

**Homework 3 for computer scientists**  
**1-INF-301: Methods in Bioinformatics**  
**Deadline: Wednesday December 17, 2025 22:00**

You can write your answers in English or Slovak.

**(1) RNA structure and dynamic programming.** In the lecture, we considered well-parenthesized RNA structures (without pseudoknots). In this task, we will consider an even more restricted set of secondary structures. Let  $(i, j)$  and  $(i', j')$  be two base pairs in an RNA structure such that  $i < j, i' < j'$ . We call them *nested* if  $i < i' < j' < j$  or  $i' < i < j < j'$ . The whole RNA structure is called *nested-only*, if all pairs of base pairs are nested.

Below we see two examples of well-parenthesized secondary structures written in the parenthesis notation. The first structure is nested-only whereas the second is not.

`.(((.)).). nested-only`  
`.((.)(.)). not nested-only`

- a) Design an algorithm which for a given RNA sequence  $x_1, \dots, x_n$  computes the highest possible number of base pairs in a nested-only secondary structure. Consider only canonical base pairs A with U and C with G. We recommend using dynamic programming. Write a recurrence and describe how it will be used to compute the answer. It is sufficient to compute the count of base pairs, you do not need to produce the structure itself. What is the asymptotic running time of your algorithm?

For example, for sequence ACCCAGAAGGACAG we get the structure `.(((.)).). . . .` which pairs the first three Cs with the first three G's in a nested fashion. When listed as a list of pairs, these would be 2-10, 3-9, 4-6 (in this example the positions in RNA are numbered starting from 1). Note that the last C-G pair 12-14 was not used in this structure because it would not be nested with the other pairs. Your algorithm should produce the number 3.

- b) Write the rules of a stochastic context-free grammar (without listing particular values of probabilities) that can produce all RNA nested-only structures and no other structures (two bases of a base pair are always produced in the same production rule).
- c) Real RNA structures are stabilized by so called stacked pairs that follow each other on both sides. We will say that base pair  $(i, j)$  is supported if base pair  $(i + 1, j - 1)$  is also part of the secondary structure. In the example below, we see a nested-only structure with 6 base pairs but only 4 of them are supported. The left parentheses of these supported pairs are highlighted by `~` below the structure.

ACGCACGCAAAGCGGCGA  
`.(((.(((. . .)))))).`  
`~ ~ ~ ~`

Modify your dynamic programming from part a) to find the nested-only structure with the highest possible number of supported base pairs (again it is sufficient to compute the number of such supported base pairs). We recommend using two matrices in your dynamic programming: one for the overall highest number of supported base pairs within some substring and one where we restrict only to solutions that have the first and the last base of the substring paired.

In parts a) and c) make sure that your algorithm correctly solves the required problem and try to make it efficient, preferably  $O(n^2)$ .

**(2) Bioinformatics tools and databases.** The goal of this task is to gain some experience with bioinformatics tools and databases.

Download a short DNA sequence from <https://compbio.fmph.uniba.sk/vyuka/mbi-data/a3.fasta>. This sequence comes from an unknown bacteria contaminating a sequencing sample (real story).

- a) Use BLAST to compare this sequence with known genomes to find out from what species it probably comes from. Go to <https://blast.ncbi.nlm.nih.gov>, select *Nucleotide BLAST*, enter the provided sequence as a query, and as the database, select *Refseq Reference Genomes*. In the box *Organism* enter *Gammaproteobacteria (taxid: 1236)*, so the program will search only one group of bacteria (in real use, we would not have this information, and thus we would have to look for it for example in all bacteria). Leave the other settings at default values and start the search.

List the details of the alignment with the highest score found by the program (% identity, E-value, source organism). What can you infer about this sequence based on the results?

- b) We would like to see if this DNA sequence encodes some proteins and what their function might be. Instead of an HMM for gene finding, we will simply consider open reading frames (ORF). An ORF is a sequence of codons that does not contain a stop codon and could therefore potentially encode some protein. We need to consider ORFs on both strands and also in all three reading frames, which are three possible shifts where codons could start within the sequence. In bacteria, genes usually do not have introns, and so the search for ORFs is a relatively successful and simple procedure for finding candidate genes.

Find the ORFs in our sequence using the tool [https://www.bioinformatics.org/sms2/orf\\_find.html](https://www.bioinformatics.org/sms2/orf_find.html). Look for ORFs with at least 180 codons. Check all three reading frames and both strands. Allow any start codon and use the standard genetic code.

You should obtain 4 ORFs, which we will name as orf1 to orf4. We will be only interested in orfs 1 and 3 with the following starts of the corresponding protein sequences: orf1 QPKA, orf3 TRVA.

List the protein sequences in your homework. You will also use them the subsequent subtasks. Most programs require protein sequences in the FASTA format, in which each sequence is preceded by a line starting with > sign followed by the name of the sequence, e.g. >orf1. Write the protein sequences in this format.

- c) Enter orf1 and orf3 into the tool for finding protein domains using profiles at <https://www.ncbi.nlm.nih.gov/Structure/cdd/wrpsb.cgi>. In the right panel, choose Pfam as a database. List found domains, their E-values and which part of the protein these domains occupy (columns Start and End). Based on the results, which of these two ORFs has a greater chance of being an evolutionary conserved protein?
- d) In this subtask, we determined the structure of orf1 and orf3 proteins using AlphaFold2. Please use the images produced by AlphaFold2 shown on the course website. In addition to the image of the structure, AlphaFold2 provides two plots representing the confidence in the structure, abbreviated pLDDT and PAE. Read about these indicators in FAQs, sections “How confident should I be in a prediction?” and “How should I interpret the relative positions of domains?”.

Based on this, describe what you can infer about reliability of the structure predictions for the two proteins. Are some parts more reliable than others? Which of the proteins has an overall more reliable prediction? Can you see the Pfam domains found in part c) on the PAE plots? Can you make any other observations?

Note: If you wish, you can run AlphaFold yourself, using the AlphaFold Colab, the ColabFold or the AlphaFold3 server. However, the results may differ; therefore in this task please base your answers on the images provided by us.