# Data Mining and Machine Learning in Bioinformatics

Dr. Holger Fröhlich, Dr. Martin Vogt
Sabyasachi Patjoshi
sabyasachi2k13@gmail.com, martin.vogt@bit.uni-bonn.de

**Due: Jul 1, 10:30 (by the end of the lecture)**

## Exercise Series 10

**General: Exercises are to be solved and submitted in fixed groups by at most 3 students. Every member of a group should help solving each task and thus be able to answer questions to each task. No late submissions are accepted. Copying solutions will automatically lead to a point reduction of at least 50%. N – 1 homework assignments and N – 2 programming tasks have to be submitted in total.**
**A group can gain additional bonus points, if it presents its solution for a particular task during the tutorials. Accordingly, each task has a defined number of points as well as bonus points.**

1. The following multiple sequence alignment is given:

```
GC-CTC
GCT-TC
-C-C-A
G---TC
AC-CT-
```

   a) Derive the consensus sequence and define match states for a profile HMM accordingly, only columns with >50% nucleotides are modeled. (3 points + 1 bonus point)
   b) Estimate the transition and emission probabilities with a pseudo-count of 1. **Tip**: You can write a small program to do the job for you. (4 points + 1 bonus point)

2. Install CRAN package HMM for handling **Hidden Markov Models**.
   a) A dishonest casino sometimes manipulates the outcome of dice rolls by using two different dice: a fair die where all numbers from 1 to 6 have equal probability and a loaded die where the numbers 1 to 5 have equal probability and 6 has a higher probability p=0.5. Most of the time the fair die is used but with probability $p_{fair->loaded}$=0.05 the dice are switched and the loaded die is used. While the loaded die is used there is some probability $p_{loaded->fair}$=0.1 to switch back to the fair die. Assume identical starting probabilities for the fair and loaded die.
   **Generate a HMM** modeling the dishonest casino's behavior and generate a sequence of 2000 observations. (4 points + 1 bonus point)

b) Now we will try to learn the HMM of A from the observed sequence. To this end, use `initHMM` to set up an initial HMM with two states, where each state has the same starting probability and six possible outcomes 1 to 6 for each state. Use i) the **Baum-Welch** algorithm and ii) the **Viterbi training** algorithm to estimate transition and emission probabilities. Use a pseudocount of 1 in each case. (Hint: How should you choose the initial emission/transition probabilities before training?) What do you observe? (4 points + 1 point)

c) What are the **probabilities to observe the sequence** of outcomes from a) using i) the Baum-Welch model ii) the Viterbi-Training model and iii) the true HMM model from a). (3 points + 1 point)

d) For the Baum-Welch model compute the **posterior state probabilities** and the **most probable hidden state sequence**. Are there differences between the sequence of position-wise most probable states and the most probable hidden state sequence? Discuss possible reasons. (4 points + 1 bonus point)

e) Finally, change the initial model by setting $p_{fair->loaded}=0.2$ and $p_{loaded->fair}=0.2$ and repeat the Baum-Welch training using a simulated sequence of 2000 observations. Is the trained model still able to distinguish the states “fair” and “loaded”? (2 points + 1 point)