GDCtools Overview

Version 2017_03_02.1

gdctools

Python and UNIX CLI utilities to simplify interaction with the NIH/NCI Genomics Data Commons.

https://github.com/broadinstitute/gdctools

The Genomics Data Commons (GDC) is the next-generation storage warehouse for genomic data. It was inspired by lessons learned and technologies developed during The Cancer Genome Atlas project (TCGA), in the hope of extending them to a wide range of future genomics projects funded through the National Cancer Institute (NCI) of the National Institutes of Health (NIH).

This GDCtools package is the offshoot of efforts at the Broad Institute to connect the Firehose pipeline developed in TCGA to use the GDC as its primary source of data. The ultimate goal of this package, though, goes beyond simply connecting Firehose to the GDC: we aim to provide a set of Python bindings and UNIX cli wrappers to the GDC application programming interface (API) that are vastly simpler to use for the majority of common operations.

- Aim: enable one to quickly use & program against GDC
- Mirror data, aggregate, freeze, sample reports … and more
- Begin in just minutes, no need to hire/train staff

```
linux%  git clone https://github.com/broadinstitute/gdctools
linux%  make
```

- Or learn virtually any of the GDC API

- It's well understood that BIG part of data-driven science is
- Aggregation, cleansing, counting & tracking
- Esp for consortium-scale datasets
- Such as TCGA: 33 cohorts, 11.5K patients, 85K data aliquots
- Firehose performed this democratizing service in TCGA

  In ~5K lines of Python, BUT *internal/monolithic (not open)*

- It's well understood that BIG part of data-driven science is
- Aggregation, cleansing, counting & tracking
- Esp for consortium-scale datasets
- Such as TCGA: 33 cohorts, 11.5K patients, 85K data aliquots
- Firehose performed this democratizing service in TCGA

  In ~5K lines of Python, BUT *internal/monolithic (not open)*

GDCtools aims to generalize this, to all data at GDC

And make it open-source for everyone

- It's well understood that BIG part of data-driven science is
- Aggregation, cleansing, counting & tracking
- Esp for consortium-scale datasets
- Such as TCGA: 33 cohorts, 11.5K patients, 85K data aliquots
- Firehose performed this democratizing service in TCGA

  In ~5K lines of Python, BUT *internal/monolithic (not open)*

GDCtools aims to generalize this, to all data at GDC

And make it open-source for everyone

Largely replaced by GDCtools + 4 lines BASH

```
gdc_mirror      -config tcga.cfg
gdc_dice        -config tcga.cfg
sample_report   -config tcga.cfg
create_loadfile -config tcga.cfg
```

This is essentially our nightly cron job

- Easily download & process ***all*** or ***subset(s)***
- Highly configurable: even to just 1 case
- Example: to mirror TARGET

```
gdc_mirror     –config target.cfg
```

- In principle: easily mix/match GDAN programs
    - Example: combine TARGET, TCGA
    - Little or zero coding
    - Just config file

- Example: to put in Google cloud buckets

```
create_loadfile –config tcga.cfg,google.cfg
```

- This is how we specify loads to FireCloud

- Example: to put in Google cloud buckets

```
create_loadfile –config tcga.cfg,google.cfg
```

- This is how we specify loads to FireCloud

- Minimalist configuration, obeys *union semantics*

```
[loadfiles]
DIR: %(ROOT_DIR)s/loadfiles/google
FILE_PREFIX: gs://broad-institute-gdac/gdc/dice
FORMAT: firecloud
```

Entire content of `google.cfg`
Simply replaces `[loadfiles]` directive from `tcga.cfg`

- Simple object framework, easy to extend & maintain
- Easy / familiar UNIX look-n-feel for computationalists

```
%  gdc_ls programs
[
  "TCGA",
  "TARGET"
]
```

What programs have **exposed** data for download?

- Simple object framework, easy to extend & maintain
- Easy / familiar UNIX look-n-feel for computationalists

```
%  gdc_ls programs
[
  "TCGA",
  "TARGET"
]
```

What programs have **exposed** data for download?

What programs have **submitted** data?

```
%  gdc_ls submission
[
  "CCLE",
  "REBC",
  "TCGA",
  "TARGET",
  "CGCI",
  "CDDP",
  "ALCHEMIST",
  "GDC",
  "Exceptional_Responders",
  "UAT08",
  “TRIO”,
  “CPTAC”
]
```

Recently submitted

Auto-generated Python bindings: coming …