Re-exam Structural Bioinformatics 2022-2023

Jan Gorodkin, Stefan Seemann and Thomas Hamelryck

1 Protein part (2/3 of points)

Theoretical part (1/2 of protein points)

Based on the following **two articles** (one is a follow up of the other),

- End-to-end differentiable learning of protein structure, Cell, 2019
- Single-sequence protein structure prediction using a language model and deep learning, Nature Biotechnology, 2022

write a summary that

- Outlines **the ideas behind the Recurrent Geometric Network (RGN)**.
  - Provide some mathematical and algorithmic detail where appropriate.
- Outline the **differences and similarities** with the approach taken by AlphaFold2.
- Outline the **claimed advantages** of the RGN2 method (second article) with respect to AlphaFold2.

Maximum 2 pages, including tables, figures and formulas.
Add the summary to a PDF file called **protein.pdf**.

Practical part (1/2 of protein points)

Using Bio.PDB, write a class that provides functionality to (a) **automatically download** a structure file specified by its PDB identifier and (b) find **polypeptide regions in the structure file that are potentially of low quality**. Explain the criteria you use to find such regions[1] and briefly outline your implementation. Apply the method to three structures[2], each of different quality (low, medium, high) and describe the results.

Maximum 1 page (use the same PDF file **protein.pdf**), including tables, code snippets, figures and formulas. Submit the code as **protein.py**.

[1] Hint: see the slides from Exercise Session 3, on the PDB data base.
[2] Justify the choice of the three structures.

2 RNA (1/3 of points)

RNA folding with constraints

The structure of an RNA molecule is often influenced by **binding to other molecules**, for example proteins. RNA structure prediction algorithms are based on scoring schemes estimated from RNA molecules in isolation (i.e. no binding partners). Using experimental data as part of the folding algorithm can support a more biologically relevant structure. Various protocols with reactants, e.g. SHAPE, provide information on to which degree individual positions are paired or unpaired, while the partner information is missing. Hence a sequence of N nucleotides can be accompanied by a vector of N elements between 0 and 1, where closer to 1 means more unpaired as measured by the experiment.

1. Describe two different ways to make use of the experimental folding data in the Nussinov scoring scheme and write up the corresponding recursions. Compare strength and weaknesses of the two and argue for the choice of implementation (next task).

2. Update your Nussinov implementation (or the one available via Absalon) to fold an RNA sequence with constraints such that you benefit from the individual positional information of paired/unpaired.

3. Fold the sequences (min loop size 3) in the file **RNAsequences.fasta** with and without the corresponding constraints provided in the file **RNAconstraints.txt**.

4. Compare the constrained and unconstrained structure versions using both the Hamming and the base pair distances. Discuss which one you find to be most biologically relevant.

Structure the report in four parts in the following manner:

1. An introduction to the Nussinov and energy folding algorithms and a justification of the relevance of using constraints. Include an outline of two ways the Nussinov algorithm can be modified to make use of constraints. Argue for what you choose.

2. A material and methods part that describes your Nussinov implementation with your choice of constraints in the folding algorithm. Include the recursions.

3. The results of applying your implementation.

4. A section consisting of a discussion of the suitability (weaknesses and strengths) of the strategy you implemented.

3 Uploading the reports

3.1 Protein part

Upload the protein report as a PDF file called **protein.pdf**. In addition, also upload the Python/Bio.PDB script separately as **protein.py**.

3.2 RNA part

Upload the RNA report as a PDF file called **rna.pdf**. In addition, also upload the Python script separately as **rna.py**.

3.3 General remarks

- **Use the file names that we specified above.**
  - Upload the files separately. **Do not upload a zip or tar file!**
  - **Do not put the code in a Jupyter notebook!** Use an ordinary Python file.
- Commit well-structured, well-documented code that can be executed.
  - Use **functions** and / or **classes** where appropriate.
  - Use **__main__** for task-specific code that is not in functions or classes, ie. for script code.
  - Use **doc strings** and **comments** where appropriate.
  - Use **try/except** and **assert** where appropriate.
  - It should be perfectly clear what your code is doing and how, even without reading the report.
- Please provide references to the literature (or the occasional blog *if and only if* it's about referring to someone's opinion or to specific information not available elsewhere) – **NOT TO WIKIPEDIA** – where needed for both the RNA and the protein part.

4 Plagiarism warning!

Note that your exams will be checked for **plagiarism** by an effective, fully automated method. Do NOT exchange code or text. Do NOT use figures from external sources without proper referencing. Quotes should be always between quotation marks and with a reference to the source.

Severe cases of plagiarism can get you permanently expelled from the university!