COMP 4970/7970 Project 5: 15 points 15% Credit
Final Submission due before 11:59 PM Thursday April 26

Instructions:

1. This is a group project. You should do your own work while working collaboratively as a group. Any evidence of copying either from a public source or from the works of other groups without due credits will result in a zero grade and additional penalties/actions against **all members of the involved groups.**
2. Select one member from your team as a group leader for the project who did not serve as group leader before during the course. The group leader will be responsible for the task allocation, progress tracking, combining works from the team members, maintaining all communications with the instructor, and for certifying efforts of the team members. **The group leader will also deliver the project presentation in class and upload the final submission to Canvas on behalf of the entire group.**
3. **No show in project presentation or final submissions by email or late submissions (even by minutes) will receive a zero grade for the entire group.** No makeup will be offered unless prior permission has been granted, or there is a valid and verifiable excuse.

Project presentation (5 points):

1. All group presentations scheduled **in class on Thursday April 26**.
2. The group leader to deliver a PowerPoint presentation for 13 minutes followed by additional 2 minutes of Q&A.
3. Effort citification (5-point scale) for each member of the team must be presented at the end of the presentation.

Final Submission (10 points):

1. Python source files containing your code only (no test data needed).
2. Completed report document (Word or PDF format) and Readme.txt file (template provided).
3. Group leader uploads all files to Canvas as a single zipped folder before the deadline on the due date.

Implementing Linear Regression for Pairwise Protein Structural Similarity Prediction

Objective: Implement linear regression for pairwise protein structural similarity prediction.

You must use standard Python programming language. You are NOT allowed to use existing packages or libraries (e.g. Biopython).

A: Raw Data:

Two sets of 150 data files (<*pdb_id*>*.fasta* and the corresponding <*pdb_id*>*_reindexed.pdb*) are supplied under 'data' directory. Each <*pdb_id*>*.fasta* file contains a single protein sequence in FASTA format.

The corresponding <*pdb_id*>*_reindexed.pdb* file contains the protein 3D structures in PDB format.

B: Curating Training and Test Datasets:

First identify all unique pairs of proteins in the raw dataset and divide the unique pairs into non-overlapping sets of training (~75%) and test (~25%) datasets using simple random sampling without replacement. Be mindful of symmetry when identifying unique pairs of proteins (i.e. Protein_1 vs. Protein_2 is same as Protein_2 vs. Protein_1).

C. Pairwise 3D Structure Comparison using TM-align

Compare all unique pairs of protein 3D structures using TM-align program located online at https://zhanglab.ccmb.med.umich.edu/TM-align/ and record the TM-scores by parsing the output of TM-align. Executable version of TM-align for Linux system is available at https://zhanglab.ccmb.med.umich.edu/TM-align/TMalign.gz (No installation required).

Please note that TM-align protein comparison is not symmetric and so take the average of the two TM-scores reported by TM-align program (one normalized by length of Chain_1 and the other normalized by length of Chain_2).

Sample output of TM-align is shown below:

```
 **************************************************************************
 *                      TM-align (Version 20170708)                       *
 * An algorithm for protein structure alignment and comparison            *
 * Based on statistics:                                                   *
 *       0.0 < TM-score < 0.30, random structural similarity              *
 *       0.5 < TM-score < 1.00, in about the same fold                    *
 * Reference: Y Zhang and J Skolnick, Nucl Acids Res 33, 2302-9 (2005)    *
 * Please email your comments and suggestions to: zhng@umich.edu          *
 **************************************************************************

Name of Chain_1: A896994
Name of Chain_2: B896994
Length of Chain_1:  154 residues
Length of Chain_2:  146 residues

Aligned length=  143, RMSD=   1.83, Seq_ID=n_identical/n_aligned= 0.245
TM-score= 0.81453 (if normalized by length of Chain_1)
TM-score= 0.85377 (if normalized by length of Chain_2)
(You should use TM-score normalized by length of the reference protein)

(":" denotes aligned residue pairs of d < 5.0 A, "." denotes other aligned residues)
MVLSEGEWQLVLHVWAKVEADVAGHGQDILIRLFKSHPETLEKFDRVKHLKTEAEMKASEDLKKHGVTVLTALGAILKKK--G-HHEAELKPLAQSHATKHKIPIKYLEFISEAIIHVLHSRHPGNFGADAQGAMNKALELFRKDIAAKYKELGYQG
 ::::::::::::::::::::::::::::::::::::::::::::::::::: :::::::::::::::::::::::::::  : :::::::::::::: :::::::::::::::::::::::::  :::::::::::::::::::: :     ::
-SLSAAEADLAGKSWAPVFANKNANGLDFLVALFEKFPDSANFFADFKGKS-VADIKASPKLRDVSSRIFTRLNEFVNNAANAGKMSAMLSQFAKEHV-GFGVGSAQFENVRSMFPGFVASVAA--PPAGADAAWTKLFGLIIDALKA-A-----GA
```

D. Position Specific Scoring Matrix (PSSM) generation:

Step 1: Download NCBI PSIBLAST to <your_blast_path> as follows:

```
$ cd <your_blast_path>
$ wget https://zhanglab.ccmb.med.umich.edu/PSSpred/blastv2.6.tar.gz
$ tar xvfz blastv2.6.tar.gz
```

The executable programs can be found at <your_blast_path>/blast/bin/

Step 2: Download NR database to <your_nr_path> as follows:

```
$ cd  <your_nr_path>
$ wget http://calla.rnet.missouri.edu/qprob/nr_database.tar.gz
$ tar xvfz nr_database.tar.gz
```

The nr database can be found at <your_nr_path>/nr_database

Step 3: Test PSSM generation for a test protein sequence located at <your_test_path>/test.fa:

```
$cd <your_test_path>
$ <your_blast_path>/blast/bin/blastpgp -d <your_nr_path>/nr_database/nr -j 3 -b
1 -a 4 -i test.fa -Q test.pssm
```

The output PSSM file can be located at <your_test_path>/test.pssm

N.B. Sample test.fa and test.pssm files are included in the project.

E. Feature Generation:

For each unique protein pair:

i) Use the average of the 20 PSSM "weighted observed percentages rounded down" and scale them down to a scale between 0 and 1 (i.e. divide by 100) from the .pssm file generated by PSIBLAST. For a pair of proteins, this would result in a total of 40 features (20 features/protein). Naturally, all features are in the range [0,1).

ii) Use your previously developed protein secondary structure predictor to predict three-class secondary structure of the protein pair from their primary sequence. Calculate the proportion of helix (H), strand (E), and coil (C) for each pair, resulting in a total of 6 features (3 features/protein). Consequently, all features are in the range [0,1).

iii) Use your previously developed protein solvent accessibility predictor to predict two-class solvent accessibility of the protein pair from their primary sequence. Calculate the proportion of exposed (E), buried (B) for each pair, resulting in a total of 4 features (2 features/protein). Again, all features are in the range [0,1).

F. Linear Regression Learning on Training Set:

Implement the gradient descent based optimizer to learn the weight vector of Linear Regression using the training set. You may choose either MCLE or MAP estimation during training as well as batch gradient descent or stochastic gradient descent (or a combination of both).

G. Linear Regression on Test Set:

Implement Linear Regression using the learned weight vector to predict the TM-score of a pair of test protein from their FASTA formatted sequence. Note that you need to predict secondary stricture and solvent accessibility for the protein pair using your previously-developed program.

N.B. Linear Regression is an offline-learning algorithm. Therefore, training and prediction should be implemented separately. The prediction algorithm should take a protein sequence in FASTA format as an input and predict TM-score in a standalone mode. You may save the parameters learned during training in a file that can be fed into the prediction engine, in an offline mode.

H. Evaluate Accuracy:

Report accuracy of average squared error (true TM-score – predicted TM-score)^2 on the test set.