

Optional Python Review Exercise

October 20, 2015

This is an VOLUNTARY EXERCISE that won't be marked nor it is part of your course grades. The aim of this exercise is to help you keep fresh your python programming skills. I will present you my solution to this problem as an excuse to review python concepts in the last class of the course. Keep in mind that this problem might prove to be more challenging than the ones we have already discussed during the course. Do not despair, you can solve it using the tools we have learned during the course and note that it is far more complex than the one you will have to solve for your final exam.

The goal is to write a python3 script that implements the Needleman-Wunsch algorithm for sequence alignment. Although this may seem to be a daunting task, in fact it is quite easy if you have a “recipe” for this algorithm. First of all you'll need to review the concepts behind dynamic programming to solve sequence alignments, to that end take a look at the following resources:

1. Read the wikipedia entry for the Needleman-Wunsch algorithm at: http://en.wikipedia.org/wiki/Needleman%E2%80%93Wunsch_algorithm. In addition to a clear and detailed step-by-step explanation of the Needleman-Wunsch procedure, it also contains the *pseudocode* for the algorithm. A pseudo-code is a like a general template that contains the basic steps that any program needs to follow in order to produce the expected result (in this case the alignment of two sequences). The only thing you actually have to do is “substitute” the lines in this template with code for a specific language (in our case python3).
2. Also, there is a great interactive tool to see how dynamic programming for sequence alignment works: <http://ultrastudio.org/en/Needleman-Wunsch>
3. Finally, you can also take a look at the excellent (and short!) paper “What is dynamic programming?” by Sean R. Eddy (NATURE BIOTECHNOLOGY VOLUME 22 NUMBER 7 JULY 2004). This paper also provides an implementation of the Needleman-Wunsch algorithm in C, so you can use it as an additional source of inspiration. I uploaded both, the paper and C code in the course Moodle page.

You can implement the algorithm as simple or complex as you wish (e.g. only aligns nucleotides with a fixed substitution matrix or allows for any kind of

string and substitution matrix, provide fixed gap penalties or user-provided,...). In any case I would advise you to start writing the simplest possible code and then, if you wish, extent it. If you make this first version modular, you should be able to extend it quite easily.

We hope that, in addition to being useful for HPBBM, this assignment will help you gain a deeper understanding of sequence alignment and dynamic programming for your BIBMS course.

Happy coding!