PR1: Peptide Classification

Published Date:
Jan. 18, 2024, 9:00 p.m.

Deadline Date:
Jan. 31, 2024, 11:59 p.m.

Description:

****
This is an individual assignment.
****
Overview and Assignment Goals:

The objectives of this assignment are the following:

- Create feed-forward neural networks and train them using your own codes and frameworks.
- Experiment with different feature extraction techniques.
- Think about dealing with imbalanced data.

Detailed Description:
Develop predictive neural networks that can determine, given an antibacterial peptide, whether it is also an antibiofilm peptide.

"Proteins are large biomolecules, or macromolecules, consisting of one or more long chains of amino acid residues. Proteins perform a vast array of functions within organisms, including catalysing metabolic reactions, DNA replication, responding to stimuli, providing structure to cells, and organisms, and transporting molecules from one location to another. Proteins differ from one another primarily in their sequence of amino acids, which is dictated by the nucleotide sequence of their genes, and which usually results in protein folding into a specific three-dimensional structure that determines its activity.

A linear chain of amino acid residues is called a polypeptide. A protein contains at least one long polypeptide. Short polypeptides, containing less than 20-30 residues, are rarely considered to be proteins and are commonly called peptides. [...] The sequence of amino acid residues in a protein is defined by the sequence of a gene, which is encoded in the genetic code. In general, the genetic code specifies 20 standard amino acids; [...] Proteins can also work together to achieve a particular function, and they often associate to form stable protein complexes." [Wikipedia, Accessed 2020-02-07, https://en.wikipedia.org/wiki/Protein]

Biofilms are tightly-connected multicellular communities of microorganisms encased in self-secreted extra-cellular matrices. They are currently one of the major causes of disease for

two main reasons. First, roughly 75% of all human infections are caused by biofilms. Second, due to the robust multicellular cellular matrix structure, they are resistant both to the host defense mechanisms and to traditional antimicrobial compounds (antibiotics). Thus, it is important to identify peptide sequences that are not only antimicrobial (can destroy or render inert the invading microorganism), but also antibiofilm (can penetrate the extra-cellular matrix so it can get to the microorganism in the first place).

You have been provided with a training set (train.dat) and a test set (test.dat) consisting of peptide sequences, one per line in the file. Peptides are encoded as strings with characters from an alphabet of 20 characters, each representing an amino-acid residue. The training set also includes the label for each sequence as 1 (antibiofilm) or -1 (not antibiofilm) as the first character in each line of the training file, separated from the sequence by a tab (\t) character.

The input to your classifiers will not be the peptides themselves, but rather features extracted from the peptides. Two simple approaches for feature extraction are the bag-of-words and the k-mer models you should have learned about in Data Mining or Machine Learning, where a word is one of the amino-acids in the peptide. **You should not use any additional external data in this assignment.**

Note that the dataset is imbalanced. We will Matthews's correlation coefficient (MCC) as evaluation metric for this assignment, which, similar to the F-1 score, combines aspects of the result's sensitivity and specificity. Given the normal confusion matrix resulting from comparing the predicted and true classes of the test samples, MCC is defined as,

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Programs:

You are required to write two separate programs for the classification. The first may only use basic Python structures (from numpy or scipy) and you should implement your own functions for training the neural network. This is also the program you will use to make CLP submissions. In addition, you should write a second program that uses a deep learning framework of your choice to train the neural network. The structure of the network may be the same or different from the one you created in the first program. You will present results from this program (which should be at least as good as those from the first program) in your report.

Considerations:

+ Try extracting different features from the peptide strings.
+ Consider oversampling the negative class to fix the apparent imbalance.
+ Try out different network configurations and activation functions.
+ Consider regularization as a way to keep weights balanced in the network.

Data Description:

The training dataset consists of 1566 records and the test dataset consists of 392 records. We provide you with the training class labels and the test labels are held out. Your task is to predict those labels for the peptides in the test set and create a test.txt file containing those labels, which you will submit to CLP. Note that CLP only accepts files with extensions .txt or .dat for your predicted labels, and .py or .ipynb or .zip or .tgz for codes.

Rules:

- This is an individual assignment. Discussion of broad level strategies are allowed but any copying of prediction files and source codes will result in an honor code violation.
- You are allowed 5 submissions per day.
- After the submission deadline, only your chosen or last submission is considered for the leaderboard.

Deliverables:

- Valid submissions to the Leader Board website: https://clp.engr.scu.edu (username is your SCU username and your password is your SCU password).

Canvas Submission for the report:

- Include a 2-page, single-spaced report describing details regarding the steps you followed for feature extraction, designing your neural network, and training your model. The report should be in PDF format and the file should be called **report.pdf**. The report needs to be structured as a technical report (title, abstract, introduction, sections, conclusion), be free from grammatical errors, and use standard page and font sizes (letter size page, 10 or 11 pt font). Be sure to include the following in the report:
    1. Name and SCU ID.
    2. Rank & MCC-score for your submission (at the time of writing the report). If you chose not to see the leaderboard, state so.
    3. Your approach.
    4. Your methodology of choosing the approach and associated parameters.
- Ensure you submitted the correct code on CLP that matches your output.
- Zip up your report and codes for both programs in an archive called **<SCU_ID>.zip or <SCU_ID>.**tgz and submit the archive to Canvas.

Grading:

Grading for the Assignment will be split on your implementation (70%) and report (30%). Extra credit (1% of final grade) will be awarded to the top-3 performing algorithms. Note that extra credit throughout the semester will be tallied outside of Canvas and will be added to the final grade at the end of the semester.

Files: available on Canvas.