COMP 5970/6970 Project 5: 100 points 20% Credit
Final Submission due before 11:59 PM Monday April 20

Instructions:

1. This is an individual project. You should do your own work. Any evidence of copying either from a public source or from the works of other without due credits will result in a zero grade and additional penalties/actions**.**
2. **Submissions by email or late submissions (even by minutes) will receive a zero grade.** No makeup will be offered unless prior permission has been granted, or there is a valid and verifiable excuse.
3. **No show for your project presentation will receive a zero grade. There is also a penalty for missing a presentation day in which you are not presenting.**

Submission:

For 5970, you are required to upload the following to canvas before **11:59 PM Monday April 20**:

1. **Source Code**: Python source files (upload .zip file in case of multiple files) containing your code only (no test data needed) and ReadMe.txt file (template provided) describing how to run your code. Note that we will NOT debug your code. If your code does not execute as described in ReadMe.txt, you will receive a zero grade.
2. **Presentation Slide**: One slide only in PPT/PPTX/PDF format to be used during the oral presentations (see below). If you submitted file spans more than a page, we will extract the first page for the oral presentation.

For 6970, you are required to upload the following to canvas before **11:59 PM Monday April 20**:

1. **Source Code**: Python source files (upload .zip file in case of multiple files) containing your code only (no test data needed) and ReadMe.txt file (template provided) describing how to run your code. Note that we will NOT debug your code. If your code does not execute as described in ReadMe.txt, you will receive a zero grade.
2. **Presentation Slide**: One slide only in PPT/PPTX/PDF format to be used during the oral presentations (see below). If you submitted file spans more than a page, we will extract the first page for the oral presentation.
3. **Project Report**: Completed report document in PDF format using template provided. Make sure to have all necessary sections of scientific writing: abstract, introduction, methods, results, discussion, references.

Presentations:

Presentations will be during the class on **Wednesday April 22** and **Friday April 24**.

Attendance is mandatory during all the presentation days. Missed presentation days without university-approved excuse will result in a penalty of 25 points for each missed class. Note that this penalty will be applied when you miss a presentation day in which you are not presenting. **No show for your project presentation will receive a zero grade.**

Everyone is required to deliver 3 minutes flash presentation accompanied by the submitted slide following the Three Minute Thesis (3MT) format, with additional 2 minutes for Q&A:

1. Your presentation should at least contain methods (i.e., implementation), results (e.g., output), and conclusion.
2. Having appropriate graphics and visuals (e.g., figures, plots) in the presentation slides to help illustrate key concepts or results will be positively graded.
3. Any additional scientific insights and/or challenges faced and/or limitations of your implementation and/or efficiency analyses and/or comparisons with alternative approaches will be positively graded.
4. Practice your talk not to exceed the time limit or finish too early.
5. No need to bring your slides. We will set things up and decide the presentation sequence.

Implementing Linear Regression for Pairwise Protein Structural Similarity Prediction

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

Note: You must use standard Python programming language. You are NOT allowed to use non-standard packages or libraries (e.g. Biopython, scikit-learn, SciPy, NumPy, etc.).

A: Raw Data:

fasta, PSSM, and tmalign files for the set of 150 proteins are supplied.

B: Curating Training and Test Datasets:

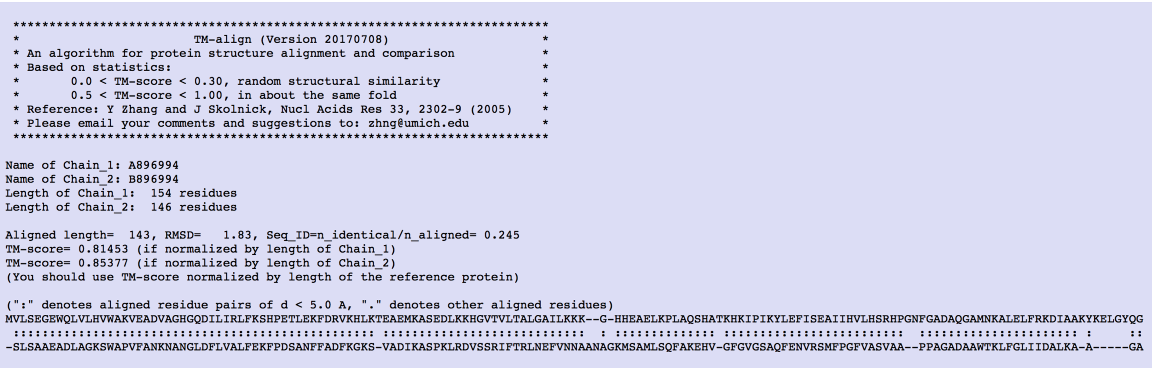
First identify all unique pairs of proteins in the raw dataset and divide the unique pairs into non-overlapping sets of training (~75%) and test (~25%) datasets using simple random sampling without replacement. Be mindful of symmetry when identifying unique pairs of proteins (i.e. Protein_1 vs. Protein_2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

C. Computing 3D Structural Similarity using TM-align

After identifying all unique pairs of protein 3D structures, compute their structural similarity via TM-align. TM-align program can be located online at https://zhanglab.ccmb.med.umich.edu/TM-align/. For convenience, we have provided the result of running TM-align program for all pairs of proteins in the tmalign directory. The similarity between a pair of proteins is reported through "TM-score" and you will have to parse the tmalign output files to get these values.

Please note that TM-align structural 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. 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. 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).

E. Linear Regression Learning on Training Set:

Implement the gradient descent based optimization algorithm to learn the weight vector of Linear Regression using the training set. You may choose to optimize MCLE via batch gradient ascent or stochastic gradient ascent (or a combination of both).

F. 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 structure 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 FASTA and PSSM files as 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.

G. Evaluate Accuracy:

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