### Open Targets Platform Bioinformatician Code Test

The goal of this test is to assess your ability to query a remote documented REST API, fetch and analyse the data, and test your code to specifications.

**Instructions:**

We would like you to build a program (in Python preferably) that can query the Open Targets REST API ([https://docs.targetvalidation.org/programmati c-access/rest-api](https://docs.targetvalidation.org/programmatic-access/rest-api)) to get a data value labelled as *association\_score.overall* for a given target or disease id. You are required to parse the output and print a simple analysis to the stdout.

1. Parse the arguments from command line, where:

*[your\_code] –t ENSG00000197386*

will run analysis for a target

*[your\_code] –d Orphanet\_399*

will run analysis for a disease

**ANSWER 1:**

For this part I developed opentarget\_query.py that accept two arguments from command line after –t you can use any other target name

Just in Terminal command line type:

python opentarget\_query.py -t ENSG00000197386

after –d you can use use any other disease id

python opentarget\_query.py -d Orphanet\_399

the application create a log file called: log\_opentarget\_query.txt to check the output carefully with date and time

as simple analysis I just filet the association score overall summary higher than 0.2 and “direct” association

opentargetclient library is used for implementation.

<https://opentargets.readthedocs.io/en/stable/tutorial.html>

1. Query the Open Targets associations data REST API endpoint (<https://platform-api.opentargets.io/v3/platform/public/association/filter>) to get target-disease association information. Use the parameter *target* to filter for association information for a target. i.e. use *ENSG00000197386* as target id
2. From the returned JSON objects, parse each entry (association) and print to stdout the combination of *target\_id, disease\_id & association\_score.overall*
3. Repeat steps 2-3 above to get disease related information using the parameter *disease* to query for association information for a disease i.e. use *Orphanet\_399* as disease\_id
4. At the end calculate and print to stdout the maximum, minimum and average and standard deviation values of *association\_score.overall* for target & disease query

**ANSWER 2, 3 , 4 and 5:**

For this part I developed endpoint\_query.py and it has no argument just to see each part, the system ask you to press enter.

Plus the stdout print , the log file “called log\_endpoint\_query.txt” will be created that save the tables in a text file for better evaluation the statistics summary which contain max , min , mean and standard deviation is calculated.

for this part there is no filtering and all association scores are shown.

For this part I used REST API with request library.

<https://docs.targetvalidation.org/programmatic-access/rest-api>

Please provide instructions to run your code in a unix-like machine.

You are welcome to use any external library, as long as you provide clear instruction to install it. Reinventing the wheel is not valued. Working code, proper testing, useful documentation and code reusability are highly valued.

*last\_updated: 4-August-2020*