Tutorial
Version 2016_04_17

gdac@broadinstitute.org
Introduction
A simple and elegant way to explore cancer data.

Backed by a powerful computational infrastructure, application programming interface (API), graphical tools and online reports.

Sitting above one of the deepest and most integratively-characterized open cancer datasets in the world.

With over 80K sample aliquots from 11,000+ cancer patients, spanning 38 unique disease cohorts.
Data and analyses utilized at numerous academic, research, and commercial sites around the world.

Example: cBio@MSKCC

TCGA data & analyses in cBioPortal—expression, mutation, copy number, significance analyses, and more—are loaded directly from Firehose.
Choose Cohort

~1500 Analyses (reports) per run
Find your favorite in 2 clicks

DataType

Inspect

SNP6 Copy number analysis (GISTIC2)
Breast Invasive Carcinoma (Primary solid tumor)
15 July 2014

Overview
Introduction
Summary

Results
Focal results
Arm-level results

Methods & Data

Copyright © 2014 Broad Institute TCGA GDAC as part of the TCGA Research Network. All rights reserved.
Many 1000s of datasets per run
Find your favorite in 2 clicks

Choose Cohort

Then DataType

Click to download
Or easily send to GenomeSpace for more analysis
Or download everything with 1 command

```
linux% firehose_get analyses latest
```

Simple 20K bash script, just 1 moving part

[Download Here]
Graphical Tools
Quickly inspect mRNASEq expression levels for a selected gene
View expression levels across all cohorts, or arbitrary subsets.
CoMut: mutation co-occurrence plots

Introduced in 2011 (Stransky et al, Science, 2011), CoMut figures have become common in TCGA research. Within a single graphic they provide a comprehensive analysis profile, enabling the reader to quickly infer relationships between co-occurring results across multiple data modalities, across common X axis of sample IDs.
But in journals, figures are static and can be small and hard to read.

And cannot be explored in real time.

And reproducing them or investigating their implications can require substantial time for data retrieval, preparation and analysis.
**Mutation significance**

**Mutation rate**

**Clusters**

**CN gain**

**CN loss**

**Clinical parameters**

---

**iCoMut**

By making such figures interactive, allowing panels to be moved, sorted and searched, iCoMut dramatically enhances that process.

Example: hovering over pixels tells you about the underlying biology.

Here we show the TCGA LGG cohort: sorted first by clinical histology, then gene (descending order of mutation count). The clinical subtypes leap off the page at you.

As does the fact that the copy-number landscape differs when IDH1/2, TP53, and ATRX mutations drop off.
Now we’ve re-sorted by CNMF copy-number clustering, and dragged it from bottom of figure to top, just above mutation panel.

Making it further apparent that the copy-number landscape differs as IDH1/2, TP53, and ATRX mutations diminish.

Also shows apparent involvement with EGFR and PTEN.
iCoMut compresses an entire Firehose run into a single, interactive & reproducible figure

Turning this …

Firehose analysis workflow
Run on 38 TCGA cohorts
> 100 tasks per run
~10 datatypes
Distills 50 TB of input data into 10GB of results (5000x)
... into this

[Image: firebrowse.org/iCoMut/?cohort=UVM]
By default, patients are sorted by histology and gene mutation.
Click on the text labels to change sorting.
Sort status of samples is reported in the info box.
iCoMut results for ACC - Adrenocortical carcinoma

Generated on Wed Nov 04 2015 1:34:46 PM with integ-2015_10_29_c | 9851a395fb32

View on Firebrowse at: http://fbdev/iCoMut/?cohort=acc

Disease Type
Date & Software Version

URL to regenerate (will reflect all interactive manipulations to figure)
Many more graphical controls …

Example: locate patient/sample of interest

Collaboratively explore questions in realtime on telecons:
in what expression cluster does patient X fall?

Without database lookup or scripting, etc
Click on to collapse a panel

- Mutation Rate
  - synonymous
  - non synonymous

- Clinical Age
- Clinical Vital Status
- Clinical Gender
- Clinical Histology
- Clinical Ethnicity

- age
- vital status
- gender
- histology
- race
Drag and drop the ⊐ or ⊑ icon to rearrange the panels.

Rearranged panels
Programmatic Tools
API-Powered: 25+ RESTful APIs in 4 categories

**WEB API**

### Analyses
Fine grained retrieval of analysis pipeline results

- **GET** /Analyses/Mutation/MAF
- **GET** /Analyses/Mutation/SMG
- **GET** /Analyses/CopyNumber/Genes/All
- **GET** /Analyses/CopyNumber/Genes/Focal
- **GET** /Analyses/CopyNumber/Genes/Thresholded
- **GET** /Analyses/CopyNumber/Genes/Amplified
- **GET** /Analyses/CopyNumber/Genes/Deleted
- **GET** /Analyses/Reports
- **GET** /Analyses/Summary

### Samples
Fine grained retrieval of sample-level data

- **GET** /Samples/mRNAseq
- **GET** /Samples/miRNAseq
- **GET** /Samples/ClinicalTier1

### Archives
Bulk retrieval of data or analysis pipeline results, as compressed archives

- **GET** /Archives/StandardData

### Metadata
Retrieve disease, sample, and datatype descriptions, sample counts, and more

- **GET** /Metadata/Counts
- **GET** /Metadata/Cohorts
- **GET** /Metadata/Cohort/{cohort}
- **GET** /Metadata/Platforms
Interactive Docs

**Implementing Notes**
This service returns sample-level log2 mRNASeq expression values. Results may be filtered by gene, cohort, barcode, sample type or characterization protocol, but at least one gene OR barcode must be supplied.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
<th>Parameter Type</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>format</td>
<td>json (default)</td>
<td>Format of result.</td>
<td>query</td>
<td>string</td>
</tr>
<tr>
<td>gene</td>
<td>egfr</td>
<td>Comma separated list of gene name(s).</td>
<td>query</td>
<td>string</td>
</tr>
<tr>
<td>cohort</td>
<td>ACC, BLCA, BRCA, CESC</td>
<td>Narrow search to one or more TCGA disease cohorts from the scrollable list.</td>
<td>query</td>
<td>string</td>
</tr>
<tr>
<td>tcca_participant_barcode</td>
<td></td>
<td>Comma separated list of TCGA participant barcodes (e.g. TCGA-GFA4EO).</td>
<td>query</td>
<td>string</td>
</tr>
<tr>
<td>sample_type</td>
<td>NB, NT, TAM, TAP</td>
<td>Narrow search to one or more TCGA sample types from the scrollable list.</td>
<td>query</td>
<td>string</td>
</tr>
<tr>
<td>protocol</td>
<td>RPKM, RSEM</td>
<td>Narrow search to one or more sample characterization protocols from the scrollable list.</td>
<td>query</td>
<td>string</td>
</tr>
</tbody>
</table>

*learn APIs and explore data by playing in real time instead of cut/paste from static HTML or PDF*

*automatically generated & updated as API and database evolve*

*choices clearly enumerated*
Proper RESTful call is ASSEMBLED FOR YOU

Request URL

http://firebrowse.org:8000/api/v1/Samples/mRNASEq?format=json&gene=egfr&page=1&page_size=250&sort_by=gene

Results returned in multiple formats

JSON for computers/programmers

TSV, CSV for scientists, algorithms
Even Easier in Python, R, and UNIX

fbget

- Low-level Python bindings: 1-1 with RESTful api
- Higher-level interface, for easy/common bioinformatics
- UNIX command line interface, too
- Automatically generated, easily synched with RESTful API
- Flexible, copiously documented and tested
- BSD-style open source license

Download

FireBrowseR: bindings for R

https://github.com/mariodeng/FirebrowseR
fbget: low level interface

4 classes, one per API category:
Samples, Analyses, Archives, Metadata

N methods per class, matching RESTful API; each defaults to returning 1 page, in JSON

```python
import firebrowse
print firebrowse

Samples() mRNASEq(gene="egfr", cohort="ucs")
```

```json
"mRNASEq": [
  {
    "cohort": "UCS",
    "expression_log2": 7.06162500904694,
    "gene": "EGFR",
    "geneID": 1956,
    "protocol": "RSEM",
    "sample_type": "TP",
    "tcga_participant_barcode": "TCGA-QN-A5NN",
    "z-score": -0.598993525060403
  },
  ...
]
```
fbget : high level interface

```python
python> import fbget
python> print fbget.mrnaseq("egfr", cohort="ucs")
```

<table>
<thead>
<tr>
<th>tcga_participant_barcode</th>
<th>gene</th>
<th>expression_log2</th>
<th>z-score</th>
<th>cohort</th>
<th>cohort</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCGA-QN-A5NN</td>
<td>EGFR</td>
<td>7.06162500905</td>
<td>-0.59899352506</td>
<td>UCS</td>
<td>TP</td>
</tr>
<tr>
<td>TCGA-QM-A5NM</td>
<td>EGFR</td>
<td>8.16734387649</td>
<td>-0.298443593752</td>
<td>UCS</td>
<td>TP</td>
</tr>
<tr>
<td>TCGA-NG-A4VW</td>
<td>EGFR</td>
<td>8.93092623547</td>
<td>0.0932667888031</td>
<td>UCS</td>
<td>TP</td>
</tr>
</tbody>
</table>

- Simpler, e.g. objects do not need to be instantiated
- Intuitive defaults for common bioinformatic use cases
- Transparently iterates:
  - ✓ To retrieve all pages of results in 1 call
  - ✓ In TSV format
fbget : UNIX CLI interface

Because sometimes even writing just a couple of lines of Python takes too long

<table>
<thead>
<tr>
<th>tcga_participant_barcode</th>
<th>gene</th>
<th>expression_log2</th>
<th>z-score</th>
<th>cohort</th>
<th>cohort_type</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCGA-QN-A5NN</td>
<td>EGFR</td>
<td>7.06162500905</td>
<td>-0.59899352506</td>
<td>UCS</td>
<td>TP</td>
</tr>
<tr>
<td>TCGA-QM-A5NM</td>
<td>EGFR</td>
<td>8.16734387649</td>
<td>-0.298443593752</td>
<td>UCS</td>
<td>TP</td>
</tr>
<tr>
<td>TCGA-NG-A4VW</td>
<td>EGFR</td>
<td>8.93092623547</td>
<td>0.0932667888031</td>
<td>UCS</td>
<td>TP</td>
</tr>
</tbody>
</table>
Example: quickly list patients

All of TCGA

```
linux% fbget patients
tcga_participant_barcode    date    cohort
TCGA-PK-A5H9    2015-04-02 00:00:00 ACC
TCGA-PA-A5YG    2015-04-02 00:00:00 ACC
TCGA-OR-A5JD    2015-04-02 00:00:00 ACC
TCGA-P6-A5OF    2015-04-02 00:00:00 ACC
TCGA-P6-A5OG    2015-04-02 00:00:00 ACC
```

Or just GBM

```
linux% fbget patients cohort=gbm
tcga_participant_barcode    date    cohort
TCGA-19-4065    2015-04-02 00:00:00 GBM
TCGA-81-5911    2015-04-02 00:00:00 GBM
TCGA-81-5910    2015-04-02 00:00:00 GBM
TCGA-12-1089    2015-04-02 00:00:00 GBM
```

This can be enhanced to yield platform data matrix, like AWG freeze list
Better than an inscrutable stack trace, don't you think?

Docs for almost all class methods and functions can also be obtained by invoking the function with zero arguments.

```python
python> fbget.mrnaseq()

mrnaseq() call has missing/None arg value(s), need at least one of: gene OR barcode
Help on function mrnaseq in module fbget:

mrnaseq(gene=None, barcode=None, **kwargs)

    High level wrapper for the FireBrowse Samples.mRNASeq method.
    By default it returns ALL pages of data, in TSV format.

```
Same is true on UNIX command line

```
linux% fbget mrnaseq

mrnaseq() call has missing/None arg value(s), need at least one of: gene OR barcode
Help on function mrnaseq in module firebrowse.fbget:

mrnaseq(gene=None, barcode=None, **kwargs)
    High level wrapper for the FireBrowse Samples.mRNASeq method.
    By default it returns ALL pages of data, in TSV format.
    
    This service returns sample-level log2 mRNASeq expression
    values. Results may be filtered by gene, cohort, barcode,
    sample type or characterization protocol, but at least one
    gene OR barcode must be supplied.
    
    For more details consult the interactive documentation at
    http://firebrowse.org/api-docs/#!/Samples
    OR use help(param_values) to see the range of values accepted
    for each parameter, the defaults for each (if any), and the
    degrees of optionality/requiredness offered by the API.
    
    Parameters:
    format (str) Format of result.
    gene (str) Comma separated list of gene name(s).
    cohort (str) Narrow search to one or more TCGA disease cohorts.
    barcode (str) Comma separated list of TCGA participant barcodes (e.g. TCGA-GF-A4E0).
    sample_type (str) Narrow search to one or more TCGA sample types.
    protocol (str) Narrow search to one or more sample characterization protocols.
    page (int) Which page (slice) of entire results set should be returned.
    page_size (int) Number of records per page of results. Maximum is 2000.
    sort_by (str) Which column in the results should be used for sorting paginated results?
```

Docs obtained by invoking functions with zero arguments
Examples Embedded Directly in Tool

```
linux% fbget --examples

# Every line of these examples can be cut and directly pasted to your
# UNIX-like command line. Comments will be ignored, while everything
# not beginning with the # comment character will be executed, as long
# as fbget is in your $PATH

# Get the RNASeq expression level of the POLE gene, for all TCGA samples
# (both tumors and normals, in RSEM form, saved to file)
fbget --outfile=fbget-test-pole.tsv mrnaseq pole

# Similar query, but constrained to just the DLBC disease cohort
fbget mrnaseq pole cohort=dlbc

# Now constrained to single patient, and showing case insensitivity
fbget mrnaseq p0LE baRc0dE=TCGA-RQ-A6JB

# What is the DLBC cohort, anyway?
fbget cohort dlbc
# DLBC Lymphoid Neoplasm Diffuse Large B-cell Lymphoma

# List all the disease cohorts offered by FireBrowse (note that aggregate
# cohorts like COADREAD,KIPAN,GBMLGG,STES are not available at the TCGA DCC)
fbget cohorts

# Display help (docstring) for the function which retrieves clinical data
fbget help clinical
```
Fin