Astex Bioinformatics Programming Test

This test contains a task we might expect a successful candidate to carry out in their day-to-day work.

# Basic Instructions

* Alongside these instructions, you should also have received a TSV file to use as part of the test
* Complete the task by writing Python code in one or more files
* Submit the test by replying with a zip/tgz file containing
  + The source code for your submission
  + Instructions on how to run your code
  + Any additional files that are needed, such as a list of dependencies
* The test should take you 1-2 hours – don’t spend much longer than this!

# The Data

The file `programming\_test\_data.tsv` contains the data you should use for this project. It is a truncated extract from some publicly available data made available by the OpenTargets project, which publishes a variety of genomic data that is useful in drug discovery. This file contains a list of genes along with information about their possible disease associations.

# The Project

The task is to create an HTTP API exposing the attached OpenTargets data. The API should provide functionality to query the data in several different ways. To achieve this you will need to:

* Read the tsv file into a suitable data structure
* Write functions or methods to extract and process the data for each endpoint
* Build the code to setup an HTTP API and expose different endpoints
* Connect your API endpoints to the analysis code you have written

## Endpoints

The API should include endpoints for the following:

* Get a list of all the unique gene names, as represented by the “approvedSymbol” column
  + Extension: allow the user to specify a therapeutic area code, perhaps as a URL argument. Return the list of only the genes that have diseases in that therapeutic area
* Get a mapping from gene name (“approvedSymbol”) to its ID (“id”).
* Get a mapping from the gene name (“approvedSymbol”) to a list of the associated diseases (as represented by the “name” column)
  + Extension 1: allow the user to specify a score, perhaps as a URL argument. Only report diseases that are above this score threshold
  + Extension 2: instead of just listing the associated disease names, map from the gene to a list of objects describing each association using the “datatypeId”, “score” and “name” columns e.g.

{ "BCAT1":

[

{

"datatypeId":"literature",

"name":"neoplasm",

"score":"0.8841359923841443"

},

…

],

…

}

## Documentation

Along with your source files, include a short description of what will be needed to run the code you have written. You can also include any comments you might want to make on your approach and what you would do next if you had more time.

# Some Hints

These hints are here to help you, but you should complete the task however works for you. If any of these hints don’t make sense, just ignore them and do it your own way!

* You might find it convenient to use a virtual environment to manage the dependencies for your code
* Use a web framework like Flask or FastAPI to build your API
  + If you aren’t familiar with these frameworks, just build the code to read and process the files
* Try to separate your code into suitable classes or modules, especially a separation between the code to perform the analysis and the code to expose the API endpoints
* You don’t have to tackle the endpoints in the order they are specified – by all means jump to the tasks you find easiest
* If you are familiar with the Python Pandas library, a Pandas dataframe is a good way to read and filter the TSV file. Otherwise, a list-of-lists or list-of-dicts will work fine to store the underlying data for the purposes of this test