CS 369 Assignment 4 2017

Due Monday May 22 6:00 pm

Write your submission in a Jupyter notebook and write any code in Python 3. You should submit the .ipynb file and a .html file with all output displayed. The primary document the markers will look at is the .html file.

In your report, include explanations of what you are doing and comments in the code. Where you are making mathematical derivations, show your working.

Submit your notebook as an .ipynb file and as an .html file with all output showing to `https://adb.auckland.ac.nz/` by 6:00 pm on the due date.

1. *[11 marks total]* Implementing and applying the overlap alignment algorithm. You will need the BLOSUM62 matrix and a method to read it in from the resources page of the course website.

   (a) State which lines need to change in the pseudocode (taken from Wikipedia) at the end of this assignment for the global alignment algorithm and make the necessary changes to turn it into pseudo-code for the overlap alignment algorithm discussed in lectures.

   (b) The pseudo-code produces a single alignment with the optimal score. Is this alignment the same regardless of the order in which the two sequences are passed as arguments to the method? Explain why or why not.

   (c) Implement the overlap alignment algorithm based on the pseudo-code you produced in part a. As input, it should take two sequences, a score matrix and a linear gap penalty. It is sufficient that your algorithm produces a single alignment with the optimal score.

   (d) Uniprot is an online protein sequence database. Find the database and download the protein sequences with accession numbers H0Y8P2 and N1J540. The first 10 residues are `XILESPEHLP` and `MHPAQLSKLL`, respectively. Print the full sequences.

   (e) Both of these sequences contain a zinc-finger, which is a small protein region that binds to zinc (see `https://en.wikipedia.org/wiki/Zinc_finger` if you are interested). There are multiple classes of zinc finger. A protein sequence from the Cys2His2 class is: `YECENCAKVFTDPSNLQ`

   Using your overlap alignment algorithm with the given BLOSUM62 matrix and a gap penalty of 8, find and print an optimal alignment between H0Y8P2 and the given zinc-finger sequence and N1J540 and the given zinc-finger sequence.

   (f) Use the score of each alignment to decide which sequence has a Cys2His2 zinc-finger and which does not. Which residues in the nominated sequence contain the zinc-finger?

2. *[13 marks total]* Suppose we wish to estimate basic secondary structure in protein (amino acid) sequences. The model we consider is a simplistic rendition of the model discussed in S C. Schmidler et al. (2004) Bayesian Segmentation of Protein Secondary Structure, doi:10.1089/10665270050081496

   We assume that at each point of the sequence, the residue is associated with one of three secondary structures: α-helix, β-strand and loops which we label H, S and T, respectively. To simplify the problem, we classify the amino acids as either hydrophobic, hydrophilic or neutral (B, I or N, respectively) so a sequence can be represented by this 3-letter alphabet.

   In a α-helix, the residues are 35% hydrophobic, 55% hydrophilic and 10% neutral. In a β-strand, the respective proportions are 55%, 15%, 30% and in a loop they are 10%, 10%, 80%.

   Assume that all secondary structures have geometrically distributed length with α-helices having mean 20 residues, β-strands having a mean of 12 residues and loops a mean of 10 residues. A β-strand is followed by an α-helix 40% of the time and a loop 60% of the time. An α-helix is followed by a β-strand 20% of the time and a loop 80% of the time and a loop is equally likely to be followed by a strand or a helix. At the start of a sequence, any structure is equally likely.

   (a) Derive the transition probabilities of a state to itself (e.g., a_{HH}) by considering that if L is geometrically distributed with parameter p then $E[L] = 1/p$. Make sure you use the parametrisation of the geometric distribution that takes values in $\{1, 2, \ldots\}$ and remember that $\sum_l a_{kl} = 1$ for any state k.

   (b) Sketch a diagram of the HMM (a hand-drawn and scanned picture is fine). In your diagram, show only state nodes and transitions. Show the emission probabilities using a separate table.

   (c) Write a method to simulate state and symbol sequences of arbitrary length from the HMM. Your method should take sequence length, a and e as arguments. Simulate and print out a state and symbol sequence of length 150.

   (d) Write a method to calculate the natural logarithm of the joint probability $P(x, \pi)$. Your method should take x, π, a and e as arguments.

   Use your method to calculate $P(x, \pi)$ for π and x given below and for the sequences you simulated in Q2c.

   $\pi =$ `S,S,S,S,T,T,S,S,S,S,S,S,S,H,H,H,H,H,H,H,H,H,H,H`

   $x =$ `I,N,I,B,N,B,N,B,B,B,N,N,B,B,B,B,I,I,I,I,I,B,N,B,I`

   (e) Implement the forward algorithm for HMMs to calculate the natural logarithm of the probability $P(x)$. Your method should take x , a and e as arguments.

   Use your method to calculate $\log(P(x))$ for π and x given above and for the sequences you simulated in Q2c.

   How does $P(x)$ compare to $P(x, \pi)$ for the examples you calculated? Does this relationship hold in general? Explain your answer.

Pseudocode for global alignment

```
 % make the F matrix
 1 for i=0 to length(A)
 2   F(i,0) <- d*i
 3 for j=0 to length(B)
 4   F(0,j) <- d*j
 5 for i=1 to length(A)
 6   for j=1 to length(B)
 7   {
 8     Match <- F(i-1,j-1) + S(Ai, Bj)
 9     Delete <- F(i-1, j) + d
10     Insert <- F(i, j-1) + d
11     F(i,j) <- max(Match, Insert, Delete)
12   }

 % backtrack and form alignment
13 AlignmentA <- ""
14 AlignmentB <- ""
15 i <- length(A)
16 j <- length(B)
17 while (i > 0 or j > 0)
18 {
19   if (i > 0 and j > 0 and F(i,j) == F(i-1,j-1) + S(Ai, Bj))
20   {
21     AlignmentA <- Ai + AlignmentA
22     AlignmentB <- Bj + AlignmentB
23     i <- i - 1
24     j <- j - 1
25   }
26   else if (i > 0 and F(i,j) == F(i-1,j) + d)
27   {
28     AlignmentA <- Ai + AlignmentA
29     AlignmentB <- "-" + AlignmentB
30     i <- i - 1
31   }
32   else (j > 0 and F(i,j) == F(i,j-1) + d)
33  {
34     AlignmentA <- "-" + AlignmentA
35     AlignmentB <- Bj + AlignmentB
36     j <- j - 1
37   }
38 }
```