





TASK 3

PROTEIN MUTATION CLASSIFICATION

You are in the group  Lossless consisting of  liyueli (liyueli@student.ethz.ch (mailto://[u'liyueli@student.ethz.ch'])),  yinxue (yinxue@student.ethz.ch (mailto://[u'yinxue@student.ethz.ch'])) and  yuzhong (yuzhong@student.ethz.ch (mailto://[u'yuzhong@student.ethz.ch'])).

1. READ THE TASK DESCRIPTION

2. SUBMIT SOLUTIONS

3. HAND IN FINAL SOLUTION

1. TASK DESCRIPTION

INTRODUCTION

Proteins are large molecules. Their blueprints are encoded in the DNA of biological organisms. Each protein consists of many amino acids: for example, our protein of interest consists of a little less than 400 amino acids. Once the protein is created (synthesized), it folds into a 3D structure, which can be seen in Figure 1. The mutations influence what amino acids make up the protein, and hence have an effect on its shape.

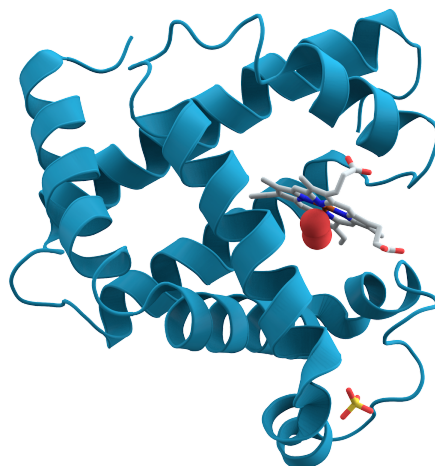


Figure 1: 3D structure of a protein

TASK

The goal of this task is to classify mutations of a human antibody protein into active (1) and inactive (0) based on the provided **mutation information**. Under active mutations the protein retains its original function, and inactive mutation cause the protein to lose its function. The mutations differ from each other by 4 amino acids in 4 respective sites. The sites or locations of the mutations are fixed. The amino acids at the 4 mutation sites are given as 4-letter combinations, where each letter denotes the amino acid at the corresponding mutation site. Amino acids at other places are kept the same and are not provided.

For example, **FCDI** corresponds to amino acid F (Phenylalanine) being in the first site, amino acid C (Cysteine) being in the second site and so on. The Figure 2 gives translation from symbols to amino acid chemical names for the interested students. The biological and chemical aspects can be abstracted to solve this task.

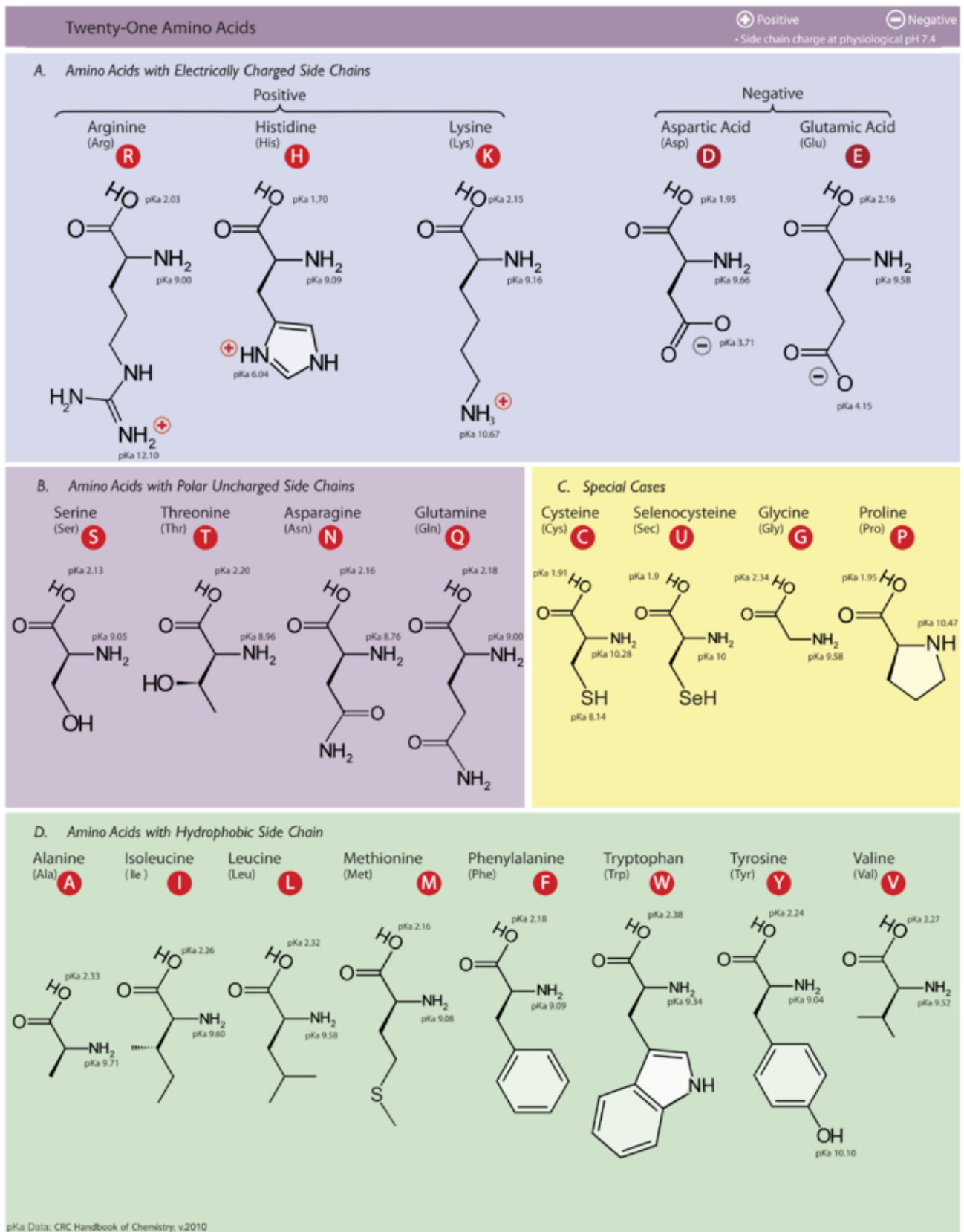


Figure 2: Table containing chemical grouping and abbreviations for common amino acids (source: https://commons.wikimedia.org/wiki/File:Amino_Acids-wide.svg)

DATASET DESCRIPTION

[Download handout \(/static/task3_ks39mcp5.zip\)](#)

The handout you download contains the following files:

- **train.csv** training set

- **test.csv** test set
- **sample.csv** a sample submission file in the correct format

Each line in train.csv corresponds to a single mutation. The dataset contains 112000 rows, where each row is associated with a mutation described by a sequence of four letters (amino acids) and its activity (label).

Sequence	Active
DKWL	0
FCHN	0
...	...
LCLA	1
...	...

The test.csv has the same structure, however the column for activity is omitted.

You should submit a csv file containing the predicted activity of mutations in the same order as in the test set. The csv should contain only 0 or 1 in each separate line. For your convenience, we provide a sample submission file which has been generate randomly.

EVALUATION

For the practical purposes, it is very important to detect nearly all active mutations such that they can be evaluated. Hence we need to maximize recall (true positive rate), but at the same time we want to have equally good precision. Therefore, we use F1 score which captures both precision and recall. The formula to calculate it is:

$$F_1 = 2 \cdot \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

To calculate the F1 score we use scikit-learn implementation:

```
from sklearn.metrics import f1_score
f1_score(y_true, y_predicted)
```

GRADING

We provide you with **one test set** for which you have to compute predictions. We have partitioned this test set into two parts (of the same size) and use it to compute a *public* and a *private* score for each submission. You only receive feedback about your performance on the public part in the form of the public score, while the private leaderboard remains secret. The purpose of this division is to prevent overfitting to the public score. Your model should generalize well to the private part of the test set. When handing in the task, you need to select which of your submissions will get graded and provide a short description of your approach. This has to be done **individually by each member** of the team. We will then compare your selected submission to our baseline. This project task is graded with either **pass (6.0)**, **partial pass (4.0)** or **fail (2.0)**. To fully pass the project (grade: 6.0), you need to perform better than the baseline in both private and public score. If you only outperform the baseline in either the private or the public score, you will get a partial pass (grade: 4.0). In addition, for the pass/fail decision, we consider the code and the description of your

solution that you submitted. The following **non-binding** guidance provides you with an idea on what is expected to pass the project: If you hand in a properly-written description, your source code is runnable and reproduces your predictions, and your submission performs better than the baseline, you can expect to have passed the assignment.

⚠ Make sure that you properly hand in the task, otherwise you may obtain zero points for this task.

FREQUENTLY ASKED QUESTIONS

⦿ WHICH PROGRAMMING LANGUAGE AM I SUPPOSED TO USE? WHAT TOOLS AM I ALLOWED TO USE?

You are free to choose any programming language and use any software library. However, **we strongly encourage you to use Python**. You can use publicly available code, but you should specify the source as a comment in your code.

⦿ AM I ALLOWED TO USE MODELS THAT WERE NOT TAUGHT IN THE CLASS?

Yes. Nevertheless, the baselines were designed to be solvable based on the material taught in the class up to the second week of each task.

⦿ IN WHAT FORMAT SHOULD I SUBMIT THE CODE?

You can submit it as a single file (main.py, etc.; you can compress multiple files into a .zip) having max. size of 1 MB. If you submit a zip, please make sure to name your main file as *main.py* (possibly with other extension corresponding to your chosen programming language).

⦿ WILL YOU CHECK / RUN MY CODE?

We will check your code and compare it with other submissions. We also reserve the right to run your code. Please make sure that your code is runnable and your predictions are reproducible (fix the random seeds, etc.). Provide a readme if necessary (e.g., for installing additional libraries).

⦿ SHOULD I INCLUDE THE DATA IN THE SUBMISSION?

No. You can assume the data will be available under the path that you specify in your code.

⦿ CAN YOU HELP ME SOLVE THE TASK? CAN YOU GIVE ME A HINT?

As the tasks are a graded part of the class, **we cannot help you solve them**. However, feel free to ask general questions about the course material during or after the exercise sessions.

⦿ CAN YOU GIVE ME A DEADLINE EXTENSION?

⚠ We do not grant any deadline extensions!

⦿ CAN I POST ON PIAZZA AS SOON AS HAVE A QUESTION?

This is highly discouraged. Remember that collaboration with other teams is prohibited. Instead,

- Read the details of the task thoroughly.
- Review the frequently asked questions.
- If there is another team that solved the task, spend more time thinking.
- Discuss it with your team-mates.

⦿ WHEN WILL I RECEIVE THE PRIVATE SCORES? AND THE PROJECT GRADES?

We will publish the private scores, and corresponding grades before the exam the latest.