

COMP3212(2023/24): Computational Biology Lab Two

Issue	12 February 2024
Deadline	23 February 2024

Objective

To fully understand and implement sequence matching and generating algorithms

Part 1

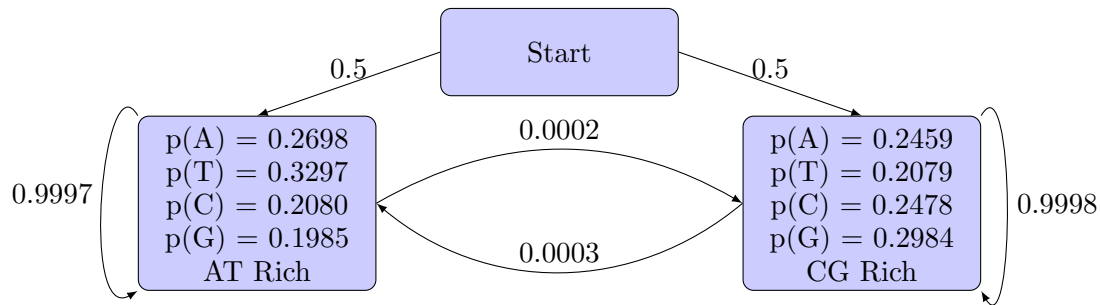
- Write a program to implement Needleman-Wunsch for proteins
- Check that it is working using HEAGAWGHEE versus PAWHEAE
- Compare to slide 60 in Lecture 09 - Sequence matching.
- Match the protein sequences using your implementation
SALPQPTTPVSSFTSGSMLGRDTDALTNTYSAL with PSPTMEAVTSVEASTASHPHSTSSYFATTYYHLY

Part 2

- Write a program to implement Smith-Waterman for proteins
- Check that it is working using HEAGAWGHEE versus PAWHEAE
- Compare to slide 34 in Lecture 10 - Local Matching
- Using your implementation find the best local match between
MQNSHSGVNLGGVFNVRPLPDSTRQKIVELAHSGARPCDISRILQVSNCGVSKILGRY
and TDDECHSGVNLGGVFNVRPLPDSTRQKIVELAHSGARPCDISRI

Part 3

- Implement an HMM to mimic the dishonest casino in the slides (Lecture 11 HMM pg 7).
- Produce a human readable output with both the sequence of numbers and the sequence of states.
- Compare your output to that generated on page 36.
- Implement an HMM to generate DNA bases along the model below. Use it to generate a long (ca 2000) base sequence
- Find some way to display this series



Part 4

- Implement the Viterbi algorithm for finding the most likely state sequence in an HMM.
- Test it on the output from your dishonest casino and your DNA sequence generator, how does it compare?
- Run this on the genome for the lambda phage, a virus that infects *E. coli* bacteria
- Can you calculate the posterior probability of the path?

Report and marking

You can ask a demonstrator or myself to mark your work in the lab, or you can submit a pdf output from your work.