

## Supplementary File

### HMMTeacher 1.0 - User Manual

February 2020

The purpose of this manual is to guide the user in the execution of HMMTeacher, through practical exercises in the learning of hidden Markov Models.

#### Frequently Asked Questions - FAQ

##### 1. What is a Hidden Markov Model or HMM?

An HMM is a modeling technique that is useful to represent situations in which a sequence of observed states, in time or space, is emitted by a sequence of hidden states or properties. Some examples are Prediction of (hidden) weather from (observed) atmospheric pressure; Prediction of (hidden) protein properties, like structure and function, from the (observed) protein sequence; Prediction of the (hidden) music composer from the (observed) melody of a song. As seen in the examples, many times, what we want to uncover is the most probable sequence of hidden states.

##### 2. What general questions can an HMM answer?

There are four common questions that an HMM is capable of answering with four algorithms:

- i. Probability of the sequence of observed states given the Model (Forward Algorithm).
- ii. Probability of a particular state being emitted by a particular hidden state (Backward Algorithm).
- iii. The most probable sequence of hidden states that emits the sequence of observed states given the Model (Viterbi Algorithm).
- iv. The optimal parameters of the HMM that produce a set of observations (Expectation-Maximization, EM Algorithm).

This manual is focused on developing simple examples that will help the user to use the tool and learn how to model problems with an HMM, answering the first three questions and interpreting the results.

##### 3. What are the input parameters or elements of an HMM?

An HMM consists of:

- i. An alphabet of  $m$  observed states<sup>1</sup>.
- ii. An alphabet of  $n$  hidden states.
- iii. A sequence of observed states, of length  $L$ .
- iv. A vector  $\vec{v}$  of  $n \times 1$ , of initial probabilities of the hidden states.
- v. A matrix,  $P$ , of  $n \times n$ , of transition probabilities between the hidden states.
- vi. An emission matrix  $E$ , of  $n \times m$ , with the probability of emitting an observed state by a hidden state.

---

<sup>1</sup> We use a letter of an alphabet to represent the states for simplicity. It could be a set of expressions too.

In order to solve an HMM, a number of parameters have to be estimated. HMM parameter data processing and estimation constitute a problem on its own. Here, for simplicity and focus on the HMMs, we assume that all the input needed, the parameters of the matrices, are known and available.

**4. Could you give a concrete example of a problem modeled with an HMM?**

To develop the concepts, we will start with an example known as the “occasionally dishonest casino” problem. This problem was taken from the book *“Biological Sequence Analysis”* by Kroug et al., 1998.

A croupier rolls the dice in sequence. In the meantime, you notice that before rolling the dice, he discreetly chooses between two dice. You assume that one of the dices is loaded (the other is a fair dice) and you want to discover in a sequence of (observed) rolls, which is the (hidden) sequence of the loaded or the fair dice.

**5. What are the observed states?**

The observed states correspond to a set of features that we can be measured or observed. In an HMM problem, the set of observed states is given. Examples: A peptide; a nucleotide; a sequence of dice rolls; the phenotype of an organism; a song.

**6. What are the hidden states?**

The normally unknown state conditions to which the observed states are subject. These conditions affect the probability of the occurrence of observed states. For example: a protein structure might be represented by a sequence of hidden amino acid properties that determine an observed sequence of amino acids; in our example problem  $F$  (for Fair),  $L$  (for Loaded), that indicate whether the dice chosen for the roll is Loaded or Fair; the set of music composers.

**7. What is the transition matrix of an HMM?**

It is a  $n \times n$  matrix of transition probabilities between two hidden states, in two consecutive moments or positions in the sequence. A transition probability is the probability of a hidden state, given another hidden state in the previous moment or position.

In our example, each cell in the matrix  $P$  is represented as a conditional probability,

$$P = [P(F_t | F_{t-1}) \ P(L_t | F_{t-1}) \ P(F_t | L_{t-1}) \ P(L_t | L_{t-1}) ]$$

Where, for instance,  $P(F_t | L_{t-1})$  is read probability of the Fair ( $F$ ) dice being chosen at roll  $t$ , given that in the previous roll,  $t - 1$ , the chosen dice is known to be the Loaded ( $L$ ) one. It can be written just  $P(F | L)$ , as this value is the same for all  $t$ 's.

The assumed dependence between two consecutive hidden states is called Markovian property, and that is what makes the sequence to have a particular order. The term “Markov chain” is an analogy which refers to this relationship between consecutive hidden states. One important detail of matrix  $P$  is that the sum of probabilities in a row equals one.

**8. What is the initial probabilities vector?**

The initial probabilities vector is a  $n \times 1$  matrix of the probabilities for each hidden state of the alphabet at a time or position  $t = 1$ . In our example, at roll 1,

$$\vec{v} = [P(F_{t=1}) \ P(L_{t=1}) ]$$

Notice that the probabilities in  $\vec{v}$  are not conditional. That is because there is no time or position before the first, i.e.,  $t = 1$ . Similar to  $P$ , the probabilities in  $\vec{v}$  sum to one.

**9. What is the emission probability matrix of an observation given a hidden state?**

It's a  $n \times m$  matrix,  $E$ , where each cell has the probability that a hidden state emits an observed state. In our example, the possible observations are from 1 to 6, and the hidden states are  $F$  (Fair) and  $L$  (Loaded).

$$E = [P(1|F) P(2|F) P(3|F) P(4|F) P(5|F) P(6|F) \quad P(1|L) P(2|L) P(3|L) P(4|L) P(5|L) P(6|L)]$$

Where, for instance,  $P(1|F)$  reads, probability of the dice to roll a 1, given (represented by “|”) that the dice is Fair. In our example, it is expected that the Fair dice would render each of the possible 6 outcomes the same proportion of times. Thus,  $P(1|F) = P(2|F) = P(3|F) = P(4|F) = P(5|F) = P(6|F) = 1/6$ .

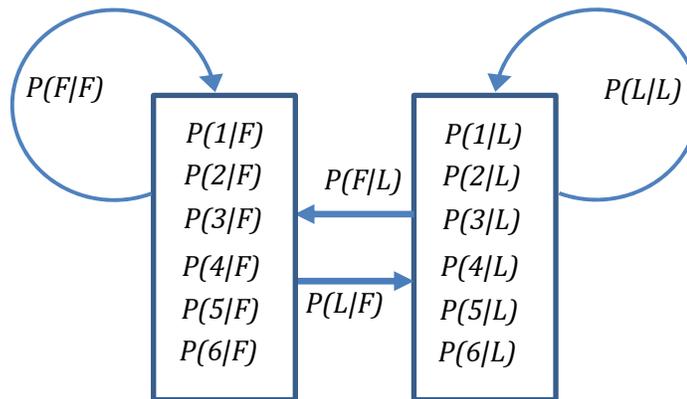
Similar to the initial vector  $\vec{v}$  and the transition matrix  $P$ , the probabilities in a row sum to one.

**10. What does it mean the expression “training of an HMM”?**

The parameters of an HMM consist in the initial vector  $\vec{v}$  of probabilities, the transition matrix  $P$  and emission matrix  $E$ . The training of an HMM, or of any model, is to estimate the probabilities of the HMM parameters from a training set. A training set could be a database of records of known cases.

**11. Is there a graphical representation of an HMM?**

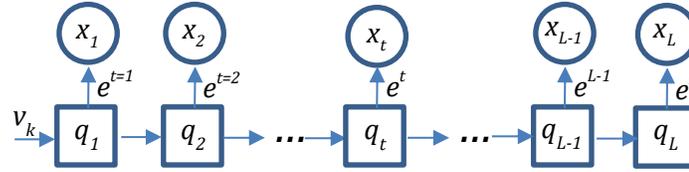
The classic representation of a Markov chain is a graph. A graph consists in a set of arrows connecting a set of nodes. Here, the nodes represent the hidden states. Each arrow represents a transition probability between two hidden states. In our example we have only two hidden states,  $F$  and  $L$ . Within each hidden state node, in the figure, the emission probabilities are found. In our example, the graph of the corresponding HMM would be



The vector of initial probabilities is not represented in the graph.

**12. How is it calculated a sequence of observed states,  $x_1x_2 \dots x_{n-1}x_L$ , from a known sequence of hidden states,  $q_1q_2 \dots q_{n-1}q_L$ ?**

If we have the initial parameters of the model  $M$ , which are  $\vec{v}$ ,  $P$ ,  $E$ , the sequence of observed states  $x_1x_2 \dots x_{n-1}x_L$ , and the sequence of  $L$  hidden states  $q_1q_2 \dots q_t \dots q_{L-1}q_L$ , then the process can be represented like this,



Where,  $x_t$  is an observation. The value of  $x_t$  is one of the symbols of the observed state alphabet. In our example,  $x_t = \{1,2,3,4,5,6\}$ .  $t$  is the index of the moment/position of the sequence of observed states,  $1 \leq t \leq L$ .  $q_t$ , a hidden state. The value of  $q_t$  is one of the symbols of the hidden state alphabet. In our example,  $q_t = \{F, L\}$ .

The formula for the probability of a sequence of observed states, emitted by a known sequence of hidden states is,

$$P(x_1x_2 \dots x_t \dots x_{n-1}x_L | q_1q_2 \dots q_t \dots q_{L-1}q_L M) = \vec{v}_k \cdot e_{ks}^{t=1} \cdot \prod_{t=2}^{t=L} p_{ij}^t \cdot e_{jh}^t$$

Where,

- $M$  is the Model which includes the vector of initial probabilities, the transition matrix  $P$  and the emission matrix  $E$ .
- $k$ , the index of the first hidden state.  $1 \leq k \leq n$ .
- $s$ , the index of the first observed state.  $1 \leq s \leq m$ .
- $i$ , the index of the hidden state at the moment/position  $t-1$ ,  $1 \leq i \leq n$ .
- $j$ , the index of the hidden state at the moment/position  $t$ ,  $1 \leq j \leq n$ .
- $h$ , the index of the observed state emitted at the moment/position  $t$ ,  $1 \leq h \leq m$ .

**13. Does this last formula correspond to the Forward algorithm of question P2?**

No. In the previous formula, the sequence of hidden states,  $q_1q_2 \dots q_t \dots q_{L-1}q_L$ , is known. The forward algorithm answers the question of what the probability of a known sequence of observed states is, emitted by an unknown sequence of hidden states. Therefore, in the Forward case, it is necessary to add up on all the possible combinations and hidden states probabilities that could emit the same sequence of observed states.

#### 14. Where can I read more about HMMs?

Instead of providing an exhaustive bibliographic research, we preferred to present a few selected sources to the ones who are interested in understanding the theory behind HMMs. In 1989, Lawrence R. Rabiner wrote "A tutorial on Hidden Markov Models and Selected Applications in Speech Recognition". It is fairly very well explained and it is available [online](#). A small but very good chapter in the book "[Statistical Methods in Bioinformatics: An introduction](#)", by Ewens and Grant, Springer editorial, 2002. It explains an HMM and its main three algorithms in a couple of pages. Impressive!

And "[Biological Sequence Analysis](#)", by Krough et al. is a classic 1998 book written by a few of the first researchers that applied this modeling technique to biological sequences, in order to improve the annotation of genes and proteins among other predictions like evolutionary trees or three dimensional structure of proteins. Other sources, for the ones who like computers, One chapter from the book "[Pattern Recognition and Machine Learning](#)", by Christopher Bishop, Springer 2006.

There is, literally, over a thousand papers in the last thirty years on different specific applications, mainly in molecular biology, genetics of populations and molecular evolution, and improvements in the basic implemented algorithms presented in this website. Many of the abstracts of these papers can be found in NCBI's Pubmed, [here](#). Finally, there are many tutorials and videos [online](#). If you know better references, we would like to include them in this manual. Please, write us an [e-mail](#)!

### 15. What HMM software exists?

There are many [implementations](#) of the algorithms for different applications.

[HMMer](#) is, maybe the best known general purpose application of HMMs to biological sequence analysis. It allows you to use a multiple sequence alignment as input to create an HMM of the patterns of the gene or protein sequences given. You can further search genomes and other sequence and HMM databases with your own HMM. It was created by Sean Eddy, which is one of the authors of the book Biological Sequence Analysis presented above. [InterproScan](#) and [PFam](#), and [RFam](#) are other applications and [databases](#) of HMMs to Biological Sequence Analysis

## Problems

- Coins Loaded or Fair.
- Dice Loaded or Fair.
- Protein 2<sup>nd</sup> structure prediction, through protein sequence?
- A functional motif is hidden in a DNA sequence.
- The Backward algorithm needs the parameters of the Observed symbol at a particular time/position and a hypothesis of which hidden state might be emitting the Observed state. Does the solution at time/position  $t$  depend on the Observed or Hidden states in the future? i.e. in a  $T > t$ ? How we can find out?

## Solved problems of modeling, resolutions, and interpretations of HMMs results.

**Problem 1:** The actual DNA sequences are not homogeneous and can be described by Hidden Markov Models with hidden states representing different types of nucleotides composition. Considering an HMM that contains two hidden states  $H$  and  $L$  for high or low  $C + G$  content, respectively. The initial probabilities for both  $H$  and  $L$  are 0.5, while the transition probabilities are the following:  $a_{HH} = 0.5$ ;  $a_{HL} = 0.5$ ;  $a_{LL} = 0.6$ ;  $a_{LH} = 0.4$ . The nucleotides T, C, A, G are emitted by the states  $H$  and  $L$  with the probabilities 0.2, 0.3, 0.2, 0.3 and 0.3, 0.2, 0.3, 0.2, respectively. Using the Viterbi algorithm, define the sequence of hidden states most likely for the “toy” sequence  $x = \text{GGCACTGAA}$ .

According to problem 1:

### Identify the hidden states.

For this case, the hidden states are explicitly described and correspond to  $H$  and  $L$ .

- **H:** High content of  $G+C$
- **L:** Low content of  $G+C$

### Identify the observed states.

In this problem, nucleotides are explicitly mentioned, so our observations will be a nucleotide sequence. This is why the observed states will be A, T, C, and G.

- Nucleotidos observados: A - T - C - G.

### Identify the chain of observed states.

At the end of the problem, we find the sequence of observed nucleotides that is to be analyzed using the described probabilities of the problem.

- Observed sequence: GGCACTGAA

### Identify the emission probability matrix.

The emission probability matrix corresponds to the probability that the model shows a certain observed state (A, T, C or G) given a certain hidden state (H or L). This data is obtained from the problem description and are used at it is. The emission probability matrix is represented below:

	A	T	C	G
H	0.3	0.3	0.2	0.2
L	0.2	0.2	0.3	0.3

**Identify the transition probability matrix.**

The transition probabilities matrix corresponds to the probability of changing from one hidden state to another. In this particular case, it refers to change from H (High GC content) to L (Low GC content) or vice versa. This information is given in the problem and it's represented in the following way.

	H	L
H	0.5	0.5
L	0.4	0.6

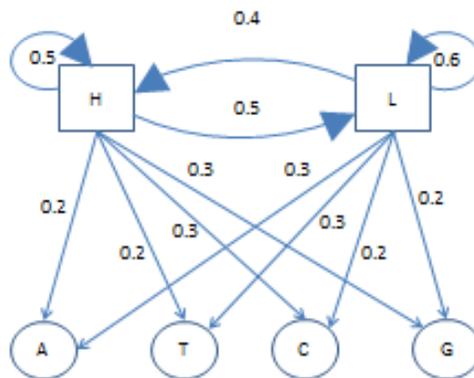
**Identify the initial probabilities.**

The initial probabilities correspond to the probability that a particular hidden state has emitted the first observed state of the sequence of observed states. For this particular problem, this information is given, but in the case that these values are not found, the probability is divided equally among all the hidden states. These probabilities are represented in the following matrix.

H	L
0.5	0.5

**Draw the graph representation of the HMM.**

The graph representation of a hidden Markov chain aims to show the interactions that occur between each observed state with the hidden states, in addition to indicating the probability in which that interaction occurs. According to the transition and emission matrix, the following graph model of the problem is:



## Resolve the question of problem 1: Execute the Viterbi algorithm.

Once all the data have been identified and having established all the transitions that can be made, we proceed to enter this information in the software.

The main page of the tool shows the following options.

### HMMTeacher

HMMTeacher guides you in the learning of the mechanics of solving the main three HMM algorithms. From the input of the data of a pre-stated HMM, passing by choosing the questions asked and the algorithms to apply, to the final report, containing the step-by-step algorithms solution.

The screenshot shows the 'Step 1 - States' configuration screen. It has a progress bar at the top with four steps: Step 1 - States (Hidden States), Step 2 - Matrix (Fill the matrix), Step 3 - Algorithm (Select an algorithm), and Step 4 - Results (Final results). Below the progress bar are 'Previous', 'Next', and 'Reset' buttons. The main content area is titled 'Select the alphabet of the observed sequence of states to use'. There are three panels: 'DNA', 'Protein', and 'Custom'. The 'DNA' panel is active and shows 'Alphabet of observed states: DNA Nucleotides (ACTG)'. A text input field contains 'GGCACTGAA'. Below it, 'Hidden states:' are listed as 'H' and 'L', each with a 'Remove' button. There is an 'Add' button at the bottom. The 'Protein' panel shows 'Alphabet of observed states: One letter code for Proteins (ARNDBCEQZGHILKMFSTWVY)'. It has an 'Example: MTLDA' and an 'Observed Protein sequence' field. 'Hidden states:' are 'H0' and 'H1', each with a 'Remove' button and an 'Add' button. The 'Custom' panel shows 'Custom alphabet' with an 'Example: HHTHTHT' and an 'Observed Custom sequence' field. 'Hidden states:' are 'H0' and 'H1', each with a 'Remove' button and an 'Add' button.

Image 1: Main window.

In this step, you can enter the observed sequences and its hidden states. For this problem, we must select the *DNA* option and enter the sequence of observed states, in this case, GGCACTGAA, the input sequence is limited to 20 characters in this tool. Next, you must enter the hidden states, in this case, H and L. You can enter up to 6 hidden states at maximum. Once ready, you can press proceed to the next step. The tool will perform validations if any of the required data is missing.

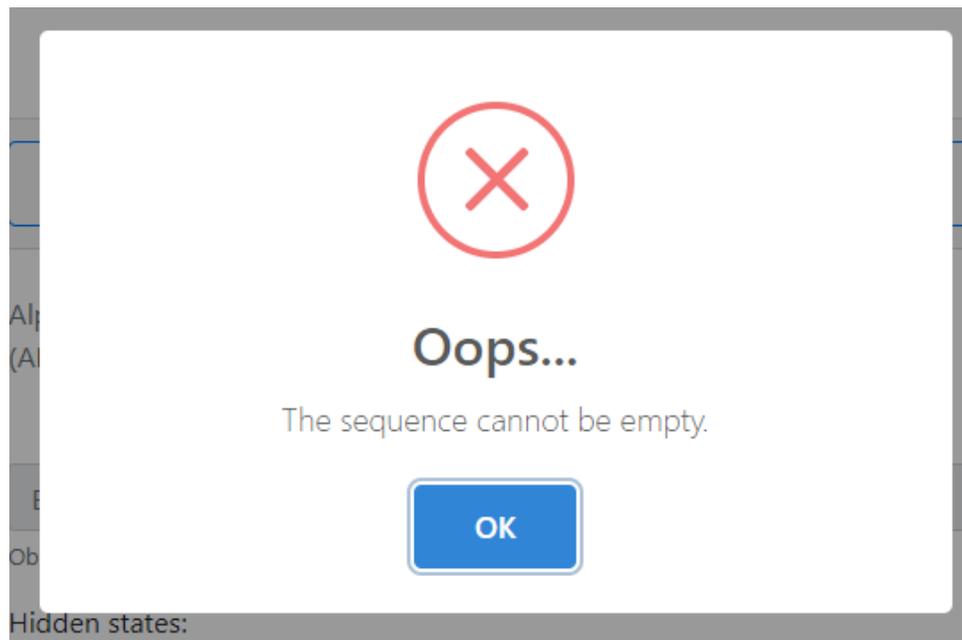


Image 2: Error message, associated with the validation of the data.

In the next step, it's possible to enter the probabilities of our model. First the initial probabilities, then the transition matrix and finally the emission matrix.

Input the data ?

**Priors Vector**

H	0.5
L	0.5

Random initial matrix

Random Clear

**Transition matrix**

	H	L
H	0.5	0.5
L	0.4	0.6

Random transition matrix

Random Clear

**Emission matrix**

	A	C	T	G
H	0.3	0.3	0.2	0.2
L	0.2	0.2	0.3	0.3

Random emission matrix

Random Clear

**Image 3: Probabilities associated with our model.**

It's possible to insert any number, although the sum of these numbers can't be greater than 1. The tool allows the insertion of random values, with the button "Random", this in case that the probabilities for the problem aren't given. After inserting all the values it's possible to move to the next step, the tool will perform validations when moving to the next step.

Select an algorithm ?

HMM algorithms

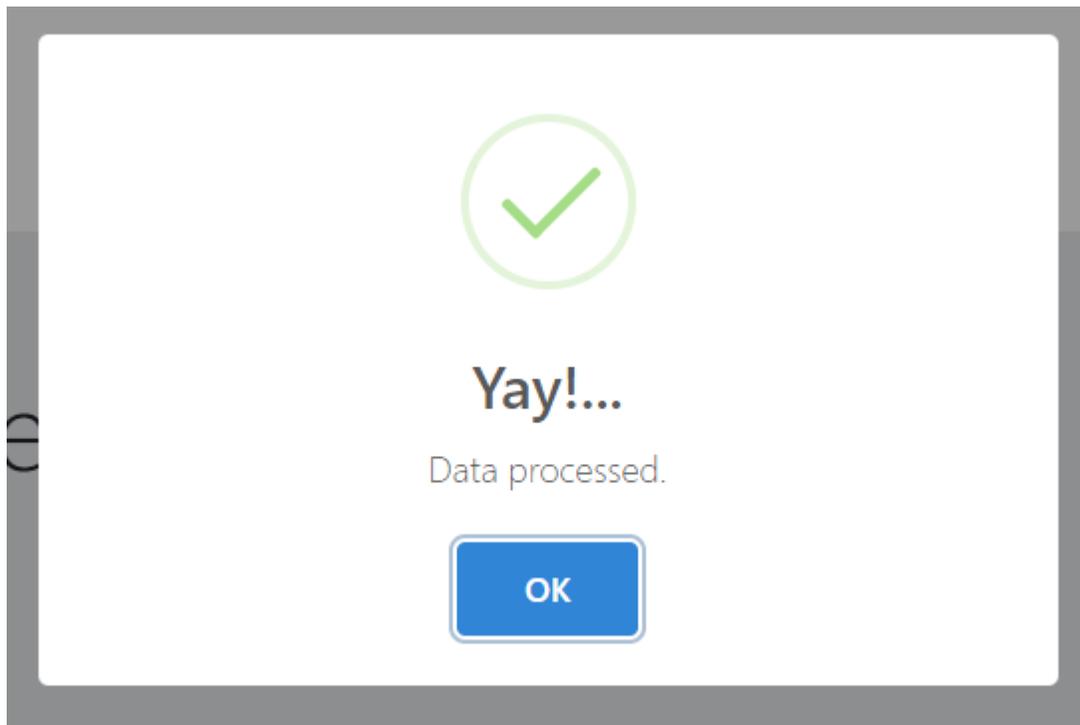
Forward

Viterbi

Backward

**Image 4: Selection of the mathematical algorithm to use.**

In this step, it's possible to choose which mathematical algorithm to use to resolve the problem. Forward, Backward and Viterbi algorithm. In this case, we must use the Viterbi algorithm to find the most probable sequence of hidden states after the observed sequence. It's important to mention that at least one algorithm needs to be chosen, and more than one can be chosen.



**Image 5: Success on analyzing the results.**

This confirmation is shown when the data is correctly processed.

[Download PDF](#)

## HMM Teacher results

2020-02-18 13:58:57.776396

<p style="font-size: x-small; margin: 0;">Observed sequence</p> <p style="margin: 5px 0;">GGCACTGAA</p>	<p style="font-size: x-small; margin: 0;">Alphabet</p> <ul style="list-style-type: none"> <li>• ACTG</li> </ul>	<p style="font-size: x-small; margin: 0;">Hidden States</p> <ul style="list-style-type: none"> <li>• H</li> <li>• L</li> </ul>	<p style="font-size: x-small; margin: 0;">Selected algorithms</p> <p style="margin: 5px 0;">1. Viterbi</p>
---	---	--	--

**Image 6: Showing the results.**

In the final step, the data is analyzed and processed. If the data was processed successfully the results will be displayed with detail. It's possible to also download a PDF file with the results.

## Backtracking

1. The matrix shows all the calculated probabilities.
2. In red is shown the calculated backtracking.
3. This, starting from the last column of the matrix (A).

	G <sub>1</sub>	G <sub>2</sub>	C <sub>3</sub>	A <sub>4</sub>	C <sub>5</sub>	T <sub>6</sub>	G <sub>7</sub>	A <sub>8</sub>	A <sub>9</sub>
H	0.1	0.012	0.00324	0.000486	7.29e-05	7.29e-06	8.748e-07	2.36196e-07	3.54294e-08
L	0.15	0.027	0.00324	0.0003888	4.86e-05	1.0935e-05	1.9683e-06	2.36196e-07	2.83435e-08

## Termination

$$P(O \cdot Q) = \max_i [\delta_L(S_i) \cdot a_{i0}]$$

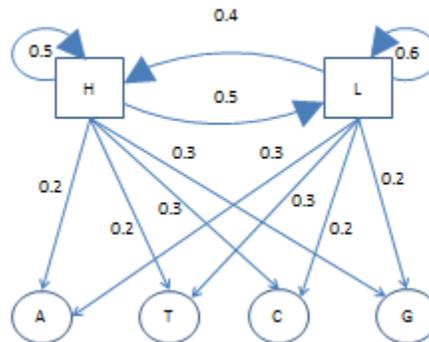
$$q_L^* = \arg \max_i [\delta_L(S_i) \cdot a_{i0}]$$

The steps of states are L L H H H L L H H with highest probability of 3.5429399999999994e-08

### Image 6: Showing the results of the selected algorithm.

In the results, it's possible to review the input data, the algorithms and the resolution for each one of them (if more than one was chosen). Lastly, in the final section of the algorithm, we can see the result of the most likely sequence that we wanted to discover.

**Problem 2:** From the hidden Markov model used in the previous problem, and the DNA sequence fragment X: GGCA, find the probability that this will occur using the forward algorithm. Recalling from the previous exercise, the HMM generated was:



We are asked to calculate the probability that the GGCA sequence will occur, using the generated model in the previous exercise.

To accomplish this with the tool, it's necessary to start from step 1 and enter the sequence of observed and hidden states. The observed sequence change to GGCA.

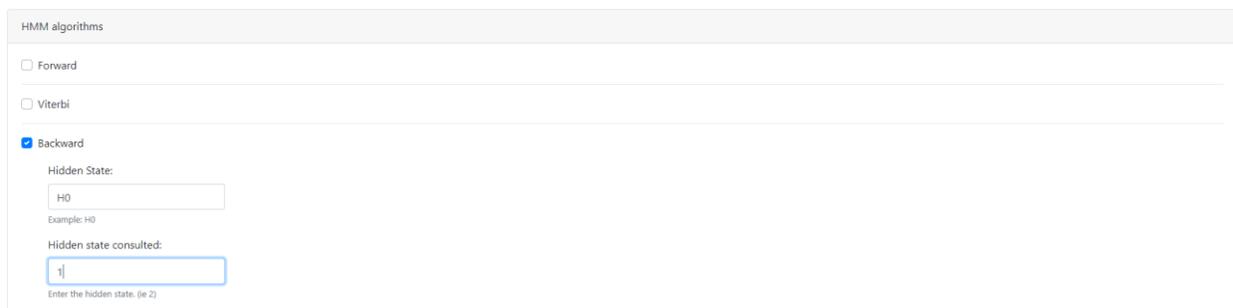
In step 2, the initial probabilities remain the same as does the transition matrix.

In the, we must choose the forward algorithm, which will tell us the probability that a certain sequence will occur according to a certain model.

As in problem 1, a window with the results is displayed. In the algorithm resolution part, the result we are looking is displayed. The probability that the GGCA sequence occur according to the model of problem 1 is 0.004.

**Problem 3:** Find a *posteriori* probabilities of the hidden states H and L at position 4 of the DNA sequence x: GGCA. Consider the HMM generated in problem 1.

The Backward algorithm must be used to calculate a posteriori probability of both hidden states in a specific position of the sequence GGCA. The input data is the same as the previous exercise, where the only difference with the previous 2 exercises is the selection of the algorithm to use, in this case, the Backward algorithm. By doing this, two additional boxes will be enabled. In these boxes, it's possible to enter the hidden state to search and the position of the sequence to verify. The Backward algorithm reverse the position of the observed sequence, so the last position of the sequence in the algorithm makes references to the first position of the original sequence. For example, in our problem sequence GGCA, the first observed state is G, while the last one is A. If a 1 is entered, the position to be evaluated is the last position or the letter A. If a 4 is entered, the position to be evaluated is the first one or G.



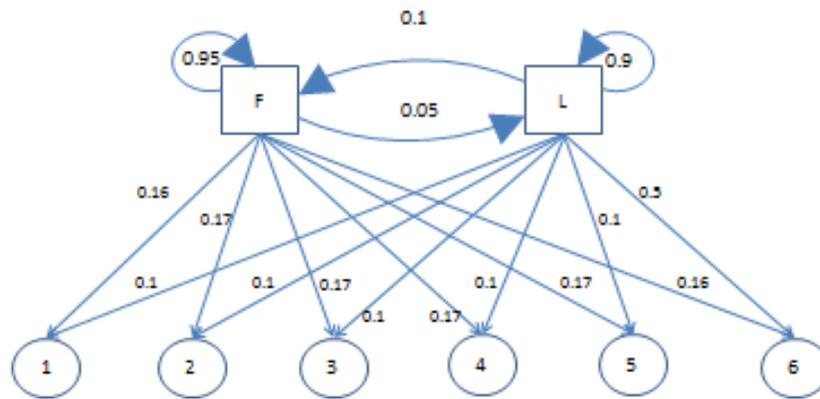
The screenshot shows a web interface titled "HMM algorithms". It has three radio buttons: "Forward", "Viterbi", and "Backward". The "Backward" option is selected. Below the "Backward" option, there are two input fields. The first is labeled "Hidden State:" and contains the text "H0". Below it, there is a small text "Example: H0". The second input field is labeled "Hidden state consulted:" and contains the number "1". Below this field, there is a small text "Enter the hidden state. (ie 2)".

**Image 7: Options in backward algorithm**  
How to enter options for the backward algorithm.

Finally, in the last step, it's possible to check the results. In this case, the value of the first observation having been emitted by the hidden state L is 50%, which in conclusion dictates that the probability that the first observed state was emitted by H is also 50% since it's known that both values must add up to 100%.

**Problem 4:** On a particular day, a casino uses a fair dice all the time (F) or uses a fair dice most of the time, but occasionally changes to a loaded dice (L). The probability of changing from a fair to a loaded dice after a throw is 0.05 and the probability of changing again is 0.1, and the loaded dice has a probability of 0.5 of showing a 6 and 0.1 for all the other numbers. Evaluate the following sequence of throws  $x$ : 6,4,6,2,1,1,6,4,3,3,5,5,6 and show the sequence of dice used for all the throws.

To solve this problem is necessary to create an HMM, establishing the interactions and the probabilities between them. The resulting model is shown below:



We must insert the given data in the tool. The observed states sequence, corresponding to the rolls of the dice ( $x$ ), and our hidden states are if the dice is loaded or fair (L, F).

Custom

Any custom alphabet, one character per symbol, without separators. Example of alphabet of observed states for a coin, H (Heads) T (Tail): HT

Observed Custom sequence (One character per observed symbol. Max length: 20 characters.)

Hidden states:

F

Remove

L

Remove

Add

**Image 8: Custom sequence entry data**  
How to enter custom data in the platform.

Next, the filling of the matrices. For this problem, the initial probabilities are not given, so we assume an equal probability for both cases, that is a 0.5 probability that the initial dice is fair (F) and a 0.5 probability that is loaded (L). The transition and emission matrices are filled with the data given in the HMM previously generated.

Input the data ?

**Priors Vector**

F 0.5

L 0.5

Random initial matrix

Random Clear

**Transition matrix**

	F	L
F	0.95	0.05
L	0.9	0.1

Random transition matrix

Random Clear

**Emission matrix**

	1	2	3	4	5	6
F	0.16	0.17	0.17	0.17	0.17	0.16
L	0.1	0.1	0.1	0.1	0.1	0.5

Random emission matrix

Random Clear

**Image 9: Filling the data of the custom entry**

The data given is used to fill the prior vector, transition matrix, emission matrix.

Next, we select the best algorithm for the particular question we want to answer. In this case, the Viterbi algorithm, since we want to know the sequence of hidden states that are emitted by the observed states.

Once the calculations are completed in the final step, it's possible to see the answer to our question:

### Backtracking

1. The matrix shows all the calculated probabilities.
2. In red is shown the calculated backtracking.
3. This, starting from the last column of the matrix (3).

	6 <sub>1</sub>	4 <sub>2</sub>	6 <sub>3</sub>	2 <sub>4</sub>	1 <sub>5</sub>	1 <sub>6</sub>	6 <sub>7</sub>	4 <sub>8</sub>	3 <sub>9</sub>	3 <sub>10</sub>
F	0.25	0.0025	0.00095625	2.907e-05	4.69481e-06	7.1361e-07	5.42344e-07	1.64873e-08	2.66269e-09	4.30025e-10
L	0.08	0.03825	0.005814	0.000938961	0.000142722	2.16938e-05	3.29745e-06	5.32538e-07	8.60049e-08	1.38898e-08

### Termination

$$P(O \cdot Q) = \max_i [\delta_L(S_i) \cdot a_{i0}]$$

$$q_L^* = \arg \max_i [\delta_L(S_i) \cdot a_{i0}]$$

The steps of states are L F F F F F F F F F with highest probability of 1.3889796980374163e-08

**Image 10: Result of the Viterbi algorithm for the given problem.**

According to the proposed model, the majority of the throws are more likely to have been thrown with fair dice, while the rest of them were in a loaded dice.

**Problem 5:** According to the data given in the previous problem, analyze the probability that the seventh throw was rolled with loaded dice, regardless of the dice and results of the other throws.

For this question, the Backward algorithm must be used, since we want to know the probability that a specific position has been issued by a particular hidden state.

All the previous data remains the same, only the algorithm to use is changed. In the Backward algorithm options, we must enter the hidden state to search, in this case, L and the position, in this case, 7.

In the last step, we can check the result of the problem, where it's shown that the probability that the observed state in position 7 has been emitted by a loaded dice is 0.0178.

**Problem 6:** With the data of problem 4, what's the probability that a sequence of throws could be repeated some other day?

To calculate the probability that a sequence of throws occurs, the Forward algorithm must be used. This algorithm answers the question of what is the probability that a specific observed sequence will occur?

To solve this problem, all the other values remains the same, only the algorithm is changed.

The result of the algorithm is shown in the corresponding section.

This type of throws has a chance of 0.1% of occurring, which makes them very difficult to repeat.

Examples of problems that could be modeled with an HMM if there's a Markovian relationship between the hidden layers.

Layer		Examples	Program or DB exists?
Observed	Hidden		
Symptom	Disease		
Phenotype	Genotype	Genotype prediction	
Genotype	Phenotype	Trait prediction (e.g. face prediction)	
Protein (Seq.)	Structure	Secondary structure predictors	Yes
Gene (Seq.)	Function	Pfam	Yes
encrypted file	decrypted file		
Movie	Meaning, cause/effect		
Story	Moraleja		
Look	What you feel/think		
Seq. of sounds	Words	Rabiner, 1989	Yes
Seq. of actions	Intention		
News	Reality	Fake news prediction	
Protein (Sec.)	Transmembrane parts	TM-HMM	Yes