Hidden Markov Models

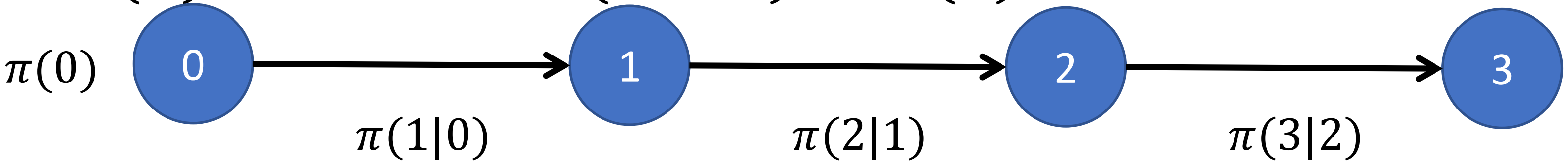
MPA-PRG: Programming in Bioinformatics

Exercise 11

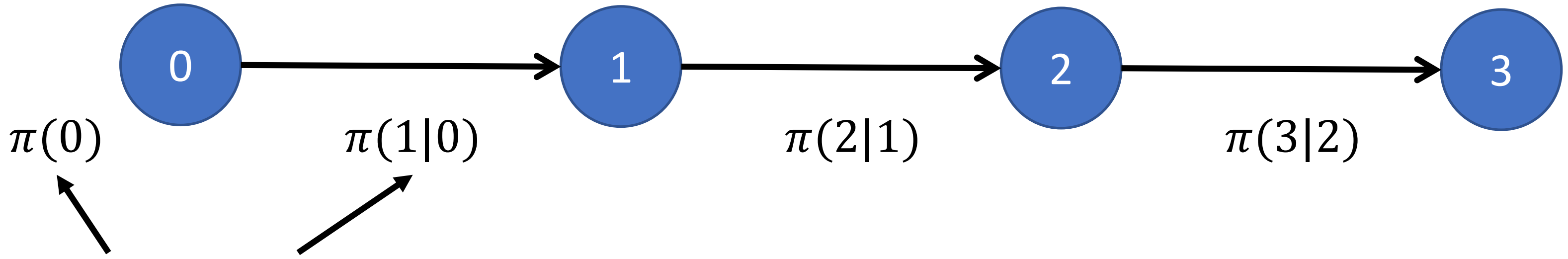
Markov Chain

- in bioinformatics, we deal with individual symbols (nt, aa), therefore we use "discrete time chains"
- Markov property ("memorylessness") – the predictions associated with a Markov process are conditional on its current state and are independent of past and future states
- a Markov chain is described by 2 structures:
    - the probability distribution vector $\pi(n) = [\pi_1(n), \pi_2(n), \dots \pi_I(n)]$ for $n = 0, 1, 2, \dots,$ where $\pi_i(n)$ denotes the probability that the process is in state i at time n
    - the transition probability matrix $P(n) = [p_{ij}(n)]$, where $i = 1, 2, \dots\ I$ and $j = 1, 2, \dots\ I$
- we can describe the Markov chain using the following formula:

$$\pi(n+1) = \pi(n) \times P \quad \leftrightarrow \quad \pi(n+1) = \pi(0) \times\ P^{n+1}$$



Markov Chain



this is a vector

e.g. for 2 states (sunny × rainy day, coding × non-coding sequence)

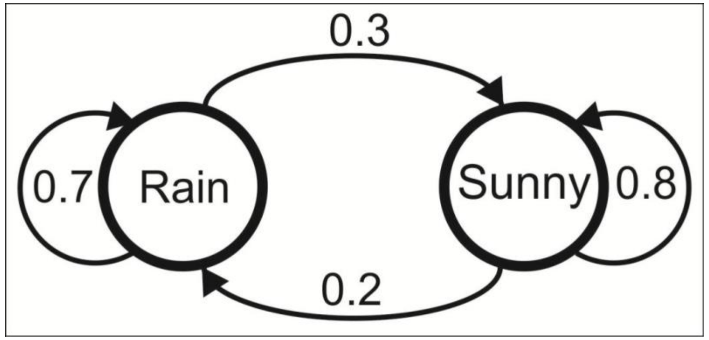
$$\pi(0) = [\pi_1(0), \pi_2(0)]$$

transition probability matrix:

$$P = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix}$$

Markov Model

RRRSSSSRRSSRRRRRSSSSS



$$P = \begin{bmatrix} 0.8 & 0.2 \\ 0.3 & 0.7 \end{bmatrix}$$

Example of MM Aplication

- decide whether the `ATGCC` sequence is a CpG island

$(A \to T) \log \frac{0{,}12}{0{,}21} + (T \to G) \log \frac{0{,}38}{0{,}29} + (G \to C) \log \frac{0{,}34}{0{,}25} + (C \to C) \log \frac{0{,}37}{0{,}30} = 0{,}035 \boxed{>0}$

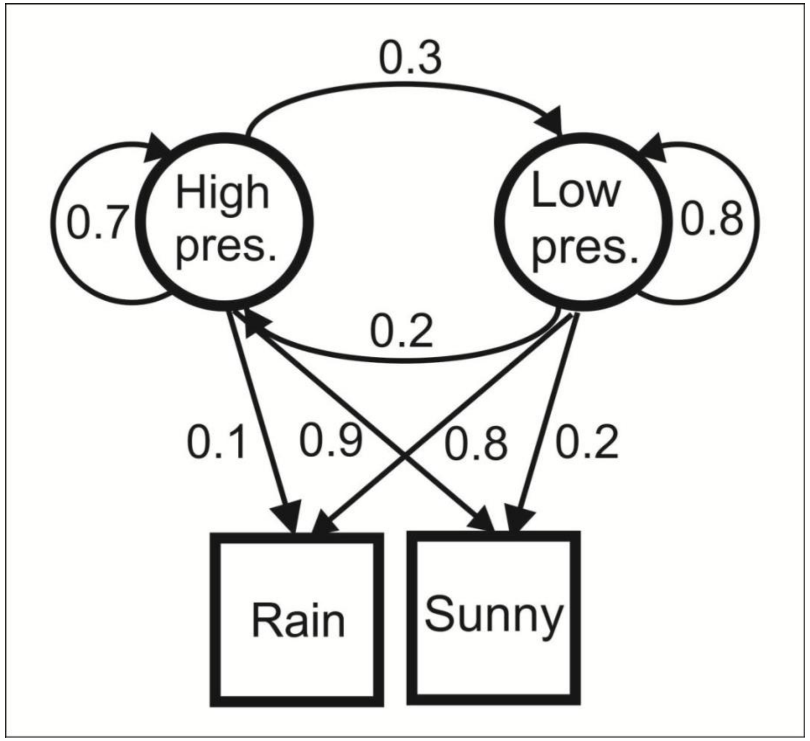
- if $\log \frac{p(x|+)}{p(x|-)} > 0$, then it belongs to a CpG island

$$P^+ = \left[\begin{array}{ccccc} & A & C & G & T \\ A & 0.18 & 0.27 & 0.43 & 0.12 \\ C & 0.17 & 0.37 & 0.27 & 0.19 \\ G & 0.16 & 0.34 & 0.37 & 0.13 \\ T & 0.08 & 0.36 & 0.38 & 0.18 \end{array}\right] \qquad P^- = \left[\begin{array}{ccccc} & A & C & G & T \\ A & 0.30 & 0.20 & 0.29 & 0.21 \\ C & 0.32 & 0.30 & 0.08 & 0.30 \\ G & 0.25 & 0.25 & 0.29 & 0.21 \\ T & 0.18 & 0.24 & 0.29 & 0.29 \end{array}\right]$$

Hidden Markov Model

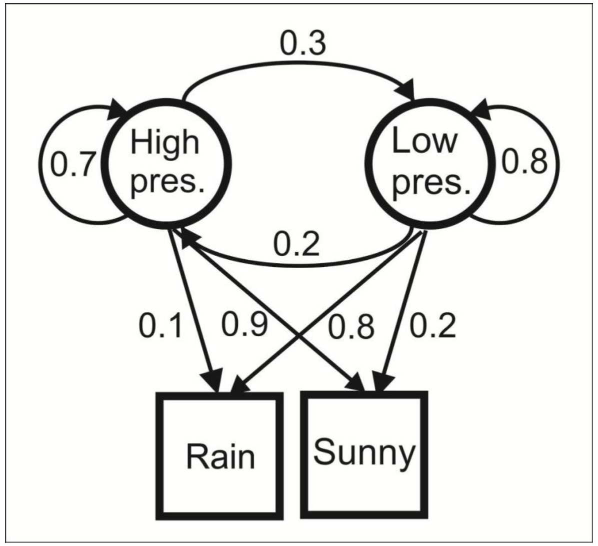
- we have states that can generate different characters of the string with a defined probability

- the states themselves are hidden, we only observe the characters that are generated by those states



HMM

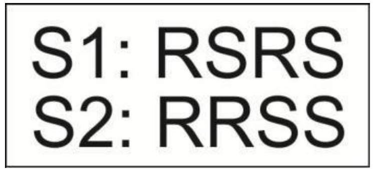
- definition:
  - HMM is a stochastic automaton $HMM = (N, M, A, B, \pi)$



- more user friendly:
  - one more matrix is needed for the definition than for classical Markov models, i.e. a total of 2 matrices and 1 vector
    - initial state – the probability distribution vector (π)
    - the transition probability matrix between hidden states (A, type $|N| \times |N|$)
    - the emission probability matrix of observations by hidden states (B, type $|N| \times |M|$), where *N* is set of hidden states and *M* is a set of emitted observations

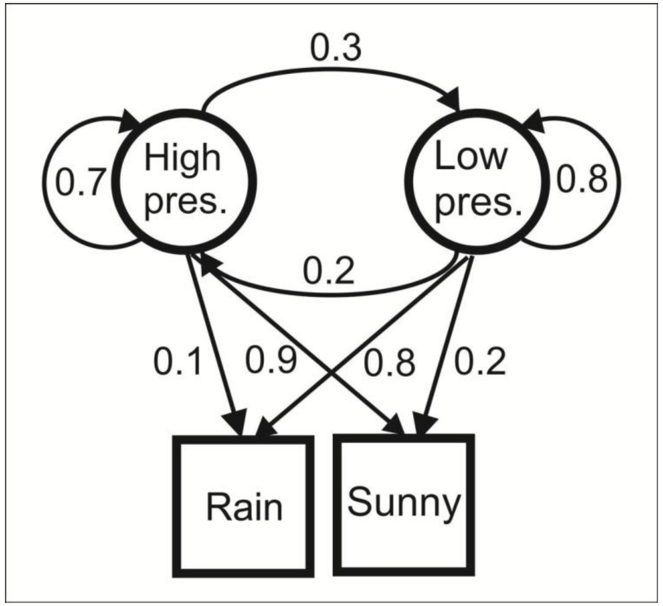
HMM

- initial state: $\pi = [0, 1]$
- 2 different observations S1 and S2:



S1: RSRS
S2: RRSS

- HMM:



HMM

- what might we be interested in?
  - how likely is it for the given model to generate a given sequence S1 or S2 → model evaluation
  - what is the most likely path through hidden states when generating a given sequence → model decoding
  - how to change the model parameters so that it generates a given sequence with higher probability → model learning

HMM Algorithms

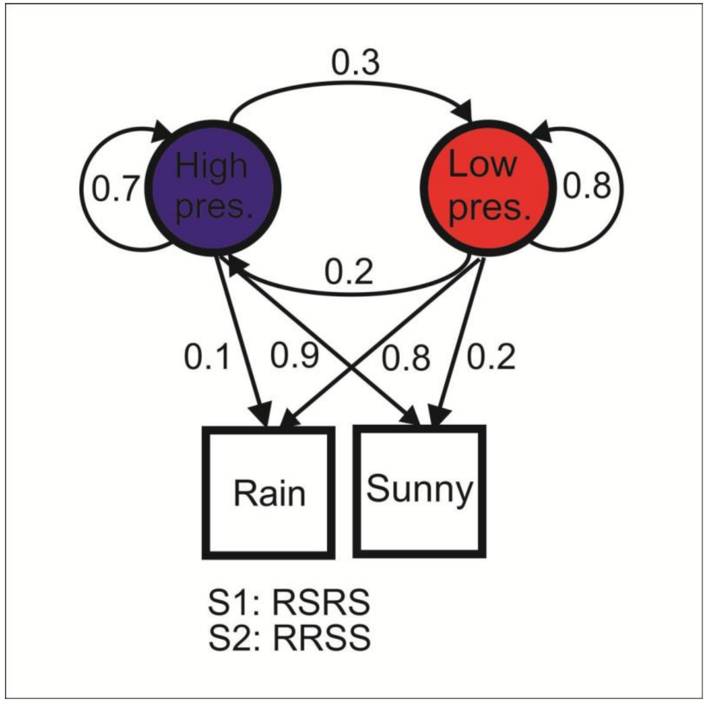
- evaluation: the Forward-Backward algorithm

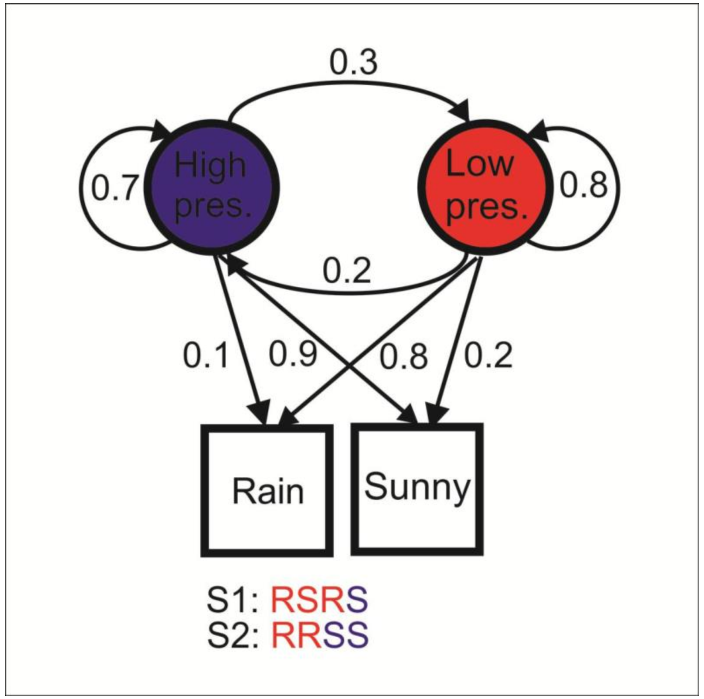
- decoding: the Viterbi algorithm

- learning: the Baum-Welch algorithm

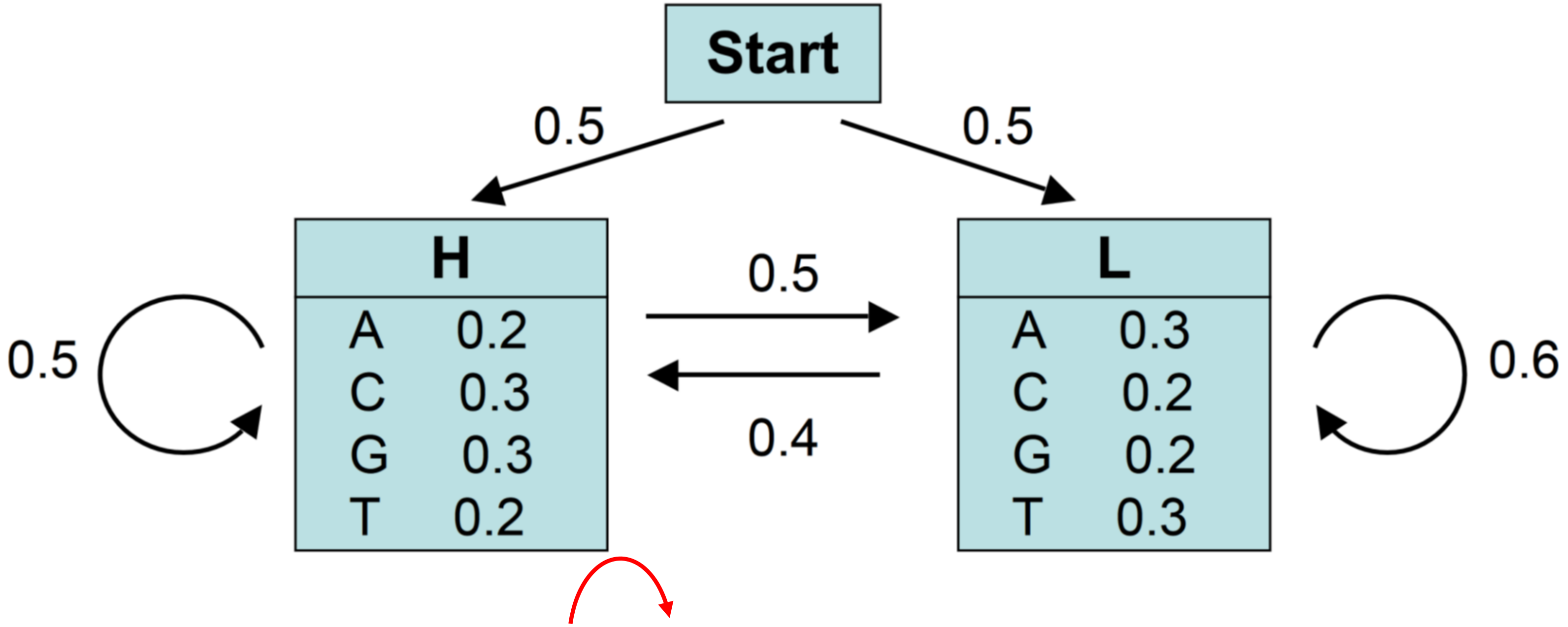
Decoding of HMM – The Viterbi Algorithm

- a dynamical programming algorithm that compute the most probable path through hidden states for given set of observations





The Viterbi Algorithm



G G C A C T G A A

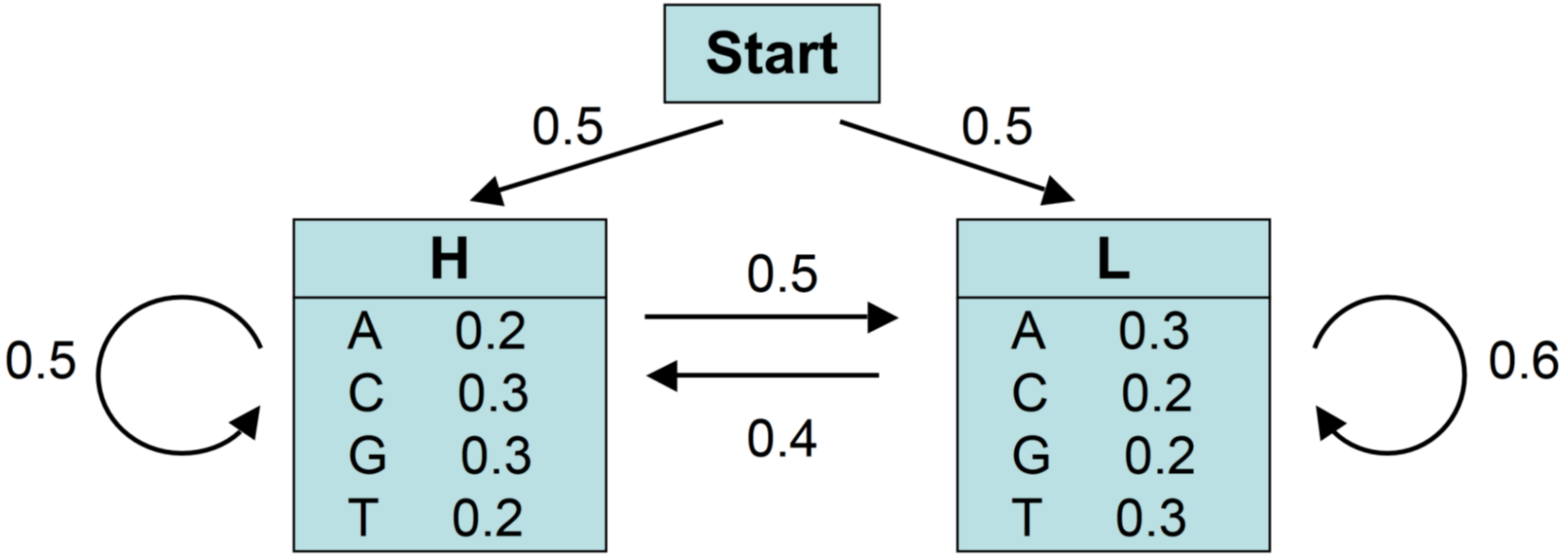
probability of the transition from state *l* to state *k*

$$p_l(i,x) = e_l(i)\max_k(p_k(j,x-1)\cdot p_{kl})$$

probability to observe element *i* in state *l*

probability of the most probable path ending at position x-1 in state *k* with element *j*

The Viterbi Algorithm



G G C A C T G A A

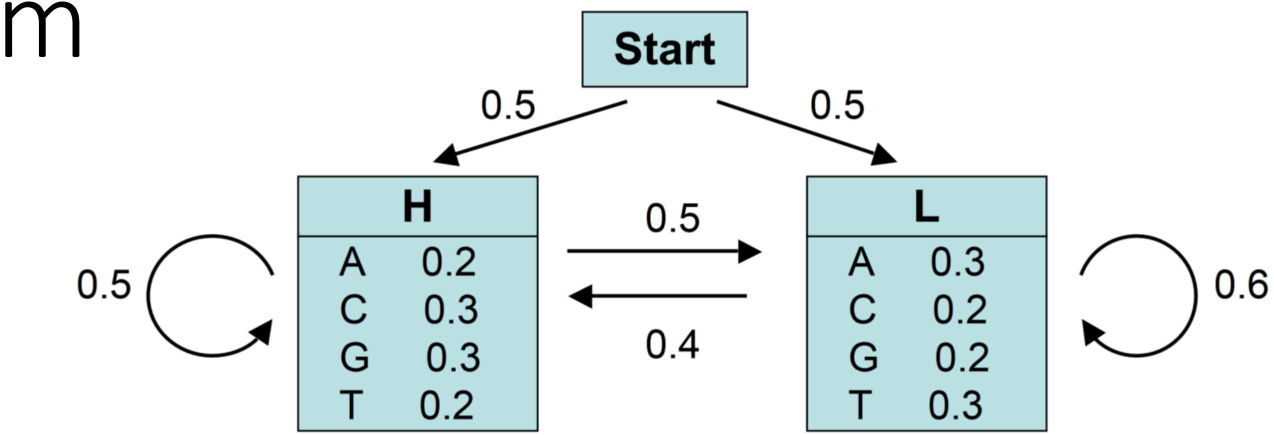
probability of the transition from state L to state H

$$p_H(A,4) = e_H(A)\max(p_L(C,3)\cdot p_{LH}, p_H(C,3)\cdot p_{HH})$$

probability to observe element A in state H

probability of the most probable path ending at position 3 in state L with element C

The Viterbi Algorithm

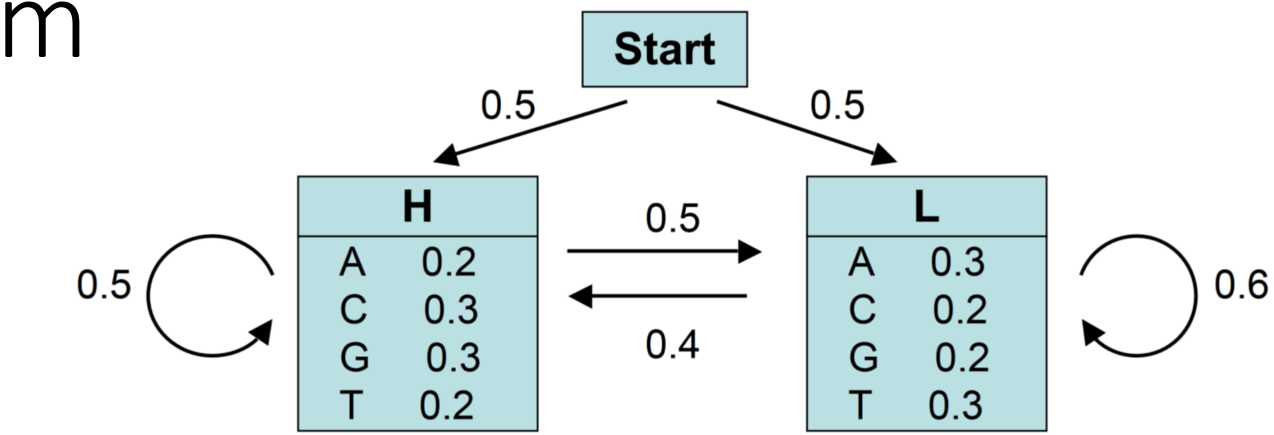


G G C A C T G A A

$$p_l(i,x) = e_l(i)\max_k(p_k(j,x-1)\cdot p_{kl})$$

$$p_H(G,1) = e_H(G)\cdot p_{SH} = 0.3\cdot 0.5 = 0.15$$
$$p_L(G,1) = e_L(G)\cdot p_{SL} = 0.2\cdot 0.5 = 0.10$$

The Viterbi Algorithm



G G C A C T G A A

$$p_l(i,x) = e_l(i)\max_k(p_k(j,x-1)\cdot p_{kl})$$

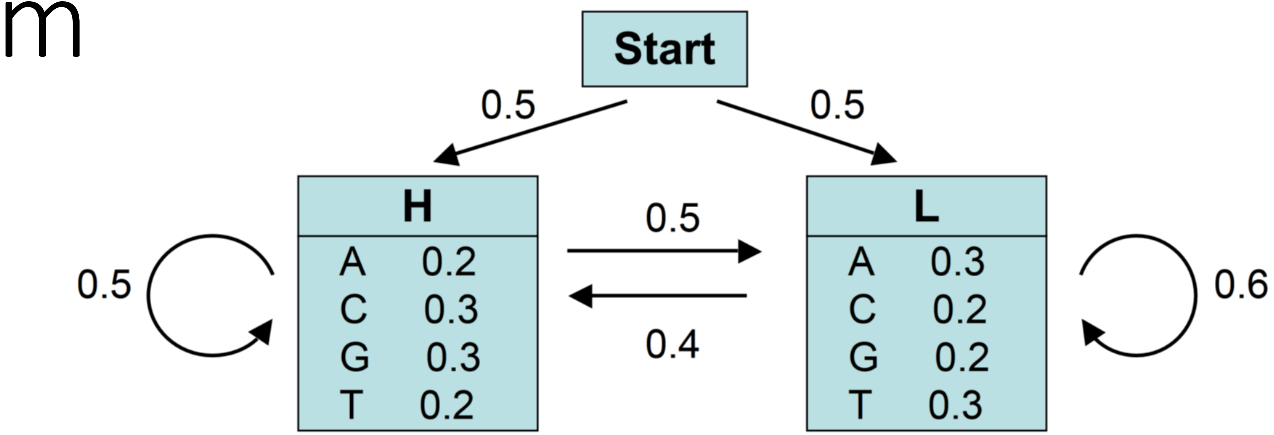
$$p_H(G,2) = e_H(G)\max(p_H(G,1)\cdot p_{HH}, p_L(G,1)\cdot p_{LH}) =$$
$$= 0.3\cdot\max(0.15\cdot 0.5, 0.1\cdot 0.4) = 0.3\cdot 0.075 = 0.0225$$

↖ obtained from p_H(G,1)

$$p_L(G,2) = e_L(G)\max(p_H(G,1)\cdot p_{HL}, p_L(G,1)\cdot p_{LL}) =$$
$$= 0.2\cdot\max(0.15\cdot 0.5, 0.1\cdot 0.6) = 0.2\cdot 0.075 = 0.015$$

↖ obtained from p_H(G,1)

The Viterbi Algorithm

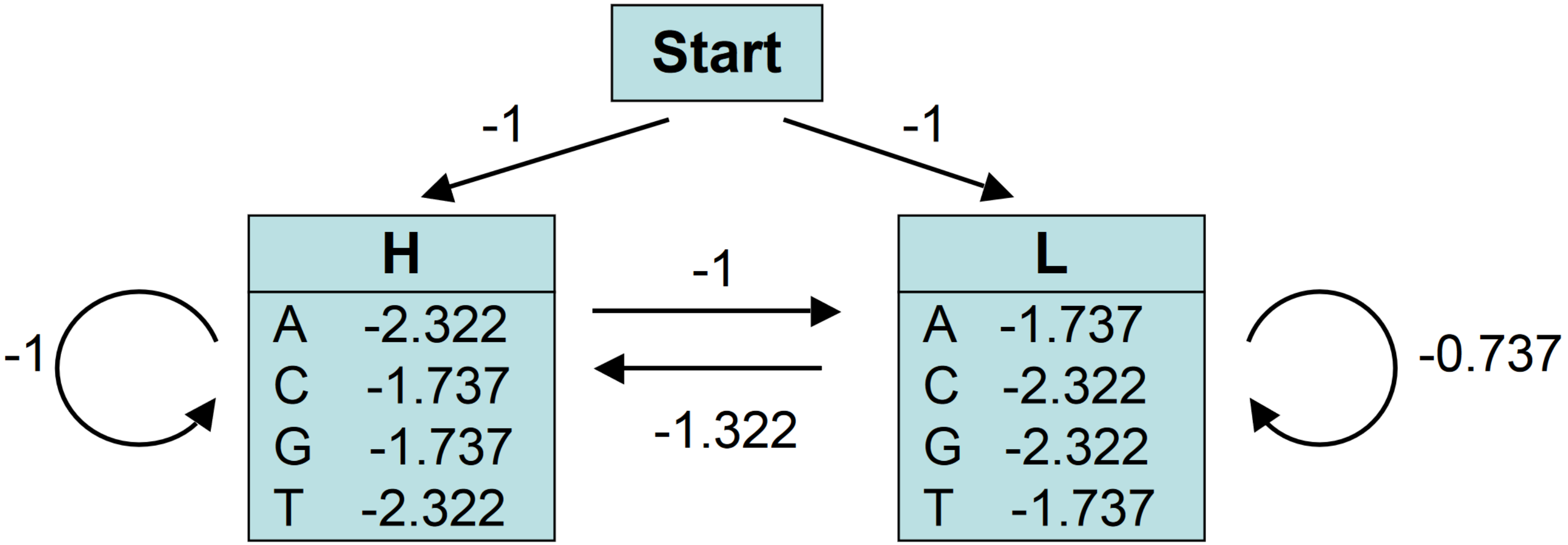


G G C A C T G A A

	G	G	C	A	C	T	G	A	A
H	0.15	0.0225	0.0034	0.0003	...				
L	0.10	0.015	0.0023	0.0005	...				

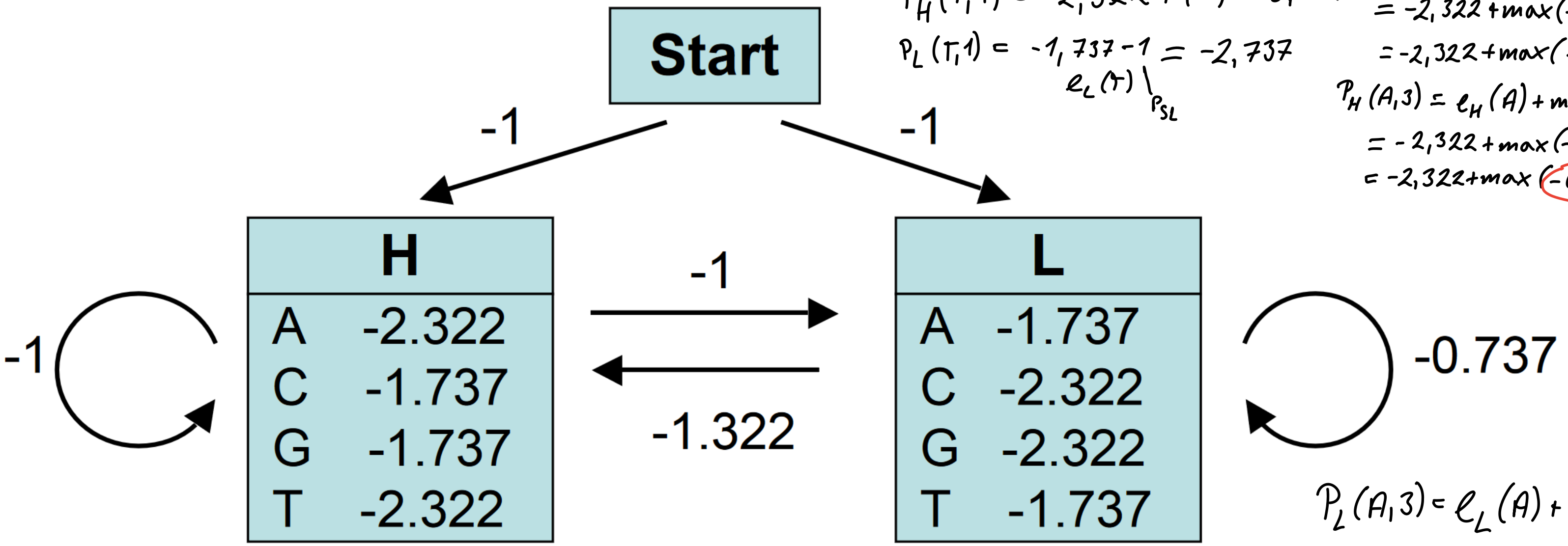
the most probable path is: HHHLLLLLL

The Viterbi Algorithm



- for the calculations, it is convenient to use the log of the probabilities (rather than the probabilities themselves) because it allows to compute sums instead of products, which is more efficient and accurate
- here, $\log_2(p)$ was used

Example 1



what hidden states generated sequence TGA?

	T	G	A
	L	L	L
H	-3,322	-5,796	-9,118
L	-2,737	-5,796	-8,270

$$P_H(T,1) = \overset{e_H(T)}{-2{,}322} + \overset{P_{SH}}{(-1)} = -3{,}322$$

$$P_L(T,1) = \underset{e_L(T)}{-1{,}737} \underset{P_{SL}}{-1} = -2{,}737$$

$$P_H(G,2) = e_H(G) + \max(P_H(T,1) + P_{HH}, P_L(T,1) + P_{LH}$$

$$= -1{,}737 + \max(-3{,}322 + (-1); -2{,}737 + (-1{,}322)) =$$

$$-1{,}737 + \max(-4{,}322; \boxed{-4{,}059}) = \underline{-5{,}796}$$

$$P_L(G,2) = e_L(G) + \max(P_H(T,1) + P_{HL}; P_L(T,1) + P_{LL})$$

$$= -2{,}322 + \max(-3{,}322 - 1; -2{,}737 - 0{,}737) =$$

$$= -2{,}322 + \max(-4{,}322; \boxed{-3{,}474}) = \underline{-5{,}796}$$

$$P_H(A,3) = e_H(A) + \max(P_H(G,2) + P_{HH}; P_L(G,2) + P_{LH})$$

$$= -2{,}322 + \max(-5{,}796 + (-1); -5{,}796 + (-1{,}322)) =$$

$$= -2{,}322 + \max(\boxed{-6{,}796}; -7{,}188) = \underline{-9{,}118}$$

$$P_L(A,3) = e_L(A) + \max(P_H(G,2) + P_{HL}; P_L(G,2) + P_{LL})$$

$$= -1{,}737 + \max(-5{,}796 + (-1); -5{,}796 + (-0{,}737)) =$$

$$= -1{,}737 + \max(-6{,}796; \boxed{-6{,}533}) = -8{,}270$$

Example 2

consider following HMM:

$$N = \{A, B, C\}$$ → sets of possible states

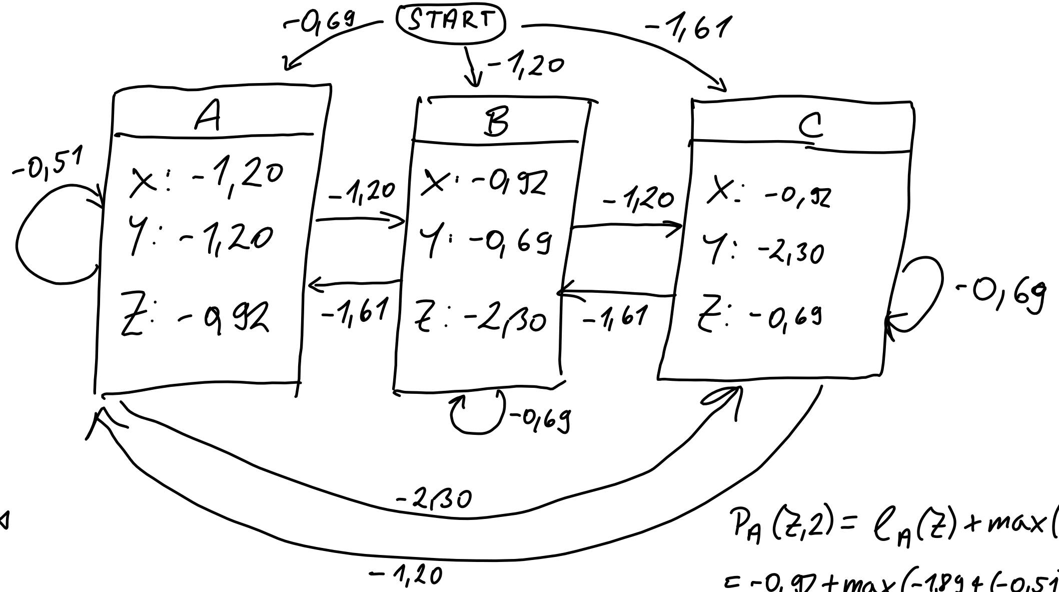
$$M = \{X, Y, Z\}$$ → sets of characters

$$A = \begin{bmatrix} -0.51 & -1.20 & -2.30 \\ -1.61 & -0.69 & -1.20 \\ -1.20 & -1.61 & -0.69 \end{bmatrix}$$

$$B = \begin{bmatrix} -1.20 & -1.20 & -0.92 \\ -0.92 & -0.69 & -2.30 \\ -0.92 & -2.30 & -0.69 \end{bmatrix}$$

$$\pi = \begin{bmatrix} -0.69 & -1.20 & -1.61 \end{bmatrix}$$

what hidden states generated sequence XZYY?



$$P_A(Z,2) = e_A(Z) + \max(P_A(X,1) + P_{AA}; P_B(X,1) + P_{BA}; P_C(X,1) + P_{CA})$$
$$= -0{,}92 + \max(-1{,}89 + (-0{,}51); -2{,}12 + (-1{,}61); -2{,}53 + (-1{,}20))$$
$$= -0{,}92 + \max(\boxed{-2{,}40}; -3{,}73; -3{,}73) = \underline{-3{,}32}$$

$$P_B(Z,2) = e_B(Z) + \max(P_A(X,1) + P_{AB}; P_B(X,1) + P_{BB}; P_C(X,1) + P_{CB}) =$$
$$= -2{,}30 + \max(-1{,}89 + (-1{,}20); -2{,}12 - 0{,}69; -2{,}53 - 1{,}61) =$$
$$= -2{,}30 + \max(-3{,}09; \boxed{-2{,}81}; -4{,}14) = \underline{-5{,}11}$$

$$P_C(Z,2) = e_C(Z) + \max(P_A(X,1) + P_{AC}; P_B(X,1) + P_{BC}; P_C(X,1) + P_{CC}) =$$
$$= -1{,}89 + (-$$

	A	A	B	B
	X	Z	Y	Y
A	-1,89 →	-3,32		
B	-2,12 →	-5,11		
C	-2,53 →	-3,91		

$$P_A(X,1) = e_A(X) + P_{SA} = -1{,}20 + (-0{,}69) = -1{,}89$$
$$P_B(X,1) = e_B(X) + P_{SB} = -0{,}92 + (-1{,}20) = -2{,}12$$
$$P_C(X,1) = e_C(X) + P_{SC} = -0{,}92 + (-1{,}61) = -2{,}53$$

Task

- in R, implement the Viterbi algorithm
- input:
  - a `AAString` of observations
  - Hidden Markov Model i.e. one of the prepared models `HMM1.Rdata`,`HMM2.Rdata` or `HMM3.Rdata`
- output:
  - matrix of calculated probabilities for all observations generated by all hidden states ($|N| \times$ number of observations)
  - a vector of hidden states, that generated the input observations

- $HMM = (N, M, A, B, \pi)$
  - N … set of hidden states
  - M … set of emitted characters
  - A … transition probability matrix
  - B … emission probability matrix
  - *pi* … initial probability distribution vector