RELATION

338 Euston Road, Regents Place, London NW1 3BG   relationrx.com

Data Science Coding Challenge

Thank you for participating in this challenge. You have 24 hours to complete the three exercises below and return the results as a reply to the original email. Please use Python as your programming language. One of the exercises offers you the option to use a Jupyter notebook, but for the others, please do not.

We do not expect you to use the full 24 hours. The exercises were designed such that they can be completed in 4-5 hours.

Please use part of the extra time to make sure your solutions are correct and complete. We would like you to pay appropriate attention to the quality and maintainability of your code. This is especially important at Relation, where many may be working on the same repository.

Our evaluation criteria is different from exercise to exercise (detailed below), but in general we are looking for:

- Full correct solutions
- Clean code practices: someone other than yourself should be able to read your solution and easily figure out what you have done
- Good software engineering practices: create dependencies file with specific versions of packages used (eg. conda yml, pip requirements), docstrings and testing, and anything else you consider as being good practice

Exercise 1

Proteins are macromolecules that perform the majority of functions in the human body. Each protein is composed of a sequence of amino acids. For the purposes of this exercise a protein is a string of capital letters:

Protein 1: “MVRIWTTIMIVLILLLRIGPNKPSLSGRQAPAQAQTSDLVPSLFPL”
Protein 2: “MLSSGVETQPVPLDSSMSAVVQELYSELPVSVSRELHADPEPS”

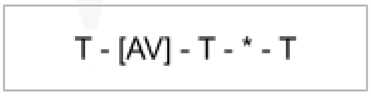
A protein structure team at Relation has identified an important pattern in the sequences of proteins. They ask you to find all human proteins that contain that pattern.

For this exercise you have been given task_1/human_proteins_dirty.csv which contains human proteins and their respective sequences in csv format.

The file has the following columns:

- protein_id: an identifier of the protein e.g. (P10234)
- sequence: the amino acid sequence of the protein as a string

The pattern (“Pattern X”) that the team has discovered is:



T - [AV] - T - * - T

Described in words, the pattern is a sequence formed of T, followed by either A or V, followed by another T, followed by any one character, followed by T

Tasks

1. The first thing you will notice is that the file is corrupt and does not work. There is 1 error in the file. Please identify it, fix it, and save it as a new file: task_1/human_proteins_clean.csv
2. After you have fixed the file please write code that will identify all sequences that contain Pattern X in task_1/human_proteins_clean.csv

Note: no specialised bioinformatics package may be used in this exercise.

Evaluation criteria

- Whether the solutions are correct
- Whether the code is clean and easily understandable to someone else

Exercise 2

The drug development team at Relation performed a screen of drug-like chemical compounds for a target of interest, the Spike protein of SARS-CoV-2. A screen, for the purpose of this exercise, is a filtering process where a large library (e.g. 100000) of chemical compounds is tested against the target protein, and only the ones that show activity against the target are retained.

The team has returned to you a list of 50 compounds, which are stored in task_2/spike_positive_compounds.csv, along with three features for each compound. The team did not tell you what the features mean.

The file has the following columns:

- drug_name: a name given to the drug
- feature_1: a list of labels that a drug can receive from the following list: {C, N, O, S}. Each drug can take 1 or more of those labels, and they are separated by a comma
- feature_2: a scalar value
- feature_3: a scalar value

You did not receive the initial library that they used. However, you do have the list of all the drugs approved in the world for other diseases in task_2/approved_drugs.csv. The drugs in this file also have the same three features reported.

Intuitively, you would not expect the vast majority of approved drugs to show activity against the SARS-CoV-2 Spike protein.

Task

1. Perform an exploratory data analysis of the screened compounds in task_2/spike_positive_compounds.csv and identify the features that you find significant for an active compound against the spike protein. You may use matplotlib, scipy etc. to help you analyse and summarise your findings.
2. Prepare a mini report for the team indicating what the significant features are and why you believe they are significant.

Note: you may use a Jupyter notebook to solve both problems together, or you may submit a script with an associated Word or PDF document.

Evaluation criteria

- Identifying and justifying significant features (2 features which award 50% each)
- A report that succinctly summarise the findings (e.g. plots/tests that demonstrate your findings)
- Clean code: someone else other than yourself will be able to read your code and figure out with ease what you have done
- Good software engineering practices. This includes, but is not limited to, dependencies file with specific version of packages used, code comments etc.

Exercise 3

The knowledge graph team at Relation is working hard on creating a prediction algorithm for whether two proteins interact with each other. Recent research indicates that we only know a small fraction of protein interactions. It is very hard to demonstrate the opposite, i.e. when two proteins do not interact with each other.

This creates a problem, because we only have positive examples. In order for the team to develop a robust prediction algorithm we need both positive and negative examples. The team says that if you can find a way to generate *soft negatives* that will help with their task.

A soft negative is a protein interaction that you believe can not actually happen based on a domain-specific heuristic. For example, you would not expect two proteins that are in different compartments of a cell to interact with each other. It's not necessarily true, but it's a better guess than random.

You are given 3 files:

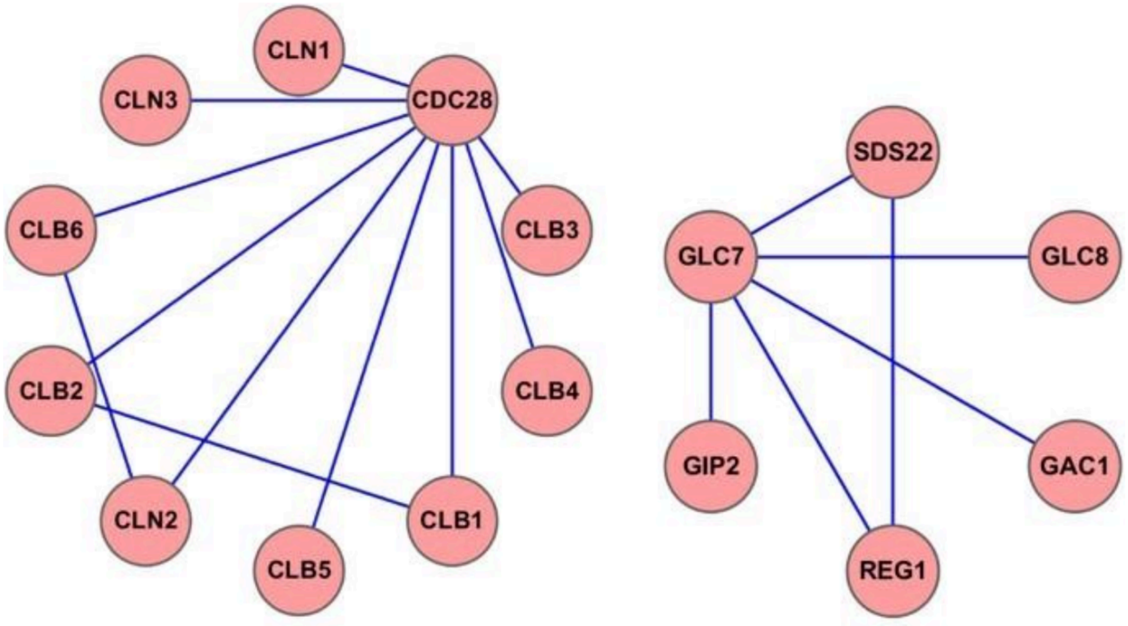
- proteins.txt: all the protein identifiers, one per line
- protein_interactions.txt: all the proteins that are already known to interact with each other, one interaction per line, i.e. protein_1 protein_2
- protein_compartments.csv: a csv file with the following columns:
    - protein_id: the id of the protein
    - compartment_id: a integer with the compartment identifier

Tasks

1. Identify all of the unique combinations of two proteins that do not reside in the same compartment, and also have not been previously identified to interact with each other
2. Identify all of the unique combinations of two proteins that are not in the same subgraph, and also do not reside in the same compartment

   Two proteins are said to be in the same subgraph if there is a path between them in the interaction graph.

   For example: In the image below every node is a protein, and an edge describes an interaction. CLB3 and CLN2 are considered to be on the same subgraph because there is a path from one to the other in the subgraph (CLB3 -> CDC28 -> CLN2, or CLB3 -> CDC28 -> CLB6 -> CLN2 etc.). GLC7 and CLB3, however, are not on the same subgraph because there is no path from one to another.



Note: no specialised graph package (e.g. networkx) is allowed for this exercise.

Evaluation criteria

1. Full correct solution to the tasks. (There is a common combinatorial pitfall to some approaches, where the right approach is used but the end result is wrong.)
2. Clean code: someone else other than yourself will be able to read the code and figure out with ease what you have done.
3. Good software engineering practices. This includes, but is not limited to, dependencies file with specific version of packages used (conda yml file or pip requirements file), docstrings, comments describing key areas of the approach, modularity (relevant functions and classes), and testing (use a simple framework and obtain good coverage).