Technical University of Munich

Chair for Bioinformatics and Computational Biology
Advanced Bioinformatics SoSe 2023
Prof. B. Rost
L. Richter

SoSe 2023
Exercise sheet 1
25.04.2023 – 02.05.2023

Exercise 1: Neural Networks

(H) Neural Networks ($\sum = 15$)

In this homework you are asked to complete and submit the program `exercise1.py` and additional files. Do **not change** the name of these files. Some of the report files has to be created completely by you. Zip all submission files into one .zip archive. Do not submit the input data file.
Remark: You might want to have a look at the types of the input and output of the function to get an idea how the function should behave. Furthermore, you are allowed to add helper functions if necessary but you may **not change** the **signature** of the given functions. Remember that the states H,E,C also have to be represented by numbers internally. Some of the tasks are specified in the text here others are more described in the respective docstrings. Later this week we will publish test code for some functions, but this still have to be completed.

1.1 (H) Loading Data ... (3)

Complete the function `load_jsonl`. This function accepts a string represented a potential path prefix and the file name of the file to load. It then returns a list of Python dictionary objects which allow to access the entry data with the same keys as in the JSON documents.

1.2 (H) One-hot Encoding 1 (3)

Complete the function `single_one_hot_endcode`. Make sure that the input:

- contains exactly one residue
- the residue is in the set of accepted amino acid symbols defined in `AMINO_ACIDS`

If these contraints are not met the function raises a `ValueError`.
The function returns a Numpy array of size 20 and type `np.int8` with all values 0 except the position which equals the input's index in `AMINO_ACIDS` which value is then 1.

1.3 (H) One-hot Encoding 2 (2)

Complete the function `one_hot_encode_sequence` as described in the docstring.

1.4 (H) One-hot Encoding 3 (2)

Complete the function the function `one_hot_encode_labeled_sequence` as described in the docstring.

1.5 (H) Q3 (2)

Implement the function `calculate_Q3` as it is described in the docstring. Please also consider unusual predictions like certain states missing.

1.6 (H) Predict Secondary Structure (3)

Implement the function `predict_secondary_structure` as describes in the docstring including the following steps. Create a simple train-test split of the input data using `sklearn.model_selection.train_test_split` with `random_state` set to 42 and use the train portion to fit your model. Call the `fit` method to so. Use the `predict()` method to predict the state for the test portion and then calculate and return the Q3.