INTRODUCTION TO BIOINFORMATICS

PRACTICE – 9th week

Hidden Markov Models

PROBABILITY THEORY

- CONDITIONAL PROBABILITY $P(A \mid B)$
- JOINT PROBABILITY $P(A, B)$

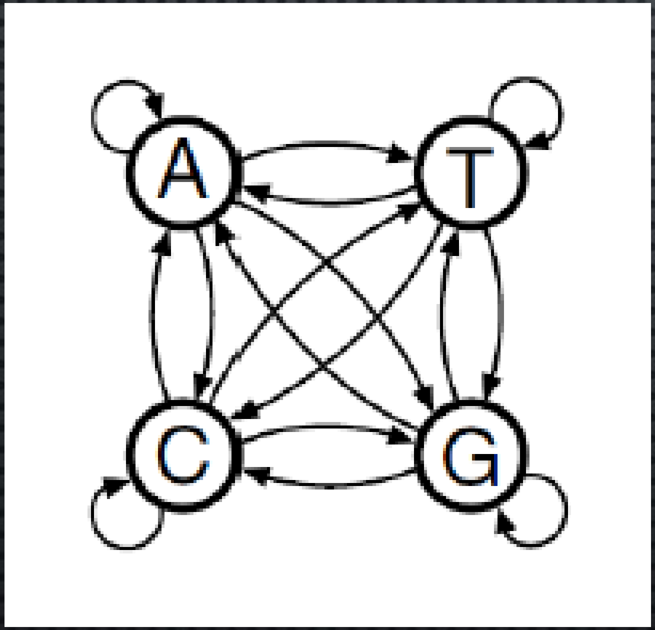
$$P(X) = \sum_{Y} P(X, Y) = \sum_{Y} P(X \mid Y) P(Y)$$

- BAYES THEOREM

$$P(X \mid Y) = \frac{P(Y \mid X) P(X)}{P(Y)}$$

MARKOV CHAINS

MARKOV CHAIN FOR GENERATING RANDOM NUCLEOTIDE SEQUENCES:



- Each state represents a symbol
- The sequence is the ('time') series of states

Markov Chains

- Markov chains can be described by state transitions
- Given a finite set of states: $x = \{x_1\ x_2\ x_3 \ldots x_L\}$
- At time point *t+1* the process stays in the same state as in time step *t* or moves to another state
- **Transition probability**: p_{ij} (the probability of moving from the *ith* state to the *jth*)

  → we can build a **state transition matrix** from them

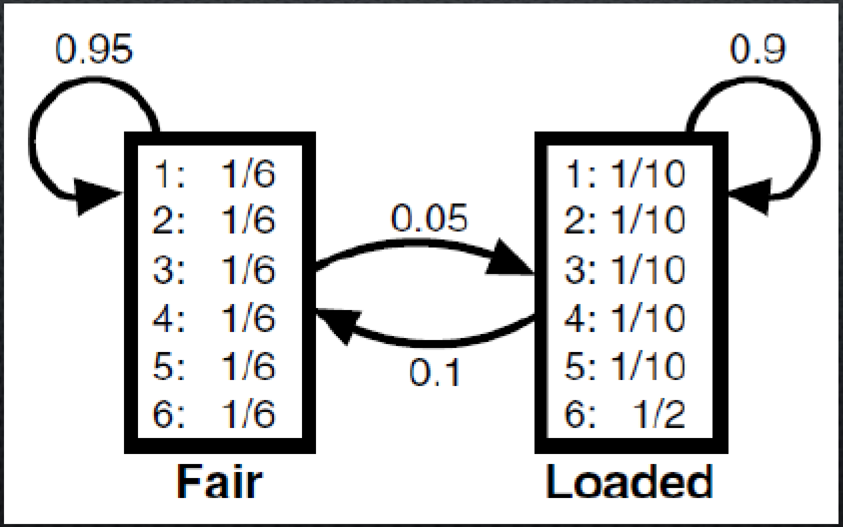
$$P(x) = P(x_L, x_{L-1}, \ldots, x_1) = P(x_L \mid x_{L-1}, \ldots, x_1) \cdot P(x_{L-1} \mid x_{L-2}, .., x_1) \cdot \ldots \cdot P(x_1)$$

- **Markov property**: next state depends only on the actual one:

$$P(x) = P(x_L \mid x_{L-1}) \cdot P(x_{L-1} \mid .x_{L-2}) \cdot \ldots \cdot P(x_2 \mid x_1) \cdot P(x_1)$$

HMM – Unfair Casino

- An unfair casino uses 2 kinds of dices: 99% of the dices is normal, 1% is loaded. Rolling 6 with the loaded dice has a probability of 0.5.
- HMM model of the unfair casino:



HMM - UNFAIR CASINO

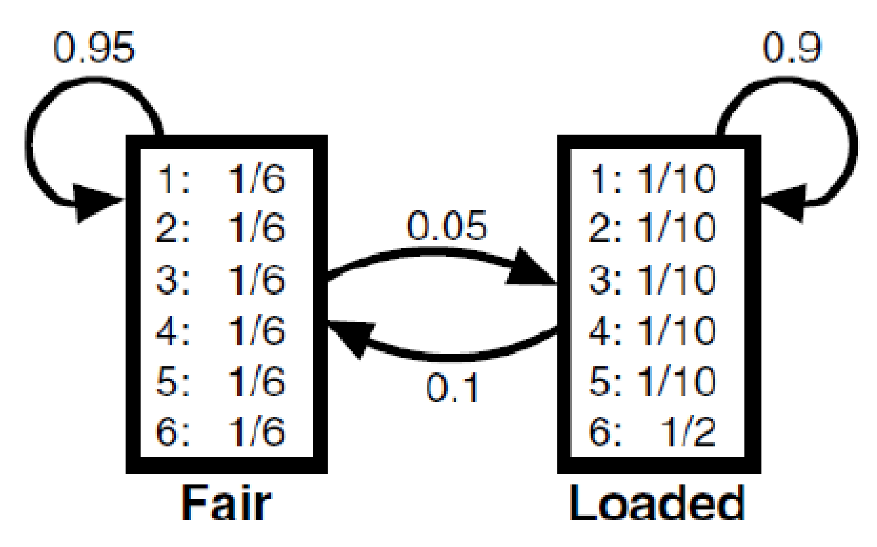
- Generating sequences of symbols (i.e. numbers)
  - The same sequence could be generated by either a loaded or normal dice
- One can obtain the sequence itself, but not the states!
- How do you figure out what were the states (dices), that are possibly generated the underlying sequences

Which one was more likely generated by a loaded dice?

SEQ1: 45**66**32**66**164**6666**411**6**5**66**5**6**312**6**33**66**233**66**3**6**21353

SEQ2: 251454242**6**251**6**152344 5**6**5554 3**6**4345325**6**34324**6**14

HMM - COMPONENTS



- **Hidden state:** (Loaded/Normal dice) and (observable) **emitted symbols** ({ACTG}/{1,2,3,4,5,6})
- **Emission matrix:** probability of observable symbols given that the hidden model is in a certain hidden state
- **Initial distribution:** the probability of the model being in a certain hidden state at time 0
- **State transition matrix:** probability of moving from one hidden state to another hidden state

Decoding - Viterbi Algorithm

Question: What is the sequence of the states that generated the following sequence?

- Figuring out that at each position the loaded or the fair dice had been used

45**66**32**66**164**6666**411**6**5**66**5**6**312**6**33**66**233**66**3**6**21353251454242**6**
251**6**1523445**6**55543**6**4345325**6**34324**6**14

Solution:
Assuming that we know that until the position 8 the loaded dice was used, how would you decide which one is the next state? Suggestions?

45**66**32**66**1

LLLLLLLL?

Decoding - Viterbi Algorithm

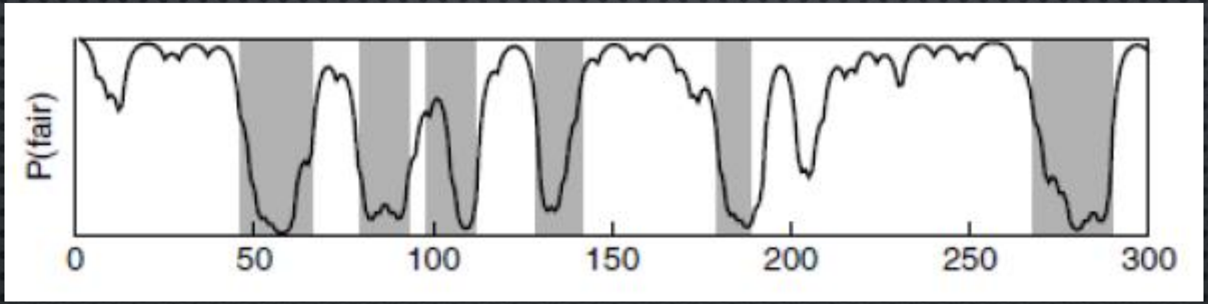
- Viterbi: finding the most probable path (path is denoted as **π**)
- Most probable path: $\pi^* = \arg\max_{\pi} P(x, \pi)$
- Let $\boldsymbol{v_k(i)}$ be the probability that the most probable path ends at ***k***. state where it emits symbol $\boldsymbol{x_i}$

$$v_l(i+1) = e_l(x_{i+1}) \max_k (v_k(i) a_{kl})$$

where $\mathbf{e_l(x_{i+1})}$ is the probability of emitting $\mathbf{x_{i+1}}$at the ***l*** state; $\mathbf{a_{kl}}$ is the transition probability from ***k*** to ***l***

Viterbi - Unfair casino

- With dynamic programming it is possible to find the most probable path



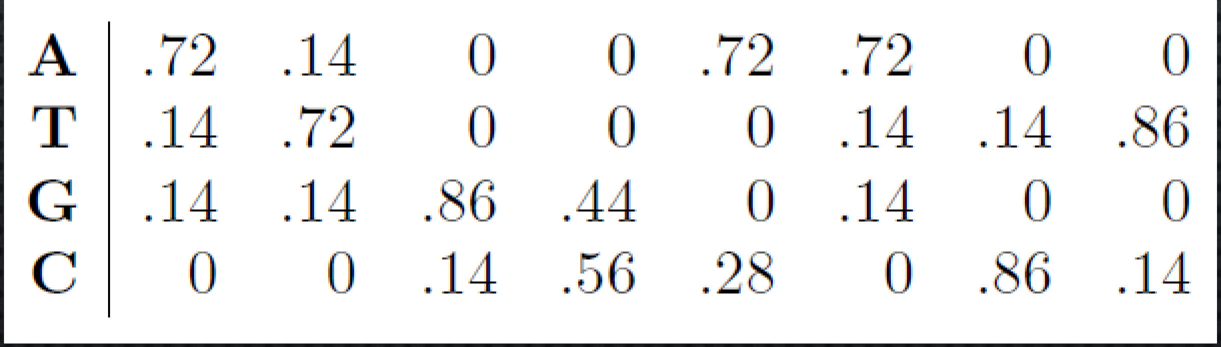
- Finding genes in a nucleotide sequence

HMM - TRAINING

- Until now we assumed that we know the model exactly. (model: transition and emission probabilities)
- In real life it is not the case
- The model parameters (emission and transition probabilities) must be estimated somehow (~training the model)
- Must have a training set, i.e. annotated genomes, annotated sequences
- Estimation:
    - Expectation-maximization algorithm
    - Baum Welch training

Profile HMM

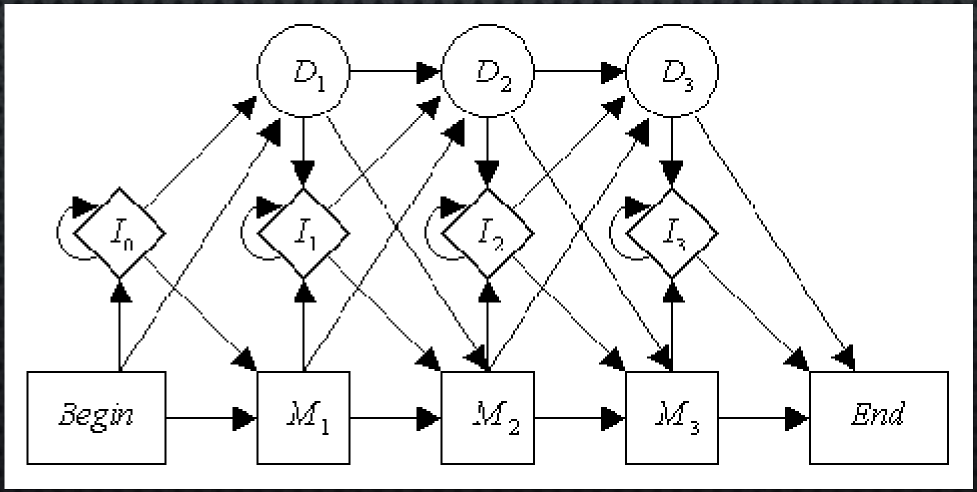
- Given a protein family we want to find other members of this family in a given database
- We can make a profile that represents the protein family and we compare the database elements to this profile
- An HMM can describe such a profile



A	.72	.14	0	0	.72	.72	0	0
T	.14	.72	0	0	0	.14	.14	.86
G	.14	.14	.86	.44	0	.14	0	0
C	0	0	.14	.56	.28	0	.86	.14

PROFILE HMM

- STATES:
  - M: MATCH STATE
  - I: INSERTION STATE
  - D: DELETION STATE (SILENT STATE – NO EMISSION)
- N: LENGTH OF SEQUENCE (NUMBER OF MATCHES)


D_1
D_2
D_3
I_0
I_1
I_2
I_3
Begin
M_1
M_2
M_3
End