CS 640: Hidden Markov Models

Generalized Hidden Markov Models are defined as follows:

1. **N hidden states**: $S_1,,\ldots,S_N$
2. **M symbols in emitted alphabet**
3. Initial probability distribution vector of length N: $\boldsymbol{\pi} = \{\pi_1 \ldots, \pi_N\}$
4. Transition probability matrix of size N x N
   where $\boldsymbol{\tau}_{ij}$ is probability of transition from state in row i to state in row j
5. Emission probability matrix of size N x M
   where $\mathbf{e}_i(c)$ probability that state i emits character c

We refer to the transition probabilities, the emission probabilities and the initial distribution vector, collectively as the parameters of the model, designated $\boldsymbol{\lambda = (\tau_{ij} , e_i(c), \pi)}$.

Let Q be the sequence of visited states: $Q = (q_1, q_2, \ldots, q_F)$
Let O be the sequence of emitted symbols: $O = (O_1,O_2, \ldots,O_T)$ (the observed sequence).

Write a generalized Hidden Markov Model that employs the Viterbi algorithm (which is a dynamic programming algorithm) to find most likely sequence of hidden states to emit an observed sequence. You may hard-code in a transition matrix, emissions matrix and start probabilities. ***Your program should read a string of any length in FASTA format from a file and output the score of the nucleotide sequence of that string, given the HMM defined below*. The string may contain whitespace and numbers, which should be ignored, as well as *nucleotide characters {A, C, T, G}***

Code the Viterbi algorithm for this HMM, filling in matrix cells $\boldsymbol{\alpha_t(i)}$, where t corresponds to sequence index and i corresponds to state:
1. N = 3, hidden states S_1, S_2, S_3
2. M=4 symbols in alphabet {a, c, t, g}
3. Initial probability distribution vector $\boldsymbol{\pi}$ = {.25, .5, .25}

4. Transition probability matrix $\boldsymbol{\tau}$ =

	S1	S2	S3
S1	.5	.4	.1
S2	0	.5	.5
S3	.3	.2	.5

5. Emission probabilities **e** =

	a	c	t	g
S1	.4	.3	.2	.1
S2	.25	.25	.25	.25
S3	.1	.2	.3	.4

Initialization: $\alpha_1(i) = \pi_i e_i(O_1)$

Iteration: $\alpha_{t+1}(i) = e_i\,(O_{t+1}) \max_{j \in \text{states}} (\alpha_t(j) * \tau_{ji})$

Sean Eddy generalized HMMS:http://www.nature.com/nbt/journal/v22/n10/full/nbt1004-1315.html