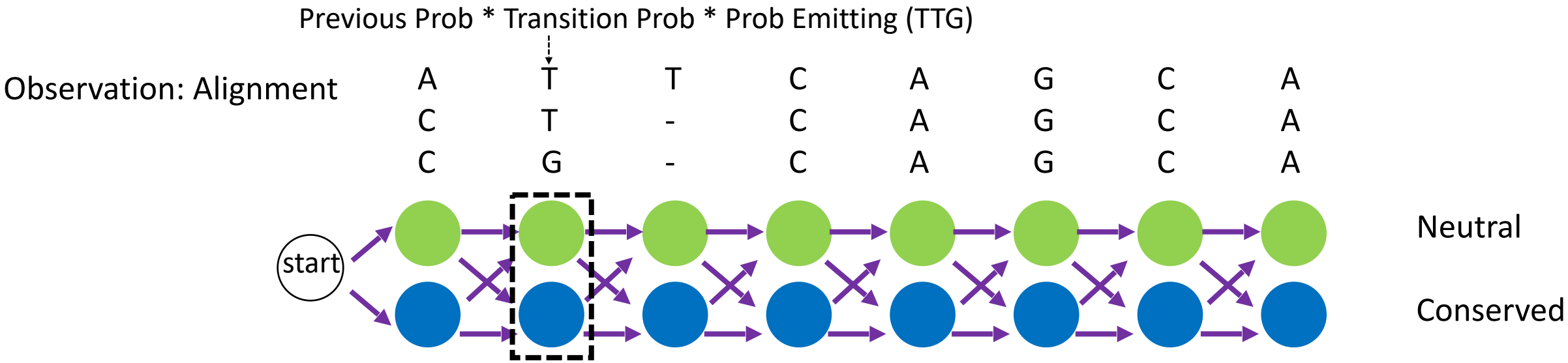
Finding the most likely series of hidden states (Viterbi Path)

- Step 1: given an observed alignment, determine the most probable series of states
  - This depends on the specified probabilities:
    - Initiation
    - Transition
    - Emission
  - Process nodes in a sliding window



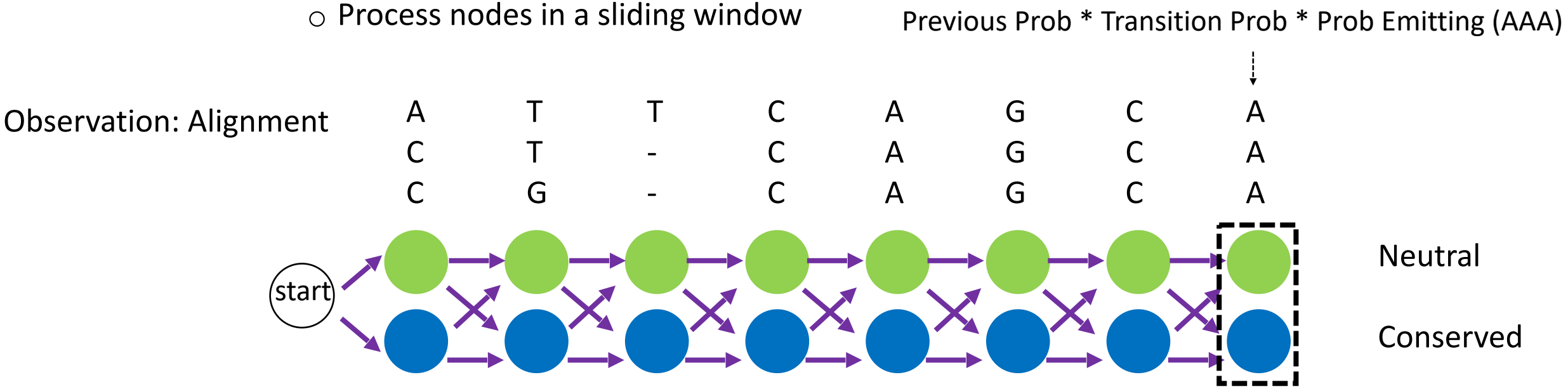
Finding the most likely series of hidden states (Viterbi Path)

- Step 1: given an observed alignment, determine the most probable series of states
  - This depends on the specified probabilities:
    - Initiation
    - Transition
    - Emission
  - Process nodes in a sliding window



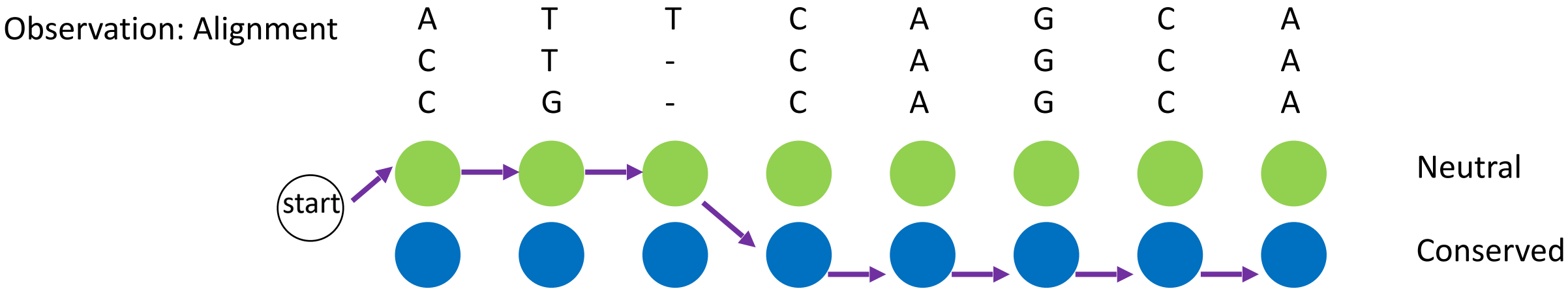
Finding the most likely series of hidden states (Viterbi Path)

- Step 1: given an observed alignment, determine the most probable series of states
  - This depends on the specified probabilities:
    - Initiation
    - Transition
    - Emission
  - Process nodes in a sliding window

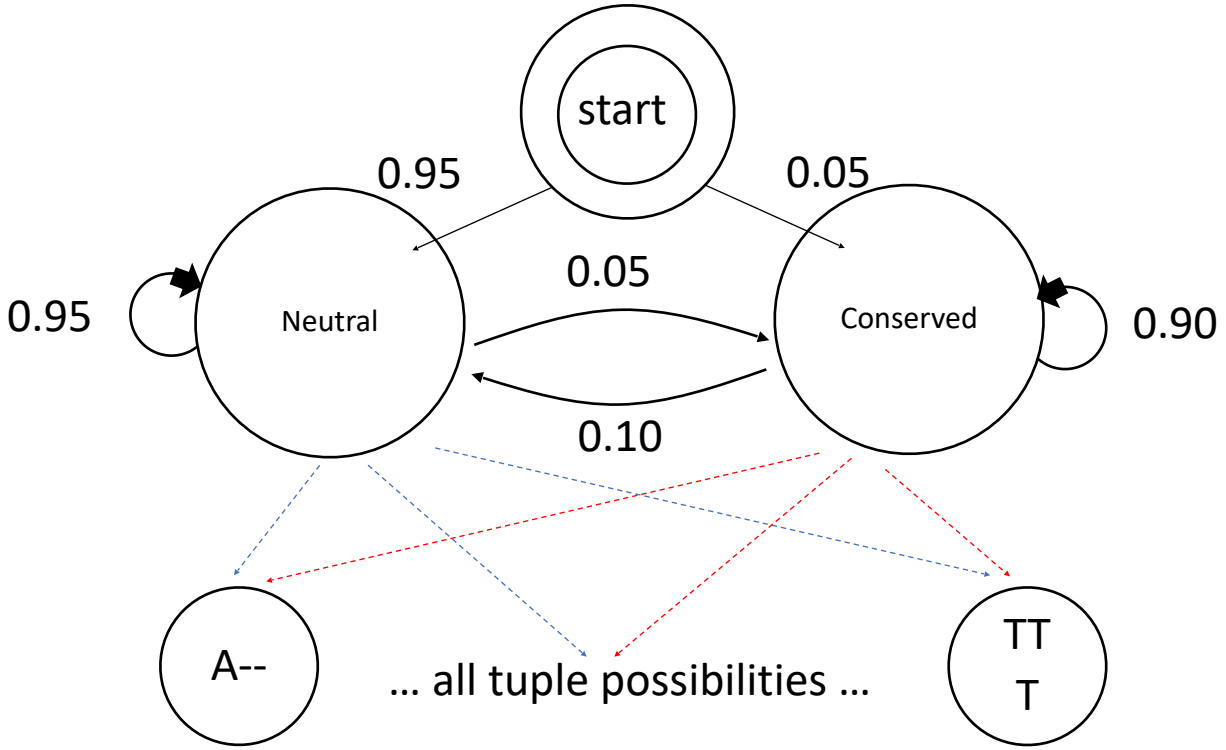


Finding the most likely series of hidden states (Viterbi Path)

- Step 1: given an observed alignment, determine the most probable series of states
  - This depends on the specified probabilities:
    - Initiation
    - Transition
    - Emission
  - Process nodes in a sliding window



HMM Diagram


start
0.95
0.05
0.95
Neutral
0.05
Conserved
0.90
0.10
A--
… all tuple possibilities …
TT
T

Input

- Original maf format
  - Sequences broken into alignment blocks based on the species included
  - Official file format specs
- Homework file format
  - Only 3 species
  - Gaps in human sequence were removed and ambiguous bases replaced with ‘A’ for simplicity

```
# chrX:152767699-152767743
hg18    ATAAAAACATTAAAAAAAATCAGCCACAGGACTTGGTCTTGGACC
canFam2 -----------------------------------------
mm9     -----------------------------------------

# chrX:152767744-152767853
hg18    CAAGTTAGAGCTAGGCCATGCTTGCTTAAAGGAGTGGCTGTAATTTTAAACAAGGCTAGTGGGAAAGT
canFam2 ------------------------------------------------------------------
mm9     ------------------------------------------------------------------
```

Setting parameters

- Emission probabilities
  - Neutral state: observed frequencies in neutral data set
  - Conserved state: observed frequencies in functional data set
- Transition probabilities
  - Given in the assignment; more likely to go from conserved to neutral
- Initiation probabilities
  - Given in the assignment; more likely to start in the neutral state

Calculating Emission Probabilities

Neutral State: Ancient Repeat Sequences

AAA	10222095
AAC	481243
AAT	420185
AAG	1415675
AA-	273456
ACA	852624
ACC	179459
ACT	99493
ACG	167810
AC-	29636
ATA	874547
ATC	113150
ATT	220714
ATG	185789

etc …

1st base: human

2nd base: dog

3rd base: mouse

Conserved State: Putative Functional Sites

AAA	2375583
AAC	21337
AAT	10886
AAG	56328
AA-	3205
ACA	33210
ACC	12122
ACT	2270
ACG	5187
AC-	374
ATA	21805
ATC	2871
ATT	7426
ATG	4369

etc …

Output

- State and segment histograms
- Parameter values
  - Initiation/transition probabilities you were given in the assignment
  - Emission probabilities you calculated from neutral and conserved data sets
- Coordinates of 10 longest conserved segments (report positions relative to the start of the chromosome)
- Brief annotations for the 5 longest conserved segments (look at UCSC genome browser, and make sure using the correct genome version, e.g. hg18)

```
State Histogram:
1=5
2=3

Segment Histogram:
1=2
2=1
```

```
Initial State Probabilities:
1=0.90000
2=0.10000

Transition Probabilities:
1,1=0.99000
1,2=0.01000
2,1=0.20000
2,2=0.80000

Emission Probabilities:
1,A--=0.20000
1,A-A=0.20000
1,A-C=0.20000
1,A-G=0.20000
1,A-T=0.20000
.
.
.
2,A--=0.10000
2,A-A=0.20000
2,A-C=0.25000
2,A-G=0.25000
2,A-T=0.20000
etc..
```

```
Longest Segment List:

116741000 116752000
116745000 116756000
etc.. (give 10 longest from state 2)

Annotations:

Start: 116741000
End: 116752000
Overlaps with exon3 of the protein coding gene cMyc

Start: 116745000
End: 116756000
Overlaps with exon4 of the protein coding gene cMyc

etc.. (give 5 longest)
```

Questions?