COMP 4970/7970 Project 2: 15 points 15% Credit
Final Submission due before 11:59 PM Thursday March 1

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 March 1**.
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 decision tree for protein RSA prediction

Objective: Implement decision tree for protein relative solvent accessibility 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 data files (*Proteins.fa* and *Proteins.sa*) are supplied. The *Proteins.fa* file contains 5,772 protein sequences in Multi-FASTA format. A multi-FASTA file contains multiple FASTA formatted sequences.

```
>sequenceID-001 description
AAGTAGGAATAATATCTTATCATTATAGATAAAAACCTTCTGAATTTGCTTAGTGTGTAT
ACGACTAGACATATATCAGCTCGCCGATTATTTGGATTATTCCCTG
>sequenceID-002 description
CAGTAAAGAGTGGATGTAAGAACCGTCCGATCTACCAGATGTGATAGAGGTTGCCAGTAC
AAAAATTGCATAATAATTGATTAATCCTTTAATATTGTTTAGAATATATCCGTCAGATAA
TCCTAAAAATAACGATATGATGGCGGAAATCGTC
>sequenceID-003 description
CTTCAATTACCCTGCTGACGCGAGATACCTTATGCATCGAAGGTAAAGCGATGAATTTAT
CCAAGGTTTTAATTTG
```

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

'e': exposed
'-': 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 (and optionally Validation) Datasets:

Divide the raw data into non-overlapping sets of training (~75%) and test (~25%) datasets using simple random sampling without replacement. You may further divide the resulting training subset to create a validation set to avoid overfitting (not mandatory to create a validation set).

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 (and optionally validation 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

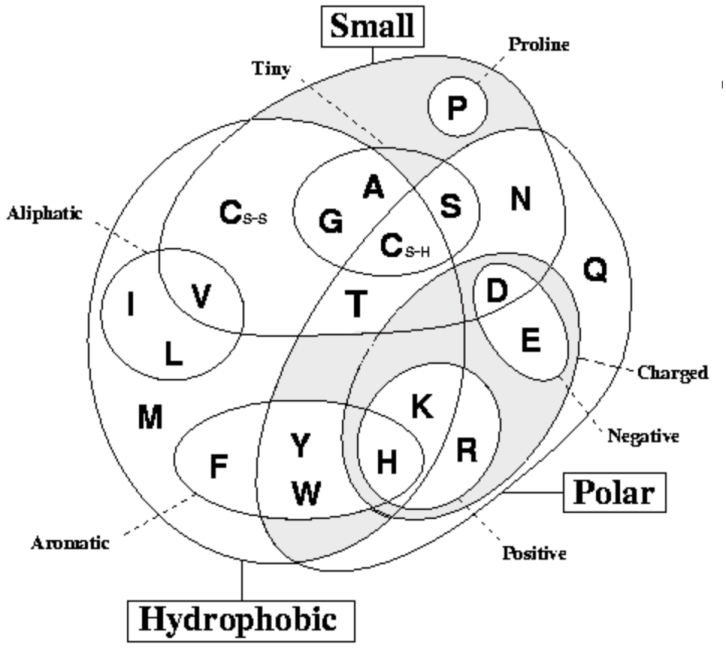

Figure 1. Venn diagram of chemical properties of 20 naturally occurring amino acid residues (Livingstone & Barton, CABIOS, 9, 745-756, 1993)

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. You may optionally reduce overfitting by implementing reduced error pruning algorithm over the validation set.

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.

F. Evaluate Accuracy:

Use Precision, Recall, Accuracy, F-1 score, Mathews Correlation Coefficient (MCC) to calculate the accuracy of the decision tree classifier implemented in step E on test dataset.