Exam Structural Bioinformatics 2023-2024

Jan Gorodkin, Stefan Seemann and Thomas Hamelryck
Deadline: Friday 26/01/2024, 17h00

1 Protein part (2/3 of points)

This assignment is an opportunity for you to explore and critically analyze the latest advancements in the structural bioinformatics of proteins, especially those leveraging machine learning and probabilistic methods (a main focus of the course). Here's a breakdown of your task:

- **Objective:** Your initial goal is to identify a recent peer-reviewed article, published in 2023-2024, that reports a significant advancement in the field of the structural bioinformatics of proteins. This advancement should be in the context of the **machine learning and/or probabilistic methods** you encountered in the course and the reading assignments.
  - **Tip**: Use Google Scholar. You could also look at machine learning conference proceedings, such as ICLR, NeurIPS, AISTATS, ICML, ISMB, RECOMB and so on.
- **Contextual Analysis:** Compare the advancement described in the article to the methods you have studied in the course and as part of your reading assignments (AlphaFold and OmegaFold, Theseus, etc). Examine how this new method addresses existing challenges in protein structural bioinformatics.
  - **Examples of such challenges:** improved 3D structure prediction, protein design, prediction of protein dynamics, probabilistic models of proteins, statistical analysis, etc.
- **Practical Illustration:** Implement a program using the Bio.PDB module in Python to demonstrate a core algorithm, aspect, feature or concept of the article. This should involve applying your program to a protein structure or a set of protein structures.
  - **Note:** You find a database of 100 high quality protein structures (**top100.tar.gz**) on Absalon (see Modules, top of the page). You are welcome to use this set, but you are free to use others.
  - **Tips:** For visualization of proteins, you can make use of Pymol. For plotting and statistical analysis, you can make use of Matplotlib and Scikit-learn.
- **Critical Evaluation:** Discuss the strengths and limitations of the new method. Reflect on its potential impact on the field, how it might influence future research, and possible applications in biotechnology or medicine.

Tips for Success

- **Thorough Research:** Ensure your article selection is relevant and represents a significant advancement in the field.
- **Clear Algorithm Explanation:** Make sure your Bio.PDB implementation is well-explained and **directly relates to the article in a non-trivial way**.
- **Effective Presentation:** Organize your report with a logical flow, and make sure your analysis is clear and concise.

Report Format

Limit your report to a maximum of two pages, excluding figures, tables, algorithms, code and references. Your report should include an introduction, a discussion of the new method and why it is an advancement, the Bio.PDB part (justification, implementation, results) and a short conclusion. Include the Bio.PDB code (or at least the most important parts of it).

2 RNA (1/3 of points)

In this exercise you will modify the Nussinov algorithm for RNA folding to naïvely accommodate stacking information in the scoring. Subsequently, you will assess to which extent your modifications provide a more realistic model compared to the original Nussinov algorithm.

You can take outset in your own implementation of the Nussinov algorithm or use one available in Absalon. Indicate which one you use.

1. In the step of the Nussinov algorithm where you check for base pairing between positions i and j, modify the scoring values to take possible base pair information on the neighboring bases $i+1$ and $j-1$ into account. For the scoring, use the stacked base pairs values below that are derived from the first Turner et al. model (by multiplying the original energy contributions by minus one and shifted such that the minimum is one):

			$(i{+}1, j{-}1)$			
	AU	CG	GC	GU	UA	UG
(i,j) AU	2.4	3.4	3.5	2.7	2.2	1.9
CG	3.4	3.7	4.6	3.4	3.4	2.7
GC	3.5	4.6	4.7	3.8	3.7	2.8
GU	2.7	3.4	3.8	0.0	2.6	1.8
UA	2.2	3.4	3.7	2.6	2.6	2.3
UG	1.9	2.7	2.8	1.8	2.3	1.0

   We provide you with a ready to paste python code of the full scoring matrix under `https://rth.dk/resources/bioinf2024/`. This version also includes the cases when $i+1$ and $j-1$ is not making up a base pair. In those cases the socre of pairing i and j is 1.

2. Once your modification is completed you will have to test if the folding improved compared to RNAfold. Obtain 100 random sequences (of length 120 nt) which were folded with RNAfold

at `https://rth.dk/resources/bioinf2024/` by inserting your KU ID to get the sequences. The sequences and a pairing masks are available as a python code block.

3. Based on the output dot-brackets strings, compare the RNA structures that were predicted by the unmodified Nussinov implementation, your modified Nussinov, and RNAfold. Calculate base pair distance for each of 3 pairwise comparisons (indicate which program you use). Produce three scatter plots between the distances and show the Pearson correlation coefficients. Plot the cumulative distributions in a single plot of the observed distances. Based on your observations discuss to what extend your revised Nussinov implementation predict structures more similar to those retrieved with RNAfold. With outset in the RNA folding energy model discuss why / why not the modified Nussionov algorith, is an improvement over the orginal one.

Report Format

Limit your report to a maximum of two pages, excluding figures, tables, algorithms, code and references. Structure the report as follows:

1. An introduction to the Nussinov and energy folding algorithms. Include an outline of how the Nussinov algorithm can be modified to accommodate the modification of the scoring.
2. A "material and methods" part that describes your Nussinov implementation.
3. The results of applying your implementation.
4. A section discussing the suitability (weaknesses and strengths) of the strategy you implemented and whether it is a step good step in the right direction.

In addition, also upload the Python script separately as rna.py and makes sure that you have added a comprehensive documentation of all essential parts of the code.

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 a 12pt font for the reports.**
- **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 with others. Do NOT use figures, formulas, tables, graphs etc. 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 EXPELLED from the university!