Prof. Dr. Daniel Huson
Zentrum für Bioinformatik
Fachbereich Informatik
Mathematisch-Naturwissenschaftliche Fakultät

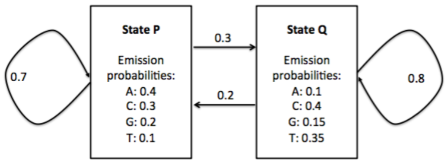
EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Grundlagen der Bioinformatik SoSe 2018

Assignment 05 **Submit electronically in Ilias by 28.5.2018, 10h**

1 HMM and Viterbi algorithm (2 points)

Illustrate how the Viterbi algorithm computes the most probable path for the sequence `GATA`, using the following HMM (assume that the probability of starting in either state is 50%):



2 Implementation of HMM generator (3 points)

Download the set of files contained in `Download-05.zip`. The Java file `HMM.java` contains code for reading and writing an HMM in the adhoc file format discussed in the lecture. The file `casino.hmm` contains a definition of the *unfair casino* HMM and can be read by the code in `HMM.java`.

Your task is to write a Java program called *Generator.java* that reads the HMM description file as input and generates a sequence of state values (in this case die throws) as output, e.g. `6665342151662542313142664536251425366352....`

To do this, implement the method called `generateSequence()` in your class `Generator.java`. Then, in `Generator.java`, construct an HMM object, read in the `casino.hmm` file and then output a simulated sequence of symbols obtained using your method `generateSequence()`. Your class should store the sequences of states that was actually used to to generate the sequence of symbols and make it available by a method called `getStates()`, as this will be needed in Problem 4 below.

3 Implementation of HMM Viterbi decoder (3 points)

Complete the implementation of the Viterbi algorithm in the method `runViterbi(String seq)` in `HMM.java`.

Then write a program called `Decoder.java` that reads in an HMM and a sequence as input and outputs the *decoding* as output.

For example, if the input of die rolls is `666534215...` then the output of the program should be two lines that look something like this:

```
666534215...
UUUUFFFFF...
```

4 Evaluation of the Viterbi decoder (2 points)

Write a Java program `Evaluator.java` that reads an hmm as input. It then uses your generator code to generate 100 sequences. For each sequence, run your Viterbi code on the sequence. Compare the string of states generated by the Viterbi with the string of states used in the generation of the corresponding sequence.

Report how many times the Viterbi algorithm decodes the correct state, or incorrect state, for the two states `U` and `F`.