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

4^{th}Assignment - Structure and Systems Bioinformatics

Hand in: 2023-05-25 20:00 CEST (source code and pdf in a single archive, uploaded via ILIAS)

Task 1 — DSSP energy function (60 P)

The DSSP algorithm and its various adaptations are commonly used to annotate a protein's secondary structure based on their tertiary structure. The heart of the algorithm is a function estimating the electrostatic binding energy from the geometry of the C=O group of one residue and the N-H group of another residue.

Your task is to take the high-resolution PDB structure `5JXV`, and write a Python 3 program to:

1. Implement DSSP's energy function (lecture 09B slide 3) and evaluate it on every pair of residues (both ways, i.e. as donor/acceptor and vice versa).
2. Store these values in a 56×56 matrix `E`: element `E[i, j]` should be the energy function assuming residue i as acceptor (O side) and residue j as donor (H side). Output this matrix into a tab-separated file `dssp_matrix.tsv`.
3. Visualize the pairwise energy matrix on a heatmap.
4. Interpret the heatmap. Find secondary structures based on the visible hydrogen bond patterns (no code needed) and list them with the following information:
   a) for helix: start and end residue position, helix type (3_{10} / alpha / pi) with an explanation how you deduced it
   b) for beta sheet: start and end positions for every pair of bound strands, with parallel / antiparallel description

You can use `PDBParser` from Biopython's `Bio.PDB` module but you have to implement the energy calculation yourself using atom coordinates. Notes:

- The PDB file contains 20 models of the same protein. Use the first model for the task.
- Report binding energies in the unit of kJ/mol. PDB files store atomic coordinates in units of Å. On the slide, e_0 stands for elementary charge and ε_0 for vacuum permittivity. (Hints: Refer to the lecture O9B slide 4 for the H-bond threshold)
- If you can't get the units right, you can still score points on the heatmap and secondary structure tasks, as the heatmap's color scale shouldn't prevent you from interpreting it correctly.
- The relevant atom labels are `C`, `O`, `N` and `H` for every residue. The sidechain atoms are irrelevant for this task.

Task 2 — Simplified Chou-Fasman algorithm (40 P)

Unlike DSSP, which annotates secondary structure elements based on tertiary structure, the Chou-Fasman method predicts them from the primary structure, i.e. the sequence. Your task is to implement its simplified, most basic version as seen in lecture 09C, and apply it to the same protein `5JXV`.

To help you get started, we have provided you the sequence, and helix / strand relative probability values and builder/breaker classes for the 20 amino acids, found in the supplementary file `chou-fasman_supplement.py`.

To recap the algorithm's outline:

- Find the first helix core based on builder/breaker class membership with a window of size 6
- Extend the core to the left and right by evaluating relative helix probability values in windows of size 4
- Find the next helix core and repeat, until the end of sequence.
- Find the first strand core based on class membership in a window of size 5
- Extend to the left and right using relative strand probability values in windows of size 4
- Find the next strand core and repeat, until the end of sequence
- Identify segments that were predicted as both helix and strand, and resolve the conflict for each such segment using relative probability values. (Hints: compare helix and strand probabilities in each conflicting segment and determine if the segment is either helix or strand structure)

While searching for cores, make sure to skip already found and extended ones (secondary structures have already been predicted).

You don't have to clean up or "fix" the resulting prediction, so do not bother with removing unrealistically short segments, just leave them as-is.

Please output 1) the helix and strand cores, 2) the helix and strand extended structures before conflict resolving, 3) the conflict resolved structure in one pdf file, and use the following output format for your secondary structure prediction results:

```
XXXXXXXXXXXXXXXXXXX... (AA sequence)
--HHHHH-----SSSS---... (SS prediction)
```

Compare the resulting annotation with the reference annotation (`ss_ref`) included in the supplementary file: calculate the Q_3 accuracy of the simplified Chou-Fasman method for this protein. Is it in line with the "advertised" 50-60%?

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.