HMMTeacher 1.0
User Manual

Manual for the webserver
https://hmmteacher.mobilomics.org
https://hmmteacher1.mobilomics.org

July, 2021.

HMMTeacher is a user-friendly web server that allows: 1. the experience of modelling using Hidden Markov Models (HMMs) without the distraction of programming, and 2. it allows the understanding of the main algorithms associated, without the need of performing the long series of mathematical operations required.

The purpose of this manual is to answer the most common questions about HMMs in Section I, and guide the user in the execution of HMMTeacher, through practical exercises, in Section II.

Section I - Frequently Asked Questions - FAQ

1. **What is a Hidden Markov Model or HMM?**
   An HMM is a modelling 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 these 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 can answer with four algorithms:
   i. Probability of the sequence of observed states given a Model (Forward Algorithm).
   ii. Probability of a particular state being emitted by a particular hidden state (Backward Algorithm) given the Model.
   iii. 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. HMMTeacher assumes a predetermined modeled situation. Therefore, the parameters are given by the user. If you do not know how to model using HMMs, in Section II, there are a few examples of modelling.

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

An HMM consists of:

i. An alphabet of *m* observed states[1].
ii. An alphabet of *n* hidden states.
iii. A sequence of observed states, of length L [2].
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.

In order to be solve an HMM, several 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, is known and available. Section II includes a few examples of models and question that can be solved.

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 *Krought 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.

Additional problems are presented and solved in the next section of this manual.

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, and that you want to unhide, or discover. These hidden conditions affect the probability of 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; in the example of the observed song (previous question), the hidden states might be the set of music composers.

[1] We use a letter of an alphabet to represent the states for simplicity. It could be a set of expressions too.

[2] L bares no relationship with n or m. The sequence of observed states could be shorter or larger than L, and it does not need to include all possible states. Similarly, the sequence could include repeated states.

7. What is the transition matrix of an HMM?

It is an $n \times n$ 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[3] another hidden state in the previous moment or position. In our example, each cell in the matrix P is represented as a conditional probability,

$$P = \begin{bmatrix} P(F_t \,|F_{t-1}) & P(L_t \,|F_{t-1}) \\ P(F_t \,|L_{t-1}) & P(L_t \,|L_{t-1}) \end{bmatrix}$$

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. This assumption of the same probabilities anywhere in the sequence has consequences[4] in the results of the model, but greatly simplifies the problem. One important detail of matrix P is that the sum of probabilities in each and every row equals one.

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.

8. What is the initial probabilities vector?

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

$$\vec{v} = \begin{bmatrix} P(F_{t=1}) \\ P(L_{t=1}) \end{bmatrix}$$

Notice that the probabilities in $\vec{v}$ are not conditional. That is because there is no time or position before the first, i.e., $t = 1$. Like 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 an $n \times m$ 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). Therefore, E is,

$$E = \begin{bmatrix} P(1|F) & P(2|F) & P(3|F) & P(4|F) & P(5|F) & P(6|F) \\ P(1|L) & P(2|L) & P(3|L) & P(4|L) & P(5|L) & P(6|L) \end{bmatrix}$$

[3] The word "given" is used in conditional probabilities, in probability theory, meaning "conditioned to", or "restricted to", or "under". This conditional probability has the set of possibilities (the domain in which the event is considered. The denominator of the proportion) restricted by the condition. More, on Probability Theory by Jaynes (page 10).

[4] This assumption is called homogeneity. The consequence is that the sequence of observed states has roughly the same behavior or properties, at any region. This is generally not true in real life. But it allows the modelling of particular situations.

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, $P(1|F) = P(2|F) = P(3|F) = P(4|F) = P(5|F) = P(6|F) = 1/6$.

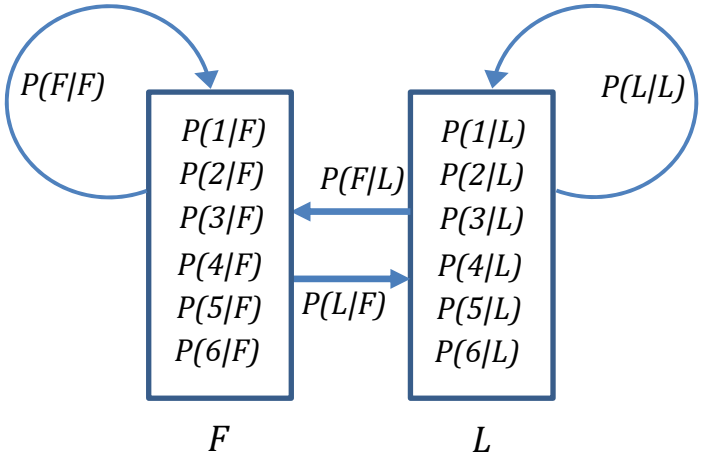
Like the initial vector $\vec{v}$ and the transition matrix P, the probabilities in each row of E, sum to one.

10. What does it mean the expression "training of an HMM"?

Besides stating the problem by defining the alphabet of observed states, the sequence of observed states, and the alphabet of hidden states, the parameters of an HMM consist of 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, for instance, 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

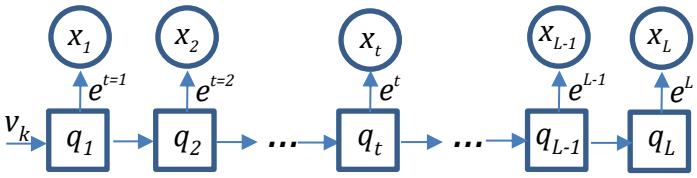


Which represent an HMM of the occasionally dishonest casino problem in the previous questions. The vector of initial probabilities is not represented in the graph.

There is at least one additional graphical representation for HMMs. It is presented in the solutions to the problems of Section II, of this manual.

12. How is it calculated a sequence of observed states, $x_1x_2 \dots x_{n-1}x_L$, from a known sequence of 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 may be represented sequentially, like this,



Note that this picture is not the HMM. It just represents what happens at each moment or position. Here, 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 of hidden states probabilities that could emit the same sequence of observed states.

14. What are the mathematical formulae for the HMM algorithms, Forward, Backward and Viterbi?

The development of the algorithms for efficient computation can be seen somewhere else (Question 15). The following formulae was derived from Biological Sequence Analysis by Krough et al., 2001, and Statistical Methods in Bioinformatics, an introduction by Grant and Ewans, 2005.

Forward algorithm:

Initialization.

$$\alpha(t=1,i) = \pi_i \bullet e_{S_i}(O_{t=1}) \quad \forall i$$

Recursion.

$$\alpha(t,i) = e_{q_t=S_i}(O_t) \bullet \sum_{j=1}^{N} \alpha(t-1,j) \cdot a_{ji} \quad \forall i$$

Termination.

$$P(O) = \sum_{i=1}^{N} \alpha(L,i)$$

Viterbi algorithm:

Initialization.

$$\delta_0(S_0) = 1$$
$$\delta_0(S_i) = 0 \quad \forall i > 0$$

Recursion.

$$\delta_t(S_i) = \pi_i \cdot e_{q_t=S_i}(O_t) \qquad 1 \le i \le N, \qquad t = 1$$
$$\delta_1(S_i) = \pi_i \cdot e_{q_1=S_i}(O_1) \qquad 1 \le i \le N$$

$$\delta_t(S_j) = e_{q_t=S_j}(O_t) \cdot \max_i[\delta_{t-1}(S_i) \cdot a_{ij}] \qquad 2 \le t \le L$$
$$1 \le j \le N$$
$$ptr_t(S_j) = \arg\max_i[\delta_{t-1}(S_i) \cdot a_{ij}]$$

Termination.

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

Backward algorithm:

Initialization.

$$\beta(L,k) = 1 \qquad \forall\ k,\ k = 1..N$$

Recursion.

$$\beta(t-1,i) = \sum_{j=1}^{N} a_{ij} \cdot e_{q_t=S_j}(O_t) \cdot \beta(t,j)$$

Termination.

$$P(q_{t^*} = S_k|O) = \frac{\alpha(t^*,k) \cdot \beta(t^*,k)}{P(O)}$$

15. Where can I read more about HMMs?

A few selected sources are available to understand the core mechanics and the theory of an HMM.

- "A tutorial on Hidden Markov Models and Selected Applications in Speech Recognition" - Lawrence R. Rabinner, 1989.
  - Well explained and show and excellent overview of the HMM process.
  - Available online.
- "What is a hidden Markov model?" – Sean R Eddy, 2004.
  - Concise, well written and understandable explanation of HMM centered in genome analysis.
  - Available online.
- 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 or PubMed.
- Finally, there are many videos online (Here and here) and tutorials of implementations in programming languages.
- If you know better references, we would like to include them in this manual. Please, write us an e-mail!

16. 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

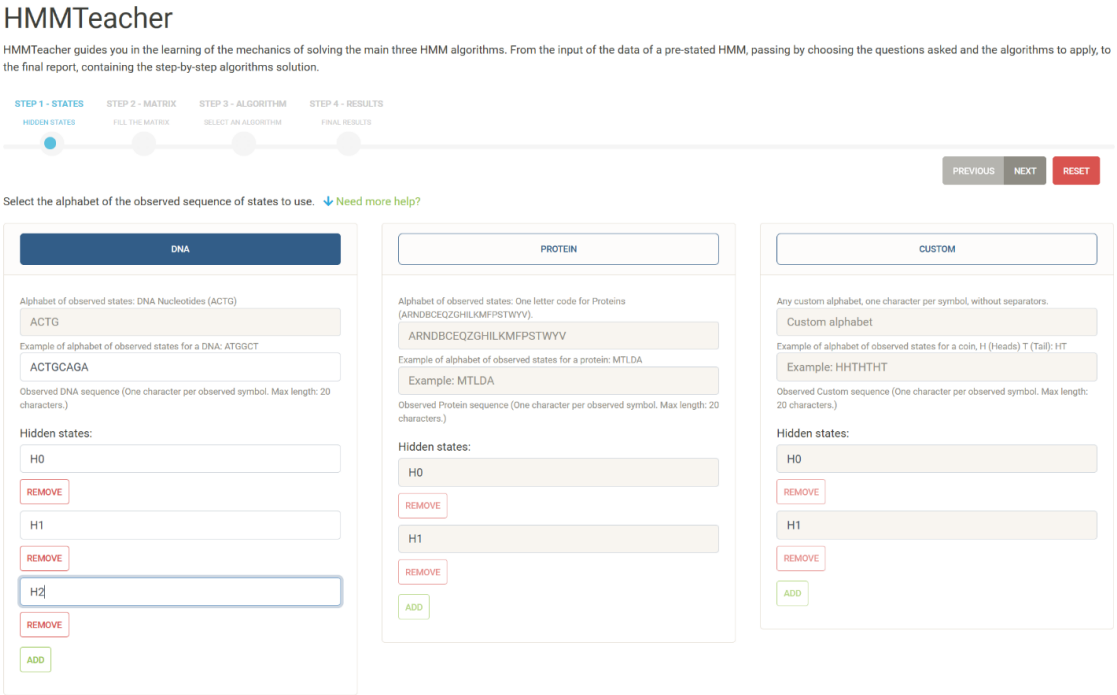
Section II – Practical problems of modelling, solutions and interpretations of HMMs results.

P0) Invent your own HMM problem (and solution!) using HMMTeacher.

HMMTeacher allows to generate at each step, random input data, in order to fill data unknown by the user. If the user does not know anything, just press the random button at the right of each matrix or vector, and the cells will be filled with random data that allows going to the next step. This will not help understanding or interpreting the results but will make the user familiar with the steps for solving HMMs.

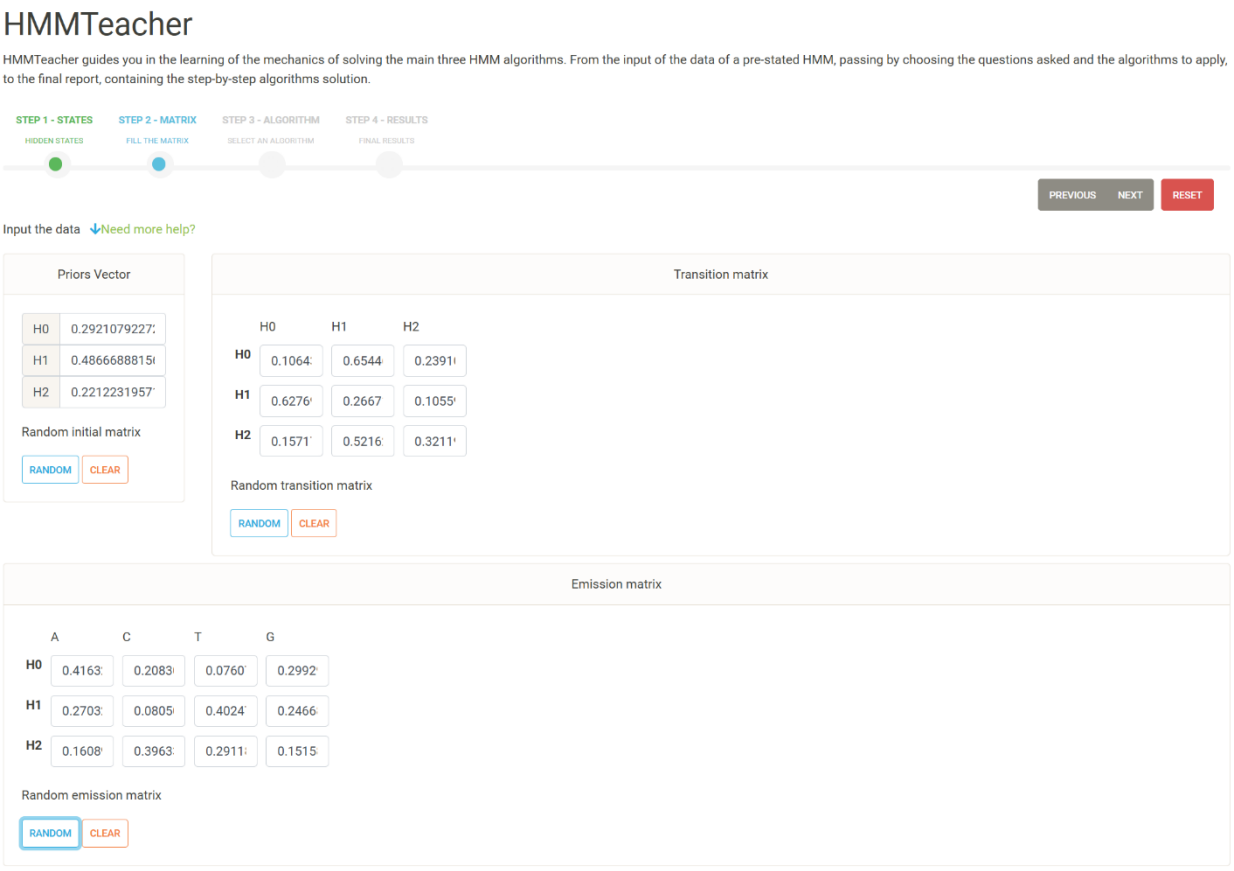
Solution:

Before starting to solve an HMM, a previously modelled problem should exist. However, HMMTeacher allows the training the mechanics of solving an HMM without the need of much information. The webpage (https://hmmteacher.mobilomics.org or https://hmmteacher1.mobilomics.org) starts guiding the user to choose a type of problem among three alternatives (Screenshot 1, DNA, Protein and Custom) and helps to generate hidden variables (the ones for which the sequence we are normally interested to discover) from problems of biological sequences, DNA and Protein, besides the possibility of starting from scratch solving a custom HMM. We will start by choosing to solve an unknown problem on DNA sequences. If we need additional guidance, we can watch the Screencasts (videos showing how to fill the cells in each step), and this manual, in the links on the top left of the page. HMMTeacher shows how the DNA alphabet is already set, and an example observed sequence that can be changed by the user. Two hidden variables are set by default, H0 and H1. We will add another hidden variable, H2 (Screenshot 1) and go to the next step.



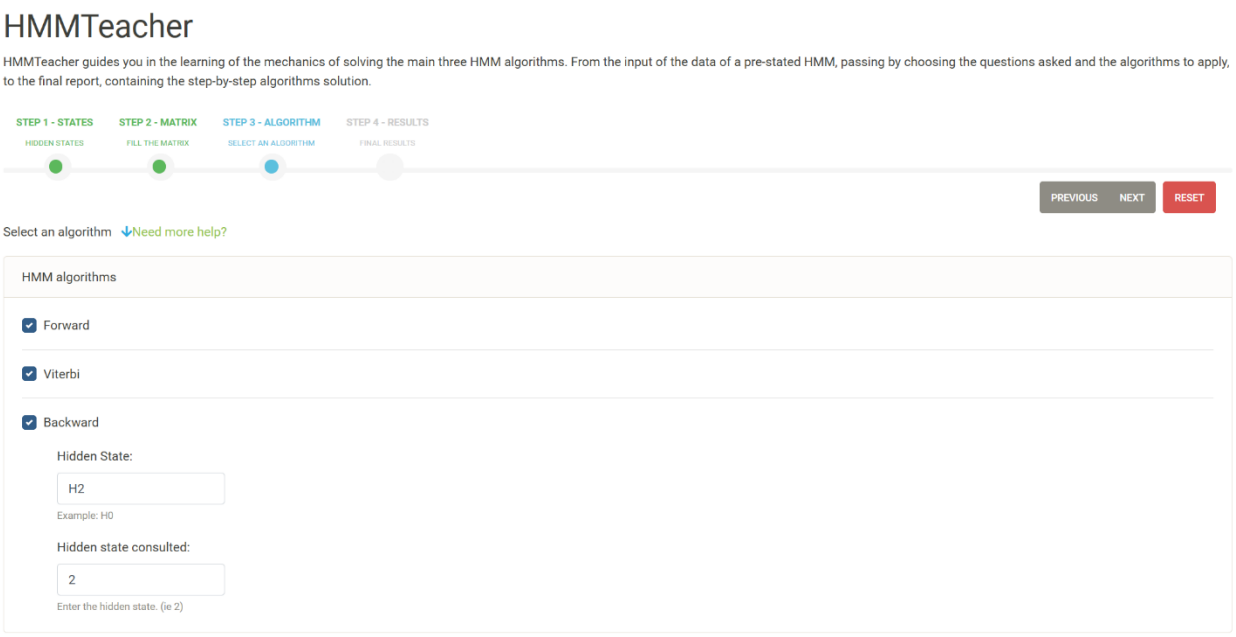
Screenshot 1: HMMTeacher first step. Selecting the DNA option and using a random sequence. An additional hidden state is added.

After setting the alphabet and hidden variables, the second step in solving an HMM is setting three elements: the vector of initial probabilities of the hidden variables, the transition probability matrix of hidden variables and the emission probability matrix between hidden and observed variables (Screenshot 2). If this information were not available, the user may generate random values and modify them. The only restriction is that the sum of the values must add to one (please, see questions 8 and 9 of the section I).

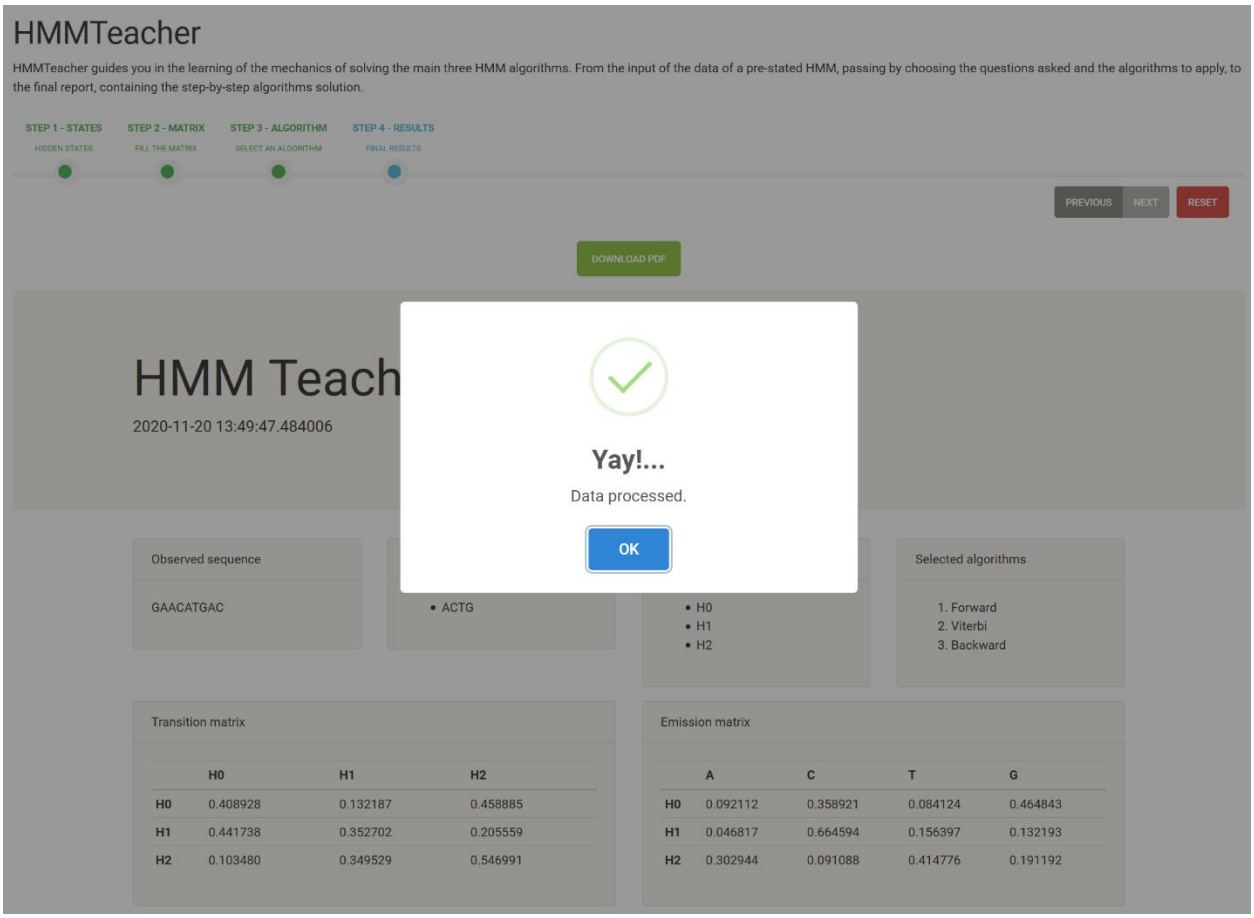


Screenshot 2: HMMTeacher second step. It is possible to press the random button in the required sections to generate random data.

The values of these matrices set the HMM. The meaning of the variables in the context of the problem is, of course, central for troubleshooting the model and the interpretation of solution. The next step is to answer questions, choosing among the alternatives Forward, Backward and Viterbi, at least one. As explained in question 2, each algorithm answers a particular question from the HMM (Screenshot 3).



Screenshot 3: HMMTeacher third step. Possible algorithms to select in order to answering questions about the HMM.



Screenshot 4: HMMTeacher fourth step. After the processing is done the results will be displayed in an orderly manner with the option to also download a PDF.

P1) In the middle of a downtown street you see a kid is betting with the people passing by. He challenges the victim to guess the result of the toss of his coin. The kid wins frequently. It is so much, you want to test the hypothesis that the kid has two coins, one fair, F, and one loaded, L. You think, somehow, he manages to choose one of the coins without being noticed (with certain transition probabilities), and then he tosses it. You record the (observed) results of Heads (H) and Tails (T), for some time (n = 20). Build an HMM with the parameters,

Initial probabilities:

F	L
0.3	0.7

Transition matrix

	F	L
F	0.45	0.55
L	0.2	0.8

Emission matrix

	H	T
F	0.2	0.8
L	0.5	0.5

And want to answer the following questions:

a) What is the probability of the observed run HHHTTTHHTHTHHTHHTHHT, of tosses?

b) What would be the most probable sequence of hidden choices of coins in this observed run, by the kid?

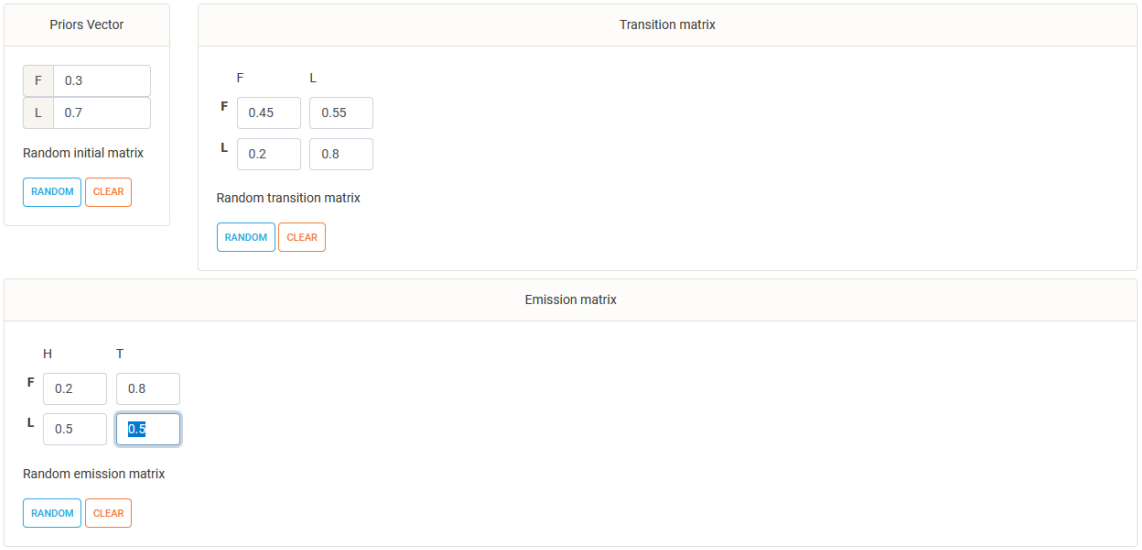
c) On the 5^{th} and 6^{th} toss, you think the coin is the loaded one. What are the probabilities on those tosses that the coin was loaded? Does the answer in b), confirms your hint?

Solution:

We start by modelling the problem in the step 1 of HMMTeacher. The observed sequence given will be “HHHTTTHHTHTHHTHHTHHT”.

Screenshot 5: Modelling a custom problem in HMMTeacher.

In the step 2 we input the probabilities given in the problem.



Screenshot 6: Inputting the given data.

To answer question **a)** we use the Forward algorithm. Selecting it in the third step give us the next result.

P(O: HHHTTTHHHTHTHHTHHTHHT) = α(20,1) + α(20,2)
P(O: HHHTTTHHHTHTHHTHHTHHT) = (1.177020226522136e-07) + (2.502567025350946e-07) = 3.679587251873082e-07

Screenshot 1: Result of the Forward algorithm.

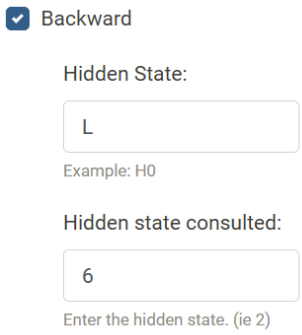
The probability of the given observed sequence might seem extremely low. But if we compare this probability to the probability of the same observed sequence using a fair coin, $\frac{1}{2^{20}} = 9.5\ x\ 10^{-7}$, we have only a slightly lower value. This shows that the result of a probability is only low or high compared to other probabilities. In this case, and as typically happens in HMMs, the order of magnitude of a probability is expected to be low considering the number of possible combinations.

Answering the question **b)** requires the use of the Viterbi algorithm. If we select this in the third step will give us the most probable sequence.

The most probable sequence of hidden states is L L L L L L L L L L L L L L L L L L L L with probability of 9.620726743040008e-09

Screenshot 2: Result of the Viterbi algorithm.

Although this might, again, look like a low probability, this is the largest probability among all $2^{20} = 1.048.576$ combinations of hidden states. Therefore, the coin used in all tosses was probably the loaded one.
To answer the question **c)** the Backward algorithms is required. We need to input the state we want to look (L) and the position to look for (5th and 6th).



Screenshot 3: Backward options in the step 3 of HMMTeacher.

With that, we can check the result for the specified position in the last step of HMMTeacher.

P(q_5 = L|HHHTTTHHHTHTHHTHHTHHT) = α(5,2) * β(5,2) / P(O)
P(q_5 = L|HHHTTTHHHTHTHHTHHTHHT) = (0.015531994 * 1.4233287117147658e-05)/3.679587251873082e-07 = 0.600804696209552

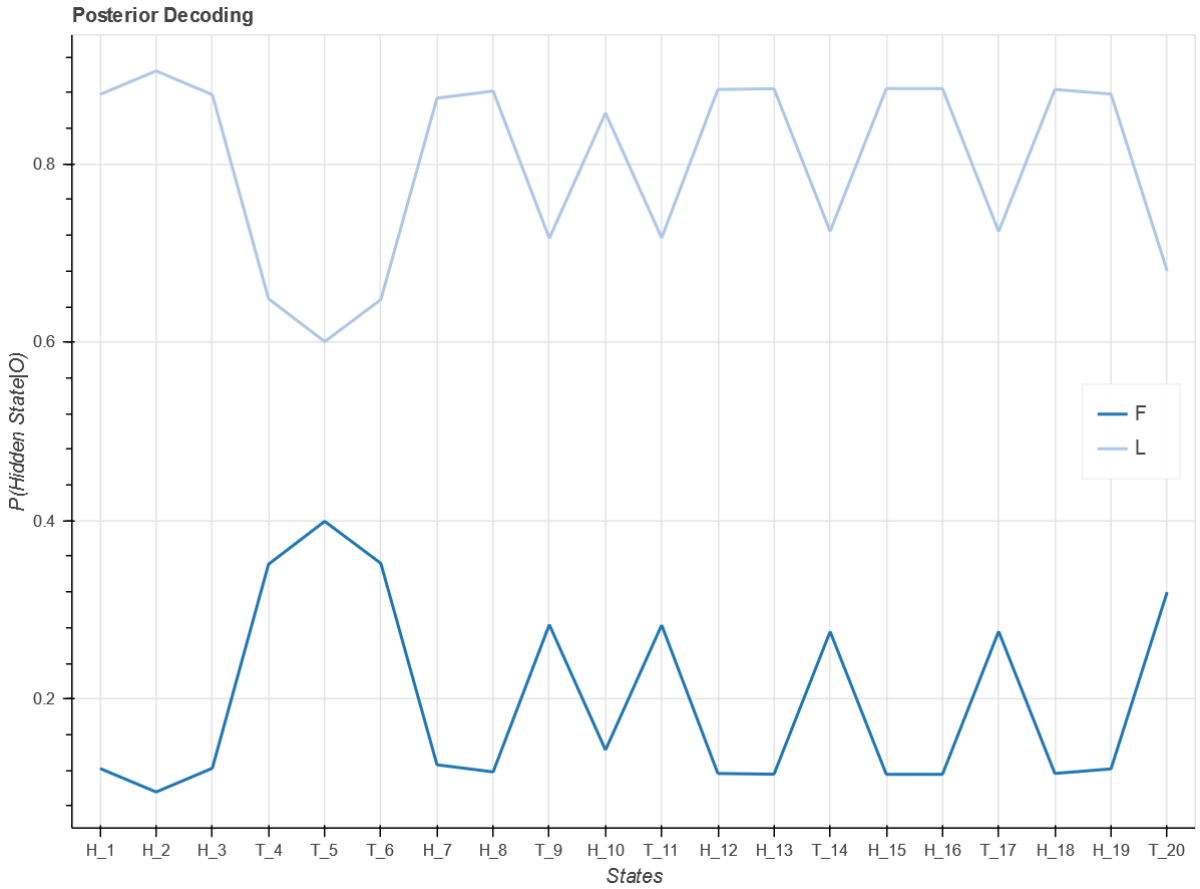
Screenshot 4: Result of the Backward algorithm for the 5th position.

$P(q_6 = L|HHHTTTHHTHTHHTHHTHHT) = \alpha(6,2) * \beta(6,2) / P(O)$

$P(q_6 = L|HHHTTTHHTHTHHTHHTHHT) = (0.008866750440000001 * 2.688730826309062e-05)/3.679587251873082e-07 = 0.6479070505824155$

Screenshot 5: Result of the Backward algorithm for the 6th position.

c) The Backward probabilities on the 5th and 6th positions are 0.6 and 0.64, respectively. But probabilities only have a meaning when compared to other probabilities. If we repeat the Backward exercise, comparing these with the probability of these positions to be emitted by the Fair dice, we have 0.399 and 0.3520, respectively. In this case, the probabilities are complementary because there are only two hidden states. Effectively, the probability of the 5th and 6th positions are larger if the emitting hidden state is Loaded, instead of Fair, which confirms the result given by Viterbi algorithm.



Screenshot 12: Posterior decoding chart of Problem 1.

The chart of the posterior probabilities at each position given the different hidden states of the observed sequence, calculated by the Backward algorithm is also called Posterior decoding. This chart is particularly useful to find patterns of (hidden) properties in observed sequences which are so long, that have very similar probabilities. Posterior decoding allows to find small regions where the probabilities have large changes, pointing out the pattern. More on Posterior decoding in Biological Sequence Analysis, chapter 3, on “Markov chains and hidden Markov models”.

P2) from the book Biological Sequence Analysis: On a 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. The croupier starts with the fair dice four out five times he rolls them.

a) Draw a graph representing the HMM.
b) Evaluate the probability of the following observed sequence of 13 rolls x: 1 5 2 4 3 6 6 6 6 6 6 2 4 1 5 3 1 4
c) Show the most probable sequence of dices used for all the tosses.
d) Analyze the probability that the seventh throw was rolled with a loaded dice, regardless the dice and results of the other throws.
   Hint: 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 hidden state.

Solution:

This is very similar to Problem 1, but with a larger alphabet of observed states. To solve this problem is necessary to state the HMM, establishing the interactions and the probabilities between each pair of states. But first, if you already used HMMTeacher, be sure of resetting the page (right most button in the start page). This ensures that any previous input parameters will not affect the modelling of the current problem.

The result is the HMM shown as a graph, below:

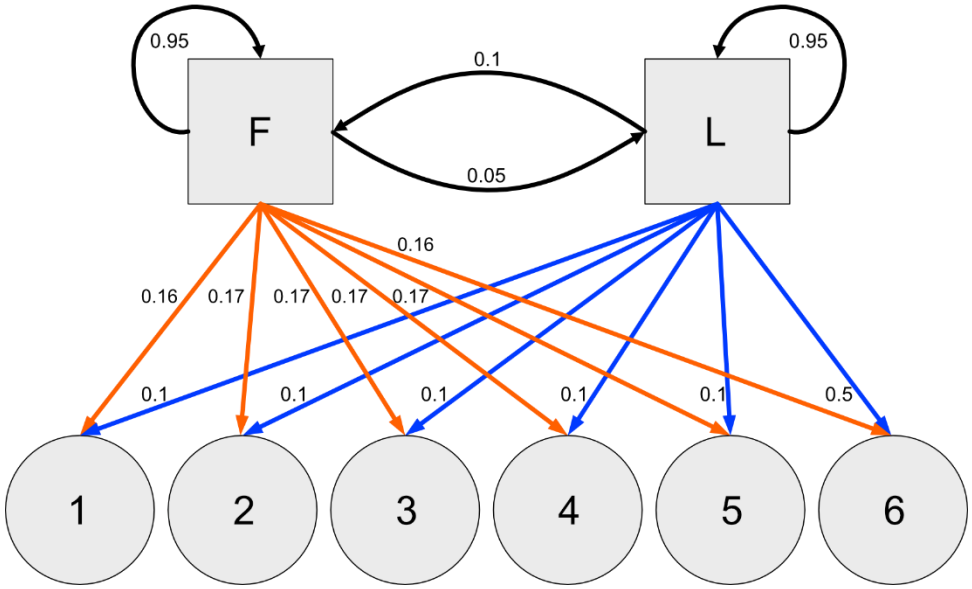
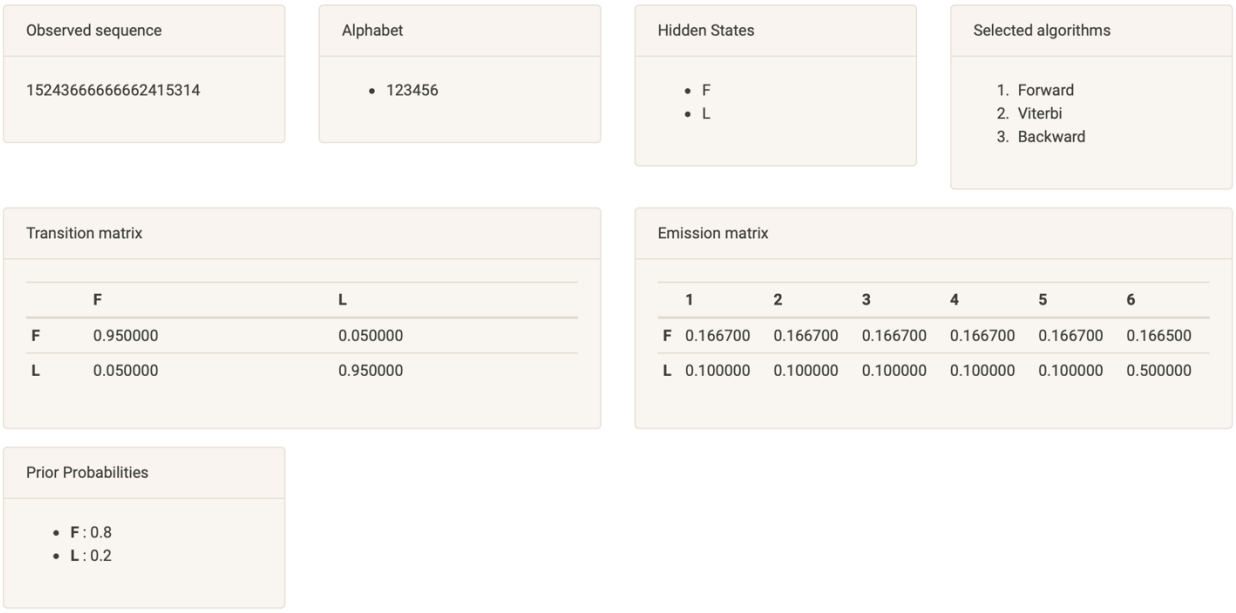


Figure 1: HMM diagram of problem 2. The square nodes are the hidden states. The circle nodes correspond to the emitted values. The edges, in orange and blue, correspond to the conditional probabilities on Fair (F) and Loaded (L) states, respectively.

The parameters are as follow (Screenshot from HMMTeacher):



Observed sequence

1524366666662415314

Alphabet

- 123456

Hidden States

- F
- L

Selected algorithms

1. Forward
2. Viterbi
3. Backward

Transition matrix

	F	L
F	0.950000	0.050000
L	0.050000	0.950000

Emission matrix

	1	2	3	4	5	6
F	0.166700	0.166700	0.166700	0.166700	0.166700	0.166500
L	0.100000	0.100000	0.100000	0.100000	0.100000	0.500000

Prior Probabilities

- **F** : 0.8
- **L** : 0.2

Screenshot 13: Parameters of problem 2.

b) The probability of the observed sequence 1 5 2 4 3 6 6 6 6 6 6 6 2 4 1 5 3 1 4 is given by the Forward algorithm (a screenshot follows):

$$P(O) = \sum_{i=1}^{N} \alpha(L, i)$$

P(O: 1524366666662415314) = α(20,1) + α(20,2)

P(O: 1524366666662415314) = (1.4588887091106564e-14) + (3.612551653376652e-15) = 1.8201438744483217e-14

Screenshot 14: Result of the backward algorithm in problem 2.

c) The most probable sequence of states is given by the Viterbi algorithm. Screenshot follows:

Matrix of δ values

	1_1	5_2	2_3	4_4	3_5	6_6	6_7	6_8	6_9	6_{10}	6_{11}	6_{12}	6_{13}	2_{14}	4_{15}	1_{16}
F	0.133360	0.021120	0.003345	0.000530	0.000084	0.000013	0.000002	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
L	0.020000	0.001900	0.000181	0.000017	0.000003	0.000002	0.000001	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0

Termination

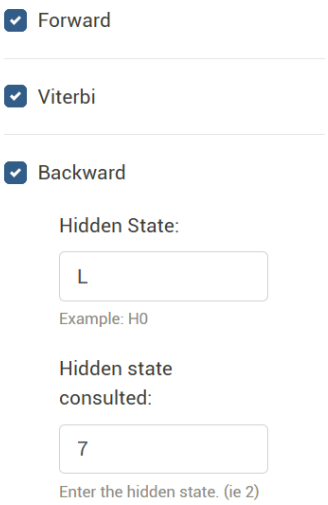
$$P(O \cdot Q) = \max_i[\delta_L(S_i) \cdot a_{i0}]$$
$$q_L^* = arg\max_i[\delta_L(S_i) \cdot a_{i0}]$$

The most probable sequence of hidden states is F F F F F L L L L L L L L F F F F F F with probability of 1.5042447977071585e-15

Screenshot 15: Screenshot of the result of the backward algorithm in problem 2.

According to the model, the croupier cheated between (around) the 6th and the 13th roll.

d) 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 HMMTeacher, in the step of choosing the question (and algorithm) to apply, select Backward as follows,



Screenshot 16: Insertion of hidden state and the position consulted.

In the last step we can check the result of the problem, where it is shown that the probability that the observed state in the position 7 has been emitted by a loaded dice is 0.3924.

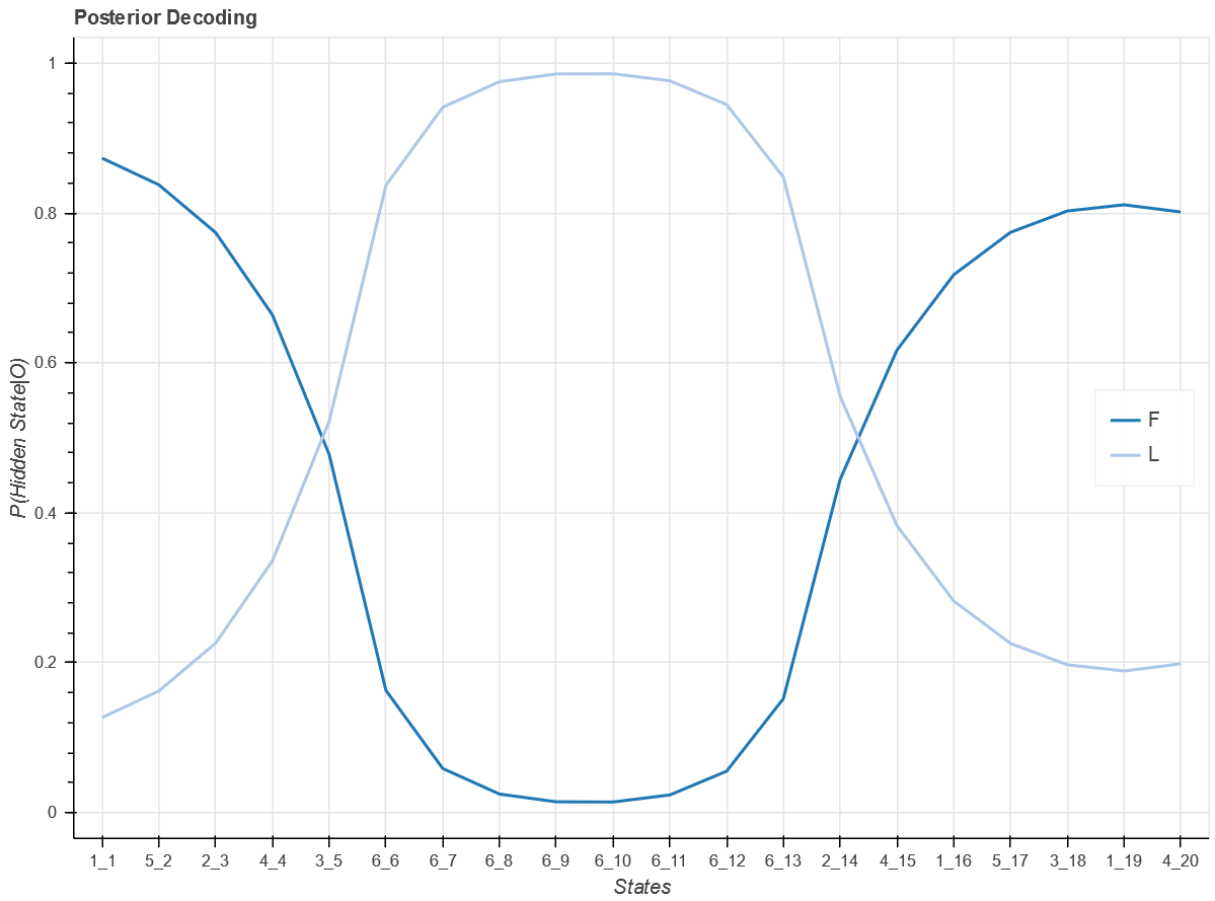
$$P(q_{t^*} = S_k|O) = \frac{\alpha(t^*,k) \cdot \beta(t^*,k)}{P(O)}$$

P(q_7 = L|15243666666662415314) = α(7,2) * β(7,2) / P(O)

P(q_7 = L|15243666666662415314) = (3.04844630357736e-06 * 5.6202338043791265e-09)/1.8201438744483217e-14 = 0.9412981691567108

Screenshot 17: Screenshot of the result of the backward algorithm in problem 2.

The Backward algorithm results, 0.94 of probability for the dice at the 7th roll being the loaded one, confirms Viterbi results.



Screenshot 18: Posterior decoding chart of Problem 2.

P3) You go to a casino, and see a croupier tossing a die, like the kid from problem 1 with a coin. You watch for some time, record the numbers of the tosses of the dice, and estimate the parameters of the HMM,

Initial probabilities:

F	L
0.45	0.55

Transition matrix

	F	L
F	0.2	0.8
L	0.5	0.5

In modelling the problem, you are not sure if the emission matrix is

Emission matrix 1

	1	2	3	4	5	6
F	0.2	0.1	0.1	0.1	0.3	0.2

L	0	0.1	0.2	0.3	0.2	0.2

Or

Emission matrix 2

	1	2	3	4	5	6
F	0.3	0.2	0.1	0.1	0.2	0.1
L	0.4	0.2	0.2	0.1	0.1	0.0

Questions:

a) What is the probability of the following observed run: 4 3 2 5 1 6 3 4 5 2 1 3 4 5 1 6 4 5 2, of rolls using each of the emission matrices?
b) Which do you think, is the correct emission matrix? Why?
c) How many times the croupier changes dice, in the observed run?

Solution:

a) The probability of the observed run of tosses with the first emission matrix is:

P(O: 4325163452134516452) = α(19,1) + α(19,2)
P(O: 4325163452134516452) = (1.585220697647461e-16) + (3.0993747766446697e-16) = 4.684595474292131e-16

Screenshot 19: Probability of the observed with the first emission matrix given by the Forward algorithm.

And with the second one is:

P(O: 4325163452134516452) = α(19,1) + α(19,2)
P(O: 4325163452134516452) = (3.6908270007084595e-17) + (7.964416159423517e-17) = 1.1655243160131977e-16

Screenshot 20: Probability of the observed with the second emission matrix given by the Forward algorithm.

b) The first emission matrix it is the correct one, as the probability of emitting the observed sequence of rolls is higher with that one.

c) Using the first emission matrix, the croupier changes dices 10 times,

Matrix of δ values

	4_1	3_2	2_3	5_4	1_5	6_6	3_7	4_8	5_9	2_{10}	1_{11}	3_{12}	4_{13}	5_{14}	1_{15}	6_{16}
F	0.045000	0.008250	0.000825	0.000124	0.000013	0.000001	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
L	0.165000	0.016500	0.000825	0.000132	0.000000	0.000002	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0

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 most probable sequence of hidden states is L L F L F L L L F L F L L L F L L F L with probability of 2.6276659200000016e-18

Screenshot 21: The most probable sequence of states given by the Viterbi algorithm with the emission matrix 1. The backtracking shown is partial up to the 16th roll.

This the most probable sequence of dices is different when using the second emission matrix. Here the croupier changes dice 11 times.

Matrix of δ values

	4_1	3_2	2_3	5_4	1_5	6_6	3_7	4_8	5_9	2_{10}	1_{11}	3_{12}	4_{13}	5_{14}	1_{15}	6_{16}
F	0.045000	0.002750	0.000720	0.000072	0.000009	0.000001	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
L	0.055000	0.007200	0.000720	0.000058	0.000023	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0

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 most probable sequence of hidden states is F L L F L F L L F L F L L F L F L F L with probability of 3.6238786560000047e-19

Screenshot 22: The most probable sequence of states given by the Viterbi algorithm with the emission matrix 2.

Comparing the probability of the most probable sequence of hidden states given by the Viterbi algorithm, between the two emission matrices, confirms the conclusion in part a) by the Forward algorithm.

P4) From L. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition". Proceedings of the IEEE, Vol. 77, No2, February 1989.

Is it possible to use HMMTeacher, to solve a discrete (non-Hidden) Markov chain? Consider a simple 3 state Markov model of the weather.

- State 1: Rainy (R)
- State 2: Cloudy (C)
- State 3: Sunny (S)

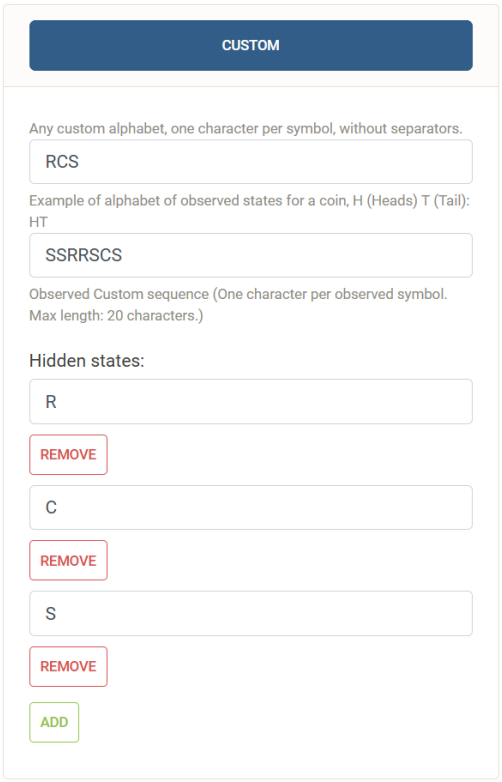
The transition matrix is, as follows:

	R	C	S
R	0.4	0.3	0.3
C	0.2	0.6	0.2
S	0.1	0.1	0.8

Given that the weather on day 1 is Sunny (state 3), we can ask the question: what is the probability (according to the model) that the weather for the next 7 days will be "sun-sun-rain-rain-sun-cloudy-sun"?

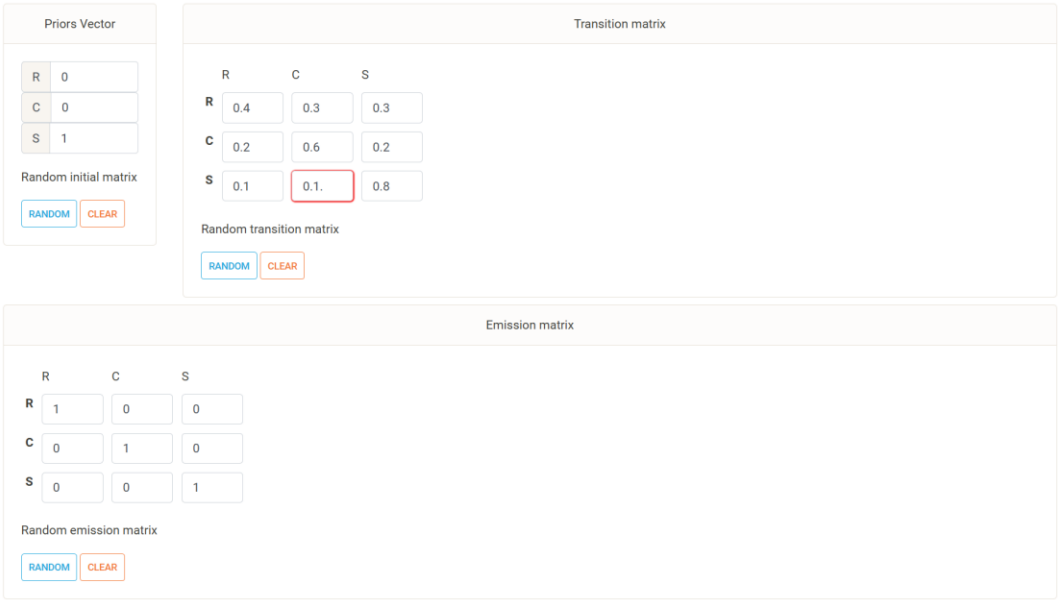
Solution:

We start by modelling the problem in HMMTeacher (Step 1). Our custom observed alphabet is RCS, the observed sequence is "SSRRSCS" and the hidden states are also R, C, S.



Screenshot 23: Modelling a non-hidden Markov Model in HMMTeacher.

As the observed sequence start with an "S", we give the prior vector option a value of 1. The transition matrix is the one given in the problem, and the emission matrix can be constructed accordingly to the emission of each state. The trick to use HMMTeacher as a solver for a Markov chain is to assign a value of 1 in each corresponding cell of the emission matrix.



Screenshot 24: Required data for problem 4.

With that, we select the Forward algorithm in the third step and in the final step the results are displayed.

$$P(O) = \sum_{i=1}^{N} \alpha(L, i)$$

P(O: SSRRSCS) = α(7,1) + α(7,2) + α(7,3)
P(O: SSRRSCS) = (0.0) + (0.0) + (0.00019200000000000009) = 0.00019200000000000009

Matrix of α values

	S_1	S_2	R_3	R_4	S_5	C_6	S_7
R	0.000000	0.000000	0.080000	0.032000	0.000000	0.000000	0.000000
C	0.000000	0.000000	0.000000	0.000000	0.000000	0.000960	0.000000
S	1.000000	0.800000	0.000000	0.000000	0.009600	0.000000	0.000192

Screenshot 25: Result with the Forward algorithm.

The probability that the weather for the next 7 days will be “sun-sun-rain-rain-sun-cloudy-sun is 0.0001920.

Viterbi algorithm would give us the most probable weather sequence in the observed week, which is, as expected the same sequence of observed weather states. After all, this is a Markov chain with no hidden states.

P5) from (https://www.cs.hmc.edu/~yjw/teaching/cs158/lectures/17_19_HMMs.pdf)

One biological application of HMMs is to determine the secondary structure (i.e. the general three-dimensional shape) of a protein. This general shape is made up of alpha helices, beta sheets, and other structures. In this problem, we will assume that the amino acid composition of these regions is governed by an HMM. To keep this problem relatively simple, we do not use actual transition values or emission probabilities. The start state is always “other”. We will use the state transition probabilities and emission probabilities below.

Transition matrix

	Alpha	Beta	Other
Alpha	0.7	0.1	0.2
Beta	0.2	0.6	0.2
Other	0.3	0.3	0.4

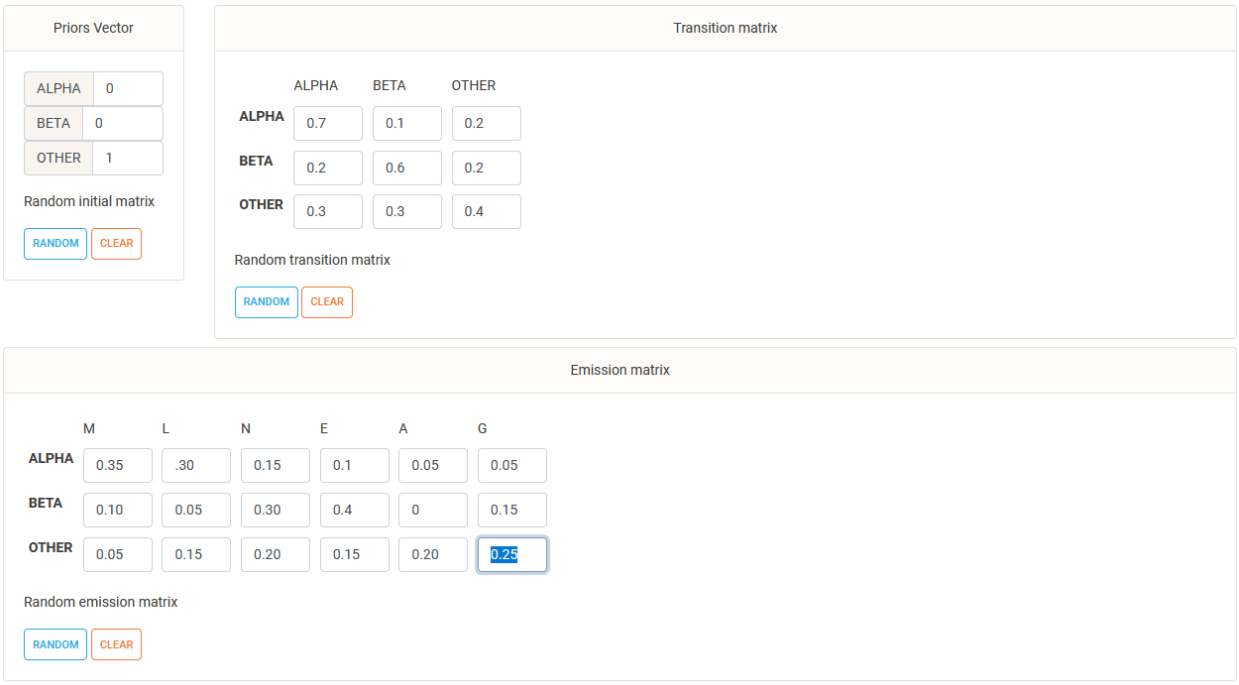
Emission matrix

Amino acid	Alpha	Beta	Other
M	0.35	0.10	0.05
L	0.30	0.05	0.15
N	0.15	0.30	0.20
E	0.10	0.40	0.15
A	0.05	0.00	0.20
G	0.05	0.15	0.25

a) What is the probability P (q = O, O = ML)?
Solution:
We start by modelling the problem in HMMTeacher. In the first step we put the data as is given in the problem.

Screenshot 66: Modelling the fifth problem.

In the second step we input the required data. As the start state is always “other” we input a 1 in the corresponding prior value option.



Screenshot 7: Data for the problem 5.

The probability of observing the ML sequence is given by the Forward algorithm.

P(O: ML) = α(2,1) + α(2,2) + α(2,3)

P(O: ML) = (0.0045) + (0.00075) + (0.0030000000000000005) = 0.00825

Screenshot 8: The result of the Forward algorithm.

b) How many paths could give rise to the sequence *O* = *MLN*? What is the total probability *P*(*O*)?
Solution:
In the first position only one hidden state is allowed, "other", as the initial probabilities state that the probability of "other" is 1. "alpha" and "beta" hidden states are allowed in the positions 2 and 3. Therefore, we have $1x3x3 = 9$ paths of hidden states can result in the observed sequence MLN. The total number of paths, i.e., possible hidden states emitting all possible observed sequences is $9 * 6^3 = 162$.

The probability of O= MLN is obtained with the Forward algorithm.

P(O: MLN) = α(3,1) + α(3,2) + α(3,3)

P(O: MLN) = (0.0006299999999999999) + (0.00054) + (0.0004500000000000001) = 0.0016200000000000001

Screenshot 9: Probability of MLN given the model (Forward).

c) Give the most likely state transition path *q** for the amino acid sequence *MLN* using the Viterbi algorithm. What is *P* (*q**, *O*)?

Solution:

Using the Viterbi algorithm, the most probable sequence of hidden states is:

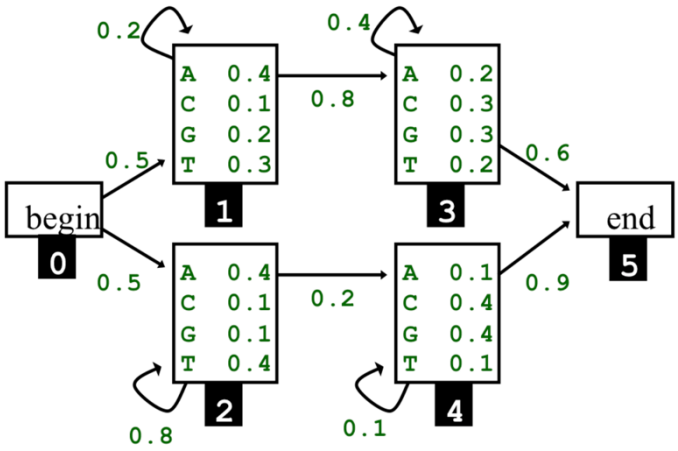
The most probable sequence of hidden states is OTHER ALPHA ALPHA with probability of 0.00047249999999999994

Matrix of δ values

	M_1	L_2	N_3
ALPHA	0.000000	0.004500	0.000472
BETA	0.000000	0.000750	0.000270
OTHER	0.050000	0.003000	0.000240

Screenshot 30: Viterbi result.

Two Problems from https://www.cs.swarthmore.edu/~soni/cs68/s17/Labs/hmm.html
P6)



Screenshot 31: Example HMM for Problem 6.

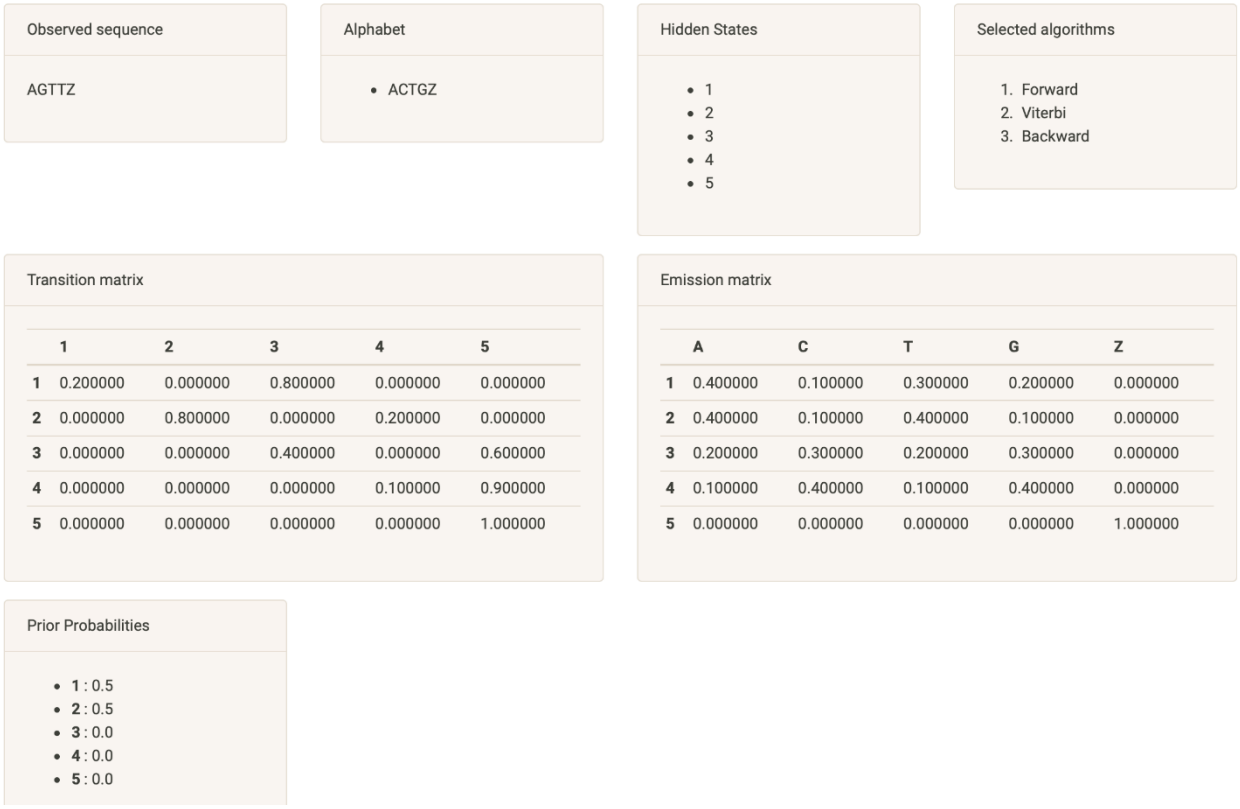
Above is an example HMM that we used in class. Use this HMM for the following questions:

a) Apply the forward algorithm to infer the probability of observing the sequence **AGTT**. Show each step of the recursion by filling a 2D matrix of forward probabilities $\mathbf{f_{k(i)}}$.
b) Compute the backward probability as well showing your steps and the resulting backward matrix, $\mathbf{b_{k(i)}}$.
c) Infer the most likely path through the states of the HMM for the sequence **TATA**. Again, show the steps of your algorithm. Your answer should indicate the optimal path as well as the probability of that path.

Solution:

This problem is particularly interesting because it includes begin and end states, which so far, we had not considered. We will include the begin state probabilities in the prior probabilities vector, whereas the end state will be considered as a hidden state that has 100% of probability of recurring. As the end state is hidden variable, it will appear in the emission matrix, so, it has to emit a nucleotide, imposing a difficulty to the modelling. In order to circumvent this and be consistent with the modelling, we will say that the end state emits a Z observed nucleotide with 100% of probability and add a Z at the end of the sequence of observed states. We could do the same thing for the begin state, but in this case, there is no need.

The parameters are as follows:



	1	2	3	4	5
1	0.200000	0.000000	0.800000	0.000000	0.000000
2	0.000000	0.800000	0.000000	0.200000	0.000000
3	0.000000	0.000000	0.400000	0.000000	0.600000
4	0.000000	0.000000	0.000000	0.100000	0.900000
5	0.000000	0.000000	0.000000	0.000000	1.000000

	A	C	T	G	Z
1	0.400000	0.100000	0.300000	0.200000	0.000000
2	0.400000	0.100000	0.400000	0.100000	0.000000
3	0.200000	0.300000	0.200000	0.300000	0.000000
4	0.100000	0.400000	0.100000	0.400000	0.000000
5	0.000000	0.000000	0.000000	0.000000	1.000000

Screenshot 32: Parameters for Problem 6. Note the extra symbol Z added to the end of the observed sequence. The begin state transition probabilities are included in the prior vector. The end state is a hidden self-recurring state with 100% of probability of emitting the Z (equivalent to a STOP).

a)

Termination

$$P(O) = \sum_{i=1}^{N} \alpha(L, i)$$

P(O: AGTTZ) = α(5,1) + α(5,2) + α(5,3) + α(5,4) + α(5,5)

P(O: AGTTZ) = (0.0) + (0.0) + (0.0) + (0.0) + (0.00038832000000000016) = 0.00038832000000000016

Screenshot 33: Forward result for the P6.

b)

Matrix of δ values

	A_1	G_2	T_3	T_4	Z_5
1	0.200000	0.008000	0.000480	0.000029	0.000000
2	0.200000	0.016000	0.005120	0.001638	0.000000
3	0.000000	0.048000	0.003840	0.000307	0.000000
4	0.000000	0.016000	0.000320	0.000102	0.000000
5	0.000000	0.000000	0.000000	0.000000	0.000184

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 most probable sequence of hidden states is 1 3 3 3 5 with probability of 0.00018432000000000008

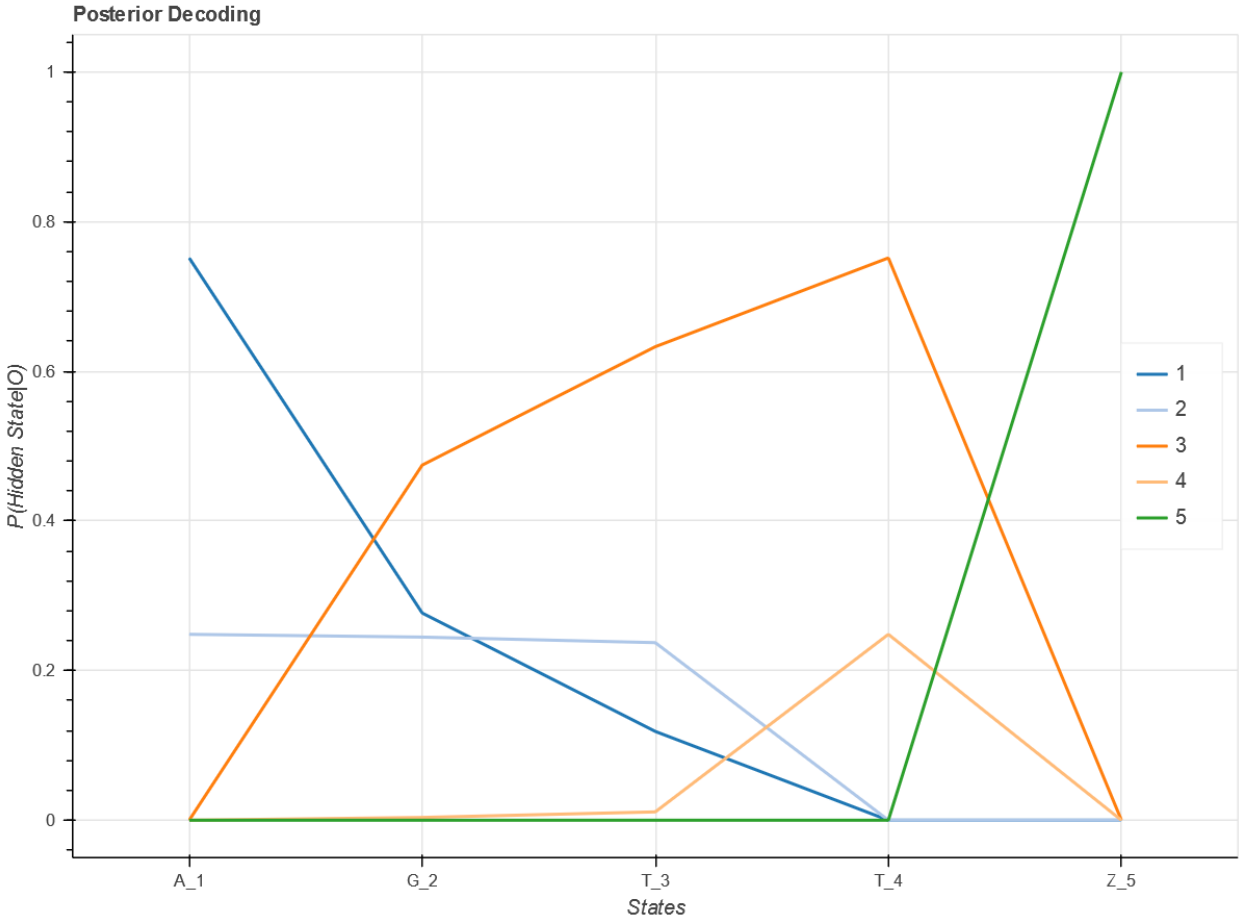
Screenshot 34: Viterbi backtracking and probability of the most probable sequence of hidden states for Problem 6.

Hidden States	A1	G2	T3	T4	Z5
1	0.75155	0.27689	0.11867	0.00000	0.00000
2	0.24845	0.24475	0.23733	0.00000	0.00000
3	0.00000	0.47466	0.63288	0.75155	0.00000
4	0.00000	0.00371	0.01112	0.24845	0.00000
5	0.00000	0.00000	0.00000	0.00000	1.00000
Sum	1.00000	1.00000	1.00000	1.00000	1.00000

Table 1: Posterior decoding for Problem 6. The cells contain the probability of each the hidden state, given the observed state at each position. The last row shows how the sum of the probabilities is 1, for each position, as expected.

Begin and end states are important in modelling HMMs, because adding them solve a mathematical restriction on the length of the observed sequences allowed as input to the HMM, that otherwise is there. For more information on the features that begin, and end states add to

HMMs, please refer to the book Biological Sequence Analysis, chapter 3 on “Markov chains and hidden Markov models”.



Screenshot 35: Posterior decoding chart of Problem 6.

d) The Viterbi algorithm shows that the sequence of hidden states “2 2 2 2” has the highest probability of producing the TATA sequence.

Matrix of δ values

	T_1	A_2	T_3	A_4
1	0.150000	0.012000	0.000720	0.000058
2	0.200000	0.064000	0.020480	0.006554
3	0.000000	0.024000	0.001920	0.000154
4	0.000000	0.004000	0.001280	0.000410
5	0.000000	0.000000	0.000000	0.000000

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 most probable sequence of hidden states is 2 2 2 2 with probability of 0.006553600000000003

Screenshot 36: Viterbi algorithm result for Problem 6.

P7) For the following set of questions, consider the following problem. Assume you have three boxes, each containing a certain number of apples and oranges. At any point in time, you select a box at random, and then a fruit from that box (i.e., an apple or orange) and record your finding (**A** for apple and **O** for orange). You immediately replace the fruit so that the total number of apples and oranges stays the same over time and repeat the process. Unfortunately, you forgot to write down the boxes you chose and simply have an account of apples and oranges. Assume the following quantity of fruits:

- **Box 1**: 2 apples, 2 oranges
- **Box 2**: 3 apples, 1 orange
- **Box 3**: 1 apple, 3 oranges

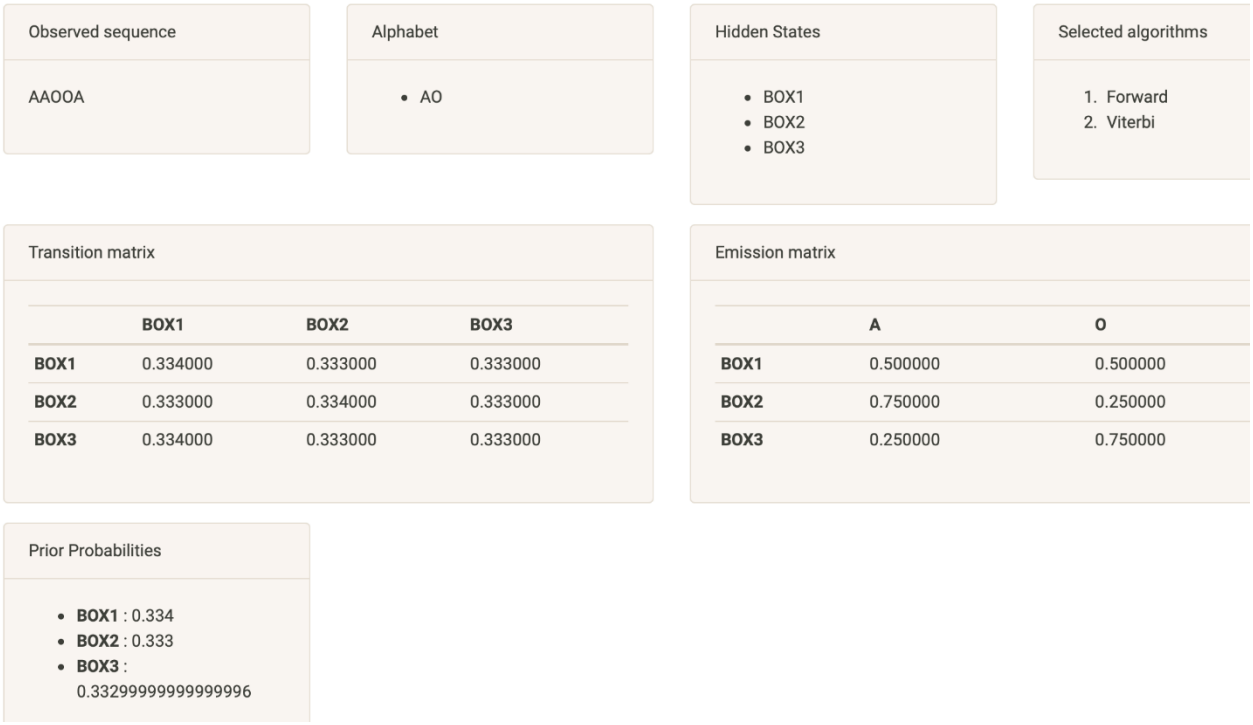
a) Draw a hidden Markov model to represent this problem. Show a state diagram in addition to two-dimensional parameter arrays **a** (for transitions) and **e** (for emission probabilities).
b) Compute the probability of seeing box sequence **π = (1,1,3,3,2)** and fruit sequence **x = (A, A, O, O, A)**. Show your work.
c) Compute the optimal set of boxes corresponding to the fruit sequence given in the previous problem (i.e., **π***). That is, which box was each piece of fruit most likely to be selected from?
d) How much better is your path than that given in part b)? Compute this value by using a log-odds ratio; that is:

$$\log \frac{P(\pi * |x)}{P(\pi_b|x)}$$

where the denominator is using the path from problem 4.

Solution:

a) The parameters of the model are the following



Screenshot 37: Parameters of the HMM for Problem 7.

b) This can be solved using the formula in the answer to question 12 of the Section I (FAQ) of this manual. As the probabilities of the transition matrix are uniform, the problem is reduced in practice to independent sampling of the boxes, only needing to multiply the same transition 0.33 by the emission of the corresponding box. We will leave this to the student (The calculation can also be found in https://www.cs.swarthmore.edu/~soni/cs68/s17/Labs/hmm_solution.txt). The result is,

$$P(\pi = 1,1,3,2\ and\ x = A,A,O,O,A) = 0.000434$$

Where π, is the sequence of hidden states (Boxes), and x is the sequence of observed Apples (A) and Oranges (O).

Note that this probability is different and lower from the result of Forward, below, which is the probability of the observed sequence, x, regardless the box where the apples and oranges came from.

Termination

$$P(O) = \sum_{i=1}^{N} \alpha(L, i)$$

P(O: AAOOA) = α(5,1) + α(5,2) + α(5,3)

P(O: AAOOA) = (0.010431419916454222) + (0.01561587598893619) + (0.0052026891317264535) = 0.031249985037116866

Matrix of α values

	A_1	A_2	O_3	O_4	A_5
BOX1	0.167000	0.083375	0.041698	0.020865	0.010431
BOX2	0.249750	0.125062	0.020849	0.010411	0.015616
BOX3	0.083250	0.041625	0.062453	0.031219	0.005203

Screenshot 38: Forward result for Problem 7.

The reason why this probability is lower than the probability of the Forward algorithm is because there are many more combinations of boxes that can render the same observed sequence in the case of Forward, summing up many more probabilities.

c) The solution for this question is Viterbi:

Matrix of δ values

	A_1	A_2	O_3	O_4	A_5
BOX1	0.167000	0.041583	0.010417	0.002609	0.000652
BOX2	0.249750	0.062562	0.005224	0.001301	0.000975
BOX3	0.083250	0.020792	0.015625	0.003902	0.000325

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 most probable sequence of hidden states is BOX2 BOX2 BOX3 BOX3 BOX2 with probability of 0.0009746074296806664

Screenshot 39: Solution with the Viterbi algorithm for the Problem 7.

d) With all the values calculated so far, we can calculate what this question is asking:

$$\log\left[\frac{P(\pi * |x)}{P(\pi_b|x)}\right] = \log\left[\frac{\frac{P(\pi = 1,1,3,2\ and\ x = A,A,O,O,A)}{P(x = A,A,O,O,A)}}{0.0009746074296806664}\right]$$

$$= \log\left[\frac{\frac{0.000434}{0.031249985037116866}}{0.0009746074296806664}\right] = 1.15381019798264\ db$$

This result, in decibels (db) means that the probability of the most probable path of boxes (solved in part c) is $10^{1.15}$ = 14.24 times larger than the probability of the path of part b.

P8A) The nucleotide composition of a region of a genome, might have a meaning in biology. For example, high C+G content is typical in promoter regions of genes. This is because promoter regions are protected in the genome by DNA methylation. A Hidden Markov Model can be built and used to find regions with high C+G composition. The hidden states will represent different types of nucleotide composition. Consider two hidden states, *H* and *L* for high and low *C* + *G* content, respectively. The initial probabilities for both *H* and *L* are 0.5, while the transition probabilities between these two hidden states are the following: H->H = 0.5; H->L = 0.5; L->L = 0.6; L->H = 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 = 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. Therefore, the observed states will be A, T, C and G.

- Observed nucleotides: 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 initial probabilities.

The initial probabilities correspond to the probability that a hidden state has emitted the first observed state of the sequence of observed states. For this 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

Identify the transition probability matrix.

The transition probabilities matrix corresponds to the probability of changing form one hidden state to another. In this 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 is represented in the following way.

	H	L
H	0.5	0.5
L	0.4	0.6

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

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 interactions occurs. According to the transition and emission matrix, the following graph model of the problem is:

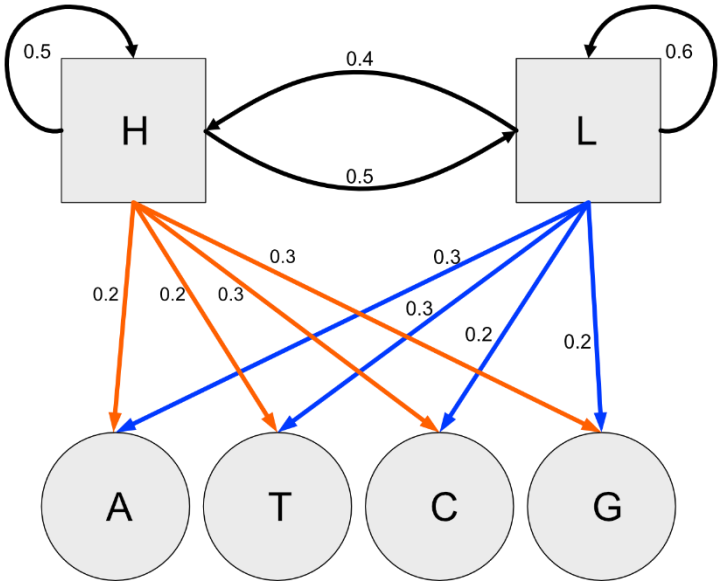
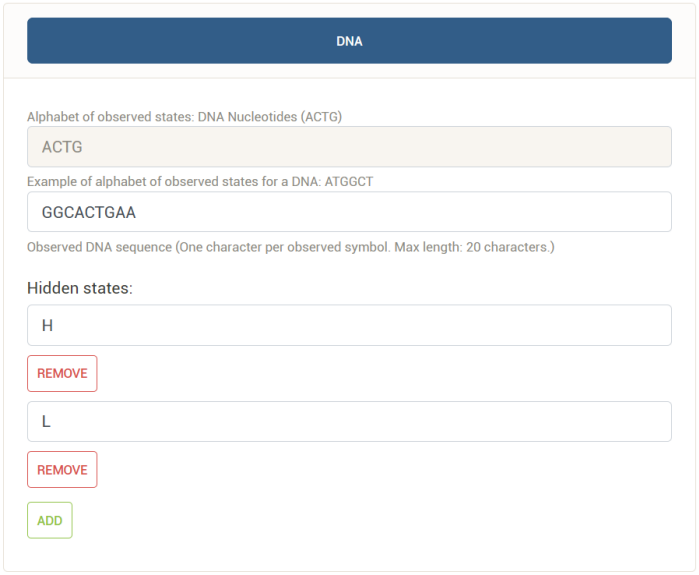


Figure 10: HMM graph for Problem 8

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

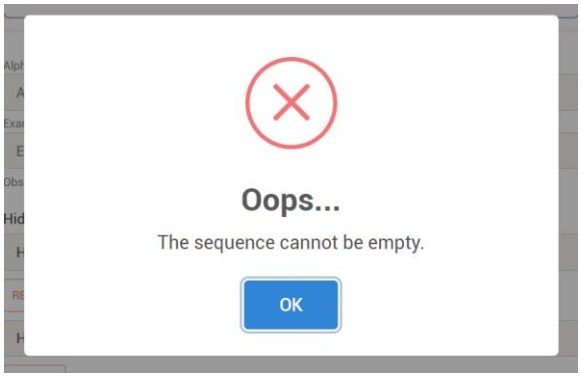
Once all the data have been identified and having established all the transitions that can be made, we proceed to enter this information in the software.

The main page of the tool shows the following options.



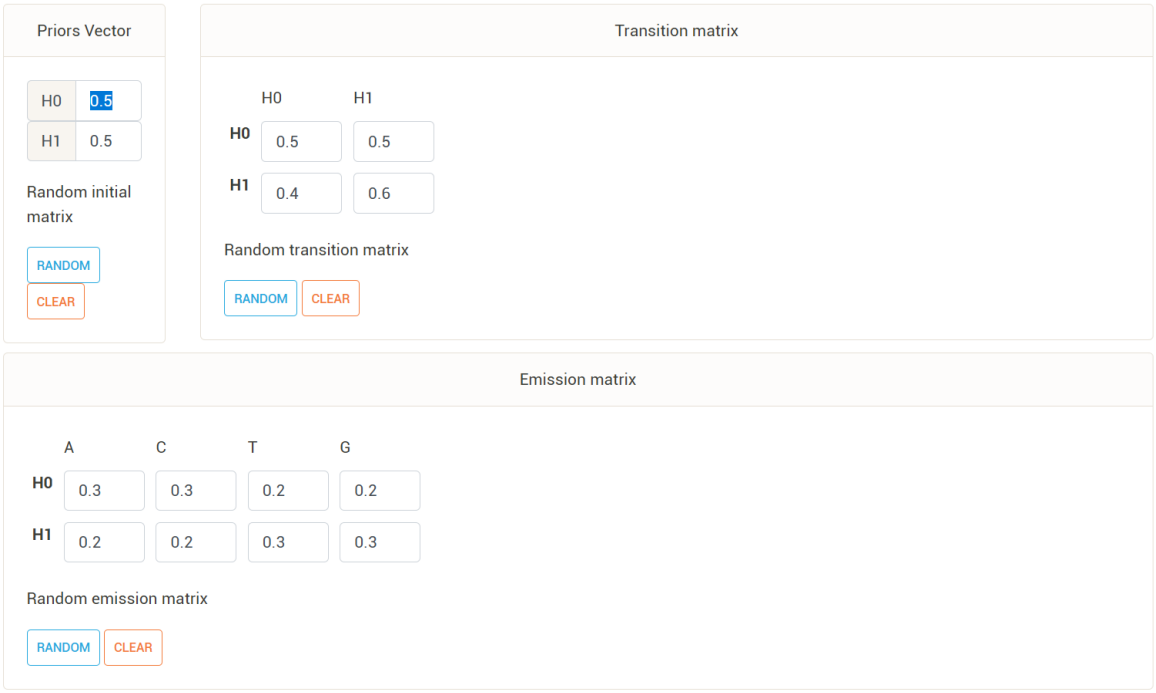
Screenshot 40: Step 1 - DNA section.

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.



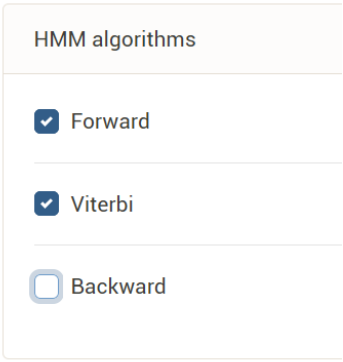
Screenshot 41: Error message for empty sequence.

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



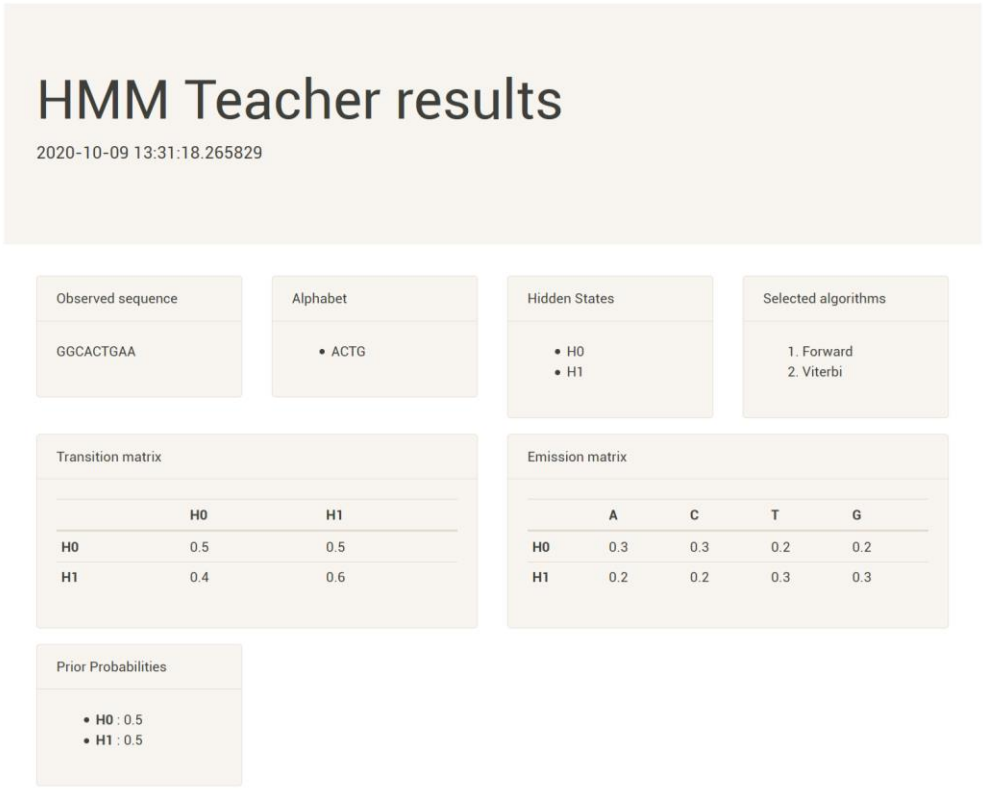
Screenshot 42: Step 2 - Filling the corresponding probabilities.

It's possible to insert any number, although the sum of these numbers cannot result different than 1. HMMTeacher allows the insertion of random values, with the button "Random", this in case that the probabilities for the problem are not given. After writing all the values, it is possible to move to the next step, the tool will perform validations when moving to the next step.



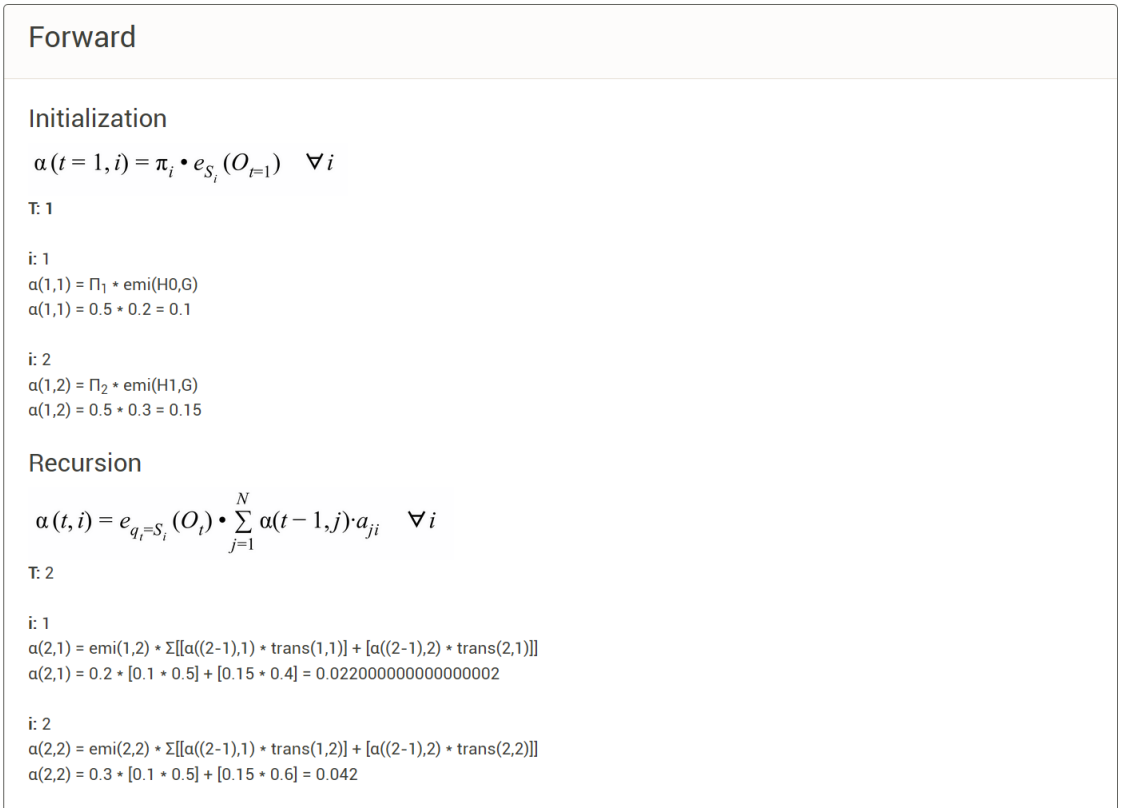
Screenshot 43: Step 3 - Selecting the HMM algorithm to answer the questions.

In this step it is 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 is important to mention that at least one algorithm needs to be chosen, and more than one can be choosing.



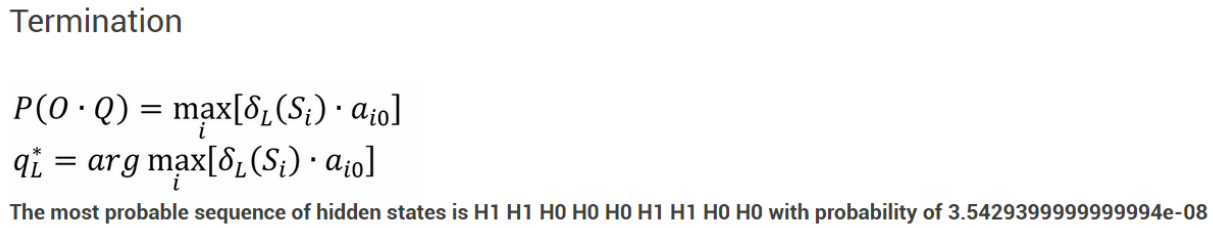
Screenshot 44: Step 4 - Results of the calculations.

In the final step, it is possible to send the data to the server to be analyzed or review one last time (going back to previous steps). After the data is sent, a new window will appear with the results, as shown in the image.



Screenshot 45: Sample of the results given. Forward algorithm and the calculations done.

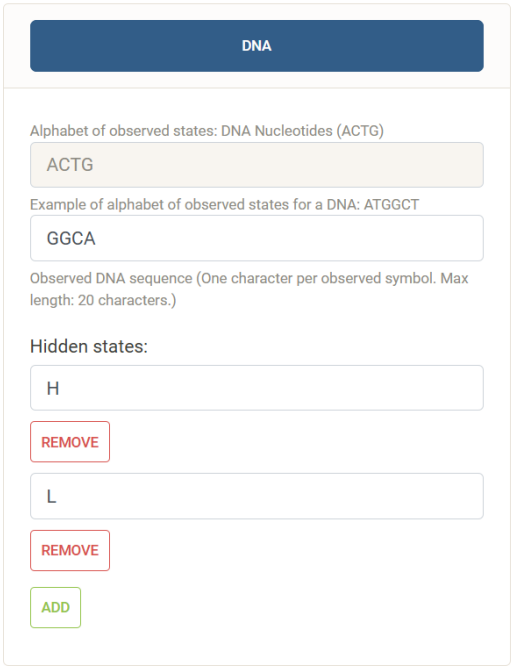
In the results it is possible to review the input data, the algorithms and the resolution for each one of them (if more than one was chosen). In the final section of the algorithm we can see the result of the most likely sequence that we wanted to discover.



Screenshot 46: Most likely sequence according to Viterbi algorithm.

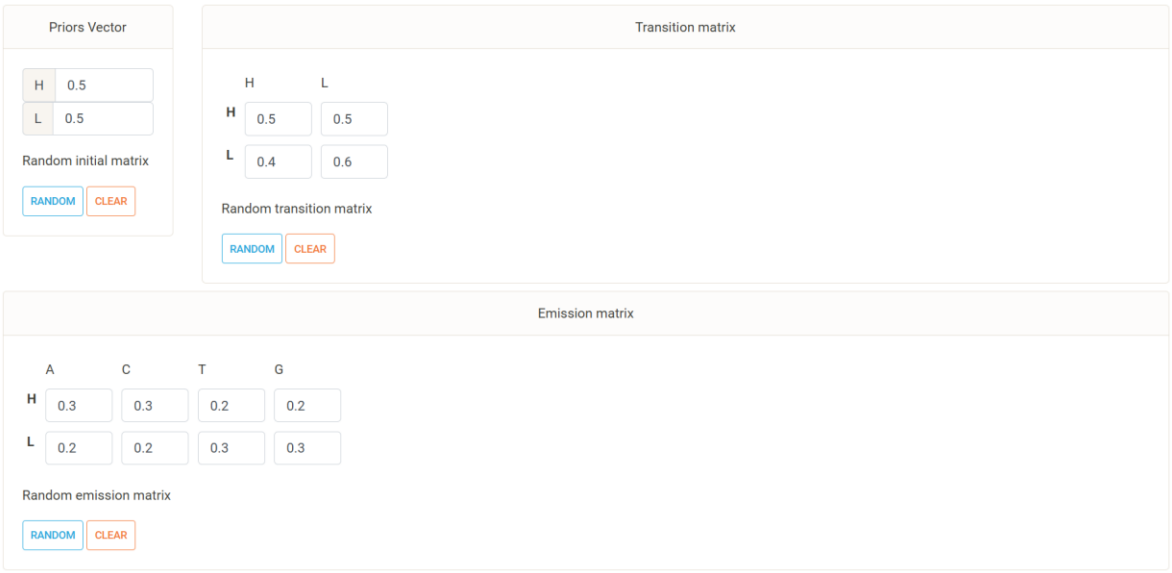
P8B) From the Hidden Markov Model used in the previous problem, and the DNA sequence fragment X: GGCA, find the probability that this will occur using the Forward algorithm. The HMM generated is shown in the Figure 2.

Solution:
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 is necessary to start from the step 1 and enter the sequence of observed and hidden states. As the observed sequence change to GGCA, it is necessary to use the custom option to change it.



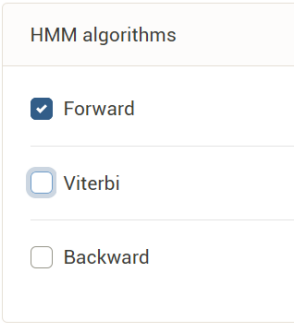
Screenshot 47: Step 1 - Custom options

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



Screenshot 48: The probabilities remain the same.

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



Screenshot 49: Step 3 - Choosing Forward algorithm

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

P(O: GGCA) = α(4,1) + α(4,2)
P(O: GGCA) = (0.0021198000000000002) + (0.0017028000000000002) = 0.0038226000000000007

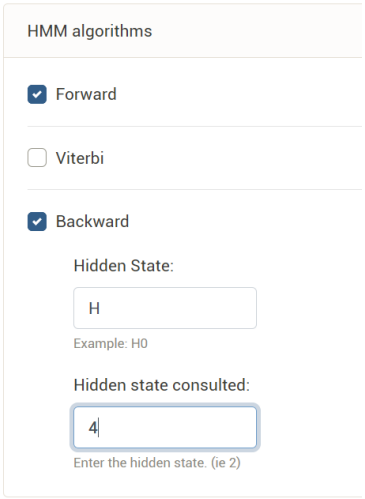
Matrix of α values

	G_1	G_2	C_3	A_4
H	0.100000	0.022000	0.008340	0.002120
L	0.150000	0.042000	0.007240	0.001703

Screenshot 50: Forward algorithm result.

P9) Find the *"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.

Solution:
The Backward algorithm must be used to calculate the *"posteriori"* probability of both hidden states in a specific position of the sequence GGCA. The input data is the same of 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.



Screenshot 51: Step 3 - Backward algorithm options.

In the last step it is 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 is known that both values must add up to 100%.

P10) Imagine that you work at ACME Chocolate Factory, confectioners extraordinaire. Your job is to keep an eye on the conveyor belt, watching the chocolates as they come out of the press one at a time.
Suppose that ACME makes two types of chocolates: ones with almonds and ones without. For the first few problems, assume you can tell with 100% accuracy what the chocolate contains. In the control room, there is a lever that switches the almond control on and off. When the conveyor is turned on at the beginning of the day, there is a 50% chance that the almond lever is on, and a 50% chance that it is off. As soon as the conveyor belt is turned on, it starts making a piece of candy.
Unfortunately, someone has let a monkey loose in the control room, and it has locked the door and started the conveyor belt. The lever cannot be moved while a piece of candy is being made. Between pieces, however, there is a 30% chance that the monkey switches the lever to the other position (i.e., turns almonds on if it was off, or off if it was on).

a) Draw a Markov Model that represents the situation and give the prior distribution on the states as well as the transition matrix.

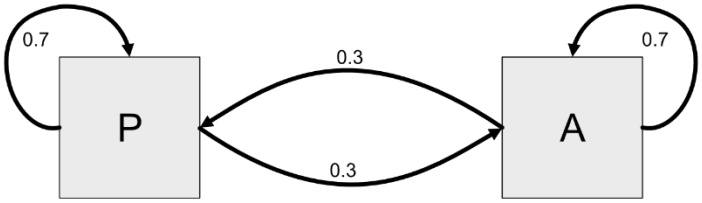


Figure 3: (P) Plain chocolates and (A) almond chocolates.

P	A
0.5	0.5

Table 2: Prior probabilities of P10 part a).

	A	P
A	0.7	0.3
P	0.3	0.7

Table 3: Transition probabilities of P10 part a).

Now assume that there is a coconut lever as well, so that there are four types of candy: Plain, Almond, Coconut, and Almond + Coconut. Again, there is to 50% chance of the lever being on at the beginning of the day, and the chance of the monkey switching the state of the second lever between candies is also 30%. Assume that the switching of the levers is independent of each other.

b) Draw a model for production of all four types of chocolate and give the prior distribution on the states as well as the transition matrix.

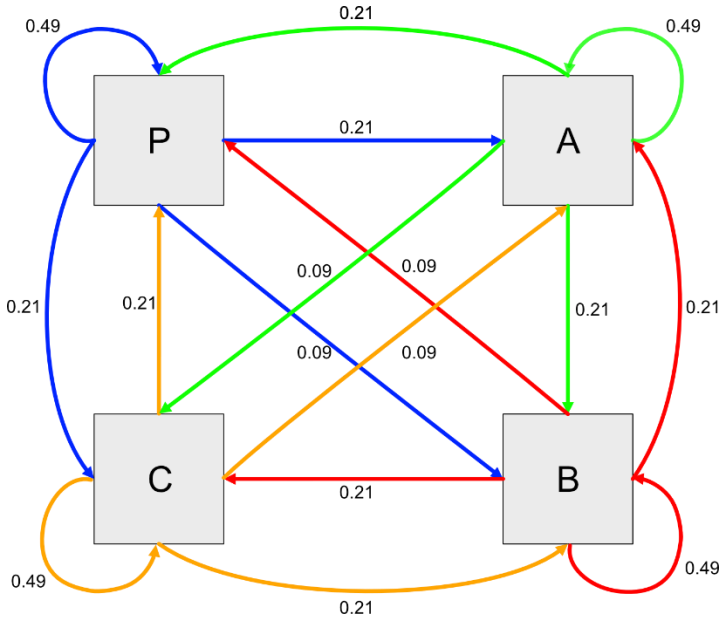


Figure 4: Model with 4 states. (P) Plain (Blue), (A), Almond (Green), (C): Coconut (Orange), (B): Both (A) and (C) (Red)

P	A	C	B
0.25	0.25	0.25	0.25

Table 4: The prior probabilities of problem P10 part b)

The probabilities are calculated with the given rules.

1. Both levers being flipped: $0.3 * 0.3 = 0.09$
2. One being flipped: $0.7 * 0.3 = 0.21$
3. Neither being flipped: $0.70.* 0.7 = 0.49$

	P	A	C	B
P	0.49	0.21	0.21	0.09
A	0.21	0.49	0.09	0.21
C	0.21	0.09	0.49	0.21
B	0.09	0.21	0.21	0.49

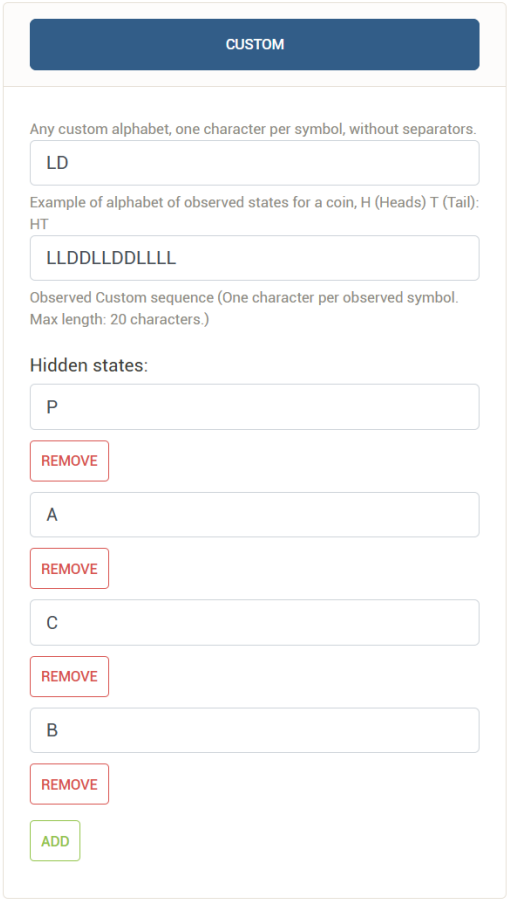
Table 5: The transition probabilities of problem P10 part b)

Assuming we can't tell what's inside the chocolate candy, only that the chocolate is light or dark, we are going to use the next emission table.

Inside	Light (L)	Dark (D)
Plain	0.1	0.9
Almond	0.3	0.7
Coconut	0.8	0.2
Both (A+C)	0.9	0.1

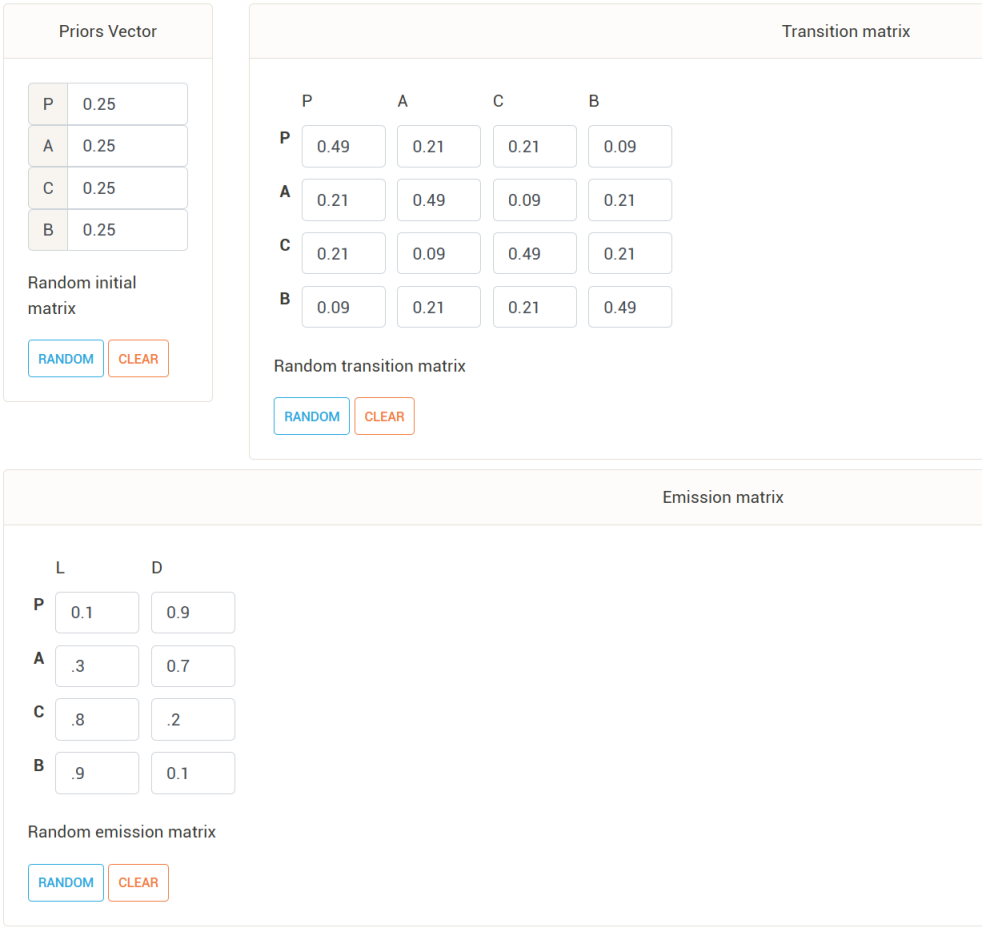
c) We want to compute how likely would be that we observed the sequence: LLDDLLDDLLLL

As first step we need to input the data in the HMMTeacher application.



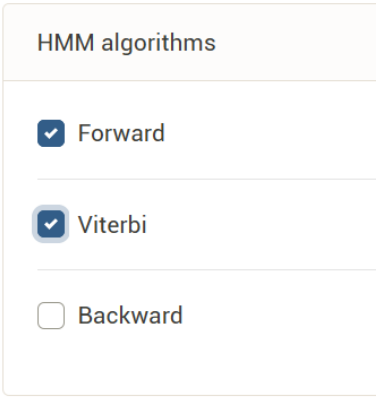
Screenshot 52: Modelling Problem 10.

In the second step it is necessary to input the gathered data.



Screenshot 53: Required data for the tenth problem.

In the third step it is necessary to select the desired algorithm.



Screenshot 54: Selected algorithms.

To finally obtain the results.

Termination

$$P(O) = \sum_{i=1}^{N} \alpha(L,i)$$

P(O: LLDDLLDDLLLL) = α(12,1) + α(12,2) + α(12,3) + α(12,4)
P(O: LLDDLLDDLLLL) = (7.796980311195184e-06) + (2.7812585931414052e-05) + (0.00012098909887063011) + (0.00015098996407774727) = 0.0003075886291909866

Screenshot 55: Forward algorithm result.

The likelihood to observe the sequence "LLDDLLDDLLLL" is 0.0003. This, using the Forward algorithm.

d) What's the most probable sequence of hidden states that would generate: LLDDLLDDLLLL

As in both cases we are making questions about the same sequence. The most probable sequence of hidden states that would generate "LLDDLLDDLLLL" is:

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 most probable sequence of hidden states is C C P P C C P P C C C C with probability of 3.629774802568882e-07

Screenshot 116: Result from Viterbi algorithm.

This, using the Viterbi algorithm.

One final point about the interpretation of probabilities: they only mean anything when compared to other probabilities. Therefore, the fact that the most probable sequence of hidden states has probability equal to $3.6\ x\ 10^{-7}$, only hints us about the large number of combinations existing for the hidden states P, A, C and B. In fact, there are $4^{12} = 16.777.216$ possible combinations. If the probability distribution of the combinations were uniform, this would be $\frac{1}{16.777.216} = 5.9x10^{-8}$. Therefore, the most probable combination found by Viterbi, is about 10 times more probable than this.