
icgc Documentation

Release 0.0.1

The Ontario Institute For Cancer Research

Mar 16, 2018

Contents

1	Installation	3
2	Contribute	5

The ICGC REST client is a simple python module that allows you to access the **International Consortium for Cancer and Genomics** web portal (<<https://dcc.icgc.org/>>) directly through Python, with a minimum of coding effort.

It lets you write queries in our Portal Query Language (**PQL**) that fetch data from the ICGC web portal as JSON objects. From there, you can use the power of Python to process and analyze the data within those objects however you see fit.

Here's an example that shows you how easy it is to get started!

```
"""
query.py

This script demonstrates running a simple PQL query against the ICGC data
portal with the icgc module.
"""
from __future__ import absolute_import, print_function

import icgc

def run():
    """
    Demonstrate PQL by displaying 1 of each request type as JSON output
    """
    for request_type in icgc.request_types():
        response = icgc.query(request_type=request_type,
                              pql='select(*),limit(1)')
        print(request_type, "===\n\n", response)

if __name__ == '__main__':
    run()
```

Here's an a simple program that demonstrates how Python can be used with the icgc Python module to automate decision making: in this case, which files we want to download from the ICGC web portal.

```
from __future__ import absolute_import, print_function
import icgc

KB = 1024
MB = 1024 * KB

def run():
    """
    Show an example of a PQL download with automated decision making.

    We download up to a maximum of 10 MB of data from the portal, of any type
    that will fit within our download limit, and save our the results as a
    tarfile named 'test.tar'.
    """
    pql = 'eq(donor.primarySite,"Brain")'

    # Find which items are available that match our pql query, and how big
    # each of the result file are.

    sizes = icgc.download_size(pql)
    print("Sizes are: {}".format(sizes))
```

```
# We'll only include a file in our tarfile if the total is below our
# 10 MB limit. Our tarfile size calculation is approximate; the
# files inside the tarfile get compressed; so the total size of the tarfile
# that we download might be smaller than we calculate.

max_size = 10 * MB
current_size = 0

includes = []
for k in sizes:
    item_size = sizes[k]
    if current_size + item_size < max_size:
        includes.append(k)
        current_size += item_size

print("Including items {}".format(includes))
print("Approximate download size={:.2f} MB".format(current_size / MB))

# Download the information, and save the results in the file "test.tar"
icgc.download(pql, includes, "test")

if __name__ == "__main__":
    run()
```

CHAPTER 1

Installation

You can install icgc using *pip* by running: `pip install icgc`

If you prefer, you can also download the source code from the url below.

CHAPTER 2

Contribute

If you'd like to contribute to this project, it's hosted on github.

See <https://github.com/icgc-dcc/icgc-python>