COMP 4970/7970 Project 4: 15 points 15% Credit
Final Submission due before 11:59 PM Thursday April 12

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 12**.
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 Logistic Regression for Protein Contact Map prediction

Objective: Implement logistic regression for protein residue-residue contact map 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>.rr*) are supplied under 'data' directory. Each *<pdb_id>.fasta* file contains a single protein sequence in FASTA format.

The corresponding *<pdb_id>.rr* file contains the all the residue-residue contacts in RR format.

RR format starts with the sequence of the protein target. The sequence is followed by the list of contacts in a five-column format:

i j d1 d2 d :

- indices i and j of the two residues in contact such that i < j, i.e. only half of the contact map is supplied. Furthermore, |i – j| > 5, i.e. the sequence separation between the two residues in contact is at least six.
- the numbers d1 and d2 indicate the distance limits defining a contact. A pair of residues is defined to be in contact when the distance between their C-beta atoms (C-alpha in case of glycine) is less then 8 Angstroms (Å). Therefore, typically d1= 0Å and d2= 8Å.
- the real number d indicates the actual intra residue distance of the two residues being in contact in Angstroms. (During classification, this column can be used for the probability of contact).

An example RR format is provided below:

```
AAYKVTLVTPTGNVEFQCPDDVYILDAAEEEGIDLPYSCRAGSCSSCAGKLKTGSLNQDDQSFLDDDQIDEGWVLTCAAYPVSDVTI
1 19 0 8 6.006
1 20 0 8 5.264
1 21 0 8 7.431
10 89 0 8 7.422
10 90 0 8 5.058
14 28 0 8 7.710
14 31 0 8 7.623
14 33 0 8 5.962
16 27 0 8 6.549
16 28 0 8 6.822
16 31 0 8 4.480
16 33 0 8 7.700
18 24 0 8 6.397
18 27 0 8 4.137
              .
              .
              .
```

N.B. The RR files are extracted from the tertiary structures of the proteins deposited in the Protein Data Bank (PDB).

B: Curating Training and Test Datasets:

Divide the raw data into non-overlapping sets of training (~75%) and test (~25%) datasets using simple random sampling without replacement.

C. 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.

D. Feature Generation:

Use the 20 PSSM values for each pair of residues (i, j), where j > i +5, in a protein from the .pssm file generated by PSIBLAST. Fist few lines for the test.pssm file are given below. For a protein sequence of N residues, there will be N x 20 PSSM values.

```
          A  R  N  D  C  Q  E  G  H  I  L  K  M  F  P  S  T  W  Y  V
  1 E    -3 -2 -2  3 -6  4  5 -4 -3 -6 -5  3 -4 -6 -4 -3 -3 -5 -5 -5
  2 A     1 -2 -2  2 -4  1  1  0 -4 -3 -1 -2 -2 -5  2  3  1 -2 -5 -1
  3 E    -1  4 -2  1 -5  1  0 -3  0 -2 -3  3 -2 -4  3  0  0 -5 -4 -1
  4 A     0  0 -2  0 -5  2  1 -1 -2 -4 -5  0 -3 -6  6 -1 -1 -6 -5 -4
  5 S     1 -2  1  5 -5 -1  3 -1 -1 -5 -5  0 -3 -6  0  1 -1 -6 -5 -4
  6 I    -1 -2 -5 -5 -4 -4 -3 -5  0  3  0 -3 -1  5 -2 -4 -1 -2  1  3
  7 C    -2 -6 -5 -6 11 -6 -6 -5 -6 -4 -4 -6 -4 -5 -6 -3 -3 -5 -5 -3
  8 S    -1  0  1 -1 -3  1  0 -3  3 -2  2  0  0  0 -5  1  0 -1  1 -2
  9 E    -1 -3 -5 -3 -4  3  2 -5 -3 -2  5 -3  2 -1 -3 -3 -3 -2 -3 -1
 10 P    -2 -3 -4 -3 -5 -1  0 -5 -3 -5 -4 -2 -4 -6  8 -1 -3 -6 -5 -4
 11 K     2  0 -3 -3 -5  0 -1 -4 -1 -2 -2  4  1 -5  4  0 -3 -5 -4  0
 12 K    -1 -2 -1  5 -5 -2  3 -4 -2 -2 -2  2 -2 -4 -4 -2 -3 -5  0  2
 13 V    -1  0 -3 -3 -3  0  0 -3 -4 -1 -3  0 -2 -4  3  1  3 -5 -3  3
 14 G    -2 -4 -3 -4 -5 -4 -4  7 -5 -7 -6 -4 -5 -6 -4 -3 -4 -5 -6 -6
 15 R    -3  2 -2 -3 -5 -3 -2 -4  1 -1 -2 -1 -2 -3  7 -2 -1 -3 -2 -2
```

Additionally, use a sliding window of 5 around each residue pairs (i.e. i ± 2 and j ± 2) for feature generation. Therefore, there will be 20 x 5 x 2 = 200 PSSM values for each non-terminal residue pairs. For terminal residues that do not have one or more neighbors on either side, use -1 as dummy PSSM values. Use the corresponding RR files to generate binary class labels: 1 if (i, j) is in contact, 0 otherwise. Note that this needs to done for all pairs of residues in a protein.

E. Logistic Regression Learning on Training Set:

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

F. Logistic Regression Classification on Test Set:

Implement Logistic Regression classifier that uses the learned weight vector to calculate the probability all pairs of residue pairs (i, j) to be in contact give a single FASTA formatted sequence. Sort the contacts by non-increasing probabilities and save them in RR format.

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

G. Evaluate Accuracy:

Report accuracy of 'top' L/10, L/5, L/2 predicted contacts to evaluate the classification performance averaged over the proteins in the test dataset, where L is the length of the protein sequence and 'top' contacts are the contacts predicted with high probabilities (i.e. present at the top of the predicted RR file, which is sorted by non-increasing probabilities).