COMP 5970/6970 Project 2: 100 points 20% Credit
Submission due before 11:59 PM Monday February 24

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 February 24**:

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 February 24**:

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 February 26** and **Friday February 28**.

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 decision tree for protein RSA prediction

Objective: Implement decision tree for protein relative solvent accessibility 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:

Two directors (*fasta* and *sa*) are supplied. The *fasta* directory contains 150 protein sequences in FASTA format. A FASTA file is as follows:

```
>sequenceID
AAGTAGGAATAATATCTTATCATTATAGATAAAAACCTTCTGAATTTGCTTAGTGTGTATACGACTAGACATATATCAG
CTCGCCGATTATTTGGATTATTCCCTG
```

The true binary relative solvent accessibility (RSA) labels of these proteins can be found in the *sa* directory. This file is also in FASTA format. RSA labels having two possible values:

'E: exposed
'B': buried

N.B. The true RSA labels are calculated using the DSSP (Dictionary of Protein Secondary Structure: Pattern Recognition of Hydrogen-Bonded and Geometrical Features. Kabsch and Sander, 1983) software at a 25% threshold.

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. Feature Extraction:

Using chemical properties of 20 naturally occurring amino acid residues as detailed in Table 1 and Figure 1, construct a feature matrix (or vector) for the training and test datasets.

Table 1. Chemical properties of 20 naturally occurring amino acid residues (Livingstone & Barton, CABIOS, 9, 745-756, 1993)

Amino acid	Abbrev.	Side chain	Hydro-phobic	Polar	Charged	Small	Tiny	Aromatic or Aliphatic	van der Waals volume	Codon	Occurrence in proteins (%)
Alanine	Ala, A	$-CH_3$	X	-	-	X	X	-	67	GCU, GCC, GCA, GCG	7.8
Cysteine	Cys, C	$-CH_2SH$	X	-	-	X	-	-	86	UGU, UGC	1.9
Aspartate	Asp, D	$-CH_2COOH$	-	X	negative	X	-	-	91	GAU, GAC	5.3
Glutamate	Glu, E	$-CH_2CH_2COOH$	-	X	negative	-	-	-	109	GAA, GAG	6.3
Phenylalanine	Phe, F	$-CH_2C_6H_5$	X	-	-	-	-	Aromatic	135	UUU, UUC	3.9
Glycine	Gly, G	-H	X	-	-	X	X	-	48	GGU, GGC, GGA, GGG	7.2
Histidine	His, H	$-CH_2-C_3H_3N_2$	-	X	positive	-	-	Aromatic	118	CAU, CAC	2.3
Isoleucine	Ile, I	$-CH(CH_3)CH_2CH_3$	X	-	-	-	-	Aliphatic	124	AUU, AUC, AUA	5.3
Lysine	Lys, K	$-(CH_2)_4NH_2$	-	X	positive	-	-	-	135	AAA, AAG	5.9
Leucine	Leu, L	$-CH_2CH(CH_3)_2$	X	-	-	-	-	Aliphatic	124	UUA, UUG, CUU, CUC, CUA, CUG	9.1
Methionine	Met, M	$-CH_2CH_2SCH_3$	X	-	-	-	-	-	124	AUG	2.3
Asparagine	Asn, N	$-CH_2CONH_2$	-	X	-	X	-	-	96	AAU, AAC	4.3
Proline	Pro, P	$-CH_2CH_2CH_2-$	X	-	-	X	-	-	90	CCU, CCC, CCA, CCG	5.2
Glutamine	Gln, Q	$-CH_2CH_2CONH_2$	-	X	-	-	-	-	114	CAA, CAG	4.2
Arginine	Arg, R	$-(CH_2)_3NH-C(NH)NH_2$	-	X	positive	-	-	-	148	CGU, CGC, CGA, CGG, AGA, AGG	5.1
Serine	Ser, S	$-CH_2OH$	-	X	-	X	X	-	73	UCU, UCC, UCA, UCG, AGU,AGC	6.8
Threonine	Thr, T	$-CH(OH)CH_3$	X	X	-	X	-	-	93	ACU, ACC, ACA, ACG	5.9
Valine	Val, V	$-CH(CH_3)_2$	X	-	-	X	-	Aliphatic	105	GUU, GUC, GUA, GUG	6.6
Tryptophan	Trp, W	$-CH_2C_8H_6N$	X	-	-	-	-	Aromatic	163	UGG	1.4
Tyrosine	Tyr, Y	$-CH_2-C_6H_4OH$	X	X	-	-	-	Aromatic	141	UAU, UAC	3.2

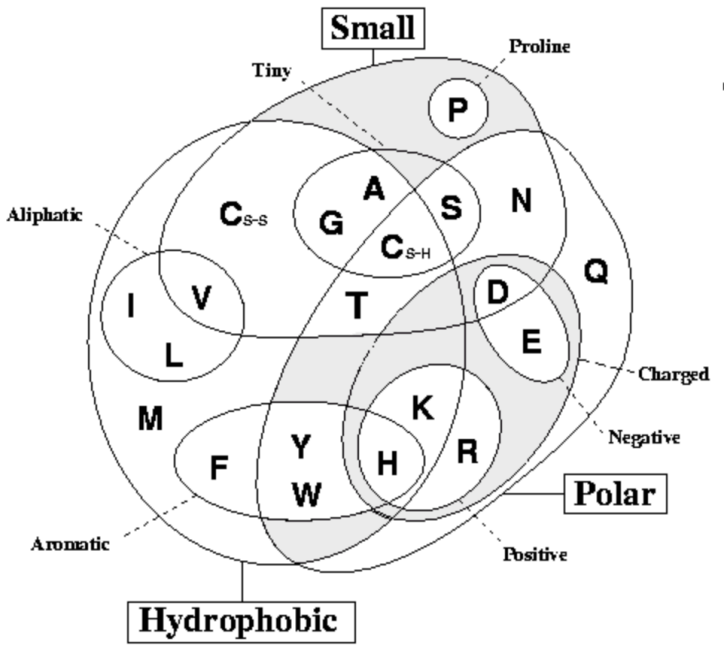

Figure 1. Venn diagram of chemical properties of 20 naturally occurring amino acid residues (Livingstone & Barton, CABIOS, 9, 745-756, 1993)

Specifically, the feature set should include the following binary attributes:

Attribute	Description
Hydrophobic	Whether a residue is hydrophobic
Polar	Whether a residue is hydrophobic
Small	Whether a residue size is small
Proline	Whether a residue is Proline (PRO, P)
Tiny	Whether a residue size is tiny
Aliphatic	Whether a residue is Aliphatic
Aromatic	Whether a residue is Aromatic
Positive	Whether a residue is Positively Charged
Negative	Whether a residue is Negatively Charged
Charged	Whether a residue is Charged

The output labels are already binary (e.g. 1 for exposed, 0 for buried or vice versa).

D. Decision Tree Learning using ID3 on Training Set:

Implement the ID3 decision tree learning algorithm that follows a greedy top-down growth of the tree using information gain to learn the best hypothesis on training dataset.

E. Decision Tree Classification on Test Set:

Implement decision tree classification algorithm that walks on the trained tree generated from step D and output predicts labels on test dataset.

N.B. ID3 decision tree 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 labels 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.

F. Evaluate Accuracy:

Use Precision, Recall, and F-1 score to calculate the accuracy of the decision tree classifier implemented in step E on test dataset.