

# Tuto 1.1 Discovering Open Data

November 17, 2020

## 1 Gravitational Wave Open Data Workshop #3

**Tutorial 1.1: Discovering open data from GW observatories** This notebook describes how to discover what data are available from the [Gravitational-Wave Open Science Center \(GWOSC\)](#).

[Click this link to view this tutorial in Google Colaboratory](#)

### 1.1 Software installation (execute only if running on a cloud platform or haven't done the installation yet!)

First, we need to install the software, which we do following the instruction in [Software Setup Instructions](#):

```
[1]: # -- Uncomment following line if running in Google Colab
     #! pip install -q 'gwosc==0.5.3'
```

**Important:** With Google Colab, you may need to restart the runtime after running the cell above.

```
[1]: #check the version of the package gwosc you are using
import gwosc
print(gwosc.__version__)
```

0.5.4

The version you get should be 0.5.3. If it's not, check that you have followed all the steps in [Software Setup Instructions](#).

### 1.2 Querying for event information

The module `gwosc.catalog` provides tools to search for events in a catalog.

The module `gwosc.datasets` provides tools for searching for datasets, including full run strain data releases.

For example, we can search for events in the [GWTC-1 catalog](#), the catalog of all events from the O1 and O2 observing runs. A list of available catalogs can be seen in the [Event Portal](#)

```
[2]: from gwosc.datasets import find_datasets
     from gwosc import catalog
```

```

##-- Print all the GW events from the GWTC-1 catalog
gwtc1 = catalog.events('GWTC-1-confident')
print('GWTC-1 events:', gwtc1)
print("")

##-- Print all the large strain data sets from LIGO/Virgo observing runs
runs = find_datasets(type='run')
print('Large data sets:', runs)

```

GWTC-1 events: ['GW150914', 'GW151012', 'GW151226', 'GW170104', 'GW170608', 'GW170729', 'GW170809', 'GW170814', 'GW170817', 'GW170818', 'GW170823']

Large data sets: ['BKGW170608\_16KHZ\_R1', 'O1', 'O1\_16KHZ', 'O2\_16KHZ\_R1', 'O2\_4KHZ\_R1', 'S5', 'S6']

catalog.events also accepts a segment and detector keyword to narrow results based on GPS time and detector:

```

[3]: ##-- Detector and segments keywords limit search result
print(catalog.events('GWTC-1-confident', detector="L1", segment=(1164556817,
↪1187733618)))

```

['GW170104', 'GW170608', 'GW170729', 'GW170809', 'GW170814', 'GW170817', 'GW170818', 'GW170823']

Using gwosc.datasets.event\_gps, we can query for the GPS time of a specific event:

```

[4]: from gwosc.datasets import event_gps
gps = event_gps('GW190425')
print(gps)

```

1240215503.0

All of these times are returned in the GPS time system, which counts the number of seconds that have elapsed since the start of the GPS epoch at midnight (00:00) on January 6th 1980. GWOSC provides a GPS time converter you can use to translate into datetime, or you can use gwpy.time.

We can query for the GPS time interval for an observing run:

```

[5]: from gwosc.datasets import run_segment
print(run_segment('O1'))

```

(1126051217, 1137254417)

To see only the confident events in O1:

```

[6]: O1_events = catalog.events('GWTC-1-confident', segment=run_segment('O1'))
print(O1_events)

```

['GW150914', 'GW151012', 'GW151226']

### 1.3 Querying for data files

The `gwosc.locate` module provides a function to find the URLs of data files associated with a given dataset.

For event datasets, one can get the list of URLs using only the event name:

```
[7]: from gwosc.locate import get_event_urls
     urls = get_event_urls('GW150914')
     print(urls)
```

```
['https://www.gw-openscience.org/eventapi/json/GWTC-1-confident/GW150914/v3/H-H1_GWOSC_4KHZ_R1-1126259447-32.hdf5', 'https://www.gw-openscience.org/eventapi/json/GWTC-1-confident/GW150914/v3/H-H1_GWOSC_4KHZ_R1-1126257415-4096.hdf5', 'https://www.gw-openscience.org/eventapi/json/GWTC-1-confident/GW150914/v3/L-L1_GWOSC_4KHZ_R1-1126259447-32.hdf5', 'https://www.gw-openscience.org/eventapi/json/GWTC-1-confident/GW150914/v3/L-L1_GWOSC_4KHZ_R1-1126257415-4096.hdf5']
```

By default, this function returns all of the files associated with a given event, which isn't particularly helpful. However, we can filter on any of these by using keyword arguments, for example to get the URL for the 32-second file for the LIGO-Livingston detector:

```
[8]: urls = get_event_urls('GW150914', duration=32, detector='L1')
     print(urls)
```

```
['https://www.gw-openscience.org/eventapi/json/GWTC-1-confident/GW150914/v3/L-L1_GWOSC_4KHZ_R1-1126259447-32.hdf5']
```

## 2 Exercises

Now that you've seen examples of how to query for dataset information using the `gwosc` package, please try and complete the following exercises using that interface:

- How many months did S6 last?
- How many GWTC-1-confident events were detected during O1?
- What file URL contains data for V1 4096 seconds around GW170817?

```
[9]: from gwosc.datasets import run_segment
     print(run_segment('S6'))
```

```
(931035615, 971622015)
```

```
[12]: (971622015 - 931035615)/(3600*24*30)
```

```
[12]: 15.658333333333333
```

```
[17]: from gwpy.time import tconvert
     t_start = tconvert(931035615)
     t_stop = tconvert(971622015)
```

```
print('t_start =',t_start)
print('t_stop =',t_stop)
```

```
t_start = 2009-07-07 21:00:00
t_stop = 2010-10-20 15:00:00
```

```
[33]: import datetime
      from dateutil.relativedelta import relativedelta

      duration = relativedelta(t_stop, t_start)

      print ('The S6 run lasted for', duration.years, 'years', duration.months,
            ↳'months', duration.days, 'days and', duration.hours, 'hours')
```

The S6 run lasted for 1 years 3 months 12 days and 18 hours

```
[36]: from gwosc.datasets import run_segment
      01_events = catalog.events('GWTC-1-confident', segment=run_segment('01'))
      print(01_events)
      l01=len(01_events)
      print('There are', l01, 'GWTC-1-confident events during the 01 run')
```

```
['GW150914', 'GW151012', 'GW151226']
```

There are 3 GWTC-1-confident events during the 01 run

```
[37]: from gwosc.datasets import run_segment
      S6_events = catalog.events('GWTC-1-confident', segment=run_segment('S6'))
      print(S6_events)
      lS6=len(S6_events)
      print('There are', lS6, 'GWTC-1-confident events during the S6 run')
```

```
[]
```

There are 0 GWTC-1-confident events during the S6 run

```
[39]: from gwosc.locate import get_event_urls
      urls_GW170817 = get_event_urls('GW170817')
      #print(urls_GW170817)
      urls_GW170817_V1_4096 = get_event_urls('GW170817', duration=4096, detector='V1')
      print('The URL containing the data for V1 for 4096 seconds around GW170817_
            ↳is',urls_GW170817_V1_4096)
```

The URL containing the data for V1 for 4096 seconds around GW170817 is

```
['https://www.gw-openscience.org/eventapi/json/GWTC-1-confident/GW170817/v3/V-V1_
  _GWOSC_4KHZ_R1-1187006835-4096.hdf5']
```

```
[ ]:
```