Prof. Dr. Oliver Kohlbacher
Prof. Dr. Manfred Claassen
Anke King, Rosanna Krebs,
Dr. Kyowon Jeong, Hadeer Elhabashy, Dr. Samuel Wein
Faculty of Science
Department of Computer Science and
Institute for Bioinformatics and Medical Informatics (IBMI)

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

1st Assignment - Structure and Systems Bioinformatics

Hand in: 2023-05-04 10:00 CEST (source code and pdf in a single archive, uploaded via ILIAS)

Task 1 – Nussinov RNA folding algorithm (60 points)

Design and implement the Nussinov RNA folding algorithm in **Python 3**. Your program has to read an RNA sequence from a `FASTA` file and predict *one* of its optimal secondary structures.

The filled dynamic programming (DP) matrix and the score of the optimal secondary structure, as well as the structure representation in both *bracket-dot* and *bpseq* formats have to be printed to the standard output. Print the DP matrix in an well-conceived way, meaning it contains all information needed to understand it, e.g. bases, and is easy to read.

Since different scoring functions and minimum loop lengths lead to different optimal structures, you should provide a command line option to set the minimum loop length and the scoring function (with reasonable default values).

The program has to have the following command syntax:

```
<last_name>_nussinov_predictor.py -i <PATH_TO_FASTA_FILE>
[--min-loop-length <MIN-LOOP-LENGTH>] [--score-GC <SCORE-GC>]
[--score-AU <SCORE-AU>] [--score-GU <SCORE-GU>]
```

Provide required packages in a `readme.txt` attached with your answer. Test your implementation with the `test.fasta` file attached with the assignment. You can validate your implementation using the provided `test.bpseq` files. They state the used parameter settings and one optimal RNA structure.

If your program works, points will be awarded for correctly filling the DP matrix (15), performing the traceback (15), correctly working minimum loop length setting (10), adaptable scoring function (10) and correctly formatted output (10). In case of incorrect behaviour, any partial points are at the tutors' discretion.

Task 2 – Multiple optimal solutions (20 points)

The Nussinov algorithm presented in the lecture and implemented in your program obtains a single well-defined solution, even if there are multiple optimal solutions. **Please provide short answers of around 1-2 sentences each.**

- Please explain why only one solution is returned with the implementation from the lecture.
- Please explain how you would modify your algorithm to obtain a random optimal solution (just written text is sufficient, pseudo-code is not required).

- Provide an estimate in O-Notation for the computational complexity of finding all optimal solutions. Explain why the method has this complexity.

Task 3 – k-loop decomposition (20 points)

For the RNA secondary structure below fill out the following table (one row per k-loop):

Base pair	Value of k	Size	Bases being accessible	Name of secondary structure
⋮	⋮	⋮	⋮	⋮

The columns are:

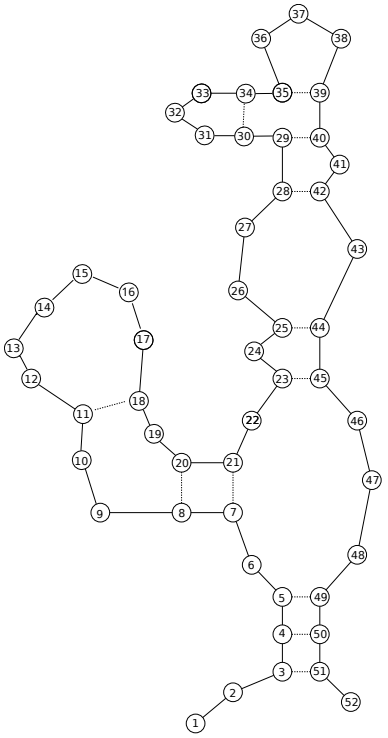
Base pair A tuple (n_1, n_2) which denotes the base pair in the figure of the k-loop if applicable.

Value of k .

Size Size of the k-loop as it was defined in the lecture.

Bases being accessible A list $n_1, n_2, \ldots$ of bases that are accessible in this k-loop. *accessible* was defined in the lecture.

Name of secondary structure Provide one of: hairpin, stacked pair, bulge loop, interior loop, multi-loop, dangling end.



Please use the numbers of the bases in the figure to refer to the bases. Dashed lines indicate base pairs.
Remember: The number of non-null k-loops of a structure equals the number of base pairs it contains.

Questions can be directed to ssbi-ss23@informatik.uni-tuebingen.de or the ILIAS course forum. We highly encourage you to use ILIAS for communication.