Predicting Druggable Proteins from Primary Sequence (amino acid sequence)

Druggable proteins are those that can interact with drug-like molecules and are used as targets for drugs that can potentially be effective in treating disease. Predicting the druggability of proteins is therefore a crucial step in the development of new drugs. Computational models that can predict the druggability of proteins using only their primary sequence is a time and cost-effective complement to the more precise yet more time consuming and laborious experimental methods that analyze the tertiary (3D) structure of proteins. In this assignment, you will explore the use of machine learning methods for predicting the durability of proteins from their primary amino acid sequence.

Q1) Identify and list different *types* of features that can be used for predicting druggable proteins from their primary amino acid sequence. You may refer to this paper and other papers in the literature for possible options.

Q2) Select at least 4 types of features you identified in Q1. For each type of selected features, train a suitable classifier based on feature vectors of that type in order to predict druggability using this training data (TR_pos_SPIDER.txt, TR_neg_SPIDER.txt) and evaluate it using the testing data available at the same link (TS_pos_SPIDER.txt, TS_neg_SPIDER.txt). Report the test results for each type of features. When reporting results, clearly indicate the features used, which classification model was used along with its hyperparameters and report the accuracy, sensitivity, specificity, precision and the F1 measure.

Q3) Try to combine the different types of features you used in Q2. You may do this either at the level of features (combining features from multiple types into a single feature vector and training a single classification model) or by building an ensemble of classifiers, where each individual classifier is trained on a different type of features. In either case, you may use feature selection to select the most informative features. Briefly explain the method you used and report the test accuracy, sensitivity, specificity, precision and the F1 measure for the same data mentioned in Q2. See notes below.

> **Note 1**: Use cross validation on training data to explore different models and their hyperparameters and find the model that gives the best cross validated accuracy (or any other measure you prefer) on this training data. Once you find the best model and hyper parameters, retrain it on the full training dataset. Then, use that best model to make predictions on the test data and report results.
>
> **Note 2**: This paper proposes a method of combining multiple types of features and provides the source code of its implementation. ***Do NOT*** copy the same method or the source code. Try different methods of your own. Your method producing a lower accuracy than what is achieved in the said paper is ***not*** a problem at all.

Q4) Compare the test results you achieved for the best model in Q3 with those of the best model in Q2. Assess whether there is a statistically significant difference between those two models. Use an appropriate statistical test to support your answer and justify its use (you may refer to this paper as an example).

Deliverables:

1. A report in PDF format providing answers to Q1 – Q4.
2. Source code. Your script should be executable from the command line and accept training and testing data in FASTA format as four file names as shown below in the given order. There should be ***no*** other inputs or parameters to set.

   Input format:
   - your_script_name <positive training data> <negative training data> <positive testing data> <negative testing data>

   Output format:

   - Output should print the test accuracy, sensitivity, specificity, precision and the F1 measure. Additionally, it should produce two text files named <predictions_pos.txt> and <predictions_neg.txt> containing the prediction results (1 if druggable or 0 if not) for the sequences in the test data files <positive testing data> and <negative testing data>, respectively.