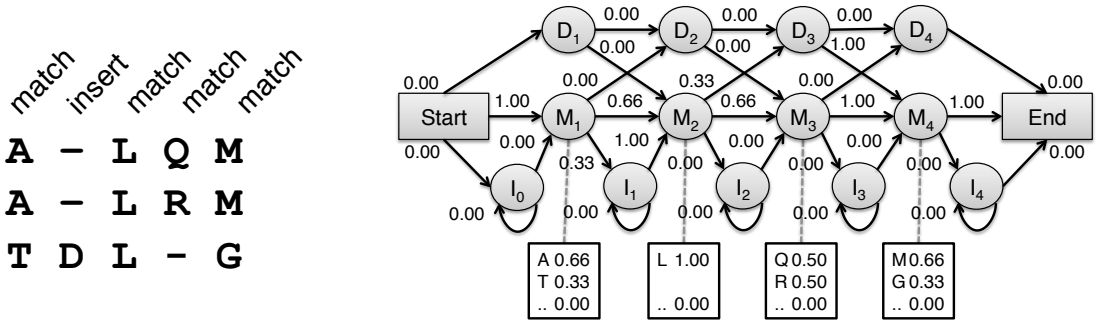
Assignment "Hidden Markov Models"

In this assignment you will implement a hidden Markov model, in particular functionality to set its parameters ("train" it) based on a multiple sequence alignment. The algorithm is described in 5.3 of "Biological Sequence Analysis" and on pages 185-187 of "Understanding Bioinformatics".



An example HMM (right) trained on an alignment of three sequences (left). This model structure is the same as in Figure 6.7A from "Understanding Bioinformatics"

Use the code skeleton in **`hmm_skeleton.py`** to:

- Read in a number of aligned protein sequences from a FASTA file (reusing code from the previous assignment).
- Implement a function to estimate the number of match states, given that we consider a sequence position to correspond a match state if at least half of the aligned sequences contain an amino acid at that position (see the figure above).
- Implement a function to count the number of transitions between states and the emissions in the match states that occur in the sequences.
- Implement a function to normalize these counts to transition probabilities (next to the arrows in the figure above) and emission probabilities (in the boxes), using formulas on "Biological Sequence Analysis" p.107 (or EQ6.24 and EQ6.25 in "Understanding Bioinformatics").
- Implement a function to sample a sequence from the HMM, i.e. generate a sequence by choosing transitions and emissions using a random number generator. A support function **`sample`**, to pick a key from a dictionary containing probability values, is supplied in the skeleton file.

Questions:

1. Read in the supplied small file **`test.fasta`** (example from "Biological Sequence Analysis"). What is the number of match states needed?

2. Train the HMM, i.e. the emission and transition probabilities. For emissions in the insert states, use the values supplied in **`pa`**. Print the estimated probabilities. Explain whether they make sense.

3. Print the emission probability matrix in a format demanded by the SeqLogo server (http://www.cbs.dtu.dk/biotools/Seq2Logo-2.0/), e.g. the PSSM

format (http://www.cbs.dtu.dk/biotools/Seq2Logo-2.0/bin/weight.txt), and use SeqLogo to generate a logo. Put both the matrix and logo in your answers.

4. List at least 10 random sequences generated using your HMM.

5. Change from using the estimated emission probabilities (EQ6.25) in your code to the background frequencies (EQ6.26) and repeat steps 2-4. How do the results change?

6. Read in the supplied large file **`test_large.fasta`** and repeat steps 2-5. Report the emission and transition probabilities, the logo (from SeqLogo), and the sampled sequences.

7. Pick a randomly generated sequence and search for it in the PFAM database (http://pfam.xfam.org/). To what family does the sequence belong?

8. What is the time complexity of estimating the emission and transition probabilities?

9. Which part of your code is deterministic and which part is randomized?

Optional (highly):

Can you implement the Viterbi algorithm (dynamic programming, page 188-190 of “Understanding Bioinformatics”) to calculate the probability of a given sequence according to the HMM?

Please, email your Python script and a PDF file containing the answers to the questions to algorithms@bioinformatics.nl no later than 14.00 on the day of the lecture.